

Q.2 a. What are the main components of Graphics system? Explain with the help of suitable diagram. (8)

Answer:

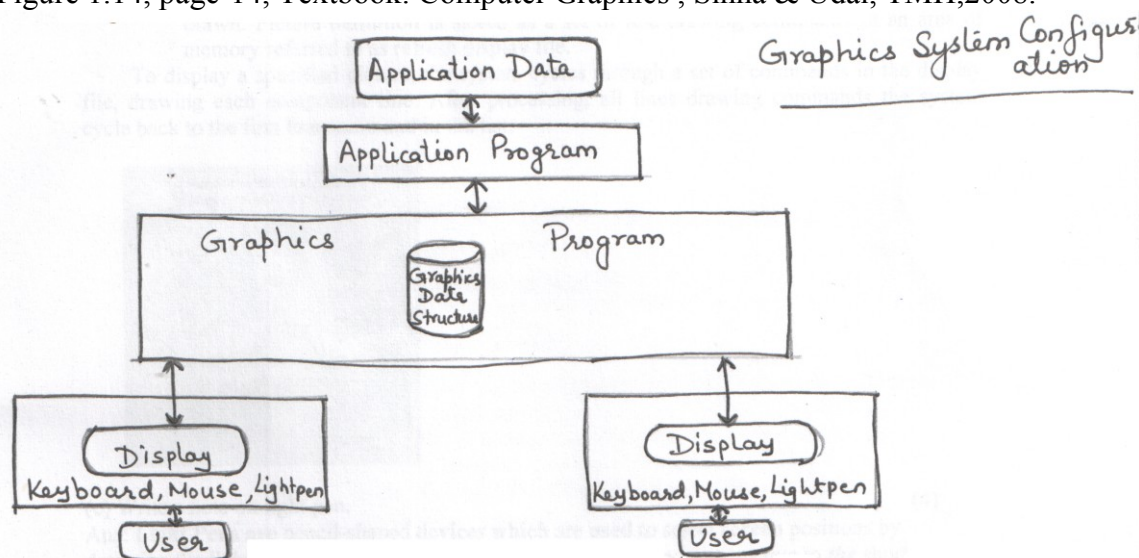
Computer Graphics is based on simple reference model which consists of 3 main components.

1. An application system
2. A graphics system
3. A user

The system consists of hardware and software. Hardware includes input & output devices for interaction and display. The software has 3 components. The first is application program which creates stores & retrieves the data or objects to be pictured on screen from second component (application data). The application program helps in producing images by sending a series of graphics output commands to third component, the graphics system. The information contains a both a detailed geometric description of what is to be viewed and the attributes describing how the images should appear. The graphics system interacts between user and the application program and is responsible for producing the picture from detailed descriptions and for passing the input to application program for processing.

Thus, the graphics system is an intermediary between application program & display device, which effects an output transformation of objects in application data, enabling the user to view the object in a particular manner and thus obtain a computer model of object in desired form. If any changes are required in computer model, user can give information to application program. Thus, 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 & represented pictorially & the mode and degree of interaction in order to create and modify the model & its visual representation. Most tasks of the programmer concern creating and editing the model and handling user interactions not actually creating pictures, since that is handled by graphics system.

Figure 1.14, page-14, Textbook: Computer Graphics , Sinha & Udai, TMH,2008.



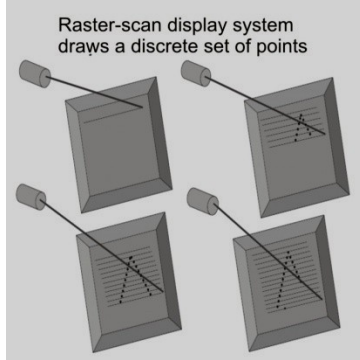
b. Highlight the major differences between raster scan and random scan display. (4)

Answer:

Raster Scan display :

- It is the most common type of monitor employing a CRT is raster scan display.

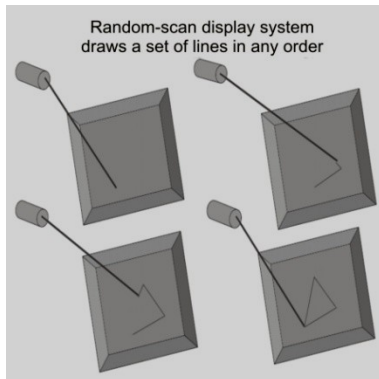
In Raster scan display the electron beam is swept across the screen one row at a time from top to bottom. Beam intensity is turned on and off to create a pattern of illuminated spots as the electron moves across each row. In a raster scan system entire screen is a matrix of pixels. Frame buffer or refresh buffer holds the set of all intensity values for all screen points. Stored intensity values are then retrieved from the refresh buffer and painted on the screen one row at a time.



Random Scan Display :

- The electron beam is directed only to the parts of the screen where a picture is to be drawn. Picture definition is stored as a set of line drawing commands in an area of memory referred to as refresh display file.

To display a specified picture, the system cycles through a set of commands in the display file, drawing each component line. After processing, all lines drawing commands the system cycle back to the first line command in the list.



c. What are the major application area of Computer Graphics?

(4)

Answer:

Computer graphics is used whenever visual representation of objects, activity, connections, values, results, positions and concepts exist or are created.

The major application of Computer Graphics.

- (1) Data presentation
- (2) Scientific visualization.
- (3) Cartography and Surveying
- (4) Simulation and animation.
- (5) Condition Monitoring.
- (6) office automation and documentation systems.
- (7) Electronic printing and publishing.
- (8) Robotics
- (9) Video Games
- (10) Manufacturing.

Q.3 a. State and explain DDA algorithm for line drawing along with its drawbacks. (6)

Answer:

The *digital differential analyzer (DDA)* is a scan-conversion line algorithm based on calculating either Δy or Δx , where $\Delta y = m \cdot \Delta x$. We sample the line at unit intervals in one coordinate and determine corresponding integer values nearest the line path for the other coordinate.

Consider first a line with positive slope. If the slope is less than or equal to 1, we sample at unit x intervals ($\Delta x = 1$) and

compute each successive y value as $y_{k+1} = y_k + m$ ----- equ 1.

Subscript k takes integer values starting from 1, for the first point, and increases by 1 until the final endpoint is reached. Since m can be any real number between 0 and 1, the calculated y values must be rounded to the nearest integer.

For lines with a positive slope greater than 1, we reverse the roles of x and y . That is, we sample at unit y intervals ($\Delta y = 1$)

and calculate each succeeding x value as $x_{k+1} = x_k + 1/m$ ----- equ 2.

Equations 1 and 2 are based on the assumption that lines are to be processed from the left endpoint to the right endpoint. If this processing is reversed, so that the starting endpoint is at the right, then either we have $\Delta x = -1$ and $y_{k+1} = y_k - m$ -----equ 3 or (when the slope is greater than 1) we have $\Delta y = -1$ with $x_{k+1} = x_k - 1/m$ -----equ 4 Equations 1 through 4 can also be used to calculate pixel positions along a line with negative slope. If the absolute value of the slope is less than 1 and the start endpoint is at the left, we set $\Delta x = 1$ and calculate y values with Eq. 1 When the start endpoint is at the right (for the same slope), we set $\Delta x = -1$ and obtain y positions **from** Eq. 3. Similarly, when the absolute value of a negative slope is greater than 1, we use $\Delta y = -1$ and Eq. 4 or we use $\Delta y = 1$ and **Eq.2**. Drawbacks: The accumulation of round-off error in successive additions of the floating-point increment, however, can cause the calculated pixel positions to drift away from the true line path for long line segments. Furthermore, the rounding operations and floating-point arithmetic are still time-consuming.

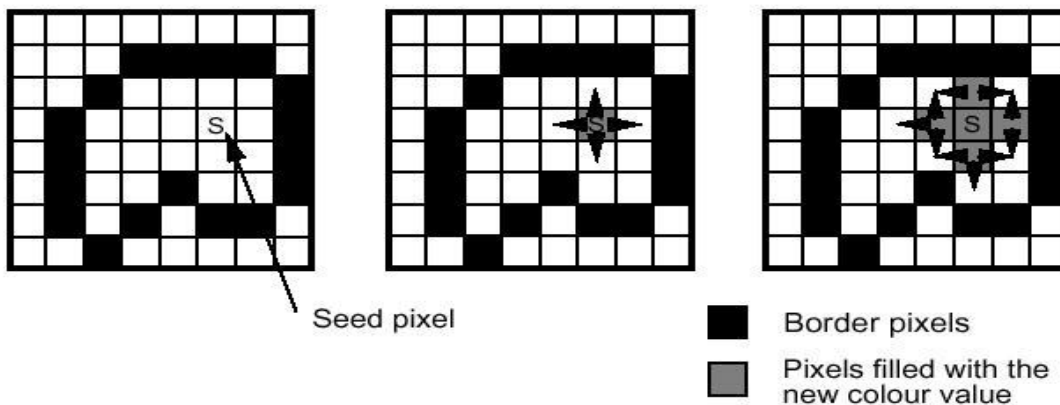
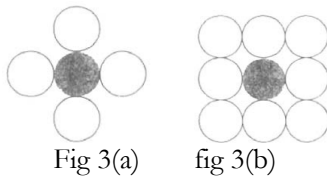
b. Define polygon filling. Explain boundary filling for a polygon with suitable example. (10)

Answer:

Filling a polygon means finding coordinates of interior points and assigning values calculated by using one of the incremental shading schemes. This requires construction of an edge list for each polygon. There are two approaches to solid area scan conversion of polygon- scan conversion and seed filling.

Boundary-Fill Algorithm

- This algorithm starts at a point inside a region called seed and paint the interior outward towards the boundary. If the boundary is specified in a single color, the fill algorithm proceeds outward pixel by pixel until the boundary color is encountered. **This** method is particularly useful in interactive painting packages, where interior points **are** easily selected. There are two methods for proceeding to neighboring pixels from the current test position. In Fig. 3(a), four neighboring points are tested. These are the pixel positions that are right, left, above, and below the current pixel. Areas **filled** by this method are called 4-connected. The second method, shown in Fig. 3(b), is **used** to fill more complex figures. Here the set of neighboring positions to **be** tested includes the four diagonal pixels. Fill methods using this approach are called 8connected.



- This is a simple method but not efficient: It is recursive method which may occupy a large stack size in the main memory.

```
void BoundaryFill(int x, int y, COLOR fill, COLOR boundary)
{
    COLOR current; current=GetPixel(x,y);
    if (current<>boundary) and (current<>fill) then
    {
        SetPixel(x,y,fill);
        BoundaryFill(x+1,y,fill,boundary);
        BoundaryFill(x-1,y,fill,boundary);
        BoundaryFill(x,y+1,fill,boundary);
        BoundaryFill(x,y-1,fill,boundary);
    }
}
```

Q.4 a. Determine the transformation matrix to reflect a polygon with vertices A(-1,0), B(0, 2), C(1, 0) about a line $y = x + 2$. (6)

Answer:

$$M_L = T_v \cdot R_{\theta} \cdot M_y \cdot R_{-\theta} \cdot T_v$$

$$M_L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[A' \ B' \ C'] = M_L \begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 & -4 & -2 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

New coordinates are :A'(-2,1), B'(-4,2),C'(-2,3)

b. Show that 2D rotation followed by scaling operation is commutative if $S_x = S_y$. (5)

Answer:

The matrix representation for a two-dimensional transformation sequence consisting of scaling followed by rotation, relative to the coordinate origin, is

$$S_{S_x, S_y} * R_{\theta} = \begin{bmatrix} S_{S_x} & 0 & 0 \\ 0 & S_{S_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_{sx} \cos \theta & -S_{sy} \sin \theta & 0 \\ S_{sx} \sin \theta & S_{sy} \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

And the matrix representation for a rotation followed by scaling, relative to the coordinate origin, is

$$R_{\theta} * S_{sx, sy} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_{sx} & 0 & 0 \\ 0 & S_{sy} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} S_{sx} \cos \theta & -S_{sx} \sin \theta & 0 \\ S_{sy} \sin \theta & S_{sy} \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

These two matrices are equivalent only when $s_x = s_y$.

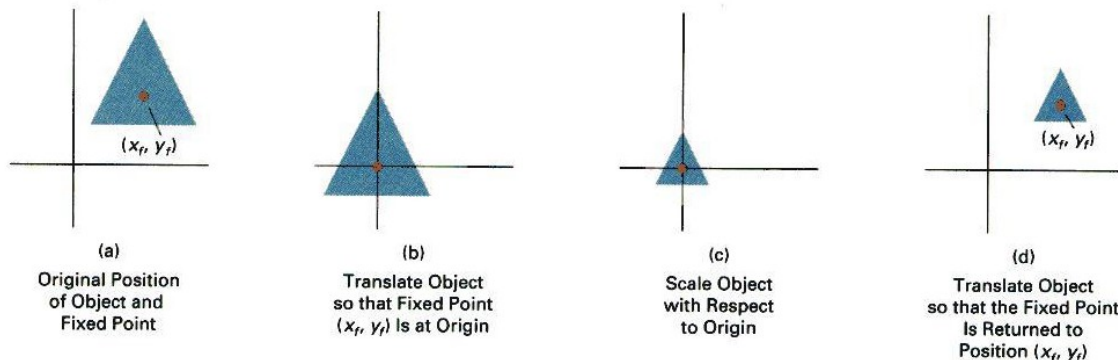
c. Explain general fixed point scaling along with its composite transformation matrix. (5)

Answer:

General Fixed-Point Scaling

Scaling with respect to an arbitrary fixed point is not as simple as scaling with respect to the origin. The procedure of scaling with respect to an arbitrary fixed point is:

1. Translate the object so that the fixed point coincides with the origin.
2. Scale the object with respect to the origin.
3. Use the inverse translation of step 1 to return the object to its original position.



The corresponding composite transformation matrix is:

Q.5 a. Explain Sutherland-Hodgeman algorithm for polygon clipping and give reason for what type of clipping regions it is not suitable? (6)

Answer:

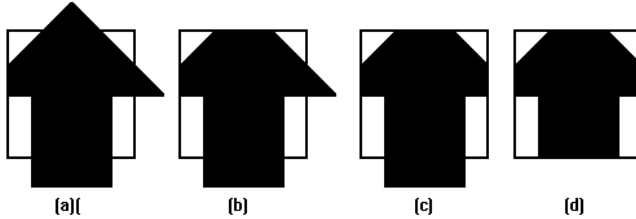
Sutherland - Hodgman Polygon Clipping

The Sutherland - Hodgman algorithm performs a clipping of a polygon against each window edge in turn. It accepts an ordered sequence of vertices $v_1, v_2, v_3, \dots, v_n$ and puts out a set of vertices defining the clipped polygon. Beginning with the initial set of polygon vertices, we could first clip the polygon against the left rectangle boundary to produce a new sequence of vertices. The new set of vertices could then be successively passed to a right boundary clipper, a bottom boundary clipper, and a top boundary clipper, as in Fig. below. At each step, a new sequence of output vertices is generated and passed to the next window boundary clipper.



Before clipping

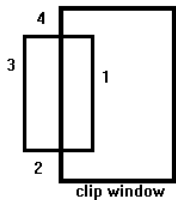
This figure represents a polygon (the large, solid, upward pointing arrow) before clipping has occurred. The following figures show how this algorithm works at each edge, clipping the polygon.



- Clipping against the left side of the clip window.
- Clipping against the top side of the clip window.
- Clipping against the right side of the clip window.
- Clipping against the bottom side of the clip window.

Four Types of Edges

As the algorithm goes around the edges of the window, clipping the polygon, it encounters four types of edges. All four edge types are illustrated by the polygon in the following figure. For each edge type, zero, one, or two vertices are added to the output list of vertices that define the clipped polygon.



The four types of edges are:

- Edges that are totally inside the clip window. - add the second inside vertex point
- Edges that are leaving the clip window. - add the intersection point as a vertex
- Edges that are entirely outside the clip window. - add nothing to the vertex output list
- Edges that are entering the clip window. - save the intersection and inside points as vertices

Some Problems with this Algorithm

- This algorithm does not work if the clip window is not convex.
- If the polygon is not also convex, there may be some dangling edges.

- b. Using Cohen-Sutherland line clipping, compute the visible portion of the line segment $A(0.6,0.8)$, $B(2.4,1.7)$ for window $(x_{min},y_{min})=(0,0)$ and $(x_{max},y_{max})=(2,2)$.** (4)

Answer:

Using Cohen-Sutherland line clipping, compute the visible portion of the line segment $A(0.6,0.8)$, $B(2.4,1.7)$ for window $(x_{min},y_{min})=(0,0)$ and $(x_{max},y_{max})=(2,2)$. (6)

Ans: Starting from leftmost bit, each bit of region code for point (x,y) is set to True (1) or false (0) as follows:

Bit 1 \equiv end point above window = $\text{sign}(y-y_{max})$

Bit 2 \equiv end point below window = $\text{sign}(y_{min}-y)$

Bit 3 \equiv end point to right of window = $\text{sign}(x-x_{\max})$

Bit 4 \equiv end point to left of window = $\text{sign}(x_{\min}-x)$

Here $\text{sign}(a) = 1$ if a is positive
0 otherwise

By Cohen Sutherland Algorithm

$A(0.6, 0.8) \rightarrow (0000)$ and $B(2.4, 1.7) \rightarrow (0010)$

Since logical AND of AB line segment is 0000, so this line is candidate of clipping.

Also, this line intersects line $x=2$

Calculate $y = y_1 + m(x - x_1) = 0.6 + 0.5(2 - 0.6) = 1.3$ where $m = (y_2 - y_1)/(x_2 - x_1)$

So, intersection point $I(2, 1.3)$

Region code for point $I \rightarrow (0000)$

Since both points A and I have region code (0000), therefore line is visible and having endpoints $A(0.6, 0.8)$ and $I(2, 1.3)$

c. Determine the parametric representation of line segment between position vector $P_1(2, 4)$ and $P_2(6, 4)$ (6)

Answer:

A line segment can be parametrically represented as

$$P(t) = P_1(t) + [P_2(t) - P_1(t)]t \text{ where } 0 \leq t \leq 1$$

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

$$y(t) = y_1 + (y_2 - y_1)t$$

Solving for x at $t=0$, $x(0) = x_1 = 2$ at $t=1$

$$x(1) = x_2 = 6$$

$$\text{Hence } x(t) = 2 + (6 - 2)t = 2 + 4t$$

$$\text{Similarly } y(t) = 4 + (4 - 4)t = 4$$

Thus the parametric representation for the line is given by

$$x(t) = 2 + 4t$$

$$y(t) = 4, \text{ where } 0 \leq t \leq 1.$$

Q.6 a. Describe the transformation matrices for 3D rotation and 3D reflection.
(6)

Answer:

$$\text{Z-Axis Rotation} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\text{Y-Axis Rotation} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\text{X-Axis Rotation} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Solution: for 3D rotation already solution is given

3D Reflection (Transformation Matrix)

Case I: for reflection through xy plane, the transformation matrix is given by the matrix

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Case II: for reflection through yz plane.

$$[T] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

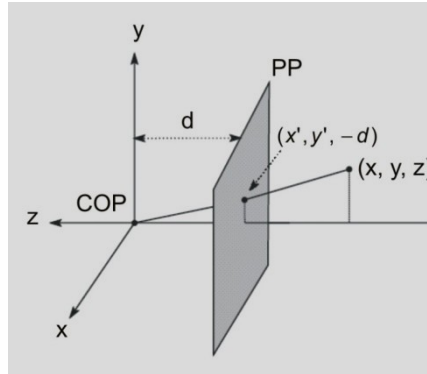
Case III: Reflection through xz plane:

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b. Describe perspective projection with the help of neat diagram. (6)

Answer:

Perspective Projection : Object positions are transformed to the view plane (PP) along the lines that converge to a point called the center of projection(COP).



When a three-dimensional object is projected onto a view plane using perspective transformation equations, any set of parallel lines in the object that are not parallel to the plane are projected into converging lines. The point at which a set of projected parallel lines appears to converge is called a vanishing point. **Each** such set of projected parallel lines will have a separate vanishing point; and in general, a scene can have any number of vanishing points, depending on how many sets of parallel lines there are in the scene. The vanishing point for any set of lines that are parallel to one of the principal axes of an object is referred to as a principal vanishing point. We control the number of principal vanishing points (one, two, or three) with the orientation of the projection plane, and perspective projections are accordingly classified as one-point, two-point, or **three-point** projections. The number of principal vanishing points in a projection is determined by the number of principal axes intersecting the view plane. Figure below illustrates the appearance of one-point and two-point perspective projections for a cube. The view plane is aligned parallel to the **xy** object plane so that only the object **z** axis is intersected.

Figure 7.19(page-202), Fig.7.20(page 203), Fig 7.21(pg 204) Textbook: Computer Graphics , Sinha & Udai, TMH,2008.

c. Define Quadric Surfaces. (4)

Answer:

Regular curved surfaces can be generated as - Quadric Surfaces, eg. Sphere, Ellipsoid, or - Superquadrics, eg. Superellipsoids

These surfaces can be represented by some simple parametric equations, eg, for ellipsoid:

$$x = r_x \cos^s \phi \cos^s \theta, \quad -\pi/2 \leq \phi \leq \pi/2$$

$$y = r_y \cos^s \phi \sin^s \theta, \quad -\pi \leq \theta \leq \pi$$

$$z = r_z \sin^s \phi$$

Where s , r_x , r_y , and r_z are constants. By varying the values of θ and ϕ , points on the surface can be computed.

Q.7 a. Explain why there is a need for visible surface detection? Also, differentiate between object precision and image precision methods for detecting visible surface. (6)

Answer:

The surfaces that are blocked or hidden from view must be "removed" in order to construct a realistic view of the 3D scene. The identification and removal of these surfaces is called the visible surface detection or

hidden – surface removal problem. The solution involves the determination of the closest visible surface along each projection line.

There are many different visible –surface detection algorithms. Each can be characterized as either an image-space method or an object-space method. An object-space method compares objects and parts of objects to each other within the scene definition to determine which surfaces, as a whole are visible. Surface visibility is determined using continuous models in the object space (or its transformation) without involving pixel based operations.

In an image-space algorithm, visibility is decided point by point at each pixel position on the projection plane. Most visible-surface algorithms use image-space methods in which the pixel grid is used to guide the computational activities that determine visibility at the pixel level.

b. Describe Back face hidden surface removal algorithm along with its limitations. (10)

Answer:

Object surfaces that are oriented away from the viewer are called back-faces. The back-faces of an opaque polyhedron are completely blocked by the polyhedron itself and hidden from view. We can therefore identify and remove these back-faces based solely on their orientation without further processing (projection and scan-conversion) and without regard to other surfaces and objects in the scene.

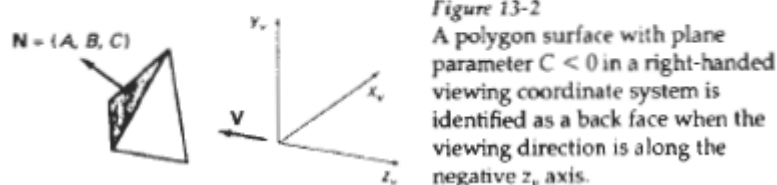
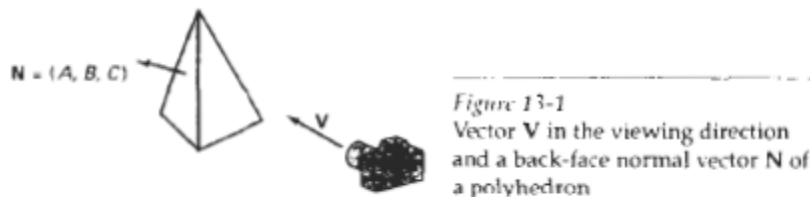
A point (x, y, z) is "inside" a polygon surface with plane parameters $A, B, C,$ and D if $Ax+By+Cz<D$.

When an inside point is along the line of sight to the surface, the polygon must be a back face (we are inside that face and cannot see the front of it from our viewing position).

We can simplify this test by considering the normal vector N to a polygon surface, which has Cartesian components (A, B, C) . In general, if V is a vector in the viewing direction from the eye (or "camera") position, as shown in Fig. 13-1, then this polygon is a back face if $V \cdot N > 0$. Furthermore, if object descriptions have been converted to projection coordinates and our viewing direction is parallel to the viewing z_v axis, then $V = (0, 0, 1)$ and so that we only need to consider the sign of C , the z component of the normal vector N .

In a right-handed viewing system with viewing direction along the negative z_v axis (Fig. 13-21, the polygon is a back face if $C < 0$. Also, we cannot see any face whose normal has z component $C = 0$, since our viewing direction is grazing that polygon. Thus, in general, we can label any polygon as a back face if its normal

vector has a z -component value: $C \leq 0$



Although this method identifies or removes back-faces quickly it does not handle polygons that face the viewer but are hidden (partially or completely) behind other surfaces. It can be used as a preprocessing step for other algorithms.

Q.8 a. What do you mean by computer-assisted animation? Differentiate it with computer-generated animation. (8)

Answer:

There are 2 main categories of computer animation

- Computer-assisted animation
- Computer-generated animation

Computer-assisted animation usually refers to 2D and 2 1/2D systems that computerizes the traditional animation process. Interpolation between key shapes is typically the only algorithmic use of computer in the production of this type of animation. Computers drastically reduce the time taken to draw intermediate frames and simplify artwork and skill requirements. Knowing the key frames, the intermediate frames can be automatically calculated by mathematical programming in computer. This is done by specifying the number of in-between frames interpolated frames required and the end frames. This method of interpolation can be linear or curvilinear to depict natural movement or for simulating accelerations or any simulating movements.

Computer-generated animation is basically concerned with motion control of objects. One of the major time consuming works of an artist of early animators is painting the sequence of frames drawn. This work can be automatically done with computer-generated frames. The method is to use seed fill algorithms and texturing & shading methods. Digital audio & video effects can be amalgamated with animation. Duplication of videos, mass reproduction, distribution, broadcasting, transfer and conversion between different video formats and data exchange have become processes that are completed in a matter of days with computer generated animation.

b. Write short notes on: (8)
(i) Frame-by-frame animation
(ii) Keyframes

Answer:

(i) This type of animation is required for creating sophisticated animation on graphics workstation with minute detail, shadows, smooth-shaded 3D objects, innumerable lightening, texturing, ray tracing scenes with global reflection & refractions. Each frame of the scene is produced one by one and is captured and stored in any suitable video frame format. Hence the complete sequence can be loaded onto frame buffer at 30 frames per second to generate an animation.

(ii) A **keyframe** is a 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* great. More key frames are specified for intricate motions than for simple, slowly varying motions.

Q.9 a. Explain the term multimedia. What are various applications of multimedia? (6)

Answer:

Multimedia is nothing but the processing and presentation of information in a more structured and understandable manner using more than one media such as text, graphics, animation, audio

and video. Thus multimedia products can be an academic presentation, game or corporate presentation, information kiosk, fashion-designing etc. Multimedia systems are those computer platforms and software tools that support the interactive uses of text, graphics, animation, audio, or motion video. In other words, a computer capable of handling text, graphics, audio, animation and video is called multimedia computer. If the sequence and timing of these media elements can be controlled by the user, then one can name it as *Interactive Multimedia*.

Common applications for Multimedia are in internet, games, producing learning or training softwares, creating reference materials such as encyclopedia, dictionary or atlas. These use visual effects and audio streamed along with the videos to make them more entertaining and effective. A graphics presentation can be in many forms: picture slides, live video, computer generated graphics and so on. These can be clubbed with audio, video, read out text written, or a music file may be played along with graphics generated. A thoughtfully presented multimedia package can enhance even static presentations like web pages: it may enhance the informative concept and interactive experience. Most applications contain hyperlinks that enable users to switch between various media elements within the subject.

Placing the media in a perspective within the instructional process is an important role of the teacher and library professional. Following are the possible areas of application of multimedia:

- Can be used as reinforcement
- Can be used to clarify or symbolize a concept
- Creates the positive attitude of individuals toward what they are learning and the learning process itself can be enhanced.
- The content of a topic can be more carefully selected and organized
- The teaching and learning can be more interesting and interactive
- The delivery of instruction can be more standardized.
- The length of time needed for instruction can be reduced.

The instruction can be provided when and where desired or necessary

b. Briefly describe how audio plays a major role in multimedia development?

(5)

Answer:

Audio has a greater role to play in multimedia development. It gives life to the static state of multimedia. Incorporation of audio is one of the most important features of multimedia, which enhance the multimedia usability to its full potential. There are several types of sound, which can be used in multimedia. They are human voices, instrumental notes, natural sound and many more. All these can be used in any combination as long as they give some meaning to their inclusion in multimedia.

- There are many ways in which these sounds can be incorporated into the computer. For example;
- Using microphone, human voice can directly be recorded in a computer.
- Pre-recorded cassettes can be used to record the sound into computer.
- Instrumental sound can also be played directly from a musical instrument for recording into the computer.

c. What is CD-ROM drive and how it differs from DVD?**(5)****Answer:**

CD-ROM is a magnetic disk of 4.7 inches diameter and it can contain data up to 680 Megabytes. It has become a standard by itself basically for its massive storage capacity, faster data transfer rate. To access CD-ROM a very special drive is required and it is known as CD-ROM drive. ROM stands for 'Read Only Memory'. It means the material contained in it can be read (as many times, as you like) but the content cannot be changed.

As multimedia involves high resolution of graphics, high quality video and sound, it requires large amount of storage space and at the same time require a media, which can support faster data transfer. CD-ROM solves this problem by satisfying both requirements. Similar to the hard disk drive, the CD-ROM drive has certain specification which will help to decide which drive suit best to your multimedia requirement.

DVD (Digital Versatile Disk) is an optical storage device that resembled the compact disk and has the ability to store about 15 times as much information and to transfer it to computer almost 20 times as fast as a CD-ROM depending on computer architecture. DVD comes in 2 formats: DVD video & DVD ROM. DVD-video is used for home movie entertainment using DVD player, DVD-ROM stores digital computer data. DVD-ROM can store interactive computer games, video file storage and photographic storage. As compared to CD-ROM technology, DVD allows better graphics with higher resolution and increased storage capacity.

TEXT BOOK

- I. Computer Graphics, Amarendra N. Sinha, Arun D Udai, TMH, 2008