

**DEPARTMENT OF INFORMATION TECHNOLOGY
CS8092 Computer Graphics and Multimedia**

**UNIT I
ILLUMINATION AND COLOR MODELS**

1. What is a shading model?

It attempts to model how light that emanates from light sources would interact with objects in a scene. Usually it does not try to simulate all of the physical principles having to do with the scattering and reflection of light.

2. What are the two types of light sources that illuminate the objects in a scene ?

The two light sources are point light source and ambient light.

3. What are the different ways by which the incident light interacts with the surface?

The three different ways are: some is absorbed by the surface and is converted to heat, some is reflected from the surface and some is transmitted into the interior of the object, as in the case of a piece of glass.

4. What are the two types of reflection of incident light?

The two types are diffuse scattering and specular reflections.

5. What are the three vectors to be found, to compute the diffuse and specular components of light?

The three vectors are: the normal vector m to the surface at P , the vector v from P to the viewer's eye, the vector s from P to the light source.

6. What is specular reflection?

Real objects do not scatter light uniformly in all directions, so a specular component is added to the shading model. Specular reflection causes highlights, which can add significantly to the realism of a picture when objects are shiny.

7. Define Diffuse Scattering.

Diffuse scattering occurs when some of the incident light slightly penetrates the surface and is re-radiated uniformly in all directions. Scattered light interacts strongly with the surface and so its color is usually affected by the nature of the surface material.

8. State Lambert's law .

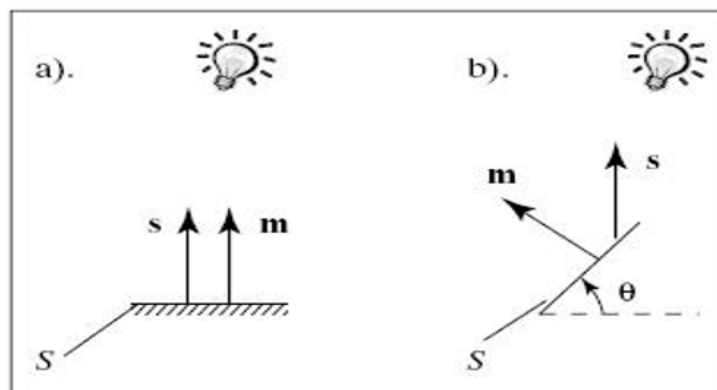


Fig (a) shows the cross section of a point source illuminating a face S when m is aligned with s. Fig (b) the face is turned partially away from the light source through angle θ . The area subtended is now only $\cos(\theta)$, so that the brightness of S is reduced by this same factor. This relationship between the brightness and surface orientation is called **Lambert's law**.

9. What is Output Primitive?

Basic geometric structures that describe a scene are referred to as Output Primitives. Points and straight line segments are the simplest geometric components of pictures. Additional output primitives that can be used to construct a picture include circles and other conic sections, quadric surfaces, spline curves and surfaces, polygon color areas, and character strings.

10. What is DDA?

The Digital Differential Analyzer is a scan-conversion line algorithm based on calculating either difference in y-coordinate (dy) or difference in x-coordinate. We sample the line at unit intervals in one coordinate and determine corresponding integer values nearest the line path for the other coordinate.

11. What are the disadvantages of DDA algorithm?

- Round-off error in successive additions of the floating-point increment can cause the calculated pixel positions to drift away from the true line path for long line segments.
- Rounding operations and floating-point arithmetic in procedure are still time-consuming.

12. What is attribute parameter?

Any parameter that affects the way a primitive is to be displayed is referred to as an attribute parameter.

13. What are the basic line attributes?

Basic attributes of a straight line segment are its type, its width, and its color.

14. What is meant by aliasing?

The distortion of information due to low frequency sampling (Under sampling) is called aliasing. We can improve the appearance of displayed raster lines by applying antialiasing methods that compensate for the under sampling process.

15. Derive the decision parameter for bresenhams line drawing algorithm

Define:

$$d1 = y - y_i = m(x_i + 1) + b - y_i$$

$$d2 = (y_i + 1) - y = y_i + 1 - m(x_i + 1) - b$$

Calculate:

$$\begin{aligned} d1 - d2 &= m(x_i + 1) + b - y_i - y_i - 1 + m(x_i + 1) + b \\ &= 2m(x_i + 1) - 2y_i + 2b - 1 \end{aligned}$$

$$\text{if } d1 - d2 < 0 \text{ then } y_{i+1} \leftarrow y_i$$

$$\text{if } d1 - d2 > 0 \text{ then } y_{i+1} \leftarrow y_i + 1$$

16. Give the final coordinates of a unit square ABCD with vertex A placed o the origin after applying a uniform scaling of 2 units with respect to the centre of the square.

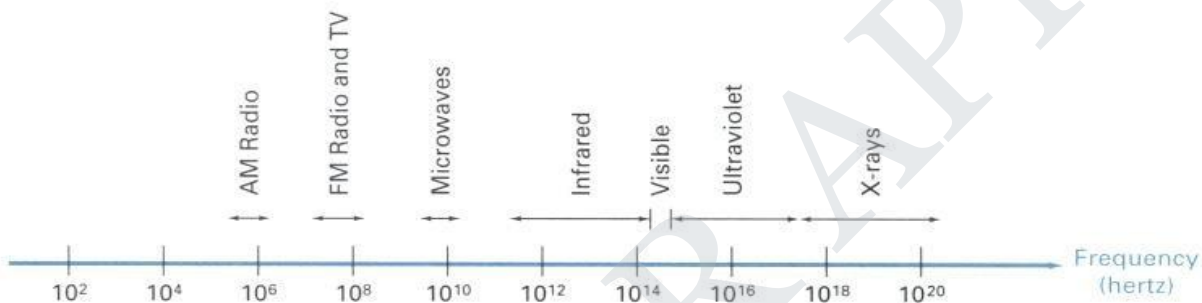
(0,0) (2,0) (2,2) (0,2)

PART B

1. Explain in detail about properties of light and basic illumination models.

Properties of Light

- Light is a narrow frequency band within the electromagnetic system.
- Other frequency bands within this spectrum are called radio waves, micro waves, infrared waves and x-rays. The below fig shows the frequency ranges for some of the electromagnetic bands.



- Each frequency value within the visible band corresponds to a distinct color.
- At the low frequency end is a red color (4.3×10^{14} Hz) and the highest frequency is a violet color (7.5×10^{14} Hz)
- Spectral colors range from the reds through orange and yellow at the low frequency end to greens, blues and violet at the high end.
- Since light is an electro magnetic wave, the various colors are described in terms of either the frequency for the wave length λ of the wave.
- The wave length and frequency of the monochromatic wave are inversely proportional to each other, with the proportionality constants as the speed of light C where $C = \lambda f$

- A light source such as the sun or a light bulb emits all frequencies within the visible range to produce white light. When white light is incident upon an object, some frequencies are reflected and some are absorbed by the object. The combination of frequencies present in the reflected light determines what we perceive as the color of the object.
- If low frequencies are predominant in the reflected light, the object is described as red. In this case, the perceived light has the dominant frequency at the red end of the spectrum. The dominant frequency is also called the hue, or simply the color of the light.
- Brightness is another property, which is the perceived intensity of the light.
- Intensity is the radiant energy emitted per unit time, per unit solid angle, and per unit projected area of the source.
- Radiant energy is related to the luminance of the source. •

The next property is the purity or saturation of the light.

- Purity describes how washed out or how pure the color of the light appears.
- Pastels and Pale colors are described as less pure.
- The term chromaticity is used to refer collectively to the two properties, purity and dominant frequency.
- Two different color light sources with suitably chosen intensities can be used to produce a range of other colors.
- If the 2 color sources combine to produce white light, they are called complementary colors. E.g., Red and Cyan, green and magenta, and blue and yellow.
- Color models that are used to describe combinations of light in terms of dominant frequency use 3 colors to obtain a wide range of colors, called the color gamut.
- The 2 or 3 colors used to produce other colors in a color model are called primary colors.

Intuitive Color Concepts

- Color paintings can be created by mixing color pigments with white and black pigments to form the various shades, tints and tones.
- Starting with the pigment for a „pure color“ the color is added to black pigment to produce different shades. The more black pigment produces darker shades.
- Different tints of the color are obtained by adding a white pigment to the original color, making it lighter as more white is added.
- Tones of the color are produced by adding both black and white pigments.

2. Discuss in detail about light sources and basic illumination models.

The mechanism of light reflection from an actual surface is very complicated it depends on many factors. Some of these factors are geometric and others are related to the characteristics of the surface.

A shading model dictates how light is scattered or reflected from a surface. The shading models described here focuses on achromatic light. **Achromatic light** has brightness and no color, it is a shade of gray so it is described by a single value its intensity.

A shading model uses two types of light source to illuminate the objects in a scene : **point light sources** and **ambient light**. Incident light interacts with the surface in three different ways:

Some is absorbed by the surface and is converted to heat. Some is reflected from the surface

Some is transmitted into the interior of the object

If all incident light is absorbed the object appears black and is known as a **black body**. If all of the incident light is transmitted the object is visible only through the effects of reflection.

Some amount of the reflected light travels in the right direction to reach the eye causing the object to be seen. The amount of light that reaches the eye depends on the orientation of the surface, light and the observer. There are two different types of reflection of incident light

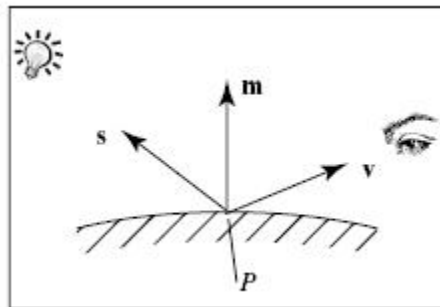
Diffuse scattering occurs when some of the incident light slightly penetrates the surface and is re-radiated uniformly in all directions. Scattered light interacts strongly with the surface and so its color is usually affected by the nature of the surface material.

- **Specular reflections** are more mirrorlike and highly directional. Incident light is directly reflected from its outer surface. This makes the surface looks shinny. In the simplest model the reflected light has the same color as the incident light, this makes the material look like plastic. In a more complex model the color of the specular light varies , providing a better approximation to the shininess of metal surfaces. The total light reflected from the surface in a certain direction is the sum of the diffuse component and the specular component. For each surface point of interest we compute the size of each component that reaches the eye.

Geometric Ingredients For Finding Reflected Light

We need to find three vectors in order to compute the diffuse and specular components. The below fig. shows three principal vectors (s , m and v) required to find the amount of light that reaches the eye from a point P .

Important directions in computing the reflected light



1. The normal vector , m , to the surface at P .
2. The vector v from P to the viewer's eye.
3. The vector s from P to the light source.

The angles between these three vectors form the basis of computing light intensities. These angles are normally calculated using world coordinates.

Each face of a mesh object has two sides. If the object is solid, one is inside and the other is outside. The eye can see only the outside and it is this side for which we must compute light contributions.

We shall develop the shading model for a given side of a face. If that side of the face is turned away from the eye there is no light contribution.

4.1.1 How to Compute the Diffuse Component

Suppose that a light falls from a point source onto one side of a face, a fraction of it is re-radiated diffusely in all directions from this side. Some fraction of the re-radiated part reaches the eye, with an intensity denoted by I_d .

An important property assumed for diffuse scattering is that it is independent of the direction from the point P, to the location of the viewer's eye. This is called **omnidirectional scattering**, because scattering is uniform in all directions. Therefore I_d is independent of the angle between m and v .

On the other hand the amount of light that illuminates the face does not depend on the orientation of the face relative to the point source. The amount of light is proportional to the area of the face that it sees.

The brightness depends on the area of the face that it sees

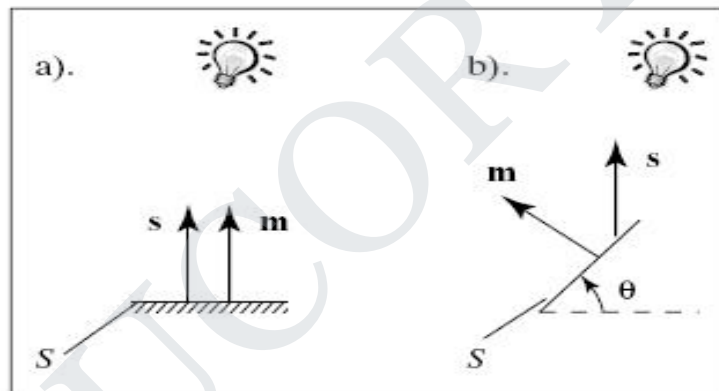


Fig (a) shows the cross section of a point source illuminating a face S when m is aligned with s .

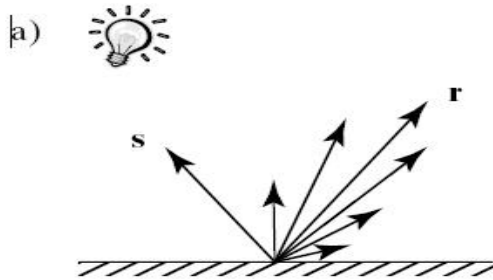
Fig (b) the face is turned partially away from the light source through angle θ . The area subtended is now only $\cos(\theta)$, so that the brightness of S is reduced by this same factor. This relationship between the brightness and surface orientation is called **Lambert's law**.

Specular Reflection

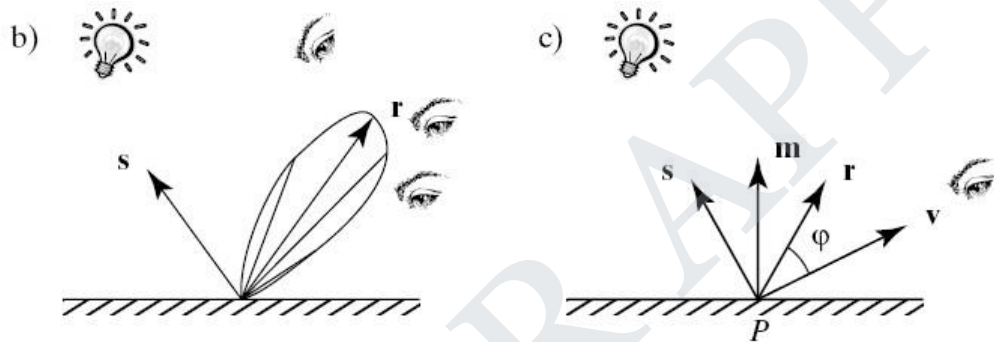
Real objects do not scatter light uniformly in all directions and so a specular component is added to the shading model. Specular reflection causes highlights which can add reality to a picture when objects are shiny. The behavior of specular light can be explained with Phong model.

Phong Model

It is easy to apply and the highlights generated by the phong model give an plastic like appearance, so the phong model is good when the object is made of shiny plastic or glass. The Phong model is less successful with objects that have a shiny metallic surface. Fig a) shows a situation where light from a source impinges on a surface and is reflected in different directions.



In this model we discuss the amount of light reflected is greatest in the direction of perfect mirror reflection, r , where the angle of incidence θ equals the angle of reflection. This is the direction in which all light would travel if the surface were a perfect mirror. At the other nearby angles the amount of light reflected diminishes rapidly, Fig (b) shows this with beam patterns. The distance from P to the beam envelope shows the relative strength of the light scattered in that direction.



Fig(c) shows how to quantify this beam pattern effect. The direction r of perfect reflection depends on both s and the normal vector m to the surface,

For surfaces that are shiny but are not true mirrors, the amount of light reflected falls off as the angle ϕ between r and v increases. In Phong model the ϕ is said to vary as some power f of the cosine of ϕ i.e., $(\cos(\phi))^f$ in which f is chosen experimentally and usually lies between 1 and 200.

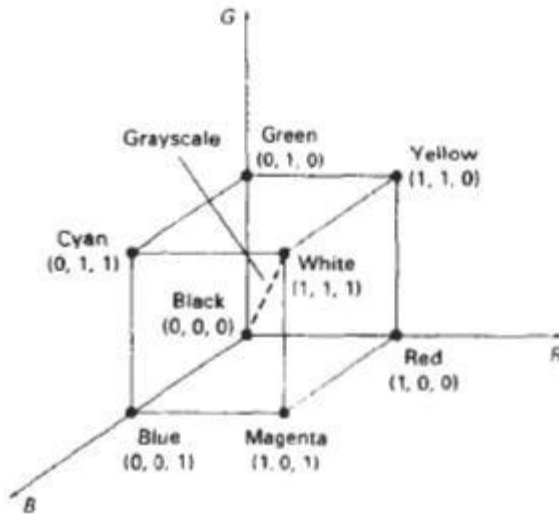
3. Explain in detail various color models.

- RGB color model
- YIQ
- CMY
- HSV
- HLS

RGB Color Model

- Based on the tristimulus theory of vision, our eyes perceive color through the stimulation of three visual pigments in the cones on the retina.
- These visual pigments have a peak sensitivity at wavelengths of about 630 nm (red), 530 nm (green) and 450 nm (blue).

- By comparing intensities in a light source, we perceive the color of the light.
- This is the basis for displaying color output on a video monitor using the 3 color primaries, red, green, and blue referred to as the RGB color model. It is represented in the below figure.



- The sign represents black, and the vertex with coordinates (1,1,1) in white.
- Vertices of the cube on the axes represent the primary colors, the remaining vertices represents the complementary color for each of the primary colors.
- The RGB color scheme is an additive model. (i.e.,) Intensities of the primary colors are added to produce other colors.
- Each color point within the bounds of the cube can be represented as the triple (R,G,B) where values for R, G and B are assigned in the range from 0 to 1.
- The color C_λ is expressed in RGB component as $C_\lambda = R\mathbf{R} + G\mathbf{G} + B\mathbf{B}$
- The magenta vertex is obtained by adding red and blue to produce the triple (1,0,1) and white at (1,1,1) in the sum of the red, green and blue vertices.
- Shades of gray are represented along the main diagonal of the cube from the origin (black) to the white vertex.

YIQ Color Model

- The National Television System Committee (NTSC) color model for forming the composite video signal in the YIQ model.
- In the YIQ color model, luminance (brightness) information is contained in the Y parameter, chromaticity information (hue and purity) is contained into the I and Q parameters.
- A combination of red, green and blue intensities are chosen for the Y parameter to yield the standard luminosity curve.
- Since Y contains the luminance information, black and white TV monitors use only the Y signal.
- Parameter I contain orange-cyan hue information that provides the flash-tone shading and occupies a bandwidth of 1.5 MHz.
- Parameter Q carries green-magenta hue information in a bandwidth of about 0.6 MHz.

- An RGB signal can be converted to a TV signal using an NTSC encoder which converts RGB values to YIQ values, as follows

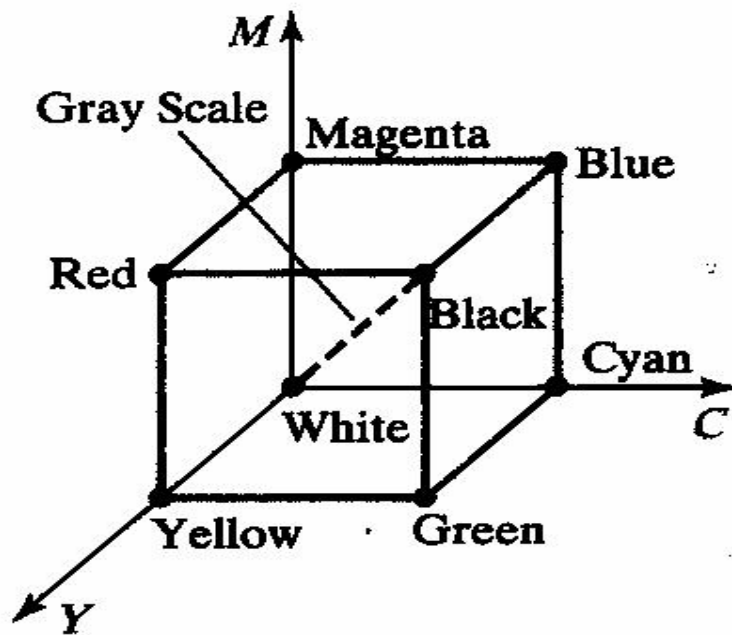
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & 0.275 & 0.321 \\ 0.212 & 0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

An NTSC video signal can be converted to an RGB signal using an NTSC decoder which separates the video signal into YIQ components, then converts to RGB values, as follows

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.620 \\ 1.000 & 0.272 & 0.647 \\ 1.000 & 1.108 & 1.705 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

CMY Color Model

- A color model defined with the primary colors cyan, magenta, and yellow (CMY) is useful for describing color output to hard copy devices.
- It is a subtractive color model (i.e.,) cyan can be formed by adding green and blue light. When white light is reflected from cyan-colored ink, the reflected light must have no red component. i.e., red light is absorbed or subtracted by the ink.
 - Magenta ink subtracts the green component from incident light and yellow subtracts the blue component.



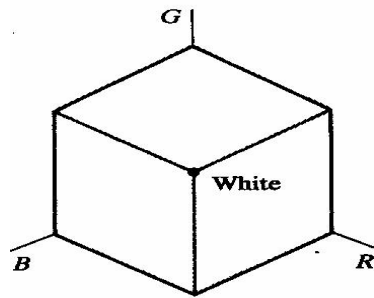
In CMY model, point $(1,1,1)$ represents black because all components of the incident light are subtracted.

- The origin represents white light.
 - Equal amounts of each of the primary colors produce grays along the main diagonal of the cube.
 - A combination of cyan and magenta ink produces blue light because the red and green components of the incident light are absorbed.
 - The printing process often used with the CMY model generates a color point with a collection of 4 ink dots; one dot is used for each of the primary colors (cyan, magenta and yellow) and one dot in black.

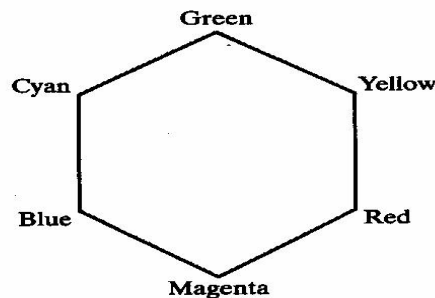
HSV Color Model

The HSV model uses color descriptions that have a more interactive appeal to a user.

- Color parameters in this model are hue (H), saturation (S), and value (V).
- The 3D representation of the HSV model is derived from the RGB cube. The outline of the cube has the hexagon shape.

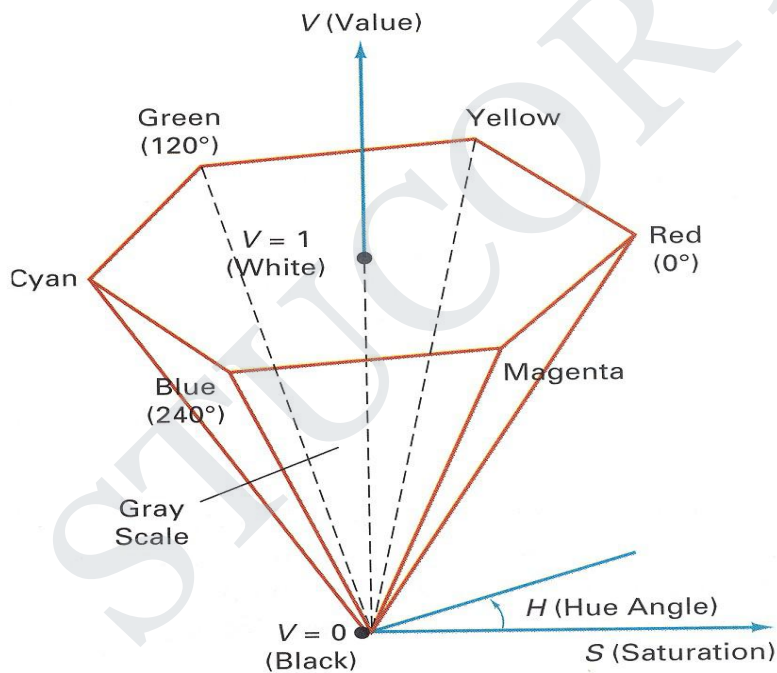


RGB Color Cube
(a)



Color Hexagon
(b)

- The boundary of the hexagon represents the various hues, and it is used as the top of the HSV hexcone.
- In the hexcone, saturation is measured along a horizontal axis, and value is along a vertical axis through the center of the hexcone.
- Hue is represented as an angle about the vertical axis, ranging from 0° at red through 360° . Vertices of the hexagon are separated by 60° intervals.
- Yellow is at 60° , green at 120° and cyan opposite red at $H = 180^\circ$. Complementary colors are 180° apart.



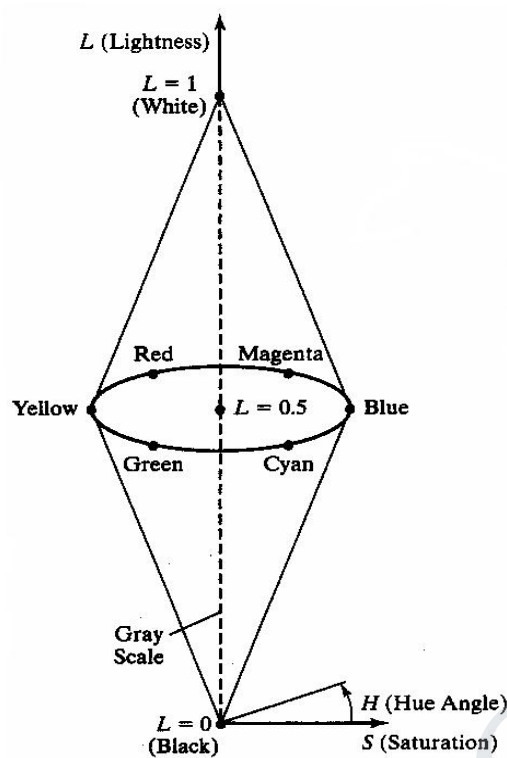
Saturation S varies from 0 to 1. the maximum purity at $S = 1$, at $S = 0.25$, the hue is said to be one quarter pure, at $S = 0$, we have the gray scale.

- Value V varies from 0 at the apex to 1 at the top. the apex representation black.
- At the top of the hexcone, colors have their maximum intensity.
- When $V = 1$ and $S = 1$ we have the „pure“ hues. White is the poin at $V = 1$ and $S = 0$.

HLS Color Model

HLS model is based on intuitive color parameters used by Tektronix.

- It has the double cone representation shown in the below figure. The 3 parameters in this model are called Hue (H), lightness (L) and saturation (s).



- Hue specifies an angle about the vertical axis that locates a chosen hue.
- In this model $H = 0^\circ$ corresponds to Blue.
- The remaining colors are specified around the perimeter of the cone in the same order as in the HSV model
- Magenta is at 60° , Red in at 120° , and cyan in at $H = 180^\circ$.
- The vertical axis is called lightness (L). At $L = 0$, we have black, and white is at $L = 1$ Gray scale in along the L axis and the "purehues" on the $L = 0.5$ plane.
- Saturation parameter S specifies relative purity of a color. S varies from 0 to 1 pure hues are those for which $S = 1$ and $L = 0.5$
- As S decreases, the hues are said to be less pure.
- At $S = 0$, it is said to be gray scale.

4. Explain DDA algorithm for line.

- Algorithm
- Definition

- Diagram
- Theory
- Implementation

The digital differential analyzer is a scan conversion algorithm based on calculation either Δy or Δx using the following equations

$$\Delta y = m \Delta x$$

$$\Delta x = \Delta y / m$$

Sample the line at unit intervals in one coordinate and determine corresponding integer values nearest the line path for the coordinates

Sample at X intervals ($\Delta x = 1$) and compute each successive Y value as $Y_{k+1} = Y_k + m$

For lines with positive slope greater than 1, reverse the roles of X and Y. Sample at unit Y intervals ($\Delta y = 1$) and calculate each successive X value as $X_{k+1} = X_k + 1/m$

Algorithm

Step 1: Input the line endpoints and store the left endpoint in (x1, y1) and right endpoint in (x2, y2)

Step 2: Calculate the values of Δx and Δy using $\Delta x = x_b - x_a$, $\Delta y = y_b - y_a$

Step 3: if the values of $\Delta x > \Delta y$ assign values of steps as Δx otherwise the values of steps as Δy

Step 4: Calculate the values of X increment and Y increment and assign the value $x = x_a$ and $y = y_a$

Step 5: for $k=1$ to steps do

$X = X + X \text{ increment}$

$Y = Y + Y \text{ increment}$

Putpixel(ceil(x), ceil(y), 15)

Step 6: End

5. Explain Bresenham's algorithm for line, circle, ellipse.

Algorithm, Definition, Diagram, Theory Implementation

In Bresenham's approach the pixel position along a line path are determined by sampling unit X intervals. Starting from the left end point (X_0, Y_0) of a given line we step to each successive columns and plot the pixel whose scan line Y-value is closest to the line path. Assuming the Kth step in process, determined that the pixel at (X_k, Y_k) decide which pixel to plot in column X_{k+1} . The choices are (X_{k+1}, Y_k) and (X_{k+1}, Y_{k+1})

Algorithm

Step 1: Input the line endpoints and store the left endpoint in (X_0, Y_0)

Step 2: Load (X_0, Y_0) in to the frame buffer

Step 3: Calculate constants Δx , Δy , $2 \Delta y$, $-2 \Delta x$, and obtain the decision parameters as

$$P_0 = 2 \Delta y - \Delta x$$

Step Algorithm

Step 1: Input radius r_x , r_y and ellipse center (X_c, Y_c) and obtain the first point on the circumference of a circle centered on the origin as

$$(X_0, Y_0) = (0, r_y)$$

Step 2: Calculate the initial values of the decision parameter in region 1 as

$$P_{10} = r_y^2 - r_x^2 r_y + 1/4 r_x^2$$

Step AI

Step 3: At each position starting at X_k position in region 1, starting at $k = 0$, perform the following test:

If $P_k < 0$, the next point to plot is (X_{k+1}, Y_k) and

$$P_{1k+1} = P_{1k} + 2 r_y^2 X_{k+1} + r_y^2$$

Otherwise the next point is (X_{k+1}, Y_{k-1}) and

$$P_{1k+1} = P_{1k} + 2 r_y^2 X_{k+1} - 2 r_y^2 Y_{k+1} + r_y^2$$

Step 4: Calculate the initial values of the decision parameter in region 2 as

$$P_{20} = r_y^2 (X_0 + 1/2)^2 + r_x^2 (Y_0 - 1)^2 - r_x^2 r_y^2$$

Step 5: At each position starting at Y_k position in region 2, starting at $k = 0$, perform the following test:

If $P_k > 0$, the next point to plot is (X_k, Y_{k-1}) and

$$P_{2k+1} = P_{2k} - 2 r_y^2 Y_{k+1} + r_x^2$$

Otherwise the next point is (X_{k+1}, Y_{k-1}) and

$$P_{2k+1} = P_{2k} - 2 r_y^2 Y_{k+1} - 2 r_x^2 Y_{k+1} + r_x^2$$

Step 6: Determine symmetry points in the other three octants

Step 7: Move each pixel position (X, Y) onto the circular path centred on

(X_c, Y_c) and plot the coordinate values as

$$X = X + X_c \quad Y = Y + Y_c$$

Step 8: Repeat steps for region 1 until $2 r_y^2 X \geq 2 r_y^2 Y$

UNIT II

TWO-DIMENSIONAL GRAPHICS

1. Define Translation.

A translation is applied to an object by repositioning it along a straight line path from one coordinate location to another. We translate a two-dimensional point by adding translation distances, t_x and t_y , to original coordinate position (x, y) to move the point to a new position (x', y') . $x' = x + t_x$, $y' = y + t_y$. The translation distance pair (t_x, t_y) is called a translation vector or shift vector.

2. Define Rotation.

A 2-D rotation is applied to an object by repositioning it along a circular path in the xy plane.

3. Define Scaling.

A scaling transformation alters the size of an object. This operation can be carried out for polygons by multiplying the coordinate values (x, y) of each vertex by scaling factors s_x and s_y to produce the transformed coordinates (x', y') . $x' = x \cdot s_x$, $y' = y \cdot s_y$.

4. Define Reflection.

A Reflection is a transformation that produces a mirror image of an object. The mirror image for a 2D reflection is generated relative to an axis of reflection by rotating the object 180 degree about the reflection axis.

5. Define Shear.

A transformation that distorts the shape of an object such that the transformed shape appears as if the object were composed of internal layers that had been caused to slide over each other is called a shear.

6. Define Window.

A world-coordinate area selected for display is called a window.

7. Define view port.

An area on a display device to which a window is mapped is called a view port.

8. What is viewing transformation?

The mapping of a part of a world-coordinate scene to device coordinates is referred to as viewing transformation.

9. Define Clipping.

Any procedure that identifies those portions of a picture that are either inside or outside of a specified region of space is referred to as a clipping algorithm or simply clipping. The region against which an object is clipped is called a clip window.

10. What are the types of Clipping?

- Point clipping
- Line clipping
- Area clipping
- Curve clipping
- Text clipping

11. What is composite transformation?

The process of building complex geometric transformation from basic transformations is called composite transformation.

This is obtained by multiplying individual transformations in sequence

PART B**1. Explain 2D Transformations****Composite Transformations**

A composite transformation is a sequence of transformations; one followed by the other. we can set up a matrix for any sequence of transformations as a **composite transformation matrix** by calculating the matrix product of the individual transformations

Translation

If two successive translation vectors $(tx1, ty1)$ and $(tx2, ty2)$ are applied to a coordinate position P , the final transformed location P' is calculated as

$$P' = T(tx2, ty2) \cdot \{T(tx1, ty1) \cdot P\}$$

$$= \{T(tx2, ty2) \cdot T(tx1, ty1)\} \cdot P$$

Where P and P' are represented as homogeneous-coordinate column vectors.

$$\begin{bmatrix} 1 & 0 & tx2 \\ 0 & 1 & ty2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & tx1 \\ 0 & 1 & ty1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx1+tx2 \\ 0 & 1 & ty1+ty2 \\ 0 & 0 & 1 \end{bmatrix}$$

Or

$$T(tx2, ty2) \cdot T(tx1, ty1) = T(tx1+tx2, ty1+ty2)$$

Which demonstrated the two successive translations are additive.

Rotations

Two successive rotations applied to point P produce the transformed position

$$P' = R(\theta2) \cdot \{R(\theta1) \cdot P\} = \{R(\theta2) \cdot R(\theta1)\} \cdot P$$

By multiplying the two rotation matrices, we can verify that two successive rotation are additive

$$R(\theta2) \cdot R(\theta1) = R(\theta1 + \theta2)$$

So that the final rotated coordinates can be calculated with the composite rotation matrix as

$$P' = R(\theta_1 + \theta_2).P$$

$$\begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_2 + \theta_1) & \sin(\theta_2 + \theta_1) & 0 \\ \sin(\theta_2 + \theta_1) & \cos(\theta_2 + \theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling

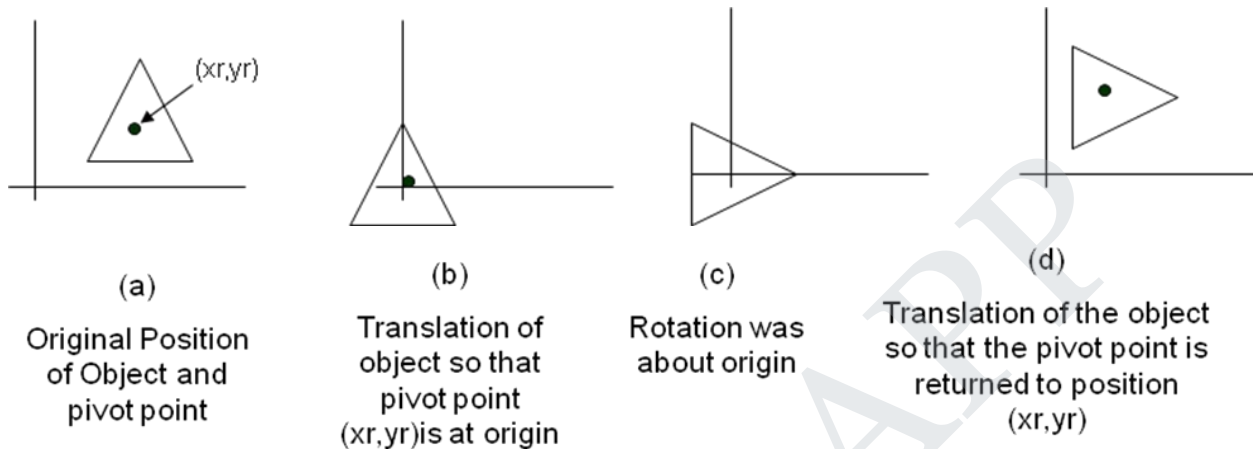
Concatenating transformation matrices for two successive scaling operations produces the following composite scaling matrix

$$\begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x2}.s_{x1} & 0 & 0 \\ 0 & s_{y2}.s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

General Pivot-point Rotation

1. Translate the object so that pivot-position is moved to the coordinate origin
2. Rotate the object about the coordinate origin

Translate the object so that the pivot point is returned to its original position



The composite transformation matrix for this sequence is obtain with the concatenation

$$\begin{bmatrix} 1 & 0 & xr \\ 0 & 1 & yr \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -xr \\ 0 & 1 & -yr \\ 0 & 0 & 1 \end{bmatrix}$$

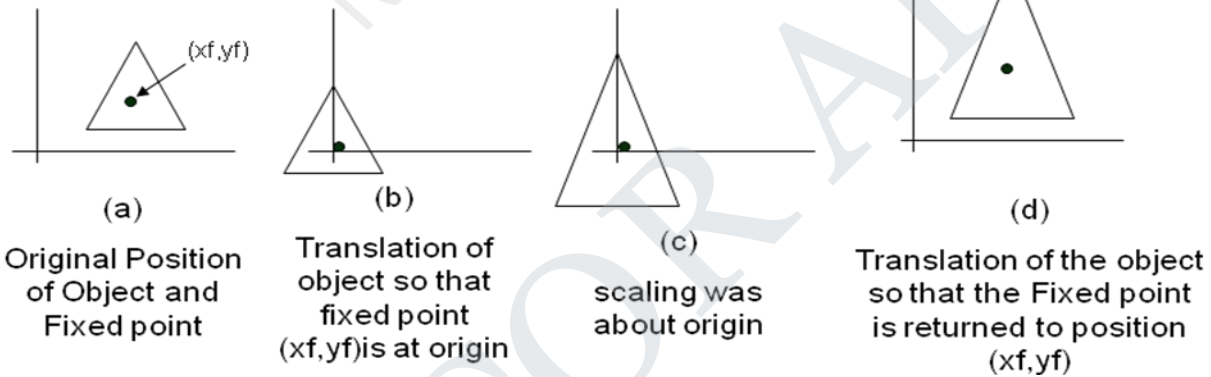
$$\begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta)+y_r\sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta)-x_r\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

Which can also be expressed as $T(x_r, y_r) \cdot R(\theta) \cdot T(-x_r, -y_r) = R(x_r, y_r, \theta)$

General fixed point scaling

Translate object so that the fixed point coincides with the coordinate origin
Scale the object with respect to the coordinate origin

Use the inverse translation of step 1 to return the object to its original position



Concatenating the matrices for these three operations produces the required scaling matrix

$$\begin{bmatrix} 1 & 0 & xf \\ 0 & 1 & yf \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -xf \\ 0 & 1 & -yf \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} sx & 0 & xf(1-sx) \\ 0 & sy & yf(1-sy) \\ 0 & 0 & 1 \end{bmatrix}$$

Can also be expressed as $T(xf,yf).S(sx,sy).T(-xf,-yf) = S(xf, yf, sx, sy)$

Note : Transformations can be combined by matrix multiplication

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 0 & tx \\ 1 & ty \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & \sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Implementation of composite transformations

```
#include <math.h> #include
<graphics.h>
typedef float Matrix3x3 [3][3]; Matrix3x3
thematrix;
```

```
void matrix3x3SetIdentity (Matrix3x3 m)
{
int i,j;
for (i=0; i<3; i++)
for (j=0; j<3; j++)
m[i][j] = (i == j);
}
```



```

/* Multiplies matrix a times b, putting result in b */ void
matrix3x3PreMultiply (Matrix3x3 a, Matrix3x3 b)
{
    int r,c;
    Matrix3x3 tmp;
    for (r = 0; r < 3; r++)
        for (c = 0; c < 3; c++)
            tmp[r][c] = a[r][0]*b[0][c] + a[r][1]*b[1][c] + a[r][2]*b[2][c];
    for (r = 0; r < 3; r++)
        for (c = 0; c < 3; c++)
            b[r][c] = tmp[r][c];
}

```

```

void translate2 (int tx, int ty)
{
    Matrix3x3 m;
    matrix3x3SetIdentity (m);
    m[0][2] = tx;
    m[1][2] = ty;
    matrix3x3PreMultiply (m, theMatrix);
}

void scale2 (float sx, float sy, wcPt2 refpt) (
    Matrix3x3 m;
    matrix3x3SetIdentity (m);
    m[0][0] = sx;
    m[0][2] = (1 - sx)* refpt.x;
    m[1][1] = sy;
    m[1][2] = (1 - sy)* refpt.y;
    matrix3x3PreMultiply (m, theMatrix);
}

```

```

void rotate2 (float a, wcPt2 refPt)
{
    Matrix3x3 m;
    matrix3x3SetIdentity (m);
    a = pToRadians (a);
    m[0][0] = cosf (a);
    m[0][1] = -sinf (a);
    m[0][2] = refPt.x * (1 - cosf (a)) + refPt.y sinf (a);
    m[1][0] = sinf (a);
    m[1][1] = cosf (a);
}

```

```

m[1][2] = refPt.y * (1 - cosf(a)) - refPt.x * sinf(a);
matrix3x3PreMultiply(m, theMatrix);
}

void transformPoints2 (int npts, wcPt2 *pts)
{
    int k;
    float tmp;
    for (k = 0; k < npts; k++)
    {
        tmp = theMatrix[0][0]* pts[k].x + theMatrix[0][1] * pts[k].y + theMatrix[0][2]; pts[k].y =
        theMatrix[1][0]* pts[k].x + theMatrix[1][1] * pts[k].y + theMatrix[1][2]; pts[k].x = tmp;
    }
}

void main (int argc, char **argv)
{
    wcPt2 pts[3] = { 50.0, 50.0, 150.0, 50.0, 100.0, 150.0 };
    wcPt2 refPt = { 100.0, 100.0 };
    long windowID = openGraphics (*argv, 200, 350);
    setBackground (WHITE);
    setColor (BLUE);
    pFillArea(3, pts);
    matrix3x3SetIdentity(theMatrix);
    scale2 (0.5, 0.5, refPt);
    rotate2 (90.0, refPt);
    translate2 (0, 150);
    transformPoints2 (3, pts);
    pFillArea(3, pts);
    sleep (10);
    closeGraphics (windowID);
}

```

Other Transformations

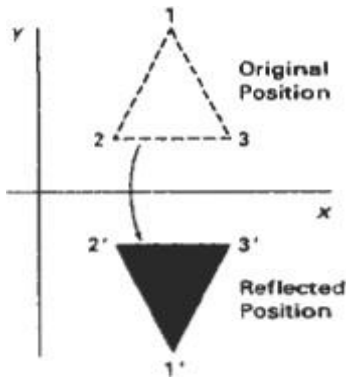
1. Reflection
2. Shear

Reflection

A reflection is a transformation that produces a mirror image of an object. The mirror image for a two-dimensional reflection is generated relative to an axis of reflection by

rotating the object 180° about the reflection axis. We can choose an axis of reflection in the xy plane or perpendicular to the xy plane or coordinate origin

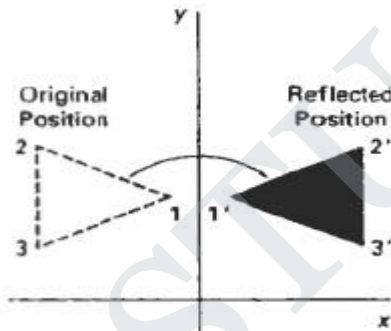
Reflection of an object about the x axis



Reflection the x axis is accomplished with the transformation

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

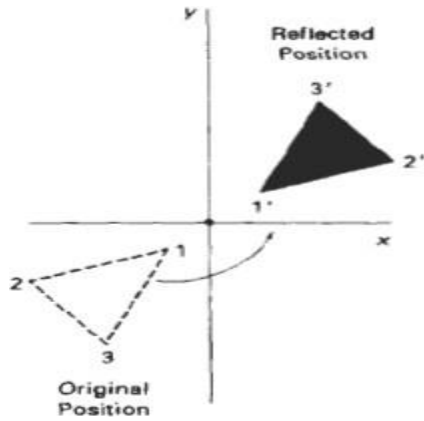
Reflection of an object about the y axis



Reflection the y axis is accomplished with the transformation matrix

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

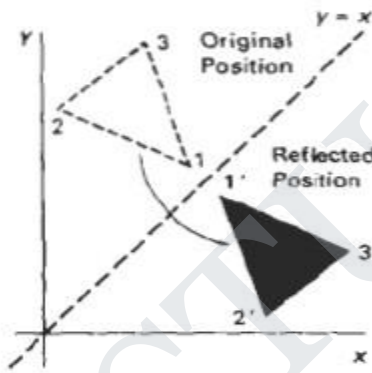
Reflection of an object about the coordinate origin



Reflection about origin is accomplished with the transformation matrix

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Reflection axis as the diagonal line $y = x$



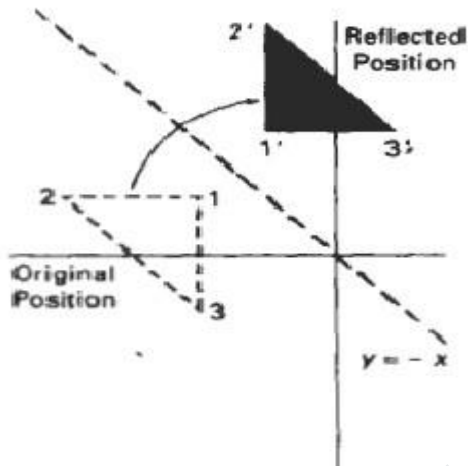
To obtain transformation matrix for reflection about diagonal $y=x$ the transformation sequence is

1. Clock wise rotation by 45°
2. Reflection about x axis
3. counter clock wise by 45°

Reflection about the diagonal line $y=x$ is accomplished with the transformation matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Reflection axis as the diagonal line $y = -x$



$y=-x$ the transformation
To obtain transformation matrix for reflection about diagonal
sequence is

1. Clock wise rotation by 45°
2. Reflection about y axis
3. counter clockwise by 45°

Reflection about the diagonal line $y=-x$ is accomplished with the

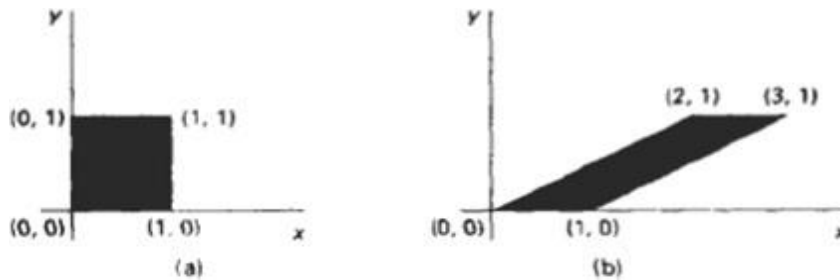
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Shear
transformation matrix**

A Transformation that slants the shape of an object is called the shear transformation. Two common shearing transformations are used. One shifts x coordinate values and other shift y coordinate values. However in both the cases only one coordinate (x or y) changes its coordinates and other preserves its values.

X- Shear

The x shear preserves the y coordinates, but changes the x values which cause vertical lines to tilt right or left as shown in figure



The Transformations matrix for x-shear is

$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which transforms the coordinates as x'

$$=x + sh_x \cdot y$$

$$y' = y$$

Y Shear

The y shear preserves the x coordinates, but changes the y values which cause horizontal lines which slope up or down

The Transformations matrix for y-shear is

$$\begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which transforms the coordinates as x'

$$=x$$

$$y' = y + sh_y \cdot x$$

XY-Shear

The transformation matrix for xy-shear

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

which transforms the coordinates as x'

$$= x + sh_x \cdot y$$

$$y' = y + sh_y \cdot x$$

Shearing Relative to other reference line

We can apply x shear and y shear transformations relative to other reference lines. In x shear transformations we can use y reference line and in y shear we can use x reference line.

X shear with y reference line

We can generate x-direction shears relative to other reference lines with the transformation matrix

$$\begin{bmatrix} 1 & sh_x & -sh_x \cdot y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

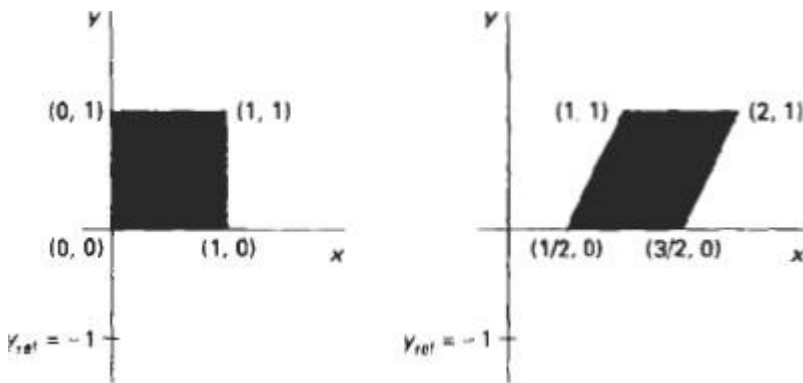
which transforms the coordinates as

$$x' = x + sh_x (y - y_{ref})$$

$$y' = y$$

Example

$$sh_x = \frac{1}{2} \quad \text{and} \quad y_{ref} = -1$$



Y shear with x reference line

We can generate y-direction shears relative to other reference lines with the transformation matrix

$$\begin{bmatrix} 1 & sh_x & -sh_x \cdot y_{ref} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

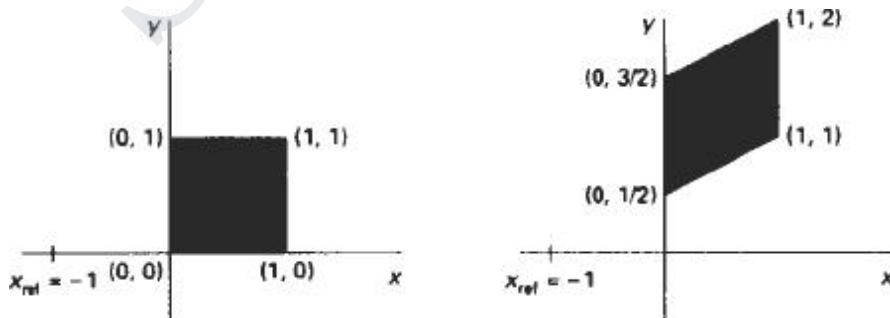
which transforms the coordinates as x'

$=x$

$$y' = sh_y (x - x_{ref}) + y$$

Example

$$sh_y = \frac{1}{2} \quad \text{and} \quad x_{ref} = -1$$



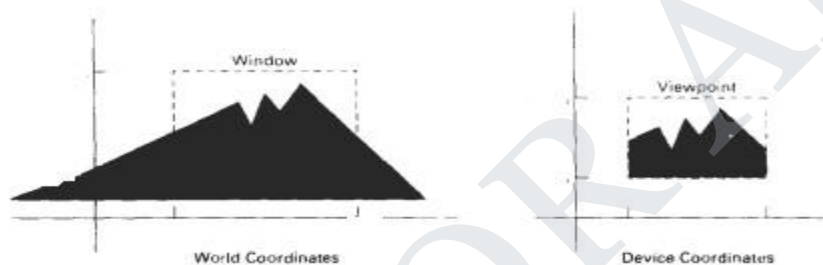
2. Explain 2D viewing

The viewing pipeline

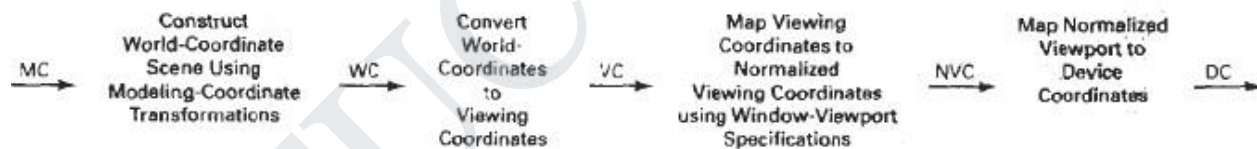
A world coordinate area selected for display is called a window. An area on a display device to which a window is mapped is called a view port. The window defines what is to be viewed the view port defines where it is to be displayed.

The mapping of a part of a world coordinate scene to device coordinate is referred to as viewing transformation. The two dimensional viewing transformation is referred to as window to view port transformation of windowing transformation.

A viewing transformation using standard rectangles for the window and viewport



The two dimensional viewing transformation pipeline



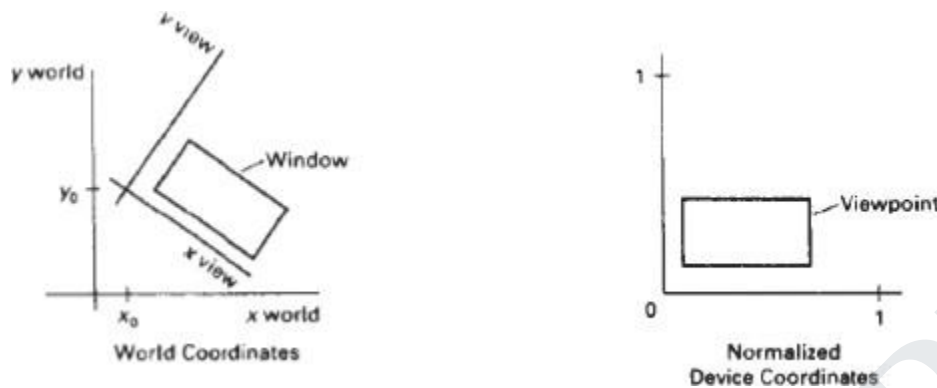
The viewing transformation in several steps, as indicated in Fig. First, we construct the scene in world coordinates using the output primitives. Next to obtain a particular orientation for the window, we can set up a two-dimensional viewing- coordinate system in the world coordinate plane, and define a window in the viewing- coordinate system.

The viewing- coordinate reference frame is used to provide a method for setting up arbitrary orientations for rectangular windows. Once the viewing reference frame is established, we can transform descriptions in world coordinates to viewing coordinates.

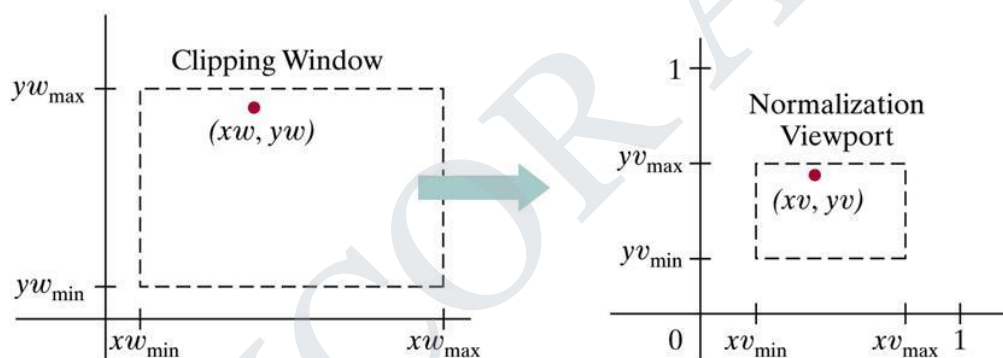
We then define a viewport in normalized coordinates (in the range from 0 to 1) and map the viewing-coordinate description of the scene to normalized coordinates.

At the final step all parts of the picture that lie outside the viewport are clipped, and the contents of the viewport are transferred to device coordinates. By changing the position

of the viewport, we can view objects at different positions on the display area of an output device.



Window to view port coordinate transformation:



A point (x_w, y_w) in a world-coordinate clipping window is mapped to viewport coordinates (x_v, y_v) , within a unit square, so that the relative positions of the two points in their respective rectangles are the same.

A point at position (x_w, y_w) in a designated window is mapped to viewport coordinates (x_v, y_v) so that relative positions in the two areas are the same. The figure illustrates the window to view port mapping.

A point at position (x_w, y_w) in the window is mapped into position (x_v, y_v) in the associated view port. To maintain the same relative placement in view port as in window

$$= \frac{XV - XV_{min}}{XV_{max} - XV_{min}} = \frac{XW - XW_{min}}{XW_{max} - XW_{min}}$$

$$= \frac{yV - yV_{min}}{yV_{max} - yV_{min}} = \frac{yW - yW_{min}}{yW_{max} - yW_{min}}$$

solving these expressions for view port position (x_v, y_v)

$$XV = XV_{min} + \frac{XW - XW_{min}}{XW_{max} - XW_{min}} (XV_{max} - XV_{min})$$

$$yV = yV_{min} + \frac{yW - yW_{min}}{yW_{max} - yW_{min}} (yV_{max} - yV_{min})$$

where scaling factors are

$$S_x = \frac{XV_{max} - XV_{min}}{XW_{max} - XW_{min}} \quad S_y = \frac{yV_{max} - yV_{min}}{yW_{max} - yW_{min}}$$

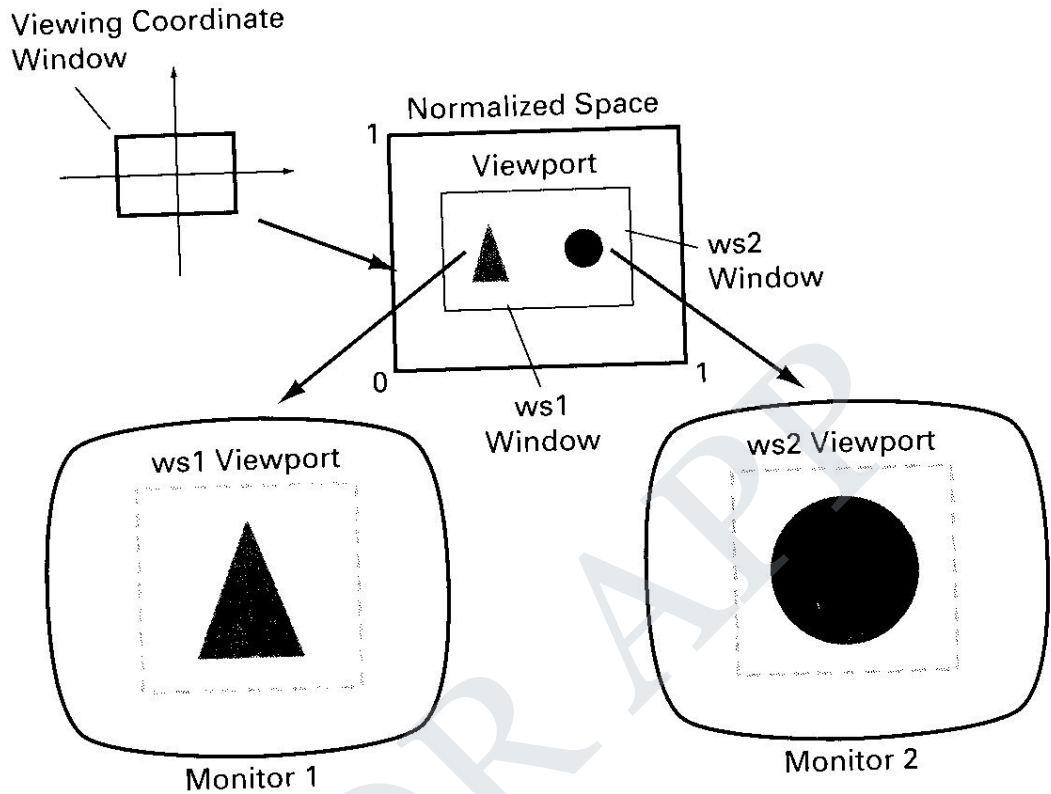
The conversion is performed with the following sequence of transformations.

1. Perform a scaling transformation using point position of ($x_w \min, y_w \min$) that scales the window area to the size of view port.
2. Translate the scaled window area to the position of view port. Relative proportions of objects are maintained if scaling factor are the same ($S_x = S_y$).

Otherwise world objects will be stretched or contracted in either the x or y direction when displayed on output device. For normalized coordinates, object descriptions are mapped to various display devices.

Any number of output devices can be open in particular application and another window view port transformation can be performed for each open output device. This mapping called the work station transformation is accomplished by selecting a window area in normalized space and a view port are in coordinates of display device.

Mapping selected parts of a scene in normalized coordinate to different video monitors with work station transformation.



3. Explain Cohen sutherland line clipping

This is one of the oldest and most popular line-clipping procedures. The method speeds up the processing of line segments by performing initial tests that reduce the number of intersections that must be calculated.

Every line endpoint in a picture is assigned a four digit binary code called a **region code** that identifies the location of the point relative to the boundaries of the clipping rectangle.

Binary region codes assigned to line end points according to relative position with respect to the clipping rectangle.

Regions are set up in reference to the boundaries. Each bit position in region code is used to indicate one of four relative coordinate positions of points with respect to clip window: to the left, right, top or bottom. By numbering the bit positions in the region code as 1 through 4 from right to left, the coordinate regions are corrected with bit positions as

bit 1: left

bit 2: right

bit 3: below

bit4: above

A value of 1 in any bit position indicates that the point is in that relative position. Otherwise the bit position is set to 0. If a point is within the clipping rectangle the region code is 0000. A point that is below and to the left of the rectangle has a region code of 0101.

Bit values in the region code are determined by comparing endpoint coordinate values (x,y) to clip boundaries. Bit1 is set to 1 if $x < x_{W_{min}}$.

For programming language in which bit manipulation is possible region-code bit values can be determined with following two steps.

- (1) Calculate differences between endpoint coordinates and clipping boundaries.
- (2) Use the resultant sign bit of each difference calculation to set the corresponding value in the region code.

bit 1 is the sign bit of $x - x_{W_{min}}$

bit 2 is the sign bit of $x_{W_{max}} - x$

bit 3 is the sign bit of $y - y_{W_{min}}$

bit 4 is the sign bit of $y_{W_{max}} - y$.

Once we have established region codes for all line endpoints, we can quickly determine which lines are completely inside the clip window and which are clearly outside.

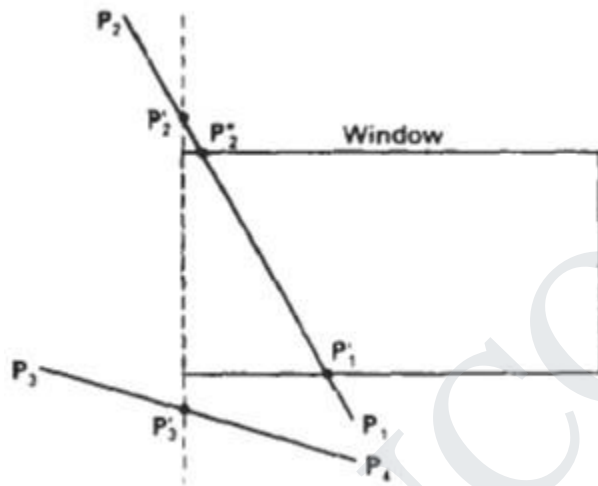
Any lines that are completely contained within the window boundaries have a region code of 0000 for both endpoints, and we accept

these lines. Any lines that have a 1 in the same bit position in the region codes for each endpoint are completely outside the clipping rectangle, and we reject these lines.

We would discard the line that has a region code of 1001 for one endpoint and a code of 0101 for the other endpoint. Both endpoints of this line are left of the clipping rectangle, as indicated by the 1 in the first bit position of each region code.

A method that can be used to test lines for total clipping is to perform the logical and operation with both region codes. If the result is not 0000, the line is completely outside the clipping region.

Lines that cannot be identified as completely inside or completely outside a clip window by these tests are checked for intersection with window boundaries.



Line extending from one coordinates region to another may pass through the clip

window, or they may intersect clipping boundaries without entering window.

Cohen-Sutherland line clipping starting with bottom endpoint left, right, bottom and top boundaries in turn and find that this point is below the clipping rectangle.

Starting with the bottom endpoint of the line from P_1 to P_2 , we check P_1 against the left, right, and bottom boundaries in turn and find that this point is below the clipping rectangle. We then find the intersection point P_1' with the bottom boundary and discard the line section from P_1 to P_1' .

The line now has been reduced to the section from P_1' to P_2 . Since P_2 is outside the clip window, we check this endpoint against the boundaries and find that it is to the left

of the window. Intersection point P_2' is calculated, but this point is above the window. So the final intersection calculation yields P_2'' , and the line from P_1' to P_2'' is saved. This completes processing for this line, so we save this part and go on to the next line.

Point P_3 in the next line is to the left of the clipping rectangle, so we determine the intersection P_3' , and eliminate the line section from P_3 to P_3' . By checking region codes for the line section from P_3' to P_4 we find that the remainder of the line is below the clip window and can be discarded also.

Intersection points with a clipping boundary can be calculated using the slope- intercept form of the line equation. For a line with endpoint coordinates (x_1, y_1) and (x_2, y_2) and the y coordinate of the intersection point with a vertical boundary can be obtained with the calculation.

$$y = y_1 + m(x - x_1)$$

where x value is set either to $x_{w_{min}}$ or to $x_{w_{max}}$ and slope of line is calculated as $m =$

$$(y_2 - y_1) / (x_2 - x_1)$$

the intersection with a horizontal boundary the x coordinate can be calculated as $x = x_1$

$$+ (y - y_1) / m$$

with y set to either to $y_{w_{min}}$ or to $y_{w_{max}}$.

Implementation of Cohen-sutherland Line Clipping

```
#define Round(a) ((int)(a+0.5))
```

```
#define LEFT_EDGE 0x1
```

```
#define RIGHT_EDGE 0x2
```

```
#define BOTTOM_EDGE 0x4
```

```
#define TOP_EDGE 0x8
```

```
#define TRUE 1
```

```
#define FALSE 0
```

```
#define INSIDE(a) (!a) #define
```

```
REJECT(a,b) (a&b) #define
```

```
ACCEPT(a,b) (!(a|b))
```

```
unsigned char encode(wcPt2 pt, dcPt winmin, dcPt winmax)
```

```
{
```

```

unsigned char code=0x00;
if(pt.x<winmin.x)
code=code|LEFT_EDGE;
if(pt.x>winmax.x)
code=code|RIGHT_EDGE;
if(pt.y<winmin.y)
code=code|BOTTOM_EDGE;
if(pt.y>winmax.y)
code=code|TOP_EDGE;
return(code);
}
void swappts(wcPt2 *p1,wcPt2 *p2)
{
wcPt2 tmp;
tmp=*p1;
*p1=*p2;
*p2=tmp;
}
void swapcodes(unsigned char *c1,unsigned char *c2)
{
unsigned char tmp;
tmp=*c1;
*c1=*c2;
*c2=tmp;
}
void clipline(dcPt winmin, dcPt winmax, wcPt2 p1,ecPt2 point p2)
{
unsigned char code1,code2;
int done=FALSE, draw=FALSE; float
m;
while(!done)
{
code1=encode(p1,winmin,winmax);
code2=encode(p2,winmin,winmax);
if(ACCEPT(code1,code2))
{
done=TRUE;
draw=TRUE;
}
else if(REJECT(code1,code2))
done=TRUE;
else
{

```

```

if(INSIDE(code1))
{
  swappts(&p1,&p2); swapcodes(&code1,&code2);
}
if(p2.x!=p1.x)
  m=(p2.y-p1.y)/(p2.x-p1.x);
if(code1 &LEFT_EDGE)
{
  p1.y+=(winmin.x-p1.x)*m;
  p1.x=winmin.x;
}
else if(code1 &RIGHT_EDGE)
{
  p1.y+=(winmax.x-p1.x)*m;
  p1.x=winmax.x;
}
else if(code1 &BOTTOM_EDGE)
{
  if(p2.x!=p1.x)
    p1.x+=(winmin.y-p1.y)/m;
  p1.y=winmin.y;
}
else if(code1 &TOP_EDGE)
{
  if(p2.x!=p1.x)
    p1.x+=(winmax.y-p1.y)/m;
  p1.y=winmax.y;
}
}
if(draw) lineDDA(ROUND(p1.x),ROUND(p1.y),ROUND(p2.x),ROUND(p2.y));
}

```

4. Explain Liang barsky's Line clipping algorithm

Based on analysis of parametric equation of a line segment, faster line clippers have been developed, which can be written in the form :

$$\begin{aligned}
 x &= x_1 + u \Delta x \\
 y &= y_1 + u \Delta y
 \end{aligned}
 \quad 0 \leq u \leq 1$$

where $\Delta x = (x_2 - x_1)$ and $\Delta y = (y_2 - y_1)$

In the Liang-Barsky approach we first the point clipping condition in parametric form :

$$x_{W_{\min}} \leq x_1 + u \Delta x \leq x_{W_{\max}}$$

$$y_{W_{\min}} \leq y_1 + u \Delta y \leq y_{W_{\max}}$$

Each of these four inequalities can be expressed as

$$\mu p_k \leq q_k, \quad k=1,2,3,4$$

the parameters p & q are defined as

$$\begin{aligned} p_1 &= -\Delta x & q_1 &= x_1 - x_{W_{\min}} \\ p_2 &= \Delta x & q_2 &= x_{W_{\max}} - x_1 \\ p_3 &= -\Delta y & q_3 &= y_1 - y_{W_{\min}} \\ p_4 &= \Delta y & q_4 &= y_{W_{\max}} - y_1 \end{aligned}$$

Any line that is parallel to one of the clipping boundaries have $p_k=0$ for values of k corresponding to boundary $k=1,2,3,4$ correspond to left, right, bottom and top boundaries. For values of k , find $q_k < 0$, the line is completely out side the boundary.

If $q_k \geq 0$, the line is inside the parallel clipping boundary.

When $p_k < 0$ the infinite extension of line proceeds from outside to inside of the infinite extension of this clipping boundary.

If $p_k > 0$, the line proceeds from inside to outside, for non zero value of p_k calculate the value of u , that corresponds to the point where the infinitely extended line intersect the extension of boundary k as

$$u = q_k / p_k$$

For each line, calculate values for parameters u_1 and u_2 that define the part of line that lies within the clip rectangle. The value of u_1 is determined by looking at the rectangle edges for which the line proceeds from outside to the inside ($p < 0$).

For these edges we calculate

$$r_k = q_k / p_k$$

The value of u_1 is taken as largest of set consisting of 0 and various values of r . The value of u_2 is determined by examining the boundaries for which lines proceeds from inside to outside ($P > 0$).

A value of r_k is calculated for each of these boundaries and value of u_2 is the minimum of the set consisting of 1 and the calculated r values.

If $u_1 > u_2$, the line is completely outside the clip window and it can be rejected.

Line intersection parameters are initialized to values $u_1=0$ and $u_2=1$. for each clipping boundary, the appropriate values for P and q are calculated and used by function

Cliptest to determine whether the line can be rejected or whether the intersection parameter can be adjusted.

When $p < 0$, the parameter r is used to update u_1 .

When $p > 0$, the parameter r is used to update u_2 .

If updating u_1 or u_2 results in $u_1 > u_2$ reject the line, when $p=0$ and $q < 0$, discard the line, it is parallel to and outside the boundary. If the line has not been rejected after all four value of p and q have been tested, the end points of clipped lines are determined from values of u_1 and u_2 .

The Liang-Barsky algorithm is more efficient than the Cohen-Sutherland algorithm since intersections calculations are reduced. Each update of parameters u_1 and u_2 require only one division and window intersections of these lines are computed only once.

Cohen-Sutherland algorithm, can repeatedly calculate intersections along a line path, even through line may be completely outside the clip window. Each intersection calculations require both a division and a multiplication.

Implementation of Liang-Barsky Line Clipping

```
#define Round(a)    ((int)(a+0.5))
int clipTest (float p, float q, gfloat *u1, float *u2)
{
    float r;
    int retval=TRUE; if
    (p<0.0)
    {
```

```

r=q/p
if      (r>*u2)
retVal=FALSE;
else
if (r>*u1)
*u1=r;
}
else
if (p>0.0)
{
r=q/p
if      (r<*u1)
retVal=FALSE;
else
if (r<*u2)
*u2=r;
}
else
if      )q<0.0)
retVal=FALSE
return(retVal);

```

```

void clipLine (dcPt winMin, dcPt winMax, wcPt2 p1, wcpt2 p2)
{
float u1=0.0, u2=1.0, dx=p2.x-p1.x,dy;
if (clipTest (-dx, p1.x-winMin.x, &u1, &u2)) if
(clipTest (dx, winMax.x-p1.x, &u1, &u2))
{
dy=p2.y-p1.y;
if (clipTest (-dy, p1.y-winMin.y, &u1, &u2)) if
(clipTest (dy, winMax.y-p1.y, &u1, &u2))
{
if (u1<1.0)
{
p2.x=p1.x+u2*dx;
p2.y=p1.y+u2*dy;
}
if (u1>0.0)
{
p1.x=p1.x+u1*dx;
p1.y=p1.y+u1*dy;
} lineDDA(ROUND(p1.x),ROUND(p1.y),ROUND(p2.x),ROUND(p2.y));
}
}
}

```



```

}
}
}

```

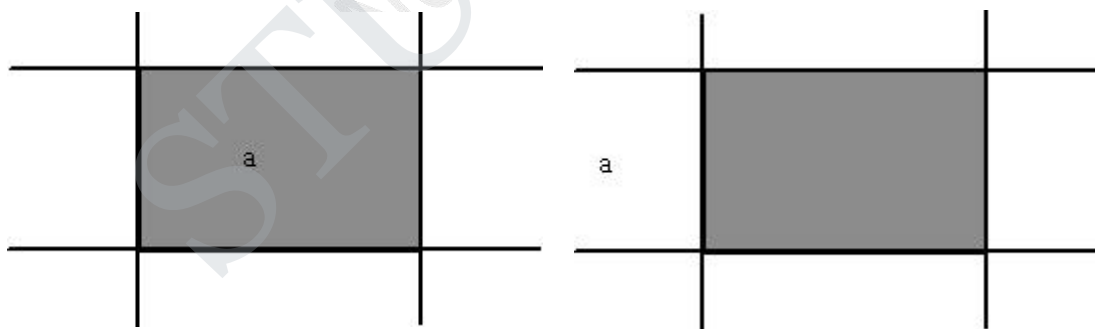
5. Explain in detail Nicholl-Lee-Nicholl line clipping.

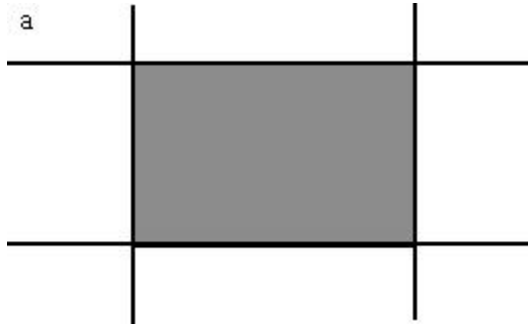
By creating more regions around the clip window, the Nicholl-Lee-Nicholl (or NLN) algorithm avoids multiple clipping of an individual line segment. In the Cohen- Sutherland method, multiple intersections may be calculated. These extra intersection calculations are eliminated in the NLN algorithm by carrying out more region testing before intersection positions are calculated.

Compared to both the Cohen-Sutherland and the Liang-Barsky algorithms, the Nicholl-Lee-Nicholl algorithm performs fewer comparisons and divisions. The trade-off is that the NLN algorithm can only be applied to two-dimensional clipping, whereas both the Liang-Barsky and the Cohen-Sutherland methods are easily extended to three- dimensional scenes.

For a line with endpoints P1 and P2 we first determine the position of point P1, for the nine possible regions relative to the clipping rectangle. Only the three regions shown in Fig. need to be considered. If P1 lies in any one of the other six regions, we can move it to one of the three regions in Fig. using a symmetry transformation. For example, the region directly above the clip window can be transformed to the region left of the clip window using a reflection about the line $y = -x$, or we could use a **90 degree** counterclockwise rotation.

Three possible positions for a line endpoint $p1(a)$ in the NLN algorithm





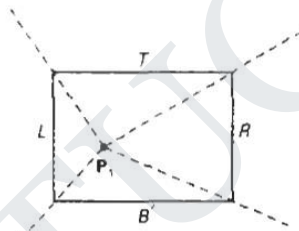
Case 1: p_1 inside region

Case 2: p_1 across edge

Case 3: p_1 across corner

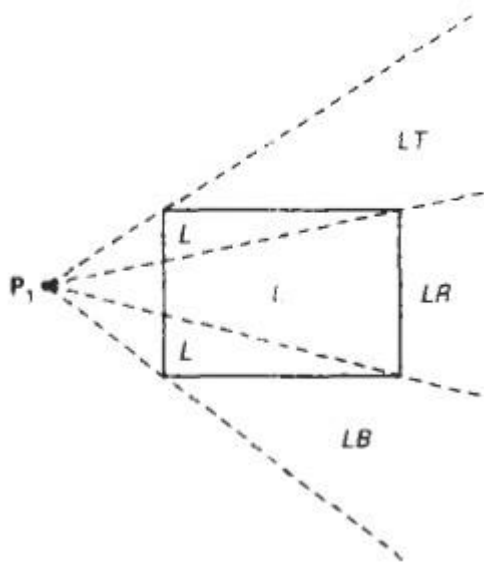
Next, we determine the position of P_2 relative to P_1 . To do this, we create some new regions in the plane, depending on the location of P_1 . Boundaries of the new regions are half-infinite line segments that start at the position of P_1 and pass through the window corners. If P_1 is inside the clip window and P_2 is outside, we set up the four regions shown in Fig

The four clipping regions used in NLN alg when p_1 is inside and p_2 outside the clip window



The intersection with the appropriate window boundary is then carried out, depending on which one of the four regions (L, T, R, or B) contains P_2 . If both P_1 and P_2 are inside the clipping rectangle, we simply save the entire line.

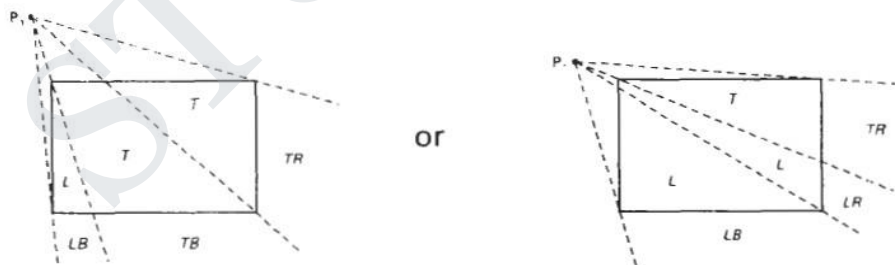
If P_1 is in the region to the left of the window, we set up the four regions, ***L, LT, LR, and LB***, shown in Fig.



These four regions determine a unique boundary for the line segment. For instance, if P_2 is in region L , we clip the line at the left boundary and save the line segment from this intersection point to P_2 . But if P_2 is in region LT , we save the line segment from the left window boundary to the top boundary. If P_2 is not in any of the four regions, L , LT , LR , or LB , the entire line is clipped.

For the third case, when P_1 is to the left and above the clip window, we use the clipping regions in Fig.

Fig : The two possible sets of clipping regions used in NLN algorithm when P_1 is above and to the left of the clip window



In this case, we have the two possibilities shown, depending on the position of P_1 , relative to the top left corner of the window. If P_2 is in one of the regions T , L , TR , TB , LR , or LB , this determines a unique clip window edge for the intersection calculations. Otherwise, the entire line is rejected.

To determine the region in which P_2 is located, we compare the slope of the

line to the slopes of the boundaries of the clip regions. For example, if P1 is left of the clipping rectangle (Fig. a), then P2, is in region LT if $\text{slope}_{P_1P_{TR}} < \text{slope}_{P_1P_2} < \text{slope}_{P_1P_{TL}}$

or

$$\frac{y_T - y_1}{x_R - x_1} < \frac{y_2 - y_1}{x_2 - x_1} < \frac{y_T - y_1}{x_L - x_1}$$

And we clip the entire line if

$$(y_T - y_1)(x_2 - x_1) < (x_L - x_1)(y_2 - y_1)$$

The coordinate difference and product calculations used in the slope tests are saved and also used in the intersection calculations. From the parametric equations

$$\begin{aligned} x &= x_1 + (x_2 - x_1)u \\ y &= y_1 + (y_2 - y_1)u \end{aligned}$$

an x-intersection position on the left window boundary is $x = x_L$, with $u =$

$(x_L - x_1) / (x_2 - x_1)$ so that the y-intersection position is

$$y = y_1 + \frac{(y_2 - y_1)(x_L - x_1)}{x_2 - x_1}$$

And an intersection position on the top boundary has $y = y_T$ and $u = (y_T - y_1) / (y_2 - y_1)$ with

$$x = x_1 + \frac{(x_2 - x_1)(y_T - y_1)}{y_2 - y_1}$$

6. Explain in detail about Sutherland Hodgeman polygon

A polygon can be clipped by processing the polygon boundary as a whole against each window edge. This could be accomplished by processing all polygon vertices against each clip rectangle boundary.

There are four possible cases when processing vertices in sequence around the perimeter of a polygon. As each point of adjacent polygon vertices is passed to a window boundary clipper, make the following tests:

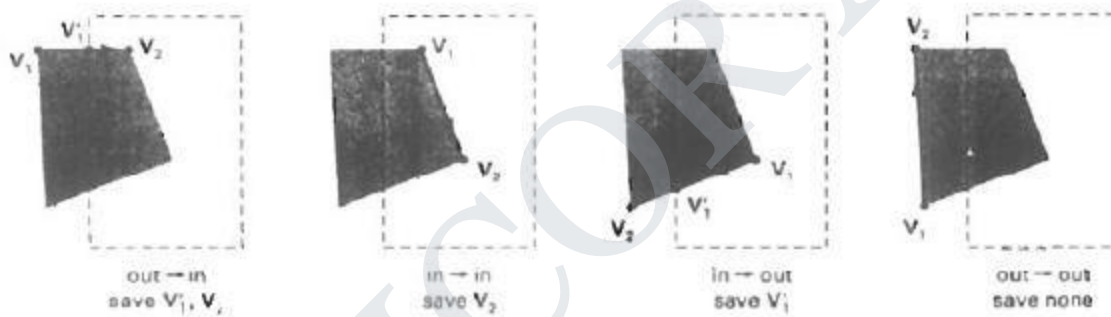
1. If the first vertex is outside the window boundary and second vertex is inside, both the intersection point of the polygon edge with window boundary and second vertex are added to output vertex list.

2. If both input vertices are inside the window boundary, only the second vertex is added to the output vertex list.
3. If first vertex is inside the window boundary and second vertex is outside only the edge intersection with window boundary is added to output vertexlist.
4. If both input vertices are outside the window boundary nothing is added to the output list.

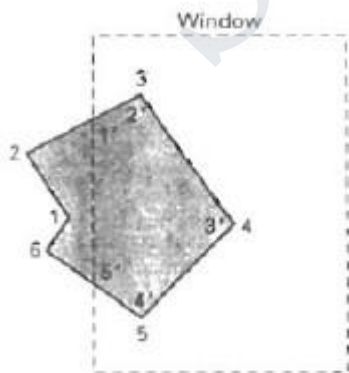
Clipping a polygon against successive window boundaries.



Successive processing of pairs of polygon vertices against the left window boundary



Clipping a polygon against the left boundary of a window, starting with vertex 1. Primed numbers are used to label the points in the output vertex list for this window boundary.

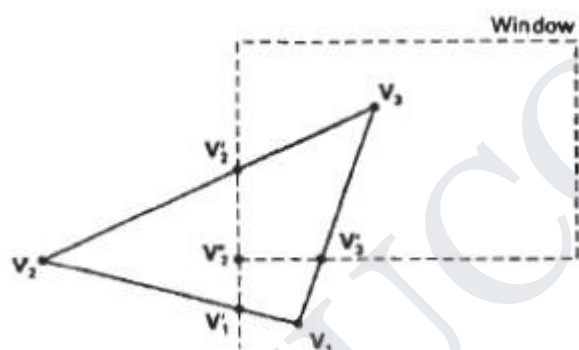


vertices 1 and 2 are found to be on outside of boundary. Moving along vertex 3 which is inside, calculate the intersection and save both the intersection point and vertex 3. Vertex 4 and 5 are determined to be inside and are saved. Vertex 6 is outside so we find and save the intersection point. Using the five saved points we repeat the process for next window boundary.

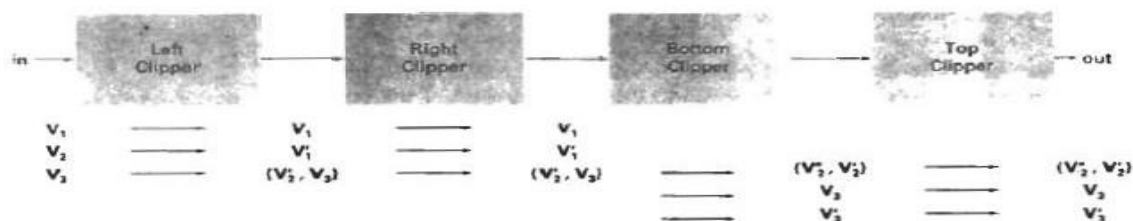
Implementing the algorithm as described requires setting up storage for an output list of vertices as a polygon clipped against each window boundary. We eliminate the intermediate output vertex lists by simply by clipping individual vertices at each step and passing the clipped vertices on to the next boundary clipper.

A point is added to the output vertex list only after it has been determined to be inside or on a window boundary by all boundary clippers. Otherwise the point does not continue in the pipeline.

A polygon overlapping a rectangular clip window



Processing the vertices of the polygon in the above fig. through a boundary clipping pipeline. After all vertices are processed through the pipeline, the vertex list is { v_2'' , v_2' , v_3 , v_3' }



Implementation of Sutherland-Hodgeman Polygon Clipping

```
typedef enum { Left, Right, Bottom, Top }
Edge; #define N_EDGE 4
```

```

#define TRUE 1
#define FALSE 0

int inside(wcPt2 p, Edge b, dcPt wmin, dcPt wmax)
{
switch(b)
{
case Left: if(p.x < wmin.x) return (FALSE); break;
case Right: if(p.x > wmax.x) return (FALSE);
break; case bottom: if(p.y < wmin.y) return
(FALSE); break; case top: if(p.y > wmax.y)
return (FALSE); break;
}
return (TRUE);
}

int cross(wcPt2 p1, wcPt2 p2, Edge b, dcPt wmin, dcPt wmax)
{
if(inside(p1, b, wmin, wmax) == inside(p2, b, wmin, wmax))
return (FALSE);
else
return (TRUE);
}

wcPt2 (wcPt2 p1, wcPt2 p2, int b, dcPt wmin, dcPt wmax )
{
wcPt2
ipt;
float m;
if(p1.x != p2.x)
m = (p1.y - p2.y) / (p1.x - p2.x);
switch(b)
{
case Left:
ipt.x = wmin.x;
ipt.y = p2.y + (wmin.x - p2.x) * m; break;
case Right:
ipt.x = wmax.x;
ipt.y = p2.y + (wmax.x - p2.x) * m; break;
case Bottom:
ipt.y = wmin.y;
if(p1.x != p2.x)
ipt.x = p2.x + (wmin.y - p2.y) / m; else

```

```

ipt.
x=p
2.x;
bre
ak;
cas
e
Top:
ipt.y=wmax.
y;
if(p1.x!=p2.
x)
ipt.x=p2.x+(wmax.y-
p2.y)/m; else
ipt.x=
p2.x;
break;
}
return(ipt);
}
void clippoint(wcPt2 p,Edge b,dcPt wmin,dcPt wmax, wcPt2 *pout,int *cnt, wcPt2
*first[],struct point *s)
{
wcPt
2 ipt;
if(!fir
st[b])
first[
b]=&
p;
else
if(cross(p,s[b],b,wmin,wmax))
{
ipt=intersect(p,s[b],b,wmin,wmax);
if(b<top)
clippoint(ipt,b+1,wmin,wmax,pout,cnt,first,
s); else
{
pout[*cnt]=
ipt;
(*cnt)++;
}
}
s[b]=p;
if(inside(p,b,wmin,wmax))
if(b<top)
clippoint(p,b+1,wmin,wmax,pout,cnt,first,s)
; else
{

```



```

pout[*cnt]=
p; (*cnt)++;
}
}
void closeclip(dcPt wmin,dcPt wmax, wcPt2 *pout,int *cnt,wcPt2 *first[], wcPt2 *s)
{
wc
Pt2
iPt;
Ed
ge
b;
for(b=left;b<=top;b++)
{
if(cross(s[b],*first[b],b,wmin,wmax))
{
i=intersect(s[b],*first[b],b,wmin,wmax);
if(b<top)
clippoint(i,b+1,wmin,wmax,pout,cnt,first,s)
; else
{
pout[*cnt]=
i; (*cnt)++;
}
}
}
}
int clippolygon(dcPt point wmin,dcPt wmax,int n,wcPt2 *pin, wcPt2 *pout)
{
wcPt2
*first[N_EDGE]={0,0,0,0},s[N_EDGE]; int
i,cnt=0;
for(i=0;i<n;i++)
clippoint(pin[i],left,wmin,wmax,pout,&cnt,first,s)
; closeclip(wmin,wmax,pout,&cnt,first,s);
return(cnt);
}
}

```

7. Explain in detail about *Weiler- Atherton Polygon Clipping*

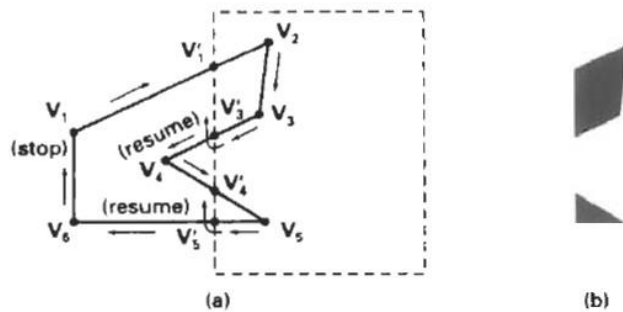
This clipping procedure was developed as a method for identifying visible surfaces, and so it can be applied with arbitrary polygon-clipping regions.

The basic idea in this algorithm is that instead of always proceeding around the polygon edges as vertices are processed, we sometimes want to follow the window boundaries. Which path we follow depends on the polygon-processing direction (clockwise or counterclockwise) and whether the pair of polygon vertices currently being processed represents an outside-to-inside pair or an inside- to-outside pair. For clockwise processing of polygon vertices, we use the following rules:

- For an outside-to-inside pair of vertices, follow the polygon boundary.
- For an inside-to-outside pair of vertices, follow the window boundary in a

clockwise direction.

In the below Fig. the processing direction in the Weiler-Atherton algorithm and the resulting clipped polygon is shown for a rectangular clipping window.



An improvement on the Weiler-Atherton algorithm is the Weiler algorithm, which applies constructive solid geometry ideas to clip an arbitrary polygon against any polygon clipping region.

UNIT III THREE-DIMENSIONAL GRAPHICS

PART A

1. Categorize the 3D representations?
Boundary representation (B-reps) and space-partitioning representations.
2. What Boundary representation?
It describes a 3D object as a set of surfaces that separate the object interior from the environment. e.g. polygon facets and spline patches.
3. What space-partitioning representation?
This is used to describe interior properties, by partitioning the spatial region containing an object in to a set of small, non-overlapping, contiguous solids. e.g. octree.
4. What is Blobby Object?
Some objects do not maintain a fixed shape, but change their surface characteristics in certain motions or when in proximity to other objects. Examples in this class of objects include molecular structures, water droplets and other liquid effects, melting objects and muscle shapes in the human body. These objects can be described as exhibiting "blobbiness" and are often simply referred to as blobby objects, since their shapes show a certain degree of fluidity.
5. What is projection?
The process of displaying 3D objects on a 2D display is called as Projection
6. What are the types of projection?
 - Perspective projection
 - Parallel projection
7. What is parallel projection?
In a parallel projection, coordinate positions are transformed to the view plane along parallel lines.
8. What is Perspective projection?
For a perspective projection object positions are transformed to the view plane along lines that converge to a point called the projection reference point.
9. What is chromaticity?
The term chromaticity is used to refer collectively to the two properties describing color characteristics: Purity and dominant frequency.
10. Define Color model.
A Color model is a method for explaining the properties or behavior of color within some particular context.
11. What are the uses of chromaticity diagram?
The chromaticity diagram is useful for the following:
 - Comparing color gamuts for different sets of primaries.
 - Identifying complementary colors.
 - Determining dominant wavelength and purity of a given color.

12. Give the transformation matrix for conversion of RGB to YIQ.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

13. What is HSV model?

The HSV(Hue, Saturation, Value) model is a color model which uses color descriptions that have a more intuitive appeal to a user. To give a color specification, a user selects a spectral color and the amounts of white and black that are to be added to obtain different shades, tint, and tones.

14. What for CMY color model used?

A color model defined with the primary colors cyan, magenta, and yellow is useful for describing color output to hard-copy devices.

15. What are the parameters in the HLS color model?

Hue, Lightness and Saturation.

16. Define Computer animation.

Computer animation refers to any time sequence of visual changes in a scene. In addition to changing object position with translations or rotations, a computer generated animation could display time variations in object size, color, transparency, or surface texture.

17. What are the steps in animation sequence?

- Story board layout
- Object definition
- Key-frame specifications
- Generation of in-between frames

18. How frame-by-frame animation works?

Here each frame of the scene is separately generated and stored. Later the frames can be recorded on film or they can be consecutively displayed in "real-time playback" mode.

19. What is morphing?

Morphing is the transformation that results in smooth interpolation between any two objects. It transforms object shape from one form to another

20. What are the methods of motion specifications?

- Direct motion specification
- Goal-directed Systems
- Kinematics and Dynamics.

21. Differentiate augmented reality and

virtual reality

Is the creation of a 3D environment which simulates real world, where the users can interact with.

VR is usually achieved by the wearing of a VR helmet or Google glasses

Augmented reality

Augmented reality blends virtual reality with the real world while interacting touch with the real world while interacting

a VR helmet or Google glasses

22. How will you represent sphere in three dimension?

A circle in 2D space is the collection of points with fixed distance from a given point in the plane. And this fact allows us to write simple algebraic equations for circles based on the distance formula. Similarly

A sphere in 3D space is the collection of points with fixed distance from a given point in space. And we use the formula for the distance between two points in 3D space to write algebraic equations for spheres.

23. Mention the properties of Bezier curve.

i. End point interpolation

Bezier curve with control points $P_0, P_1, P_2 \dots P_n$ will always interpolate P_0 and P_n and does not pass through all control points.

ii. Convex Hull

Bezier curve will always lie within convex hull.

$$\sum_{k=0}^n B_k, n(u) = 1$$

24. Differentiate Orthographic and oblique parallel projection

Orthographic parallel projection

oblique parallel projection

Projection is parallel if object is projected on view plane using parallel line. In orthogonal parallel projection, projection plane is not perpendicular to view plane

In oblique parallel projection, view plane projection plane is perpendicular to view plane

25. How do we identify the principle vanishing point?

Vanishing point is a point in the image plane where the projected set of parallel lines appears to intersect in space. If the projected set of parallel lines appears to meet at infinity, then the vanishing point is called principle vanishing point

PART B

1. Explain various 3D object representations.

Representation schemes for solid objects are divided into two categories as follows:

1. Boundary Representation (B-reps)

It describes a three dimensional object as a set of surfaces that separate the object interior from the environment. Examples are polygon facets and spline patches.

2. Space Partitioning representation

It describes the interior properties, by partitioning the spatial region containing an object into a set of small, nonoverlapping, contiguous solids(usually cubes).

Eg: Octree Representation

Polygon Surfaces

Curved Lines and Surfaces

Quadric Surfaces

Spline Representations

2. Explain 3D Transformations.

Geometric transformations and object modeling in three dimensions are extended from two-dimensional methods by including considerations for the z-coordinate.

2.1.1 Translation

In a three dimensional homogeneous coordinate representation, a point or an object is translated from position $P = (x, y, z)$ to position $P' = (x', y', z')$ with the matrix operation.

$$(1) \quad \begin{array}{ccccccc} x' & & 1 & 0 & 0 & t_x & x \\ y' & = & 0 & 1 & 0 & t_y & y \\ z' & & 0 & 0 & 1 & t_z & z \\ 1 & & 0 & 0 & 0 & 1 & 1 \end{array} \quad \text{-----}$$

$$(or) \quad P' = T.P \text{-----} (2)$$

Parameters t_x , t_y and t_z specifying translation distances for the coordinate directions x, y and z are assigned any real values.

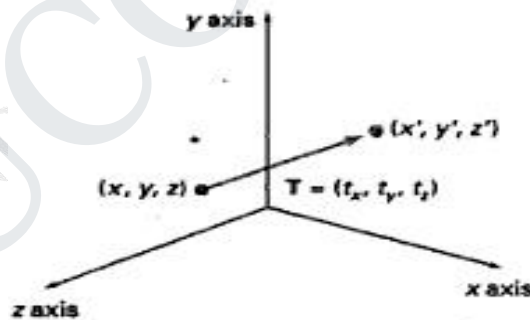
The matrix representation in equation (1) is equivalent to the three equations

$$x' = x + t_x$$

$$y' = y + t_y$$

$$z' = z + t_z \text{-----} (3)$$

Translating a point with translation vector $T = (t_x, t_y, t_z)$



Inverse of the translation matrix in equation (1) can be obtained by negating the translation distance t_x , t_y and t_z .

This produces a translation in the opposite direction and the product of a translation matrix and its inverse produces the identity matrix.

2.1.2 Rotation

- To generate a rotation transformation for an object an axis of rotation must be designed to rotate the object and the amount of angular rotation is also be specified.
- Positive rotation angles produce counter clockwise rotations about a coordinate axis.

Co-ordinate Axes Rotations

The 2D z axis rotation equations are easily extended to 3D. $x' = x \cos \theta - y \sin \theta$

$$y' = x \sin \theta + y \cos \theta$$

$$z' = z$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2)$$

Parameters θ specifies the rotation angle. In homogeneous coordinate form, the 3D z axis rotation equations are expressed as

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3)$$

which we can write more compactly as

$$\mathbf{P}' = \mathbf{R}_z(\theta) \cdot \mathbf{P} \quad (4)$$

The below figure illustrates rotation of an object about the z axis.

Transformation equations for rotation about the other two coordinate axes can be obtained with a cyclic permutation of the

coordinate parameters x, y and z in equation (2) i.e., we use the replacements

$$x \rightarrow y \rightarrow z \rightarrow x \text{-----} (5)$$

Substituting permutations (5) in equation (2), we get the equations for an x-axis rotation

$$\begin{aligned} y' &= y \cos \theta - z \sin \theta \\ z' &= y \sin \theta + z \cos \theta \text{-----} (6) \\ x' &= x \end{aligned}$$

which can be written in the homogeneous coordinate form

$$\begin{array}{cccccc} x' & & 1 & & 0 & & 0 & & 0 & & x \\ y' & = & 0 & & \cos \theta & & -\sin \theta & & 0 & & y \\ z' & & 0 & & \sin \theta & & \cos \theta & & 0 & & z \\ 1 & & 0 & & 0 & & 0 & & 1 & & 1 \end{array} \text{-----} (7)$$

$$\text{(or)} \quad \mathbf{P}' = \mathbf{R}_x(\theta) \cdot \mathbf{P} \text{-----} (8)$$

Rotation of an object around the x-axis is demonstrated in the below fig

Cyclically permuting coordinates in equation (6) give the transformation equation for a y axis rotation.

$$\begin{aligned} z' &= z \cos \theta - x \sin \theta \\ x' &= z \sin \theta + x \cos \theta \text{-----} (9) \\ y' &= y \end{aligned}$$

The matrix representation for y-axis rotation is

$$\begin{array}{cccc} & & \cos \theta & \\ & & 0 & \sin \theta \\ & & 0 & 0 \\ & & x & \end{array}$$

$$\begin{array}{rcl}
 y' & = & \begin{bmatrix} 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ z \\ 1 \end{bmatrix} \quad \text{-----(10)}
 \end{array}$$

$$(or) \quad \mathbf{P}' = \mathbf{R}_y(\theta) \cdot \mathbf{P} \quad \text{-----(11)}$$

An example of y axis rotation is shown in below figure

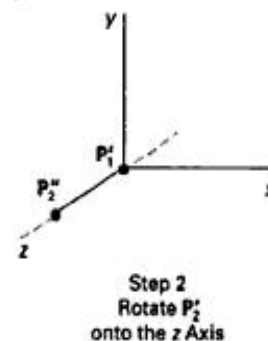
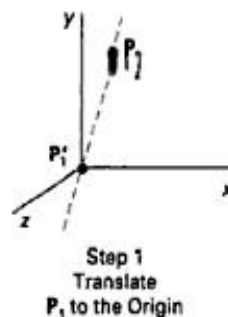
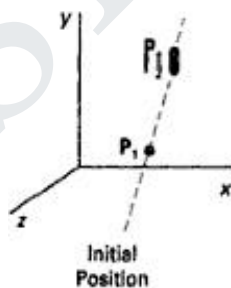
- An inverse rotation matrix is formed by replacing the rotation angle θ by $-\theta$.
- Negative values for rotation angles generate rotations in a clockwise direction, so the identity matrix is produced when any rotation matrix is multiplied by its inverse.
- Since only the sine function is affected by the change in sign of the rotation angle, the inverse matrix can also be obtained by interchanging rows and columns. (i.e.,) we can calculate the inverse of any rotation matrix R by evaluating its transpose ($R^{-1} = R^T$).

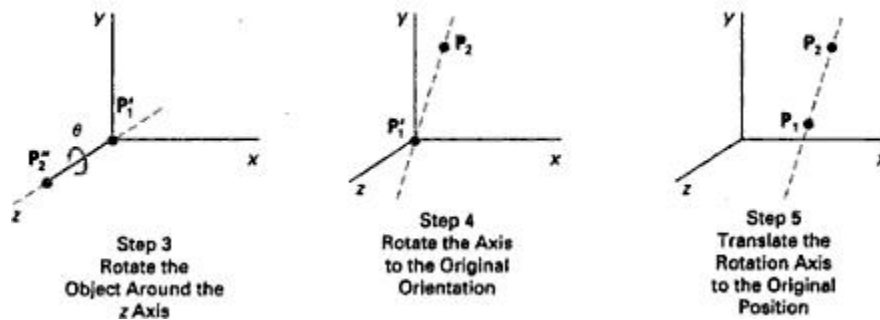
General Three Dimensional Rotations

- A rotation matrix for any axis that does not coincide with a coordinate axis can be set up as a composite transformation involving combinations of translations and the coordinate axes rotations.
- We obtain the required composite matrix by
 1. Setting up the transformation sequence that moves the selected rotation axis onto one of the coordinate axis.
 2. Then set up the rotation matrix about that coordinate axis for the specified rotation angle.

3. Obtaining the inverse transformation sequence that returns the rotation axis to its original position.
- In the special case where an object is to be rotated about an axis that is parallel to one of the coordinate axes, we can attain the desired rotation with the following transformation sequence:
 1. Translate the object so that the rotation axis coincides with the parallel coordinate axis.
 2. Perform the specified rotation about the axis.
 3. Translate the object so that the rotation axis is moved back to its original position.
 - When an object is to be rotated about an axis that is not parallel to one of the coordinate axes, we need to perform some additional transformations.
 - In such case, we need rotations to align the axis with a selected coordinate axis and to bring the axis back to its original orientation.
 - Given the specifications for the rotation axis and the rotation angle, we can accomplish the required rotation in five steps:
 1. Translate the object so that the rotation axis passes through the coordinate origin.
 2. Rotate the object so that the axis of rotation coincides with one of the coordinate axes.
 3. Perform the specified rotation about that coordinate axis.
 4. Apply inverse rotations to bring the rotation axis back to its original orientation.
 5. Apply the inverse translation to bring the rotation axis back to its original position.

Five transformation steps





2.1.3 Scaling

- The matrix expression for the scaling transformation of a position $P = (x,y,z)$ relative to the coordinate origin can be written as

$$\begin{array}{rcl} x' & = & s_x x \\ y' & = & s_y y \\ z' & = & s_z z \\ 1 & & 1 \end{array} \quad \text{-----(11)}$$

(or) $\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$ ----- (12)

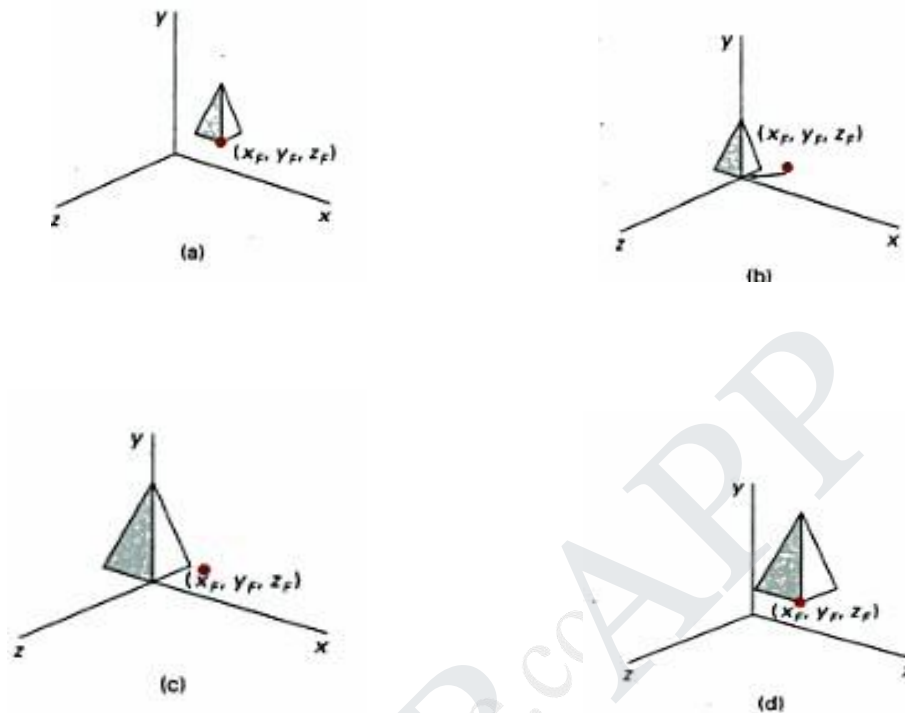
where scaling parameters s_x , s_y , and s_z are assigned any position values.

- Explicit expressions for the coordinate transformations for scaling relative to the origin are

$$\begin{array}{l} x' = x \cdot s_x \\ y' = y \cdot s_y \text{----- (13)} \\ z' = z \cdot s_z \end{array}$$

- Scaling an object changes the size of the object and repositions the object relative to the coordinate origin.
- If the transformation parameters are not equal, relative dimensions in the object are changed.
- The original shape of the object is preserved with a uniform scaling ($s_x = s_y = s_z$).
- Scaling with respect to a selected fixed position (x_f, y_f, z_f) can be represented with the following transformation sequence:
 1. Translate the fixed point to the origin.
 2. Scale the object relative to the coordinate origin using Eq.11.

3. Translate the fixed point back to its original position. This sequence of transformation is shown in the below figure .



- The matrix representation for an arbitrary fixed point scaling can be expressed as the concatenation of the **translate-scale-translate** transformation are

$$T(x_f, y_f, z_f) \cdot S(s_x, s_y, s_z) \cdot T(-x_f, -y_f, -z_f) =$$

$$\begin{array}{cccc} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{array} \quad \text{-----(14)}$$

- Inverse scaling matrix m formed by replacing the scaling parameters s_x , s_y and s_z with their reciprocals.
- The inverse matrix generates an opposite scaling transformation, so the concatenation of any scaling matrix and its inverse produces the **identity matrix**.

3. Explain 3D viewing.

In three dimensional graphics applications,

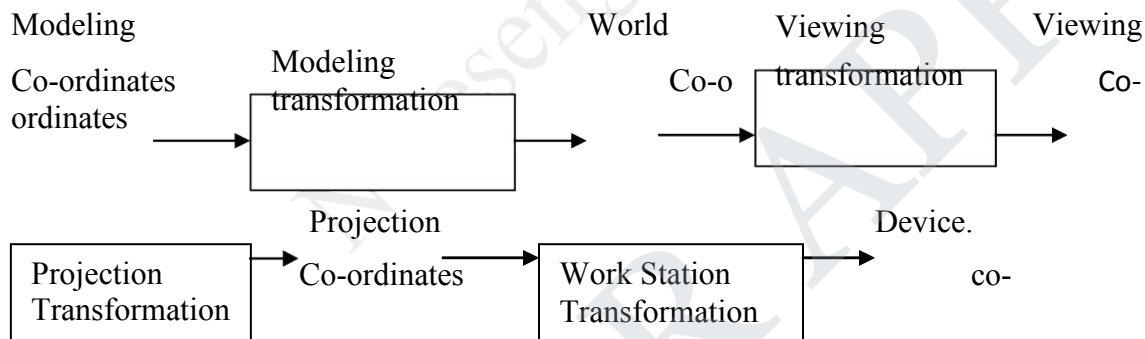
- we can view an object from any spatial position, from the front, from above or from the back.
- We could generate a view of what we could see if we were standing in the middle of a group of objects or inside object, such as a building.

2.1.4 Viewing Pipeline:

In the view of a three dimensional scene, to take a snapshot we need to do the following steps.

1. Positioning the camera at a particular point in space.
2. Deciding the camera orientation (i.e.,) pointing the camera and rotating it around the line of sight to set up the direction for the picture.
3. When snap the shutter, the scene is cropped to the size of the 'window' of the camera and light from the visible surfaces is projected into the camera film.

In such a way the below figure shows the three dimensional transformation pipeline, from modeling coordinates to final device coordinate.



Processing Steps

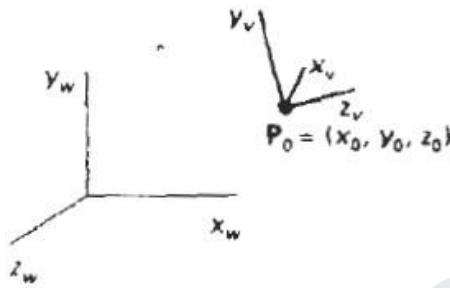
1. Once the scene has been modeled, world coordinates position is converted to viewing coordinates.
2. The viewing coordinates system is used in graphics packages as a reference for specifying the observer viewing position and the position of the projection plane.
3. Projection operations are performed to convert the viewing coordinate description of the scene to coordinate positions on the projection plane, which will then be mapped to the output device.

4. Objects outside the viewing limits are clipped from further consideration, and the remaining objects are processed through visible surface identification and surface rendering procedures to produce the display within the device viewport.

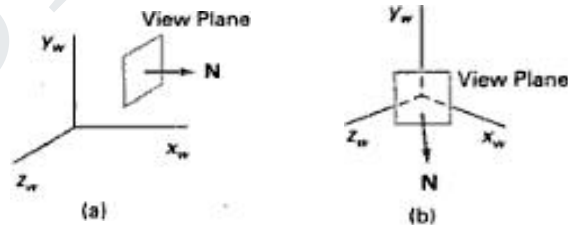
2.1.5 Viewing Coordinates Specifying

the view plane

- The view for a scene is chosen by establishing the viewing coordinate system, also called the **view reference coordinate system**.

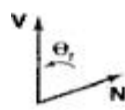


- A **viewplane** or **projection plane** is set-up perpendicular to the viewing Z_v axis.
- World coordinate positions in the scene are transformed to viewing coordinates, then viewing coordinates are projected to the view plane.
- The **view reference point** is a world coordinate position, which is the origin of the viewing coordinate system. It is chosen to be close to or on the surface of some object in a scene.
- Then we select the positive direction for the viewing Z_v axis, and the orientation of the view plane by specifying the **view plane normal vector, N**. Here the world coordinate position establishes the direction for N relative either to the world origin or to the viewing coordinate origin.



- Then we select the up direction for the view by specifying a vector V called the **view-up vector**. This vector is used to establish the positive direction for the y_v axis.

Specifying the view –up vector with a twist angle θ_t

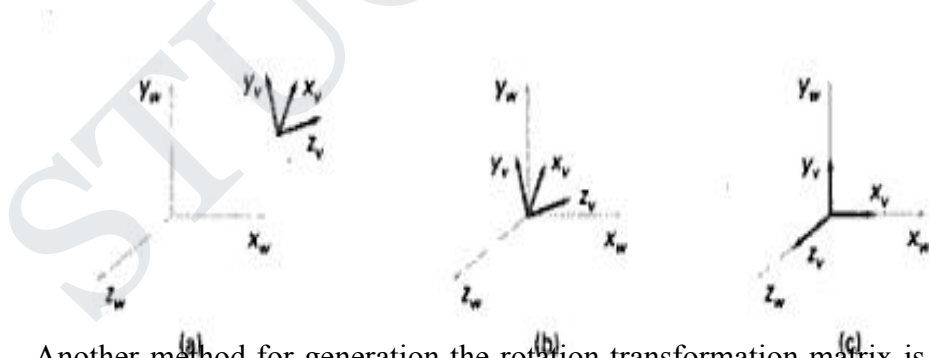


Transformation from world to viewing coordinates

- Before object descriptions can be projected to the view plane, they must be transferred to viewing coordinate. This transformation sequence is,
 1. Translate the view reference point to the origin of the world coordinate system.
 2. Apply rotations to align the x_v , y_v and z_v axes with the world x_w, y_w and z_w axes respectively.
- If the view reference point is specified at world position (x_0, y_0, z_0) this point is translated to the world origin with the matrix transformation.

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The rotation sequence can require up to 3 coordinate axis rotations depending on the direction chosen for N. Aligning a viewing system with the world coordinate axes using a sequence of translate – rotate transformations



- Another method for generation the rotation transformation matrix is to calculate unit vvn vectors and form the composite rotation matrix directly.

Given vectors N and V, these unit vectors are calculated as $n = N / (|N|)$

$$= (n_1, n_2, n_3)$$

•

$$u = (V * N) / (|V * N|) = (u_1, u_2, u_3) \quad v = n * u = (v_1, v_2, v_3)$$

- This method automatically adjusts the direction for v , so that v is perpendicular to n .
- The composite rotation matrix for the viewing transformation is

$$R = \begin{pmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

which transforms u into the world x_w axis, v onto the y_w axis and n onto the z_w axis.

- The complete world-to-viewing transformation matrix is obtained as the matrix product.
 $M_{wc,vc} = R.T$

This transformation is applied to coordinate descriptions of objects in the scene transfer them to the viewing reference frame.

4. Explain Color models.

Color Models

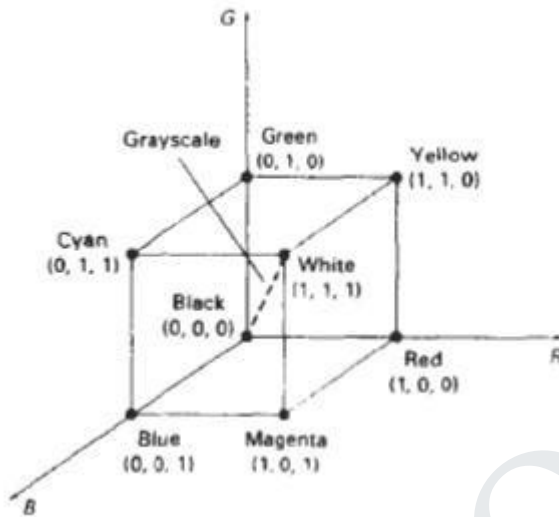
Color Model is a method for explaining the properties or behavior of color within some particular context. No single color model can explain all aspects of color, so we make use of different models to help describe the different perceived characteristics of color.

RGB Color Model

- Based on the tristimulus theory of vision, our eyes perceive color through the stimulation of three visual pigments in the cones on the retina.

•

- These visual pigments have a peak sensitivity at wavelengths of about 630 nm (red), 530 nm (green) and 450 nm (blue).
- By comparing intensities in a light source, we perceive the color of the light.
- This is the basis for displaying color output on a video monitor using the 3 color primaries, red, green, and blue referred to as the RGB color model. It is represented in the below figure.



- The sign represents black, and the vertex with coordinates (1,1,1) in white.
- Vertices of the cube on the axes represent the primary colors, the remaining vertices represents the complementary color for each of the primary colors.
- The RGB color scheme is an additive model. (i.e.,) Intensities of the primary colors are added to produce other colors.
- Each color point within the bounds of the cube can be represented as the triple (R,G,B) where values for R, G and B are assigned in the range from 0 to 1.
- The color C_λ is expressed in RGB component as C_λ

$$= R\mathbf{R} + G\mathbf{G} + B\mathbf{B}$$

- The magenta vertex is obtained by adding red and blue to produce the triple (1,0,1) and white at (1,1,1) in the sum of the red, green and blue vertices.
- Shades of gray are represented along the main diagonal of the cube from the origin (black) to the white vertex.

•

2.5.5 YIQ Color Model

- The National Television System Committee (NTSC) color model for forming the composite video signal in the YIQ model.
- In the YIQ color model, luminance (brightness) information is contained in the Y parameter, chromaticity information (hue and purity) is contained into the I and Q parameters.
- A combination of red, green and blue intensities are chosen for the Y parameter to yield the standard luminosity curve.
- Since Y contains the luminance information, black and white TV monitors use only the Y signal.
- Parameter I contains orange-cyan hue information that provides the flesh-tone shading and occupies a bandwidth of 1.5 MHz.
- Parameter Q carries green-magenta hue information in a bandwidth of about 0.6 MHz.
- An RGB signal can be converted to a TV signal using an NTSC encoder which converts RGB values to YIQ values, as follows

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & 0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

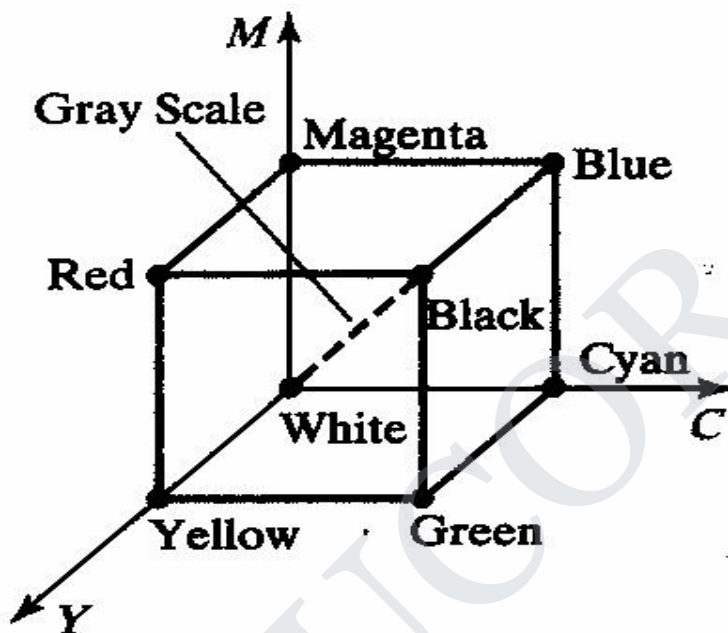
- An NTSC video signal can be converted to an RGB signal using an NTSC decoder which separates the video signal into YIQ components, then converts to RGB values, as follows:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.620 \\ 1.000 & 0.272 & 0.647 \\ 1.000 & -1.108 & -1.705 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

•

CMY Color Model

- A color model defined with the primary colors cyan, magenta, and yellow (CMY) in useful for describing color output to hard copy devices.
- It is a subtractive color model (i.e.,) cyan can be formed by adding green and blue light. When white light is reflected from cyan-colored ink, the reflected light must have no red component. i.e., red light is absorbed or subtracted by the ink.
- Magenta ink subtracts the green component from incident light and yellow subtracts the blue component.



- In CMY model, point (1,1,1) represents black because all components of the incident light are subtracted.
- The origin represents white light.
- Equal amounts of each of the primary colors produce grays along the main diagonal of the cube.
- A combination of cyan and magenta ink produces blue light because the red and green components of the incident light are absorbed.
- The printing process often used with the CMY model generates a color point with a collection of 4 ink dots; one dot is used for each of the primary colors (cyan, magenta and yellow) and one dot in black.

- The conversion from an RGB representation to a CMY representation is expressed as

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Where the white is represented in the RGB system as the unit column vector.

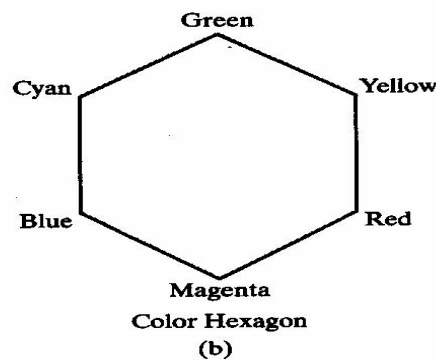
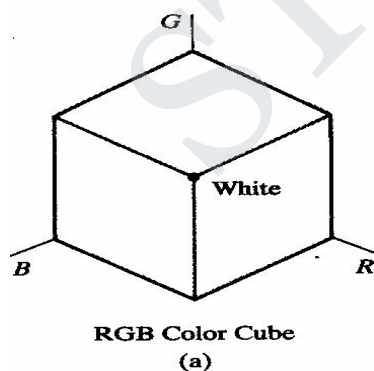
- Similarly the conversion of CMY to RGB representation is expressed as

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

Where black is represented in the CMY system as the unit column vector.

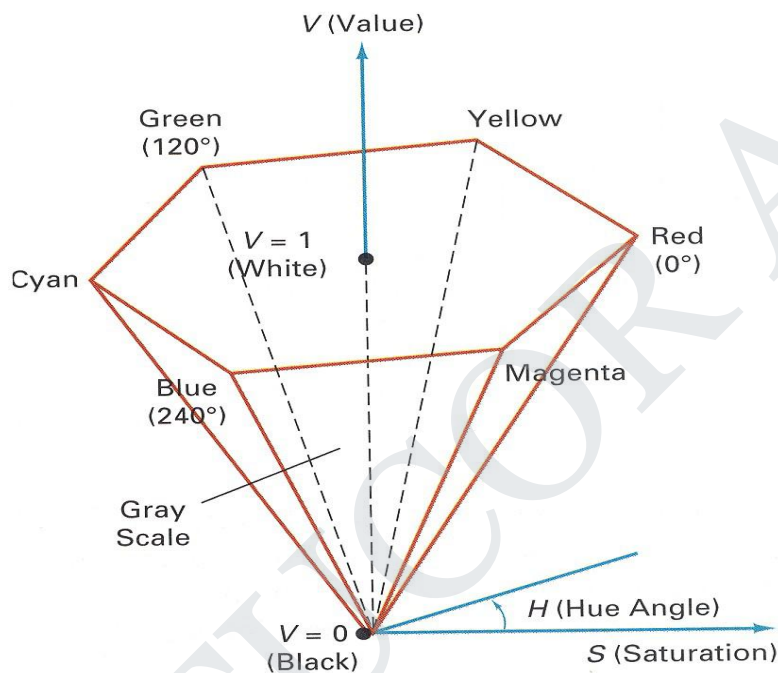
HSV Color Model

- The HSV model uses color descriptions that have a more interactive appeal to a user.
- Color parameters in this model are hue (H), saturation (S), and value (V).
- The 3D representation of the HSV model is derived from the RGB cube. The outline of the cube has the hexagon shape.



•

- The boundary of the hexagon represents the various hues, and it is used as the top of the HSV hexcone.
- In the hexcone, saturation is measured along a horizontal axis, and value is along a vertical axis through the center of the hexcone.
- Hue is represented as an angle about the vertical axis, ranging from 0° at red through 360° . Vertices of the hexagon are separated by 60° intervals.
- Yellow is at 60° , green at 120° and cyan opposite red at $H = 180^\circ$. Complementary colors are 180° apart.



- Saturation S varies from 0 to 1. the maximum purity at $S = 1$, at $S = 0.25$, the hue is said to be one quarter pure, at $S = 0$, we have the gray scale.
- Value V varies from 0 at the apex to 1 at the top.
 - the apex representation black.
- At the top of the hexcone, colors have their maximum intensity. •

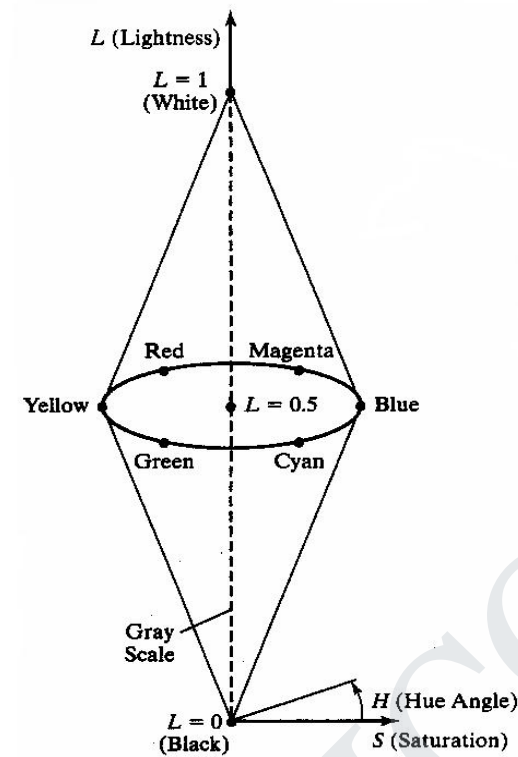
When $V = 1$ and $S = 1$ we have the „pure“ hues.

- White is the point at $V = 1$ and $S = 0$.

•

HLS Color Model

- HLS model is based on intuitive color parameters used by Tektronix.
- It has the double cone representation shown in the below figure. The 3 parameters in this model are called Hue (H), lightness (L) and saturation (s).



- Hue specifies an angle about the vertical axis that locates a chosen hue. • In this model $H = 0^\circ$ corresponds to Blue.
- The remaining colors are specified around the perimeter of the cone in the same order as in the HSV model.
- Magenta is at 60° , Red in at 120° , and cyan in at $H = 180^\circ$.
- The vertical axis is called lightness (L). At $L = 0$, we have black, and white is at $L = 1$ Gray scale in along the L axis and the “pure hues” on the $L = 0.5$ plane.
- Saturation parameter S specifies relative purity of a color. S varies from 0 to 1 pure hues are those for which $S = 1$ and $L = 0.5$

- As S decreases, the hues are said to be less pure.
- At $S = 0$, it is said to be gray scale.

5. Explain computer animation.

Computer animation refers to any time sequence of visual changes in a scene.

- Computer animations can also be generated by changing camera parameters such as position, orientation and focal length.
- Applications of computer-generated animation are entertainment, advertising, training and education.

Example : Advertising animations often transition one object shape into another.

Frame-by-Frame animation

Each frame of the scene is separately generated and stored. Later, the frames can be recoded on film or they can be consecutively displayed in "real-time playback" mode

Design of Animation Sequences

An animation sequence is designed with the following steps: •

Story board layout

- Object definitions
- Key-frame specifications
- Generation of in-between frames.

Story board

- The story board is an outline of the action.
- It defines the motion sequences as a set of basic events that are to take place.
- Depending on the type of animation to be produced, the story board could consist of a set of rough sketches or a list of the basic ideas for the motion.

Object Definition

- An object definition is given for each participant in the action.
-

- Objects can be defined in terms of basic shapes such as polygons or splines. •

The associated movements of each object are specified along with the shape.

Key frame

- A key frame is detailed drawing of the scene at a certain time in the animation sequence.
- Within each key frame, each object is positioned according to the time for that frame.
- Some key frames are chosen at extreme positions in the action; others are spaced so that the time interval between key frames is not too much.

In-betweens

- In betweens are the intermediate frames between the key frames.
- The number of in between needed is determined by the media to be used to display the animation.
- Film requires 24 frames per second and graphics terminals are refreshed at the rate of 30 to 60 frames per seconds.
- Time intervals for the motion are setup so there are from 3 to 5 in-between for each pair of key frames.
- Depending on the speed of the motion, some key frames can be duplicated. • For a 1 min film sequence with no duplication, 1440 frames are needed.
- Other required tasks are
 - Motion verification
 - Editing
 - Production and synchronization of a sound track.

General Computer Animation Functions

Steps in the development of an animation sequence are,

1. Object manipulation and rendering

-

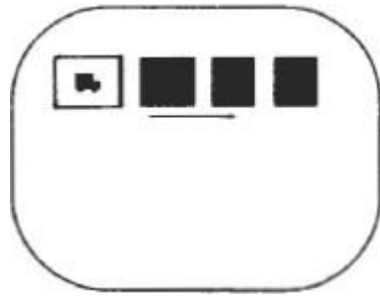
2. Camera motion
3. Generation of in-betweens

- Animation packages such as wave front provide special functions for designing the animation and processing individuals objects.
- Animation packages facilitate to store and manage the object database.
- Object shapes and associated parameter are stored and updated in the database.
- Motion can be generated according to specified constraints using 2D and 3D transformations.
- Standard functions can be applied to identify visible surfaces and apply the rendering algorithms.
- Camera movement functions such as zooming, panning and tilting are used for motion simulation.
- Given the specification for the key frames, the in-betweens can be automatically generated.

Raster Animations

- On raster systems, real-time animation in limited applications can be generated using raster operations.
- Sequence of raster operations can be executed to produce real time animation of either 2D or 3D objects.
- We can animate objects along 2D motion paths using the color-table transformations.
 - Predefine the object as successive positions along the motion path, set the successive blocks of pixel values to color table entries.
 - Set the pixels at the first position of the object to „on“ values, and set the pixels at the other object positions to the background color.
 - The animation is accomplished by changing the color table values so that the object is „on“ at successive positions along the animation path as the preceding position is set to the background intensity.

-



6. Explain Parallel projection in detail

a. Projections

- Once world coordinate descriptions of the objects are converted to viewing coordinates, we can project the 3 dimensional objects onto the two dimensional view planes.

- Parallel Projection** - Here the coordinate positions are transformed to the view plane along parallel lines.

Parallel projection of an object to the view plane

Parallel Projections

- Parallel projections are specified with a **projection vector** that defines the direction for the projection lines.
- When the projection is perpendicular to the view plane, it is said to be an **Orthographic parallel projection**, otherwise it said to be an **Oblique parallel projection**.

Orientation of the projection vector V_p to produce an orthographic projection (a) and an oblique projection (b)

7. Explain about perspective projection method

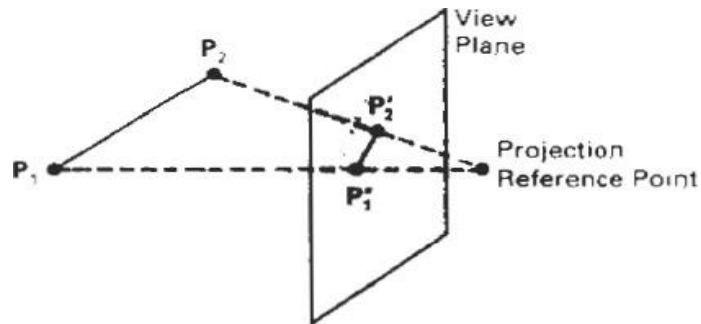
a. Projections

- Once world coordinate descriptions of the objects are converted to viewing coordinates, we can project the 3 dimensional objects onto the two dimensional view planes.

 - Perspective Projection** – Here, object positions are transformed to the view plane along lines that converge to a point called the **projection reference point**.

Perspective projection of an object to the view plane

•



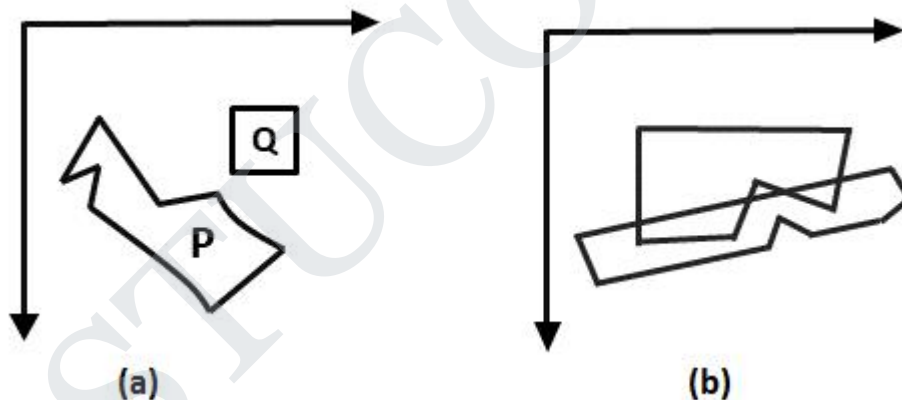
8. How do you represent 3D objects using Octrees?

Painter Algorithm

It came under the category of list priority algorithm. It is also called a **depth-sort algorithm**. In this algorithm ordering of visibility of an object is done. If objects are reversed in a particular order, then correct picture results.

Objects are arranged in increasing order to z coordinate. Rendering is done in order of z coordinate. Further objects will obscure near one. Pixels of rear one will overwrite pixels of farther objects. If z values of two overlap, we can determine the correct order from Z value as shown in fig (a).

If z objects overlap each other as in fig (b) this correct order can be maintained by splitting of objects.



Depth sort algorithm or painter algorithm was developed by Newell, Sancha. It is called the painter algorithm because the painting of frame buffer is done in decreasing order of distance. The distance is from view plane. The polygons at more distance are painted firstly.

The concept has taken color from a painter or artist. When the painter makes a painting, first of all, he will paint the entire canvas with the background color. Then more distance objects like mountains, trees are added. Then rear or foreground objects are added to picture. Similar approach we will use. We will sort surfaces according to z values. The z values are stored in the refresh buffer.

Steps performed in-depth sort

1. Sort all polygons according to z coordinate.
2. Find ambiguities of any, find whether z coordinate overlap, split polygon if necessary.
3. Scan convert each polygon in increasing order of z coordinate.

Painter Algorithm

Step1: Start Algorithm

Step2: Sort all polygons by z value keep the largest value of z first.

Step3: Scan converts polygons in this order.
Test is applied

1. Does A is behind and non-overlapping B in the dimension of Z as shown in fig (a)
2. Does A is behind B in z and no overlapping in x or y as shown in fig (b)
3. If A is behind B in Z and totally outside B with respect to view plane as shown in fig (c)
4. If A is behind B in Z and B is totally inside A with respect to view plane as shown in fig (d)

The success of any test with single overlapping polygon allows F to be painted.

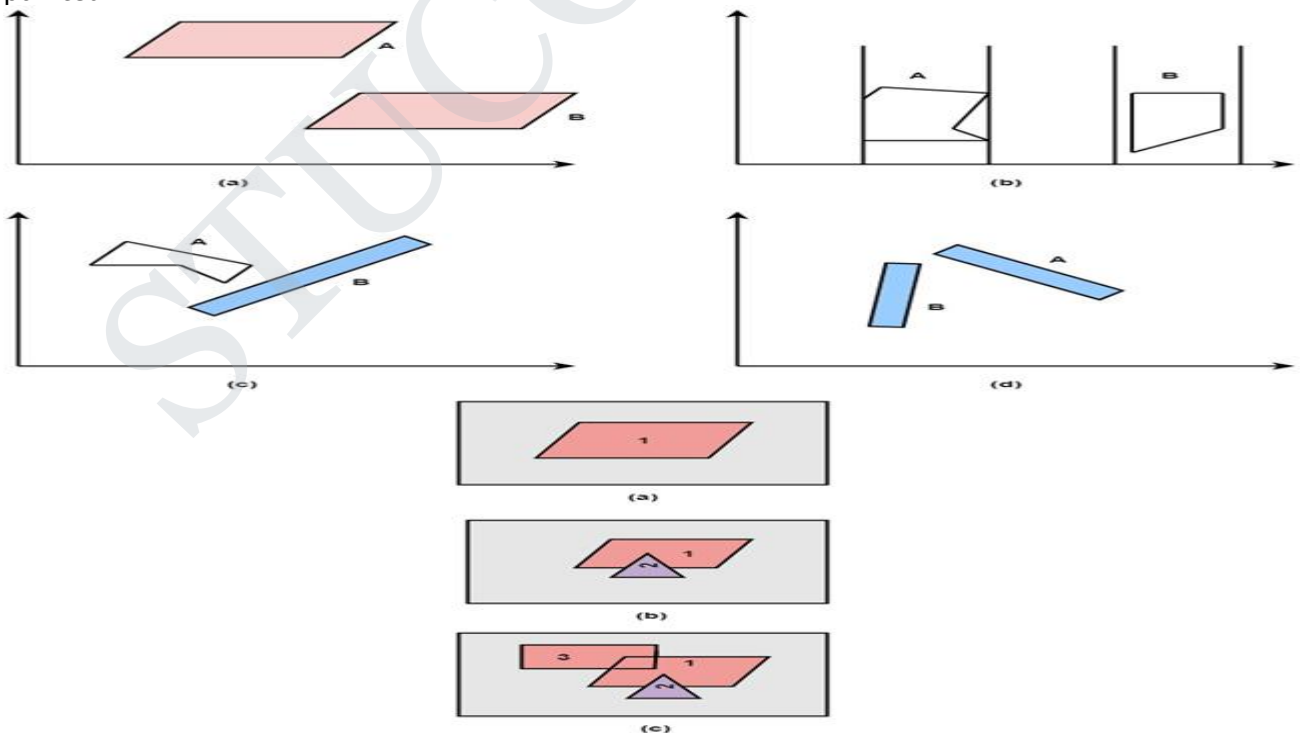


Figure showing addition of surface one by one and then painting done using painter algorithm

9. Explain the following visible surface detection algorithms

Painter Algorithm

Step1: Start Algorithm

Step2: Sort all polygons by z value keep the largest value of z first.

Step3: Scan converts polygons in this order.
Test is applied

1. Does A is behind and non-overlapping B in the dimension of Z as shown in fig (a)
2. Does A is behind B in z and no overlapping in x or y as shown in fig (b)
3. If A is behind B in Z and totally outside B with respect to view plane as shown in fig (c)
4. If A is behind B in Z and B is totally inside A with respect to view plane as shown in fig (d)

The success of any test with single overlapping polygon allows F to be painted.

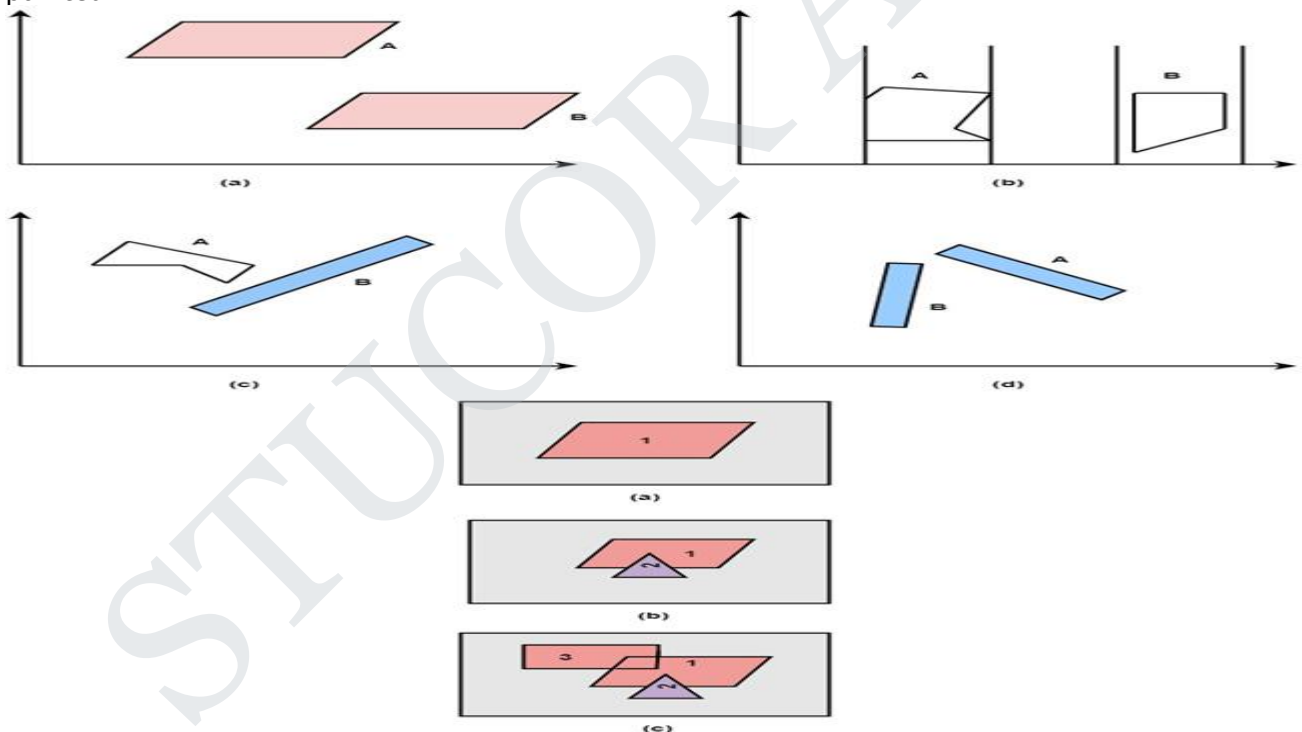


Figure showing addition of surface one by one and then painting done using painter algorithm

2.8.1 Depth Buffer Method

A commonly used image-space approach to detecting visible surfaces is the depth-buffer method, which compares surface depths at each pixel position on the projection plane. This procedure is also referred to as the z-buffer method.

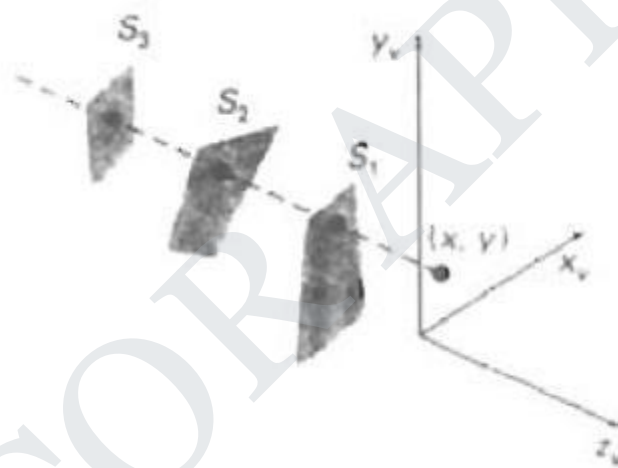
Each surface of a scene is processed separately, one point at a

time across the surface. The method is usually applied to scenes containing only polygon surfaces, because depth values can be computed very quickly and the method is easy to implement. But the method can be applied to nonplanar surfaces.

With object descriptions converted to projection coordinates, each (x, y, z) position on a polygon surface corresponds to the orthographic projection point (x, y) on the view plane.

Therefore, for each pixel position (x, y) on the view plane, object depths can be compared by comparing z values. The figure shows

three surfaces at varying distances along the orthographic projection line from position (x, y) in a view plane taken as the (x_v, y_v) plane. Surface S_1 , is closest at this position, so its surface intensity value at (x, y) is saved.



We can implement the depth-buffer algorithm in normalized coordinates, so that z values range from 0 at the back clipping plane to Z_{\max} at the front clipping plane.

Two buffer areas are required. A **depth buffer** is used to store depth values for each (x, y) position as surfaces are processed, and the **refresh buffer** stores the intensity values for each position.

Initially, all positions in the depth buffer are set to 0 (minimum depth), and the refresh buffer is initialized to the background intensity.

We summarize the **steps of a depth-buffer algorithm as follows**:

1. Initialize the depth buffer and refresh buffer so that for all buffer positions (x, y) ,

$$\text{depth}(x, y) = 0, \quad \text{refresh}(x, y) = I_{\text{backgnd}}$$

2. For each position on each polygon surface, compare depth values to previously stored values in the depth buffer to determine visibility.

- Calculate the depth z for each (x, y) position on the polygon.
- If $z > \text{depth}(x, y)$, then set

$$\text{depth}(x, y) = z, \quad \text{refresh}(x, y) = I_{\text{surf}}(x, y)$$

where I_{backgnd} is the value for the background intensity, and $I_{\text{surf}}(x, y)$ is the projected intensity value for the surface at pixel position (x, y) . After all surfaces have been processed, the depth buffer contains depth values for the visible surfaces and the refresh buffer contains the corresponding intensity values for those surfaces.

Depth values for a surface position (x, y) are calculated from the plane equation for each surface:

$$z = \frac{-Ax - By - D}{C} \quad (1)$$

For any scan line adjacent horizontal positions across the line differ by 1, and a vertical y value on an adjacent scan line differs by 1. If the depth of position (x, y) has been determined to be z , then the depth z' of the next position $(x + 1, y)$ along the scan line is obtained from Eq. (1) as

$$z' = \frac{A(x + 1) - By - D}{-C} \quad (2)$$

$$\text{Or } z' - z = \frac{A}{-C} \quad (3)$$

On each scan line, we start by calculating the depth on a left edge of the polygon that intersects that scan line in the below fig. Depth values at each successive position across the scan line are then calculated by Eq. (3).

Scan lines intersecting a polygon surface

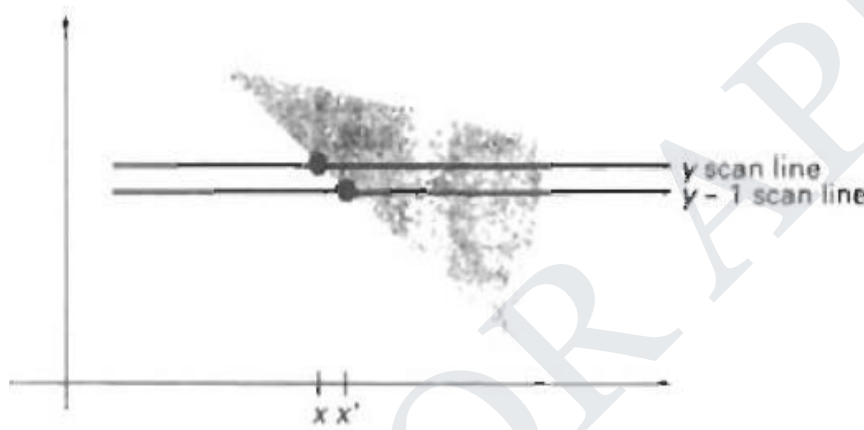
•

We first determine the y-coordinate extents of each polygon, and process the surface from the topmost scan line to the bottom scan line. Starting at a top vertex, we can recursively calculate x positions down a left edge of the polygon as $x' = x - 1/m$, where m is the slope of the edge.

Depth values down the edge are then obtained recursively as

$$z' = z + \frac{A/m + B}{C} \text{-----} (4)$$

Intersection positions on successive scan lines along a left polygon edge



If we are processing down a vertical edge, the slope is infinite and the recursive calculations reduce to

$$z' = z + \frac{B}{C} \text{-----} (5)$$

•

UNIT IV MULTIMEDIA SYSTEM DESIGN & MULTIMEDIA FILE HANDLING PART A

1. Give some Multimedia applications.
 - Document imaging
 - Image processing and Image recognition
 - Full-Motion Digital Video Applications
 - Electronic Messaging
2. What are the multimedia elements?
Facsimile, Document images, Photographic images, Geographical information system maps, Voice commands and voice synthesis, Audio messages, Video messages, Full-motion stored and live video, Holographic images, Fractals.
3. What is Holography?
It is defined as the means of creating a unique photographic image without the use of a lens.
4. What is hologram?
The photographic recoding of the image is called a hologram, which appears to be an unrecognizable pattern of stripes and whorls but which when illuminate by coherent light as by a laser beam, organizes the light in to 3D representation of the original object.
5. What are the important processes in image processing?
Image recognition, image enhancement, image synthesis, and image reconstruction.
6. What are complex image enhancement capabilities?
Image calibration, Real-time alignment, Gray-scale normalization, RGB hue intensity adjustment, Color separation, Frame averaging.
7. What is VGA mixing?
Here, the image acquisition memory also serves as the display source memory, thereby fixing its position and size on screen.
8. What is Dual-buffered VGA mixing / scaling?
Double buffer schemes maintain the original images in a decompression buffer and the resized image in a display buffer.
9. What is hypermedia documents?
In hypermedia documents in addition to text, embedded or linked multimedia objects such as image, audio, hologram, or full -motion video.
10. What are the sub-systems in DSP?

Memory management, hardware-interrupt handling, Multitasking, Inter task synchronization and communication, Multiple timer services, Device-independent I/O.

11. What are the types of images based on multimedia?
Visible images, non-visible images, abstract images.
12. What does non-visible images refer?
Non-visible images are those that are not stored as images but are displayed as images e.g. pressure gauges, temperature gauges.
13. What are abstract images?
Abstract images are really not images that ever existed as real-world objects or representations. Rather they are computer-generated images based on some arithmetic calculations. e.g. fractals.
14. What is DVI?
The Digital Video Interface (DVI) standard was defined to provide a processor independent specification for a video interface that could accommodate most compression algorithms for fast multimedia displays.
15. What is MIDI?
This is the interface standard for file transfer between a computer and a musical instrument such as a digital piano.
16. What is Apple's Quick time?
The QuickTime standard, developed by Apple Computer, is designed to support multimedia applications. It is viewed as a multimedia interface that is evolving to become a standard part of the Apple as well as MS-Windows based systems.
17. What is JPEG?
The Joint Photographic Experts Group, formed as a joint ISO and CCITT working committee, is focused exclusively on still-image compression.
18. What is called Asymmetrical compression based on Compression?
These are applications that need to be compressed once but are read many times.
19. What are the considerations in Multimedia storage?
Massive storage volumes, large object sizes, multiple related objects, temporal requirements for retrieval.
20. What are the strengths of object oriented s/w?
Encapsulation, Association, Classification.
21. Write the characteristics of Multimedia Database
 - Reduce time and space to store and retrieve
 - Eliminates lost/ missing files by automatic indexing
 - Provide simultaneous document access by multiple users
 - Provide rapid and correct response to request

22. How are image annotation useful for information systems?

Image annotation or description about the image, is used by image retrieval systems to organize and locate images of interest from the database.

23. What are the basic objects in multimedia?

- Text
- Image
- Audio
- Full motion video

24. Name a few analog and digital video broad cast standards.

Analog: NTSC, PAL, SECAM

Digital: HDTV, UDTV

25. Define Cadence.

Cadence is a term used to define the regular rise and fall in the intensity of sound.

26. Say some loss less compression standards?

Pack bits encoding, CCITT Group3 1D, CCITT Group3 2D, CCITT Group4, Lempel-Ziv and Welch algorithm LZW.

27. Say some lossy compression standards?

JPEG (Joint photographic Experts Group), MPEG (Moving Picture Experts Group), Intel DVI, CCITT H.261 video coding algorithm, Fractals.

28. What are the advantages of CCITT Group 3 1D?

- It is simple to implement in both h/w and s/w.
- It is a worldwide standard for facsimile, which is accepted for document imaging application. This allows document-imaging applications to incorporate fax documents easily.

29. What is the disadvantage of CCITT Group 3 2D Scheme?

It is complex and relatively difficult to implement in software.

30. What is Luminance?

Luminance refers to brightness. This is a measure of the brightness of the light emitted or reflected by an object.

31. What are the levels of definition in JPEG standards?

Baseline system, Extended system, special loss less function.

32. Define Quantization.

It is a process of reducing the precision of an integer, thereby reducing the number of bits required to store the integer.

33. What are the controls in VCR paradigm?

Play, fast, forward, rewind, search forward, and rewind search.

34. What are types of moving picture?

- Intra picture, Unidirectional predicted pictures, Bi-directional predicted pictures.
35. What are the factors that affect video performance?
Microprocessor speed, Play back window size, Frame rate.
 36. What is fractal?
A fractal is a multidimensional object with an irregular shape or body that has approximately the same shape or body irrespective of size. i.e., irrespective of whether it gets smaller or bigger in size.
 37. What are multimedia file formats?
Rich-Text Format(RTF), Tagged image file format(TIFF), Resource image file format(RIFF), Musical instrument digital interface(MIDI), Joint Photographic Experts Group, Audio Video Interchanged Indeo file Format(AVI), TWAIN.
 38. What is digital pen?
A digital pen is a powerful input device that allows the user to write, draw, point and gesture.
 39. What are the components of PEN?
Electronic pen and digitizer, Pen driver, Recognition context manager, Recognizer, Dictionary, Display driver.
 40. What are the display performance issues?
N/w b/w, Decompression or decoding, Display technology.
 41. What is roping?
Roping causes straight lines to appear twisted or helical. This is caused by poor convergence as successive pixels in the line show different edge colors.
 42. Write the four basic technologies used for flat panel displays.
Passive-matrix monochrome, Active-matrix monochrome, Passive-matrix color, Active-matrix color.
 43. What are the components of Laser printer?
Paper feed mechanism, Laser assembly, Corona assembly, Fuser, Toner cartridge.
 44. What are the main characteristics of voice recognition system?
Separation b/w words, Speaker dependency and speaker-independent recognition, Use of phonemes, Vocabulary size.
 45. Write down the TIFF file format specification.
TIFF- Tagged Image File Format
Provide tags to store information's like resolution, color, compression scheme used, date and time, author of creation.
It supports raster images described using CMY, YCbCr, JPEG.,
 46. Define Run-length Encoding
Encoding technique that replaces consecutive string of characters by 2 bytes, where first byte indicates the frequency of occurrence of the character and the second byte indicates the character itself.

47. Name the types of Voice Recognition Systems.
 - Speaker Dependent system
 - Speaker independent system
48. What is quantization? Give a brief note.
Quantization is a process of converting a range of input values into a smaller set of output values that closely approximates the original data.
49. What is palatte animation?
Palatte animation, is a technique used in computer graphics in which colors are changed in order to give the impression of animation.
50. What are the six levels of RAID technology?
Level 0, Level 1, Level 2, Level 3, Level 4, Level 5

PART B

1. Give the multimedia applications and explain.

1.Document Imaging The fundamental concepts of storage, compression and decompression, and display technologies used for multimedia systems were developed for document image management. Organizations such as insurance agencies law offices,country and state governments, and the federal government manage large volumes of documents. Document image technology is adopted by Department of Defence for applications ranging from military personnel records to maintenance manuals and high-speed printing systems. Almost all document image system use workflows that are customized for the purpose for which they are being used. The workflow defines the sequence for scanning images, performing data entry based on the contents of the Images, indexing them and storing them on optical media. Document Image Hardware requirements: Realtime image decompression and display place an important role on image processing hardware. Image decompression and display hardware supports 4 to 8 planes. 4 planes provide 16 colors and 8 planes provide 256 colors. The image planes are also called bit planes, because, they are addressed by a bit in a bytes. Images must be processed at the rate of tens to hundreds of pixels per nano-second. For high-resolution images, processing of the order of 10 pixels/ ns is enough for monochrome still images. Gray scale images consist of pixels that have shades of gray ranging from 16 to 256. Color images feature color hues instead of shades of gray. Most high-resolution monitors support 16 to 256 colors display capability. The number of colors that can be depicted depends on the number of bits used to define the palette.

2.Image processing and Image Recognition Image processing involves image recognition, Image enhancement,image synthesis, and image reconstruction. An image processing system may actually alter the contents of the image itself. Image processing systems employ the compression and decompression techniques, a wide range of algorithm for object recognition, comparing images of objects with pre-defined objects, extrapolating finer details to view edges more clearly, gray-scale balancing and gray-scale and color adjustments. Let us briefly review the various aspects of image processing and recognition. *Image enhancement: Most image display systems feature some level of image adjustment. Increasing the sensitivity and contrast makes the picture darker by making bonderline pixels black or increasing the gray-scale level of pixels. Capabilities built in the compression boards might include the following * Image calibration: The overall image density is calibrated, and the image pixels are adjusted to a predefined level. * Real time alignment: The image is aligned in real-time for skewing caused by improper feeding of paper. * Gray-Scale normalization: The overall gray level of an image or picture is evaluated to determine if it is skewed in one direction and if it needs cOlTection. * RGB hue intensity adjustment: Too much color makes picture garish and fuzzy. Automatic hue intensity adjustment brings the hue intensity within pre-defined ranges. * Color Separation: A picture with very little color contrast can be dull and may not bring out the details. The hardware used can detect and adjust the range of color separation. * Frame averaging: The intensity level of the frame is averaged to overcome the effects of very dark or very light areas by adjusting the middle tones. Image Animation Computers-created or scanned images can be displayed sequentially at controlled display speeds to provide image animation that simulates real processes. The basic concept of

displaying successive images at short intervals to give the perception of motion is being used successfully in designing moving parts such as automobile engines. Image annotation Image annotation can be performed in one of two ways: as a text file stored along with the image or as a small image stored with the original image. The annotation is overlayed over the original image for display purposes. It requires tracking multiple image components associated with a single page, decompressing all of them, and ensuring correct spatial alignment they are overlayed. Optical Character Recognition Data entry is the most expensive component of data processing, because it requires extensive clerical staff work to enter data. Automating data entry, both typed and handwritten, is a significant application that can provide high returns. Optical Character Recognition (OCR) technology is used for data entry by scanning typed or printed words in a form. Initially, people used dedicated OCR scanners. Now, OCR Technology is available in software. OCR technology, used as a means of data entry, may be used for capturing entire paragraphs of text. The capturing text is almost always entered as a field in a database or in an editable document Handwriting recognition Research for Handwriting recognition was performed for CADI CAM systems for command recognition. Pen-based systems are designed to allow the user to write commands on an electronic tablet. Handwriting recognition engines use complex algorithms designed to capture data in real time as it is being input or from an image displayed in a window, depending on the application. Two factors are important for handwriting recognition. They are the strokes or shapes being entered, and the velocity of input or the vectoring that is taking place. The strokes are parsed and processed by a shape recognizer that tries to determine the geometry and topology of the strokes. It attempts to compare it to existing shapes, such as predefined characters. The stroke is compared with the prototype character set until a match is found or all pre-defined prototypes have been checked without a match. Multimedia system will use handwriting recognition as another means of user input. Non-Textual Image Recognition Image recognition is a major technology component in designing, medical and manufacturing fields. Let us review the basic concepts of image recognition architecture. For example, a general Image recognition system,- the Image Understanding Architecture has the design which calls for three processing layers. (i) 512 x 512 array of custom pixel processors that extract basic features such as lines and object boundaries. (ii) The features of an object extracted by the first layer are tracked by the DSP array, and that information is fed into 512-M byte RAM. (iii) At the highest level, sophisticated AI algorithms perform the difficult task of object and scene recognition.

3. Full motion Digital video Applications Full motion video has applications in the games industry and training, as well as the business world. Full motion video is the most complex and most demanding component of multimedia applications. For business applications, some core requirements are needed. (i) Full-motion video clips should be sharable but should have only one sharable copy. (ii) It should be possible to attach full-motion video clips to other documents such as memos, chapter text, presentation, and so on. The following features should be available: (a) Features, of a VCR metaphor, such as, rewind, fast-forward, play, and search. (b) Ability to move and resize the window displaying the video clip. (c) Ability to view the same clip on a variety of display terminal types with varying resolution capabilities without the need for storing multiple copies in different form (d) Ability to adjust the contrast and brightness of the video clip. (e) Ability to adjust the volume of the associated sound. (f) It should enable the users to place their own indexing marks to locate segments in video clip.

4. Electronic Messaging The first generation mail system provided a basic text link between users and provided a valuable communications medium for users within a department or enterprise. These systems were the first alternative to paper based inter-office memos. The second generation of electronic mail system expands this capability tremendously by providing cross-platform and cross-network electronic mail with a capability to attach other files ranging from editable text files to bit mapped graphics and program executables. A multimedia enabled electronic messaging system requires a sophisticated infrastructure consisting of the following to support it: • Message storage and forward facility. • Message transfer agents to route messages to their final destinations across various nodes in a multilevel network. • Message repositories to store documents in a filing cabinets. • Repositories for dense multimedia components such as images, videos, frames, audio messages, and full-motion video clips. • Ability for multiple electronic hypermedia messages to share the same multimedia components residing in various repositories on the enterprise network. • Local and global Directories to locate users and servers across an enterprise network. • Automatic database synchronization of dynamic electronics messaging databases. • Automatic protocol conversions and data formats conversions. • Administrative tools to manage enterprise-wide networks. 5. A Universal Multimedia Application It is an application that works on universal data type. This means that the application manipulates datatypes that can be combined in a document, displayed on a screen, or printed, with no special manipulations that the user needs to perform. A document of this type may be a

phonebook, a color brochure with pictures and drawings, a memo, a phone message, a video phone message, or live teleconferencing. The application is truly distributed in nature. An important consideration for such a universal application is the methodology for dissemination of the information on a network. Figure describes the user screen for a universal multimedia application. In this screen, mix of windows for displaying still video and document images, a video conference window with a live session in progress, a remote live desk top, and a couple of other windows for applications such as electronic mail and desk top publishing. To maintain all of these windows requires a substantial amount of CPU power. Digital Signal Processing assistance is needed to manage the multiple simultaneous decompressions for JPEG, MPEG and windows applications. Full-Motion Video Messages In addition to textual messages, electronic mail capability allows embedding of voice messages and video messages. Video messages may consist of video snapshots or live video with full-motion picture and sound. Two technological concepts at play in the implementation of full motion video messages: i) The storage and transmitted of a very large volume of data at a high rate, (ii) Decompression of that data to present a continuous play back . Viewer interactive Live video: • The key difference between full motion video and viewer interactive video is that full motion video is a play back of stored video clips while viewer interactive video is a live. It may be possible to manage decompression and display of stored video clips more easily. • Interactive Live video are the interesting applications used for direct interaction, medical application, manufacturing application and various process control application. • Whereas full motion video are useful for for messagins, information dissemination. Audio and Video Indexing. Indexing is an important and complex subject for multimedia design. Marking a position is called Indexing. Audio and video indexing are used in full-motion video in a manner similar to any video sequence, i.e., just as it would in a home movie, taped performance and so on. The needs of the application must be a strong consideration for the type of indexing provideq with the system. Key points for indexing of stored video clips: * Indexing is useful only if the video is stored, indexing information is lost. * When sound and video are decompressed and managed separately, synchronization is very important. * Depending on the application, indexing information must be maintained separately for sound and video components of a video clip.

2. What are the multimedia elements? Explain.

Data elements for Multimedia Systems

Facsimile Facsimile transmissions were the first practical means of transmitting document images over telephone lines. The basic technology, now widely used, has evolved to allow higher scanning density for better-quality fax

Document images Document images are used for storing business documents that must be retained for long periods of time or may need to be accessed by a large number of people. Providing multimedia access to such documents removes the need for making several copies of the original for storage or distribution

Photographic images Photographic images are used for a wide range of applications . such as employee records for instant identification at a security desk, real estates systems with photographs of houses in the database containing the description of houses, medical case histories, and so on

Geographic information systems map (GIS) Map created in a GIS system are being used widely for natural resources and wild life management as well as urban planning. These systems store the geographical information of the map along with a database containing information relating highlighted map elements with statistical or item information such as wild life statistics or details of the floors and rooms and workers in an office building Voice commands and voice synthesis Voice commands and voice synthesis are used for hands-free operations of a computer program. Voice synthesis is used for presenting the results of an action to the user in a synthesized voice.

Applications such as a patient monitoring system in a surgical theatre will be prime beneficiaries of these capabilities. Voice commands allow the user to direct computer operation by spoken commands Audio message Annotated voice mail already uses audio or voice message as attachments to memos and documents such as maintenance manuals. Video messages Video messages are being used in a manner similar to annotated voice mail. Holographic images All of the technologies so far essentially present a flat view of information. Holographic images extend the concept of virtual reality by allowing the user to get "inside" a part, such as, an engine and view its operation from the inside.

Fractals Fractals started as a technology in the early 1980s but have received serious attention only recently. This technology is based on synthesizing and storing algorithms that describes the information.

3. Explain about Multimedia databases.

Multimedia Databases Images, sounds and movies can be stored, retrieved and played by many databases. In future, multimedia databases will become a main source of interaction between users and multimedia elements. Multimedia storage and retrieval Multimedia storage is characterized by a number of considerations. They are:

- (i) massive storage volumes
- (ii) large object sizes
- (iii) multiple related objects

(iv) temporal requirements for retrieval
Massive Data Volumes A single multimedia document may be a combination of different media. Hence indexing of documents, films and tapes is more complex. Locating massive data volumes requires searching through massive storage files. Locating and indexing systems can be understood only by a few key staff personnel. Hence it requires a major organizational effort to ensure that they are returned in proper sequence to their original storage location.
storage technologies There are two major mass storage technologies used currently for storage of multimedia documents.

(i) Optical disk storage systems.

(ii) High-speed magnetic storage. Advantages of Optical disk storage systems:

- (i) Managing a few optical disk platters in a juke box is much simpler than managing a large magnetic disk farm.
- (ii) Optical disk storage is an excellent storage system for off line archival of old and infrequently referenced documents for significant periods of time.
Multimedia object storage Multimedia object storage in an optical medium serves its original purpose, only if it can be located fast and automatically. A key issue here is random keyed Access to various components of hypermedia database record. Optical media provides very dense storage. Speed of retrieval is another consideration. Retrieval speed is a direct result of the storage latency, size of the data relative to display resolution, transmission media and speed, and decompression efficiency. Indexing is important for fast retrieval of information. Indexing can be at multiple levels. Multimedia document retrieval The simplest form of identifying a multimedia document is by storage platter identification and its relative position on the platter (file number). These objects can then be grouped using a database in folders (replicating the concept of paper storage in file folders) or within complex objects representing hypermedia documents. The capability to access objects using identifiers stored in a database requires capability in the database to perform the required multimedia object directory functions. Another important application for sound and full motion video is the ability to clip parts of it and combine them with another set. Indexing of sound and full-motion video is the subject of intense debate and a number of approaches have been used.
Database Management Systems for Multimedia Systems Since most multimedia applications are based primarily on communications technologies, such as electronic mail, the database system must be fully distributed. A number of database storage choices are available. The choices available are:
 - * Extending the existing relational database management systems, (RDBMSs) to support the various objects for multimedia as binary objects.
 - * Extending RDBMSs beyond basis binary objects to the concepts of inheritance and classes. RDBMSs supporting these features provide extensions for object-programming front ends and/or C++ support.
 - * Converting to a full fledged object oriented database that supports the standard SQL language.
 - * Converting the database and the application to an object-oriented database and using an object-oriented language, or an object-

enabled SQL for development. Multimedia applications combine numerical and textual data, graphics from GUI front-ends, CAD/CAM systems and GIS applications, still video, audio and full-motion video with recorded audio and annotated voice components. Relational databases, the dominant database paradigm, have lacked the ability to support multimedia databases. Key limitations of relational database systems for implementing multimedia applications stem from two areas: the relational data model and the relational computational model. RDBMSs have been designed to manage only tabular alphanumeric forms of data (along with some additional data types stored in binary form such as dates). RDBMS Extensions For Multimedia Binary Large Object (BLOB) is a data type which has been adapted by most of the leading relational databases. BLOBs are used for objects such as images or other binary data types. The relational database is extended to access these BLOBs to present the user 'with a complete' data set. Extended relational databases provide a gradual migration path to a more object-oriented environment. Relational database tables include location information for the BLOBs which may be stored outside the database on separate image or video servers. Relational databases have the strength of rigorous set management for maintaining the integrity of the database. Object-Oriented Databases for Multimedia In object databases, data remains in RMS or flat files. Object databases can provide the fastest route to multimedia support. Object programming embodies the principles of reusable code and modularity. This will ease future maintenance of these databases. Object database capabilities such as message passing, extensibility, and the support of hierarchical structures, are important for multimedia systems. We can develop the application fastest class definitions. ODBMSs are extensible. They allow incremental changes to the database applications. Extensibility: Extensibility means that the set of operations, structures and constraints that are available to operations are not fixed, and developers can define new operations, which can then be added as needed to their application. Object-oriented software technology has three important concepts. They are: Encapsulation: It is the ability to deal with software entities as units that interact in pre-defined and controllable manner, and where the control routines are integral with entity. Association: It is the ability to define a software entity in terms of its differences from another entity. Classification: It is the ability to represent with a single software entity a number of data items that all have the same behavior and the same state attributes. Object orientation helps to organize the software in a more, modular and re-usable manner. Encapsulation allows for the development of open systems where one part of the application does not need to know the functioning of other part. It also provides autonomy; Autonomy means we can interface to a variety of external programs can be built in one class of objects and the storage of the data in another class of objects. Database Organization for Multimedia Applications Data organization for multimedia systems has some key issues. They are:

- (1) Data independence
- (2) Common distributed database architecture
- (3) Distributed database servers
- (4) Multimedia object management. Data Independence Flexible access by a number of databases requires that the data be independent from the application so that future applications

can access the data without constraints related to a previous application. Key features of data independent designs are:

- 1.Storage design in independent of specific applications.
- 2.Explicit data definitions are independent of application program.
- 3.Users need not know data formats or physical storage structures.
- 4.Integrity assurance in independent of application programs.
- 5.Recovery in independent of application programs.

Distributed Data servers : Distributed database

Distributed Data servers : Distributed database servers are a dedicated resource on a network accessible to a number of applications. The database server is built for growth and enhancement, and the network provides the opportunity for the growth of applications and distributed access to the data.

Multimedia Object Management The object management system must be capable of indexing, grouping and storing multimedia objects in distributed hierarchical optional storage systems, and accessing these objects on or keyed basis. The design of the object management system should be capable indexing objects in such a manner that there is no need to maintain multiple storage copies.

Transaction management for Multimedia Systems Multimedia transactions are very complex transactions. We define a multimedia transaction as the sequence of events that starts when a user makes a request to display, edit, or print a hyper media document. The transaction is complete when the user releases the hypermedia document and stores back the edited versions or discards the copy in memory (including virtual memory) or local storage . In most simple applications based on text and textual or numeric data a transaction managed generally by the server that provides the storage for the data. Even these transactions becomes more complex when data has to be retrieved from multiple data servers that can be accessed simultaneously by a large number of users. Conflicts arise when two users attempts to read from and write to same data record. A multi-phase commit methodologies are used to address conflicts in relational database

4. Explain about defining objects for multimedia system.

DEFINING OBJECTS FOR MULTIMEDIA SYSTEMS

The basic data types of object using in multimedia include text, image, audio, holograms and fullmotion video.

TEXT It is the simplest of data types and requires the least amount of storage. Text is the base element of a relational database. It is also the basic building of a document. The major attributes of text include paragraph styling, character styling, font families and sizes, and relative location in a document

HYPERTEXT It is an application of indexing text to provide a rapid search of specific text strings in one or more documents. It is an integral component of hypermedia documents. A hypermedia document is the basic complex object of which text is a sub object. Sub-objects include images, sound and full motion video. A hypermedia document always has text and has one or more other types of sub-objects

IMAGES Image object is an object that is represented in graphics or encoded form. Image object is a subobject of the hypermedia document object. In this object, there is no direct relationship between successive representations in time. The image object includes all data types that are not coded text. It do not have a temporal property associated with them. The data types such as document images, facsimile systems, fractals, bitmaps, meta files, and still pictures or still video frames are grouped together. Figure 3.6 describes a hierarchy of the object classes Non-Visible: This type of images are not stored as images. But they are displayed as images. Example:

Pressure gauges, and temperature gauges. Abstract: Abstract images are computer-generated images based on some arithmetic calculations. They are really not images that ever existed as real-world objects. Example of these images is fractals.

AUDIO AND VOICE Stored-Audio and Video objects contain compressed audio information. This can consist of music, speech, telephone conversation and voice commands. An Audio object needs to store information about the sound clip. Information here means length of the sound clip, its compression algorithm, playback characteristics, and any annotations associated with the original clip.

FULL MOTION AND LIVE VIDEO Full motion video refers to pre-stored video clips. Live video refers to live and it must be processed while it is being captured by the camera. . From a storage perspective, we should have the information about the coding algorithm used for compression. It need decoding also. From a processing perspective, video should be presented to user with smooth and there should not be any unexpected breaks. Hence, video object and its associated audio object must be transferred over the network to the decompression unit. It should be then played at the fixed rate specified for it. For successful playback of compressed video, there are number of technologies. They are database storage, network media and protocols, decompression engines and display engines.

5. Explain about Multimedia data interface standards?

MULTIMEDIA DATA INTERFACE STANDARDS

File Formats for Multimedia Systems:

- (iii) Device-independent Bitmap (DIB): This file format contains bit map, color, and color palette information.
- (iv) RIFF device Independent Bitmap (RDIB): Resource Interchange File Format (RIFF) is the standard file format defined for Microsoft Windows and OS/2. It allows a more complex set of bit maps than can be handled by DIB.
- (v) Musical Instrument Digital interface (MIDI): This is the interface standard for file transfer between a computer and a musical instrument such as a digital piano. It is also, used for full-motion video and voice-mail messaging systems. It has the advantage of ready availability of MIDI device controller boards for personal computers. RIFF Musical Instrument Digital Interface A MIDI format within a RIFF envelope provides a more complex interface. Palette File Format (PAL) An interface that allows defining a palette of 1 to 256 colours in a representation as RGB values. Rich Text Format (RTF) This file format allows embedding graphics and other file formats within a document. This format is used by products such as Lotus Notes. This format is also the basis for the use of OLE. Waveform Audio File Format (WAVE) A digital file representation of digital audio. Windows Metafile Format (WMF) This is a vector graphic format used by Microsoft Windows as an interchange format. Multimedia Movie Format (MMM) This is a format used for digital video animation. Apple's Movie Format This format was defined as the standard for file exchange by Quick Time enabled systems. Digital Video Command Set (DVCS) This is the set of digital video commands simulating VCR controls. Digital Video Media Control Interface Microsoft's high level control interface for VCR controls, including play, rewind, record and so on. Vendor - Independent Messaging (VIM) Developed by a consortium of Vendors providing a standardized format for cross-product messages. Apple's Audio Interchange File Format Apple's standard file format for compressed audio

and voice data. SDTS GIS Standard The Spatial Data Transfer Standard (SDTS) is designed to provide a common storage format for geographic and cartographic data.

VIDEO PROCESSING STANDARDS INTELS DVI DVI is an acronym of Digital Video Interface. DVI standard is to provide a processor independent specification for a video interface. That video interface should accommodate most compression algorithms for fast multimedia displays. An example of custom-designed chip which supports DVI is Intel's i750 B. This chip is designed for enhancing low-end, software based PC video. Advantages of the DVI Chip

- (vi) It can operate software video processing real time.
- (vii) It can share the processing with the host CPU.
- (viii) It can handle additional vector-quantization-type algorithms in conjunction with host processing. DVI silicon chip relies on a programmable video processor. It gives potential to DVI chips to run a range of compression algorithms. **APPLE QUICK TIME** Quick Time standard is developed by Apple Computer. It is designed to Support multimedia applications. It is integrated with the operating system. Quick time refers to both the extensions to the Mac Operating system and to the compression/decompression functionality Of the environment. Quick Time is designed to be the graphics standard for timebased graphic data types. Quick Time's definition has been extended to include (i) System Software, (ii) File Formats, (Hi) Compression! decompression algorithms,
- (ix) **Human Interface Standards.** Figure Shows the components in the Quick Time Architecture. Movie Image ToolBox compression Manager Component Manager Quick Time adjust automatically to the hardware being used by the user. MPEG is another competing standard which is comparatively higher-end, hardware-assisted standard. It can produce better resolutions at faster rates. **MICROSOFT AVI** A VI is an acronym for Audio Video Interleave Standard. It is similar to Apple's Quick Time. It offers low-cost, low-resolution video processing for the average desktop user. It is a layered product. A VI is scalable. It allows users to set parameter such as window size, frame rate, quality and compression algorithm through a number of dialog boxes. AVI-compatible hardware allows enhancing performance through hardware-accelerated compression algorithms such as DVI and MPEG. A VI supports several compression algorithms

5. Give in detail Huffman encoding.

Huffman Encoding

A modified version of run-length encoding is Huffman encoding. It is used for many software based document imaging systems. It is used for encoding the pixel run length in CCITT Group 3 1-dGroup 4. It is variable-length encoding. It generates the shortest code for frequently occurring run lengths and longer code for less frequently occurring run lengths. Mathematical Algorithm for Huffman encoding: Huffman encoding scheme is based on a coding tree. It is constructed based on the probability of occurrence of white pixels or black pixels in the run length or bit stream. Table below shows the CCITT Group 3 tables showing codes or white run lengths and black run lengths

6. Explain fractal compression.

Algorithm, Diagram, explanation.

7. Explain data and file format standards.
TIFF, RIFF, MIDI, JPEG, MPEG, TWAIN

8. Explain about Digital pen.
Electronic pen,
digitizer, RCM,
recognizer, dictionary,
display driver.

5. Explain about cache management for storage systems.

- Low-level disk caching
- Cache organization for hierarchical storage systems
- Cache organization for distributed client-server systems

6. Explain the MPEG Encoding technique for video.

Block Diagram

Explanation

Encoding file format

7. How does RAID technologies help in efficient storage and retrieval of Multimedia data? Explain.

RAID use

Levels of RAID

Diagram and Explanation of each level

8. Discuss in detail JPEG Image compression Technique.

Block Diagram

Explanation

Encoding file format

UNIT V HYPERMEDIA PART A

1. What are the design issues for multimedia authoring?

Display resolution, Data formats for captured data, Compression algorithms, Network interfaces, and Storage formats.

2. What are the types of Multimedia authoring Systems?

Dedicated Authoring system, Timeline-Based Authoring, Structured Multimedia Authoring, Programmable Authoring Systems, Multisource Multi-User Authoring systems, Telephone Authoring Systems.

3. Classify the User interface development tools?

Media editors, An authoring application, Hypermedia object creation, Multimedia object locator and browser.

4. What is the purpose of zooming?

Zooming allows the user to see more detail for a specific area of the image.

5. What is panning?

Panning implies that the image window is unable to display the full image at the selected resolution for display. In that case the image can be panned left to right or right to left as well as top to bottom or bottom to top. Panning is useful for finding detail that is not visible in the full image.

6. What are the steps needed for Hypermedia report generation?

Planning, Creating each component, Integrating components.

7. Define mail message.

Mail message is a message of a well-defined type that must include a message header and may include note parts, attachments, and other application-defined components. Note parts may include text, bitmaps, pictures, sound, and video components.

8. What are the components of a distributed Multimedia system?

Application s/w, Container object store, Image and still video store, Audio and video component store, Object directory service agent, Component service agent, User interface service agent, Networks.

9. What are the characteristics of Document store?

Primary document storage, Linked object storage, Linked object management.

10. What are key issues in data organization for multimedia systems?

Data independence, Common Distributed Database Architecture, Multiple Data servers.

11. What are the key elements in object server architecture of multimedia applications?

Multimedia application, Common object management, Object request broker, Object name server, Object directory manager, Object server, Object manager, Network manager, Object data store.

12. What are the functions performed by object request broker?

Object recompilation, Playback control, Format conversion.

13. What are the issues in database replication techniques?

Sharing of all data objects by all users on the networks, Providing acceptable performance to all users, allowing all users to update the database depending on the tasks being performed by them.

14. What are the types of database replication?

Round-robin replication, manual replication, scheduled replication, immediate replication, replication-on-demand, predictive replication, replicating references, no replication.

15. What are the primary n/w topologies used for multimedia?

Traditional LANS, Extended LANS, High-speed LANS, WANS.

16. Give the primary goal of MAPI.
Separate client applications from the underlying messaging services, Make basic mail-enabling a standard feature for all applications, Support messaging-reliant workgroup applications.
17. What is the purpose of MIME?
Multipurpose Internet Mail Extension specification defines mechanisms for generalizing the message content to include multiple body parts and multiple data types.
18. What are the characteristics of image and still video stores ?
Compressed information, Multi-image documents, Related annotations, Large volumes, Migration b/w high-volume media such as an optical disk library and high-speed media such as magnetic cache storage, shared access.
19. What are the services provided by a directory service agent?
Directory service, Object assignment, Object status management, Directory service domains, Directory service server elements, n/w access.
20. What are the services provided by User Interface Agent?
Window management, Object creation and capture, Object display and play back, Services on workstations, Using display s/w
21. List the pros and cons of Linking and embedding multimedia objects

Linking Is the practice of providing a link of source file in the destination document. Adv: Reduces File size Any change in source document will reflect in destination document as well.	Embedding Placing the data or graphic in the destination document directly. Dis Adv: Increases file size Any change in source document will not be reflected in destination document
--	--
22. Define Hypermedia.
Media (Text, Audio, Video, Image) which provide link to other media is called hypermedia.
23. What are the standard types of multimedia object servers?
Audio server
Video server
Image database

PART B

1. Explain hypermedia messaging.
HYPER MEDIA MESSAGING
Messaging is one of the major multimedia applications. Messaging started out as a simple textbased electronic mail application. Multimedia components have made messaging much more complex. We see how these components are added to messages.

5.2.1 Mobile Messaging Mobile messaging represents a major new dimension in the users interaction with the messaging system. With the emergence of remote access from users using personal digital assistants and

notebook computers, made possible by wireless communications developments supporting wide ranging access using wireless modems and cellular telephone links, mobile messaging has significantly influence messaging paradigms. Handheld and desktop devices play an important growth area for messaging, require complementary back-end services to effectively manage communications for large organizations. Hypermedia messaging is not restricted to the desktops; it is increasingly being used on the road through mobile communications in metaphors very different from the traditional desktop metaphors.

5.2.2 Hypermedia Message Components A hypermedia message may be a simple message in the form of text with an embedded graphics, sound track, or video clip, or it may be the result of analysis of material based books, CD ROMs, and other on-line applications. An authoring sequence for a message based on such analysis may consist of the following components.

1. The user may have watched some video presentation on the material and may want to attach a part of that clip in the message. While watching it, the user marks possible quotes and saves an annotated copy.
2. Some pages of the book are scanned as images. The images provide an illustration or a clearer analysis of the topic
3. The user writes the text of the message using a word processor. The text summarizes the highlights of the analysis and presents conclusions. These three components must be combined in a message using an authoring tool provided by the messaging system. The messaging system must prompt the user to enter the name of the addressee for the message. The message system looks up the name in an online directory and convert it to an electronic addresses well as routing information before sending the message. The user is now ready to compose the message. The first step is to copy the word processed text report prepared in step 3 above in the body area of the message or use the text editor provided by the messaging system. The user then marks the spots where the images are referenced and uses the link and embed facilities of the authoring tool to link in references to the images. The user also marks one or more spots for video clips and again uses the link and embed facilities to add the video clips to the message. When the message is fully composed, the user signs it (electronic signature) and mails to the message to the addressee (recipient). The addressing system must ensure that the images and video clips referenced in the message are also transferred to a server 'local' to the recipient. Text Messages In earlier days, messaging systems used a limited subset of plain ASCII text. Later, messaging systems were designed to allow users to communicate using short messages. Then, new messaging standards have added on new capabilities to simple messages. They provide various classes of service and delivery reports. Typical

Electronic mail message Other capabilities of messaging systems include a name and address directory of all users accessible to the messaging system. **Rich-Text Messages** Microsoft defined a standard for exporting and importing text data that included character set, font table, section and paragraph formatting, document formatting, and color information-called Rich Text Format (RTF), this standard is used for storage as well as Import and export of text files across a variety of word-processing and messaging systems. When sections of this document are cut and pasted into another application, the font and formatting information is retained. This allows the target application to display the text in the nearest equivalent fonts and formats. Rich-text messages based on the RTF formats provide the capability to create messages in one word processor and edit in another at the recipient end. Most messaging systems provide richtext capability for the field of a message.

Voice Messages Voice mail systems answer telephones using recorded messages and direct the caller through a sequence of touch tone key operations until the caller is connected to the desired party or is able to leave a recorded message. **Audio' (Music)** The Musical Instrument Digital interface (MIDI) was developed initially by the music industry to allow computer control of and music recordings from musical instruments such as digital pianos and electric keyboards. MIDI interfaces are now being used for a variety of peripherals, including digital pianos, digital organs, video games with high-fidelity sound output, and business presentations. **Full-Motion Video Management** Use of full-motion video for information repositories and memos are more informative. More information can be conveyed and explained in a short full-motion video clip than can be conveyed in a long text document. Because a picture is equivalent to thousand words. **Full Motion video Authoring System** An authoring system is an important component of a multimedia messaging system. A good authoring system must provide a number of tools for the creation and editing of multimedia objects. The subset of tools that are necessary are listed below:

1. A video capture program - to allow fast and simple capture of digital video from analog sources such as a video camera or a video tape. .

2. Compression and decompression Interfaces for compressing the captured video as it is being captured.

3. A video editor with the ability to decompress, combine, edit, and compress digital video clips.

4. Video indexing and annotating software for marking sections of a videoclip and recording annotations. Identifying and indexing video clips for storage. Full-Motion Video Playback Systems The playback system allows the recipient to detach the embedded video reference object, Interpret its contents and retrieve the actual video clip from a specialized video server and launch the Playback application. A number of factors are involved in playing back the video correctly. They are:

1.How the compression format used for the storage of the video clip relates to the available hardware and software facilities for decompression.

2.Resolution of the screen and the system facilities available for managing display windows. The display resolution may be higher or lower than the resolution of the source of the video clip.

3.The CPU processing power and the expected level of degradation as well as managing the degraded output on the fly.

4.Ability to determine hardware and software facilities of the recipient's system, and adjusting playback, parameters to provide the best resolution and performance on playback. The three main technologies for playing full motion video are microsoft's video for windows:

Apple's Quicktime, and Intel's Indeo. Video for Windows (VFW): It is the most common environment for multimedia messaging. VFW provides capture, edit, and playback tools for full-motion video. The tools provided by VFW are: The VidCap tool, designed for fast digital video capture. The VidEdit tool designed for decompression, edition, and compressing full-motion digital video. The VFW playback tool. The VFW architecture uses OLE. With the development of DDE and OLE, Microsoft introduced in windows the capability to link or multimedia objects in a standardized manner. Hence variety ;;windows based applications can interact with them.

We can add full-motion video to any windows-based application with the help of VFW. The VFW playback tool is designed to use a number of codecs (software encoder/decoders) for decompressing and playing video files. The default is for AVI files. Apple's QuickTime An Apple QuickTime product is also an integrated system for playing back video files. The QuickTime product supports four compression methodologies. Intel's Indeo Indeo is a digital video recording format. It is a software technology that reduces the size of uncompressed video files through successive compression methodologies, including YUV sub sampling, vector quantization, Huffman's run-length encoding, and variable content encoding. Indeo technology is designed to be scalable for playing back video; It determines the hardware available and optimizes playback for the hardware by controlling the frame rate.

The compressed file must be decompressed for playback. The Indeo technology decompresses the video file dynamically in real time for playback. Number of operating systems provide Indeo technology as standard feature and with other software products (eg. VFW).

5.2.3Hypermedia Linking and Embedding Linking and embedding are two methods for associating multimedia objects with documents. This topic deals with ⌘ Linking as in hypertext applications. Hypertext system associate keywords in a document with other document. ⌘ Linking multimedia objects stored separately from the document and the link provides a pointer to its storage. ⌘ Linking and embedding in a context specific to Microsoft Object linking and Embedding. Linking in hypertext documents Hypertext documents are indexed to locate keywords within the text component of the hypermedia document. An extension of this capability is to locate information within the linked component.

There are two types of links passive links and active links Active Links- performs functions on their own based on readers customization. Active links are more intelligent and may use artificial intelligence technologies to monitor the nature of tasks performed by the user.

Passive Links- allow associating one document with another in a number of ways including the author to name the subject of a link and access it based on the content. Linking and Embedding: Linking and embedding are two ways of associating multimedia objects with a hypermedia document or a database record. Let us discuss it in detail. Linking Objects When an object is linked, the source data object, called the link source, continues to stay whenever it was at the time the link was created. This may be at the object server where it was created, or where it has been copied. Only reference is required in the hypermedia document. The reference is also known as link. This link reference includes information about the multimedia object storage, its presentation

parameters, and the server application that is needed to display/play or edit it. When this document is copied, the link reference is transferred. But the actual multimedia document remains in its original location.

A linked object is not a part of the hypermedia document and it does not take up storage space within the hypermedia document. If the creator, or authorised user edits the original stored multimedia object, subsequent calls to the linked object bring the copy. Embedded Objects If a copy of the object is physically stored in the hypermedia document, then the multimedia object is said to be embedded. Any changes to the original copy of that object are not reflected in the embedded copy.

When the hypermedia document is copied, the multimedia object is transferred with it to the new locations. Graphics and images can be inserted in a rich-text document or embedded using such techniques as OLE. Voice and audio components can be included in a text message; or they can be part of a full voice-recorded message that has embedded text and other components.

2. Explain the components of distributed multimedia systems.

DISTRIBUTED MULTIMEDIA SYSTEMS

5.7.1. Components of a Distributed multimedia systems:

If the multimedia systems are supported by multiuser system, then we call those multimedia systems as distributed multimedia systems. A multi user system designed to support multimedia applications for a large number of users consists of a number of system components. A typical multimedia application environment consists of the following components:

1. Application software.
2. Container object store.
3. Image and still video store.
4. Audio and video component store.
5. Object directory service agent.
6. component service agent.
7. User interface and service agent.
8. Networks (LAN and WAN).

3. Give the multiserver n/w topologies.

Traditional LANS (Ethernet or Token Ring) Ethernet: Ethernet: It is a Local Area Network hardware, communication, and cabling standard originally developed by Xerox corporation that link up to 1024 nodes in a bus network. It is a high speed standard using a baseband (single-channel) communication technique. It provides for a raw data transfer rate of 10 Mbps, with actual throughput in the range of 2-3 Mbps. It supports a number of sessions in a mix of live video, audio electronic mail and so on.

Token Ring: It is a Local Area Network architecture that combines token passing with a hybrid star/ring topology. It was developed by IBM. Token Ring Network uses a multistation Access unit at its hub ..

Extended LANS: Ethernet and token rings are extended to get high transfer rates. For extension they use hubs, bridges and routers. In extended LANS each segment operates at the normal LAN bandwidth.

Switching Hubs: • These have fast switching and these network can support the requirements for full motion video standard such as MPEG2. • The important advantage of this approach is that workstation do not require additional LAN hardware if they are already connected to a LAN. • The user workstation continues to operate on low-cost, low speed LAN connection.

Bridges and Routers: • Bridges and Routers transfer a packet of data from LAN segment to another LAN segment. • Bridges are devices that interconnects local or remote networks no matter what higher level protocols are involved.

• Bridges form a single logical network, centralized network administration. • Routers are protocol dependent device that connects subnetworks together. It is useful in breaking down a very large network into smaller subnetworks. **Switching and Routing Latency:** • Switching latency is defined as the time it takes a switching hub to interconnect one LAN segment to another LAN segment. • Routing delay is defined as the delay experienced by a packet of data within the router.

ATM (Asynchronous Transfer Mode) It is a network architecture that divides messages into fixed size units (called cells) of small size and that establishes a switched connection between the originating and receiving stations. ATM appears to be a potential technology for multimedia systems for connecting object servers and user workstations. ATM is actually a good candidate for two reasons: as a hub and spoke technology, it adapts very well to the wiring closest paradigm; and it allows workstations to operate at speeds defined by the workstation. Figure below illustrates LAN topology using an ATM Switching System.

FDDI II (Fiber Distributed Data Interface II) It is a standard for creating highspeed computer networks that employ fiber-optic cable. FDDI II operates exactly like token ring, with one difference: FDDI employs two wires through all the hosts in a network.

FDDI II is a single media LAN and its full bandwidth supports all users.

FDDI II appears to be a very useful high-speed technology for connecting servers on an additional separate network and providing the dedicated high bandwidth necessary for rapid transfer and replication of information objects.

WANS (Wide Area Network) This includes LANs, dial up ISDN, T1 (1.544 Mbits/sec) and T3 (45.3 Mbits/sec) lines and regular telephone dial-up lines. The two big issues here are: • ∴ WANs may have a mix of networking and communication protocols. • ∴ WAN has a variety of speeds at which various parts of it where it communicates. Protocol Layering: Layering helps to isolate the network from the application. Layering of protocols started with the release of the ISO model.

3. Explain Distributed Client-server operation.

The agents so far we have discussed combine to form a distributed client-server system for multimedia applications. Multimedia applications require functionality beyond the traditional client server architecture. Most client-server systems were designed to connect a client across a network to a server that provided database functions. In this case, the client-server link was firmly established over the network. There was only one copy of the object on the specified server. With the development of distributed work group computing, the picture has changed for the clients and servers. Actually in this case, there is a provision of custom views in large databases. The advantage of several custom views is the decoupling between the physical data and user. The physical organization of the data can be changed without affecting the conceptual schema by changing the distributed data dictionary and the distributed data repository.

Clients in Distributed Work Group Computing Clients in distributed workgroup computing are the end users with workstations running multimedia applications. The client systems interact with the data servers in any of the following ways 1. Request specific textual data. Request specific multimedia objects embedded or linked in retrieved container objects.

3. Require activation of a rendering server application to display/ playback multimedia objects.

4. Create and store multimedia-objects on servers.

Request directory information. on locations of objects on servers Servers in Distributed Workgroup Computing Servers are storing data objects. Includes the following functions, ∅ storage for a variety of object classes, ∅ transfer objects on demand on clients. ∅ hierarchical storage for moving unused objects to optical disk libraries or optical tape libraries. ∅ system Administration functions for backing up stored data. ∅ direct high-speed LAN and WAN server-to-server Transport for copying multimedia objects.

5. How to manage Distributed objects.

We see the nature of communication between servers and the managing of distributed objects. Interserver communications: Object replication, object distribution, object recompilation and object management and network resources are some of the design requirements that play a role in defining interserver The following lists the types of communications that one server may make to another server:

1. Obtain a token from an object name server for creating a new multimedia object; the object is not accessible by others users until complete and released.
2. Search the object class directory for the current locations of that object and the least expensive route for accessing it.
3. Perform a shared read lock on the object to ensure that it is not archived or purged while it is being retrieved.
4. Replicate a copy of the object; update the object name server directory.
5. Copy an object for non-persistent use.
6. Test and set an exclusive lock on an object for editing purposes'
7. create new versions.
8. Pause the retrieval of an object to support a user action or to pace the retrieval to the speed supported by the network.
9. A Sound server architecture

6. Explain the design issues for multimedia authoring systems.

Types of Authoring system

Advantages and Disadvantages of each.

7. Describe the common forms of navigation for information Access.

Access structure defines the way objects can be accessed and how navigation takes place through the information objects. The common forms of navigations for information access are:

Direct: Direct information access is completely predefined. User must have knowledge about the object that needs to be accessed. That information includes object representations in a compound object. **Indexed:** Index access abstracts the real object from the access to the object. If the object ID of the object is an index entry that resolves to a filename on a specific server and disk partition, then the information access mechanism is an indexed mechanism.

Random Selection: In this form, the user can pick one of several possible items. The items need not be arranged in any logical sequence; and they need not be displayed sequentially. The user need not have much knowledge about the information. They must browse through the information.

Path selection or Guided tour: In a guided tour, the application guides the user through a predefined path across a number of objects and operations. The user may pause to examine the objects at any stage, but the overall access is controlled by the application. Guided tours can also be used for operations such as controlling the timing for discrete media, such as slide show. It can be used for controlling a sound track or a video clip.

Browsing: It is useful when the user does not have much knowledge about the object to access it directly.

8. Write short note on metaphors for multimedia applications

Special Metaphors for Multimedia Applications In this section let us look at a few key multimedia user interface metaphors.

The organizer metaphor One must begin to associate the concept of embedding multimedia object in the appointment diary or notepad to get an obvious view of the multimedia aspects of the organizer. Other use of multimedia object in an organizer is to associate maps or voice mail directions with addresses in address books. The Lotus organizer was the first to use a screen representation of the office diary type organizer.

Telephone Metaphor: The role of the telephone was changed by the advent of voice mail system. Voice mail servers convert the analog voice and store it in digital form. With the standards for voice mail file formats and digital storage of sound for computer. Now, computer system is used to manage the phone system. The two essential components of a phone system are speakers and microphones. They are included in most personal computers. Figure 8.8 shows how a telephone can be created on a screen to make it a good user interface. The telephone keypad on the screen allows using the interface just as a telephone keypad is used. Push buttons in dialog boxes

and function selections in memos duplicate the function provided by the keypad. Push buttons, radio buttons, list boxes, and data entry fields and menu selections allow a range of functionality than can be achieved by the telephone.

9. Explain about the components of a distributed multimedia systems.

1. Obtain a token from an object name server for creating a new multimedia object; the object is not accessible by others users until complete and released.
2. Search the object class directory for the current locations of that object and the least expensive route for accessing it.
3. Perform a shared read lock on the object to ensure that it is not archived or purged while it is being retrieved.
4. Replicate a copy of the object; update the object name server directory.
5. Copy an object for non-persistent use.
6. Test and set an exclusive lock on an object for editing purposes'
7. create new versions.
8. Pause the retrieval of an object to support a user action or to pace the retrieval to the speed supported by the network.
9. A Sound server architecture

STUCOR APP