Q.2 a. List out the four uses of computer graphics. **(2)**

Answer:

2 (a). The following are the uses of computer graphics.

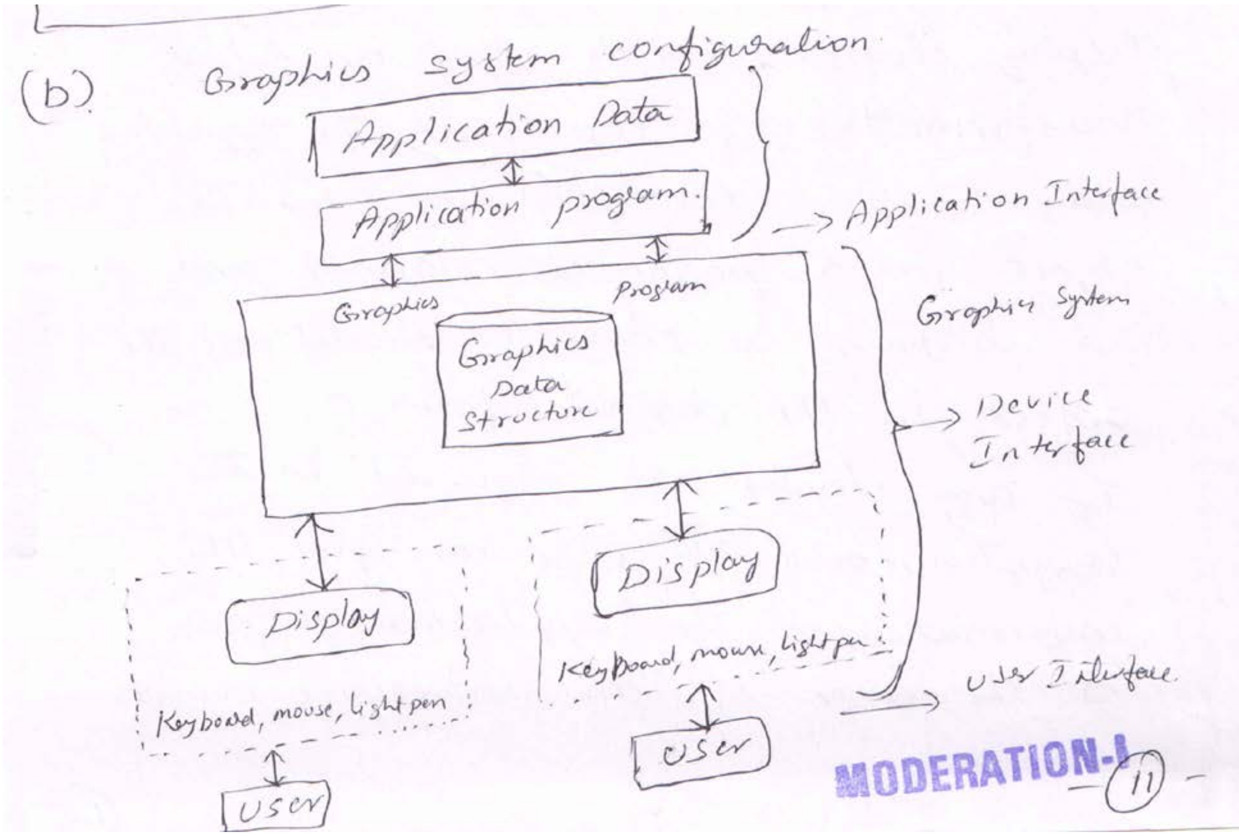
(1) Data Presentation
(2) Scientific visualization.
(3) Cartography and Surveying.
(4) Simulation and animation.
(5) Graphics arts and advertising.
(6) condition monitoring.
(7) office automation and documentation systems.
(8) Electronic printing and publishing.
(9) Design and analysis.
(10) manufacturing
(11) Robotics.
(12) video games.

$\frac{1}{2} \times 4 = 2$

Any 8 should be written ~~1/4 × 8~~ = 2M.

b. With a neat diagram. Explain graphics system configuration. **(8)**

Answer:


(b) Graphics system configuration
Application Data
Application program.
Application Interface
Graphics
Program
Graphics Data Structure
Graphics System
Device Interface
Display
Keyboard, mouse, light pen
User
Display
Keyboard, mouse, light pen
User
User Interface
MODERATION-I
11

DC 66 Computer Graphics.

Computer graphics is based on a simple reference model as shown in the diagram. This configuration consists of 3 components:

1. An application system.
2. A graphics system.
3. A user (operator, client, system server)

The System consists of hardware & Software. Hardware consists of input & output devices for interaction and display.

The Software consists of 3 components.

(a) application program.
(b) application data.
(c) graphics system.

The graphics system is an intermediary between the application program and the display device, which effects an output transformation of objects in the application data, enabling the user to view the object in a particular manner and, i.e. obtain a computer model of the object in the desired form.

If any changes are required in the computer model, the user can give the information to the application program.

So the user of an interactive computer

graphics program specifies what classes of data items or objects are to be generated and represented pictorially, how the application program is to be generated and represented pictorially and. The mode and degree of interaction in order to create and modify the model and its visual representation. The task of creating pictures is done by graphics system.

diagram 3 marks. Explanation. 5 marks.

c. Explain the essential components of GUI. (6)

Answer:

(c). The essential components of GUI are.

(1). Graphics pointer:- It is a mouse cursor. or a symbol that appears on the display Screen and that we move to select objects, icons or any menu commands within the interface.

(2) Pointing devices:- It is a device, such as a mouse or a track ball, which enables The user to select objects, an item from the menu.

(3) Icons: Small images That represent commands, files or windows, By moving the pointer to the icon and pressing a mouse button, one can execute a command or convert the icon to a window.

4) Desktop: It is the area on The display Screen where icons are often grouped It is Known as desk top because the icons are intended to represent real objects on a real workspace.

5). Windows:- It divides the screen into different areas. In each window, one can run different programs or different components of a program or display a different file.

6). Menus: most graphical user interfaces allow the execution of commands by making a choice from a menu.

[Each component 1 mark]

Q.3 a. Write the algorithm for raster display of a line using Bresenham's Algorithm. (8)

Answer:

3.6 BRESENHAM'S LINE ALGORITHM

Bresenham's line algorithm selects optimum raster locations to represent a straight line. In this algorithm, pixels along x and y directions are incremented by one unit depending upon the slope m. If the slope of the line (in first quadrant) is less than half the pixel point in the x direction is shifted, and if the slope is more than half the pixel in the y direction is shifted. The increment in either direction (y or x respectively) is determined by examining the closeness between actual line location and the nearest screen grid location.

The principle of Bresenham's algorithm can be explained as demonstrated in Fig. 3.5. In this figure a straight line [(2, 2) (10, 8)] is to be plotted on the screen.

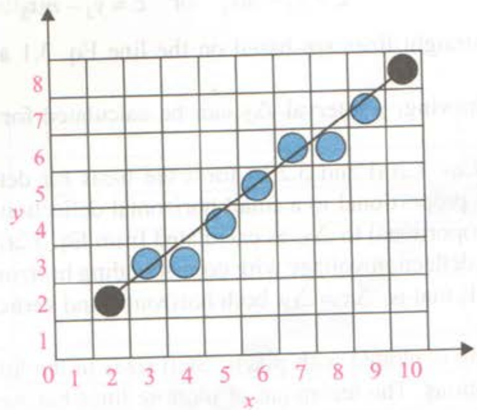


Fig. 3.5 *Line plotting on the screen grid display*

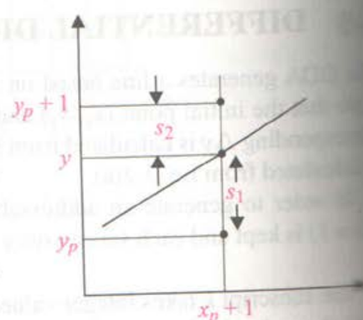


Fig. 3.6 *Selection of pixel posi*

The location of the first and last pixel is indicated by black circles. However, pixels in between are shown as grey pixels. Bresenham's algorithm is based on the pixel position closer to the line path. Although it is simple to plot manually, the algorithm for pixels following the closeness theory is complicated. The algorithm is developed on the basis of testing the sign of the integer parameter whose value is proportional to the difference between the distance of the two pixel positions from the actual line path.

Let us take the example of scan conversion of a line having a positive slope of less than one. The pixel position on the line path between the two end points $[(x_o, y_o), (x_1, y_1)]$ can be found by getting the y position corresponding to the x position every time. Each time a new x position is found by adding the x interval equal to unity, the corresponding y value will be one, which is close to the path line. In Fig. 3.6, it can be seen that after initial position of pixel $[(x_p, y_p), (x_o, y_o)]$, the starting point of the line, the next pixel in x direction will be on x_{p+1} and then corresponding to x_{p+1}, either y_p or y_{p+1} will be selected.

The selection of y_p or y_{p+1} against x_{p+1} is a crucial one, that is, whether to plot point as (x_{p+1}, y_p) or $(x_{p+1},$ $y_{p+1})$. This is decided mathematically by the distance s_1 and s_2 as shown in Fig. 3.6. Coordinate y at pixel column positions x_{p+1} is calculated as

$$y = m(x_p + 1) + c$$

Then,

$$s_1 = (y - y_p) = m(x_p + 1) + c - y_p$$

$$s_2 = (y_p + 1) - y = y_p + 1 - m(x_p + 1) - c$$

Now,

$$s_1 - s_2 = 2m(x_p + 1) - 2y_p + 2c - 1$$

A decision parameter r_p for the rth step in the line algorithm can be obtained by rearranging Eq. 3.8 so that it involves only integer calculations.

Therefore, from Eq. 3.8

$$s_1 - s_2 = 2\frac{\Delta y}{\Delta x}(x_p + 1) - 2y_p + 2c - 1$$

or $$\Delta x(s_1 - s_2) = 2\Delta y\, x_p - 2y_p\, \Delta x + 2\, \Delta y + \Delta x\, (2c - 1)$$

Let $$r_p = \Delta x\, (s_1 - s_2) \quad (3.9)$$

Hence, $$r_p = 2\Delta y\, x_p - 2y_p\, \Delta x + 2\Delta y + \Delta x(2c - 1)$$

Let f = 2 $\Delta y + \Delta x$ (2c – 1). This has all terms as constant.

Thus, $$r_p = 2\Delta y\, x_p - 2y_p\, \Delta x + f$$

Since $\Delta x > 0$ and f is a constant, the sign of the decision parameter is same as $(s_1 - s_2)$. The decision parameter r_p is independent and indicates the position of the pixel at y_p, when it is negative and closer to the line path $(s_1 < s_2)$, that is, $(x_p + 1, y_p)$ is closer to the actual line as compared to $(x_p + 1, y_p + 1)$, and the pixel at $(x_p + 1, y_p)$ will be plotted as the next point. If $(s_1 > s_2)$, the decision parameter will be positive and the next point to be plotted will be $(x_p + 1, y_p + 1)$.

The change in x direction is in unit steps. Therefore, it is necessary to obtain the values of successive parameters using incremental integer calculations. At step $p + 1$, the decision parameter is evaluated from Eq. 3.9.

$$r_{p+1} = 2\Delta y\, x_{p+1} - 2y_{p+1}\, \Delta x + f \quad (3.10)$$

Subtracting Eq. 3.9 from Eq. 3.10

$$r_{p+1} - r_p = 2\Delta y\, (x_{p+1} - x_p) - 2\Delta x(y_{p+1} - y_p)$$

and here $x_{p+1} = x_p + 1$, such that

$$r_{p+1} = r_p + 2\Delta y - 2\Delta x(y_{p+1} - y_p)$$

The term $(y_{p+1} - y_p)$ is either 0 or 1 depending on the sign of r_p. The calculation of decision parameters from starting to end point at each integer x position will be recursive and can be programmed easily in loop. The first parameter position at starting pixel position is calculated from

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

Based on the above, the computer program for line drawing may be developed with the following steps, for slope $|m| < 1$.

1. Declare variables.
2. Input the end points and load the starting point to frame buffer, that is, plot on the screen.
3. Calculate constants Δx, Δy, $2\Delta y$ and $2\Delta y - \Delta x$, and calculate the starting value for the decision parameter as
   $$r = 2\Delta y - \Delta x$$
4. At each r_p along the line, starting at $p = 0$, test if $r_p < 0$, the next point plot is $(x_p + 1, y_p)$, or else the next point is $(x_p + 1, y_p + 1)$ and
   $$r_{p+1} = r_p + 2\Delta y - 2\Delta x$$
5. Repeat the above steps till the end point is reached.

These steps are well explained by digitizing a line between two sample points, as shown in Solved

b. Explain scan line seed fill algorithm for filling a polygon with a single color. (6)

Answer:

3.12.4 Scan Line Seed Fill Algorithm

In seed filling algorithm, the stack size is very large as in each loop the algorithm pushes 4 or 8 pixels onto the stack. There are duplicate pixels too which are recursively propagated making the process slow (more in case of 8 interconnected pixels). But, the scan line seed filling algorithm minimizes the duplicate pixels, by pushing on to the stack only one pixel in any uninterrupted unfilled span of pixels in a single scan line, or a row of pixels in a defined boundary region, that is, a closed polygon. Instead of proceeding along 4 or 8 connected pixels, this algorithm processes in raster pattern, that is, along left to right along each scan line in the region.

The algorithm flows in the following manner:

1. A seed pixel located on the scan line within the area popped from a stack containing the seed pixel is selected.
2. The line or span containing the seed pixel is filled to its right and left including the seed pixel itself until the boundary is found.
3. The extreme left and extreme right unprocessed pixel in the span are saved as *x*-left and *x*-right, respectively.

76 ☆ *Computer Graphics*

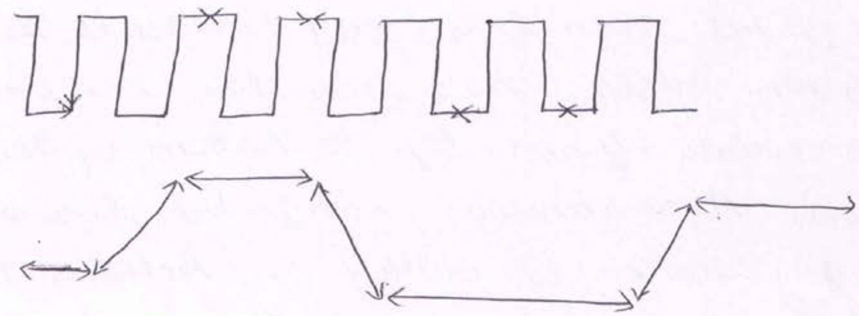
4. The scan lines above and below the current scan line are examined in the range *x*-left and *x*-right for any simple crossover. The extreme right pixel in all the unfilled spans on these scan lines within the same range is marked as a seed pixel and pushed onto the stack.

c. What is aliasing & antialiasing? (2)

Answer:

3 (c). Aliasing and Antialiasing.

The term aliasing comes from the sampling theory in signal processing. If a rapidly varying signal is sampled too infrequently, the samples appear to represent a signal that vary at a lower frequency. The frequency of the original signal appears to be replaced by its lower frequency. This is called aliasing.



Aliasing happens because of the discrete nature of the pixel in raster display. The pixel display in the fixed rectangle causes jagged appearance of black rectangle which should be perfectly displayed.

Antialiasing is a technique used to reduce aliasing effect. This can be done by a display with high resolution because the jags are then smaller in size relative to the object. Another method is blurring to smoothen an image.

| Aliasing – 1m Antialiasing – 1m | – (18) –

Q.4 a. Derive the Transformation matrix in 2D for: **(2x2)**
(i) Shearing about x axis
(ii) Reflection about y axis

Answer:

$$[X][T] = [x\ \ y]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [x\ \ y] = [x^*\ \ y^*]$$

Hence, there is no change in the original coordinate matrix.

Example 2

Let b = c = 0 and d = 1

$$[X][T] = [x\ \ y]\begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} [ax\ y] = [x^*\ y^*]$$

This shows that the element "a" of transformation matrix changes the scale of coordinate x because $x^* = ax$, but there is no change in the direction of y, that is, $y^* = y$. The corresponding change due to transformation can be seen in Fig. 4.1.

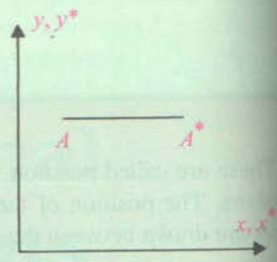


Fig. 4.1 *Scaling transformation along x axis*

Example 3

Let b = c = 0

$$[X][T] = [x\ \ y]\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}[ax\ dy] = [x^*\ y^*]$$

This indicates a scaling of both the x and y coordinates of the original position vector A (Fig. 4.2).

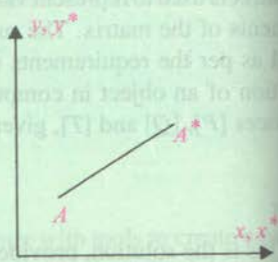


Fig. 4.2 *Scaling transformation along both the axes*

Example 4

(a) Let b = c = 0, d = 1 and a = −1 then

$$[X][T] = [x\ \ y]\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}[-x\ \ y] = [x^*\ y^*]$$

This shows reflection about y axis as shown in Fig. 4.3.

(b) Let b = c = 0, a = 1 and d = −1

$$[X][T] = [x\ \ y]\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\ [x - y] = [x^*\ y^*]$$

This shows reflection about x axis as shown in Fig. 4.4.

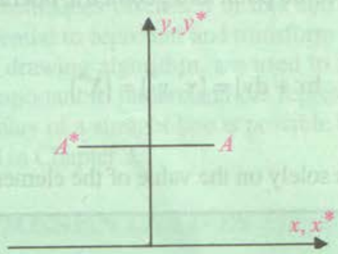


Fig. 4.3 *Reflection about y axis*

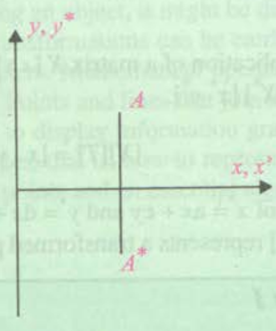


Fig. 4. 4 *Reflection about x axis*

b. Give the transformation matrix for reflection of a point along the line y = mx + c. **(6)**

Answer:

4.13 REFLECTION ABOUT AN ARBITRARY LINE

The reflection of an object as discussed earlier is either passing through the line $x = 0$ or $y = 0$ or $y = x$ or $y = -x$. All these lines pass through origin. However, the operation of reflection through an arbitrary line (not passing through origin) is required to follow the following procedures.

1. Translate the line as well as the object so that the line passes through the origin.
2. Rotate the line and the object about the origin until the line is coincident with one of the coordinate axes.
3. Reflect the object through the coordinate axis.
4. Apply the inverse rotation about the origin to shift the line at translated position.
5. Apply inverse translation or back the object that is, move line to its original position.

In matrix operation

$$[T] = [T_{trans}][R_\theta][R_{ref}][R_\theta]^{-1}[T_{trans}]^{-1}$$

where

$[T_{trans}]$ = Translation matrix

$[R_\theta]$ = Rotation matrix

$[R_{ref}]$ = Reflection matrix

$[R_\theta]^{-1}$ = Inverse rotation matrix

$[T_{trans}]^{-1}$ = Inverse translation matrix

Let an arbitrary line be $y = mx + c$ and the position vectors of vertices be $A\ [x_1\ y_1\ z_1]$, $B\ [x_2\ y_2\ z_2]$ and $C\ [x_3\ y_3\ z_3]$ as shown in Fig. 4.15.

As per the above information, the different matrix required for reflection about given arbitrary line are as follows:

$$[T_{trans}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -c & 0 \end{bmatrix}$$

$$R_\theta = \begin{bmatrix} \cos(-\tan^{-1}m) & \sin(-\tan^{-1}m) & 0 \\ -\sin(-\tan^{-1}m) & \cos(-\tan^{-1}m) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ as } \theta = \tan^{-1}m$$

$$[R_{\text{ref}}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[R_\theta]^{-1} = \begin{bmatrix} \cos(\tan^{-1}m) & \sin(\tan^{-1}m) & 0 \\ -\sin(-\tan^{-1}m) & \cos(-\tan^{-1}m) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T_{\text{trans}}]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & c & 0 \end{bmatrix}$$

Solved Exercises 4.3 and 4.14 discuss problems based on the above.

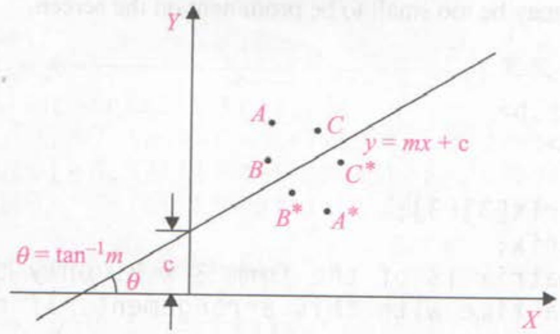


Fig. 4.15 *Reflection about an arbitrary line*

SUMMARY

c. Explain the steps required to rotate an object in 3D about an arbitrary point. (6)

Answer:

4(c) Rotation about an arbitrary point.

1. Translate the object or body at the origin.
2. Rotate by any angle as given.
3. Translate back to its original location.

In matrix form it can be shown as.

$$[T] = [T_{trans}]\,[R_\theta]\,[T_{trans}]^{-1}$$

T_{trans} = Translation matrix.

R_θ = Rotation matrix by angle θ.

T_{trans}^{-1} = Inverse translation matrix.

Rotation about any arbitrary point [m n] as given below.

$$[x^* \; y^* \; 1] = [x \; y \; 1]\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -m & -n & 1\end{bmatrix}\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ m & n & 1\end{bmatrix}$$

$$= [x \; y \; 1]\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ -m(\cos\theta-1)+n\sin\theta & -n(\cos\theta-1)-m\sin\theta & 1\end{bmatrix}$$

Steps 2m	matrix representation 2m

Q.5 a. Explain the Barsky 2D line clipping algorithm. (8)

Answer:

5.7 PARAMETRIC LIANG–BARSKY 2D LINE CLIPPING ALGORITHM

An efficient algorithm was devised by Liang Y. and Barsky B. in the year 1984. It uses inequalities from basic parametric line equations, which distinguish the clipping region and the visible region window. The inequalities discussed are used to establish the geometrical relationship between the and the parametric line.

As discussed in Section 5.5, parametric lines with their ends (x_1, y_1) and (x_2, y_2) can be expressed as follows:

$$x(u) = x_1 + u(x_2 - x_1)$$

$$y(u) = y_1 + u(y_2 - y_1) \qquad \text{where } u(0 \le u \le 1) \text{ is a parameter}$$

The interior of the window can be written as

$$x_{min} \le x \le x_{max}$$

$$y_{min} \le y \le y_{max}$$

Substituting, the parametric line equation with these

$$x_{min} \le x_1 + u(x_2 - x_1)x \le x_{max}$$

$$y_{min} \le y_1 + u(y_2 - y_1)y \le y_{max}$$

This can be rewritten to define interior of window as:

$$-u\Delta x \le x_1 - x_{min} \qquad \text{for left } (1^{st}) \text{ boundary}$$

$$u\Delta x \le x_{max} - x_1 \qquad \text{for right } (2^{nd}) \text{ boundary}$$

$$-u\Delta y \le y_1 - y_{min} \qquad \text{for bottom } (3^{rd}) \text{ boundary}$$

$$u\Delta y \le y_{max} - y_1 \qquad \text{for top } (4^{th}) \text{ boundary}$$

General form of this can be written as:

$$u \times d_i = q_i,$$

where, $i = 1, 2, 3$ and 4 for left, right, bottom and top boundaries, respectively.

$$d_1 = -\Delta x, \quad d_2 = \Delta x, \quad d_3 = -\Delta y, \quad d_4 = \Delta y$$

$$q_1 = x_1 - x_{min}, \quad q_2 = x_{max} - x_1, \quad q_3 = y_1 - y_{min}, \quad q_4 = y_{max} - y_1$$

This is an alternate formulation for checking against boundaries as compared to Cohen–Sutherland algorithm.

If $d_i = 0$, the line is parallel to the *i*th edge of the window.

If $q_i < 0$ it lies outside the *i*th boundary side.

If both the conditions prevail, the line is trivially rejected as it lies completely outside the clipping region. If $q_i = 0$, then P_1 is on the *i*th edge of the window boundary. $q_i \geq 0$ implies that P_1 is inside the visible portion of the window.

The intersection point can be calculated by $u = q_i/d_i$. If the line is not parallel to any of the window boundaries, the intersection points can be labeled as I_i and any intersection point outside the parametric range ($0 \leq u \leq 1$) of the line, for example, I_4 in the Fig. 5.11, can be ignored. The remaining intersection points outside the window can be eliminated by finding the maximum (u_{min}) and minimum parameter value (u_{max}) of the lower and upper set of parametric values, respectively. With these calculated maximum and minimum parameter values (which lie within the window), the visible portion of line can be drawn by using parametric equation of line.

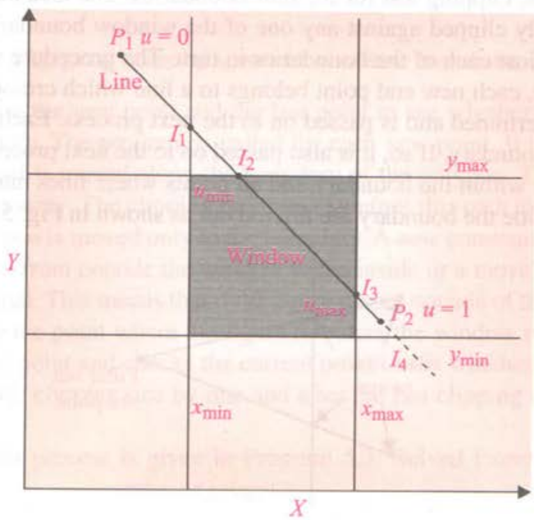


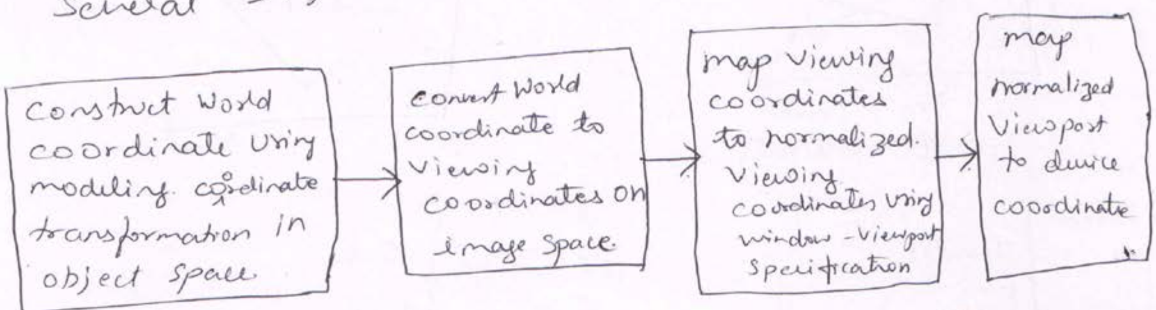
Fig. 5.11 *Various window and line parameters*

Liang–Barsky algorithm makes use of the parametric representation of the line to speed up the intersection computations. It represents the four inequalities required for 2D clipping in a uniform way. Program 5.2 describes the flow of Liang–Barsky algorithm through C code, and Solved Exercise 5.7 presents the functioning of this algorithm.

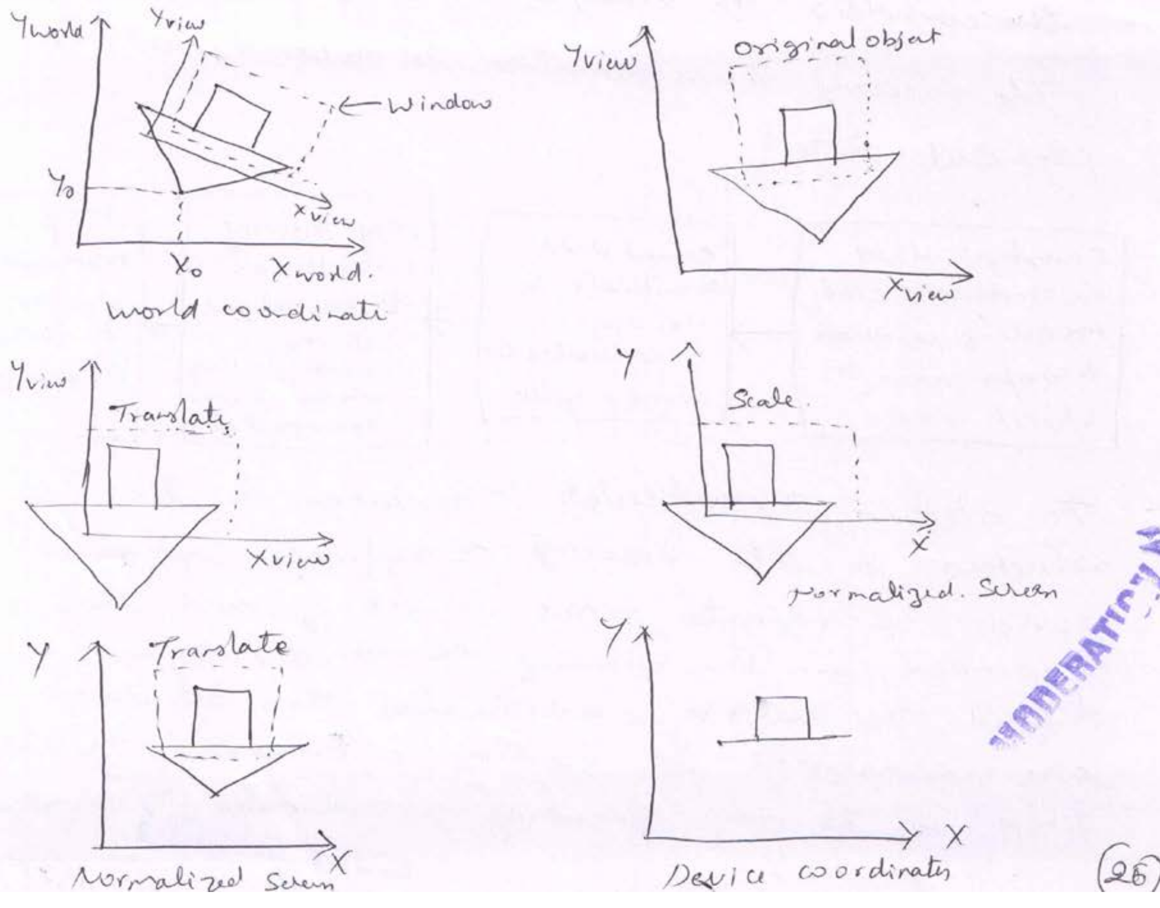
b. Explain the steps involved in 2D viewing transformation. (8)

Answer:

5(b). The mapping of a part of world coordinate Scene to device coordinates is referred to as viewing transformation.
The 2D viewing transformation is referred to as the window to viewport transformation.
The viewing transformation is carried out in several steps



To obtain a particular orientation for the window, a 2D viewing coordinate system in world coordinate plane is set up. and thus a window in the viewing coordinate system defined. The world coordinates then developed are converted to viewing coordinates. The next step is to map viewing coordinates to normalized viewing coordinates using window viewport specification. For mapping, the object together with its window is translated till the lower left corner of the window is set at the origin. The object and window is then scaled down to obtain the dimension of the viewport. The effect of this is conversion of object and window into image and viewport. The next step is another translation to move viewport to its correct position on the screen. The final transformation takes place for transforming of viewport to device coordinate along with clipping.



(25)

Q.6 a. Define Perspective and Parallel projections. What is the main difference between them? (4)

b. Derive the transformation matrix for the projection of a point (x, y, z) onto the plane z = 0 with centre of projection at (0, 0, $-z_c$). (8)

c. What do you mean by vanishing points? What are different types of vanishing points? (4)

Answer: A B and C

7.11 PERSPECTIVE PROJECTION

Plane geometry projections of objects are formed by the intersection of projector lines with the projection plane or the picture plane. Projectors are lines from an arbitrary point called the centre of projection, through each point in an object. If the centre of projection is located at a finite point in 3D space, the result is a perspective projection. The position of the picture plane relative to the object determines the size of the perspective view. When the object is in the picture plane, its perspective will be in true size. When the object is placed in front of the picture plane, its perspective will be enlarged. When the object is placed behind the picture plane, the perspective will be in reduced size. The picture plane in this case is placed between eye and the object.

All the projections converge at a point known as the vanishing point. This effect will be seen vividly by looking at a straight long road as shown below in Fig. 7.16 or a cube as shown in Fig. 7.17. Such an effect can also be noticed in photographs. Photographs are therefore a perspective view. Perspective is chiefly employed in architectural drawings.

Fig. 7.16 *Two parallel lines alongside the road, merging together at the vanishing point*

The general 4 × 4 transformation matrix is given by Matrix 7.1. If the fourth column of this general matrix [3 × 1] is non-zero, the result is a perspective 3D transformation matrix. In perspective transformation, parallel lines converge, and object size is reduced with increasing distance from the centre of projection. A single-point perspective transformation with a single vanishing point is given by

$$[x\ y\ z\ 1]\begin{bmatrix}1 & 0 & 0 & p\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} = [x\ y\ z\ px+1]$$

Here $h = px + 1 \neq 0$. The ordinary coordinates are obtained by dividing by h, i.e. $h = px + 1$, which gives

$$[x^*\ y^*\ z^*\ 1] = \left[\frac{x}{px+1}\ \ \frac{y}{px+1}\ \ \frac{z}{px+1}\ \ 1\right]$$

This gives the perspective transformation value of the coordinate when projectors are placed on the x axis. Similarly, when the projectors are placed on the y axis, it can be shown as

$$[x\ y\ z\ 1]\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & q\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix} = [x\ y\ z\ qy+1],$$

here $h = qy + 1 \neq 0$; hence the ordinary coordinates are

$$[x^*\ y^*\ z^*\ 1] = \left[\frac{x}{qy+1}\ \ \frac{y}{qy+1}\ \ \frac{z}{qy+1}\ \ 1\right]$$

Similarly, when the projectors are placed on the z axis, it can be shown as

$$[x\ y\ z\ 1]\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & r\\0 & 0 & 0 & 1\end{bmatrix} = [x\ y\ z\ rz+1],$$

here $h = rz + 1 \neq 0$; hence the ordinary coordinates are

$$[x^*\ y^*\ z^*\ 1] = \left[\frac{x}{rz+1}\ \ \frac{y}{rz+1}\ \ \frac{z}{rz+1}\ \ 1\right]$$

Now, a perspective projection onto a 2D viewing plane is obtained by multiplying an orthographic projection matrix with a perspective transformation matrix. For example, a perspective projection onto the $z = 0$ plane is obtained by

$$[T] = [P_r][P_z] = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & r\\0 & 0 & 0 & 1\end{bmatrix} \times \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & r\\0 & 0 & 0 & 1\end{bmatrix}$$

Now $[x^*\ y^*\ z^*\ 1] = [x\ y\ z\ 1] \times \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & r\\0 & 0 & 0 & 1\end{bmatrix} = \left[\frac{x}{rz+1}\ \ \frac{y}{rz+1}\ \ 0\ \ 1\right]$ in ordinary coordinates.

The above transformation is for perspective projection onto the $z = 0$ plane. With the help of geometry, one can find the coordinates of a projected point. Consider a projector on the z axis at z_c distance away from the origin, and an object at P which is projected on the $z = 0$ plane, as shown in Fig. 7.17.

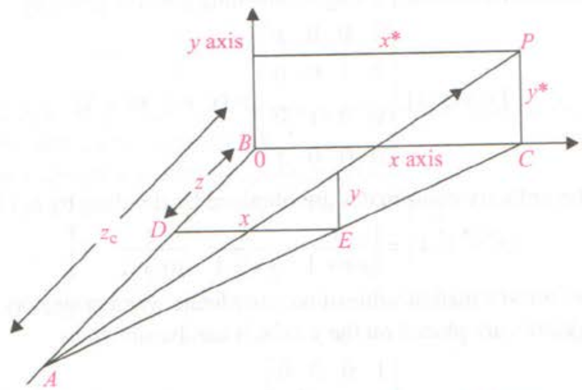


Fig. 7.17 *Finding coordinates of a projected point*

The coordinates of projected point P^* can be found by two similar triangles ΔABC and ΔADE as follows:

$$\frac{x^*}{x} = \frac{z_c}{z_c - z}$$

$$x^* = \frac{x}{\frac{z_c - z}{z_c}} \quad \text{or} \quad x^* = \frac{x}{1 - \frac{z}{z_c}}$$

and

$$\frac{y^*}{y} = \frac{\sqrt{x^{*2} + z_c^2}}{\sqrt{x^2 + (z_c - z)^2}}$$

$$= \frac{\sqrt{\left(\frac{x}{1 - z/z_c}\right)^2 + z_c^2}}{\sqrt{x^2 + (z - z_c)^2}} = \frac{\sqrt{x^2 + z_c^2\left(\frac{z}{z_c}\right)}}{\sqrt{x^2 + (z_c - z)^2}\,\sqrt{(1 - z/z_c)^2}}$$

$$= \frac{\sqrt{x^2 + (z_c - z)^2}}{\sqrt{x^2 + (z_c - z)^2}} \frac{1}{\sqrt{(1 - z/z_c)^2}}$$

or

$$\frac{y^*}{y} = \frac{1}{1 - z/z_c}$$

$$\Rightarrow \quad y^* = \frac{y}{1 - z/z_c}$$

z^* is zero because it is projected on the $z = 0$ plane.

To understand the effect of perspective transformation, consider the line PQ which is parallel to the z axis and which, after perspective transformation, lies on the xy plane when the projector is placed on the z axis. This can be shown as in Fig. 7.18.

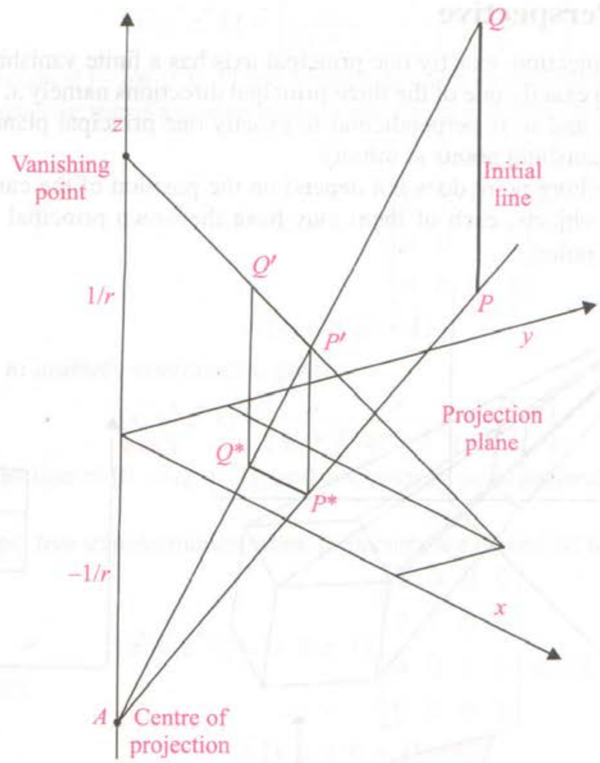


Fig. 7.18 *Perspective transformation*

Here the distance of the point *A* on *z* axis is $z_c = -1/r$ from the origin. To show that the vanishing point lies in a positive direction on the *z* axis at $1/r$ distance away from the origin, consider a point which lies on the positive axis *z* at infinity i.e. [0 0 1 0]. Now, take a perspective transformation of this point when the projector lies on the *z* axis

$$[x^* \ y^* \ z^* \ 1] = [0 \ 0 \ 1 \ 0] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = [0 \ 0 \ 1 \ r]$$

which is equivalent to [0 0 $1/r$ 1] in actual coordinates. This shows that the point is at an finite point on the positive *z* axis and that this is the vanishing point.

Q.7 a. Explain back face detection method of hidden surface removal. (10)

Answer:

7(a) An object can be well approximated using Polyhedrons. A smooth surface can be well approximated using small polygons such that normal to the plane of any elemental polygon is represented the average surface normal of the actual smooth surface at that point.

If any three points (x_1, y_1, z_1), (x_2, y_2, z_2) & (x_3, y_3, z_3) on any plane surface are known the unknown parameters A, B, C, & D of the plane surface equation

$Ax + By + Cz + D = 0$ can be found as follows:

All the three points (x_1, y_1, z_1), (x_2, y_2, z_2) & (x_3, y_3, z_3) should satisfy the equation

MODERATION-II

(29)

$Ax + By + Cz + D = 0$ as it lies on the surface

Hence,

$$Ax_1 + By_1 + Cz_1 + D = 0$$
$$Ax_2 + By_2 + Cz_2 + D = 0$$
$$Ax_3 + By_3 + Cz_3 + D = 0$$

Also, any albitrary point (x, y, z) lying on the surface should satisfy the equation of the desired surface.

$$Ax + By + Cz + D = 0.$$

A unique solution of equations for $A, B, C, \& D$ can only be obtained if.

$$\begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x & y & z & 1 \end{vmatrix} = 0 \quad \text{By Cramer's Rule}$$

or

$$\begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix} x + \begin{vmatrix} z_1 - z_2 & x_1 - x_2 \\ z_2 - z_3 & x_2 - x_3 \end{vmatrix} y + \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix} z - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} = 0.$$

which is in the form of $Ax + By + Cz + D = 0$.

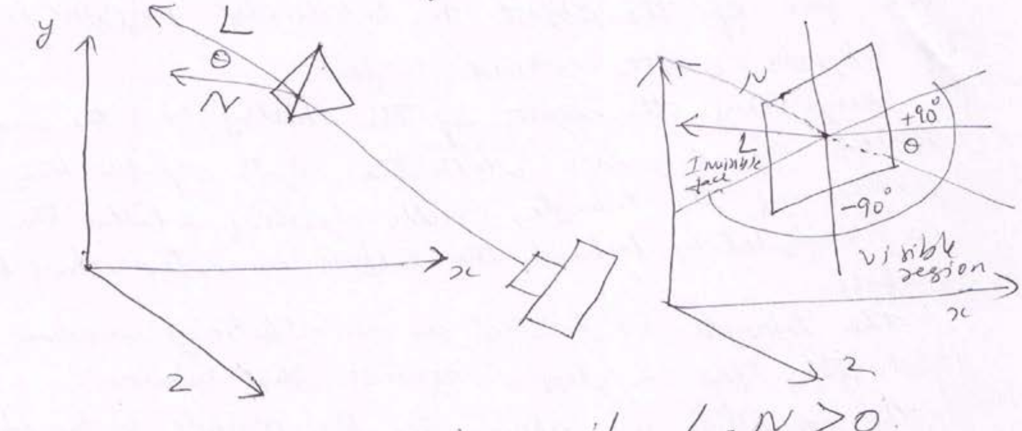
where $A = \begin{vmatrix} y_1 - y_2 & z_1 - z_2 \\ y_2 - y_3 & z_2 - z_3 \end{vmatrix}$ $B = \begin{vmatrix} z_1 - z_2 & x_1 - x_3 \\ z_2 - z_3 & x_2 - x_3 \end{vmatrix}$

$C = \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix}$ & $D = -\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$

If any point $(x, y, \text{or } z)$ lies inside the polygon surface, $Ax + By + Cz + D < 0$, and if it along the line of sight to the surface, the polygon must be back face.

— (30) —

If L is line of sight or viewing vector
N is the unit surface normal vector.



Polygon is back face if $L \cdot N > 0$.
The face becomes invisible or shadowed.
on aligning the light vector to that of the viewing vector which is normally along negative z axis for right-handed viewing system,

i.e. $L = 0\hat{i} + 0\hat{j} + L_z K$.

$L \cdot N = L_z K$,

Hence, only z component of normal vector N is required to be considered, and the sign of K is checked for negative value. Thus, any polygon is a back face if it has a surface normal with negative z component value.

The cos θ component in $L \cdot N = LN\cos\theta$ is positive for $-90°$ to $+90°$, Hence the surface is visible from the region.

Back face removal method is well applicable for convex objects, where any face can either be completely hidden or fully visible.

| Diagram – 2M matrix & equation – 4M Explanation – 4M | (31)

b. **Explain the following w.r.t. hidden line removal** **(6)**
(i) Direct method

(ii) Using visible surface detection methods

Answer:

7(b) (i) Direct method

In this method, each and every line is compared to each face of the object or subdivided polygons in case of objects with curved surface.

comparing the depth of the starting and the ending point of any line with the depth of the three vertices of the triangle, enable checking whether the line is completely behind the surface or intersecting the surface.

The triangle is treated as an albitrary window and the line is clipped against that window.

The equation of plane for the triangle is formed and the point of intersection of the line and the plane is calculated. Only the visible part of the line against the triangle is drawn.

(ii) Using visible surface detection methods.

. In back face detection method all the faces of the objects are sorted according to their depth. The face which is nearest to the viewplane is drawn first. Then the polygons in the ascending order of depth are successively drawn, clipping against the polygons that are drawn before that as window. while drawing hidden lines can be drawn in dasted, or dotted lines.

plotting part of Painter's algorithm is modified such that the faces are scan converted with surface interiors as background colour. The edges having the fore graund colour are selected. This method is possible only on raster monitors.

Screen subdivision method can be used to draw only with the edge of the faces, not the interiors of the visible surfaces, Thus leaving only the wireframe with hidden lines eliminated.

Subdivision (i) – 3 marks Subdivision (ii) – 3M

Q.8 a. Briefly explain any two different devices used for producing animation. (8)

Answer:

8 (a). The different devices used for producing animation are
Film Projectors.
Cathode Ray Tube.
Liquid crystal display.

Film projectors:- A film projector is an optical device from which light shines through a microfilm and projects the magnified image in front of it over a white wall or board. The light is made to pass through a series of lenses to focus the image marked on the microfilm properly. The projector is placed at any suitable distance from the wall or screen. The microfilm reel is made to pass through the film gate through sprocket wheels. The film is illuminated by a shuttered light, the shutter of which opens and closes as the film passes by. The lower sprocket roller pulls the film down, one frame at a time. It is also fitted with an optical sound head that reads the sound track which runs parallel to the micro film story board.

A microfilm output with the projector 35mm or 70mm generally results in more accurate translation from other colour system display than video tape.

Film projections requires less reso[illegible] of RAM, Processor speed or graphics capabilities as compared to the graphics audio visual file displays in computer generated animation.

Cathode Ray tube: (CRT)

A CRT is used on video displays on television or a computer monitor or any graphics game workstation. In this the frame buffer, is analogous to microfilm. The rapid variation in the animation Sequence is generated using a software program that manipulates the frame buffer in the computer VDU.

The rapid variation in the frame buffer is produced using Several hardware as well as Software techniques. One of the technique is to use the memory besides the frame buffer., which contains the parts of the image. A windowed area is made to move over this memory area to Scan or to convert this visible area to the frame buffer.

In the other technique use video pages available in any graphics resolution of the computer VDU. The successive frames are drawn completely on to the next pages available. The visible page is made to flip rapidly over these pages to show a series of frames to produce animation

Liquid Crystal Displays: LCD

LCD is rapidly replacing CRT used in the computer displays are well as TV displays because of its small size, weight and aesthetic quality. This contains the frame buffer and the graphics accelerators and the pixels that are arranged in grids. The LCD consists of a substance sandwitched between a pair

of wired glass known as liquid crystal, which
behaves like both a liquid and a solid. The
molecules in a liquid crystal tend to be oriented in
the same manner, much like the molecular
arrangement in a solid crystal, on application of
voltage across any given set of conductors and
are said to be polarized.

Any 2 devices 4M each

b. Explain different video formats. (8)

Answer:

8(b) NTSC Format

National Television system committee.
the NTSC standard consists one TV frame in the NTSC standard consists of two interlaced fields, each field consists of 262.5 horizontal scan lines and the whole TV frame has fixed vertical resolution of 525 interlaced horizontal scan lines stacked on top of each other, with varying amounts of "lines" making up the horizontal resolution.
It has a precise 59.94 Hz of field rate.
A field is a set of even lines or odd lines.
The odd and even fields are displayed sequentially, thus interlacing the full frame which is displayed about every 1/30th of a second.
Aspect ratio of NTSC is 4:3, Resolution 640 by 480.

various divisions within PAL.

1. PAL (720 × 486)
2. D-1 PAL (720 × 576)
3. D-1 PAL Square Pixel (768 × 576.)

MODERATION-A

-(35)-

PAL Format. -
Phase Alternation by Line
It has 625 horizontal lines making up the vertical resolution. Fifty fields are displayed and interlaced per second, making for a 25-frame-per second system.
various divisions within the PAL.
1. PAL (resolution 720 x 486)
2. D-1 PAL (resolution 720 x 576)
3. D-1 PAL Square Pix (768 x 576)

SECAM Format
Systeme Electronique couleur Avec Memoire
It has 625 lines & 25 frames per second.

RGB Format
RGB format is replacing the region-specific formats with device-specific format. It stores information in Red, Green, & Blue intensities and the horizontal synchronous signal is stored in each scan line.

| NTSC - 4m, PAL - 2m, SECAM - 1m, RGB-1m |

Q.9 a. Explain BMP file format. (10)

Answer:

9(a) BMP File format.
microsoft's windows ® bitmap (BMP) file was jointly introduced by microsoft windows & IBM.

9

A bitmap file is a binary file that may be separated into four sections.

1. File Header
2. Image Header
3. Colour Table
4. Pixel Data itself.

File header is primarily used for ascerning software to confirm that the file is a windows BMP file, to find out exactly how large the file is and to learn where the actual image data is located within the file. It is of 14-bytes in size, with the first two bytes as ASCII codes for the letters 'B' and 'M'.

The 40-bytes Image Header gives detailed information about the image and its data format, such as the height and width of the image, the number of bits used per pixel for colours and information about compressibility of image data.

The colour Table may or may not be present, depending on the format of the image data. For a 24-bit colour image it is not present. when it does exist, it is either a colour palette or a set of bit masks used to extract the colour information from the image data.

MODERATION-I (37)

The rest of the file is the pixel data.
The pixel data in a BMP file is the scan line padded to a 32-bit (4 byte) boundary.
For example: if an image has 8 bits colour per plane and size of 1024 x 221, then each scan line (horizontal row of data in the image) consists of 221 pixels, each of which requires 3-bytes each for blue, green and red (8x3 = 24-bits) to encode. Thus, 221 x 3 = 663-bytes of data per line are required. The standard format requires that scan lines must be multiples of 4-bytes so 1 null byte is added to the end of the data for each line to make a total of 664 (166x4) bytes per scan line.

Four Sections each 2M i.e 2x4=8M
Example 2M.

b. Explain compact disk and digital versatile disk. **(6)**

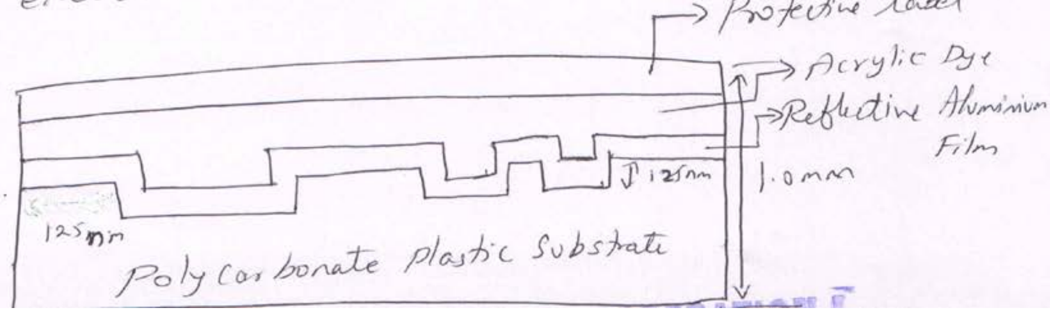
Answer:

9 (b). Compact Disk (CD)

A CD is a round plastic platter 12mm (4¾") diameter and 1mm thick with a hole in its centre. It is an optical device on which sound is recorded in a digital format by assigning digital values to electrical signals of the sound. It reproduces sound with high quality. As the method of storing and retrieving involve no physical contact between the device and the medium it eliminates wear to the optical disk.

Video storage may also be stored.

A polycarbonate layer of the CD, having the data impressed on to it, is coated with a mirror-like gold or aluminium metal film. When the digital data is prepared, the optical disk that has a light-sensitive base layer is exposed to laser light pulses as the disk spins. The pulse of laser light etches microscopic pits onto the disk and encoded form of digital information.



Digital Versatile Disk (DVD)

This is an optical storage device that resembles the compact disk and has the ability to store 15 times more as compared to CD & 20 times faster than CD-ROM.

DVD's come in two formats: DVD-Video & DVD-ROM.

DVD-video format is used for home movie entertainment using a DVD player. The DVD-ROM stores digital computer data.

CD —	Explanation	3M.
	Figure	1M.
DVD —	2M. for explanation.	

Text Book

Computer Graphics C Version, Hearn D, N. Sinha, Arun D Udai, TMH, 2008