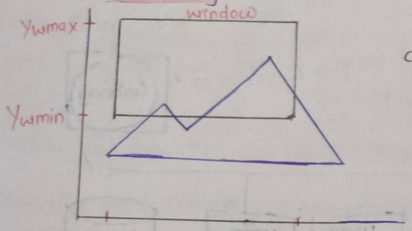
mputer graphics

UNIT III

Windowing : The process of selecting the part of real world sence to display it on some device called Windowing.

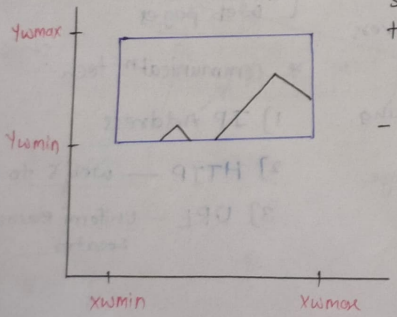
Portion of the graph that is to be displayed in world-co-ordinate. The method of selecting & enlarging a portion of a drawing is called windowing



- Any conventional cartision co-ordinate system reffered to as a World coordinate.

- Window : is Area chosen for this display is called window
- Viewport - An Area on display device to which a window is mapped.

- The picture parts within the selected area are then rapped to specific areas of the device co-ordinates.



- An area on a display device to which a window is mapped is called viewport

Device -coordinate

*viewing transformation :-

- VT is the process of transforming 2D world-coordinate object to device coordinates.
- Object inside the world or clipping window are mapped to the viewport, which is area on the screen where world co-ordinates are mapped to be displayed.

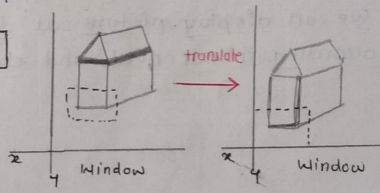
ViewPort :- Area on the device co-ordinate where graphics to be displayed.

- The process of mapping the part of world co-ordinate scene to device coordinate is reffered to as a viewing transformation or window to viewport transformation or windowing transformation

* Steps for window to view port transformat°

1) Translate window to origin !

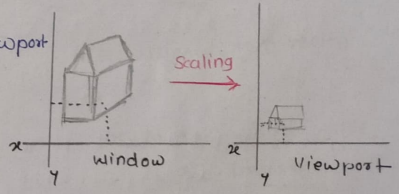
$$Tx = XWmin \quad Ty = -YWmin$$



2) Scaling of the window to match its size to the viewport

$$sx = \frac{(XVmax - XVmin)}{(XWmax - XWmin)}$$

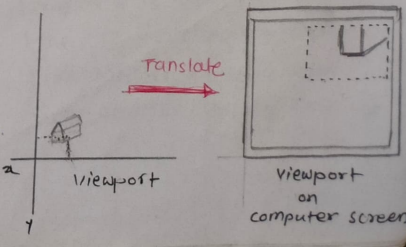
$$sy = \frac{(Yvmax - Yvmin)}{(YWmax - YWmin)}$$



3) Again translate viewport to its correct possition on screen.

$$Tx = -Xvmin \qquad Ty = -Yvmin$$

$$\therefore VT = T * S * T_1$$



$VT = T * S * T1$

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_{wmin} & -y_{wmin} & 1 \end{bmatrix}$$

Translate viewwindow to the origin.

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling of window to viewport size.

$$T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_{vmin} & y_{vmin} & 1 \end{bmatrix}$$

Translating viewport on Screen

* Advantages of Viewing Transformation.

- We can display picture at device or display system according to our need and choice.

* Line clipping
* Cohen Sutherland Line Clipping Algorithm

- It detected whether line lies inside the screen or it is outside the screen.
- All lines come under any one of the following categories.
  1) Visible
  2) Not visible
  3) clippling case
- Visible - line lies in window
- Not visible - Not lies in window
- Clippling case - neither visibler nor Invisible then its clippling case

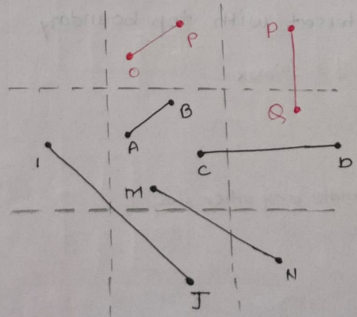
region 1	region 2	region 3
region 4	region 5	region 6
region 7	region 8	region 9

9 region.

1001	1000	1010
0001	0000	0010
0101	0100	0110

Xmin Xmax

bits assigned to 9 region.



Line AB is the visible case
Line OP is an invisible cas
Line PQ is an invisible lin
Line IJ is clipping case
Line MN is clipping case
Line CD is clipping case.

Advantages of Cohen — easy to understand, simple to implement

→ It calculates position of both end of the line

1) It calculates end-points very quickly and rejects and accepts lines quickly.

2) It can clip picture much large than screen size.

* Algorithm of cohen Sutherland Line Clipping :

Step 1 : Calculate positions of both endpoints of the line.

Step 2 : Perform OR operation on both of these end-pts.

Step 3 : If OR operation gives 0000 then line is visible
Else Perform AND operation on both endpts
If AND ≠ 0000 ⟶ line is invisible
Else AND = 0000 ⟶ line is consider as clippling case

step 4 : if a line is clipped case, find an intersection with boundries of the window. $m = (y_2 - y_1)(x_2 - x_1)$

(a) if bit 1 is "1" intersects with left boundary of ▭ [w]

$y_3 = y_1 + m(x - x_1)$, $x = x_{wmin}$

b) if bit 2 is "1" —11— right boundary

$y_3 = y_1 + m(x - x_1)$, $x = x_{wmax}$

c] if bit 3 is "1" line intersects with bottom boundary

$x_3 = \frac{x_1 + (y - y_1)}{m}$, $y = y_{wmin}$

d] if bit 4 is "1" line intersect with top boundary

$x_3 = \frac{x_1 + (y - y_1)}{m}$, $y = y_{wmax}$

Disadvantage

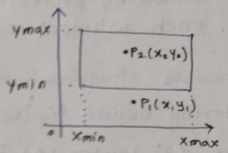
Repeated clipping is expensive

Only applicable to ▭ window, can't handle any other shap.

It can be improved using more region.

Point clipping

- Point clipping
- It is used to determining whether the point is inside the window or not
- 1) $x \leq x_{max}$
- 2) $x \geq x_{min}$
- 3) $y \leq y_{max}$
- 4) $y \geq y_{min}$

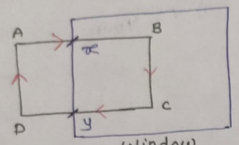


* Sutherland Hodgeman Polygon clipping Algorithm.

- The process of removing those part of a polygon that lie outside a clippling window
- It is performed by processing the boundary of polygon against each window corner or edge.

* Four Possible situations while processing

- by refering beside figure,
- If the 1st vertex (A) is outside, & 2nd vertex(B) is inside then 2nd vertex added to output list



Outside → inside → o/p inside vertex intersected pt to the window

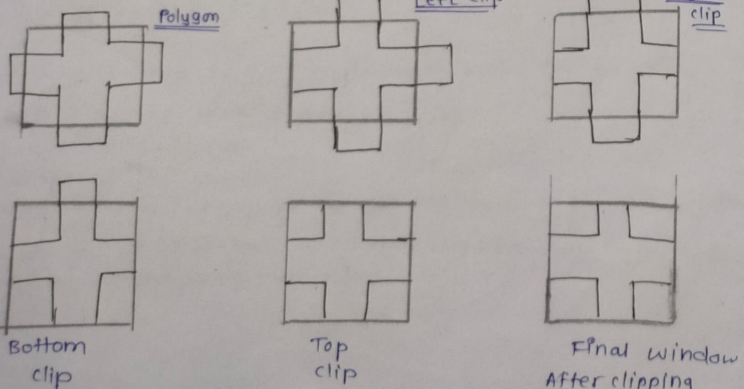
- Both vertex are inside A&C then 2nd vertex is o/p. (2)

Inside → inside → o/p inside vertex

- 1st vertex inside & 2nd outside the window (C & D) then the edge which intersected with window is added to o/p list (y)

inside → outside → o/p (y) intersection pt

- If both vertex are outside window the no o/p added (AD)



* Algorithm sutherland Polygon clipping

Step1: Read co-ordinates of all vertices of the polygon.

Step2: Read co-ordinates of all clipping window.

Step3: Consider the left edge of the window

Step4: Compare the vertices of each edge of the polygon individually with the clipping plane

Step5: Save the resulting intersection & vertices in the new list of vertices according to four passible relationship between the edge and the clipping boundary.

Step6: Repeat step 4 & 5 For remaining edges or the clipping window.

Step7: stop.

Advantages :-

- It is easy implement
- It clip polygon against all edges of clipping polygon
- Well suited for h/w implemntation.

Disadvantage :-

- Only works for convex clipping.
- time consumming.
- Sometime produce single polygon instead multiple. Polygon for concave polygon.

* Jai Mata di *
* Jai shree Krishna *

* 3D Transformation and Projection.

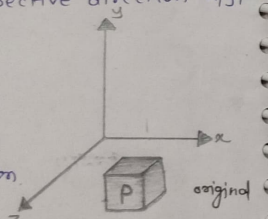
- 3D transformation is the process of manipulating the view of a 3-D object with repes respect to its original position by modifying its physical attributes through various method of transformation like,
  1) Translation.
  2) Scaling.
  3) Rotation.
  4) Shear.
  5) Reflection.

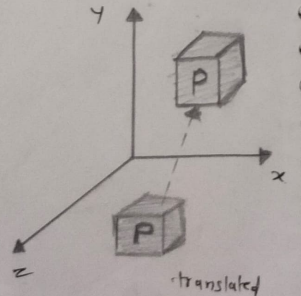
* Translation :-

- Translation allows to changes the position by adding some integers to each co-ordinates.
- Translation in 3D is similar to translation in 2D.
- In 3D there is z co-ordinate to specify the amount of motion in the direction of each of the co-ordinate axes.
- Tx, Ty, Tz translatn in the respective direction x, y, z.
- $x' = x + tx$, $y' = y + ty$, $z' = z + tz$ — $P(x,y,z)$, $P'(x',y',z')$
- Equivalent homogeneous matrix representation.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{old} \\ y_{old} \\ z_{old} \\ 1 \end{bmatrix}$$

$P' = T \cdot P$

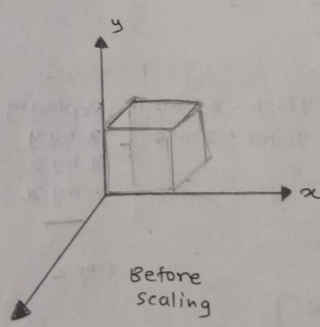


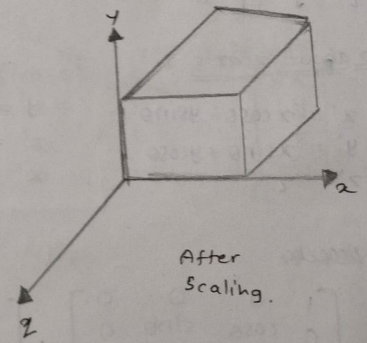


Scaling

- scaling is used to change the size of an object.
- The size of an object can be increased or decreased.
- if the value of sx & sy are less than 1 then object size is decrease.
- if the value of sx & sy are greater than 1 then object size is increases.
- Scaling Factor are sx, sy & sz
- let $S = [sx, sy, sz]$ be a vector of the scaling parameters in all 3 dimention.
  1) $P' = S \cdot P$
  2) $x' = sx \cdot x$
  3) $y' = sy \cdot y$
  4) $z' = sz \cdot z$
- Matrix representation.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$





* Rotation

- Rotation about x axis. } about axis
- Rotation about y axis }
- Rotation about principle axis
- Rotation about Arbitary line

- To define the direction of rotation in 3D, we use the right hand rule.

* Rotation about principle axis.

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

$P'(x', y' z')$
$P(x, y, z)$

* $P' = R_z \cdot P$

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

Rotation about x axis :-

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

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

} Replacing
x by y
y by z
z by x

Anticlockwise

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

For Anticlk wise

For clk wise rotation

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

* Rotation about y-axis

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

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

* Rotation about z-axis.

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

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

* Rotation About an Arbitrary line.

- In real-world rotation about the principle axis is a rare case
- Appl^{n} programmer needs to rotate the object around any arbitary axis in space.
- When the object is to be rotated around a line which is not parallel to the principle axis it requires more efforts.
- This case is more complicated than the previous cases.

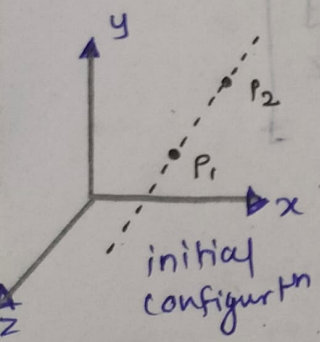
step1 : Translate the rotation axis such that it passes through the origin.

step2 : Rotate the translation axis such that is passes coincides one of the principle axes.

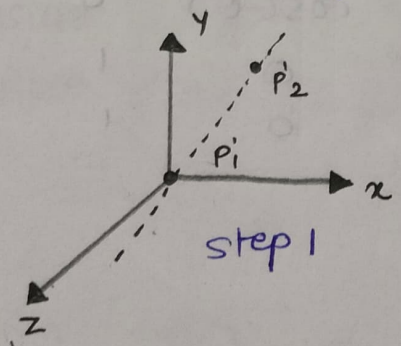
step3 : Perform the actual rotation ~~to bring rotation axis to the original orientation~~. operation

step4 : Perform inverse rotation to bring rotation axis to the original orienth

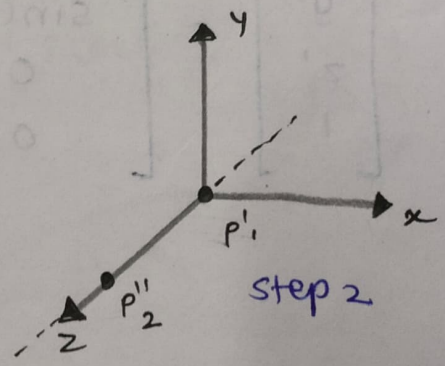
step5 : Perform inverse translation to move back the rotation axis to its original location.



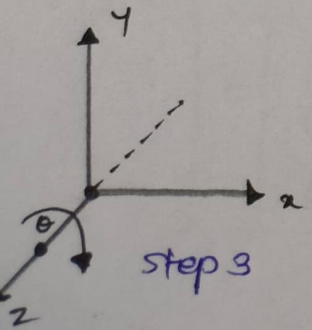
initial configurtn



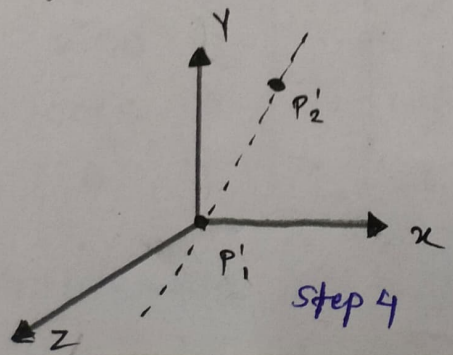
step 1



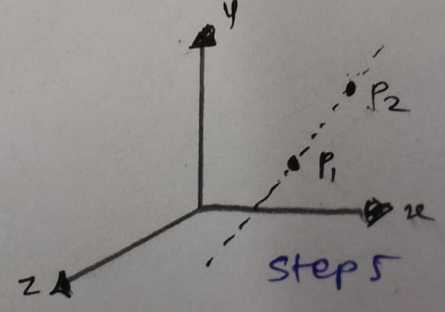
step 2



step 3



step 4



step 5

Reflection

Reflection about an axis is equivalent to rotation by 180° about that axis.

Transformation matrices for reflection about $x=0$, $y=0$, $z=0$

$$Ref(x=0) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Ref(y=0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Ref(z=0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

* $P' = M \cdot P = Ref(x=0) \cdot P$

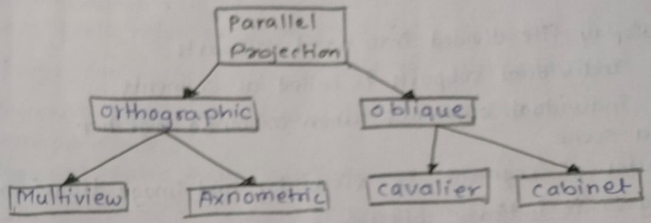
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$x' = -x$

$y' = y$

$z' = z.$

* Types of Parallel Projections.



UNIT 4

Segments

- The display file divided into no of subparts.
- These individual subparts is called as segments.
- These individual segments when combined together form a scene.
- Attributes of segment is visibility and image transformation
- Visibility :- to indicate whether a particular segment is visible or Not visible
- Image Transformation :- The different types of transformation like * translation , * Scaling , * rotation & *reflection can be applied on individual objects in the scene
- The different operations namely can be performed on segment namely creating a segment , opening/closing of a segment, deletion and renaming of a segment.

Advantages :-

- Segment visibility of each object can be controlled
- The individual objects in the display file is represented as segment.
- Animation can be applied on different segments.
- So that flow of animation can be controlled easily.

Disadvantages :-

- To maintain the segment table extra overhead is required.
- The ST works in association with display file interpreter
- sufficient no. of segment needs to be created for efficienty
- The content of the display file is not displayed by default

* Segment table

- All information related to segment is maintained in a table that table refer as segment table.
- Arrays can be hold used to hold the attributes of a segment
- Segment table and display file need to be work in correspondence.
- Segment table have attributes as follow :-

1) Segment Name :- used to uniquely identify a particular segment
   segment name is in integer number start with 1
   - Used to decied which segment wbwd be visible & which would not be visible.
2) Segment start :- start position of segment
3) Segment size :- No. of instruction required to draw a Particular object is nothing but the size of Particular segment.
4) Segment table :- structure

segment name	segment start	segment size	visibility	Scale x	Scale y	tran x
1	—	—	ON	—	—	—
2	—	—	OFF	-		-

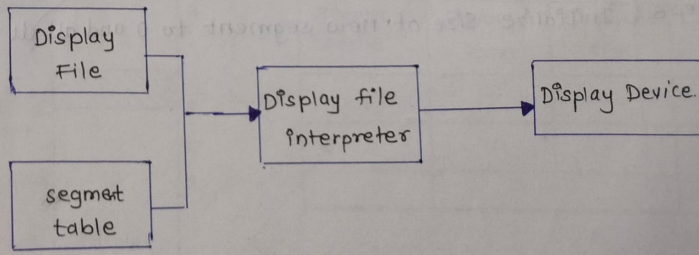


fig: Corresponding between display file & Segment file.

* Algorithm for Segment creation :-

step1 : If any segment is open, give error message : "Segment is still open" go to step 8.

step 2 : Read the name of the new segment.

step 3 : If the segment name is not valid, give error message "segment name not a valid name" and go to step 8

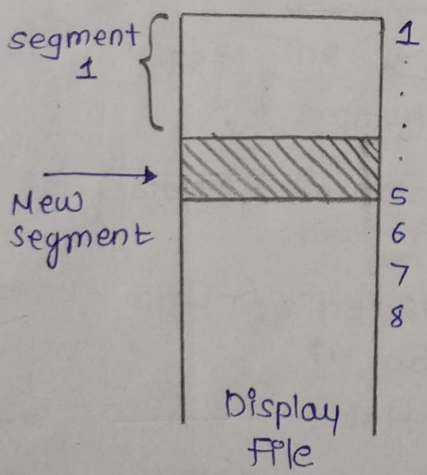
step 4 : If given segment name already exist, give error message : "segment name already exists in name list" and goto step 8

step 5 : Make next free storage area in display file as start of new segment.

Step 6 : Initialize size of new segment to 0 and all its attributes to their default values.

step 7 : Inform that the new segment is now open

step 8 : Stop.



Segment name	Segment start	Segment size	Visibility
1	—	—	—
2	14	0	ON

New segment.

fig 3 - Segment Creation

Closing a Segment :

After completing entry of all display file instructions, the segment needs to be closed.

Algorithm :

Step 1 : If any segments is not open, give error message : "No segment is open now" goto step 6

tep 2 : Change the name of currently opened segment to any unnamed segment, lets say 0

ep 3 : Delete any other unnamed segment instruction which may have been saved and initialize above unnamed segment with no instructions.

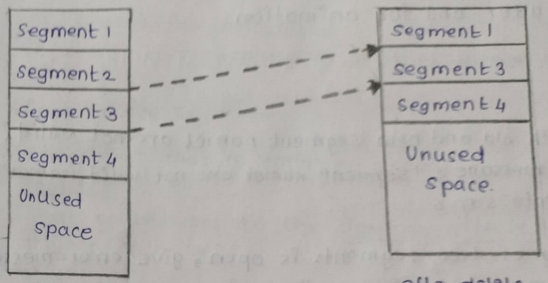
ep 4 : Make the next free storage area available in display file as start of the unnamed segment

p 5 : Initialise size of unnamed segment to 0

p 6 : Stop.

Deleting a Segment

- To delete a particular segment from display file, we must just delete that one segment



Before delete

After delete

* Algorithm

step 1 : Read the name of the segment to be deleted.

step 2 : If segment name is not valid, give error message "segment name is not a valid name" and goto step 8

step 3 : If the segment is open, give error message : "Can't delete an open segment" and goto step

step 4 : If size of segment is less than 0, no processing is required and goto step 8

step 5 : The segment which follow the deleted segment are shifted by its size.

step 6 : Recover deleted space by resetting index of next free instruction.

step 7 : The starting position of shifted segments is adjusted by subtracting the size of deleted segment from it

step 8 : Stop.

] Renaming a Segment :-

- This is done to achieve Double Buffering ie. idea of storing two images, one to show & other to create, alter and for animation.

* Algorithm

step 1 : If both old and new segment names are not valid, error message : "segment names are not valid names" and goto step 6

step 2 : If any of two segments is open, give error message "Segment are still open" and go to step 6.

step 3 : If new segment name given already exists in the display list, give error message : "Segment name already exists" and goto step 6

tep 4 : The old segment table entry are copied into new position.

ep 5 : Delete the old segment

6 : Stop.

* Illumination Models and Shading Algorithm.

* Illumination Mode :-

- Refered As lighting model.
- It is used for calculation of intensity of light that is seen at a given point on the surface of an object.

* Light Source :-

- Any object that is emitting radiant energy is a light source.
- That contributes to the lighting effects for other object in a scene.
- We can model light sources with a variety of shapes & characteristics.
- Most emitters serve only as a source of illumination for a scence.
- A light source can be defined with a number of properties.
- We can specify
  - its position
  - color of the emitted light
  - emission direction
  - its shape.

Ambient Light :

- In our basic illumination model, we can incorporate b/g lighting by setting a general brightness level for a scene.
- Ambient light refers to a general level of illumination that does not come directly from a light source.
- It consist of light that has been reflected and Rereflected so many times that it is no longer coming from any particular direction.

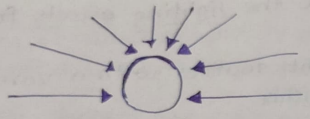


Fig Ambient lighting.

Diffuse Reflection

When a surface is lit by the light, it absorbs some amount of light while the remaining is reflected.

When the incident light is not reflected in one direction but is reflected in all directions.

Reflection of light largely depends on the surface material

The smooth surface su reflects more light as compared to the rough surface.

he shininess of the surface depends on the amount & irection of reflected light.

uch a surface has equal shininess all over this is known s diffuse Reflection.

4] Specular Reflection :-

- When light falls on any shiny or glossy surface most of it is reflected back such reflection is known as specular Reflection.
- light is directional but scatter in preffered direct^n.
- shiny material has high spectularity
- In perfect specular reflection, an incoming ray of light is reflected from the surface intact.
- The reflected ray makes the same angle with the surface as the incoming way.

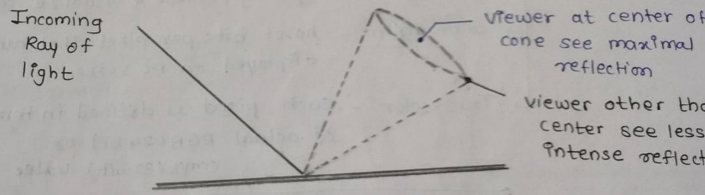
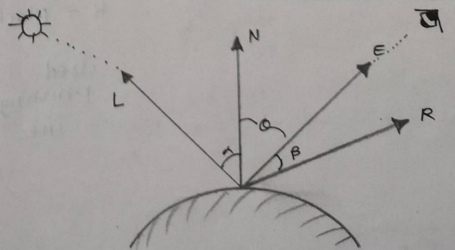


Fig : Specular Reflection

5] Phong Model :

- It is empirical model
- for specular reflection which provides us with the formula for calculation the reflected intensity.

$$I_{spec} = W(\theta) . I_p . \cos^n \theta$$



Color Models

* Color Models
- A system uses 3 primary colors
- By 3 primary color it creates a large rang of color
- Color model is a system used to describe a color

* Color is the aspect of things that is caused by differing qualities of the light being reflected or emitted by them

- Color model is and orderly system for creating a whole range or larger range of colors from a small set of primary etc colors

- 3 parameters
  - color space — specify, create & visualize color.
  - color depth — no. of bits per pixel that can be displayed on PC screen.
  - True color — Each pixel is defined in terms of actual RGB(24 bi) or CMYK(32 bit) value

types of color model.

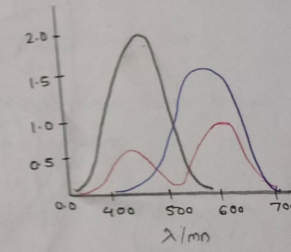
Addictive colors
↓
RGB
(RED Green Blue)
use light

Substrictive colors.
↓
CMYK
C - Cyan
M - Magenta
Y - Yellow
K - Black
used printing ink.

* CIE chromaticity diagram.
- International standard for primary colors proposed by commission International de l'Eclairage (1931)
- Not a computer model but theoretical one
- Imagine X, Y and Z represent vectors in a 3D addictive additive color space
- The X, Y and Z are imaginary primary colors.
- Mathematically defined as positive color matching function
- Additive method

$C = aX + bY + cZ$

- Normalised amounts are then calculated as
  - $x' = \frac{a}{a+b+c}$
  - $y' = \frac{b}{a+b+c}$
  - $z' = \frac{c}{a+b+c}$



- $x' + y' + z' = 1$ & it lies in the rang of (0,1)
- Given 2 normalized amounts, 3rd one easily computed.
  - $z' = 1 - (x' + y')$
- To compute the actual amount from the normalized amount For ex.
  - $\frac{x'}{y'} = \frac{a}{b}$, $a = b \cdot \frac{x'}{y'}$
  - $z' = 1 - (x' + y')$, $\frac{y'}{z'} = \frac{b}{c}$ so, $c = b \cdot \frac{z'}{y'}$

Advantages

- Comparing color gamuts for different set of primary colors
- identifying complementary colors.
- Determining dominant wavelength and purity of color

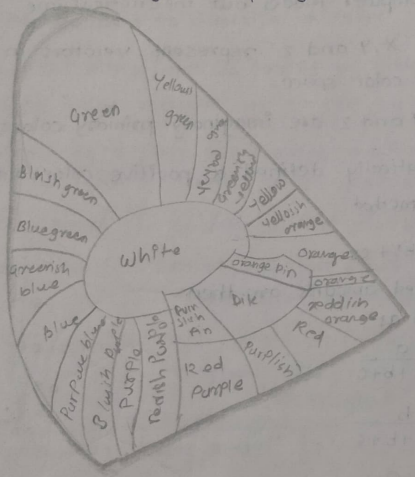
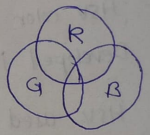


Fig Chromaticity diagram

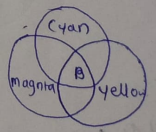
* RGB

- RGB is additive color model for ~~me~~ computer displays uses light to display color
- Color result from transmitted light.
- Red + Green + Blue = White

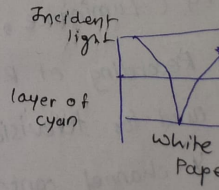


CMYK color model

- standard color model used in offset printing for full color documents.
- Because such printing uses inks of these four basis color, it is often called four-color printing.
- Where 2 colors of RGB overlaps, we see new color formed by mixing of 2 additive primaries These new color

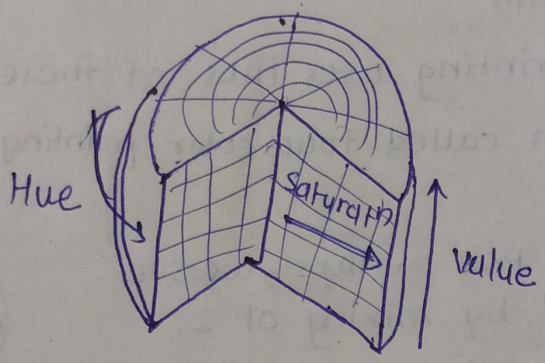


- Greenis blue cyan
- Blushed red magenta
- bright yellow yellow
- Black



HSV

- Hue Staturation , Value
- Describes colors (hue or tint)
- HSV color model based on polar co-ordinates
- Developed in 1970's for CG application
- HSV used today in color pickers.
- Image editing slw
- less commonly in image ~~any~~ analysis & computer vision.



YIQ (Luminance , Inphase, Quadrature) color model.

- Recording of RGB cm
- used for television broadcasting.
- Y-channel contains luminane infor (sufficient for Black & white television sets)
- I-channel & Q-channel contains color information

Inphase = RED - YELLOW

Quadrature = Blue - Yellow

UNIT IV

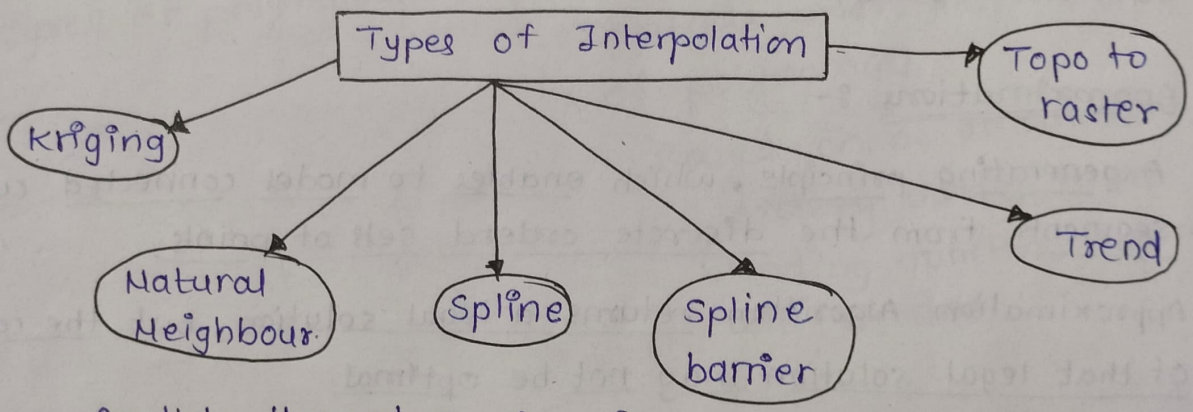
Curve, fractals & Animation.

* Curves

- A curve is an infinitely large set of points.
- In computer graphics, we often need to draw different types of object onto screen.
- Objects are not flat at all the time and we need to draw curve is an infinitely / many times to draw an object.

* Interpolation and Approximation.

- **Interpolation** :- It is a method of constructing new data points within range of discrete set of known data points.
- The main task is to find suitable mathematical expression for known curve.
- This technique is used when we have to draw curve by determining intermediate points between known sample pts.



1) kriging - Predicts the value of a function at a given pt by computing a weighted average of the known values of the function in the neighborhood of the pt.

2) Natural Neighbour - Find colest closest subset of i/p samples to query point.

Applies weigh to i/p sample based on proportion areas to interpolate value

3) spline - estimation of values is done using mathematical functions that minimize overall surface curvature. Results in smooth surface that passes exactly through i/p pts.

4) Spline with barries - same as spline, only diff. is that this tool honors discontinuities encoded in both barriers & i/p pts data.

5] Topo to Raster - - Uses technique specifically designed to create surface that more closely represents natural drainage surface
- better preserves stream n/w from i/p contour data.

6) Trend - It is global polynomial interpolation that fits, smooth surface defined by mathematical functions to i/p sample pts.

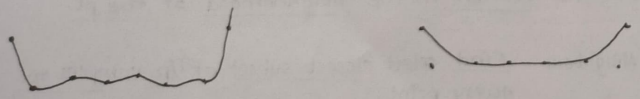
The trend surface changes gradually and capture coarsescale pattern in data.

* Approximations :-

- A generating principle, which enables to model connected curve segments from the discrete ordered sets of points.

Approximation Algorithm returns a legal solution, but the cost of that legal solution may not be optimal.

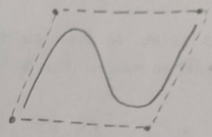
If the curve doesn't pass through the given control pts & we approximate the shape, it is called approximation



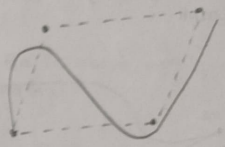
Interpolation vs Approximation

* Convex hull :-

- The curves generally satisfy the convex hull property.
- The convex polygon boundary that encloses a set of control pts is called as convex hull.
- If the curve satisfy in convex hull property then it should lie within the convex hull else should not.

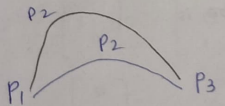


(a) Curve satisfies convex hull P.

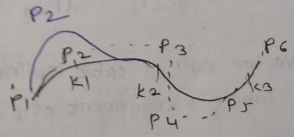


b) Curve donot satisfies convex hull pr.

- Bezier curve
  - global control
  - depent on cntrl pt. & degree of polynomial

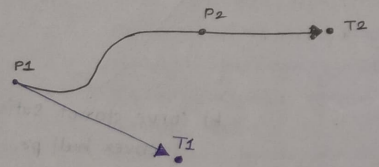


β-spline - local control
Degree of Polynomial
Does not dependent on cntrl pt
Depends upon the order of polynomial.
Based on no. of cntrl pt if ordery varies then blending functions degree will be diff



* Spline Interpolation method - hermite interpolation.

- Hermite spline is named after the french Mathetician charles Hermite
- It is an interpolation piecewise cubic polynomial with a specific tangent at each control point
- It can be adjusted locally as each curve section is dependent only on its endpt



- P1 : Start pt Hermite curve
- P2 : endpt of Hermite curve
- T1 : Tangent to the start pt
- T2 : Tangent to the end pt
- P[u] is the parametric cubic point function.

Parameteric cubic pt functⁿ for any curve section is

$P[0] = P_K$
$P[1] = P_{K+1}$
$P'[0] = dP_K$
$P'[1] = dP_{K+1}$

} Derivative at pt P_K & P_{K+1} respectively

Vector eqⁿ of cubic spline is : $\boxed{P[u] = au^3 + bu^2 + cu + d}$

where x component of p is : $\boxed{x[u] = a_x u^3 + b_x u^2 + c_x u + d_x}$

Spline Interpolation method - hermite inter (continue)

→ Matrix form of above eqⁿ is $P(u) = [u^3\ u^2\ u\ 1] \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$

→ Derivatives of $P(u) = [u^3\ u^2\ 3au^2 + 2bu + c + 0$

→ Matrix form of $P'(u)$ is : $P'(u) = [3u^2\ 2u\ 1\ 0] \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$

- Substitue end pt value of u as 0 & 1 in above eqⁿ & combine all 4 parametric eqⁿ in matrix

$$\begin{bmatrix} P_K \\ P_{K+1} \\ dP_K \\ dP_{K+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

- Solve it for polynomial co-efficient

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_K \\ P_{K+1} \\ dP_K \\ dP_{K+1} \end{bmatrix}$$

$H_K(u)$ for $K = 0,1,2,3$ are refered to as blending function

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = M_H \begin{bmatrix} P_K \\ P_{K+1} \\ dP_K \\ dP_{K+1} \end{bmatrix}$$

- Put value of above eqⁿ in eqⁿ of $P(u)$

$$P(u) = [u^3\ u^2\ u\ 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_K \\ P_{K+1} \\ dP_K \\ dP_{K+1} \end{bmatrix}$$

$P(u) = P_K[2u^3 - 3u^2 + 1) + P_{K+1}(-2u^3 + 3u^2) + dP_K(u^3 - 2u^2 + u) dP$

$P(u) = P_K H_0(u) + P_{K+1} H_1(u) + dP_K H_2(u) + dP_{K+1} H_3(u)$

B-spline curve.

- B-spline basic contains the Bernstein basis as the special case.
- The B-spline basis is non-global, its local
- A B-spline curve is defined as a linear combination of control points P_i and B-spline basis function $N_{i,k}$ t given by

$$C(t) = \sum_{i=0}^{n} P_i N_{i,k}(t),$$

$$n \geq k-1, \quad t \in [t_{k-1}, t_{n+1}]$$

- $\{P_i ; i = 0, 1, 2 \ldots n\}$ are the control point.

k is the order of the polynomial segments of the B-spline curve.

order k means that the curve is made up of piecewise polynomial segments of degree k-1,

$N_{i,k}(t)$ are the "normalized B-spline blending function"

$t_i ; i = 0, \ldots n+k$

The $N_{j,k}$ function are described as follow:

$$N_{i1}(t) = \begin{cases} 1, & \text{if } u \in (t_i, t_{i+1}) \\ 0, & \text{otherwise} \end{cases}$$

nd if $k > 1$

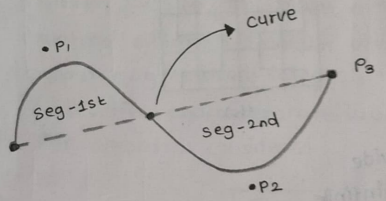
$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1}} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

d

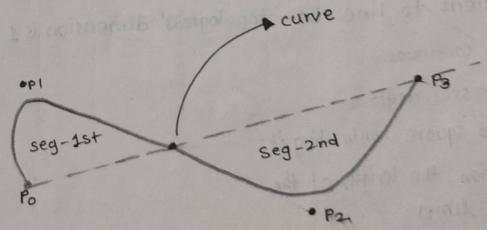
$$t \in [t_{k-1}, t_{n+1}]$$

* Properties of B-spline curve

- The sum of the B-spline basis functions for any parameter value is 1
- Each basis function is +ve or 0 for all parameter values.
- Each basis function has precisely 1 max value expect k=1.
- The max. order of the curve is = no. of vertices of defining Polygon.
- The degree of B-spline polynomial is independent on the no. of vertices of defining polygon.
- The curve exhibits the variation diminishing property
- The curve generally follows the shape of defining polygon.



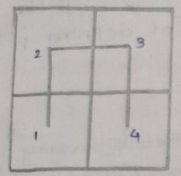
Before changing the Position of control Pt P1



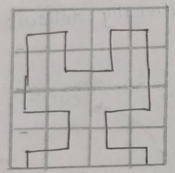
After changing the Position of control Pt P1

* Hilbert Curve

- It was introduced by the German mathematician David Hilbert
- Also known as Hilbert Space-filling curve.
- The basic entity of curve is u shaped.
- It requires successive approximation.
  - a) 1 approximation the square is divided into 4 quadrants and connect center points.
  - b) In the second approximation further every quadrants are divided which cannot be center of each.



(a)


(b)

- There is no limit to subdivide.
- Ideally length of curve is infinite.
- Every subdivision the length increase by 4
- The curve is equivalent to line. Its topological dimension is 1

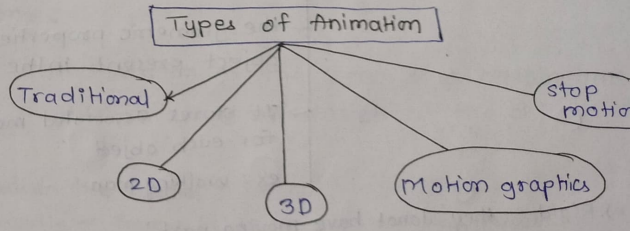
On applying this process continuosly

- The curve never crosses itself
- Curve gets closer to square containting it
- With each subdivision, the length of the curve increases 4 times

* Animation :-

- Computer Animation refers to the time sequence of visual changes
- Animation is a key concept in envirnoment, entertainment, education, simulation, games scientic & engineering studies
- Scope of Animation is extended to various transformation operations like scaling, rotation, variation in color, transparency, surface property, shape etc.
- Animation is achived by displaying successive frames with minor difference
- It is a process of taking sequential drawing, models, or even puppets to create a illusion of sequnces.
- Creating a smooth motion view from these sketched, computer-generated images, frame rate, or the no. of consecutive characters are usually shot, images displayed Per second is considered.

*



Design of Animation sequence

- In a frame by Frame animation, each frame is generated using interpolation or some other method.
- Frames are stored on film or they are consecutively displayed in playback mode.

* Animation sequence
  - Storyboard layout :
    - center of Action
    - the set of motion sequence taking place in the action
    - Defines main action of events
    - consist of sketches of the imp ideas of animation
    - Its basic event in animation
  - Object definition :
    - Scene is collection of object
    - the geometric properties of each object present in the scene
    - It stores associated movements for each object
    - ex : walking man.

a) Rigid : they donot have moving part

b) Articulated : such obj. are formed using parent child hierarchy. subpart may rigid but 1 to 2 subpart movement allowed

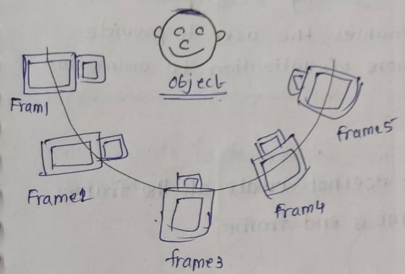
) Dynamic : such model employs the law of physics for the simulation of the object body

Particle based : follow rule of nature

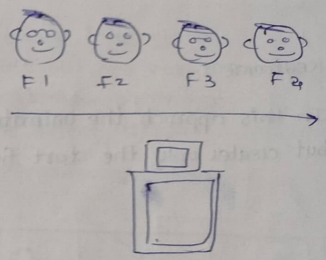
Behaviour ; such type of animation mimic the behaviour of real animals of the planet.

3) Path specification
   - a) The obj. is stationary, camera in motion
   - b) The camera is static / stationary, object in motion.

- This specification tells about how the movement is added to the object There are 2 types. a & b ↑



a) object is stationary camera in motion.



b) camera is stationary obj. in motion.

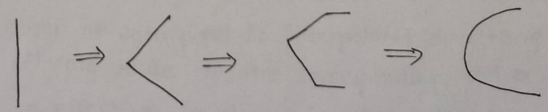
4] Keyframe :

- Actual Representation of the scene at a certain time
- Defines position, scale & orientation of the obj

5) In between the frames :

- Intermediate frames between a pair of keyframes.

24 Frames ⟶ film, 60 Frames ⟶ non-interlaced display



shape transformation

* Animation language

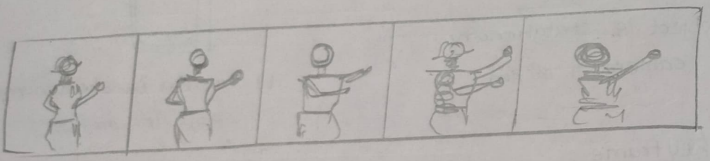
1) linear list notation : describe by start & ending frame

2) Genral purpose : ↑ level lang. Comp. prog. lang. like C, C++, Java used slw app. dev. also support animation using diff. features along with drawing of graphical obj.

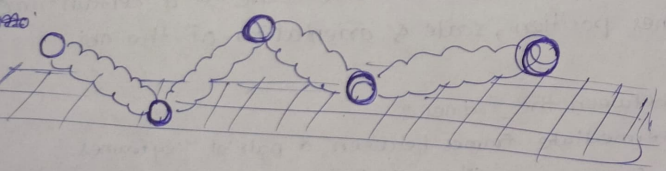
3) Graphical language : Enables the user to provide more visualization to animation

* keyframe :-

- In this approch the animator doesnot create all the frames but creates only the start frame & end frame.

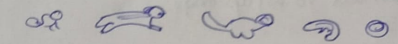


key frame animation can be applied to motion, shape, color etc.

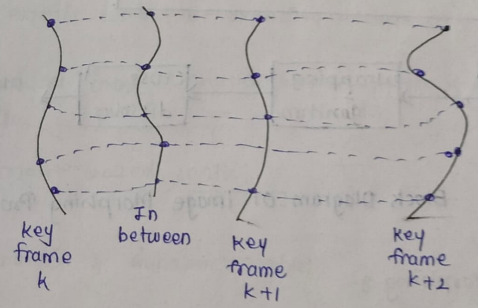


Approces For key frame Animation.

1) Shape Interpolation [used in movies advertisement world
shape of an object changes due to the app of geometric transformtn.

Given frame 

2) Parameter interpolation : The shape interpolation is fast but not very accurate
one way is to interpolte the parameter of the 2 object rather than construct the object itself



* Morphing :-

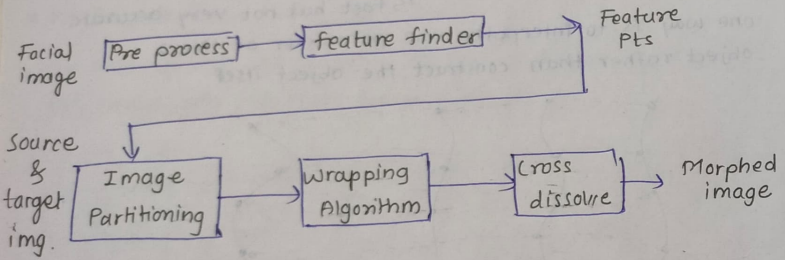
- Is a special effect in motion pictures and animation that changes one image into another through a series of continous transition.
- cross dissolving : - transform 1 image to another
- colour of each pixel is interpolated over time from the 1st img value to the corresponding 2nd img value
- so effective in displaying the actual metamorphosis

b) Wrarping: Refers geometric transformation of graphical objects from one co-ordinate sys to another

- Works with only one obj
- slowly changes - wrap it - over a series of frames.

— Two methods to wrap img.

1) Forward mapping:-
2) Reverse mapping



Block Diagram of image Morphing Process.

Applicatn of Morphing :-

- Medical appliation — How organ will look.
- movies, animation
- Marketing advertising
- Design of vehicles

* Motion Specification

— Method of motion secificatn
  1) Direct motion
  2) kinematics & Dynamics.

* Methods of controlling Animation

a) full explicit control or explicitly declare cntrl

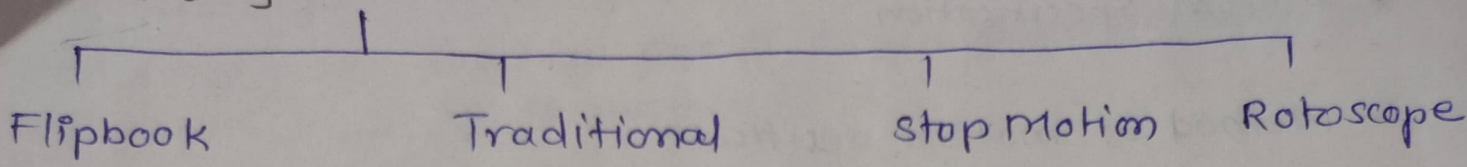
b) Procedural cntrl (Physical based system & Actor based

c) constriant based control

d) kinematics & dynamic control

e) Control by analyzing live action.

* Frame by Frame Animation



* Advantages Frame by Frame

- It is traditional
- It offers More creative control
- It is Fantasy-like.

* Disadvantage.

- time consuming
- Expensive
- Its not Realistic looking.