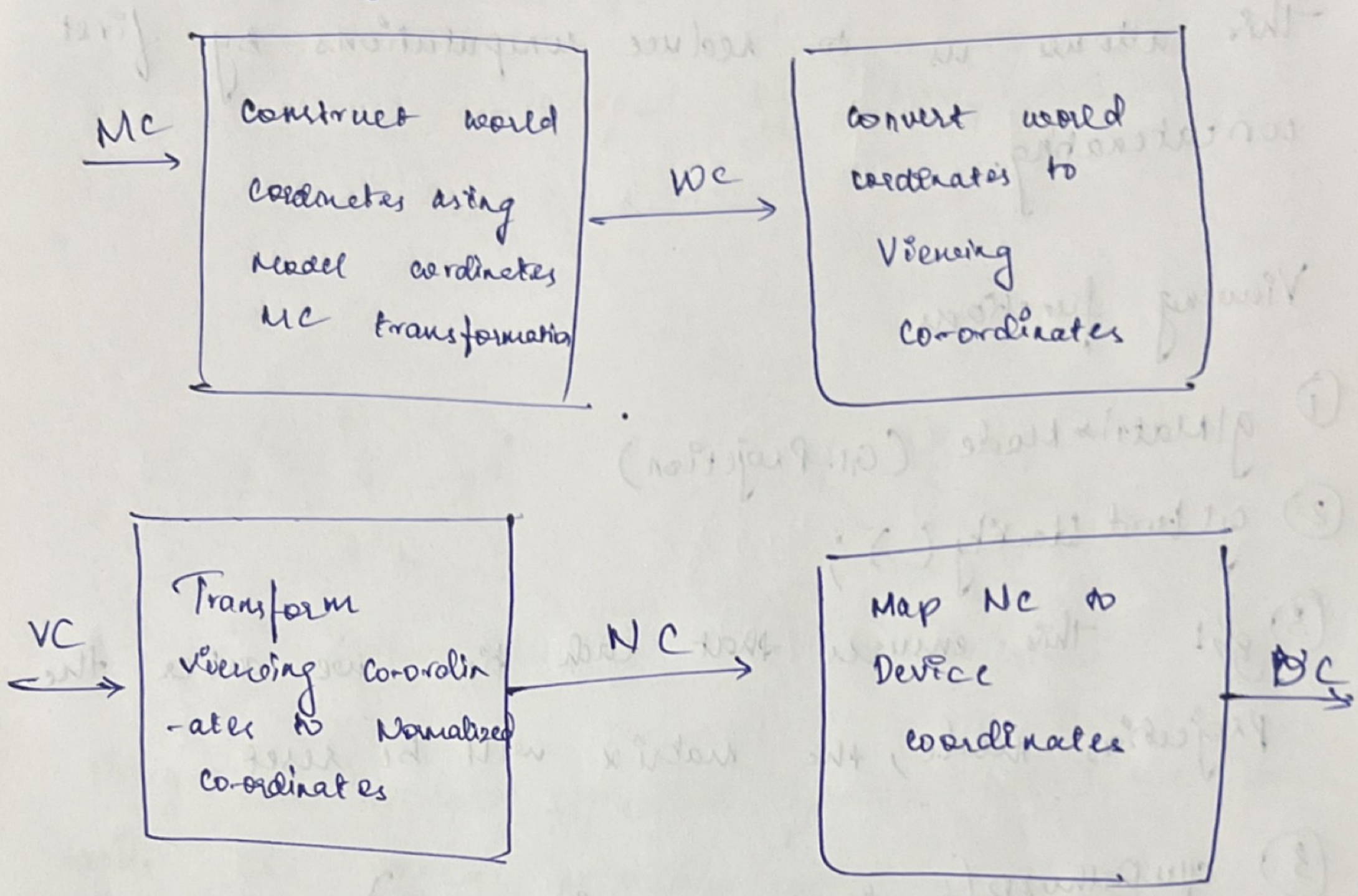
1

CG ASSIGNMENT

Name → Kevin A Barnes
USN → 1BY20CS085

1. 2D Viewing Transformation



* Mapping of a 2D dimensional, world-cordinates description to device coordinates is called 2-D Veiwing transformation.

* Once a world-coordinates scene has been constructed, we could set up a seperate two dimensional viewing co-ordinates reference frame

clipping is usually performed in normalized coordinates. This allows us to reduce computations by first concatenating.

Viewing functions

① glMatrixMode (GL_Projection)

② glLoadIdentity ();

③ glI This ensures that each time we enter the projection mode, the matrix will be reset.

③ gluOrtho2D (xwmin, xwmax, ywmin, ywmax);

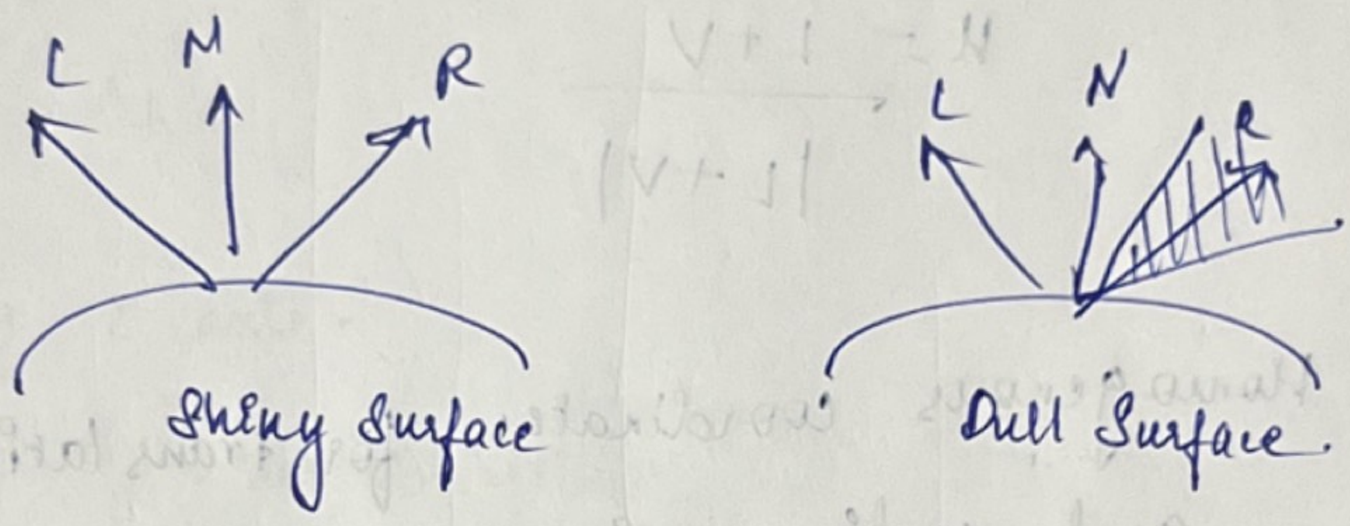
Specifies an orthogonal projection for mapping the scene to screen.

④ glViewport (xvmin, yvmin, width, height)

Specify the position of lowerleft corner of the viewport.

2. Build Phong Lighting Model with equations

An Empirical model for calculating the specular reflection range, developed by Phong Bui Tong and called Phong G Model.



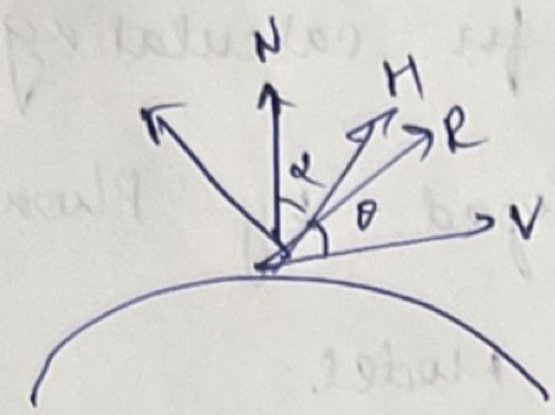
Using the spectral reflection function $w(\theta)$ we can write the Phong - reflection model as

$$I_{spec} = w(\theta) I_1 \cos^{n_s}\phi$$

$$I_{spec} = \begin{cases} k_s I_1 (V.R)^{n_s} & \text{if } V.R > 0 \\ 0.0 & \text{if } V.R \leq 0. \end{cases}$$

Projection of L onto the direction of Normal Vector has magnitude equal to dot product

$$R + L = (2 N.L) N$$



The Halfway Vector is obtained as

$$H = \frac{L+V}{|L+V|}$$

3. Apply Homogenous coordinates for translation, rotation and scaling via Matrix

Soln Three dimensional Translation

A position $P = (x, y, z)$ in 3 dimensional space is translated to a location.

$x' = x + t_x$ $\quad y' = y + t_y$ $\quad z = z + t_z$

$P = T \cdot P$

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

3- Dimensional Rotational

- Rotate an object about any axis in space
- Easiest rotation axes to handle are those that are parallel to cartesian axes
- Also can be use combination of coordinate axis.

Along z axis -

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

$$P' = R_z(\theta) \cdot P$$

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

Three dimensional Scaling

$$P' = S.P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

4. Outline the differences between raster scan display and random scan displays.

Raster Scan	Random Scan.
* Produces jagged lines that as a discreet point sets	* Random system produces smooth lines
* Less expensive	* More Expensive
* Modification difficult.	* Modification easy
* Resolution low	* Resolution easy
* Solid pattern easy to fill.	* Solid pattern difficult to fill

5. Demonstrate Open GL functions for window Management

Soln, Glut Createwindow - Used to create a new window.

- Glut Create window - Used to create another window within same window.

- Glut set window -
  Used to set a particular id for a window id

- Glut get window - Used to get window ID

- Glut PostRedisplay - To display the window again and again

- GLUT Reshape window - Used for transformation of world cordinates & displaying

- GLUT popwindow/glut Pushwindow -
  works just like matrix in an window.

6. Explain OpenGL visibility Detection Functions

1. OpenGL polygon culling Functions

Removes backface, frontface of an object

glCullFace (Mode);

glEnable (GL_CULL face);

glDisable (GL_CullFace);

b. Depth - Buffer functions

glut - Init DisplayMode (GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH)

glut - RGB | GLUT - DEPTH)

gl Linear (GL - Depth - BufferBIT)

this works as initialization function.

gl Clear (GL - Depth - BufferBIT)

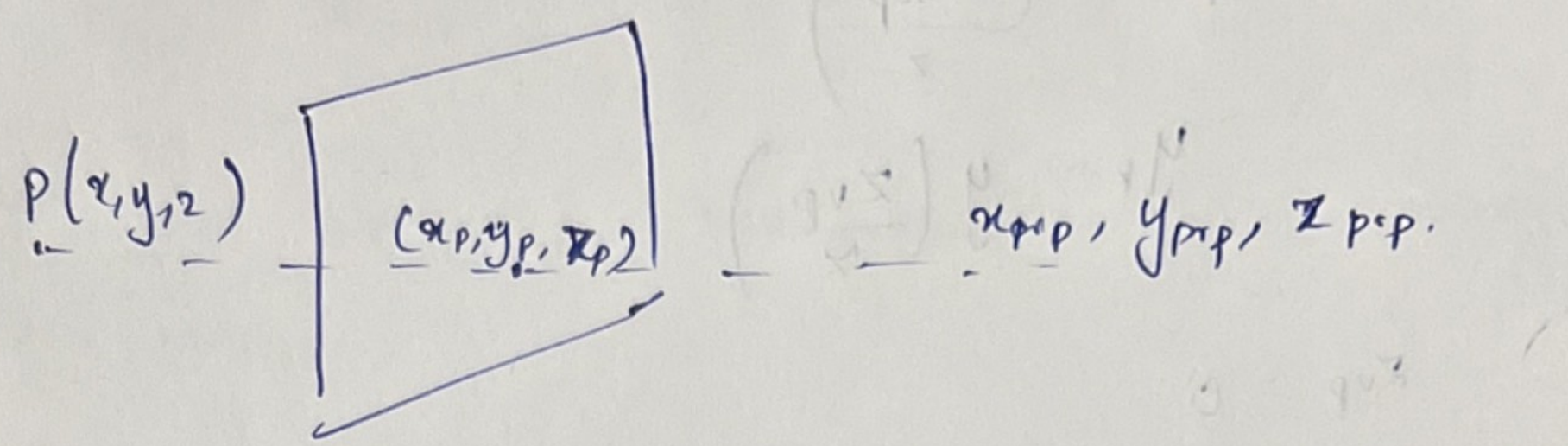
gl ClearDepth (Max Depth)

glEnable (GL - Depth - Test)

(c) OpenGL wireframe surface visibility Methods

gl Polygon Mode

Write special cases discussed with perspective Projections



Consider,

$$x' = x - (x - x_{prp})u$$

$$y' = y - (y - y_{prp})u$$

$$z' = z - (z - z_{prp})u$$

$$u = \frac{z_{vp} - z}{z_{prp} - z}$$

$$x_p = x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) + x_{prp}\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

Special cases:-

(1) $x_{prp}, y_{prp} = 0$

$$x_p = x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) \qquad y_p = y\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right)$$

(2) $x_{prp}, y_{prp}, z_{prp} = (0,0,0)$;

$$x_p = x\left(\frac{z_{prp}}{z}\right)$$

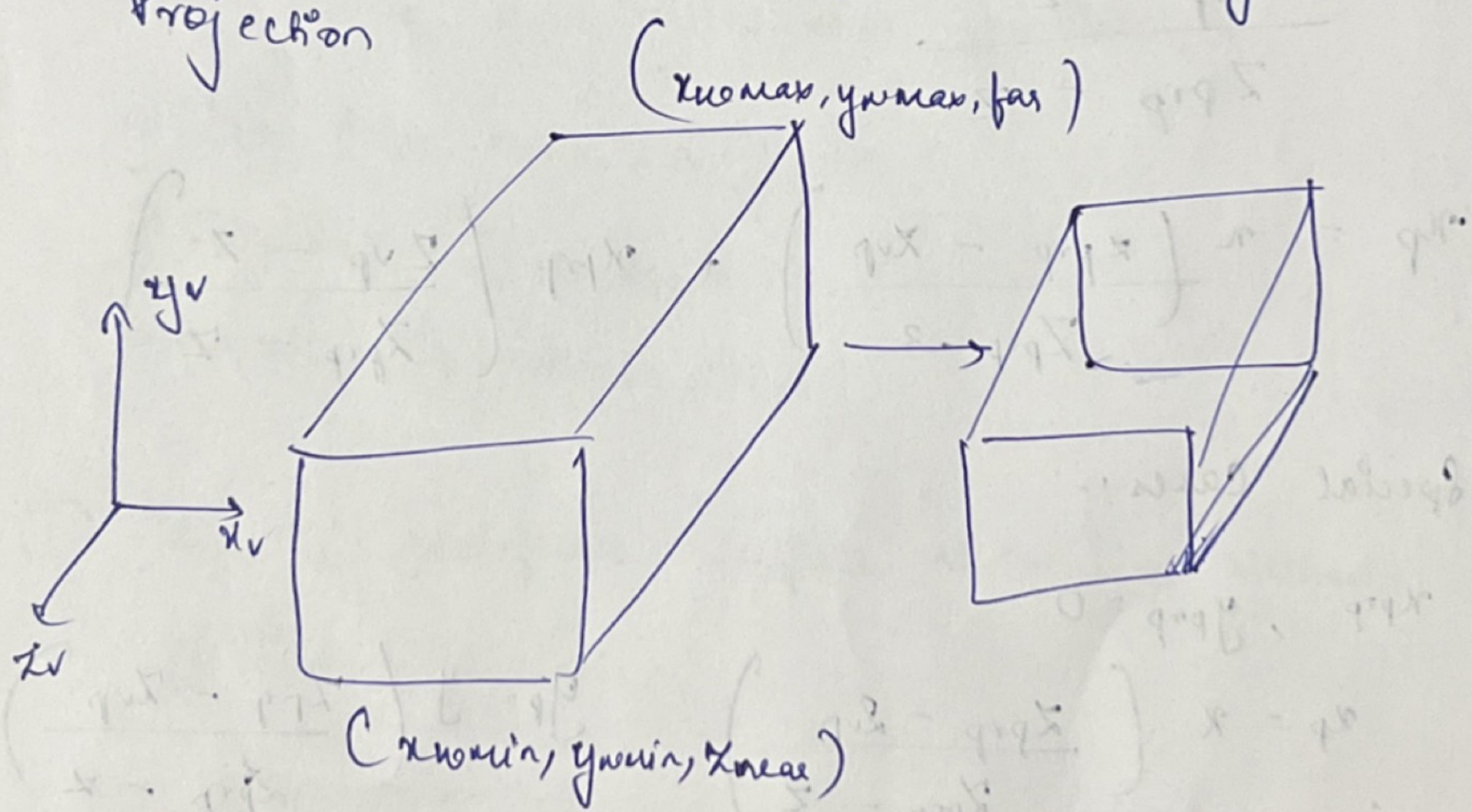
$$y_p = y\left(\frac{z_{vp}}{z}\right)$$

(3) $z_{vp} = 0$

$$x_p = x\left(\frac{z_{prp}}{z_{prp} - z}\right) - x_{prp}\left(\frac{z}{z_{prp} - z}\right)$$

$$y_p = y\left(\frac{z_{prp}}{z_{prp} - z}\right) - y_{prp}\left(\frac{z}{z_{prp} - z}\right)$$

8. Explain Normalisation for an Orthogonal Projection



we consider a unit cube for this normalized view volume with each x, y, z coordinates normalised in range 0 to 1

∴, we get Normalised transformation for orthogonal view volume

$$M_{ortho,\,norm} = \begin{bmatrix} \frac{2}{x_{wmax} - x_{wmin}} & 0 & 0 & -\frac{x_{wmax} + x_{wmin}}{x_{wmax} - x_{wmin}} \\ 0 & \frac{2}{y_{wmax} - y_{wmin}} & 0 & -\frac{y_{wmax} + y_{wmin}}{y_{wmax} - y_{wmin}} \\ & & \frac{2}{z_{max} - z_{far}} & \end{bmatrix}$$

9. Explain Bezier Curve and its properties with equations Bezier curve are parametric curves

$$\sum_{k=0}^{n} P_i B_i^n(t)$$

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i$$

n - polynomial degree

t = variable

i = index

n = control points number - 1

t = 0 - 1 (Range).

we used the above formulas.

10. Cohen - Sutherland line clipping Algorithm.

Cohen Sutherland Algorithm works on Region code.

Region code is 4-bit code

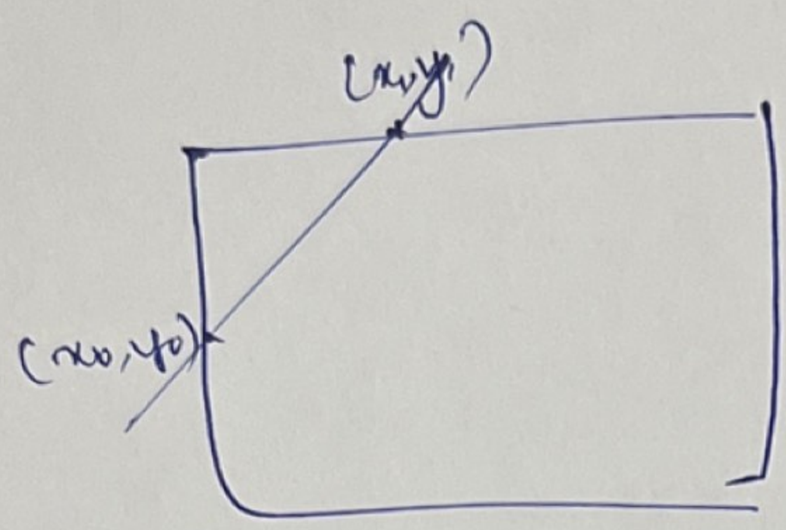
(ABRL)

(TBRL)

1001	1000	1010
0001	0000	0010
0101	0100	0110

$$m = (y - y_0)/(x - x_0)$$

$$m(x - x_0) = (y - y_0)$$

$$x = x_0 + (y - y_0)/m$$

$$y = y_0 + m(x - x_0).$$