Date.................... Page No. 01 Experiment No....................

Name of Experiment: Computer Graphics

Experiment Result....................

I

1) what is refresh CRT?
It is the total no of times per second the pixels are displayed on the Screen is called refresh rate CRT.

CRT → A cathode ray tube is vaccume tube in which images are produced when an electron beam hits phosphor coated screen.

2) Define Ellipse?
A ellipse is defined as a set of point such that the sum of the distance from the 2 fixed position is same for all points is called Ellipse.

3) What is blanking?
All the parts outside the region must be retained but within the window boundary region is discarded is called blanking (or) Exterior clipping.

4) what is a viewport?
An area on a display device to which a window is mapped is called a viewport.

5) Define reflection?
A reflection is a transformation that produces a mirror image of an object.

6) what is composite transformation?
It is a Sequency of transformation (or)

when a transformation is applied one (2) more time to the same object then that transformation is called composite transformation.

7) What do you mean by device coordinate system?

8) What is parallel projection?

Parallel projection generating a view of a solid object by projecting points on the object surface along Parallel line onto the display plane.

9) What is the use of control points?

Control points are used to decide the shape of the curve.

10) what is intensity cueing?

Intensity cueing is a method for indicating depth with wire frame displays is to vary the intensity of objects according to their distance from the view position.

11) what is constraints?

Constraint is a set of rule for altering input co-ordinate values to produce a specified alignment of the displayed co-ordinates.

12) Mention the combinational keys of Keyboard.

Ctrl, ALt, shift

Date.................... Page No. 03 Experiment No....................

Name of Experiment....................

Experiment Result....................

II

13) What are the difference between random and raster displays?

Raster	Random
* It has less resolution	* It has high resolution
* The lines produced in zigzag manner	* It produce smooth lines.
* Cannot used in animation	* used in animation
* less cost	* high cost.
* Easy to produce high realistic images with shading & hidden surface techniques.	* Difficult to produce realistic images.

14) Write a program to draw a circle using DDA tech

```
#include <stdio.h>
# include <conio.h>
#include <graphics.h>
#include <math.h>
void main()
{
    int gd=DETECT, gm, tmp, i=1, r;
    float x, y, x1, y1, x2, y2, ep;
    initgraph(&gd, &gm, "c:\\tc\\bgi");
    printf("Enter Radius:");
    scanf("%d", &r);
    while(r > pow (2,i))
     i++;
     ep = 1/ pow (2,i);
     x1 = r
     y1 = 0
```

```
x=r
y=0;
do
{
    x2=x1+(y1*ep);
    y2=y1-(x2*ep);
    putpixel(x2+200, y2+200, 6);
    x1=x2;
    y1=y2;
    delay(100);
}
while ((y1-y)<ep // (x-x1)>ep);
getch();
}
```

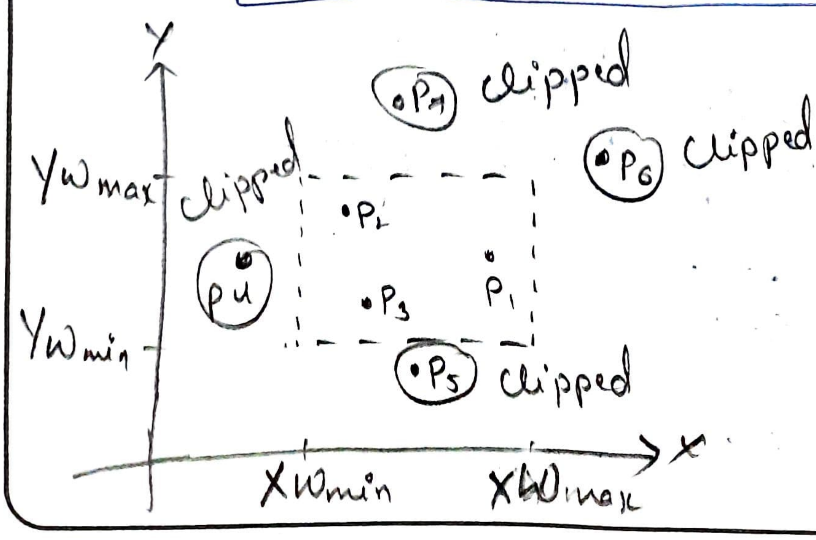
15) Explain point clipping.

Clipping of points is done by comparing each point with the boundary if the rectangle window.

Point clipping against a window specification is done by testing the co-ordinate values to deter-mine whether they are within the boundaries or not. using the following conditions:

$$\boxed{\begin{array}{l} Xw_{min} <= X <= Xw_{max} \\ Yw_{min} <= Y <= Yw_{max} \end{array}}$$

The point (x, y) lies inside the clip area.

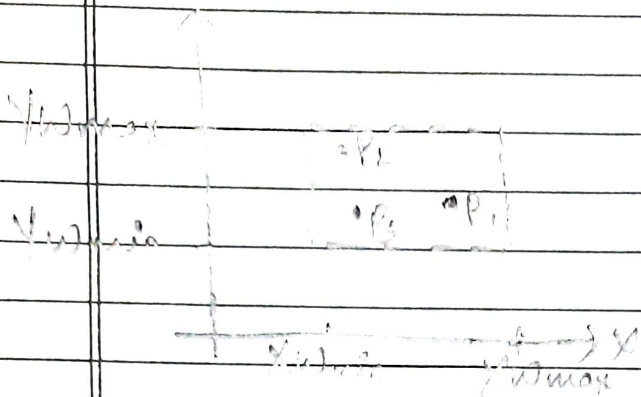


Points P4, P5; P6, P7 are outside the window are clipped.

Name of Experiment

Experiment Result

After point clipping



Points P_1, P_2, P_3 are within the window are not clipped

16) Explain homogeneous transformation.

An animation involves sequence of geomatric transformation we need to perform translation, rotations & scaling to the picture into their proper positions.

We must calculate transform co-ordinate one step at a time.

1. First, co-ordinate positions are scale.
2. Then the scale co-ordinates are rotated
3. Finally the rotated co-ordinates are translated.

1) Translation:

The translation is represented as a multiplication the 3x3 matrix is

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\boxed{P_{new} = T(t_x - t_y)\,P}$$

2) Rotation:

Rotation transformation equation about the co-ordinate origin are written as

$$\begin{bmatrix} x_{new} \\ y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\boxed{P_{new} = R(\theta).P}$$

3) <u>Scaling:</u>

Scaling transformation related to the co-ordinate origin are written as

$$\begin{bmatrix} x_{new} \\ y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\boxed{P_{new} = S(S_x . S_y).P}$$

17) Explain uniform scaling transformation with an example.

It is the ability to change the size of the object. This can be done by multipling the co-ordinate values x, y of each vertex while scaling factor S_x and S_y to produce tranform co-ordinate x_{new} & y_{new}

The scaling equation in matrix form

$$P\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = P\begin{bmatrix} x \\ y \end{bmatrix} * S\begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

The scaling equation 3*3 matrix

$$P_{new}\begin{bmatrix} x_{new} \\ y_{new} \\ 1 \end{bmatrix} = P\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} * \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Name of Experiment

Experiment Result

$$x_{new} = x * S_x$$
$$y_{new} = y * S_y$$

Any positive numeric values can be assign to the scaling factor S_x & S_y
value >1 produce enlarge object
value < 1 produce compressed (decrease size) object.
value = 1 the size of object will not change regains the same.

18) Explain the properties of curves.

i) Control point: Control point or knots are used to control the shape of a curve. The curve must pass through the point that control the curve's shape in a predictable way.

Control points indicate by points.

ii) Multiple value: A curve is not a graph of a single valued function. It may be multivalued with all co-ordinate system

iii) Axis Independence: The shape of an object must not change when the control points are calculated in a different co-ordinate system.

iv) Global or Local control: A designer can manipu-lates the control. That may change shape only near the control point or change the shape

Altered control point

original Control point

v) Versatility: A curve that allows limited variety of shapes. The designer can control the versatility of a curve by adding (or) removing the control point.

vi) Order of Continuity: A complex shape is created by joining several curves together end-to-end.

19) Explain basic functions of segments.

There are 3 basic functions are

1) open Segment(n): open a display file segment.
2. Close Segment : Close the open Segment.
3. Delete Segment(n): Remove from the display file segment.

To Create a new segment, we open it and then call graphics primitives to add to the segment the lines and text to be displayed. Then close the segment. To remove a segment from the display file, we delete it.

III 20) Briefly explain color model.

A color model is an abstract mathematical model describing the way of colors can be represented as tuples of numbers.

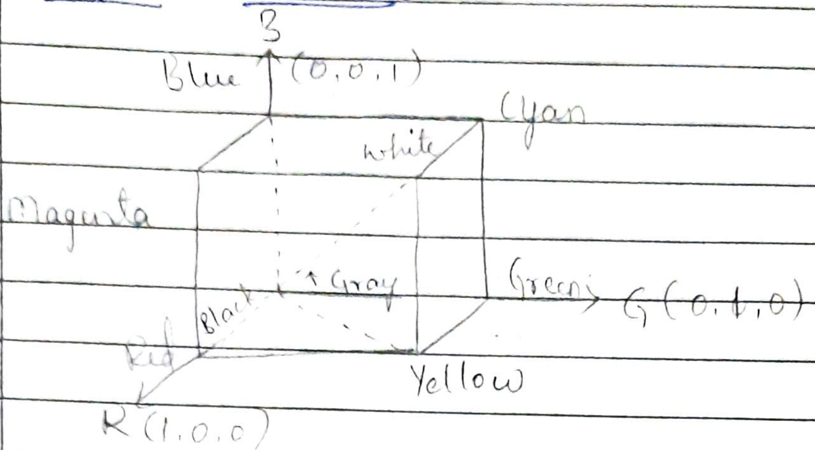
The three color models are:

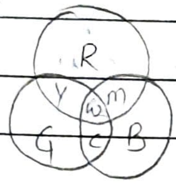
1. RGB color model
2. CMY color model
3. YIQ color model

Name of Experiment.........

Experiment Result.........

1) RGB Color model:





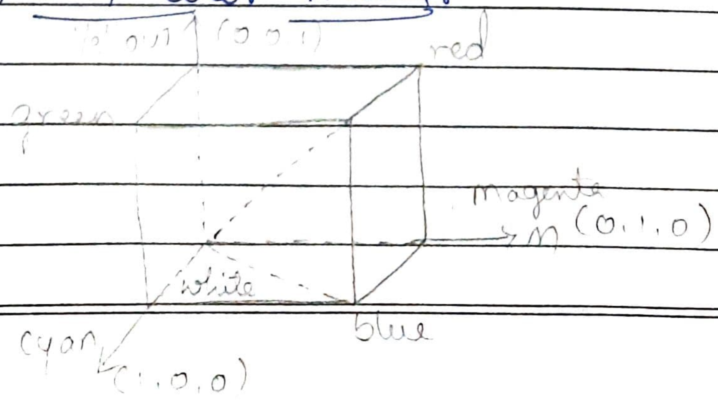
Additive Colors.

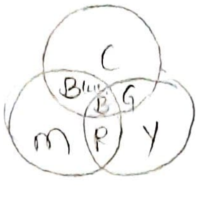
The RCB color model is composed by using primary colors. Red, Green and Blue. This model uses cartion co-ordinate system. It is used in color CRT monitors and color Raster graphics. This model is also called additive. The RGB values are at 3 corners and cmy are at other there corners white color is at the corner of the origin (forth -est) and black is there origin.

The 1° colors can be added to produce the secondary colors.

Eg: Magenta = Red + Blue
Yellow = red + Green
Cyan = Green + Blue

2) CMY color model:





The CMY color model is a sub-set of RGB color model it is mainly used in color print production these colors are called subtractive colors. It means Cyan, ~~Mag~~ Magenta, Yellow & Black inks are applied to white surface to subtract same color from white to create a another color.

Eg: Cyan = white - red
Megenta = white - green
Yellow = white - blue

3) YIQ model:

This color model is used by the US commercial color Television Board casting.

It is a re-coding of RGB for transmites efficient and compactibility for black and white television.

The Y channel contains Luminance information I and Q channels contains color information.

The Luminance can be computed as a sum of RGB components.

The color dif on chrominance components are formed by subtracting Luminance from RGB.

21) Write an algorithm to draw a straight line using Bresenham's tech and trace with 2 end points (20, 10) and (30, 18).

$dx = (x_2 - x_1)$
$30 - 20$
$\boxed{dx = 10}$

$dy = (y_2 - y_1)$
$18 - 10$
$\boxed{dy = 8}$

$P_0 = 2dy - dx$

Name of Experiment........

Experiment Result........

$P_0 = 2(18) - 30$

$= 36 - 30$

$\boxed{P_0 = 6}$

K	P_0	x , y	
0	6	21	11
1	2	22	12
2	-2	23	12
3	14	24	13
4	10	25	14
5	6	26	15
6	2	27	16
7	-2	28	16
8	14	29	17
9	10	30	18

$P_0(6) < 0$ — F

$P_{k+1} = P_k + 2dy - 2dx$

$6 + 2(8) - 2(10)$

$= 6 + 16 - 20$

$\boxed{P_1 = 2}$

$P_1(2) < 0$ — F

$P_{k+1} = 2 + 2(8) - 2(10)$

$2 + 16 - 20$

$\boxed{P_2 = -2}$

$P_2(-2) < 0$ — T

$\cancel{P_{k+1} = -2 + (2)(8) - (2)(10)}$

$= \cancel{-2 - 16 + 20}$

$P_{k+1} = P_k + 2dy$

$= -2 + 2(8)$

$= -2 + 16$

$\boxed{P_3 = 14}$

$P_3(14) < 0$ — F

$P_{k+1} = P_k + 2dy - 2dx$

$= 14 + 2(8) - 2(10)$

$= 14 + 16 - 20$

$\boxed{P_4 = 10}$

$P_4(10) < 0$ — F

$P_{k+1} = 10 + 2(8) - 2(10)$

$\boxed{P_5 = 6}$

$P_5(6) < 0$ — F

$P_{k+1} = 6 + 16 - 20$

$\boxed{P_6 = 2}$

$P_6(2) < 0$ — F

$P_{k+1} = 2 + 16 - 20$

$\boxed{P_7 = -2}$

$P_7(-2) < 0$ — T

$P_{k+1} = -2 + 16 - 20$

$\boxed{P_8 = 14}$

$P_8(14) < 0$ — F

$P_{k+1} = 14 + 16 - 20$

$\boxed{P_9 = 10}$

$P_9(10) < 0$ — F

22) Briefly explain character attributes.

The appearance of the displayed characters is control by its attributes. Attributes can be set both for the entire character strings [text] (or) Individual characters.

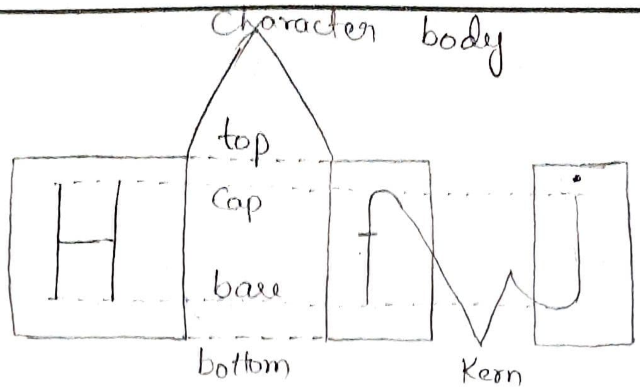
1. Font (typeface)
2. Style
3. Color
4. Size (width / height)
5. Orientation
6. Path
7. Spacing
8. Alignment

1. Font (typeface): The text can be set with a font such as Times New Roman, Courier etc. by using the function SetTextFont(tf).

2. Style: The character can be set with a font style such as Italic, bold & underline etc. By using the function SetTextStyle(ts).

3. Color: The text can be displayed with different color using the function SetTextColorIndex (tc).

4. Size: We can adjust the textsize by scaling the overall dimensions of the character using the function SetCharacterHeight(ch) where ch is assigned a value greater than 0.



The character height is defined as the distance between the capline and baseline.

Eg: Height 1
Height 2
Height 3
(effect of the different height)

The width of the text can be set using the function SetCharacterExpansionFactor(cw).

Eg: width 0.5
width 1.0
Width 2.0
(effect of different character width)

5) Orientation: The orientation is used to display the characters according to the direction of the character upvector using the function SetCharacterUpVector(upvect)
-upvect parameters takes two values.
Eg: upvect = (1,1), then the direction of the text is 45°.

upvector

6) Path: Path is used to arrange the string vertically (or) horizontally using the function SetTextPath(tp) where tp can be left, right, upper down.

V
e
r
t
i
c
a
l

Horizontal

Name of Experiment..............

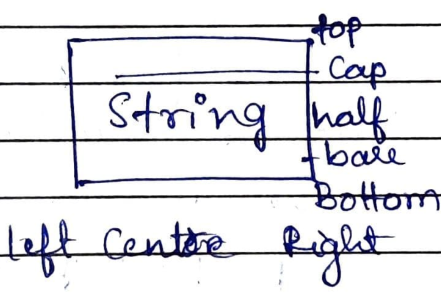
Experiment Result..............

7) Spacing: Spacing is used to give the space between the characters using the function SetCharacterSpacing (cs)

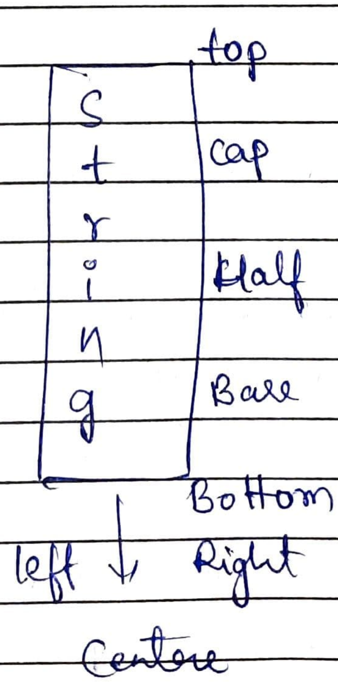
Eg: Spacing 0.0
Spacing 0.5
S p a c i n g 1.0

8) Alignment: Alignment attributes can be sent using the function SetTextAlignment (h, v)
where h-controls the horizontal alignments by assigning the value of left, centre, right.
where v-controls the vertical alignment by assigning the value of top, cap, half, base @ bottom.

Eg:



Horizontal alignment



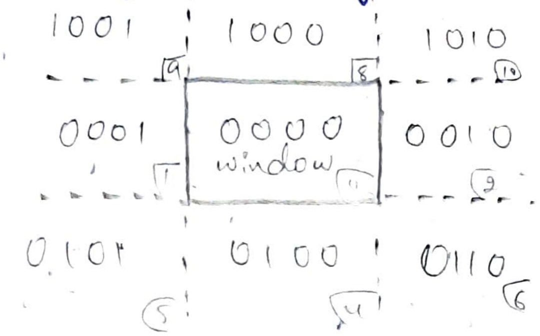
Vertical alignment

23) Explain Cohen and Sutherland line Clipping algorithm.

It is invented by Dan cohen & Ivan Sutherland. It is the efficient algorithm which performs initial tests on a line.

The algorithm divides a 2D space into 9 regions of which only the middle part (viewport) is visible.

Every line end point in a picture is assigned a 4-digit binary code, called a region code.



4bit	3bit	2bit	1bit
above	below	right	left

All lines fall into one of these categories.
1. Both endpoints lie inside the window → Accept
2. Both endpoints outside the window → reject
3. Neither 1 nor 2 then clip part of line outside the borders.

Algorithm:
1. A 4-bit region code number is assigned to an endpoint (x,y).
2. Once the region codes for all line endpoints are known, then we can determine which line are completely inside the window (or) outside the window.
3. The lies within the window have a region code of 0000 for both endpoints accept the line.
4. The lines which outside the window are rejected.

Name of Experiment..........

Experiment Result..........

6. Perform the logical AND operation with both region codes.
7. If the result is not 0000, it means line is completely outside the window. Rejected
8. If the result is 0000, the line is neither inside nor outside
9. Then Intersection points are calculated using equation parameters.

For a line (x_1, y_1) & (x_2, y_2)

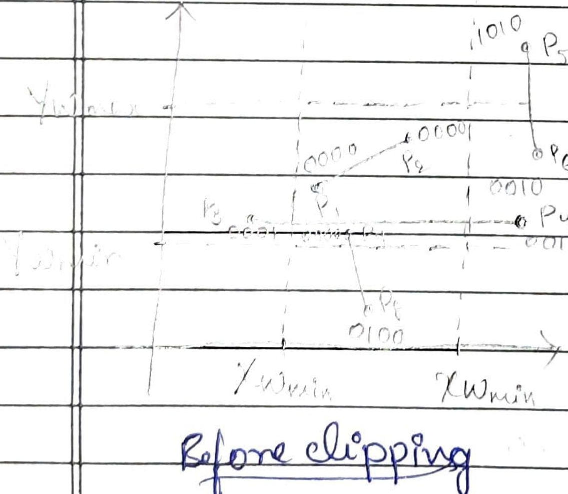
The x co-ordinate of the intersection point with a horizontal boundary can be calculated

$$\boxed{x = x_1 + \frac{(y - y_1)}{m}}$$ where $y = Yw_{min}$ or Yw_{max}

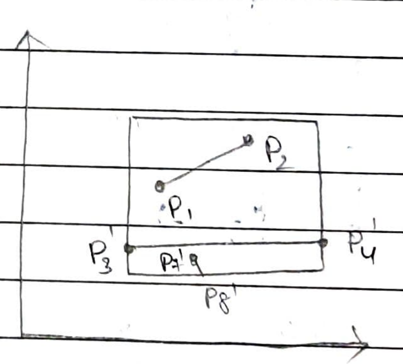
The Y co-ordinate of the intersection point with a horizontal boundary can be calculated

$$\boxed{y = y_1 + m(x - x_1)}$$ where $x = Xw_{min}$ or Xw_{max}

Slope of the line $\boxed{m = \frac{(y_2 - y_1)}{(x_2 - x_1)}}$



Before clipping



After clipping

$\boxed{P_1 \& P_2}$ 0000
Logical 0000
AND 0000
Not clip

$\boxed{P_5 \& P_6}$ 1010
0010
1000
clip

$\boxed{P_3 \& P_4}$ 0001
0010
0000
Not clip

$\boxed{P_7 \& P_8}$ 0000
0100
0000
Not clip

24) a) Write a note on Shear transformation.

The shape of an object can be distorted (twist (or) pull) by sliding internal layers of the original object is called shear.

Two types of shear transformation

1. X - direction shear.
2. Y - direction shear.

1. X-direction:

In x-direction shear related to the x-axis is produced with the transformation matrix. It changes only the x - coordinates.

* Each point in the image is display horizonally by a distance proporitional to the Y co-ordinate.

The matrix representation of x-direction shear.

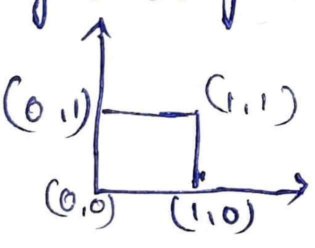
$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

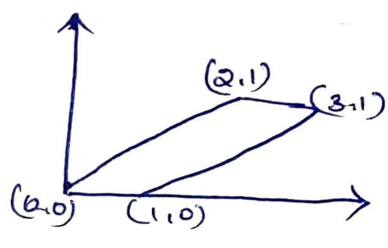
The transform co-ordinate positions are

$X_{new} = x + sh_x y$

$Y_{new} = y$

Eg: Terning the ~~square~~ square into parallelogram by setting $sh_x = 2$





for $(0,0) = X_{new} = 0 + 2.0 = (0,0)$

$Y_{new} = 0$

for $(1,0) = X_{new} = 1 + 2.0 = (1,0)$

$Y_{new} = 0$

for $(1,1) = X_{new} = 1 + 2.1 = (3,1)$

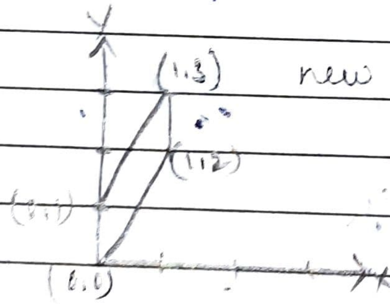
$Y_{new} = 1$

for (0,1) = x_{new} = 0+2.1 = (2,1)
y_{new} = 1

2. Y-direction shear:
A Y-direction shear relative to Y-axis is produce with the transformation matrix.
It changes only the Y- co-ordinates each point in the image is displayed vertically by a distance propotional to the x-coordinate.
The matrix representation of Y-shear.

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

The tranform co-ordinates positions are
$x_{new} = x$
$y_{new} = x\, sh_y + y$
Eg: Turning the square into parallelogram by setting $Sh_y = 2$

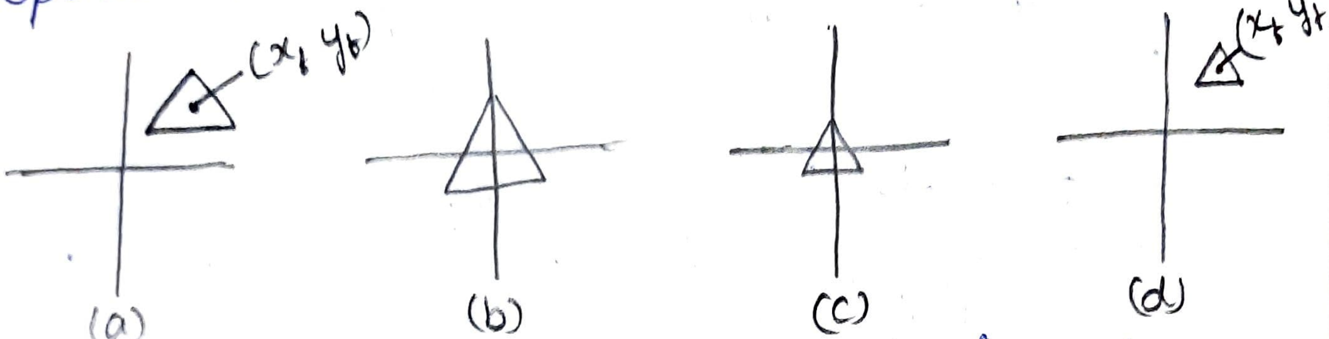


new position are

for (0,0) = x_{new} = 0
y_{new} = 0.2+0 = (0,0)
for (0,1) = x_{new} = 0
y_{new} = 0.2+1 = (0,1)
for (1,1) = x_{new} = 1
y_{new} = 1.2+1 = (1,3)
for (1,0) = x_{new} = 1
y_{new} = 1.2+0 = (1,2)

(b) Explain fixed point scaling transformation.

The graphics package provides a scaling function for scaling the object about the co-ordinate origin for this we need to perform the following sequence translate - scaling - translate operation.



(a) Original position of an object & fixed point
(b) translate object so that fixed point (x_f, y_f) is at origin.
(c) Scaling the object with respect to the co-ordinate origin.
(d) Translate the object so that the fixed point is return to the position to the position (x_f, y_f)

The composite transformation matrix is:

$$\begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} S_x & 0 & x_f(1-S_x) \\ 0 & S_y & y_f(1-S_y) \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(x_f, y_f) \cdot S(S_x, S_y) \cdot T(-x_f, -y_f)$$

$$\boxed{S(x_f, y_f, S_x, S_y)}$$

Name of Experiment..........

Experiment Result..........

25) Explain 3D rotational transformation.

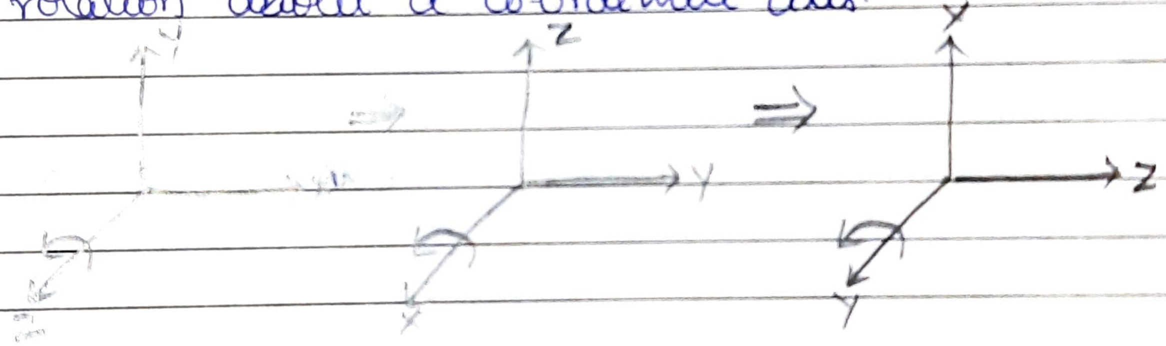
3D rotation:

To rotate an object in 3D tranformation we must decide 2 things.

- An axis of rotation
- The angle θ to be rotated

Rotation is the ability to repositioning an image along a circular path in the x,y,z plane by specifying the angle of rotation θ which the object is to be rotated.

Position rotation angles produce counter clockwise rotation about a co-ordinate axis.



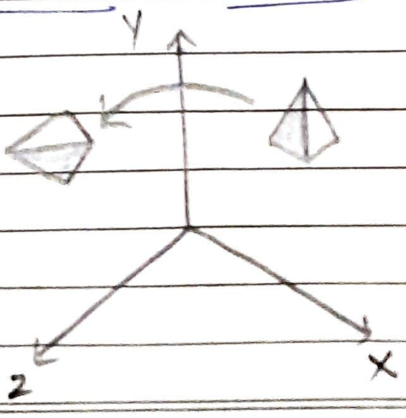
3 types of 3D rotation

(a) x - axis rotation

(b) y - axis rotation

(c) z - axis rotation

(a) x - axis rotation:



Transformation equations for x-axis rotation

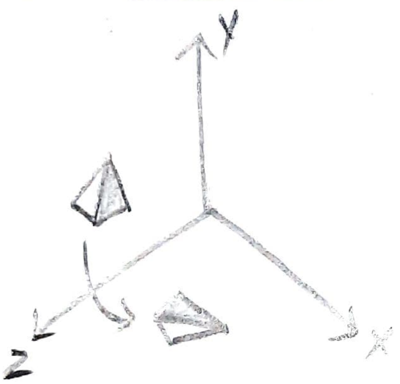
$$\boxed{\begin{aligned} y' &= y\cos\theta - z\sin\theta \\ z' &= y\sin\theta + z\cos\theta \\ x' &= x \end{aligned}}$$

The matrix form is

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

$$\boxed{P' = R_x(\theta) \cdot P}$$

(b) Y-axis rotation:



Transformation equation for Y-axis rotation

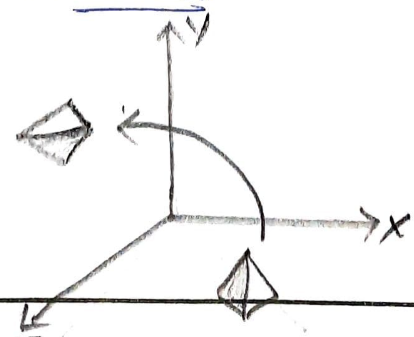
$$\boxed{\begin{aligned} z' &= z\cos\theta - x\sin \\ x' &= \cancel{z\sin\theta}\ z\sin\theta + x\cos\theta \\ y' &= y \end{aligned}}$$

The matrix form is

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

$$\boxed{P' = R_y(\theta) \cdot P}$$

(c) Z-axis rotation:



Name of Experiment............

Experiment Result............

Transformation equation along z-axis

$$\boxed{\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \\ z' &= z \end{aligned}}$$

The matrix form is

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

$$\boxed{P' = R_z(\theta).P}$$

26) Write a note on polygon table

The graphics package organizes the polygon surface data into table.

The table may contain geometric, topological and attribute properties.

Polygon data table can be organized into 2 groups.

→ Geometric tables

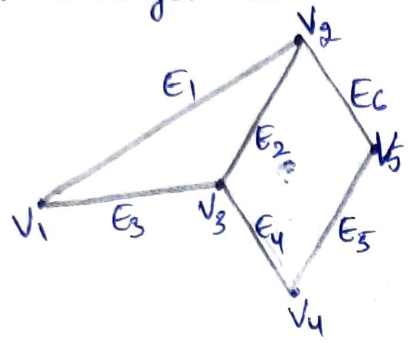
→ Attribute tables

* Geometric data tables: It contains vertex co-ordi-nates and parameters to identify the orientation of the polygon surfaces.

To store geometric data 3 table are created

1) A vertex table
2) An edge table
3) A polygon table

Vertex table stores the co-ordinate values of each vertex in the object.
Edge table consists of a pointer to each endpoint of that edge.
Polygon table defines a polygon by providing pointers to the edges that make up the polygon.



Vertex table
V_1: x_1, y_1, z_1
V_2: x_2, y_2, z_2
V_3: x_3, y_3, z_3
V_4: x_4, y_4, z_4
V_5: x_5, y_5, z_5

Edge table
E_1: V_1, V_2
E_2: V_2, V_3
E_3: V_1, V_3
E_4: V_3, V_4
E_5: V_4, V_5
E_6: V_2, V_5

Polygon Surface table
S_1: E_1, E_2, E_3
S_2: E_2, E_4, E_5, E_6

* Attribute table: Contains parameters specifying the degree of transperency and texture of the object

27) (a) Explain rubber band method.

Straight lines, Rectangles & circles can be constructed and positioned using Rubber-Band-Methods. we first select a screen position for one end point of the line, as the cursor moves around the line stretches out from the starting position to the current position of the cursor. when we finally select second position then it constructs line.

(b) Explain light pen.

- Light pen is a pointing device similar to a pen. It is used to select a displayed menu item (or) draw picture on the monitor screen. It consists of a photocell and an optical system placed in a small tube.
- When the tip of a light pen is moved over the monitor screen and the pen button is pressed, its photocell sensing element detects the screen location and sends the corresponding signal to the CPU.

IV

28) Explain ellipse generating algorithm.

Algorithm for ellipse

Step 1: Input r_x, r_y and center of ellipse (x_c, y_c) calculate first point as $(x_0, y_0) = (0, r_y)$

Step 2: For region 1
Compute initial decision parameter value as

$$\boxed{P1_0 = r^2_y - r^2_x r_y + \frac{1}{4} r^2_x}$$

Step 3: At each x_k position in region 1, with $k=0$ test $P1_k$
if $P1_k < 0$, then the next point (x_k+1, y_k)
then, $\boxed{P1_{k+1} = P1_k + r^2_y + 2r^2_y x_{k+1}}$

if $P1_k > 0$, then the next point (x_k+1, y_k-1)
then, $\boxed{P1_{k+1} = P1_k + r^2_y + 2r^2_y x_{k+1} - 2r^2_x y_{k+1}}$

Step 4: Determine symmetry points in the other 3 quadrants & plot them.

Teacher's Signature : ____________

Step 5: Repeat step 3 & step 4 while $2r_y^2 x > = 2r_x^2 y$

Step 6: for region 2

Calculate the initial value of the decision parameter using the point (x_0, y_0) for region 1 as

$$\boxed{P2_0 = r_y^2 (x_0 + 1/2)^2 + r_x^2 (y-1) - r_x^2 r_y^2}$$

Step 7: if $P2_k <= 0$ then the next point is (x_k+1, y_k-1)

$$\boxed{P2_{k+1} = P2_k + r_x^2 + 2r_y^2 x_k + 1 - 2r_y^2 y_{k+1}}$$

Step 8: Determine the symmetry points in othe 3 quadra-nt & plot them

Step 9: Repeat Step 7 & 8 while $y > 0$

29) (a) write a program to perform scaling transformation. Explain with suitable example.

(b) Explain properties of lines..

(a) Line type

(b) Line width

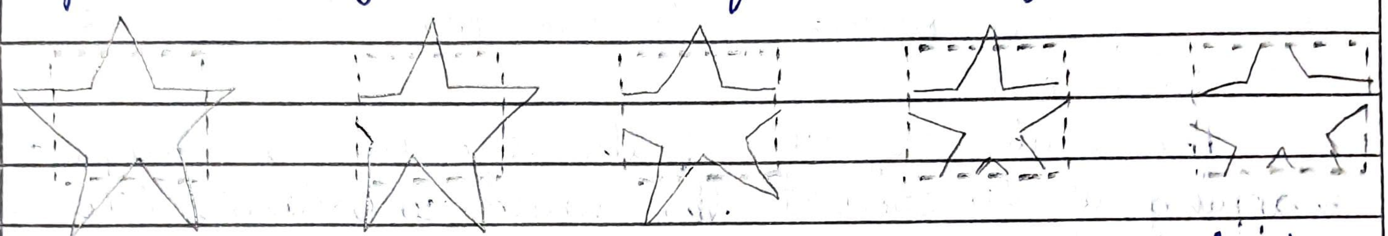
(c) Line color

(a) Line type:

30) Explain Sutherland and Hodgeman polygon clipping.

It is user for polygon clipping. It operates on the vertices of the polygon.

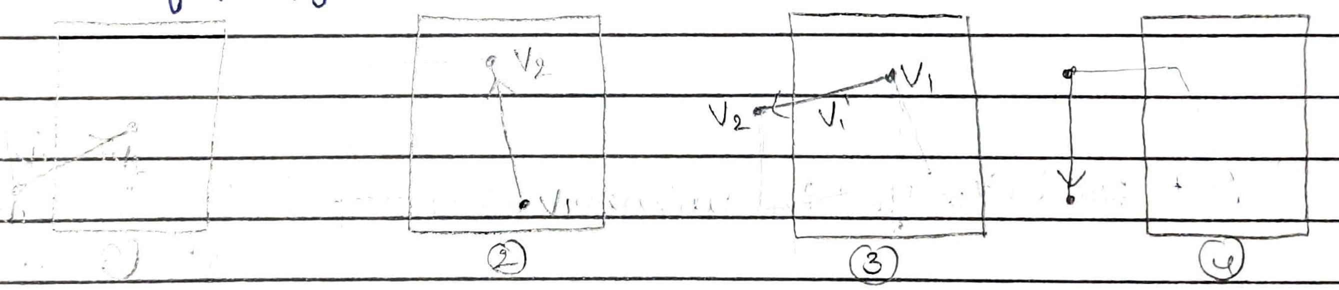
It clip the polygon on all 4 edges [Left, Right, bottom, top] by clipping the entire polygons against one edge, then taking the resulting polygon and clipping against 2nd edge and so on for all 4 edges.



Original polygon clip left clip Right clip Bottom clip top

There are 4 possible cases when processing vertices in sequence.

The 4 cases are illustrated in figure for successive pair of polygon vertices.



①	②	③	④
out → in	in → in	in → out	out → out
Save: V1', V2	Save V2	Save V1'	Save: none

Case 1: If first vertex is outside the window, second vertex is inside, both intrects point of polygon edge and Second vertex is saved.

Save: V1', V2

Teacher's Signature : ____________

<u>case 2</u>: If both vertices are inside the window boundary, only second vertex is added (saved). Save : V2.

<u>case 3</u>: If the first vertex is inside, second vertex is outside only the edge intersection with the window boundary is added (saved). Save : V_1'

<u>case 4</u>: If both input vertices are outside the window boundary, nothing is saved. Save: None.

31) (a) Explain window to viewport transformation carried out.

The main objective of the window to viewport mapping is to convert the world co-ordinate (x_w, y_w) of any point to ~~this~~ its normalized device co-ordinates (x_v, y_v).

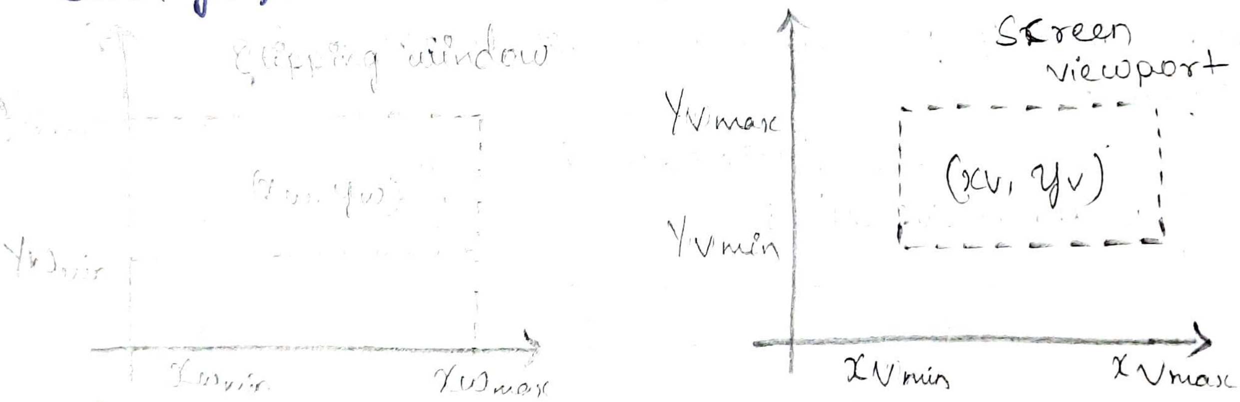


Fig: A point (x_w, y_w) in window is mapped to viewport at (x_v, y_v)

Both window & viewport is described by 4 co-ordinates

A window is specified X_{wmin}, X_{wmax}

A viewport is specified X_{vmin}, X_{vmax}

$$\boxed{\begin{aligned} X_v &= X_{vmin} + (X_w - X_{wmin}) * S_x \\ Y_v &= Y_{vmin} + (Y_w - Y_{wmin}) * S_y \end{aligned}}$$

where
$$S_x = \frac{(x_{v_{max}} - x_{v_{min}})}{(x_{w_{max}} - x_{w_{min}})}$$

$$S_y = \frac{(y_{v_{max}} - y_{v_{min}})}{(y_{w_{max}} - y_{w_{max}})}$$

(b)

32) (a) Write a note on projections.

The technique for achieving realism in three dimensional graphics is projection.

Projection can be defined "as a mapping of a point P(x, y, z) into its image P'(x', y', z') in the view plane".

2 types of Projection

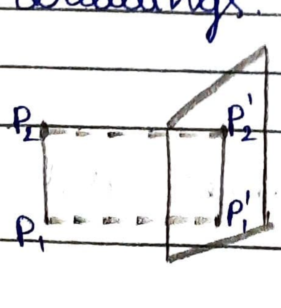
- Parallel
- Perspective

Parallel projection: Parallel projection involved generating a view of a solid object by projecting points on the object surface along parallel lines onto the display plane.

Parallel lines in the world coordinate scene, project parallel lines on the 2D display plane. It preserves relative proportions of objects.

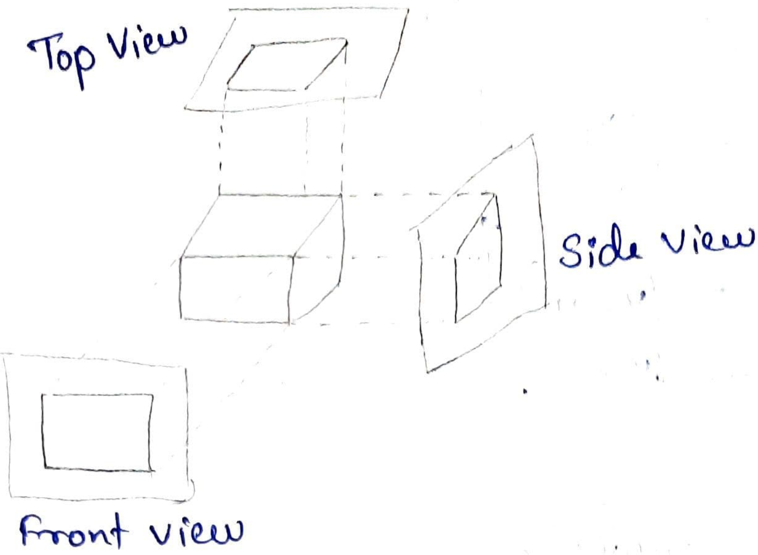
It provides accurate views of the various sides of an object but not realistic.

It is used in engineering and architectural drawings.



Parallel projection of an object to the view plane.

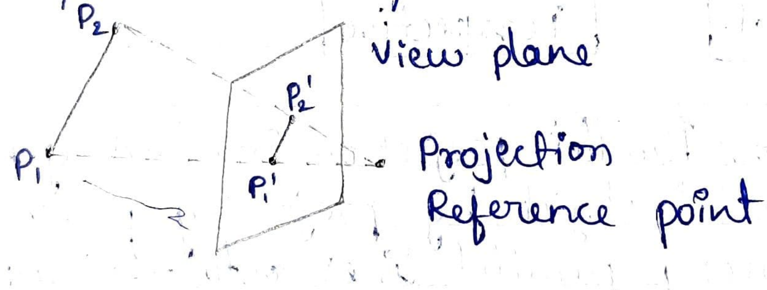
Teacher's Signature : ____________



2) Prespective Projection: In perspective projection, object positions are transformed to the view plane along lines are converge to a point called projection reference point.

Here the objects for from the viewing position to be displayed smaller than the original objects. When the object is closer the size is the same.

Eg: Aeroplane and Ship



Here parallel lines in a scene that are not parallel to the display plane.

Here the objects appears more realistic. This is the way that our eyes and camera lens form the images. They all converge at a single point called the center of projection and the intersection of these converging lines.

(b) Explain Octree.

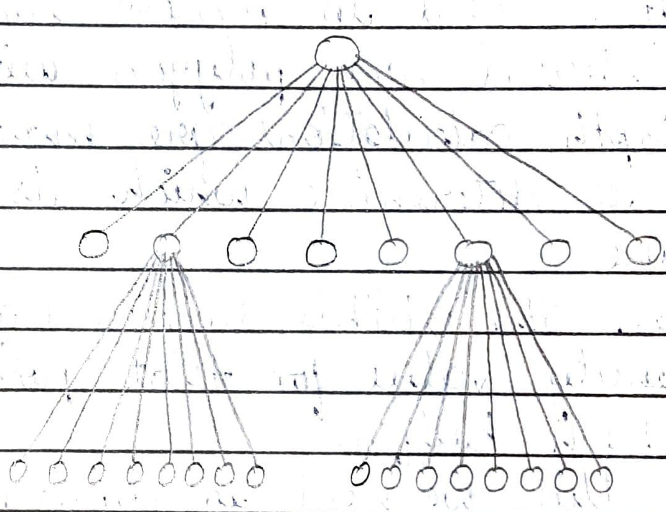
Octrees are hierarchical tree structures used to represent solid objects in graphics system.

An octree is a tree data structure in which each internal node has exactly eight children.

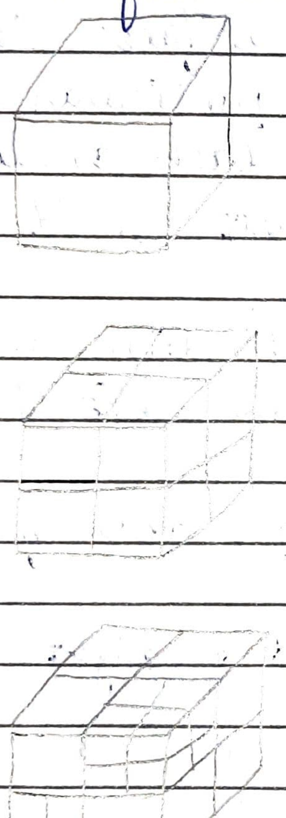
It is used to partition the 3-dimensional space by recursively subdividing it into 8 octants.

Octrees can be generated by using the procedure of quadtree.

It is used in Medical imaging and other applications for an object cross section view.



The corresponding octree



Recursive subdivision of a cube into octants.

33) Explain Z buffer algorithm for hidden surface removal.

Algorithm

Step 1: Initialize the depth buffer and refresh buffer

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

Step 2: For each position on each polygon surface, compare depth values to previously stored value in the depth buffer.

(a) Calculate the depth z for each (x,y) position on the polygon.

(b) If z > depth (x,y) then set

depth (x,y) = z refresh (x,y) = I surf (x,y)

(c) If z < depth (x,y) This polygon is closer to the observer than others already recorded for this pixel.

34) (a) Explain Scan-line method.

Scan-Line method:

It is used to remove hidden surfaces. Here it index intersect each polygon with a particular scan line. Depth calculations are made for each overlapping surface to determine which is nearest to the viewplane.

When the visible surface has been determined, the intensity value for that position is entered into the refresh buffer.

For this we need to construct edge table for each scan line.

Scan lines are processed from left to right.

(b) Write a program to animate man walk with umbrella.

⇒ Refere CG record.