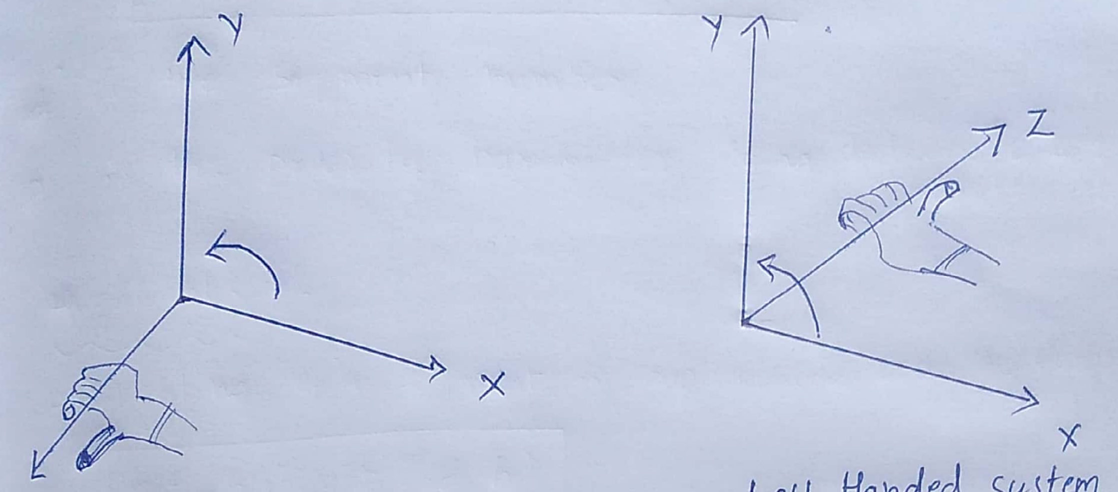
THREE DIMENSIONAL GRAPHICS

4th UNIT

3D graphics programs allow objects to be created on an X-Y-Z scale [width, height, depth].



Right Handed system

Left Handed system.

- If the thumb of right hand point in the positive Z-direction, then the co-ordinates are called <u>Right Handed System</u>.
- If the thumb of left hand points in the positive Z-direction, then the co-ordinates are called <u>Left handed system</u>.

<u>Projection</u>: The technique for achieving realism in three dimensional graphics is <u>projection</u>.

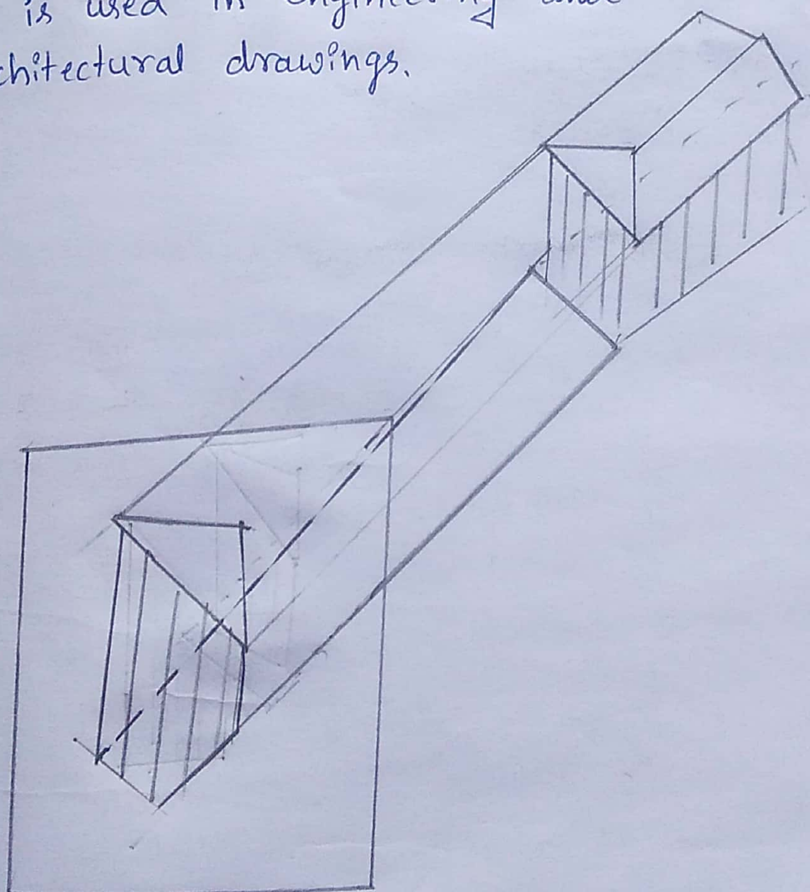
Projection can be defined "as a mapping of a point $P(x, y, z)$ onto its image $P'(x', y', z')$ in the view plane".

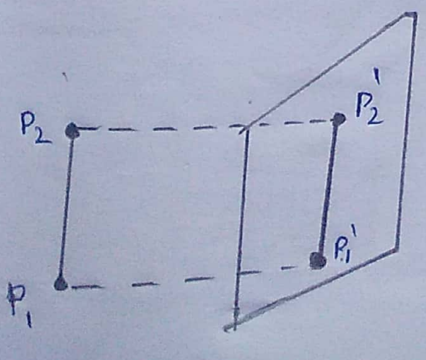
2 types of projection

- Parallel
- perspective

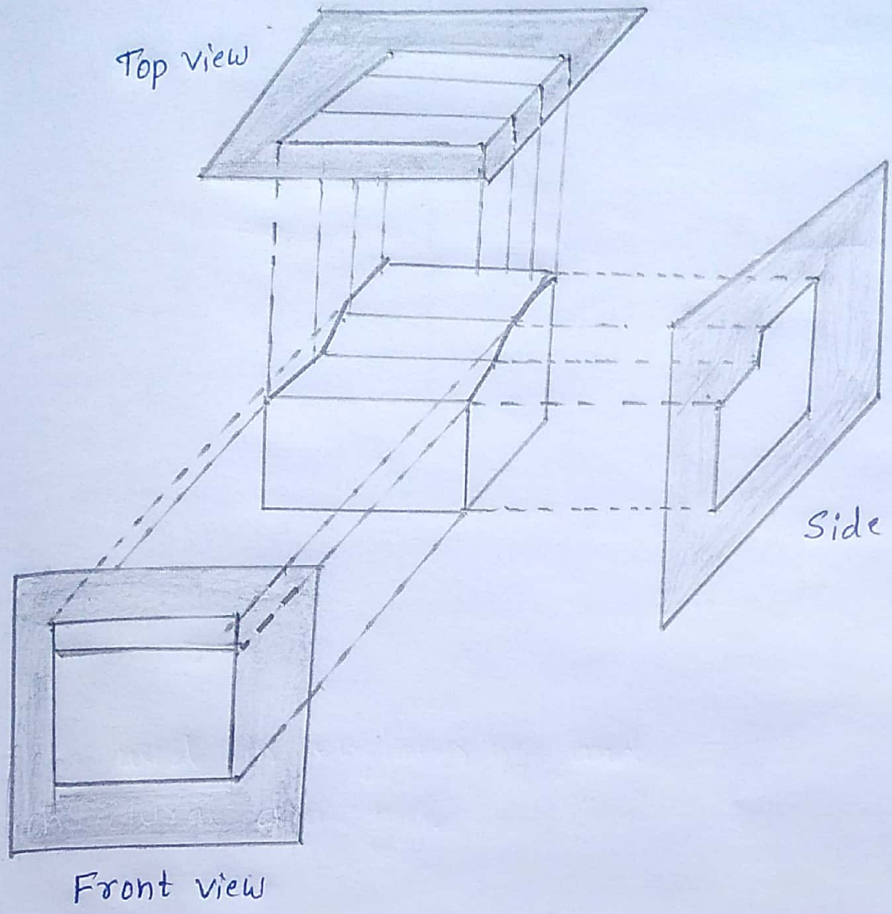
1. Parallel projection : parallel projection involved generating a view of a solid object by projecting points on the object surface along parallel lines onto the display plane.

- 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.





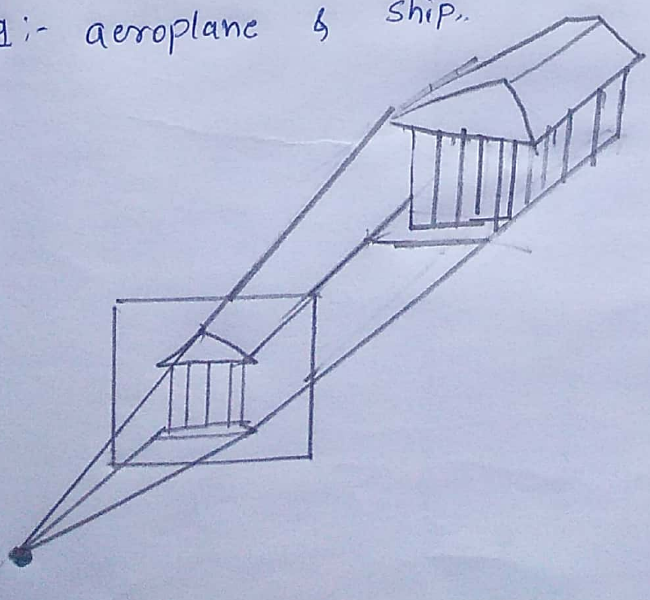
parallel projection of an object to the view plane.

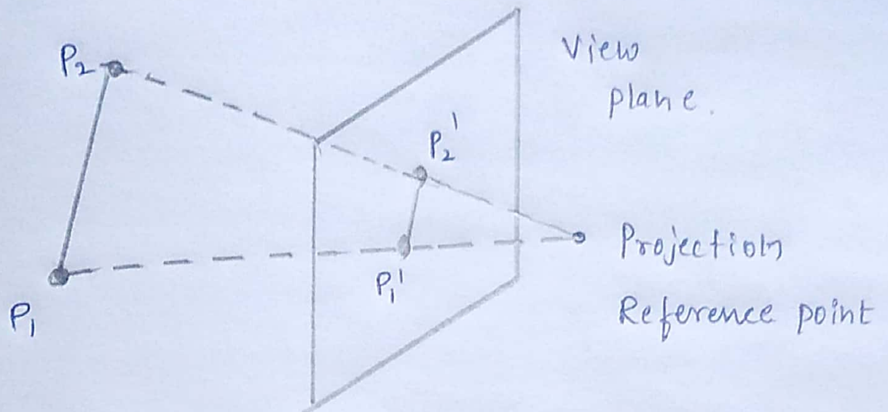


2. Perspective Projection: In perspective projection, object positions are transformed to the view plane along lines are converge to a point called projection reference point.

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

Exg :- aeroplane & ship.

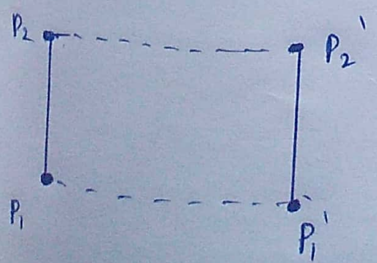
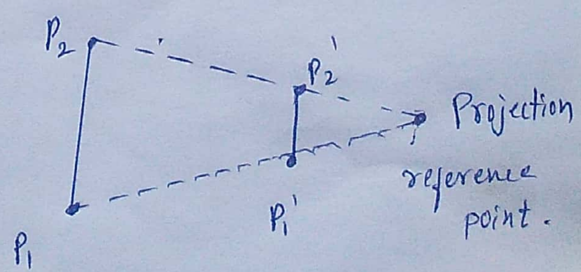




Here parallel lines in a scene that are not parallel to the display plane.

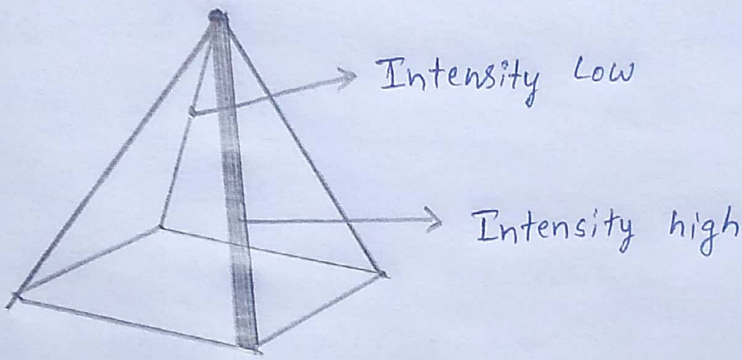
Here the objects appears more realistic. This is the way that our eyes and camera lens form the images. They all converge at a single point called the center of projection and the intersection of these converging lines.

Difference between parallel & ~~Projection~~ Perspective projection.

Parallel projection	Perspective Projection
• The projection are parallel to each other	• The projection intersect at the centre of position
• Less realistic in parallel.	• Visual effect is similar to projection.
• Preserves the parallel Lines.	• Do not preserves the parallel Lines.
• The center of projection is at Infinity	• The center of projection is a finite point.
	

Intensity Cueing (or) Depth 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 viewing position".

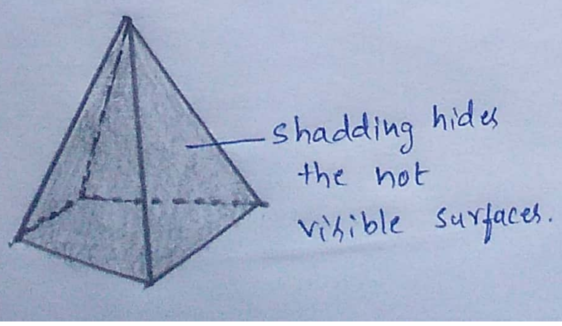


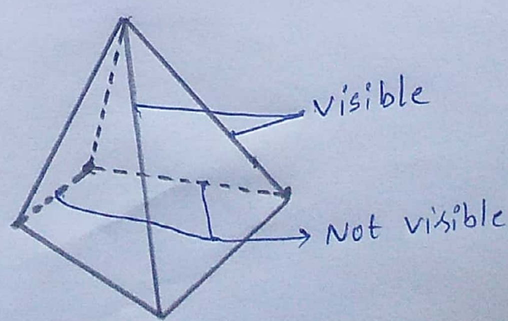
Intensity is to vary the intensity of lines according to their distance from the viewing position.

From the Above pic shows an object with front lines highlighted and drawn wider than back lines. The range of intensity brightness gradually decreases for lines far from the viewer.

Hidden Surface Removal :

When images are viewed as surfaces filled with color (or) shaded pattern, Hidden surface removal technique is used to remove surfaces hidden by visible surface.





The visible lines are highlighted & display them in a different color.

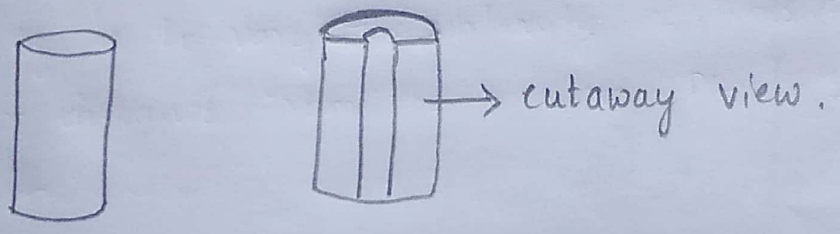
Non-visible lines are dashed ~~leg~~ lines.

---

* Exploded and Cutaway Views :

Exploded and cutaway views of object is used to show the internal structure and relationship of the object parts.

It removes the part of the visible surfaces to show internal structure.

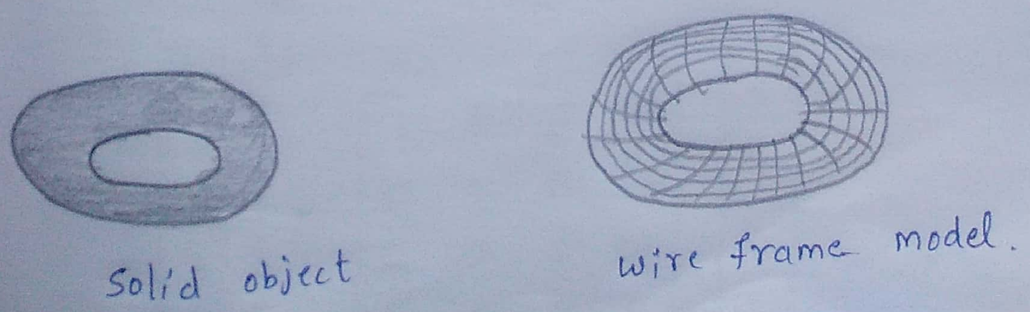


---

* Wire frame model [Polygonal mesh]

" It is a visual presentation of a 3D or physical object used in 3D computer graphics".

It represents the shape of a solid object with its characteristic lines and points. It is used to difine complex solid objects.



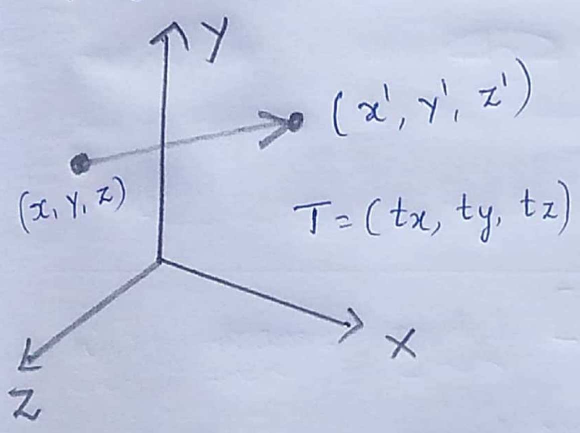
Imp

3D Transformations :

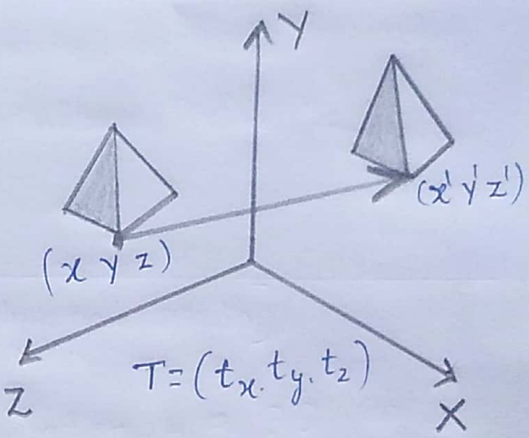
3D transformation are extended from 2D method by including Z- co-ordinate.

1. Translation.
2. Rotation.
3. Scaling.

1. <u>Translation</u> : In a 3D co-ordinate representation a point P (x, y, z) is translated to position p' (x', y', z') with the matrix operation.



Translation of <u>point 'P' to p'</u>



Translating an <u>object</u>

The translation vectors are $T(t_x, t_y, t_z)$

The matrix represention equations are

$$x' = x + t_x$$
$$y' = y + t_y$$
$$z' = z + t_z.$$

$$\boxed{p' = T \cdot P}$$

~~Using~~

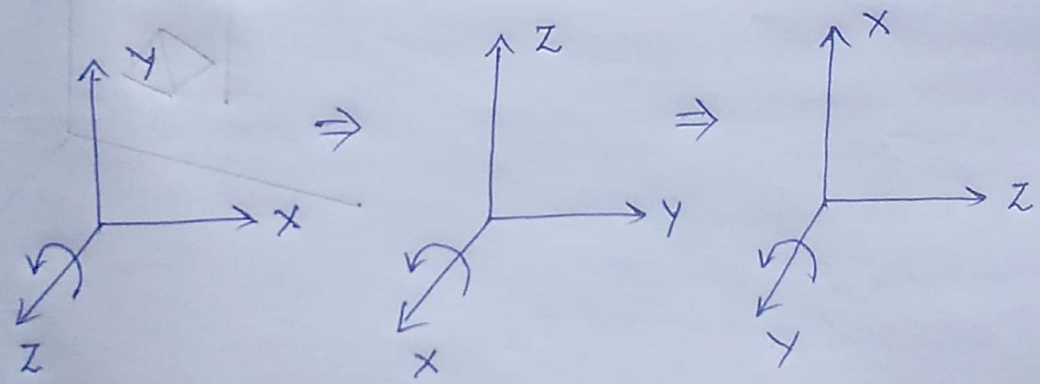
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scanned with CamScanner

2. Rotation : To rotate an object in 3D transformations we must decide 2 things
- An axis of rotation
- The angle θ to be rotated

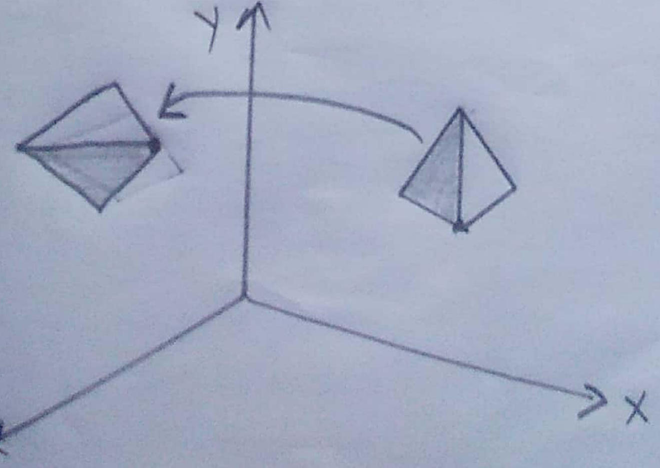
" Rotation is the ability to repositioning an image along a circular path in the x, y, z plane, by specifying the angle of rotation θ which the object is to be rotated".

Positive rotation angles produce counter clockwise rotations 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

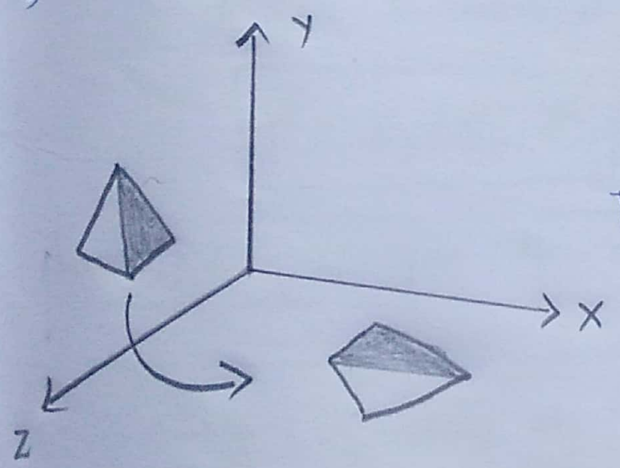
$$\boxed{\begin{aligned} y' &= y\cos\theta - z\sin\theta \\ z' &= y\sin\theta + z\cos\theta \\ x' &= x \end{aligned}}$$

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

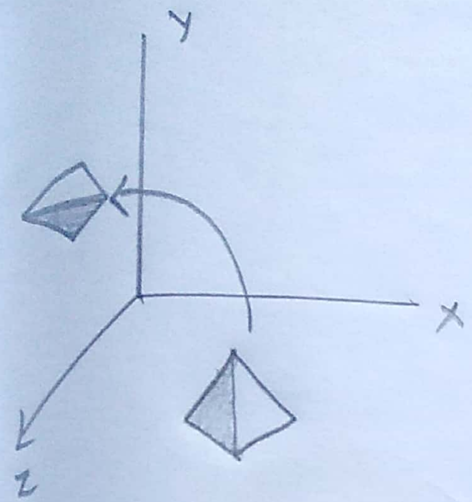
$$\boxed{\begin{aligned} z' &= z\cos\theta - x\sin\theta \\ x' &= z\sin\theta + x\cos\theta \\ y' &= y \end{aligned}}$$

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



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

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

$$\boxed{P' = R_2(\theta) \cdot P}$$

NOTE 'θ' specifies the rotation angle

3] Scaling : "Scaling an object with transformation changes the size of the object and repositions the object".

In a 3D co-ordinate representation, an object is scaled from the position $P(x, y, z)$ to the position $P'(x' y' z')$.

The equations are

$x' = x \cdot S_x$

$y' = y \cdot S_y$

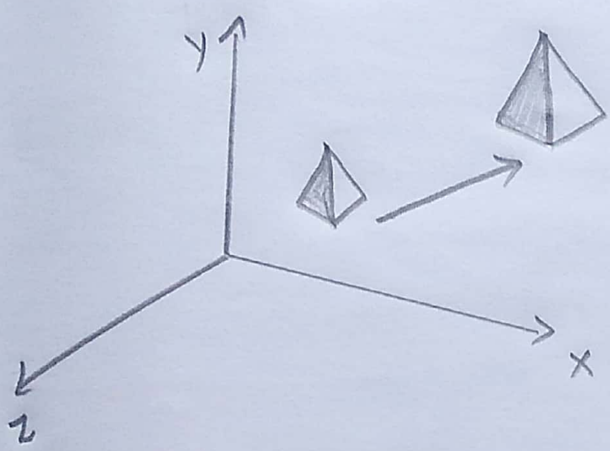
$z' = z \cdot S_z$

where S_x, S_y, S_z are scaling vectors.

The matrix form is

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\boxed{P' = S \cdot P}$$



Polygon Tables :

The graphics package organizes the polygon surface data into tables.

The table may contain geometric, topological and attribute properties.

polygon data tables can be organized into 2 groups.

→ Geometric tables

→ Attribute tables

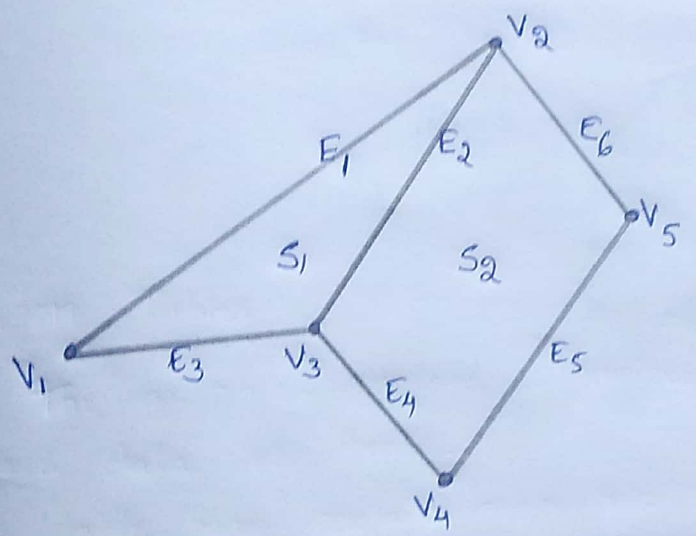
- **Geometric data tables :** It contain vertex co-ordinates and parameters to identify the orientation of the polygon surfaces.

To store geometric data 3 tables 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

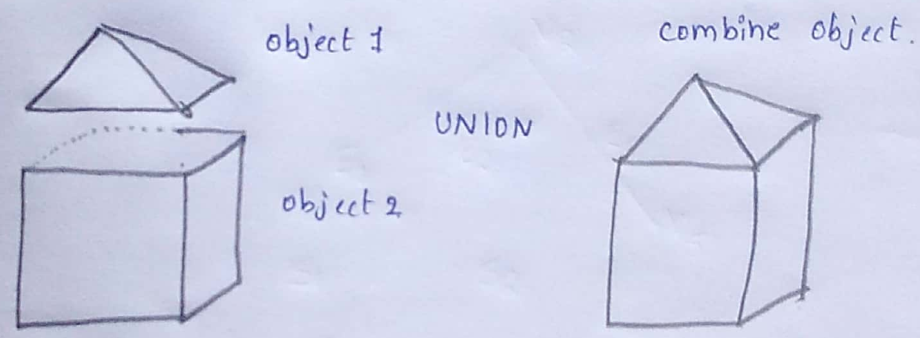
* Constructive - Solid Geometry Method (CSG)

CSG is a ~~bb~~seful technique for solid modelling is to combine the volumes occupied by overlapping 3D objects using set of operations.

A new object is created by applying the UNION INTERSECTION, and DIFFERENCE operation to 2 specified objects.

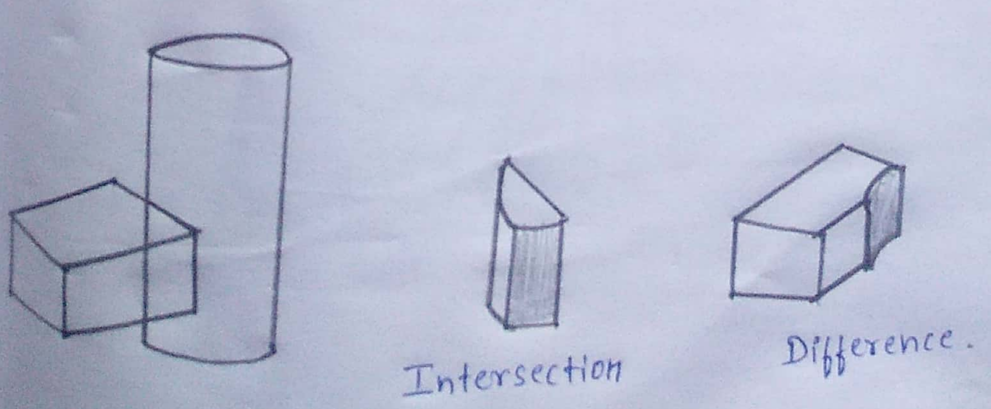
1. UNION : It is a process of combine 2 objects and create a new object.

Exa :-



2. Intersection : It overalapps the 2 object using intersection operation.

3. Difference ; The solid is obtained with a difference operation by substracting overlapping object.



IMP

* Octtrees :

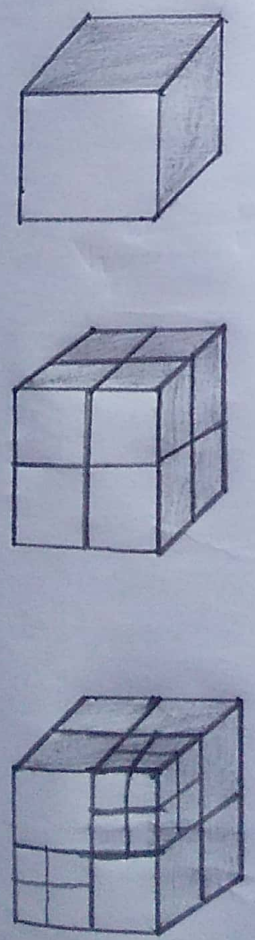
Octrees are hierarchical tree structures used to represent solid objects in graphics system.

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

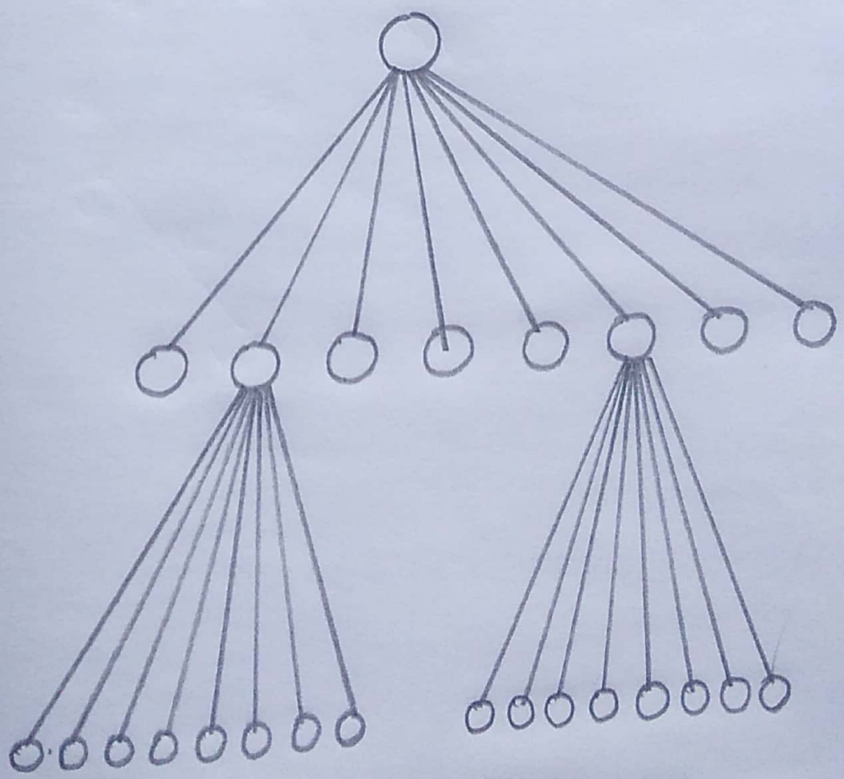
It is used to partition the 3-dimensional space by recursively subdividing it into 8 octants.

octtrees can be generated by using the procedure of quad tree ;

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



Recursive subdivision of a cube into octants



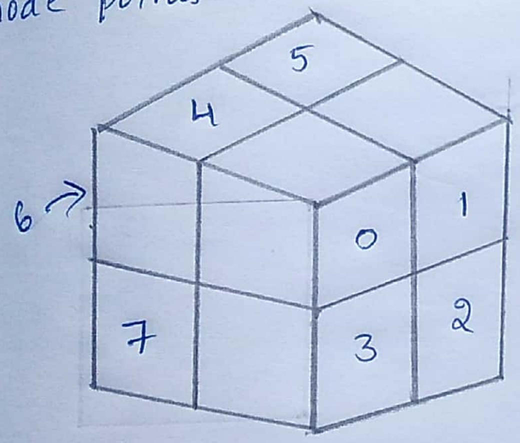
The corresponding octree.

Octree Encoding

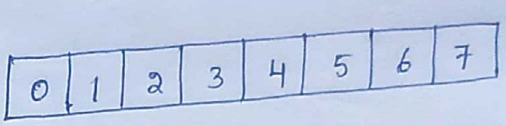
An octree encoding scheme divides regions of 3 dimensional space (cubes) into octants and stores eight data elements in each node of the tree.

Each elements of a 3dimensional space are called volume elements or Voxels.

Any heterogeneous octant is subdivided into octants and the corresponding data element in the node points to the next node in the octtree.



Region of a 3D space



Data Elements in the Representative octree node.

Algorithm for generating octrees

step 1: accept object definitions

step 2: Use the minimum & maximum positive values

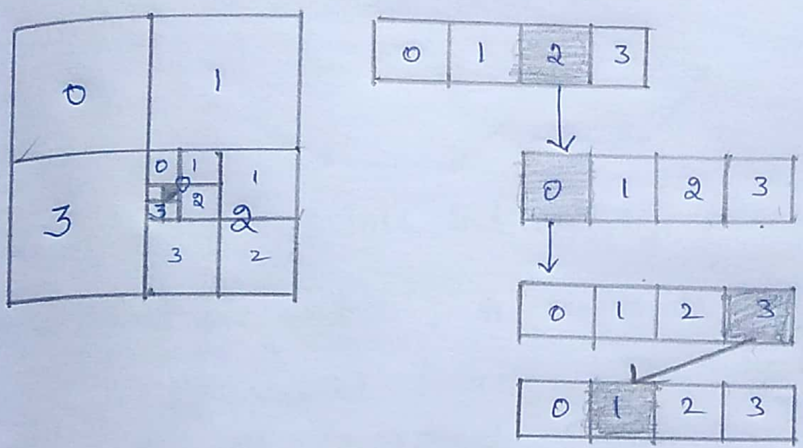
step 3: 3D object is tested, octant by octant to generate octree.

step 4: Set operations Union, intersection & Difference can be applied to octree.

step 5: A new, selected or left out objects are formed by union, intersection, difference respectively.

Quadtree : Quadtree is a hierarchical tree data structure in which each internal node has exactly Four children.

It recursively subdividing it into Four quadrants.



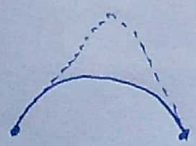
Quadtrees are dividing 2D region into quadrants.

→ If all pixels within a quadrant have the same color (homogeneous quadrant) the corresponding data element in the node stores that color.

→ other the quadrant is said to be heterogeneous & itself divided into quadrants untill it become homogeneous.

---

* Curves : " A curve is an infinitely large set of points. Each point has 2 neighbours except endpoint", OR "It is an arc between 2 endpoints".



Smooth curves and surfaces must be generated in many computer graphics applications.

Imp

Properties for designing curves:

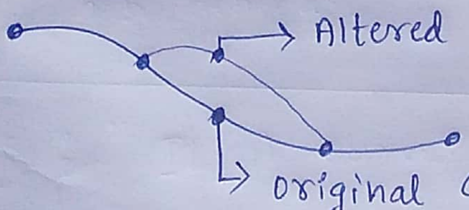
1. **Control point**; Control point or Knots are used to control the shape of a curve. The curve must pass through the point that control the curve's shape in a predictable way.

control points indicate by points.

2. **Multiple value**; A curve is not a graph of a single valued function. It may be multivalued with all co-ordinate system.

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

4. **Global or Local control**: A designer can manipulates the control. That may change shape only near the control point or change the shape.



5. **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.

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

The order of the curve determines the minimum number of control points necessary to define the curve.

The order of the curve also affects how the curve behaves when a control point is moved.

a) No Continuity : The curves do not meet at all.

b) Zero order continuity : It means the 2 curves meet.

c) First order continuity ; It requires the curve to be tangent at the point of intersection.

d). Second order continuity ; It requires that curvature must be the same.

Imp

Bezier curves : It is developed by French engineer "Pierre Bezier" in 1962.

It is used to design automobile body design of the Renault car.

" Bezier defines the curve $P(u)$ in terms of the location of $n+1$ control points P_i

$$\boxed{P(u) = \sum_{i=0}^{n} P_i \, B_{i,n}(u)}$$

where $B_{i,n}(u)$ is a blending function

$$B_{i,n}(u) = C(n,i)\, u^i (1-u)^{n-i}$$

$c(n,i)$ is the binomial co-efficient

$$c(n,i) = \frac{n!}{i!\,(n-i)!}$$

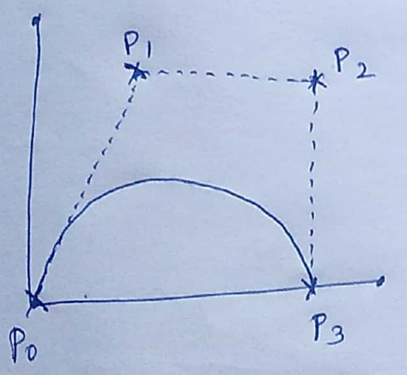
The equation for the x, y, & z parametric function

$$x(u) = \sum_{i=0}^{n} x_i B_{i,n}(u)$$

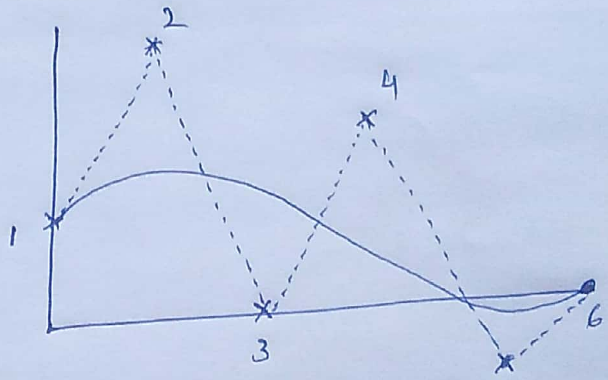
$$y(u) = \sum_{i=0}^{n} y_i B_{i,n}(u)$$

$$z(u) = \sum_{i=0}^{n} z_i B_{i,n}(u)$$

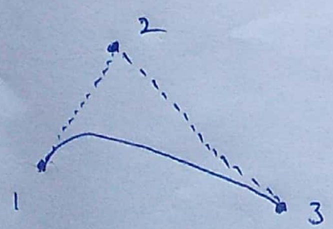
Bezier curve with different control points

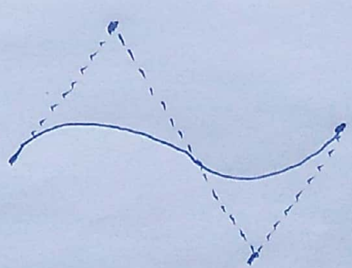


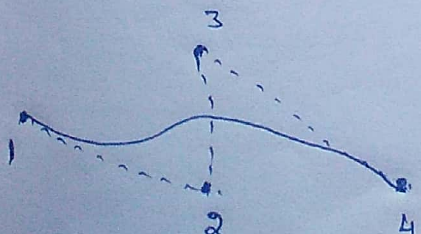
4 control point



6 control point.







Properties of Bezier Curves :

- A Bezier curve always passes through the first and last control point.
- It lies within the convex polygon boundary of the control points.
- If Bezier blending function all are positive and their sum is always 1.

  i.e $\sum_{i=0}^{n} B_{i,n}(u) = 1$.

- The degree of the polygonomial defining the curve segment is one less that the number of defining polygon point.

  Exa : 4 Control points, the degree of the polynomial is 3.

---

* Hidden Surface removal

The process used to determine which surfaces and parts of surfaces are not visible from a certain viewpoint is called hidden surface detection or Visible surface detection.

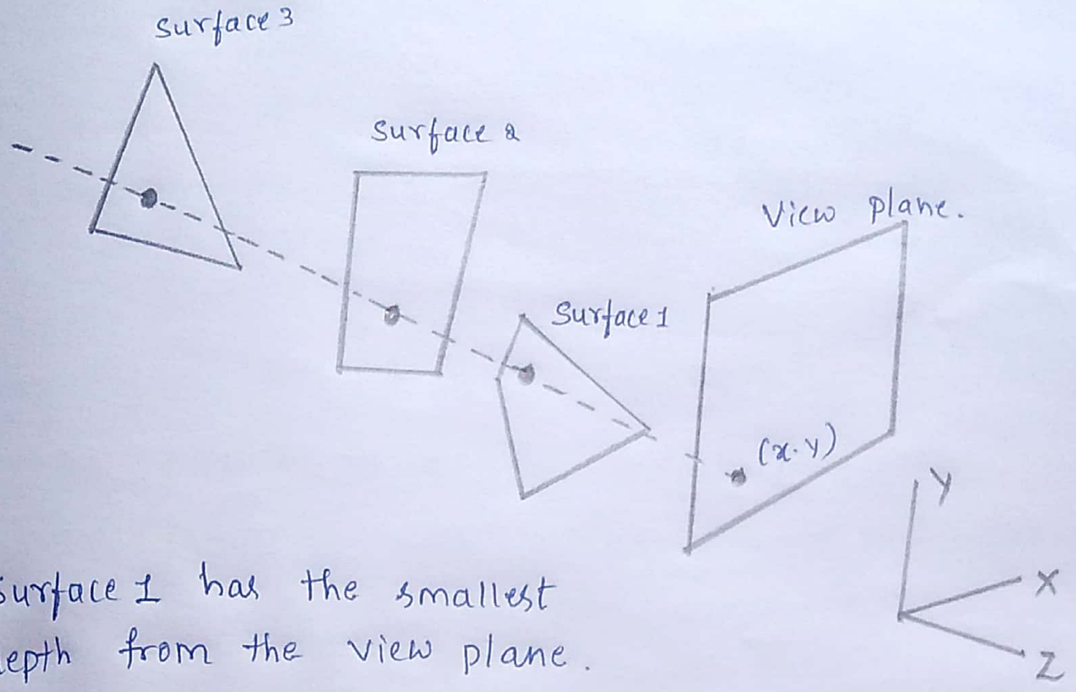
Imp ***

Depth Buffer method :

It is also called Z- buffer method, because object depth is measured using Z-axis.

Image space method is used to detect visible surface in depth buffer method, which compares surface depths at each pixel position on the plane.

Each surface of a scene is processed separately, one point at a time across the surface.



Surface 1 has the smallest depth from the view plane.

The z values range from 0 at the back clipping plane to Zmax. at the front clipping plane.

Two buffer areas are required,
one a depth buffer is used to store depth values
Second refresh buffer is used to stores the intensity values for each position.

Imp
Depth Buffer 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 coloser to the observer than others already recorded for this pixel.

V.IMP

Scan - Line method

It is used to remove hidden surfaces.

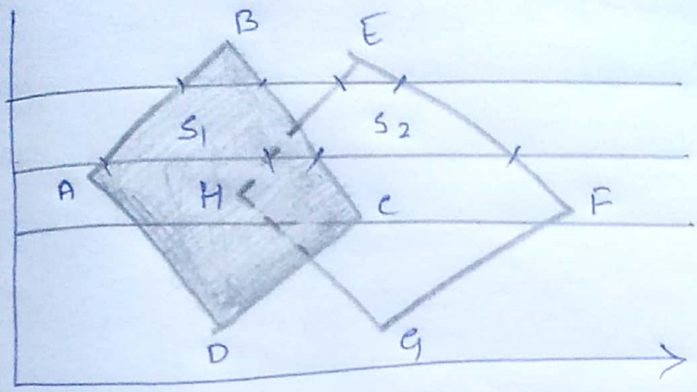
Here it intersect each polygon with a particular scan line. Depth calculations are made for each overlapping surface to determine which is nearest to the view plane.

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.

next page.



ABCD are the end points of Surface S_1

EFGD are the end points of Surface S_2

Dashed line indicates hidden surface.

1. Scan line 1.

The active edge table for scan line 1.

edge	surface
AB	S_1
BC	S_1
EH	S_2
EF	S_2

Here AB & BC are in S_1
EH & EF are in S_2

Depth calculations are not necessary.
Surface S_1 is stored in refresh buffer.

2] Scan Line 2

The active edge table for scan line 2

edge	surface
AB	S_1
EH,	S_1, S_2
BC.	S_1, S_2
EF	S_1

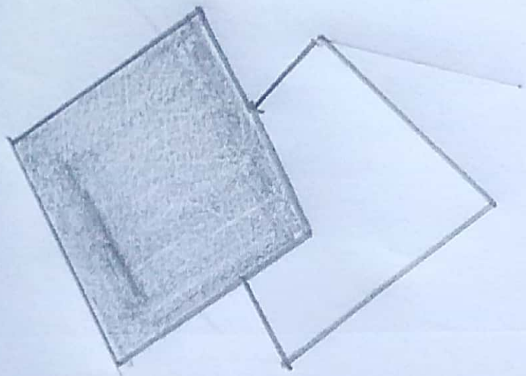
Here EH & BC placed on both surface S_1, & S_2.

Depth calculations are required.

depth surface S_1 is assumed less than surface S_2.
Intensity of surface S_1 are loaded into refresh buffer.

Repeat the same for Scanline 3.

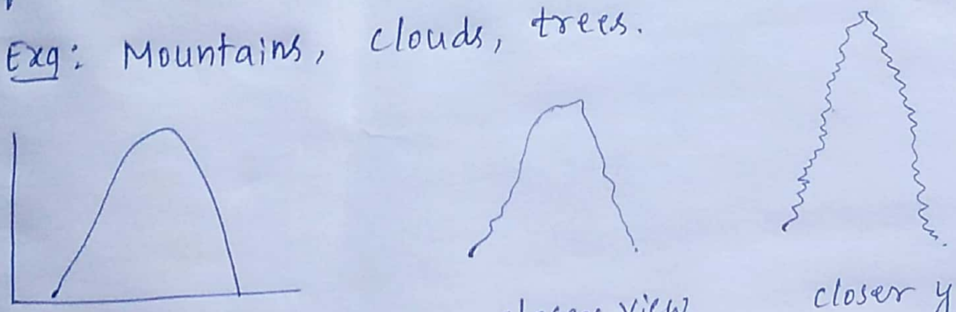
After depth calculations the the surface looks like.



Fractals :

Fractals geometry representations for the objects are applied in many fields to describe the features of natural phenomenan.

Exg: Mountains, clouds, trees.



Distant mountain   closer view   closer yet.

In computer graphics, we use fractal methods to generate images of natural objects and for viewing of various mathematical and physical system.

Applications of Fractals :

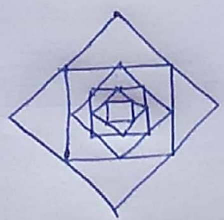
1. Fractal representation are used for modelling clouds, water, trees, plants etc.
2. Fractal patterns are also found in distribution of star, moon, music, traffic flow etc. can be used in graphics application.

Classification of Fractals

Fractals are classified into 3 categories

1. Self-Similar
2. Self-affine.
3. Invariant fractals.

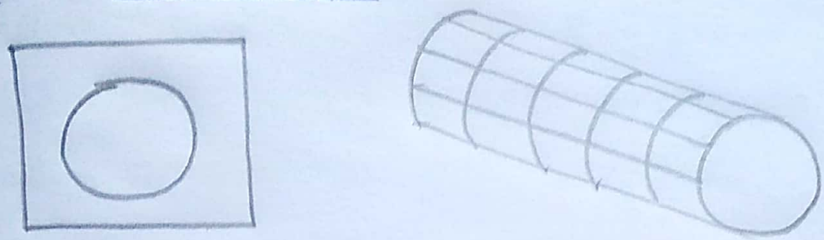
1. **Self-Similar :-** Starting with an initial shape, the object subparts are built by applying scaling parameters to the overall shape.
   - → If the same scaling factors are applied then it is known as Determinate self-similar fractal.
   - → Radom variations are applied to the scaled down subparts, then it is called statiscally self-similar.



2. **Self affine fractals**
   - → They are formed with different scaling parameters S_x, S_y, S_z in different co-ordinate directions.
   - → Random variations can be included to obtain Statistically self-affine fractals.
   - → Used to model terrain, water, clouds.

3. **Invariant fractals :** These fractals are formed with non-linear transformation. Include self-squaring & self-Inverse fractals.

* Sweep representation



Sweep representation are useful for constructing three-dimensional objects that passes translational, rotational or other symmetries.

Such object can be constructed by specifying a 2D shape and a sweep that moves the shape through a region of space.

Questions :

1. Define Right handed system & Left handed system.
2. what is projection? Explain its types?
3. Explain the diff b/w parallel & perspective projection.
4. What is Depth cueing? Explain
5. All types of transformation in 3D (Translation, rotation, scaling)
6. Explain 3D rotation in x, y, z axis.
7. Define polygon table? Explain with an example
8. Define octtree, quadtree, Explain
9. Explain the properties for designing curves
10. Explain Bezier curve & its properties.
11. Explain Depth buffer algorithm.
12. Explain Scan line method with Example.
13. Explain fractals & its types.

Segments

Imp

Definition: A segment is a set of output primitives that are joined for modifications purpose.

It is a logical unit and a collection of display file instructions representing graphics primitives that can be manipulated as a single unit.

---

* The Basic function for segmenting the file

The 3 basic functions are

Imp

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.

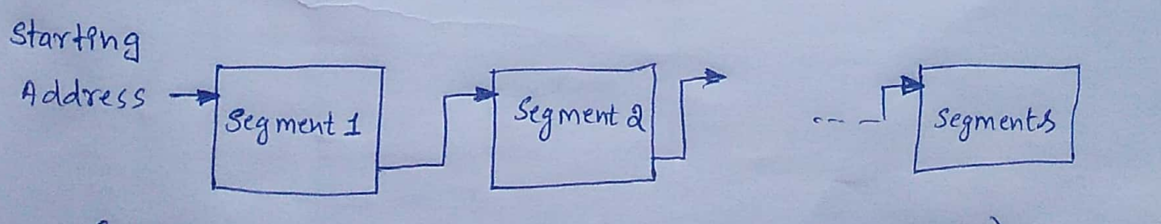
To create a new segment, we open it and then call graphics primitives to add to the segment the lines and text to be displayed. Then close the segment. To remove a segment from the display file, we delete it.

---

* Segment files:

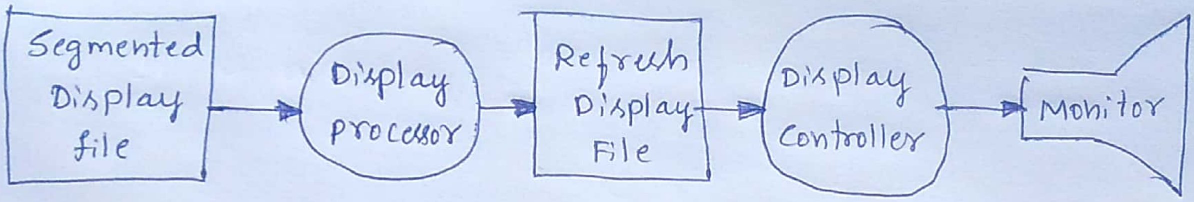
→ A segment file is any list of segments maintained by a graphics system.

→ Segment files often are stored as linked structures.



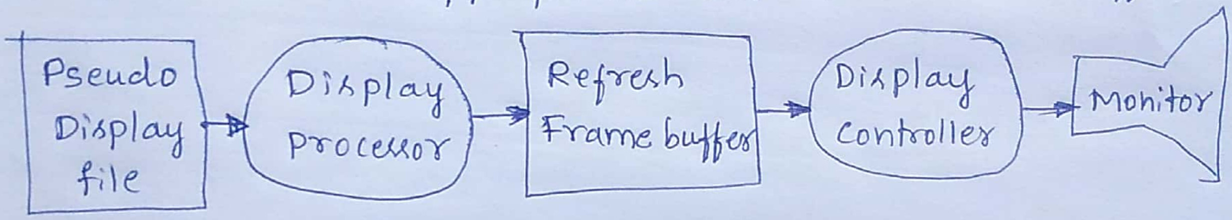
(Segment files stored as linked structures)

→ Several forms - a segmented display file is a display file program for a simple vector system.



(segment file organization for Random Scan display)

→ A pseudo display file contains segment definitions from which appropriate bits in the frame buffer



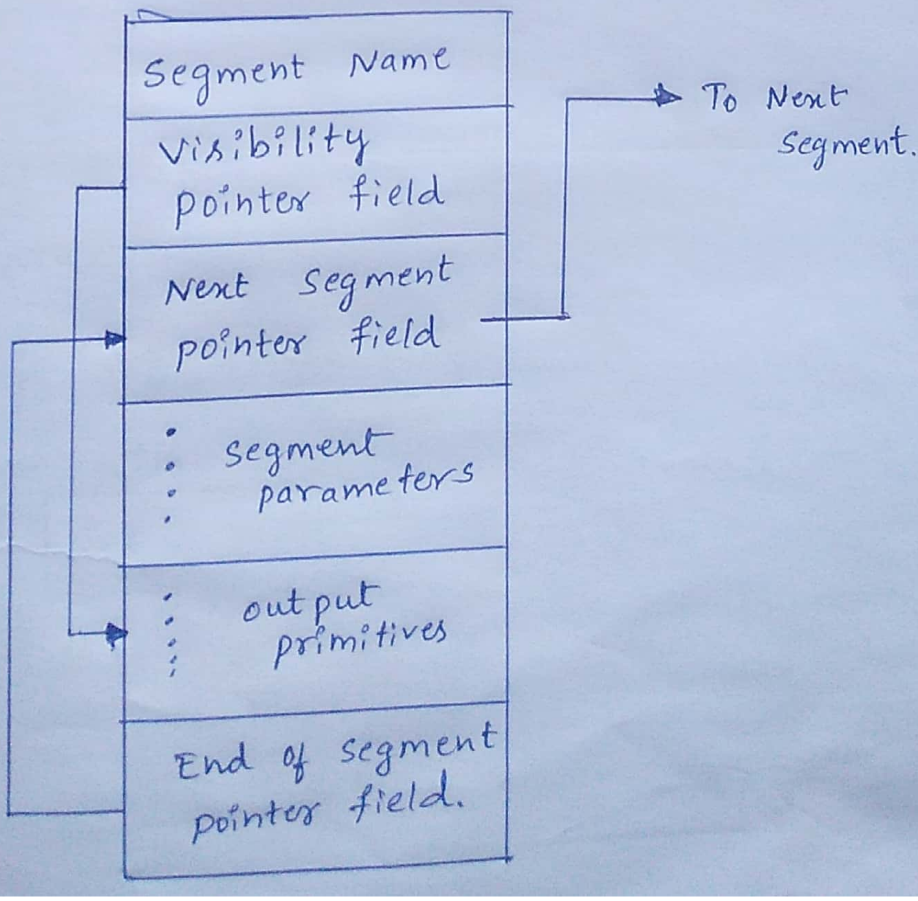
(Raster Scan Display)

---

* Memory Management for Segment Storage

Segment format

IMP



→ Blocks must be assigned as segments when segments are created.

→ Blocks must be returned to the storage pool as segments are deleted.

→ options

1. Fixed size blocks
   → Easy to manage
   → Lead to fragmentation

2. Variable sized blocks
   → Avoid fragmentation
   → more memory management.

→ make changes only at the end of the refresh cycle.

** Segment Attributes
*Imp.

To create a segment the following attributes are required

1. Visibility
2. Priority
3. Highlights
4. Transformation.

1. Visibility:- visibility of a segment can be set by the user with function

$$\boxed{\text{Set\_visibility}(id, v)}$$

where '<u>v</u>' is visible (posted) or invisible (unposted) which controls the display of segments and 'id' the segment number.

2. Priority :-

Segment priority can be set with the function

$\boxed{\text{set\_Segment\_priority(id, p)}}$

'id' is the Segment number

'p' is the priority parameter lying between '0' & '1'

3. Highlights :-

Highlighting of a segment is implemented with the function

$\boxed{\text{set\_highlight (id, h)}}$

'id' is the segment number

'h' is highlighted or normal, It depends on the type of display device.

4. Transformation :-

To achieve segment transformation, the user can specify the transformation (size, position, orientation) matrix which needs to be applied to the segments.

Transformation of a segment is implemented using

$\boxed{\text{set\_segment\_transformation(id, matrix)}}$

matrix specifies the elements of the transformation matrix.

Segment states :- The Segment states can be painted or unpainted.

14 Define Segment ? (1)

15 Explain the basic function of Segment (3)

16 write the Segment format for Segment Storage. (5)

17 Explain the Segment attributes (5)