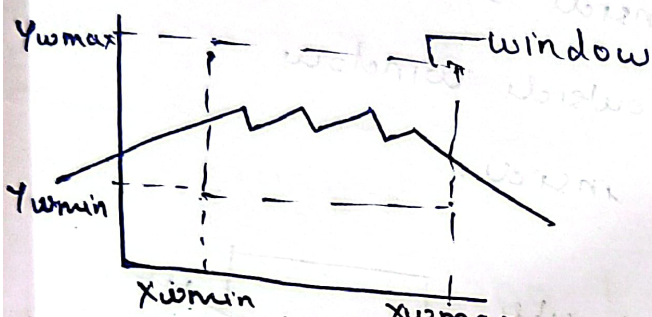
Unit-2

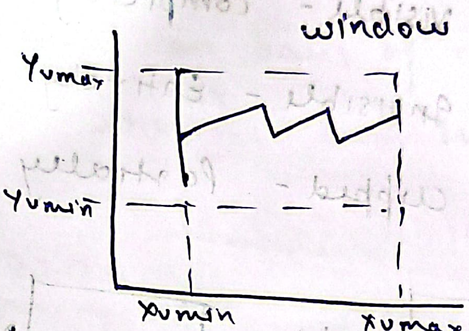
2-D viewing

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

window - method of selecting and enlarging a portion of drawing. It is selected from world co-ordinate.



world co-ordinate



Device co-ordinate

viewing transformation - mapping of a part of world co-ordinate scene to device ~~coo~~ co-ordinate. is called viewing transformation.

A point at position (xw, yw) in window is mapped to (xv, yv) in viewport. In order to maintain relative placement we require.

$$\frac{X_v - X_{vmin}}{X_{vmax} - X_{vmin}} = \frac{x_w - X_{wmin}}{X_{wmax} - X_{wmin}} \quad \text{eq}^n\ 1$$

$$\frac{Y_v - Y_{vmin}}{Y_{vmax} - Y_{vmin}} = \frac{y_w - Y_{wmin}}{Y_{wmax} - Y_{wmin}} \quad \text{eq}^n\ 2$$

solving above 2 eq^n

$$x_v = x_{vmin} + (x_w - x_{wmin}) \times \frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}}$$

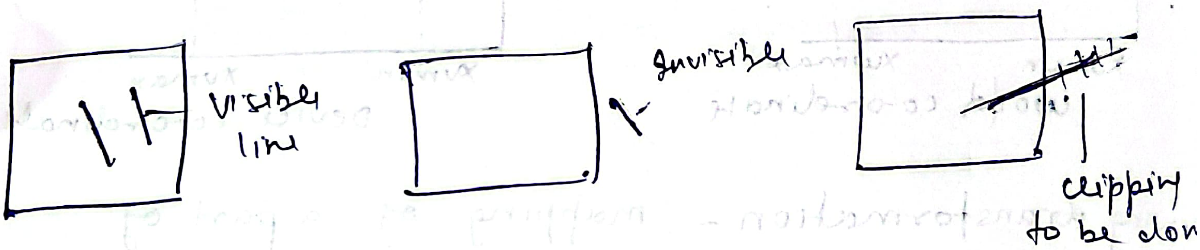
$$Y_v = Y_{vmin} + (Y_w - Y_{wmin}) \times \frac{Y_{vmax} - Y_{vmin}}{Y_{wmax} - Y_{wmin}}$$

Clipping - for deciding the visible and invisible portion a particular process clipping is used. It determines which portion to show and which to reject.

line - visible: completely inside window

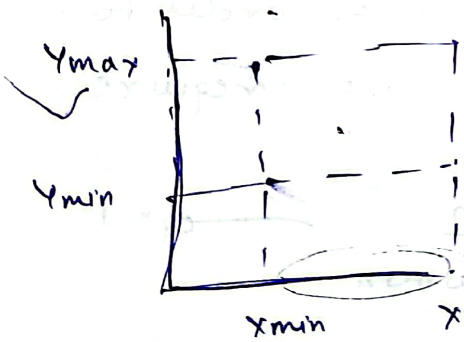
invisible - Entirely outside window.

clipped - Partially inside.



Types of clipping

Point clipping - It is used to determine whether the point is inside window or not.



conditions -

$X \leq X_{max}$

$X \geq X_{min}$

$Y \leq Y_{max}$

$Y \geq Y_{min}$

If all this condition is satisfied then point is inside window.

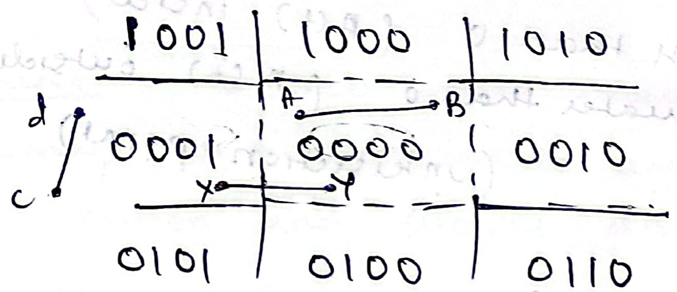
Line clipping -

Cohen Sutherland Algorithm - In this algorithm, it is detected whether line lies inside screen or not. There are three cases -

1) visible (entirely in)
2) Not visible (entirely out)
3) clipping case. (Partially in)

All category of line are found on basis of region. There are nine region as shown. and they are assigned codes.

TBRL



0000

Not

Case - take And of end points.

A & B = 0000

if and is not 0000 then line is outside.

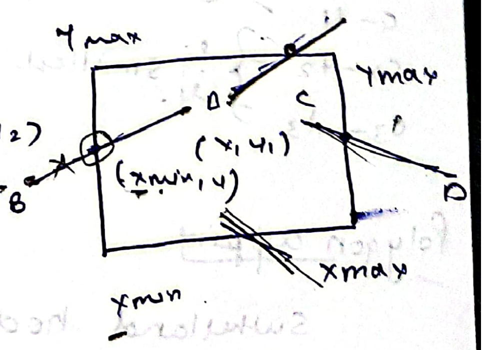
otherwise - it may be partially inside or completely inside.

For - AB

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

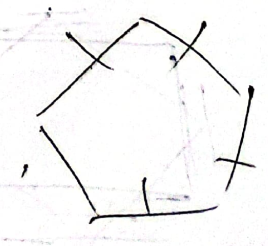
$$= \frac{y - y_1}{x_{min} - x_1}$$

$$y - y_1 = m(x_{min} - x_1)$$



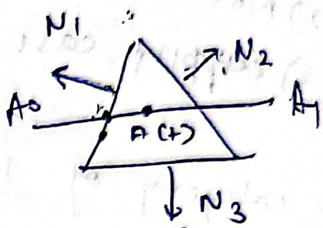
For CD -

$$y - y_1 = m(x_{max} - x_1)$$



Cyrus-Beck line clipping algorithm -

In sutherland line clipping algorithm we are constrained to have rectangular viewport. If viewport have different shape then we are not able to clip so, cyrus clipping algo overcome that disadvantage.



N_i - Normal to edge.

P_{ei} = Point belonging to edge.

$$A(t) = A_0 + t\,(A_1 - A_0)$$

Calculate → $N_i\,(A(t) - P_{ei})$

- less than 0 ($A(t)$ inside)
- greater than 0 ($A(t)$ outside)
- $= 0$ (intersection point)

$$N_i(A_0 + t(A_1 - A_0) - P_{ei}) = 0$$

$$N_i A_0 + N_i t A_1 - N_i t A_0 - N_i P_{ei} = 0.$$

$$t_1 = \frac{N_i(P_{ei} - A_0)}{N_i(A_1 - A_0)}$$

- $d < 0$ entering point
- $d > 0$ leaving point.

$e_1 - t_1$ — e

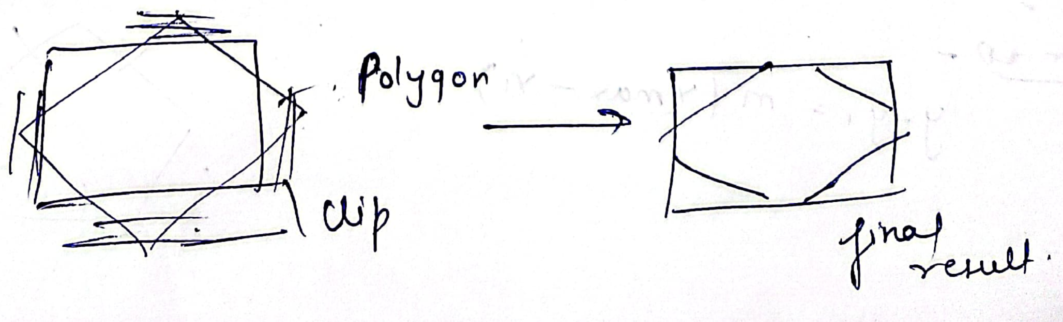
$e_2 - t_2$ } l } smallest

$e_3 - t_3$ } l

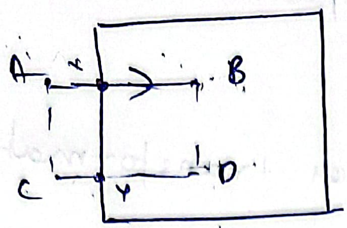
l
e } l g greatest.
e

Polygon clipping

Sutherland hodgeman Polygon clipping algorithm



Perform these step on all sides, Top, Bottom, left, Right.



A → B outside inside.
x (intersection)
(XB) (Save)
B → D 9 - 9 (save C)
D → C 9 - 0
save Y
C → A - 0 - 0 (ignore).

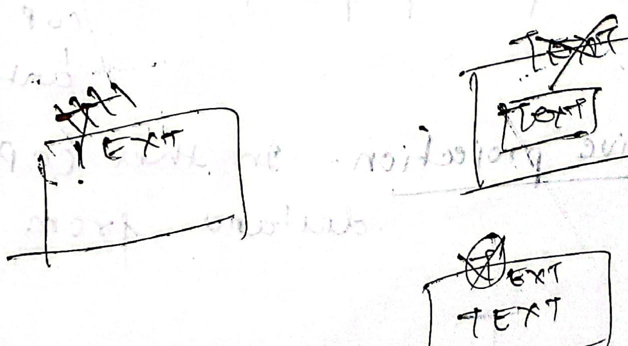
Text clipping

•) All or none string (take whole string or reject it)
•) All or none character. (take whole character or reject)
•) Individual character. (take partial characters ~~algo~~ also)

Segment - to view entire image or a part of image ~~with~~ we need to organise image information and file. is divided into segment. Segment corresponds to component and is associated with set of attributes like scaling, rotation.

Type - Posted - visible attribute is set to 1
Unposted - " " " " to 0

Segment table - is used to keep the record of length of similar segment.



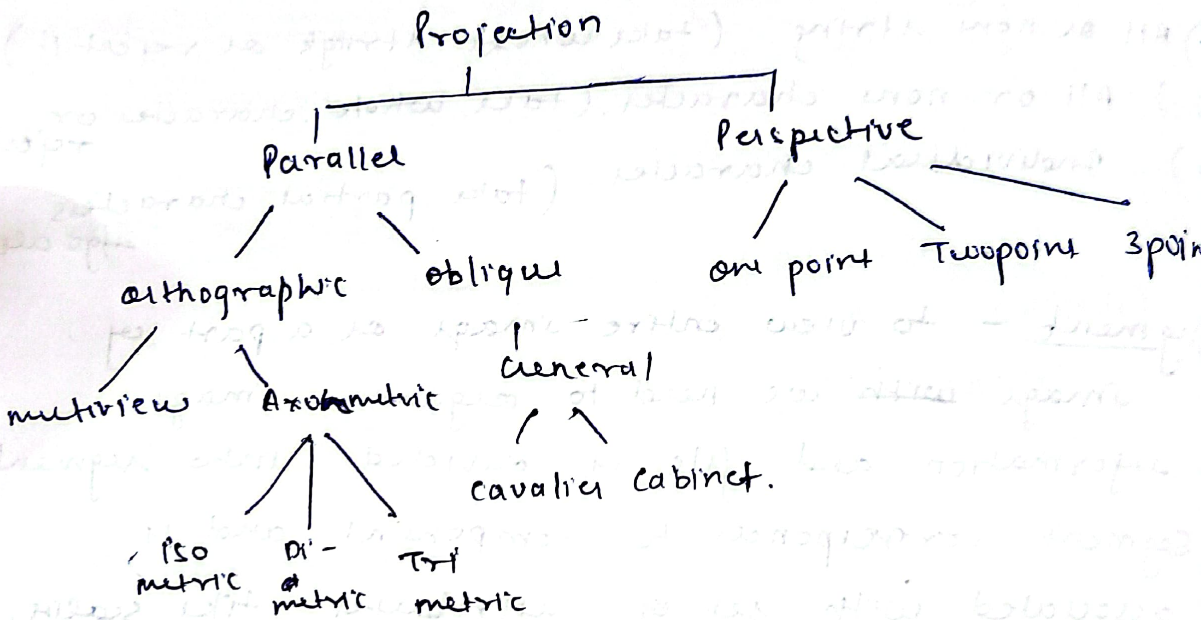
Projection -

Representing n-dimensional object into (n-1) dimension. is know as Projection.

Ex- 3D to 2D.

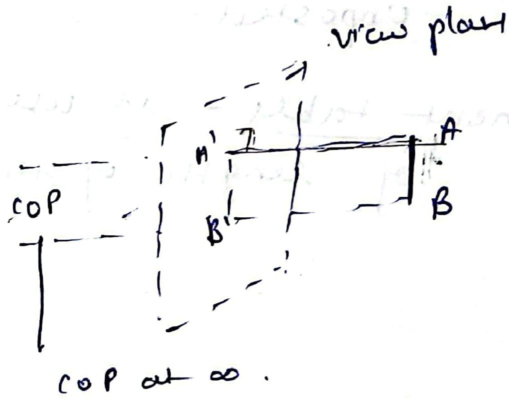
It is also defined as mapping as transformation of object in projection plane.

Types - Parallel
Perspective



Parallel Projection - In this co-ordinate position are transferred to view plane along parallel line.

A projection is said to parallel if centre of projection is at infinite distance from projected plane.



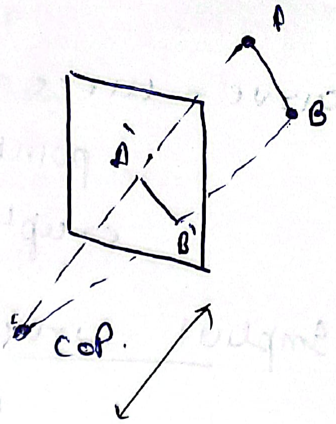
centre of Projection

Perspective projection - In this COP is at finite distance from projected plane.

visual effect is similar to human system.

object appears smaller.

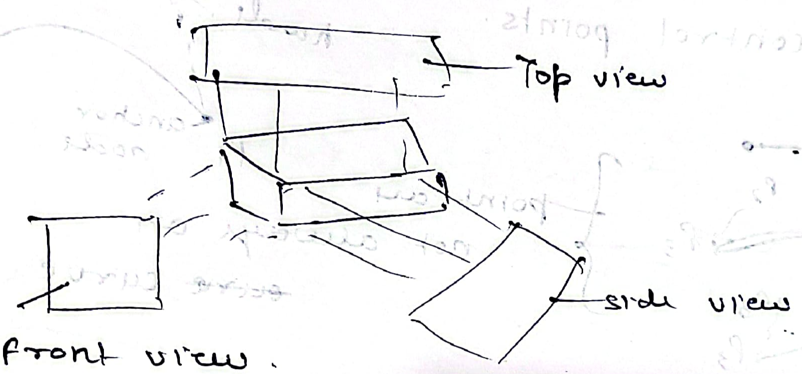
difficult to determine exact size.



Parallel Projection

1) orthographic - In orthographic projection the direction of projection is normal to projection of plane.

Three types of orthographic projection -



oblique projection - In oblique projection the direction of projection is not normal to projection plane.

Type

└ cavalier - If the ~~the~~ projected line make angle of 45° with plane as a result line of length of object do not change.

cabinet - If the angle is 63.4 then it is cabinet ~~po~~ projection. viewing surface are projected at 1/2 their actual length.

Curves

Curve - curve is infinitely large set of points. Each point has two neighbour except end point.

Implicit curve -

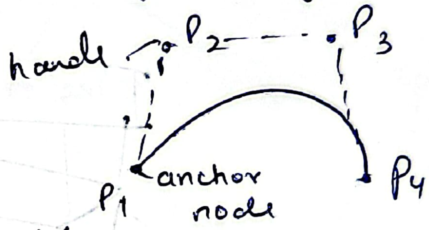
$$f(x, y) = 0$$

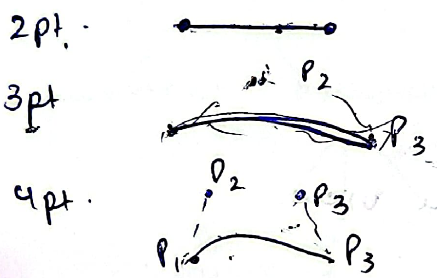
Explicit curve -

$$y = f(x)$$

Parametric curve - $P(t) = \cancel{f(t)}$
$= x(t), y(t)$

Bezeir curve - Bezeir curve are defined by control points.





points are not always on ~~curve~~ curve.

order of curve = (n-1) point.

4 point curve is Bezeir curve.

Properties - They always pass through first and last point.

- contained in convex hull.
- degree = (n-1) pt.
- Shape of defining polygon is usually followed by Bezeir curve.
- Bezeir curve exhibit global control. means moving a control point alters the shape of whole curve.

$$P(u) = \sum_{i=0}^{n} P_i \, B_{i,n}(u) \quad 0 \le u \le 1$$

Basis function.

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

n = degree

P = control point.

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

cubic bezeir

$$P(u) = P_0 B_{0,3}(u) + P_1 B_{1,3}(u) + P_2 B_{2,3}(u) + P_3 B_{3,3}(u)$$

$$B_{0,3}(u) = C(3,0)\, u^0 (1-u)^3 = \frac{3!}{0!\,3!}(1-u)^3 = (1-u)^3 \quad \text{--- (1)}$$

$$B_{1,3}(u) = C(3,1)\, u^1 (1-u)^{3-1} = 3u\,(1-u)^2 \quad \text{--- (2)}$$

$$B_{2,3}(u) = C(3,2)\, u^2 (1-u)^{3-2} = 3u^2(1-u) \quad \text{--- (3)}$$

$$B(3,3)(u) = C(3,3)\, u^3 (1-u)^{3-3} = u^3 \quad \text{--- (4)}$$

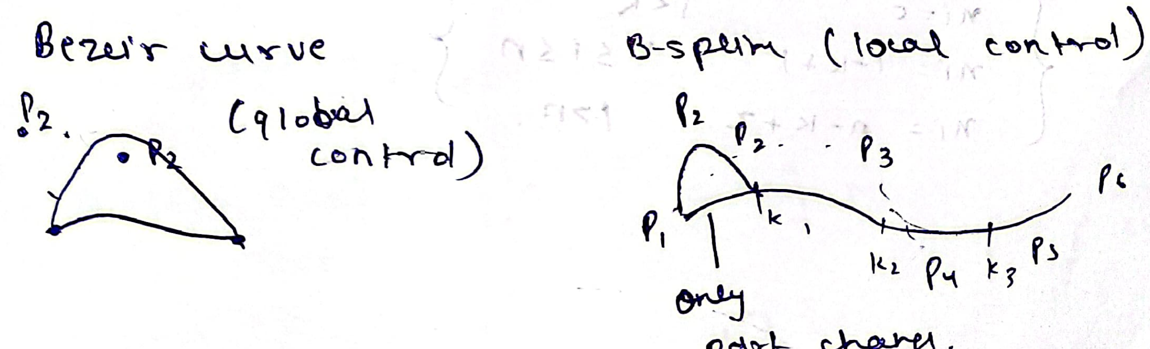
$P_0 = (1,1)$

$P_1 = (2,2)$

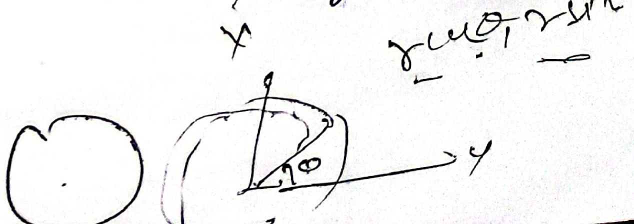
$P_2 = (3,3)$

$$P(u) = P_0 (1-u)^3 + P_1 3u(1-u)^2 + P_2 3u^2(1-u) + P_3 u^3$$

B-spline curve - It does not depends on control point. It depends on order of polynomial.



$$x^2 + y^2 = r^2$$



Properties

- The sum of B-spline basis function at any parameter 'u' equals to 1 i.e

$$\sum_{i=1}^{n+1} N_{i,k}(u) = 1$$

n+1 = control point
k = order of B-spline curve.

- Basis function is +ve or zero except for k=1

- maximum order of curve is equal to no. of vertices of defining polygon.

- B-spline allows local control.

- curve lies within convex hull.

B spline curve represented as -

$$P(u) = \sum_{i=1}^{n+1} P_i N_{i,k}(u).$$

where. $u_{min} \le u \le u_{max}$.

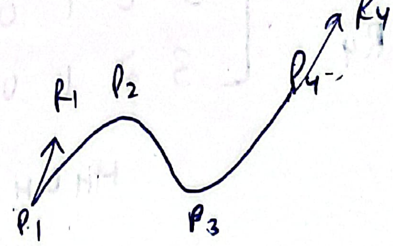
$$N_{i,k}(u) = \frac{u - x_i \; N_{i,(k-1)}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) \cdot N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

$$N_{i,k}(u) = \begin{cases} 1 & x_i \le u \le x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{cases} x_i = 0 & i < k \\ x_i = i - k + 1 & k \le i \le n \\ x_i = n - k + 2 & i > n. \end{cases}$$

Hermite curve

- Interpolation spline curve (all points are used)
- cubic polynomial function



making hermite function.

$t=1$

$0 \le t \le 1$

* first control point
* last control point

} step position vector

* first derivative of points.

$R_1 = P_1'$

$R_4 = P_4'$

$Q(t) = [x(t) \quad y(t) \quad z(t)]$

$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x$

$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y$

$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z$.

$$Q(t) = \underbrace{[t^3 \quad t^2 \quad t \quad 1]}_{(T)} \cdot \underbrace{\begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}}_{(C)}$$

hermite. $Q(t) = T \cdot C$

$C = M \cdot G$

M = basis matrix - Provide Blending eqn.

G = geometric vector matrix

$Q(t) = T \cdot M \, G$.

$Q_x(t) = P_1(t)_{t=0} = [0 \quad 0 \quad 0 \quad 1] \cdot M_H G_H$.

$Q(x)(t) = P_4(t)_{t=1} = [1 \quad 1 \quad 1 \quad 1] \cdot M_H \cdot G_H$.

$Q'_x(t) = R_1(t)_{t=0} = [0 \quad 0 \quad 1 \quad 0] \cdot M_H \cdot G_H$.

$Q'(x)t = R_4(t) = [3 \quad 2 \quad 1 \quad 0] \cdot M_H G_H$.

$$\begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} M_H G_H \quad a.$$

$$M_H G_H = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

$$M_H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad G_H = \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

$$Q(t) = T \cdot M_H G_H$$

$$= \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$

cubic hermit curve

$$= \underbrace{(2t^3 - 3t^2 + 1)}_{H_1} P_1 + \underbrace{(-2t^3 + 3t^2)}_{H_2} P_4 + \underbrace{(t^3 - 2t^2 + t)}_{H_3} R_1 + \underbrace{(t^3 - t^2)}_{H_4} R_4$$

$$Q(t) = H_1 P_1 + H_2 P_2 + H_3 R_1 + H_4 R_4$$

H_1, H_2, H_3, H_4 are Hermite Blending function.

Illumination model -

Illumination model is used to calculate the intensity of light that is reflected a given point on surface.

An object is illuminated from ambient light and from light source.

Ambient light - light ~~from~~ coming from the nearby object after reflection.

The amount of ambient light incident on each object is constant for all surface & over all direction.

I_a = ambient intensity

Independent of surface orientation, view location

but diff surface reflect diff amount of ambient light.

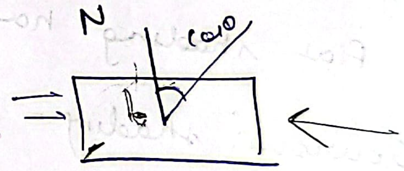
$$I_{amb} = k_a \cdot I_a$$

diffuse reflection - light reflected in all direction.

The reflected light is independent of view location. but depends on surface orientation.

The brightness at each point is proportional to $\cos\theta$

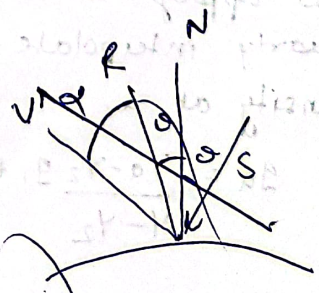
$$I_{diff} = k_d \cdot I_p \cos\theta$$



specular reflection - It is the white light seen on smooth shiny object.

Depends on surface orientation, viewer location.

generally seen on metal surface,

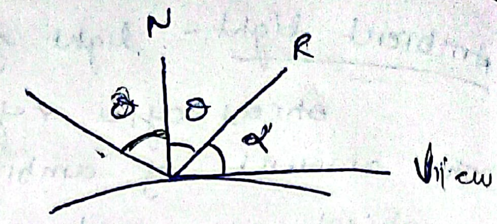


~~R = Reflection vector~~

~~$I_s = R k_d \cos^n\alpha$~~

~~n = specular intensity.~~

$I = I_p K_s (\cos\alpha)^n$



I_p = ~~the~~ light intensity

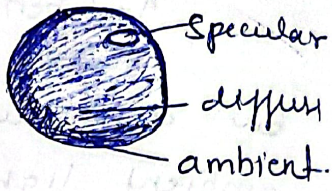
α the angle

K_s = specular reflectivity.

$R \cdot V = \cos\alpha$

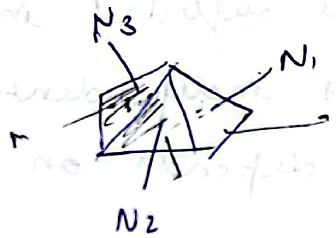
Do not depend upon color of object.

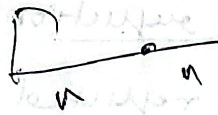
total $I = I_{amb} + I_{diff.} + I_{spec}$



Shading -

Flat shading - constant intensity shading.

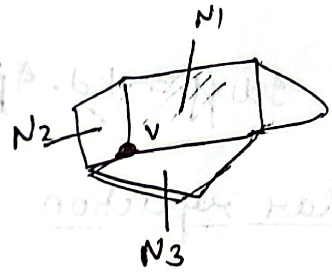




find intensity of each face and shade with it.

Flat shading has high discontinuity.

Gouraud shading - find vertex and calculate.

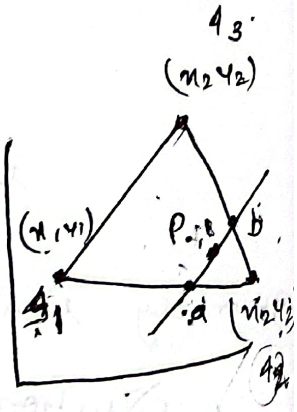


average unit normal vector.

$$N_V = \frac{N_1 + N_2 + N_3}{|N_1 + N_2 + N_3|}$$

Now apply illumination model. -

3 - linearly interpolate the vertex intensity over surface.



Intensity at a.

$$I_a = \frac{y_a - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_a}{y_1 - y_2} I_2$$

$$I_b = \frac{y_b - y_2}{y_2 - y_3} I_3 + \frac{y_3 - y_b}{y_2 - y_3} I_2.$$

(Interpolate Intensity)

$$y_p = \frac{x_b - x_p}{x_b - x_a} y_a + \frac{x_p - x_a}{x_b - x_a} y_b$$

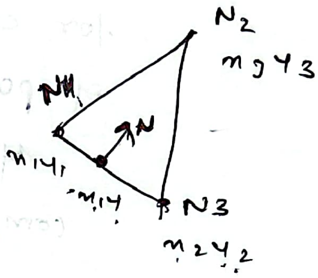
- discontinuity removed.
- sharp intensity can not be displayed
- Bright, Dark streaks appear.

Phong shading -

A more accurate interpolation based approaching for rendering polygon.

It interpolates normal vectors.

$$N = \frac{y - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y}{y_1 - y_2} N_2$$

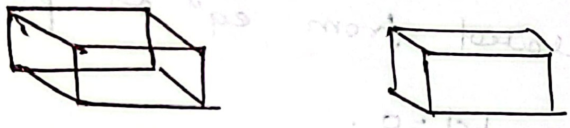


Apply the illumination model along each face.

- Realistic shading
- bit slower than gouraud.

Hidden line and surface

Polygon
- Image space - pixel
- Object space - part of polygon.



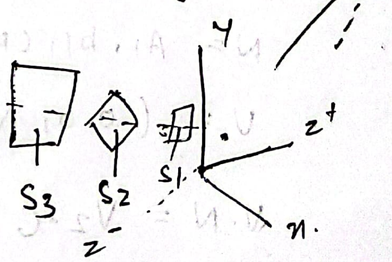
Object with hidden line — object without hidden line.

We need to remove these line and surface

Algorithm

Depth buffer or z-buffer (works for opaque surface)

Each surface is processed separately one point at a ~~xy~~ time across the surface.



Two buffer
- frame buffer (store intensity)
- depth buffer. (store depth)

$D(x,y) = z_{max}(\infty)$

Step-1 set the buffer values.

depth buffer $(x,y) = 0$

frame buffer (x,y) = background color.

Step-2 Process each polygon.

for each projected (x,y) pixel of polygon calculate z (depth)

if $z >$ depth buffer,

compute surface color

set depth buffer $(x,y) = z$,

frame buffer = surface color.

Adv - Easy to implement

reduces speed problem.

Disadv - large memory

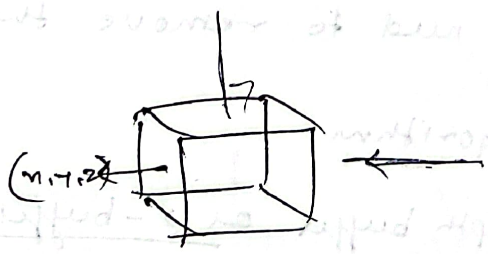
time ↑

Depth value is calculated from eqⁿ of plane.

$$ax + by + cz + d = 0.$$

$$z = \frac{-ax - by - d}{c}.$$

Back face removal algorithm -



i) $Ax + by + cz + d < 0$ inside surface.

ii) $V \cdot N > 0$ backface.

$N = Ai, bj, ck.$

$V = (0, 0, V_z)$

$V \cdot N = V_z \cdot C$ sign (c) ≤ 0 Backface, > 0 frontface.

Drawback - do not work with partial surface.

A- buffer - Accumulation (works for transparent object also).

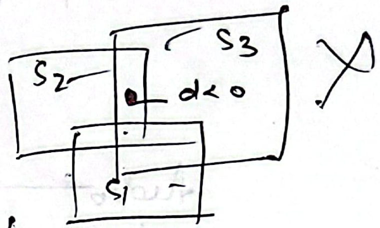
2 field

Depth field — +ve, -ve.

Intensity field.

$d<0$ - multiple surface

$d>0$ - single surface



d<0	→	S1	→	S2	→	S3	→

Scan line algorithm

Edge table — Active edge table — Polygon table.

Edge table

X	Ymax	Δx	ID	→

AET

Scan line	edge	surface		→

ID	Polygon coeff.	Shading	A/out	→

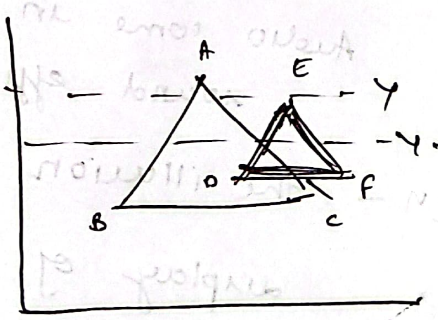
X: when y is small.

$\Delta x = 1/m$

ID - id of edge.

Polygon coeff = A, B, C, d.

$Ax + by + cz + d = 0$.



AET

L1: AB — ABC

AC — ABC

L2 = AB = ABC

AC, ED = ABC, EDF (Depth comparison) (top)

EF → EDF

Multimedia

Multimedia def is a word derived from multi and media multi mean many and media - tool that is used to represent or do certain things, delivery medium, a form of mass communication ~~text~~ ex- text, graphic, voice, image, etc.

Text
Graphic
Audio —— Multimedia
Video
animation

Text → A broad term for something that contain words to express something.
- most basic element

Graphic - Two dimensional figure or illustration
- can be produced by drawing or computer.

Audio - Produced by vibration, perceived by sense of hearing.
Audio come in form of speech, sound effects, music.

Animation - the illusion of motion created by display of static images.
Animation enchances the experience and conveys more information.

Video - technology of [illegible],
processing, transmitting moving pictures.

- video is more towards photo realistic image sequence compared to animation.
- take lots of space.

Hypermedia - combination of hypertext, graphics, audio, video linked together.

Linear multimedia -

A multimedia is linear when:

- It is not interactive
- user have no control over content that is showed.

Example - movie.

Non-linear media -

A multimedia is non linear when:

- It is interactive
- User have control over what is shown to them

Example - games.

Importance of multimedia

Business - Marketing / Training / Trade show

Education - Elearning / Simulation

~~Entat~~ Entertainment - Games / movies

Home - sms / tv

Public places. - smart card / security

Multimedia tools - tool used to create multimedia

Ex - photoshop, flash.

Animation -

A collection of static images joined together and show consecutively so that they appear to move.

- It's about bringing things to life.

Use - Artistic purpose

Story telling

display data

Principles of Animation

Timing

Secondary Action

follow through and overlapping action

Straight ahead action and pose to pose action

Staging

Appeal

solid drawing

Ease in - out

Arcs

Anticipation

Squash & stretch

Exaggeration -

Types of ~~Ani~~ Animation -

cell Animation - method used for creating hand drawn animation.

Digital Animation - electronically generated movement on computer screen.

- Basic ——— simple transitions
- Intermediate ——— special effects with color, photo etc
- Advanced ——— 3D animation, more realistic

Process of Animation

- Planning
- Implementation.