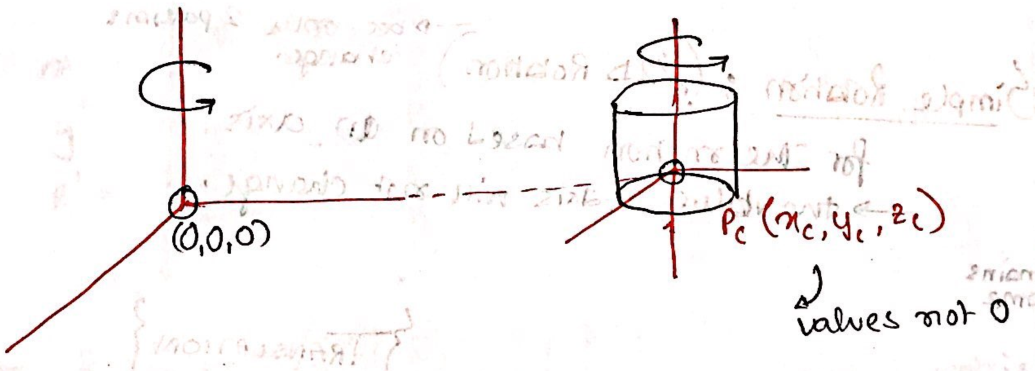
{Geometric Transformation [Motion]}

Classification (Based on Center Rotation)

- IDEA → Revolution → Geometric Transformation → $P_c(0,0,0)$ (center of rotation global)
- IDEA → rotation → Coordinate Transformation / modelling transformation (center of rotation local)



{Geometric Transformation}

Classification: (Based on uniformity of scaling)

scaling → size, (small kora, big kora)

- Rigid body motion → no change in shape (uniform)
- Affine motion → shape might change (non-uniform)

Components of motion of transformation:

- rotation — R
- translation — T
- scaling — S

$\odot(x, y)$

$\odot(x, -y)$

Rotation matrix

resulting co-ordinates.

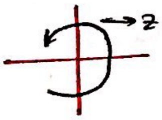
$$\begin{vmatrix} x' \\ y' \\ z' \end{vmatrix} = R \cdot \begin{vmatrix} x \\ y \\ z \end{vmatrix}$$

coordinates before rotation

Rotation can be based on 3 axis's
$\rightarrow (x, y, z)$

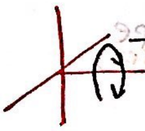
<u>Simple Rotation</u> : (//2D Rotation) → bec only 2 params change.

for the motion based on an axis
→ the value of axis will not change.

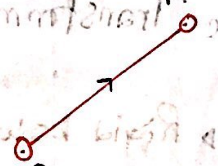


→ z remains same

{TRANSLATION}



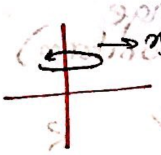
→ x remains same



P'

P

$(P' = P + T)$



→ xy remains same.

Can also be written as →

$x' = x + t$

$y' = y + t$

$z' = z + t$

$P'.x = P.x + t_x$

$P'.y = P.y + t_z$

$P'.z = P.z + t_z$

translation is very simply achieved by adding or subtracting.

Scanned with CamScanner

add / subtract

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

$$\begin{matrix} x' = \\ y' = \\ z' = \end{matrix} \begin{bmatrix} x \pm t_x \\ y \pm t_y \\ z \pm t_z \end{bmatrix}$$

for 3 axised Rotation.

→ we can mutiply by each rotation (based on axis) matrix and generate result

OR → we can combine all 3 of 'em to make a composite matrix and multiply with it to generate result.

→ 3×3 can only be used to +/−

→ 4×4 " be used for ×/÷ → $\underbrace{(x, y, z, w)}$

↳ {Homogeneous Coordinate System}

3D → 3value rep → Cartesian

3D → 4 valued representation → Homogeneous

(x, y)

$(x, -y)$

When 4^{th} value $= 1$
↳ Cartesian & Homogeneous same

$(x, y, z, w) = (x', y', z')$

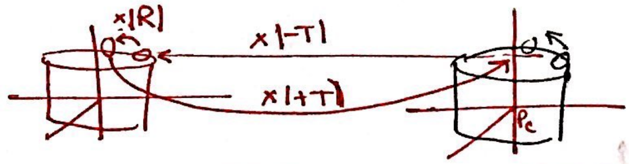
$\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}, 1\right) \rightarrow$ equals

So, $x/w = x'$ | $y/w = y'$ | $z/w = z'$ | $\frac{w}{w} = 1$

Homo + Cartesian $(15, 4, 6, 1)$ $=$ Truly Homo $(30, 8, 12, 2)$ $=$ Truly Cartesian $(15, 4, 6, 0)$

↓

Cartesian but rep as the form of Homogeneous.



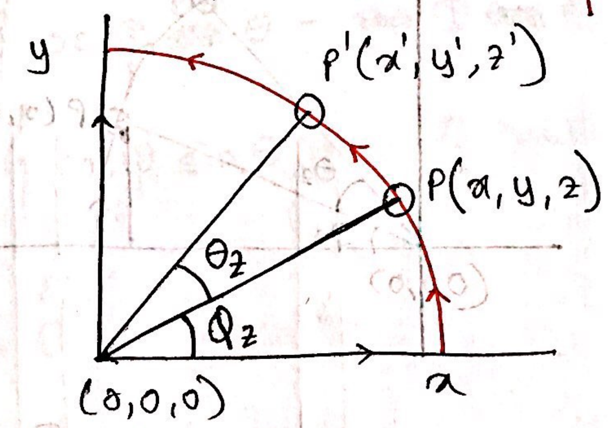
$$|P'| = |+T| \cdot |R| \cdot |-T| \cdot |P|$$

- $|+T|$ → Bring to local origin
- $|R|$ → Rotate → Geometric Rotation
- $|-T|$ → Translate to origin
- Bring to local origin, Translate to origin → Coordinate translation

{Mathematics Of Rotation} → from +ve z view

{ counter clkwise = +
clkwise = −

→ for −ve z view vice versa



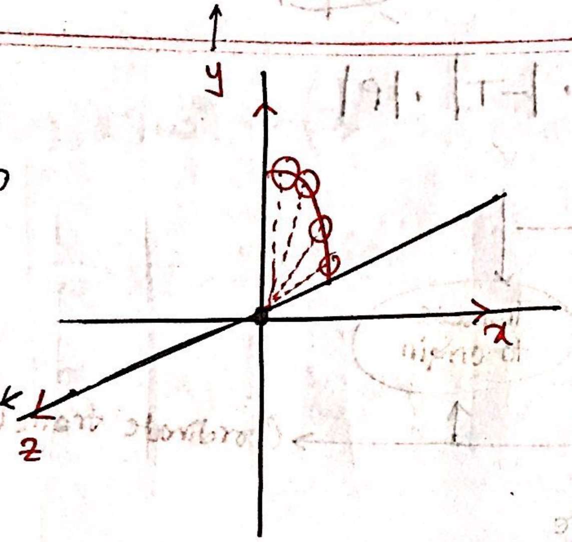
Right hand Rule → for all rotations (axis)

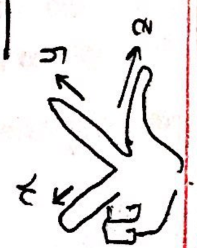
→ one +ve will go Right

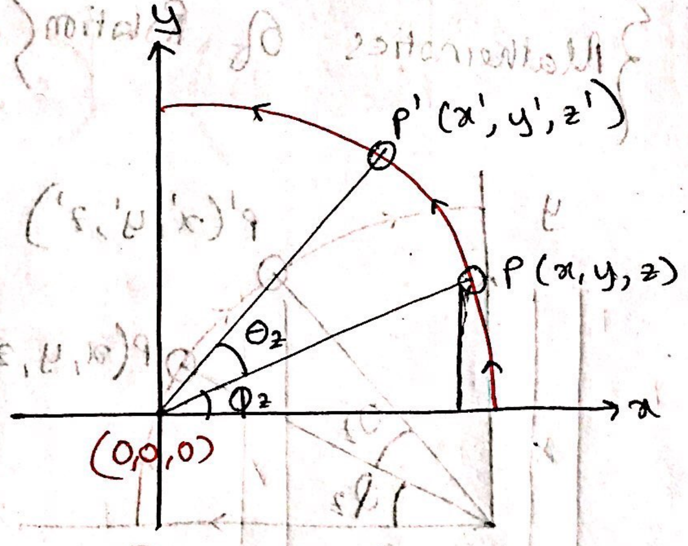
→ the one +ve will go up

z - goes to up

y - goes to Right







$$\left.\begin{aligned} x &= r\cos\phi \\ y &= r\sin\phi \end{aligned}\right\} \longrightarrow P$$

$$\left.\begin{aligned} x' &= r\cos(\theta+\phi) \\ y' &= r\sin(\theta+\phi) \end{aligned}\right\} \longrightarrow P'$$

and,

$$x = r\cos\varphi$$
$$y = r\sin\varphi$$ — (i)

$$x' = r\cos(\varphi+\theta)$$
$$y' = r\sin(\varphi+\theta)$$ — (ii)

since movement based on z axis. ← $z' = z$

$$x' = r\{\cos\varphi \cos\theta - \sin\varphi \sin\theta\}$$

$r\cos\varphi = x$
$r\sin\varphi = y$

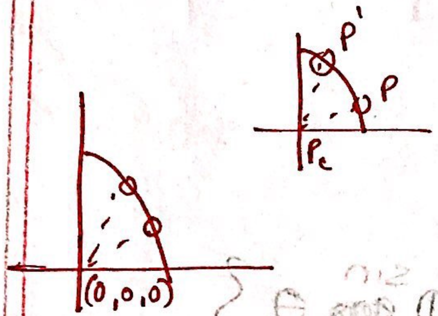
$$y' = r\{\cos\varphi \sin\theta + \sin\varphi \cos\theta\}$$

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$ — (iii)

$z = z'$

So, $\underline{\underline{3 \times 3}}$ representation:

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



→ (1) translate to origin
(2) Rotate
(3) bring back to local origin.

$$\begin{vmatrix} x' \\ y' \\ z' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & x_c \\ 0 & 1 & 0 & y_c \\ 0 & 0 & 1 & z_c \\ 0 & 0 & 0 & 1 \end{vmatrix} \times \begin{vmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \times \begin{vmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{vmatrix} \times \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix}$$

$\times$

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & (-x_c\cos\theta + y_c\sin\theta) \\ \sin\theta & \cos\theta & 0 & (x_c\sin\theta - y_c\cos\theta) \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

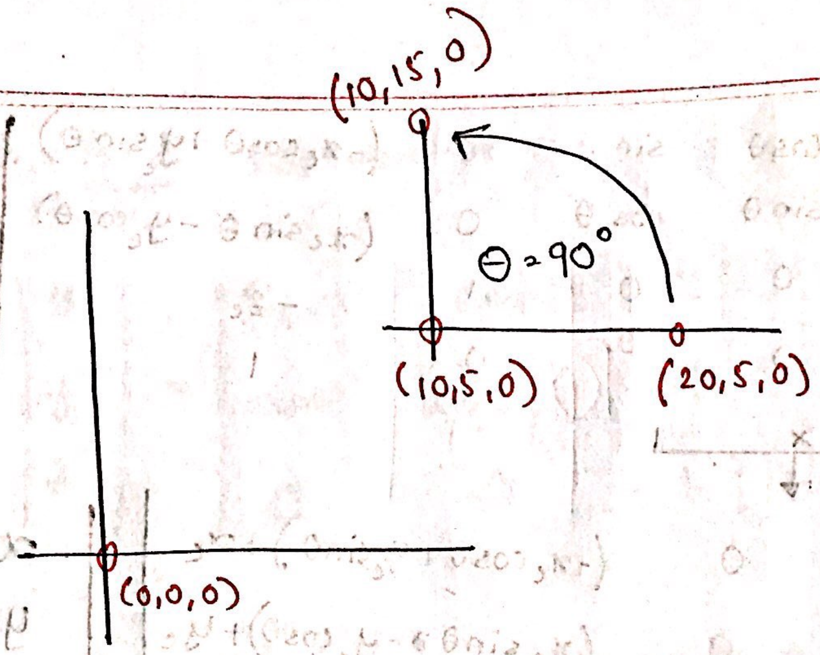
Scanned with CamScanner

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

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 & (-x_c\cos\theta + y_c\sin\theta) + x_c \\ \sin\theta & \cos\theta & 0 & (x_c\sin\theta - y_c\cos\theta) + y_c \\ 0 & 0 & 1 & -z_c + z_c = 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \Big| \; x\cos\theta - y\sin\theta$$

→ Not needed further simplification :



Find that $x', y', z' = \{10, 15, 0\}$

Prove ?

the order has to be swapped for this

$$|P'| = |T| \cdot |R_{\theta x}| \cdot |R_{\theta y}| \cdot |R_{\theta z}| \cdot |-T| \cdot |P|$$

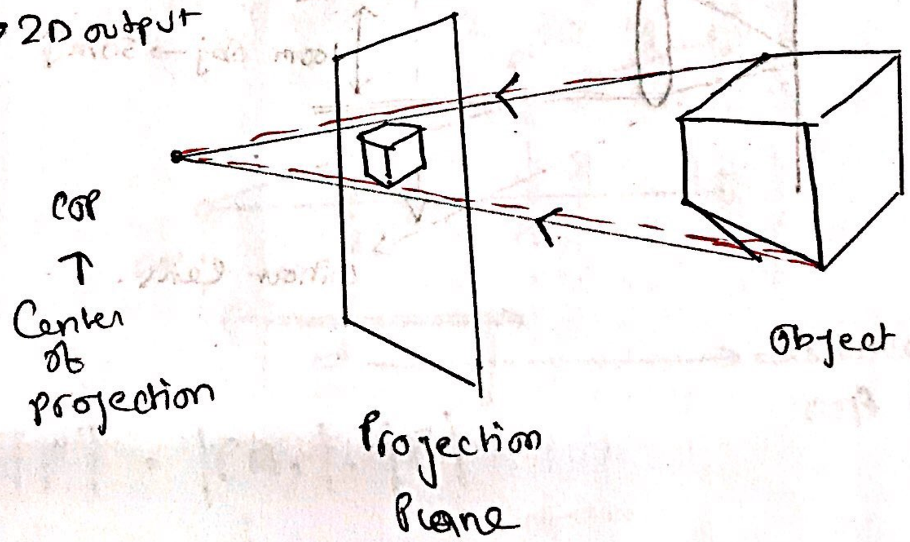
also, for the other way around

$$|P| = |T| \cdot \underbrace{|R_{\theta z}|}_{\text{3rd}} \cdot \underbrace{|R_{\theta y}|}_{\text{2nd}} \cdot \underbrace{|R_{\theta x}|}_{\text{happens 1st}} \cdot |-T| \cdot |P'|$$

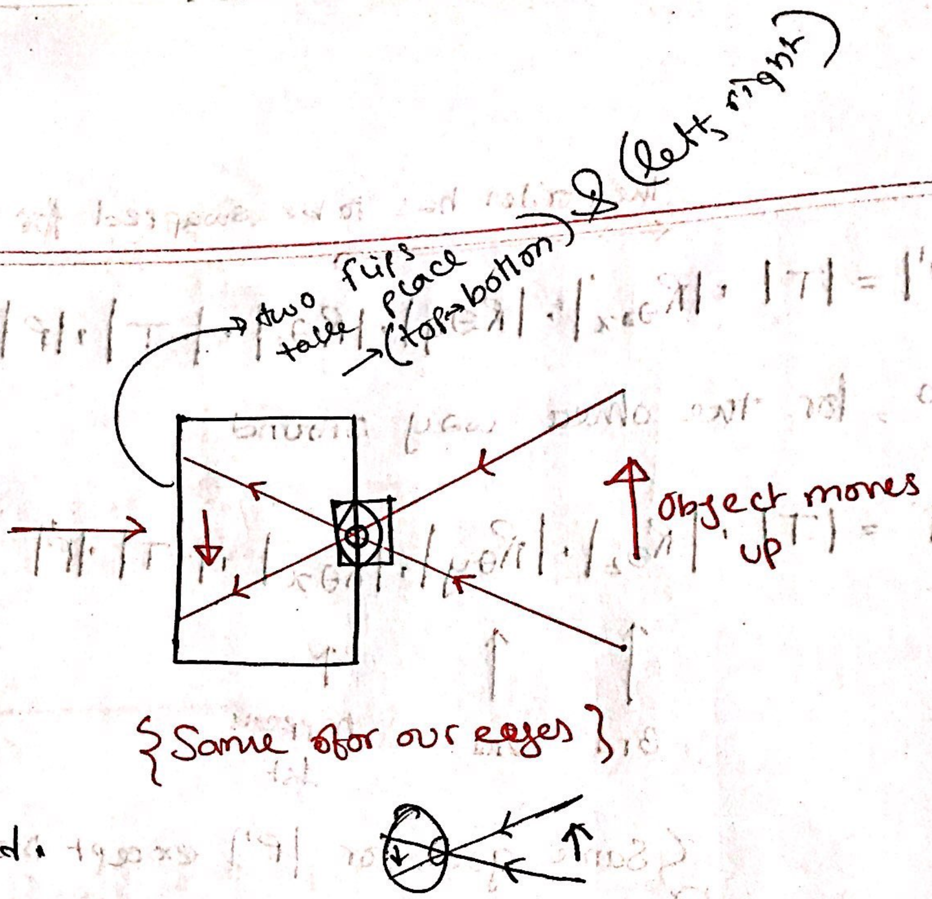
{Same goes for $|P'|$ except the order of the rotations matrix are different}

Projection:

$$\xrightarrow{\text{3D input}} \boxed{Pr} \longrightarrow \text{2D output}$$

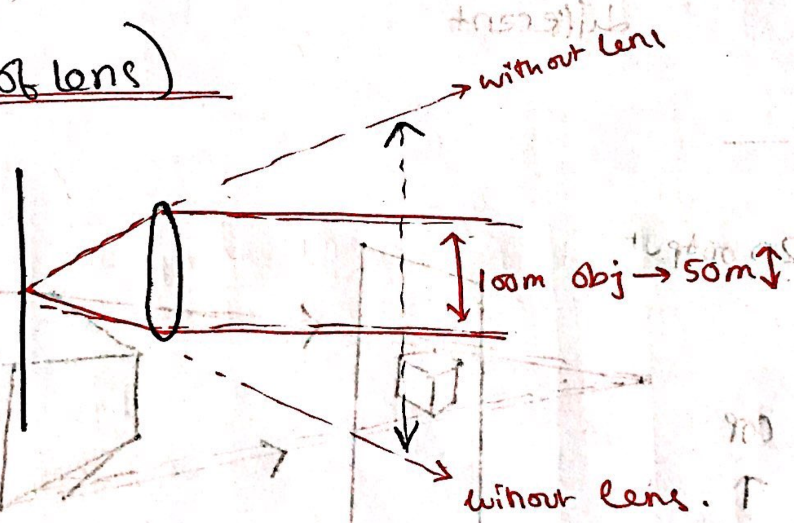


Camera :

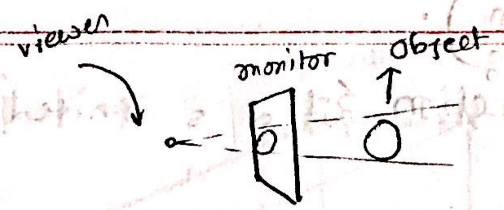


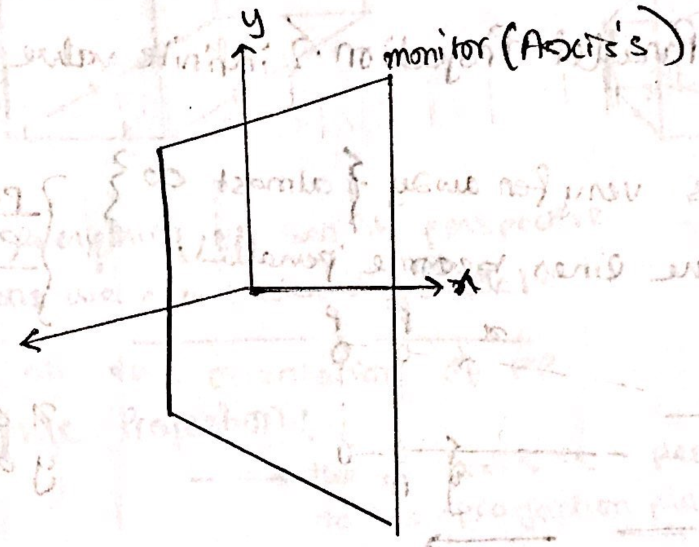
The Idea is that since display is behind COP the object looks flipped.

(use of lens)

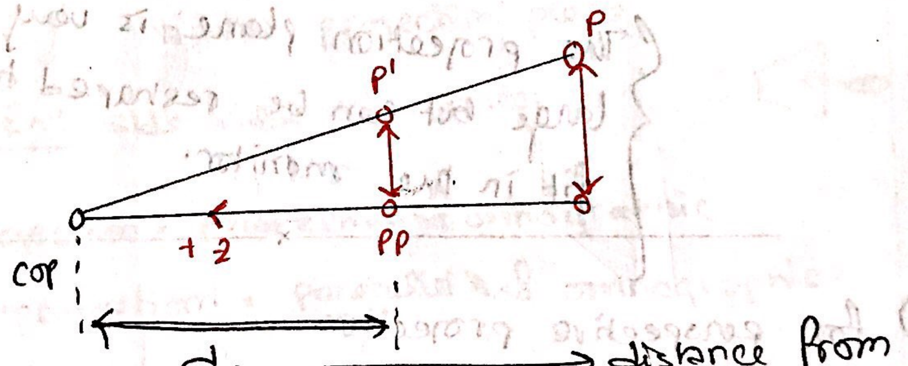


SEE PINHOLE CAMERA





(Simpler version)



d → distance from cop to projection.

$$|P'| = |Per| \cdot |P|$$

$|P'|$ → Projection

$|Per|$ → Projection matrix.

$|P|$ → object

(Classification):

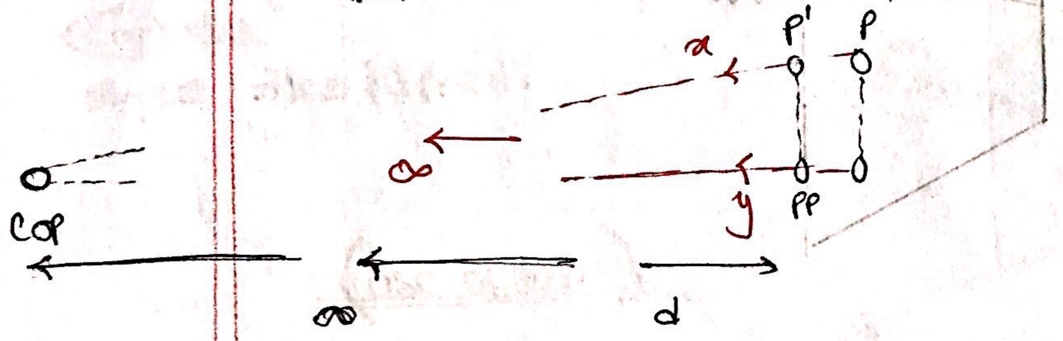
Based on d :

(i) Perspective Projection {if d is limited}

(ii) Parallel Projection. {infinite value of d}

(ii)

when cop is very far away {almost ∞} then the lines become parallel. } parallel projection



x and y are parallel

{ The projection plane is very large but can be reshaped to fit in the monitor.

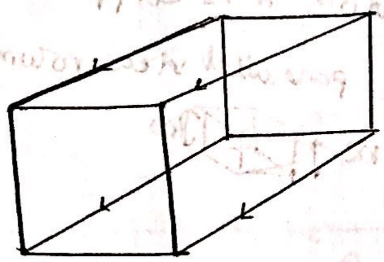
(i) for perspective projection

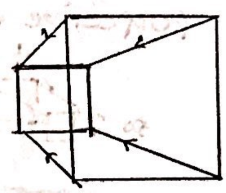
the larger d the smaller the object becomes.

① Persp —+→ ① Ortho

② Parallel ←+→ ② Oblique

Jah





in real world everything we see it perspective.
But using lens we can make it parallel.

X

→ Based on the orientation of PP.

① Orthographic Projection:

→ the z axis is perpendicular to the projection plane.

② Oblique projection:

→ the z axis is not perpendicular to the projection plane.

Ex: side view of monitor

projections

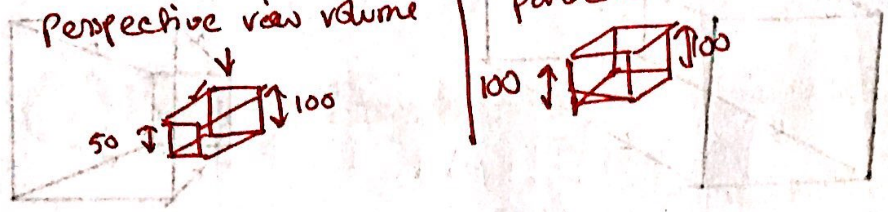
perspective projection = perspective & orthographic

orthographic projection = parallel & orthographic.

Oblique projection = parallel & oblique.

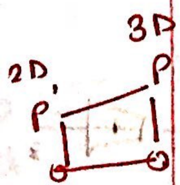
X There is no ~~parallel~~ perspective + oblique projection.

If object is within view-volume then it is shown, otherwise it is clipped.



~ (Mathematics of Projection) ~

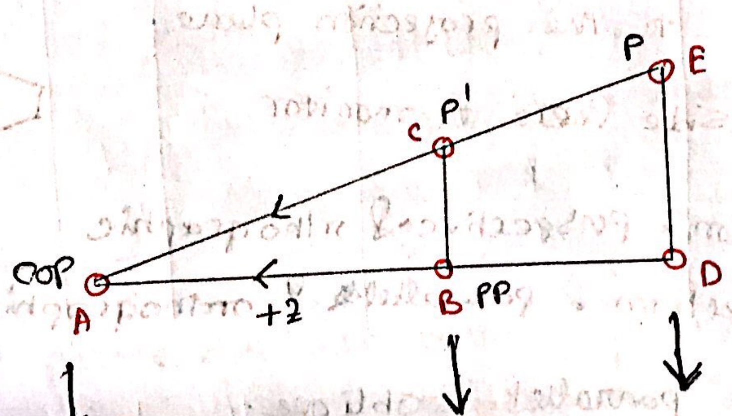
P' → is the final pixel or monitor coordinate.



Calculation and view are determined in 3D for the object | And PP is used to project the 2D pixel.

ABC → Δ one triangle

ADE → another triangle.

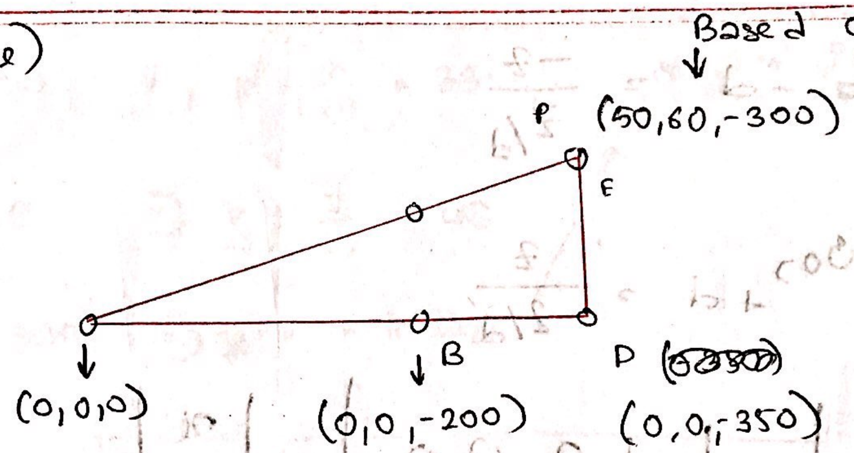


{These are 3 independent entities} ~~for~~

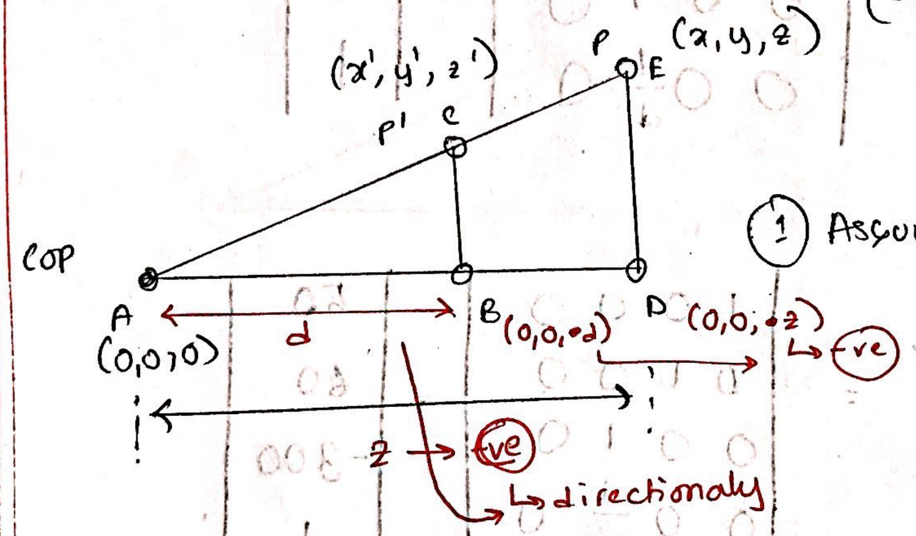
So Relation.

$$\frac{BC}{AB} = \frac{DE}{AD} \quad \cdots (i)$$

(Example)



$(z \,\&\, z')$ are $-ve$



1. Assuming origin at COP!

d & z is directionally negative

$$\therefore \Rightarrow \left\{ \frac{y'}{d} = \frac{y}{z} \right\} \qquad \text{(BC, DE)}$$

$$\therefore y' = \frac{yd}{z} = \frac{y}{z/d} \quad \text{--- (ii)}$$

Since x, y, z ration of the liner are same.

Same for x

$$\frac{x'}{d} = \frac{x}{z}$$

$$\therefore x' = \frac{x}{z/d} \quad \text{--- (iii)}$$

$$-z' = -d = \frac{-z}{z/d}$$

$$\text{so } +d = \frac{z}{z/d}$$

$$\begin{vmatrix} x' \\ y' \\ z' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{vmatrix} \cdot \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix}$$

for B(0,0,-200)

E(50,60,-300)

$$\begin{vmatrix} x'' \\ y'' \\ z'' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{-200} & 0 \end{vmatrix} \cdot \begin{vmatrix} 50 \\ 60 \\ -300 \\ 1 \end{vmatrix}$$

$$\begin{vmatrix} x'' \\ y'' \\ z'' \\ W \end{vmatrix} = \begin{vmatrix} 50 \\ 60 \\ -300 \\ 3/2 \end{vmatrix}$$

$w = z/d \longrightarrow 4^{th}$ value.

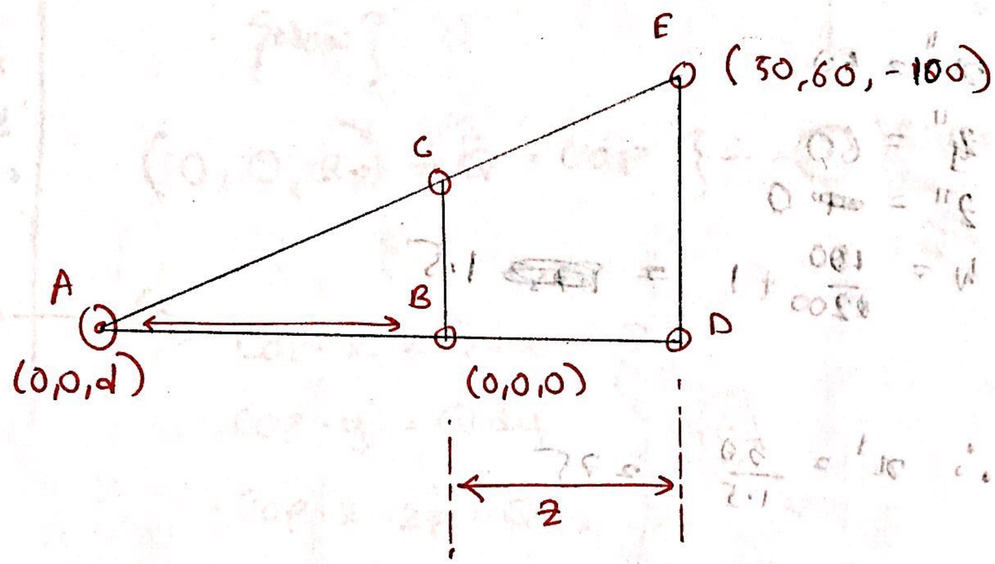
$x'' = 50 \Rightarrow x' = \frac{x''}{w} = 33.3$

$y'' = 60 \Rightarrow y' = \frac{y''}{w} = 40$

$z'' = -300 \Rightarrow z' = -200$

since $z' = z$ and $z' = -200$ so correct

ORIGIN at PP



$$\frac{y'}{d} = \frac{y}{d-z}$$ {since z is inherently -ve.}

$$\Rightarrow y' = \frac{y}{\frac{(d-z)}{d}} = \frac{y}{(1 - z/d)}$$

$$x' = \frac{x}{(1 - z/d)}$$

~~$z' = \frac{z}{(1 - z/d)}$~~ $z' = 0$ } since origin at PP

$\circ(7,-1)$

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

$\Rightarrow$ $x'' = 50$

$y'' = 60$

$z'' = 0$

$h = \dfrac{100}{200} + 1 = 1.5$

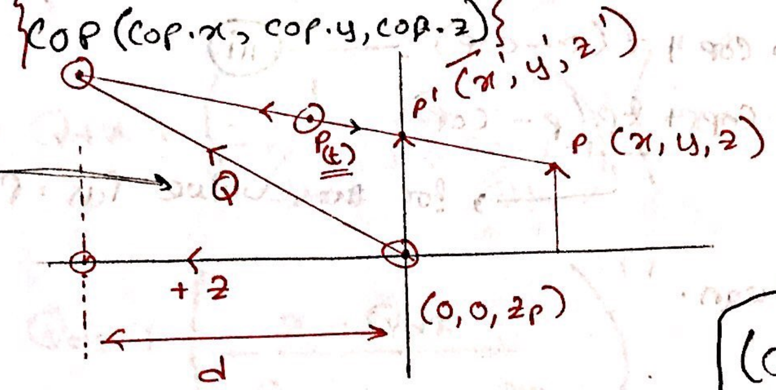
$\therefore x' = \dfrac{50}{1.5} = 2.75$

$y' = 60/1.5 = 40$

$z' = 0 = 0$

After Mid

(lecture 3)



Q : (Qdx, Qdy, Qdz)

{Previously}
$(0,0,z_p) + (0,0,d) = COP$

So,
$COP.x - 0 = Qdx$
$COP.y - 0 = Qdy$
$COP.z - z_p = Qdz$

{Now}

$(0, 0, z_p) + \bar{Q} = COP$ } --- (i)

Since directed so,
$Q = COP - (0,0,z_p)$

So, $COP.x = Qdx$
$COP.y = Qdy$
$COP.z = z_p + Qdz$
} — (ii)

Suppose (P to COP, a parametric line)
→ then $P(t)$ is any point on the line.

Hence,

we'll use this, → $P_{(t)} = COP + t(P - COP)$ ---- (iii)
OR $= P + t(COP - P)$

So, $P_{(t)}$ will be equals P' for a certain value of t.

For a certain value $P(t) = P'$, for $t = t_p$.

$$P(t) = COP + t\,(P - COP) \quad \text{--- (iii)}$$

$$P' = COP + \underbrace{t_p}_{\text{for this value } P(t) = P'}\,(P - COP)$$

For (IInd) eqn.

$$x' = Q_{dx} + t_p\,(x - Q_{dx})$$

$$y' = Q_{dy} + t_p\,(y - Q_{dy})$$

$$z' = Q_{dz} + z_p + t_p\,(z - Q_{dz} - z_p)$$

--- (iv)

we'll select t_p from z' and put it in x' and y'

$$t_p = \frac{\cancel{z'} - \cancel{z_p} - Q_{dz}}{z - z_p - Q_{dz}} = t_p$$

Since $z' = z_p$

P' all values of z same

(z_p)

$$t_p = \frac{-Q_{dz}}{z - z_p - Q_{dz}}$$

divided by $-Q_{dz}$

$$t_p = \frac{1}{-\frac{z}{Q_{dz}} + 1 + \frac{z_p}{Q_{dz}}} \quad \text{--- (V)}$$

(Now putting ~~z~~ z_p (of eqn v) in x') →

$$x' = Q_{dx} + \left(\frac{1}{\frac{-z}{Q_{dz}} + 1 + \frac{z_p}{Q_{dz}}}\right)\left(x - Q_{dx}\right)$$

$$x' = Q_{dx} + \left(\frac{x - Q_{dx}}{-z/Q_{dz} + 1 + \frac{z_p}{Q_{dz}}}\right)$$

$$= \frac{-z\frac{dx}{dz} + Q_{dx} + z_p\frac{dx}{dz} + x - Q_{dx}}{-z/Q_{dz} + 1 + \frac{z_p}{Q_{dz}}}$$

$$= \frac{-z\frac{dx}{dz} + \cancel{Q_{dx}} + z_p\frac{dx}{dz} + x - \cancel{Q_{dx}}}{-z/Q_{dz} + 1 + \frac{z_p}{Q_{dz}}}$$

= R.H.S

→ x, z, Q_{dz}, z_p, d_x Know valves.

$$\boxed{x' = \frac{x - z\frac{dx}{dz} + z_p\frac{dx}{dz}}{-z/Q_{dz} + 1 + z_p/Q_{dz}}} \quad \text{--- (vi)}$$

Similarly for $y' \longrightarrow$

$$\boxed{y' = \frac{y - z\frac{dy}{dz} + zp\frac{dy}{dz}}{\left\{-z/Qdz + 1 + zp/Qdz\right\}}} \longrightarrow \text{vii}$$

ω

and

$$z' = zp \text{ --- viii}$$

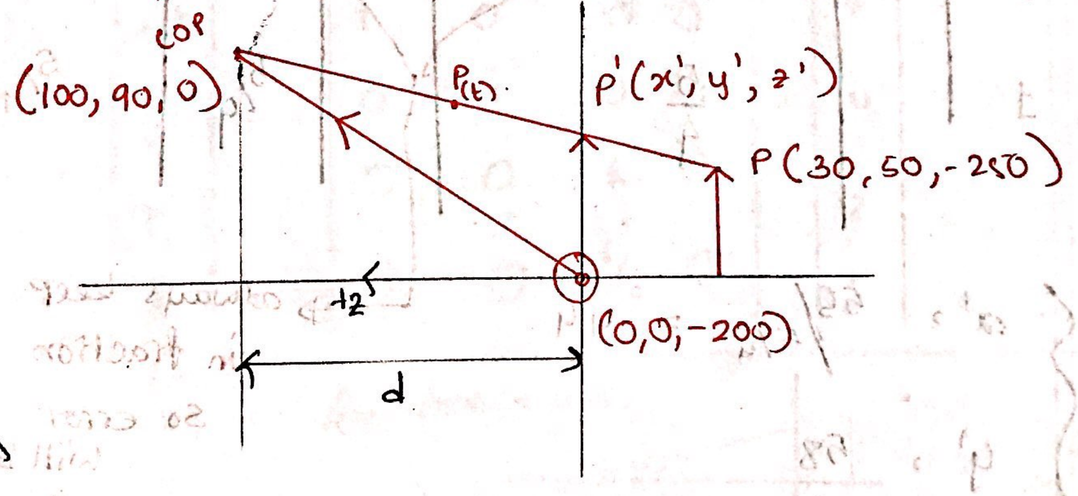
$$z' = \frac{zp\left(-z/Qdz + 1 + zp/Qdz\right)}{\left(-z/Qdz + 1 + zp/Qdz\right)}$$

$$= \frac{-z\frac{zp}{Qdz} + zp\left(1 + \frac{zp}{Qdz}\right)}{\left(-z/Qdz + 1 + zp/Qdz\right)} \longrightarrow \text{viii}$$

Now, like previous problems.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{dx}{dz} & z_p\frac{dx}{dz} \\ 0 & 1 & -\frac{dy}{dz} & z_p\frac{dy}{dz} \\ 0 & 0 & -\frac{z_p}{Qd_z} & z_p\left(1+\frac{z_p}{Qd_z}\right) \\ 0 & 0 & -\frac{1}{Qd_z} & 1+\frac{z_p}{Qd_z} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

let say Now (Problem)



So, values

$Qd_x = 100$

$Qd_y = 90$

$Qd_z = 200$

$z_p = -200$

Same matrix stands

$$= \begin{bmatrix} 1 & 0 & -\frac{1}{2} & -100 \\ 0 & 1 & -\frac{9}{20} & -90 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{200} & 0 \end{bmatrix} \cdot \begin{bmatrix} 30 \\ 50 \\ -250 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 30 + 125 - 100 \\ 50 + 112.5 - 90 \\ -250 \\ \frac{5}{4} \end{bmatrix} = \begin{bmatrix} 55 \\ 72.5 \\ -250 \\ 5/4 \end{bmatrix} \text{ OR } \begin{bmatrix} 55 \\ 145/2 \\ -250 \\ 5/4 \end{bmatrix}$$

always keep in fraction so error will be less.

divide by $w = 5/4$

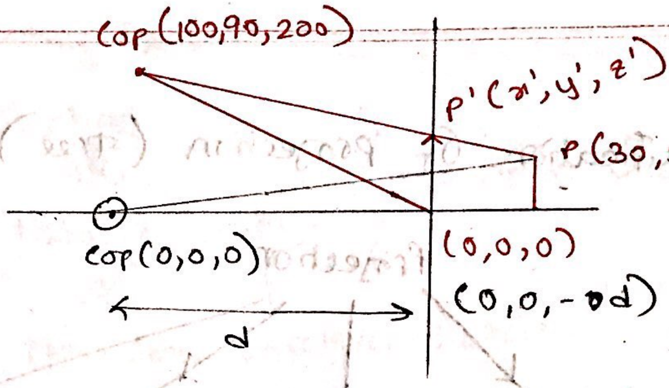
$x' = 55/(5/4) = 44$

$y' = 58$

$z' = -200$ } z_p & z' same so correct

$1 = 1$

Now,



Simplification : (for black line) {origin at cop}

$Q_{dx} = 0$

$Q_{dy} = 0$

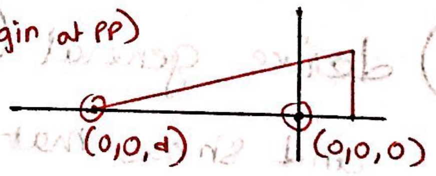
$Q_{dz} = d$

$z_p = -d$

So Now,

$$\begin{vmatrix} x' \\ y' \\ z' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{vmatrix} \cdot \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix}$$

for (origin at PP)



$Q_{dx} = 0$

$Q_{dy} = 0$

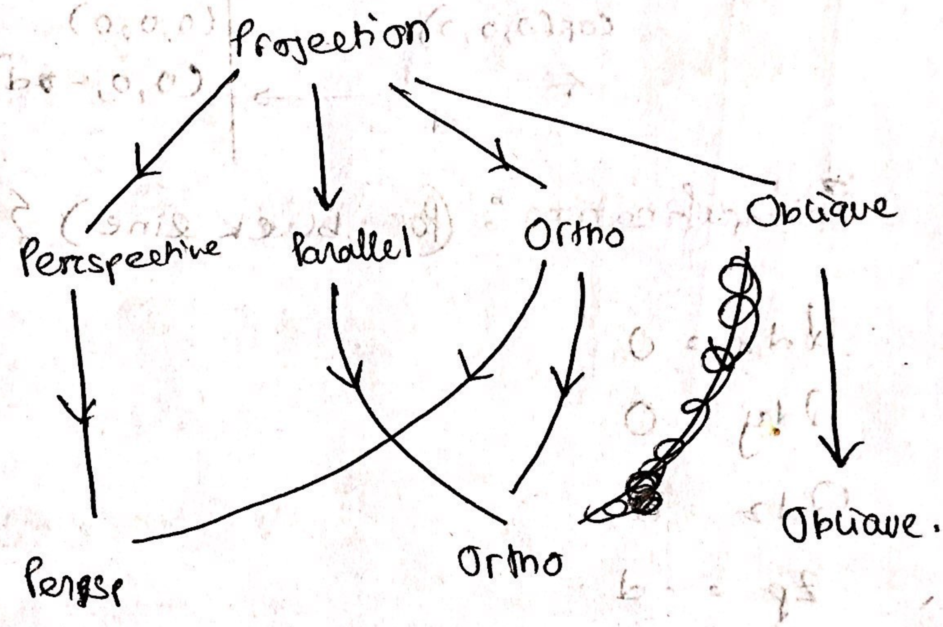
$Q_{dz} = d$

$z_p = 0$

$$\begin{vmatrix} x' \\ y' \\ z' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/d & 1 \end{vmatrix} \cdot \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix}$$

Questions

① Classification of projection (tree)



(2) derive (4x4) projection matrix while / by origin at cop.

(3) derive (4x4) projection matrix while origin on projection plane.

(4) derive general (4x4) matrix. and show that other projection matrix's can be represented using this by simplification.

(5) draw projection diagram. (all 3) → origin at cop
" " PP
cop at differ plane.

Scanned with CamScanner

(Quiz) → Only math.

→ Rotation
OR
Today's math.
OR
Non-zero rotation matrix.

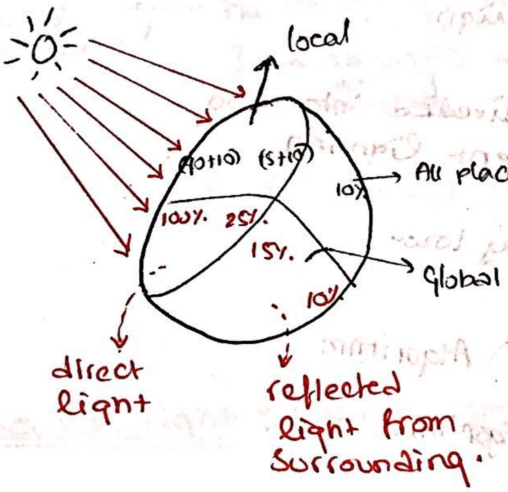
→ open book

Scanned with CamScanner

(*) Need to see Mohi's notes

CHAPTER:

Illumination & Shading:



Ø brightness variation can be caused due to surface orientation this phenomena is known as shading.

→ The difference that can show shape is shading

→ but light blocking causes shadow

Lighting Model:

this model has no difference in source light but light from reflection

Global light Model

Ø acknowledges all reflection and integrates them.

Ø light from source + light from reflection.

Ø very complex calculation.

local light Model.

Ø Divides source into two

Ambient light

Ø a brightness value that comes from and on every side

Ø Needs to be add.

Ø for slight difference the shading doesn't happen

Ø cannot create shape.

Source light

Ø light coming from the source Ray direction.

→ Per pixel accuracy

Global model → one source

→ calculates multiple deflection to calculate the brightness of a point

→ slower

→ accuracy high

local model → source is divided into two

→ uses ambient lighting

→ fast

→ accuracy low

(Recursive Ray-Tracing) Algorithm

(Phong lighting model) Algorithm

(Local Lighting Model)

(1) Ambient light:

$I_a \rightarrow$ intensity of amb light

absorption coefficient of ambient light ← $K_a \rightarrow I/I_a$ (Ratio of reflected light vs incident light)

$I \rightarrow$ amount of light reflecting from a point or intensity

Higher K_a → brighter object.

$$I = K_a I_a \text{ --- (i)}$$

→ higher K_a.

→ smaller K_a

(2) Source light: (directly coming from source and has a direction)

→ Lambert's lighting model (Rough surface)

→ Phong " " (Even surface)

{Lambert's lighting Model} (Diffuse Reflection)

↳ means reflection from rough surface.

Intensity of reflected light ← $I = K_d I_s \cos\theta$ --- (ii)

$= K_d I_s \hat{N} \cdot \hat{L}$ --- (iii)

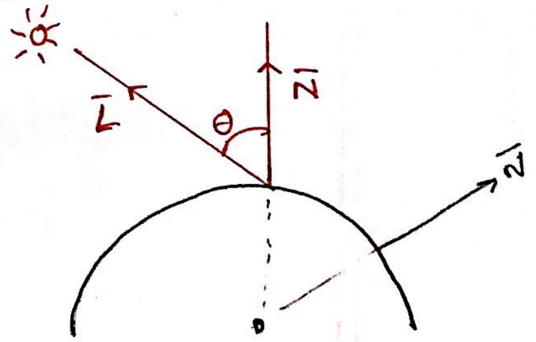
→ $I = K_d I_s \cos\theta$ --- (ii)

→ $I = K_d I_s \hat{N} \cdot \hat{L}$ --- (iii)

$I_s \rightarrow$ intensity of source light

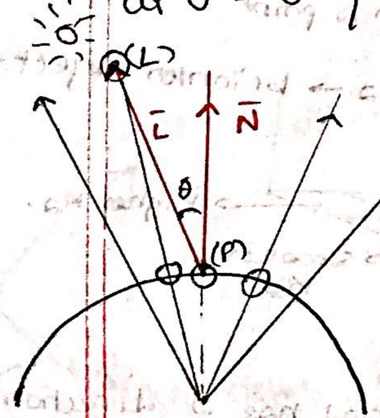
$K_d \rightarrow$ reflecting/absorption coefficient of rough object

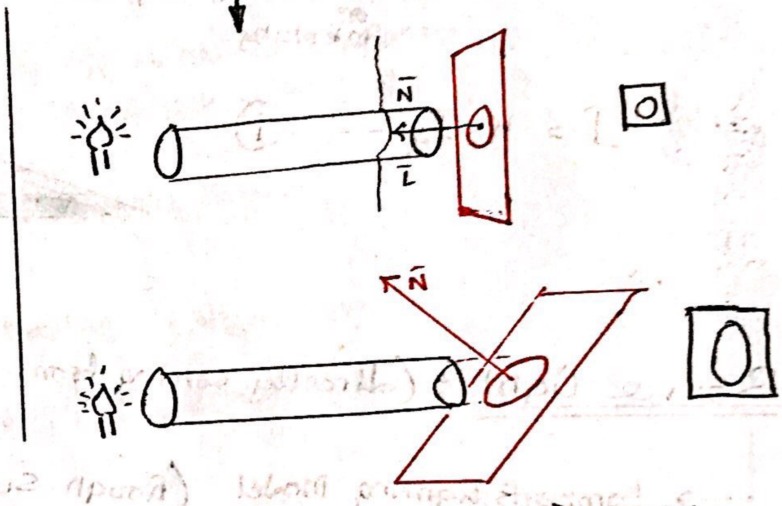
$\theta \Rightarrow$ angle between surface normal and angle of incidence.



$\bar{N} \rightarrow$ direction of liane from center to surface.

at $\theta = 0$ {the point is very bright.?}





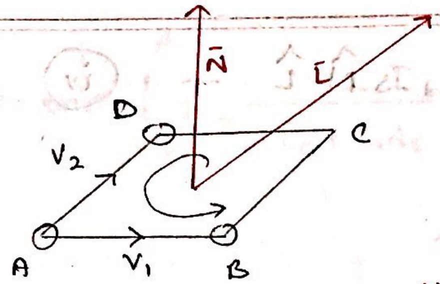
as angle increases the light per unit decreases.

Ø for circle:

$\bar{N}$ = (point on surface − center) {need to be normalized}

$\bar{L} = L - P$ {

I = cannot be −ve, the minimum value is Zero.

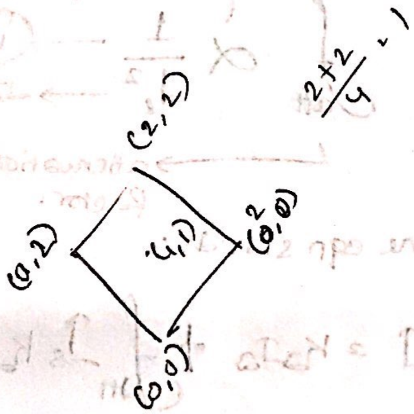


$N = V_1 \times V_2$

$$\bar{N} = \begin{vmatrix} i & j & K \\ V_1 \cdot x & V_1 \cdot y & V_1 \cdot z \\ V_2 \cdot x & V_2 \cdot y & V_2 \cdot z \end{vmatrix}$$

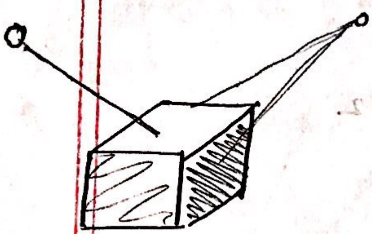
OR

$$\bar{N} = \begin{vmatrix} dx & dy & dz \\ V_0 \cdot x & V_0 \cdot y & V_0 \cdot z \\ V_1 \cdot x & V_1 \cdot y & V_1 \cdot z \end{vmatrix}$$



Lambertian shading.

$$I = K_a I_a + K_d I_s \hat{N} \cdot \hat{L} \quad \text{--- (iv)}$$



local light model will always create 3D model but because of the positioning of source we might or might not be able to differentiate between the 3D model.

(Attenuation of Light)

{ the decrease in brightness of light due to travelled path }

(1) Light Source attenuation

(2) Atmospheric "

→ object can be bright or less bright due to the distance of the obj from the light source. Due to attenuation.

$$f_{att} \propto \frac{1}{d_e^2} \quad \text{--- (v)}$$

inversely proportional.

d_e^2 → distance from object to light source.

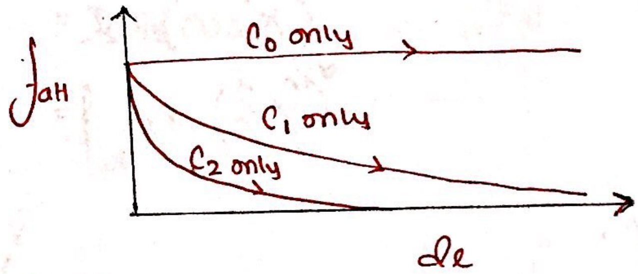
f_{att} → attenuation factor. → value [0 to 1]

So the eqn stands:

$$I = K_a I_a + f_{att} I_s K_d \hat{N} \cdot \hat{L} \quad \text{--- (vi)}$$

↳ multiplication factor.
close → brighter

$$f_{att} = \min\left(1, \frac{1}{c_0 + c_1 d_e + c_2 d_e^2}\right)$$



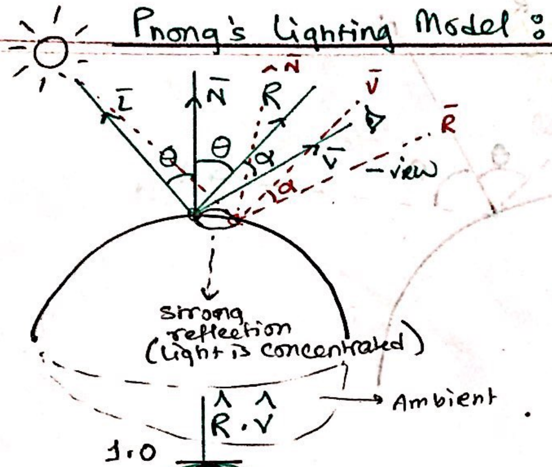
Thursday (Quiz 2)

→ non zero center of rotation.

→ Rotation

→ Projection.

Phong's Lighting Model : (Specular Reflection)



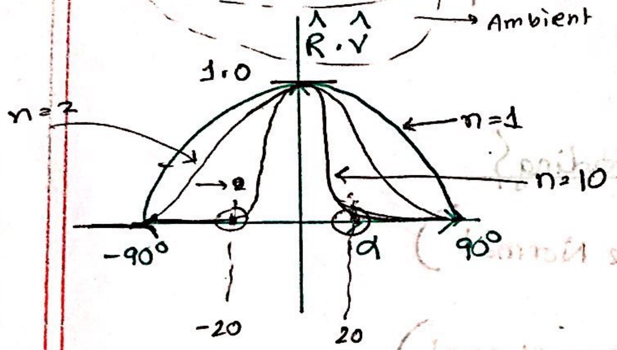
shiny ness

$I = I_s K_s (\hat{R} \cdot \hat{V})^n$ ---- (i)

dot product is cos

$= I_s K_s (\cos^n \alpha)$ ---- (ii)

$= \int_{all} I_s K_s (\hat{R} \cdot \hat{V})^n$ ----



$\boxed{\text{⊖ when the view point changes the spot of strong reflection changes as well.}}$

$= \int_{all} \omega(\theta) I_s K_s (\hat{R} \cdot \hat{V})^n$

$\omega(\theta)$ = weight of intensity.

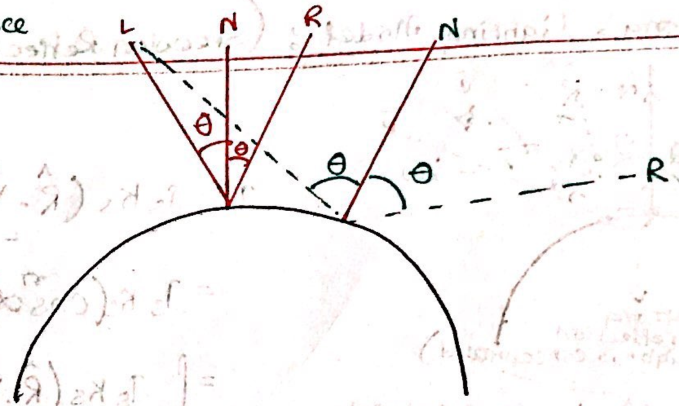
(accumulating everything)

local light model:

$$I = \underbrace{I_a K_a}_{\text{ambient}} + \int_{all} I_s \big(K_d \hat{N} \cdot \hat{L} + \underbrace{\omega(\theta) K_s (\hat{R} \cdot \hat{V})^n}_{\text{spot}} \big) \quad \text{---- (iii)}$$

R. is N and L dependant.

→ N and L er difference is same as N and R er diff.



{Types of Shading}

used in games.

1. Flat Shading : (Face Normal)
2. Smooth Shading : (Vertex Normal)

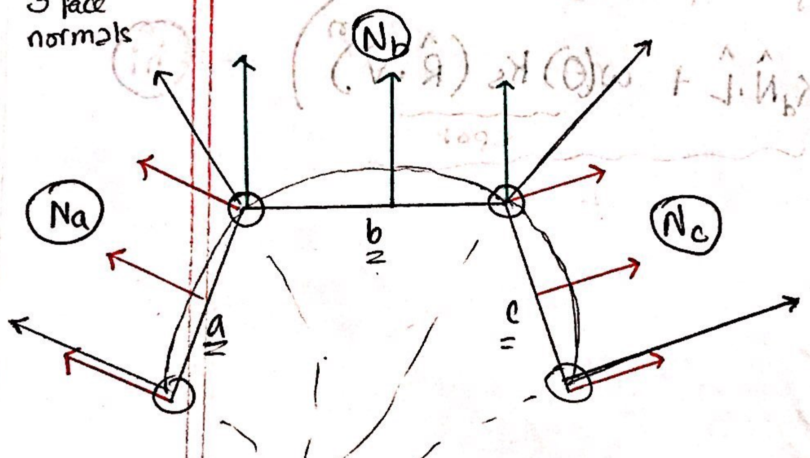
flat surface has N constant.

has different N all throughout the surface

→ Phong Shading (N-interpolation)

→ Gouraud " (I-interpolation)

This has 3 face normals

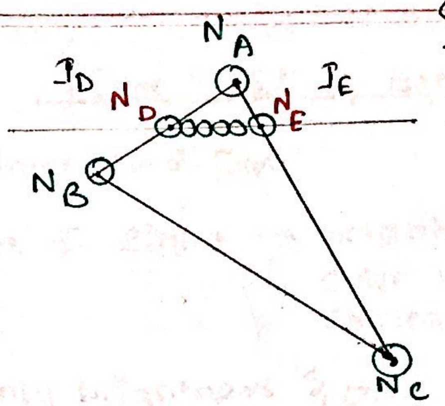


black ones are vertex normal.

→ flat ke virtually curve banaise.

Interpolation $t=0$, and $t=1$

→ finding out values between two extremes



Gouraud calculates I two times, and interpolates.

like I_D and I_E.

0.8 → 0.2

decreases

→ very blunt

→ very fast.

Phong shading calculates N_D and N_E and interpolates by calculating I at every point

→ very sharp

→ slow

Scanned with CamScanner

{Color Model of Light}

I → intensity of light.

3 parts of light → brightness
color
saturation (colorness)

I → only brightness {only black & white
diff grey levels}

In Real world

$$I \rightarrow I_R,\quad I \rightarrow I_G,\quad I \rightarrow I_B$$

{whoever has the highest intensity will dominate}

{color coming from source}

(1) Color → monitor → R, G, B $W = R+G+B$

(2) Pigment → paper print → $C = W-R$, $M = W-G$, $Y = W-B$

{sucking up the light}

$C = R+G+B - R$
$= G+B$

$M = R+B$

$Y = G+R$

for making monitor color

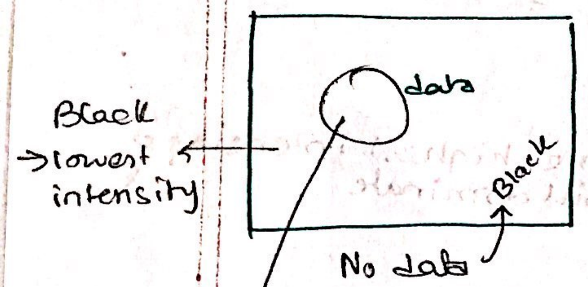
for making paper color.

RGB (obj that emit color)

1) Emission

2) Additive

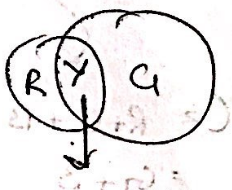
→ increase in color effective intensity increases



has to be brighter to be displayed

data↑, intensity↑

for overlap



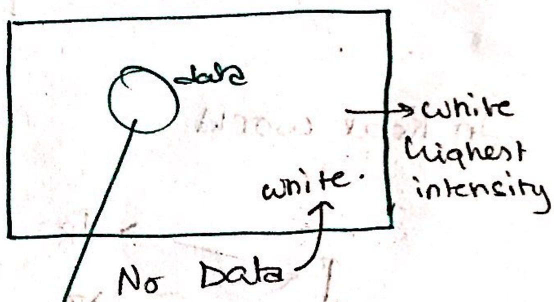
brighter

CMY (object that don't emit color but seen through reflection)

Reflection.

Subtractive

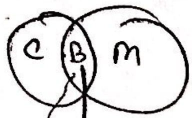
→ increase in color means effective decrease in intensity.



has to be darker than white to be visible.

data↑, intensity↓

for overlap.



darker.

→ $R+G+B-R-G$

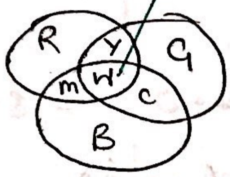
$= \underline{\underline{B}}$

Scanned with CamScanner

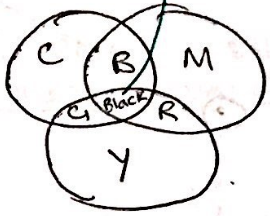
RGB   CMY

For,

additive and highest value so white



Subtractive and lowest value so black.



$$(R, G, B) \longrightarrow (C, M, Y)$$

$$(0.7, 0.6, 0.8) \xrightarrow{=} (0.3, 0.4, 0.2)$$

$(1-0.7)$ OR $(1-0.3)$

$(1-0.6)$ OR $(1-0.4)$

$(1-0.8)$ OR $(1-0.2)$

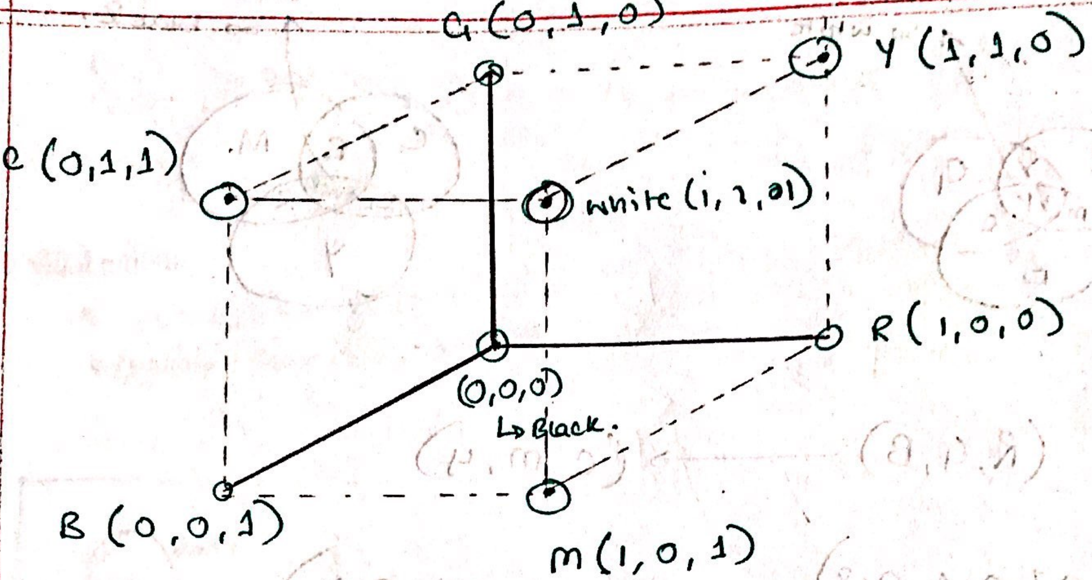
For Pure Red

$$(1, 0, 0) = (0, \underbrace{1, 1})$$

only red with high intensity.

common reflector red, so high red color

{RGB Color . Cube → 2^{24} = 16 million}



each axis is 8 bit (0 ~ 256)

RGB color → (16 million color)
$(2^8 \times 2^8 \times 2^8)$

{ for CMY Cube
diagonal opposites will be switched
(*) draw yourself.
→ Just switch polar opposites.

32 bit color { 4 billion color }
RGBA → Alpha / transparency.
CMYK
CMYB
↳ can be used to correctly produce real life object

(RGB) → theoretically float value (0~1)

(*) Good for hardware.

(*) No one uses RGB in software for these factors.

(0.8, 0.6, 0.2)

→ Brightness = ?

→ Color = ?

→ Saturation = ?

(for Representation)

(i) HSV / HSB

(ii) HLS / HSL

(used in television)

↑

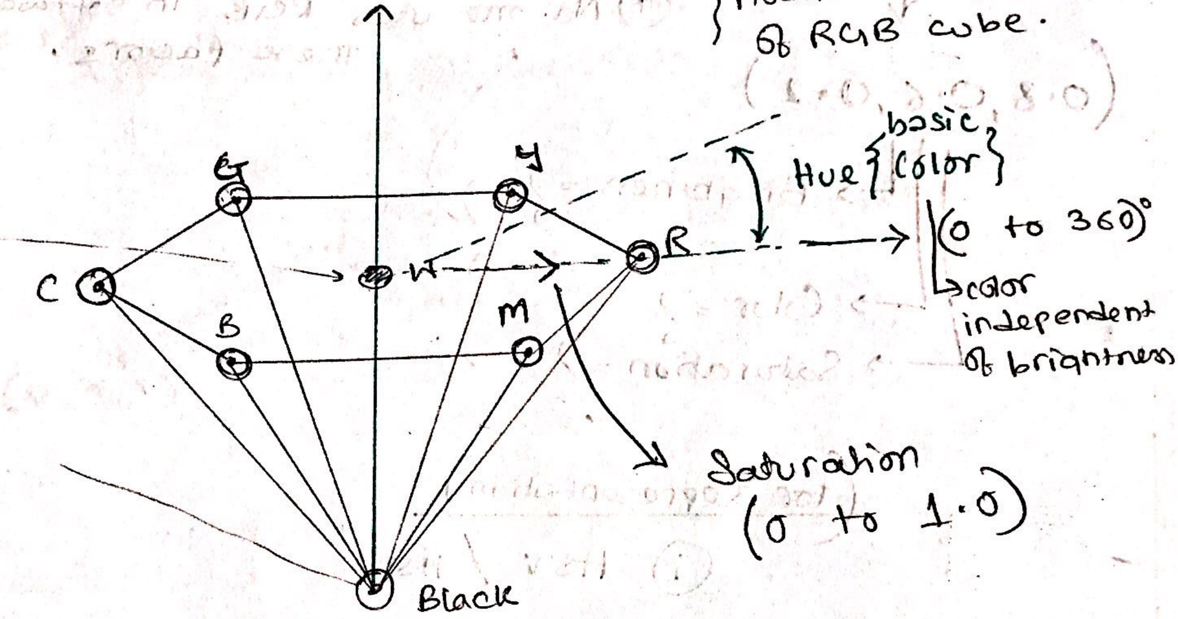
→ RGB model { backend has RGB }

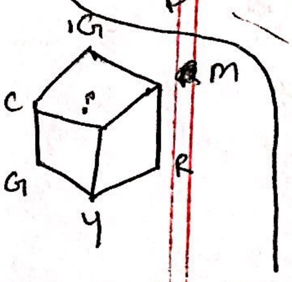
hardware.

✻ HSV / HSB Color Model:

Value / Brightness → V or B (0 to 1.0)

} modified form of RGB cube.





RGB

(0.8, 0.6, 0.2)
↕ ?
(0.4, 0.3, 0.1)

→ This two RGB values (color) is same — [The mutual proportion of the 3 values.

but upper one has higher brightness

(Relationship)

(RGB) $\xrightarrow[\text{conversion}]{}$ (HSV)

(0.8, 0.6, 0.2)
⇒ V = (0.8)
⇒ S = 0.75

input
- R = 0 to 1.0
- G = 0 to 1.0
- B = 0 to 1.0

(0.4, 0.3, 0.1) ⇒ V = (0.4)
⇒ S = 0.75

output =
- H = 0° to 360° → (Hue) → basic color.
- S = 0 to 1.0 → (Saturation)
- V or B = 0 to 1.0 → (Brightness)

X

<u>Code:</u>

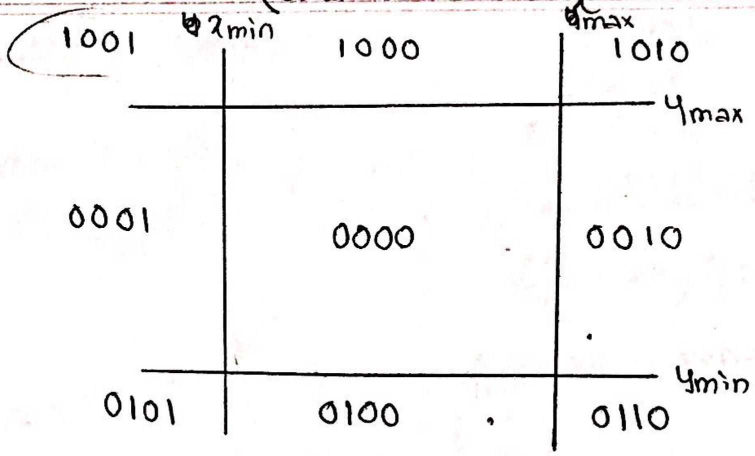
```
float H, S, V;   // global variable.
void RGB_to_HSV (float R, G, B)
    float M = Max (R, G, B);   // max value betn them
    float m = Min (R, G, B);   // min.
    float l = M - m;   // diff betn max & min
    float V = M;   // value or Brightness.
    if (V == 0) {
            S = 0;   // done for program safety.
    }
    else {
        S = l/V;   // Basic Rule for saturation.

    }
```

```
float max (R, G, B) {
   float m = 0.0;
   if (R > m) {
       m = R;
   }
   if (G > m) {
       m = G;
   if (B > m) {
       m = B;
   }
```

" Same for min

labs (Cohen Sutherland)

$a = -350$
$b = -300$
$c = 40$
$d = 1.17$
x_2



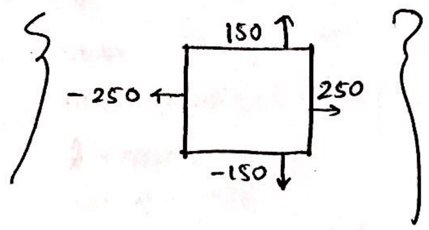
(8) Code
1000 → 1000
1001
0000 → 0
0010 → 0
0110 → 0
1000
1010 → 1000
0101 → 0001

0010
0100
0000

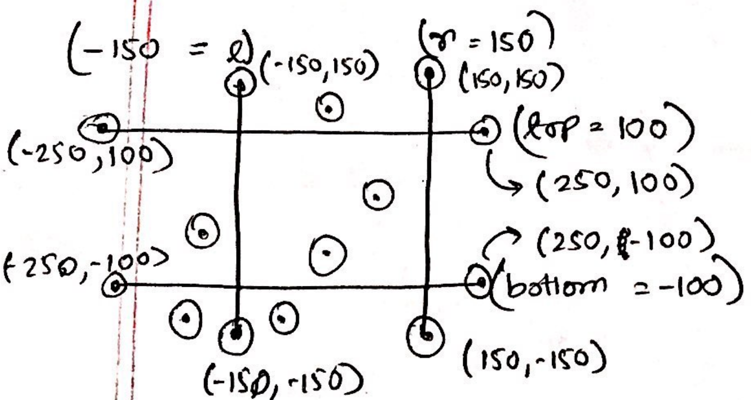
0001
0101
0001

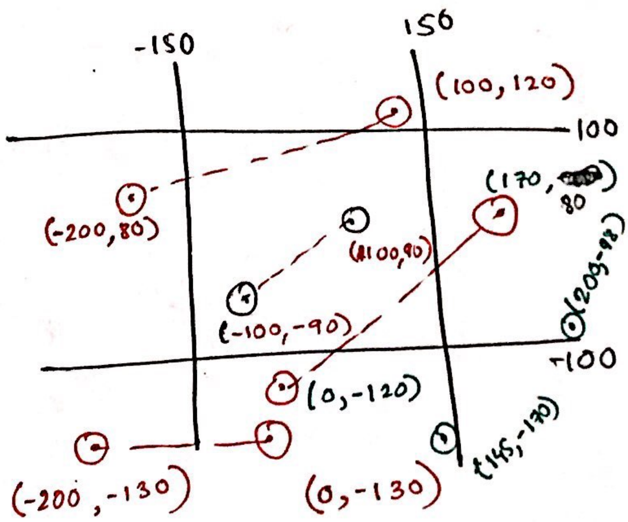
0010
0100
0000

accept → bitwise OR = 0
reject → bitwise AND = 1



Range of my actual window.





RGB lecture Cont'd

RGB to HSV Conversion

Input	Output
R, G, B	H, S, V
in.R, G, B	out. H, S, V

$V = max(R, G, B)$

$S = \frac{max - min}{max}$;

two structures { HSV, RGB

return data type

```
HSV RGB_to_HSV (RGB input)
    HSV output;
    float max, min, l;
    max = Max(in.R, in.G, in.B);
    min = Min(in.R, in.G, in.B);
    l = max - min;
    out.V = max;
    if (out.V == 0) {
        out.S = 0;
    }
    else {
        out.S = l / max;
    }
    if (out.S == 0) {
        out.H = NaN; // Not a no. // H is undefined.
    }
```

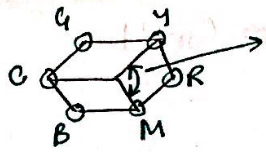
{Code continued}

if(S == 0) { H is undefined;}

OR can be written as

(in.R, in.G, in.B)

G dominant
(60 to 180°)



Red dominant
-60° to 60° }
OR
300° to 60° }

else {

if (in.R >= max) { // Red dominant.

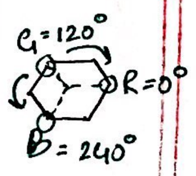
$H = \frac{G-B}{max-min} \times 60$; // → out.H $= \left(\frac{in.G - in.B}{l}\right) \times 60$; // can have -ve or +ve value, the ratio is betn (-1 ~ 1)

if (out.H < 0) {

out.H += 360;

}

}



Has B in +ve side and R in -ve side

120-60 => 60 and 120+60 => 180

60 - 180 Range for G so we add 120° so reference = 120°

else if (in.G >= max) { // G dominant

out.H = $\left(\frac{in.B - in.R}{l}\right) \times 60 + 120$;

}

else { // B dominant.

out.H $= \left(\frac{in.R - in.G}{l}\right) \times 60 + 240°$;

}

} // end to {S≠0}'s else

return out;

} // end.

for (1, 0.3, 0.3) // moderate red.

✓ H = 0°;

for (0.6, 0.4, 0.4) // light red

✓ H = 0°;

for (1, 0, 0) // Pure red

✓ H = 0°;

for (0.8, 0.6, 0.4)

✓ H = 15°

for (0.4, 0.3, 0.2)

✓ H = 15°

} Same hue as in same basic color just diff brightness

for Blue reference = 240°

and B dominant at (240° ~ 320°)

$-48 + 360 = 312$

OR

$-48 \% 360 = 312$

$360 \overline{) -48} (-1$

$\underline{+ -360}$

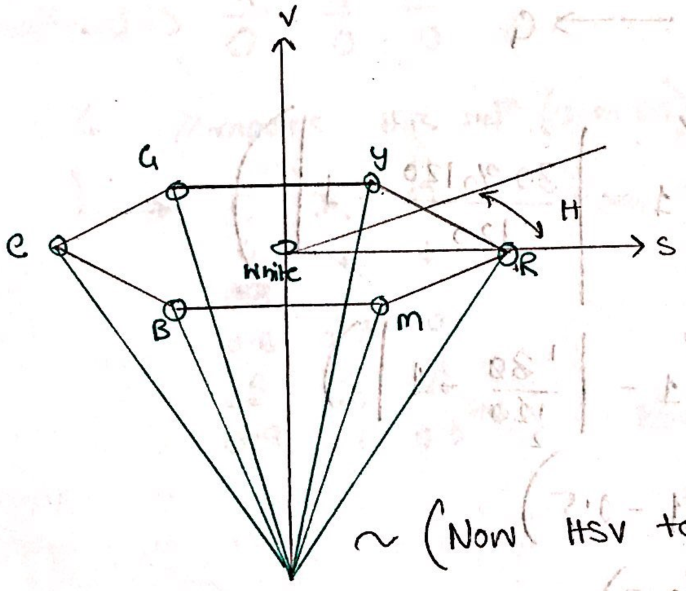
312 //

for % when less than 0 we do this

Suggestions; (for coding);

int H;

$H = (\text{int}) \left[\left\{ \left((G-B)/C \right) * 60 \right\} \% 360 \right];$



~ (Now HSV to RGB) ~

in $\begin{cases} H = 0^\circ \text{ to } 360^\circ \\ S = 0 \text{ to } 1 \\ V = 0 \text{ to } 1 \end{cases}$ out $\begin{cases} R = (0 \sim 1.0) \\ G = (0 \sim 1.0) \\ B = (0 \sim 1.0) \end{cases}$

(i) temp term $\leftarrow C = V * S;$ // not -ve

(ii) also a temp term where $H' = \frac{H}{60}$

$X = C\left(1 - \left| H' \bmod 2 - 1 \right|\right)$

OR {same}

$X = C\left(1 - \left| \left(\frac{H \bmod 120}{60}\right) - 1 \right|\right)$

dominant

R = 0 G = 2 B = 4

RGB

$(0.9, 0.6, 0.3)$

$(30^\circ, 0.66\ldots, 0.9)$

HSV

Here,

$G \; = \; C = \frac{2}{3} \times 0.9$

$= 2 \times 0.3$

$= 0.6 \longrightarrow G$

and

Blue $= X = 0.6 \times \left(1 - \left|\frac{30 \% 120}{120} - 1\right|\right)$

$= 0.6 \times \left(1 - \left|\frac{\cancel{30}^{1}}{\cancel{120}_{2}} - 1\right|\right)$

$= 0.6\left(1 - 0.5\right)$

$= 0.6\left(0.5\right)$

$= 0.3 \longrightarrow B$

for RGB

$(0.6, 0.9, 0.3)$

HSV

$(90^\circ, 0.66\ldots, 0.9)$

$C = 0.6$, $X = 0.3$

if H undefined $\Rightarrow$ $\frac{R'}{0}$ $\frac{G'}{0}$ $\frac{B'}{0}$

$0 \leq H' < 1$ // means H is betn (0 to 60).

previous

for $(30^\circ, 0.66, 0.9)$ $(0.9, 0.6, 0.3)$

$\rightarrow$ C X 0

$\downarrow$ $\downarrow$ $\downarrow$

R'	G'	B'
0.6	0.3	0
$\underline{\underline{R}}$	$\underline{\underline{G}}$	$\underline{\underline{B}}$
0.9	0.6	0.3

$M = V - C;$

$= 0.9 - 0.6$

$= 0.3;$

So,

$$\begin{cases} R = R' + m \\ G = G' + m \\ B = B' + m \end{cases}$$

RGB → HSV
RGB → HSL
HSV → RGB
HSL → RGB

structured program / Algorithm will be given in final

(RGB to HSV)

$V = \max(R,G,B)$

$S = \frac{max - min}{max}$

H

if (R == max){

$H = \left(\frac{G-B}{max-min} \times 60\right) \bmod 360^\circ$

else if (G == max){

$H = \left(\frac{B-R}{max-min} \times 60\right) + 120^\circ$

else if (B == max){

(RGB to HSL / HLS)

$L = \frac{max + min}{2}$

$S = \frac{max - min}{1 - |2L - 1|}$

(same code for Hue) → H

HSV to RGB

$C = V \times S$

same X for both → $X = C \times \left(1 - \left|\frac{H}{60} \bmod 2 - 1\right|\right)$

OR

$\frac{H \bmod 120}{60}$

$M = V - C$

HSL to RGB

$C = (1 - |2L - 1|) \times S$

$M = L - C/2$

(defn of C & M are diff but X is same :)

→ Now all remains is the table.

(Table)

	R'	G'	B'		
1	C	X	0	for	(0° ≤ H < 60) 0° ≤ H < 60
2	X	C	0	for	(60° ≤ H < 120) 60 ≤ H ≤ 120
3	0	C	X	for	(120 ≤ H < 180°) 120° to 180°
4	0	X	C	for	(180° ≤ H < 240°) 180 to 240°
5	X	0	C	for	(240° ≤ H < 300) 240° to 300°
6	C	0	X	for	(300 ≤ H < 360°) 300° to 360°

And $(R, G, B) = (R'+m, G'+m, B'+M)$

RGB		H	S	V/L
(0.0, 0.8, 0.8)	HSV	180°	1	0.8
	HSL	180°	1.0	0.4

180°

HSV to RGB

C = 0.8

X = 0.8

M = 0

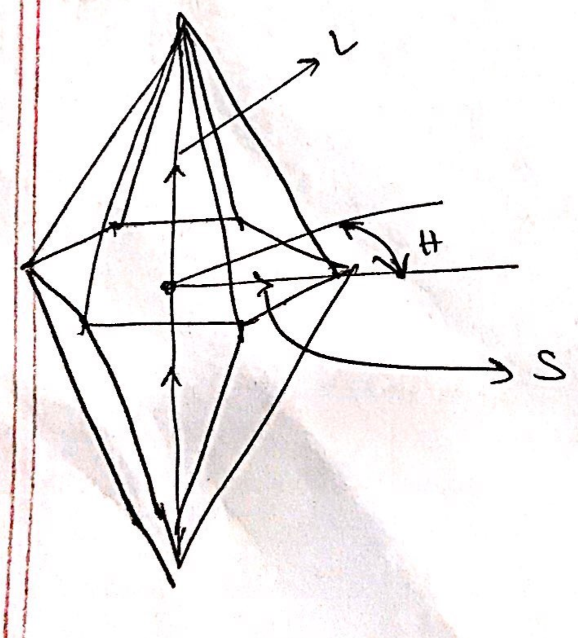
HSL to RGB

C = 0.8

X = 0.8

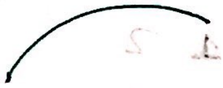
M = 0

So (RGB) ⟶ (0, 0.8, 0.8)

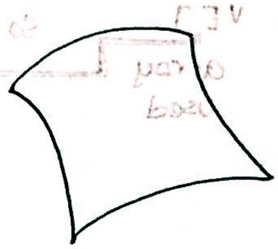


CHAPTER { CURVES & CURVED SURFACES }

curve



curved surface.



{Rendering}

Surface Rendering

Volume Rendering

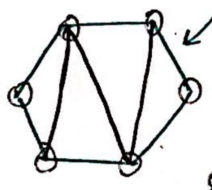
Polygonal Mesh

Curved Surface.

X

Polygonal Mesh.

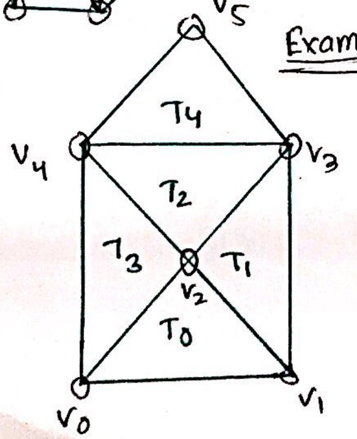
Any given polygon can be said to be set of triangles!



P_0	P_1	P_2

} coordinates of a triangle (3)

Structure (1) xplicit



Example

V_0	V_1	V_2	$\rightarrow T_0$
V_1	V_3	V_2	$\rightarrow T_1$
V_2	V_3	V_4	$\rightarrow T_2$
V_0	V_2	V_4	$\rightarrow T_3$
V_3	V_5	V_4	$\rightarrow T_4$

Ø Has high data redundancy, Common vertices come up a lot.

Scanned with CamScanner

most optimized in case of data size.

Structure (2) Pointed to a Vertex list

***.PLY**

V_0 ← V[ ] array used

V_1

V_2

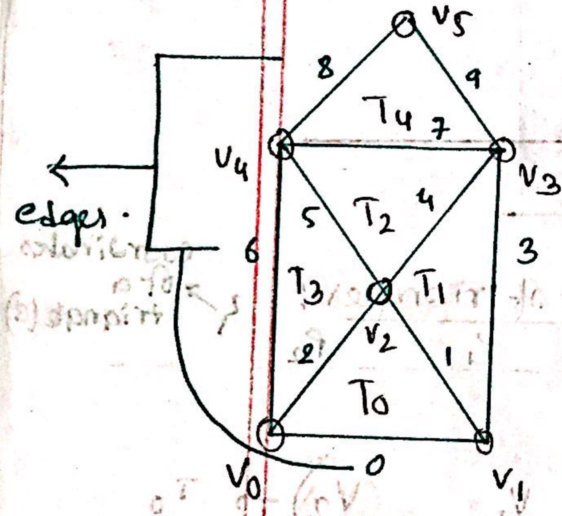
⋮

V_5

so it becomes (from prev example)

0	1	2
0	3	2
2	3	4
0	2	4
3	5	4

Structure (3) Pointed to an Edge list



V[ ] → vertex list

E[ ] → Edge list

(1) Hermite

(2) Beziev

{Curves & Surfaces Contd*}

2 points : $P_{(t)} = At + B$ --- line

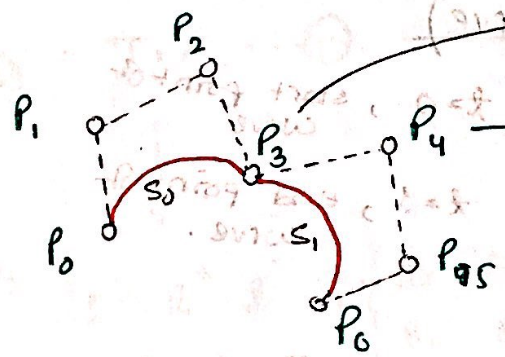
3 Points : $P_{(t)} = At^2 + Bt + C$ --- curve

4 Points : $P_{(t)} = At^3 + Bt^2 + Ct + D$ --- cubic curve

{Continuity of curves}

① Geometric → 0 order, 1st order, 2nd order

② Parametric

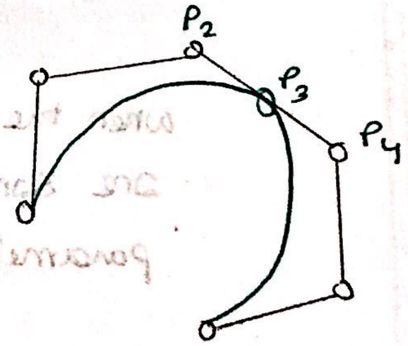


→ Geometric Continuity {0 order}

→ control Points

P_3 is common betn two curves

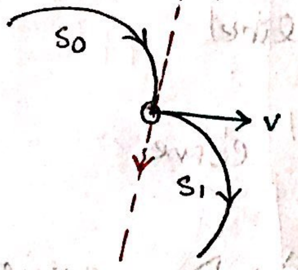
→ would have been 1st order curve if P_2, P_3, P_4 were co-linear



→ quadratic curve

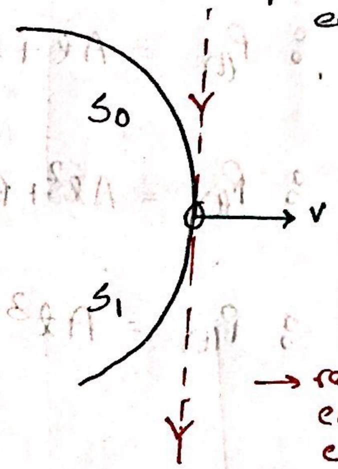
0 order

{only touches}



1st order

{1st derivative equal}



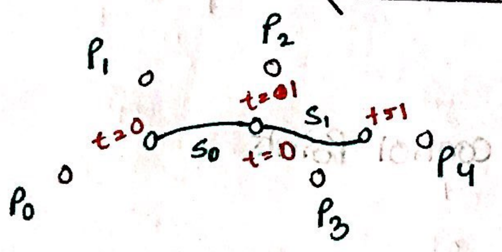
→ rep's a continuous curve

2nd order

{2nd derivative equal}

→ rep an even more continuous curve.

(Parametric Curve)



$t=0$, start point of curve

$t=1$, end point of curve.

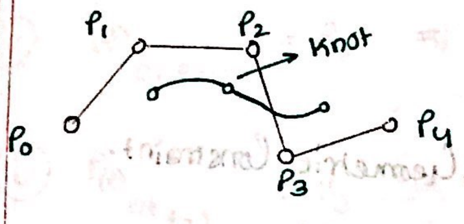
$P_0, P_1, P_2, P_3 \rightarrow$ (for S_0)

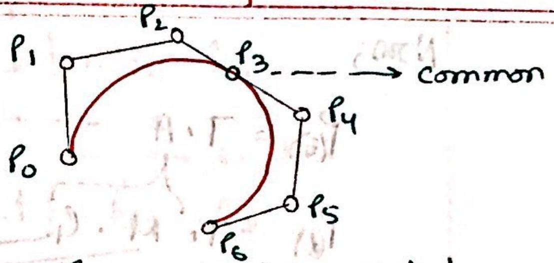
$P_1, P_2, P_3, P_4 \rightarrow$ (for S_1)

→ 3 overlaping Points (P_1, P_2, P_3)

when the parametric end and start point are same for two curves, then it is parametric continuety.

Geometric Continuety has a Point on the touching position





∅ only in approximated curve there is parametric continuety
→

∅ only in interpolated curve there is geometric continuety.
→ touches a control point in the common segment.

Standard Eqn of Cubic Polynomial

$$P_{(t)} = At^3 + Bt^2 + Ct + D \quad \text{---(i)}$$

where $T = |t^3 \; t^2 \; t \; 1|$
$= (1 \times 4)$ matrix
row column

$$P_{(t)} = T \cdot A \quad \text{---(ii)}$$

and,

$$A = \begin{vmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \\ D_x & D_y & D_z \end{vmatrix} \rightarrow (4\times3) \text{ matrix}$$

So, $\underset{(1\times4)}{|t^3 \; t^2 \; t \; 1|} \cdot \underset{(4\times3)}{\begin{vmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \\ D_x & D_y & D_z \end{vmatrix}}$

$= |P_{tx} \; ; \; P_{ty} \; , P_{tz}|$

$\rightarrow (1\times3)$ matrix.

Now,

$$P(t) = T \cdot A \quad \text{--- (ii)}$$

$$P(t) = T \cdot M \cdot G \quad \text{--- (iii)}$$

G → Geometric Constraint.

M → Basis matrix

T	M	G
$\lvert 1\times4 \rvert$	$\lvert 4\times4 \rvert$	$\lvert 4\times3 \rvert$
4 values	16 values	12 values
Known	unknown	Know

unknown → we need to find this out.

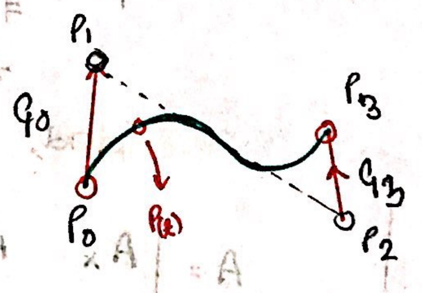
Hermite Curve :

→ How to solve eqn (iii) → Derivation of the Basis Matrix.

vector

$G_0 = (P_1 - P_0)$

$G_3 = (P_3 - P_2)$



in case of hermite eqn (B) can be written in this form:

$$P_t = T \cdot M_H \cdot G_H \quad \text{--- (i)}$$

G_H → Geom const of hermite curve.

$$G_H = \begin{vmatrix} P_0 \\ P_3 \\ G_0 \\ G_3 \end{vmatrix} = (4\times3) \text{ matrix}$$

P_0 --- $\underbrace{P_{0x}, P_{0y}, P_{0z}}$ → that's why.

G_0, G_3 → has P_1 & P_2 hidden

$P_{(t)} \rightarrow$ is a point on the curve

(v) $P_{(t)}$ at $t=0$ $= P_0 = \begin{array}{cccc} t^3 & t^2 & t & 1 \\ \end{array} \begin{vmatrix} 0 & 0 & 0 & 1 \end{vmatrix} \cdot M_H \cdot G_H$

for $t=1$

(v) $P_{(t)}$ at $t=1$ $= P_3 = \begin{array}{cccc} t^3 & t^2 & t & 1 \\ \end{array} \begin{vmatrix} 1 & 1 & 1 & 1 \end{vmatrix} \cdot M_H \cdot G_H$

$\frac{\partial P_{(t)}}{\partial t}$ at $t=0$ $= G_0$ --> } the derivative at $t=0$ and $t=1$

$= \begin{array}{cccc} 3t^2 & 2t & 1 & 0 \\ \end{array} \begin{vmatrix} 0 & 0 & 1 & 0 \end{vmatrix} \cdot M_H \cdot G_H$ → at $t=0$

$= \begin{vmatrix} 3 & 2 & 1 & 0 \end{vmatrix} \cdot M_H \cdot G_H$ → at $t=1$

$\frac{\partial P_{(t)}}{\partial t}$ at $t=1$ $= G_3$

Hence, we can form the G_H matrix with P_0, P_3, G_0, G_3

$$\cancel{G_H} = \begin{vmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{vmatrix} \cdot M_H \cdot \cancel{G_H}$$

So M_H is the inverse of this matrix

Scanned with CamScanner

Hence we can say, $\rightarrow$ inverse

$$M_H = \begin{vmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{vmatrix}^{-1}$$

$$\left|\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}\right|\left|\begin{matrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{matrix}\right| \qquad \left|\begin{matrix} 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}\right|$$

$$M_H = \begin{vmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}$$

direct result $\rightarrow$ (*) Practice inversing (4x4) matrix.

So,

ii

$$P(t) = B_H \, G_H \text{ , where } B_H = T \cdot M_H$$

$\rightarrow$ lecture 3

{Curves & Surfaces}

$$\begin{array}{ccc} T & M & G \\ |1\times 4| \cdot & |4\times 4| \cdot & |4\times 3| \end{array}$$

$$\begin{array}{cc} |1\times 4| \cdot & |4\times 3| \\ B & G \end{array}$$

Hermite Curve:

$$P_{(t)} = T_H \cdot M_H \cdot G_H \text{ --- (i)}$$

$$= B_H \cdot G_H \text{ --- (ii)}$$

$B_H \rightarrow$ Blending matrix

$$\underbrace{|t^3 \quad t^2 \quad t \quad 1|}_{B_{H_0}} \cdot \begin{vmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}$$

B_{H_1}

$$\therefore \Rightarrow B_{H_0} = 2t^3 - 3t^2 + 1$$

$$B_{H_1} = -2t^3 + 3t^2$$

$$B_{H_2} = t^3 - 2t^2 + t$$

$$B_{H_3} = t^3 - t^2$$

Final Eqn. $P_{(t)} = B_{H_0} P_0 + B_{H_1} P_3 + B_{H_2} G_0 + B_{H_3} G_3$ — x, y, z } calculat separate $P_{(t)}$ at all meshes

$P_{(t)}$ at $t = 0.5$

$B_{H_0} = 0.5$

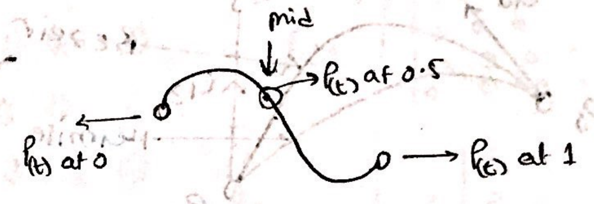
$B_{H_1} = 0.5$

$B_{H_2} = 0.125$

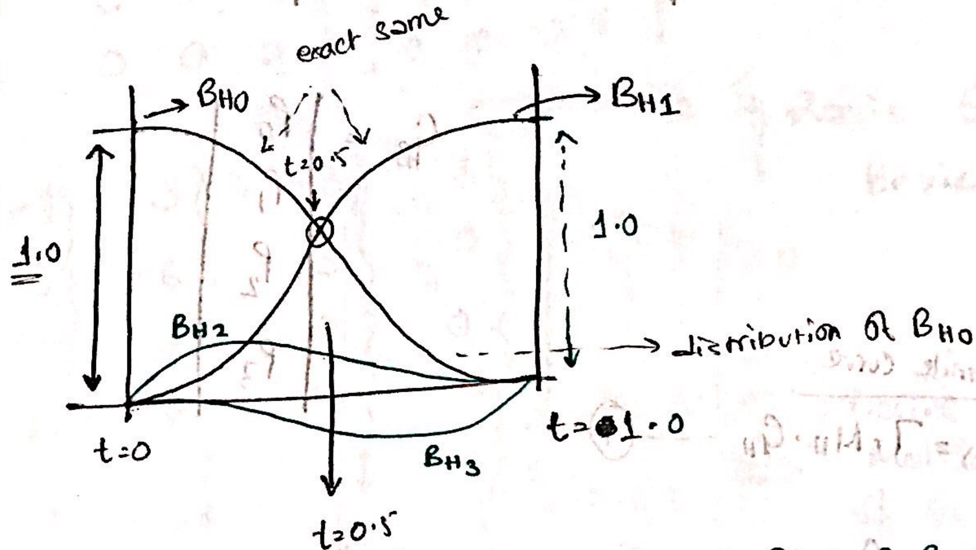
$\underbrace{B_{H_3} = -0.125}$

For any value of t, sum is 1

while $t \to (0 \sim 1)$



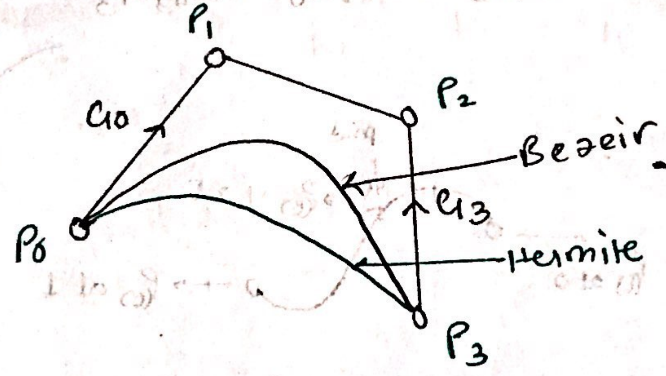
$P_{(t)} = 0.5 P_0 + 0.5 P_3 + 0.125 G_0 - 0.125 G_3$



at only $t = 0.5$, B_{H2} & B_{H3} is polar opp but same magnitude. so sum equals 0.

(*) sum of B_{H_1} and B_{H_0} is always 1.

(Bezier Curve:)



So,

$$G_H = \begin{vmatrix} P_0 \\ P_3 \\ G_{10} \\ G_{13} \end{vmatrix} \qquad G_{HB} = \begin{vmatrix} P_0 \\ P_3 \\ 3G_{10} \\ 3G_{13} \end{vmatrix}$$

$$G_B = \begin{vmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{vmatrix}$$

Hermite Curve

$P_{(t)} = T \cdot M_H \cdot G_H$ --- (i)

Bezier Curve.

$P_{(t)} = T \cdot M_H \cdot G_{HB}$ --- (ii)

(Bezier Curve)

$$P_{(t)} = T \cdot M_H \cdot G_{HB} \quad \text{-- (ii)}$$

$$= T \cdot \cancel{B}\, M_H \cdot \underbrace{\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{vmatrix}}_{M_B} \cdot G_B$$

IDEA

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{vmatrix} \cdot \begin{vmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{vmatrix} = \begin{vmatrix} P_0 \\ P_3 \\ -3P_0 + 3P_1 \\ -3P_2 + 3P_3 \end{vmatrix} = \begin{vmatrix} P_0 \\ P_3 \\ 3G_0 \\ 3G_3 \end{vmatrix}$$

So

$$\begin{array}{c} M_H \\ \begin{vmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \end{array} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{vmatrix}$$

$$= \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \rightarrow M_B \left\{ \text{Basis of Bezier} \right\}$$

$$B_0 \quad B_1 \quad B_2 \quad B_3$$

Q Derive the basis matrix ~~from~~ of Bezier curve from Hermite curve.

Scanned with CamScanner

So,

$$P_{(t)} = T \cdot M_B \cdot G_B$$

$$= B_B \cdot G_B$$

So,

$B_{B_0} = -\cancel{3}t^3 + 3t^2 - 3t + 1 = (1-t^{\cancel{3}})^3 = 1 - 3t + 3t^2 - t^3$

$B_{B_1} = 3t^3 - 6t^2 + 3t \cancel{\phi} = 3t(1-t^{\cancel{2}})^2 = 3t - 6t^2 + 3t^3$

$B_{B_2} = -3t^3 + 3t^2 = 3t^2(1-t)$

$B_{B_3} = t^3$

and,

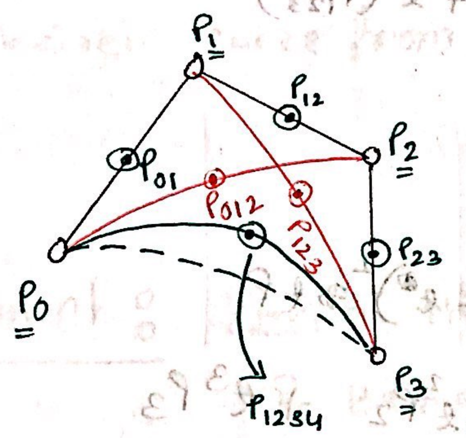
$\longrightarrow$ $P_{(t)} = B_{B_0} P_0 + B_{B_1} P_1 + B_{B_2} P_2 + B_{B_3} P_3$

So final Eqn: (stands)

$$P_{(t)} = (1-t^3) P_0 + (1-t)^2 . 3t . P_1 + 3t^2(1-t) P_2 + t^3 P_3$$

Scanned with CamScanner

{General Eqn of Bezier Curve}



$= P_0 + t(P_1 - P_0)$

Two points based eqn

$$\left.\begin{aligned} P_{01} &= (1-t)P_0 + P_1(t) \\ P_{12} &= (1-t)P_1 + P_2(t) \\ P_{23} &= (1-t)P_2 + P_3(t) \end{aligned}\right\} \rightarrow (i)$$

(3 Points based eqn)

$$P_{(t)} = P_{012} = (1-t)P_{01} + P_{12}(t)$$

$$= (1-t)\{(1-t)P_0 + P_1(t)\} + (t)\{(1-t)P_1 + P_2(t)\}$$

$$= (1-t)^2 P_0 + 2t(1-t)P_1 + t^2(P_2)$$

$P_{(t)} = P_{123}$ = same just slight diff

$$= (1-t)^2 P_1 + 2t(1-t)P_2 + t^3(P_3)$$

$$P_{(t)} = P_{0123} = (1-t)P_{012} + t\,(P_{123})$$

$$\cancel{=(1-t)\{(1-t}$$

$$= (1-t)^3 P_0 + 3(1-t)^2\, t\, P_1 + 3(1-t)\,t^2 P_2 + t^3 P_3$$

Same as before

$$\left\{\begin{aligned}
B_0 &= (1-t^3) &&= 1 - 3t + 3t^2 + t^3 \\
B_1 &= (1-t)^2 \cdot 3t &&= 3t - 6t^2 + 3t^3 \\
B_2 &= (1-t)\,t^2 \cdot 3 &&= 3t^2 - 3t^3 \\
B_3 &= t^3 &&= t^3
\end{aligned}\right.$$

So,

$$\begin{vmatrix} t^3 & t^2 & t & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 & 3 & -3 & 1 \\ +3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}$$

Q Derive the basis matrix of a cubic beizeir curve from its general equation.

Assignment : Derive the (5×5) basis matrix of Beezier curve for quad from its general equation.

(5×5) {5 points}

P_0 P_1 P_2 P_3 P_4