Computer Graphics

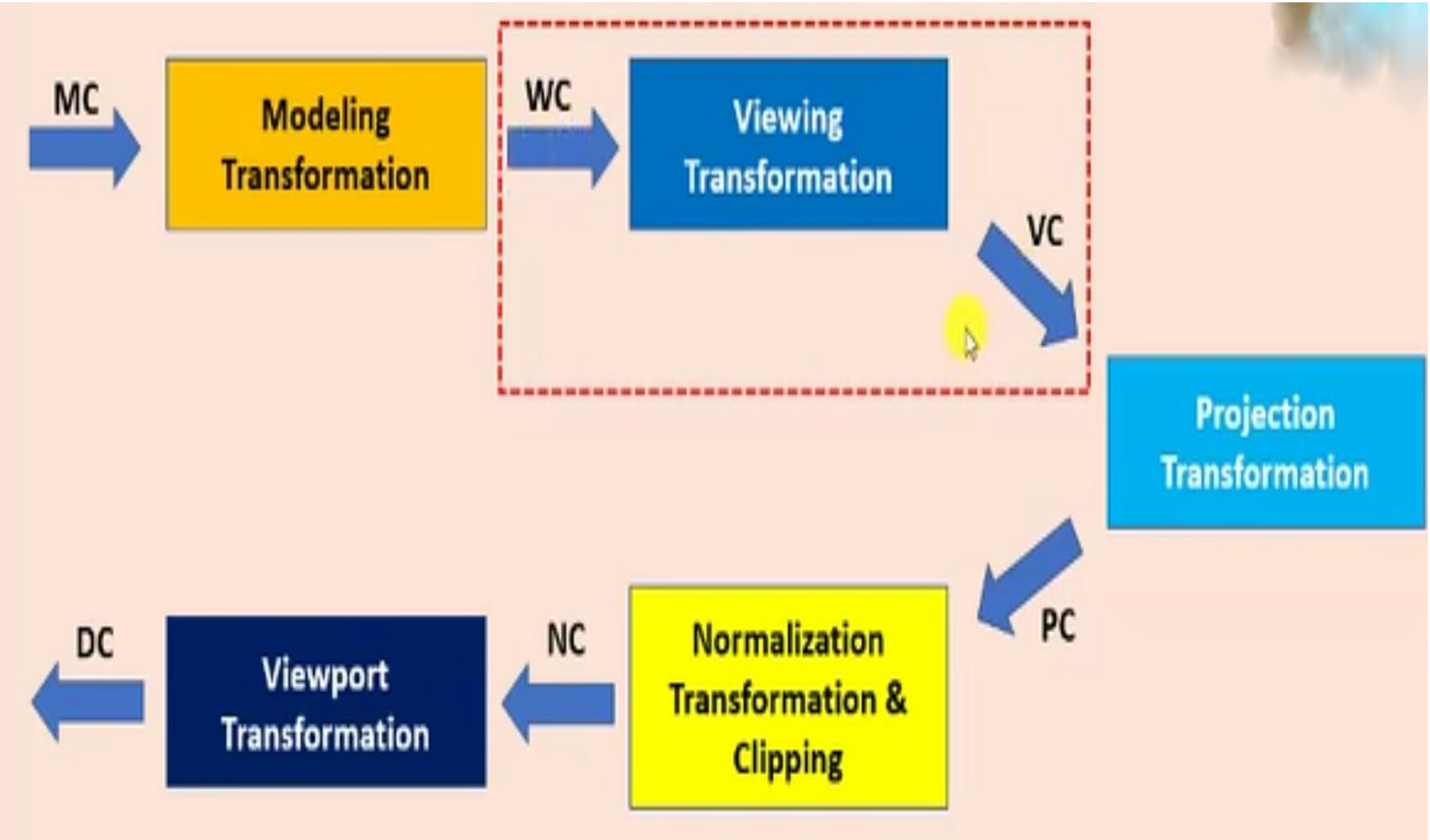
Unit 4 – Part 1

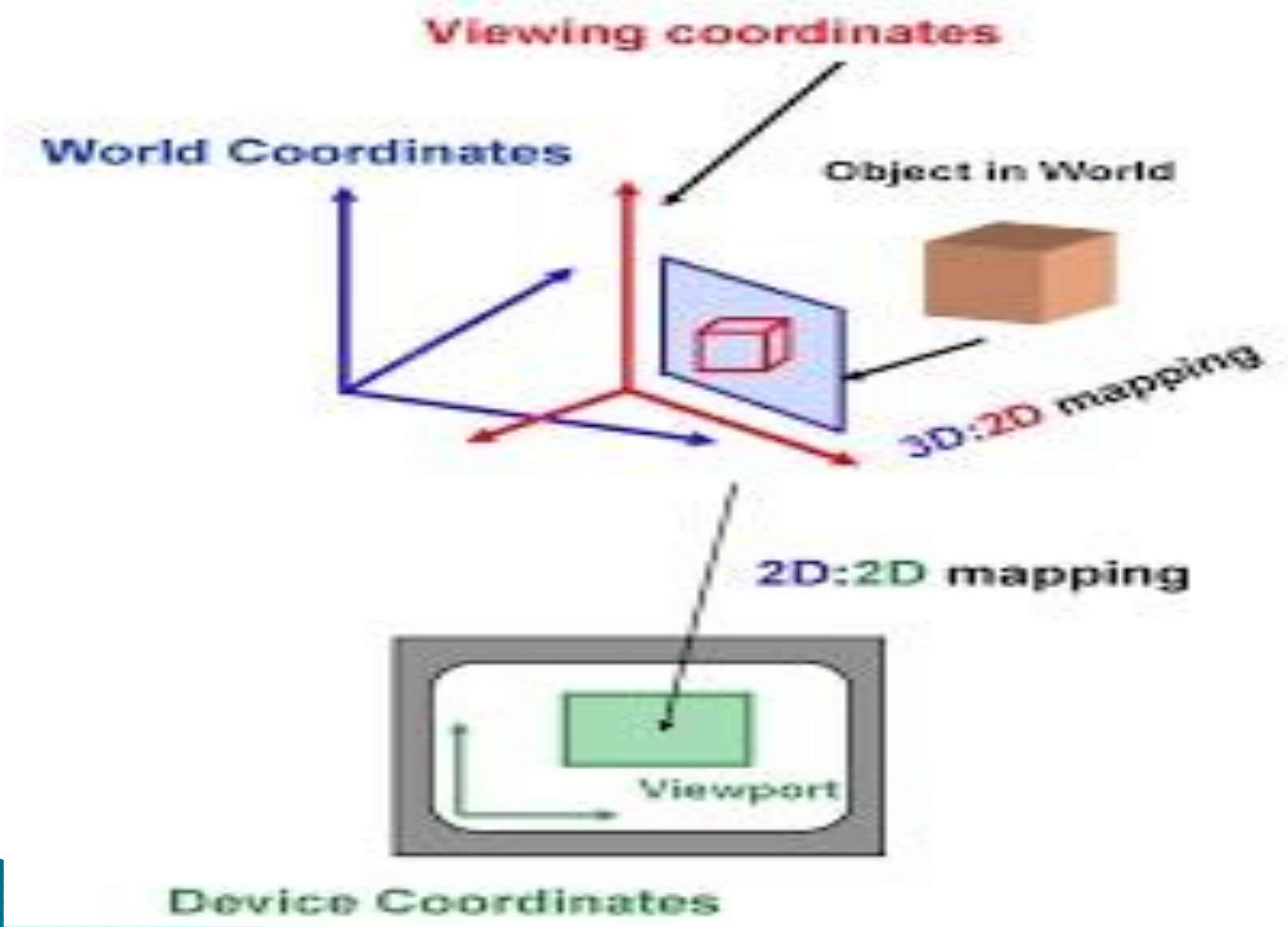
–By Manjula. S

3D Viewing

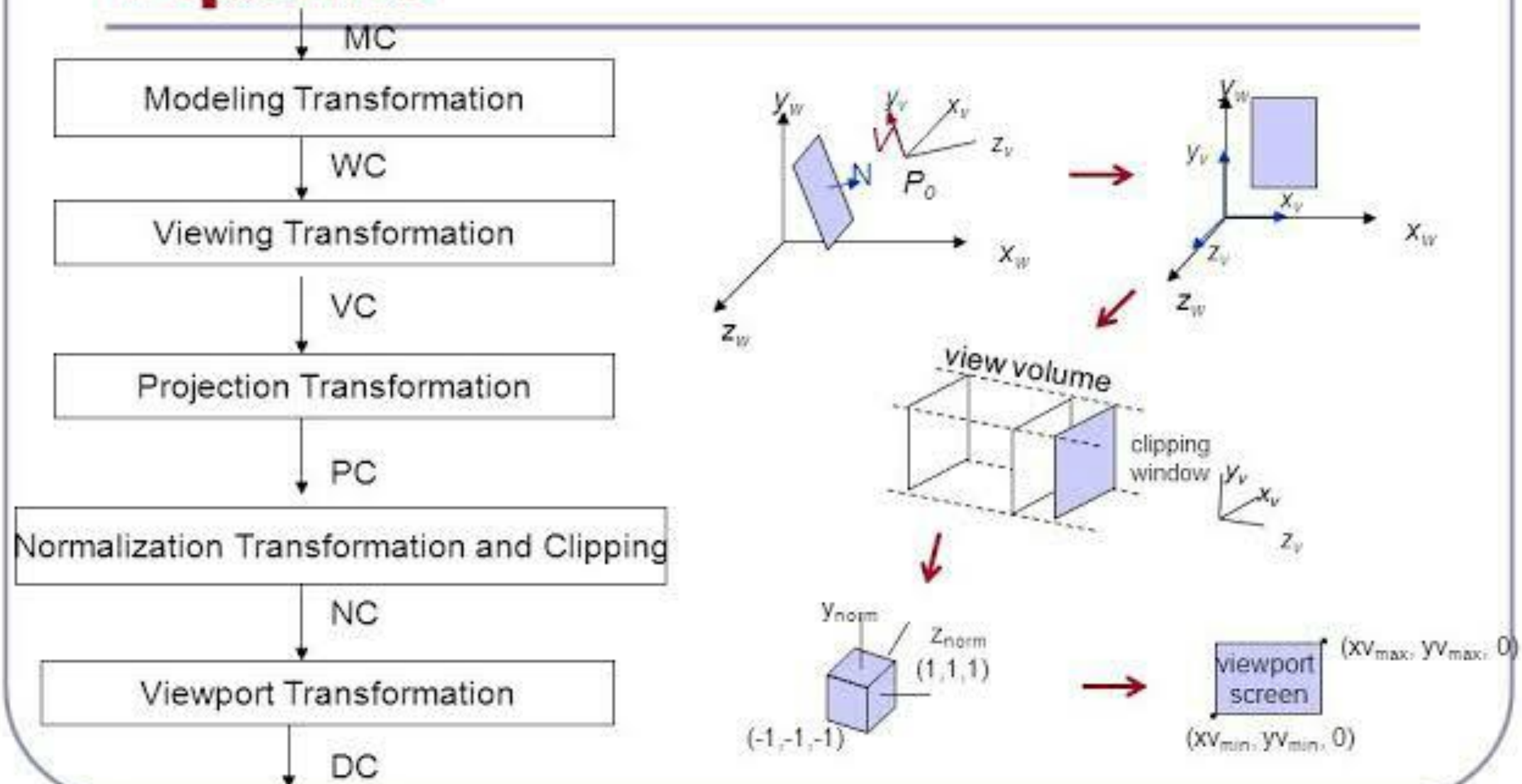
- Viewing Pipelining
- Viewing parameters
- Transformation of world coordinates to viewport coordinates

3D viewing Pipeline

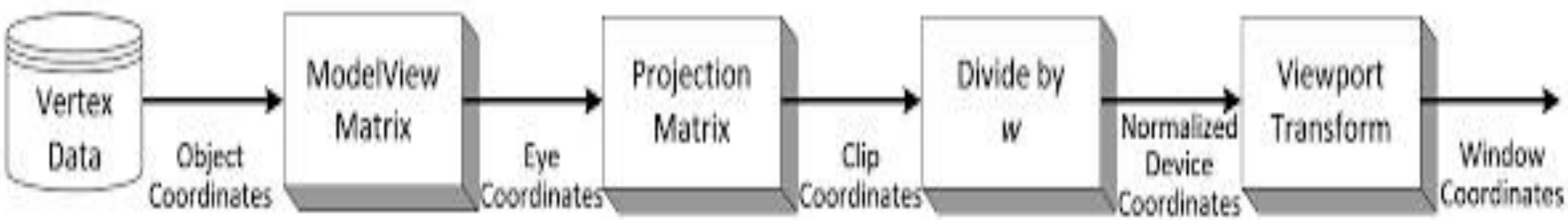



Viewing coordinates
World Coordinates
Object in World
3D:2D mapping
2D:2D mapping
Viewport
Device Coordinates

3D Viewing Transformation Pipeline



Viewing Pipelining



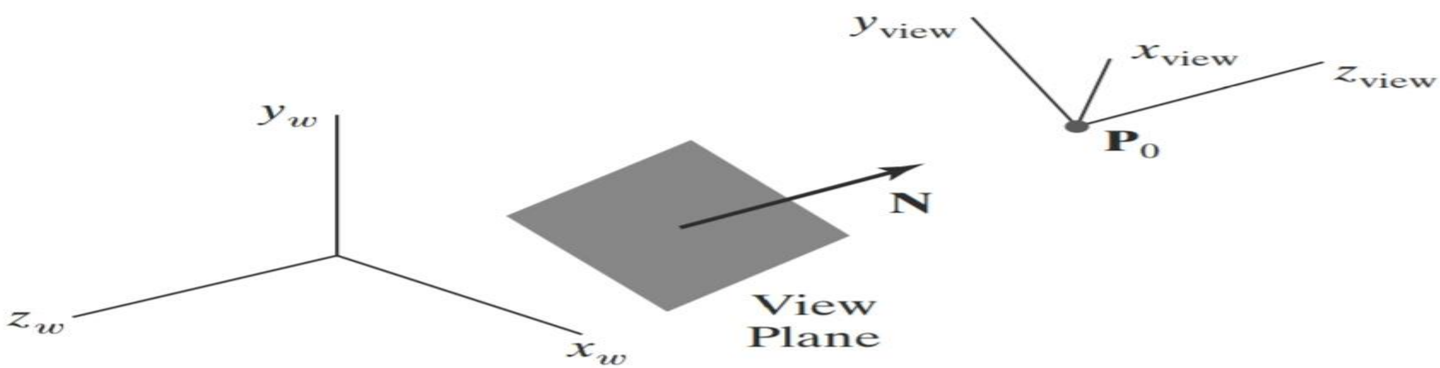
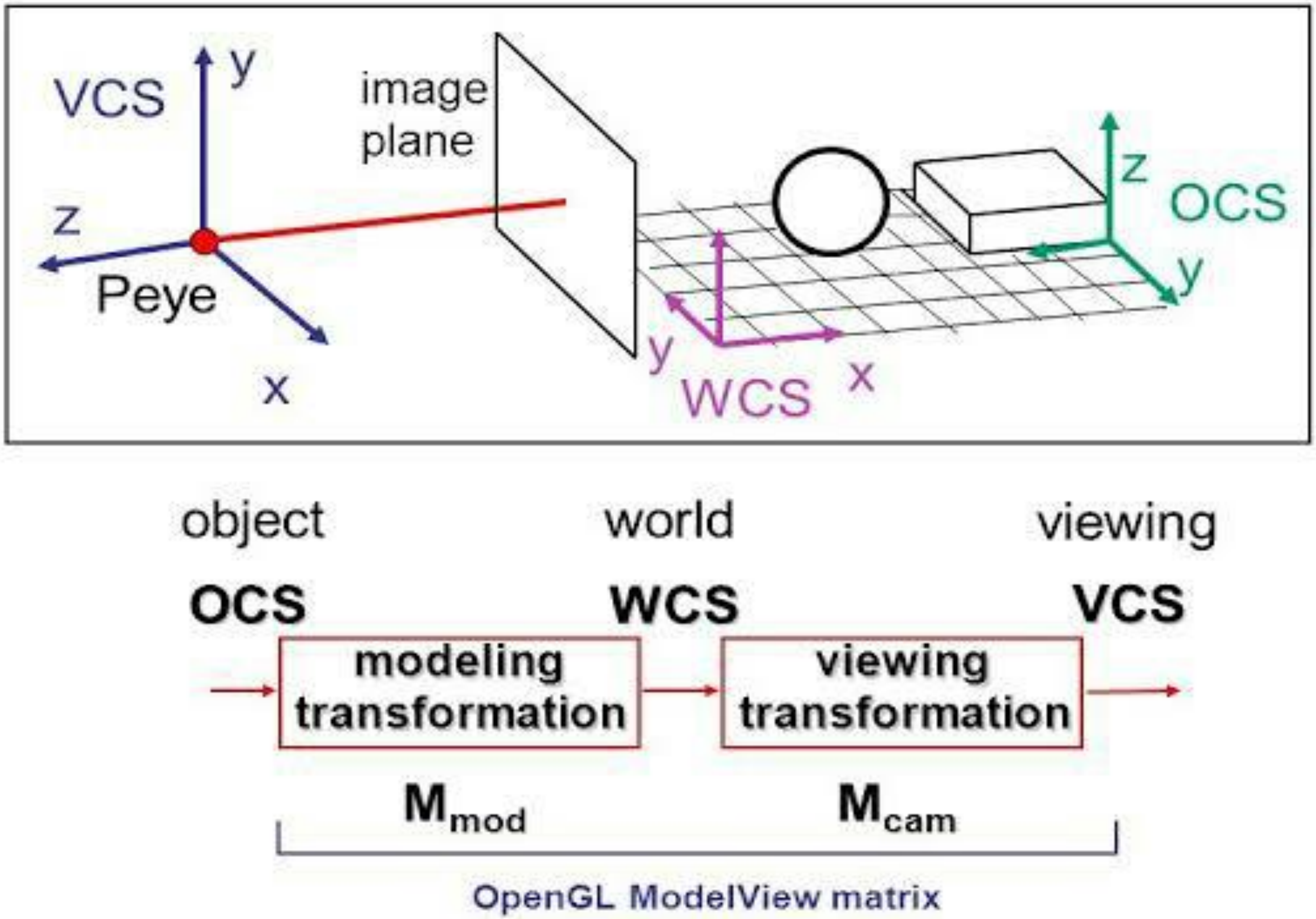


FIGURE 8
Orientation of the view plane and view-plane normal vector **N**.

Viewing Transformation



15

Viewing parameters

- View
- Viewplane
- View reference point(VRP)
- Normal vector
- Viewing distance

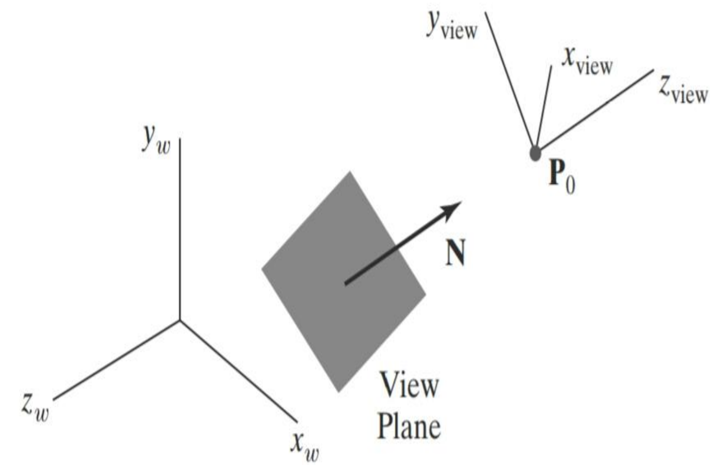


FIGURE 8
Orientation of the view plane and view-plane normal vector **N**.

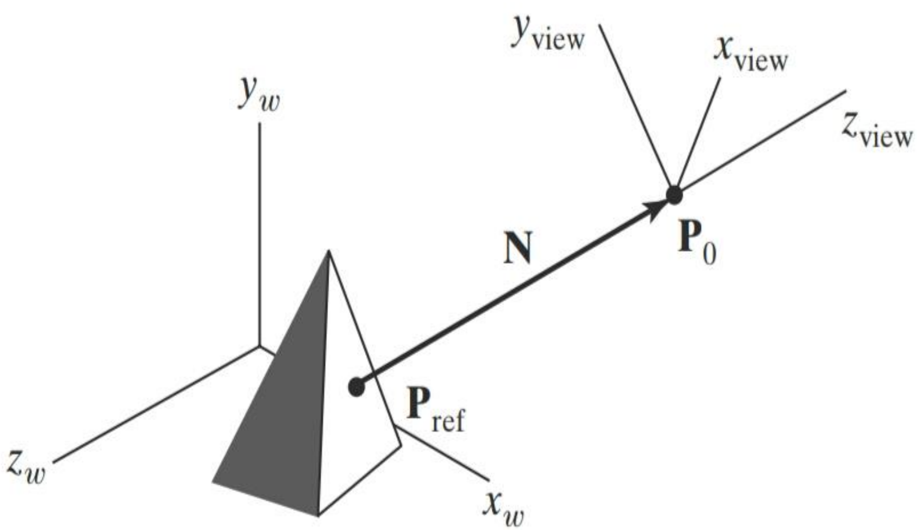
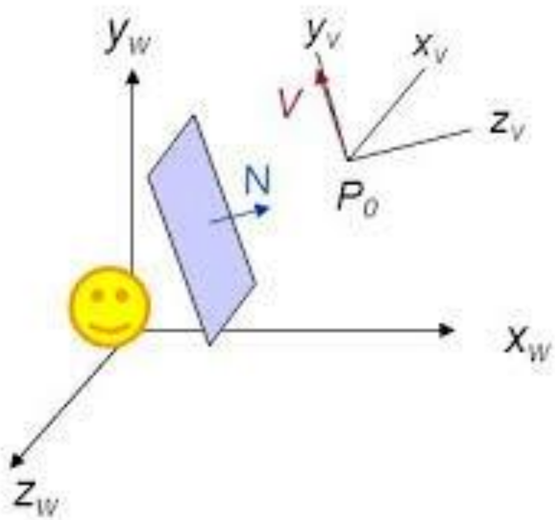


FIGURE 10
Specifying the view-plane normal vector **N** as the direction from a selected reference point P_{ref} to the viewing-coordinate origin P_0.

3D Viewing



V view up vector

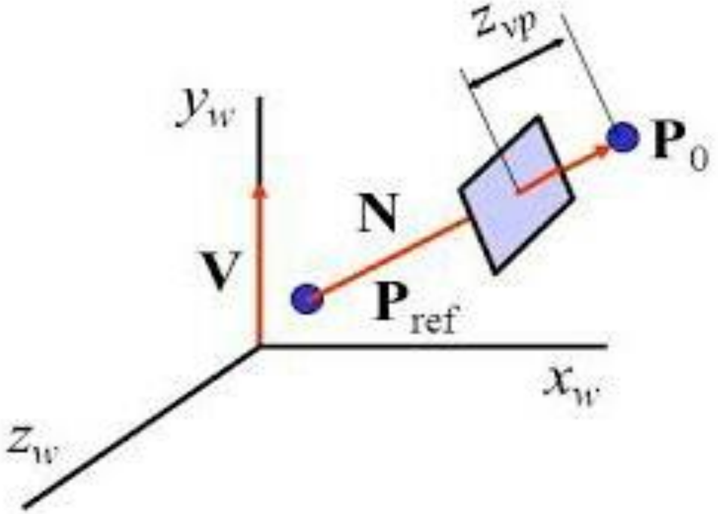
$P_0 = (x_0, y_0, z_0)$ view point

N viewplane normal

Viewplane is at point z_{vp} in negative z_v direction

V is perpendicular to N

3D viewing coordinates



Specification of projection:

$\mathbf{P}_0$: *View* or *eye point*

$\mathbf{P}_{ref}$: *Center* or *look-at point*

$\mathbf{V}$: *View-up vector* (*projection along vertical axis*)

z_{vp} : *positie view plane*

$\mathbf{P}_0, \mathbf{P}_{ref}, \mathbf{V}$: define *viewing* coordinate system

Several variants possible

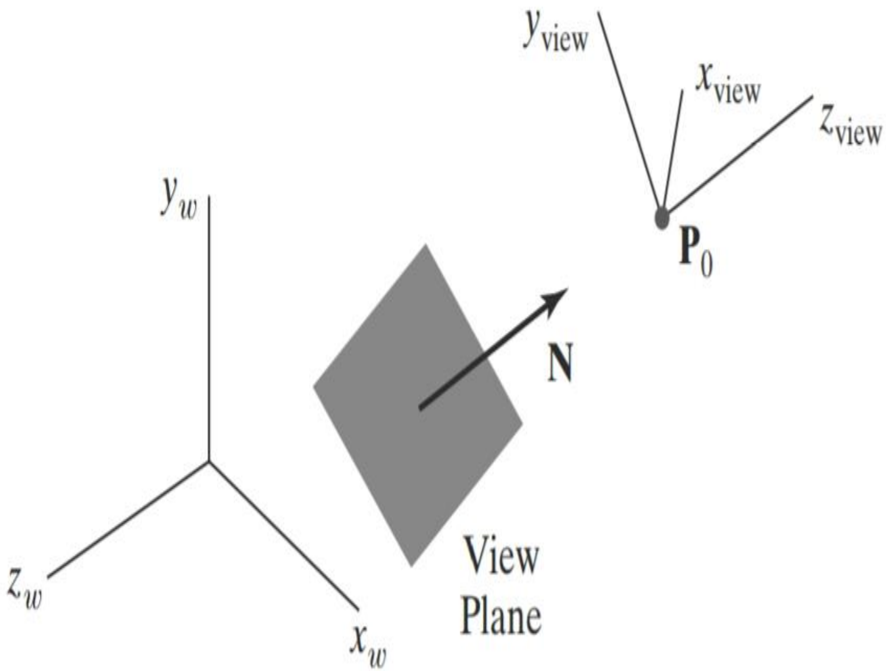


FIGURE 8
Orientation of the view plane and view-plane normal vector **N**.

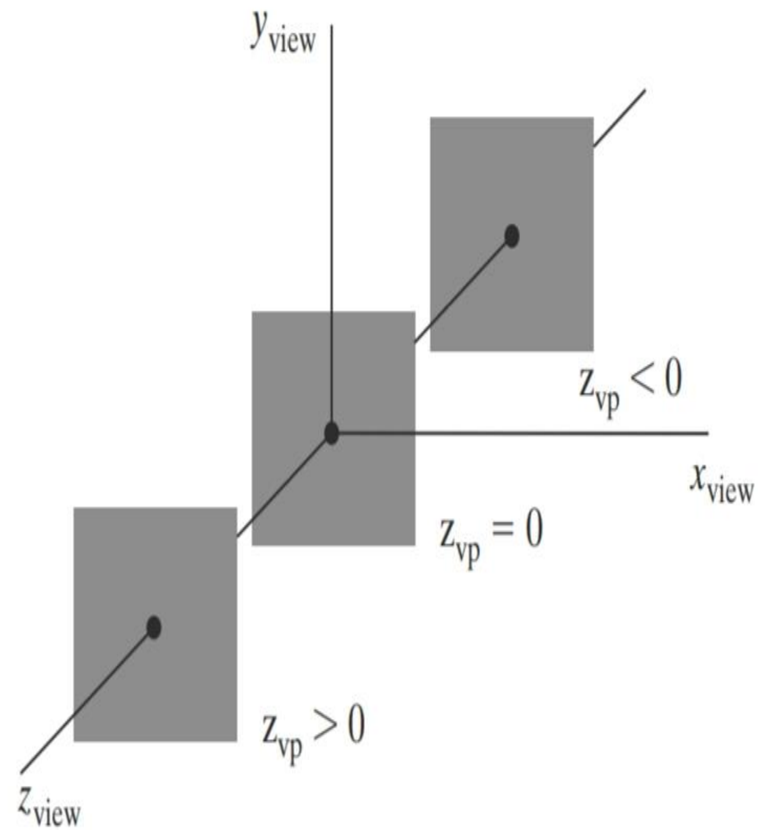


FIGURE 9
Three possible positions for the view plane along the z_{view} axis.

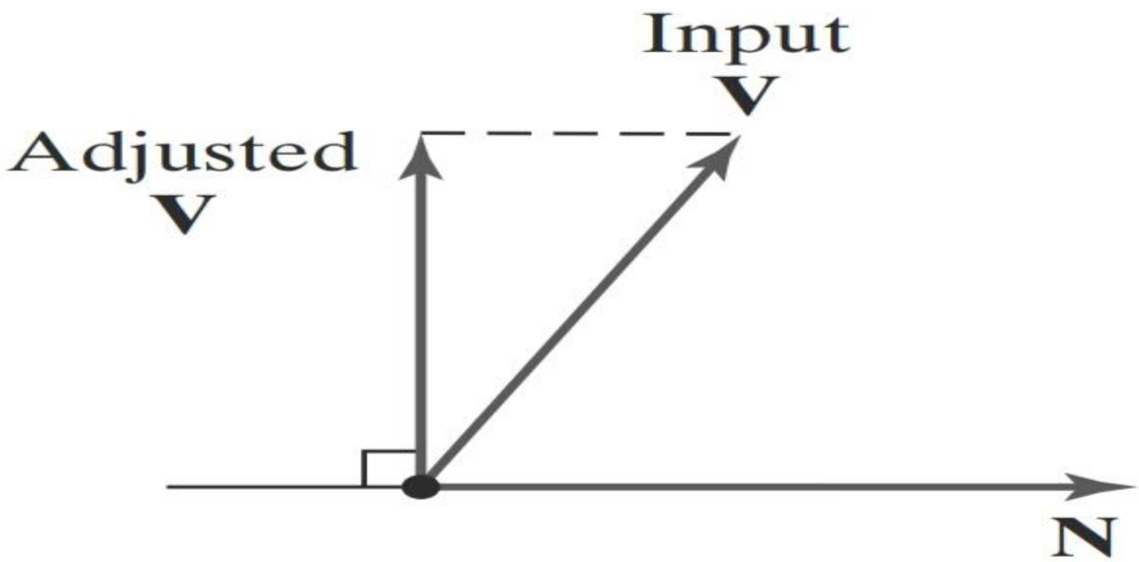


FIGURE 11
Adjusting the input direction of the view-up vector **V** to an orientation perpendicular to the view-plane normal vector **N**.

Transformation of window port to viewport

$$\mathbf{n} = \frac{\mathbf{N}}{|\mathbf{N}|} = (n_x, n_y, n_z)$$

$$\mathbf{u} = \frac{\mathbf{V} \times \mathbf{n}}{|\mathbf{V} \times \mathbf{n}|} = (u_x, u_y, u_z) \qquad (1)$$

$$\mathbf{v} = \mathbf{n} \times \mathbf{u} = (v_x, v_y, v_z)$$

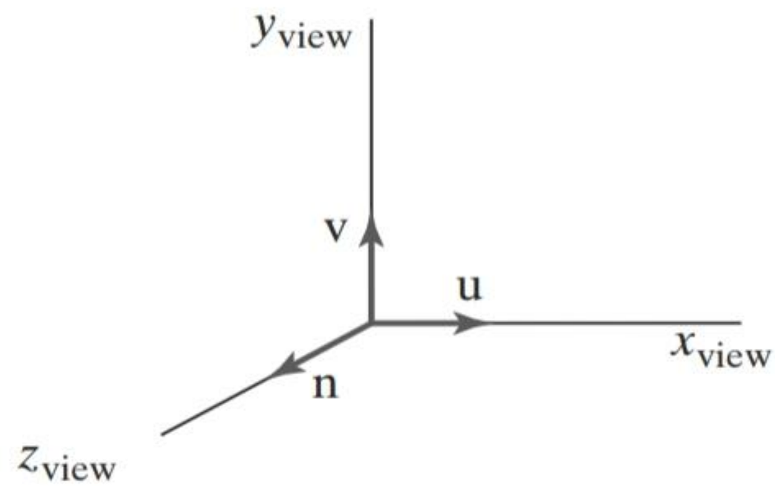


FIGURE 12
A right-handed viewing system defined with unit vectors **u**, **v**, and **n**.

Transformation of window port to viewport

- T(xv,yv,zv)
- Rx
- Rz
- Ry

Rm=T.Rx.Rz.Ry

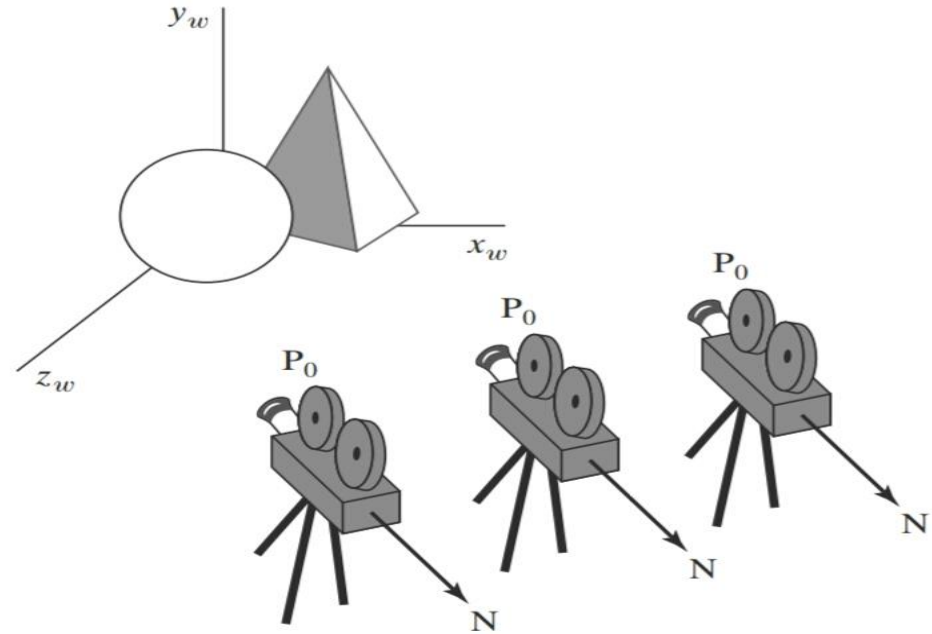


FIGURE 13
Panning across a scene by changing the viewing position, with a fixed direction for **N**.

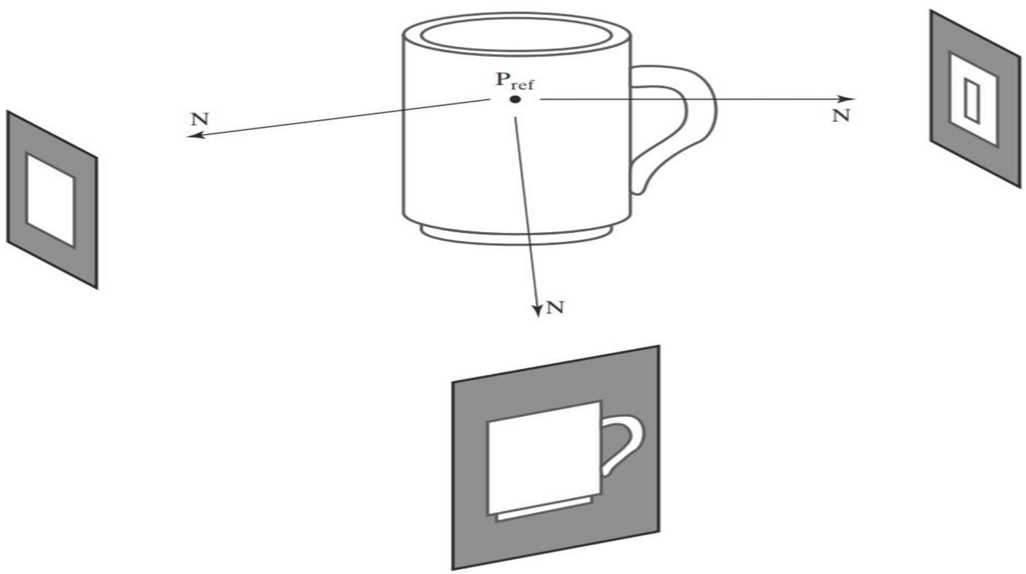
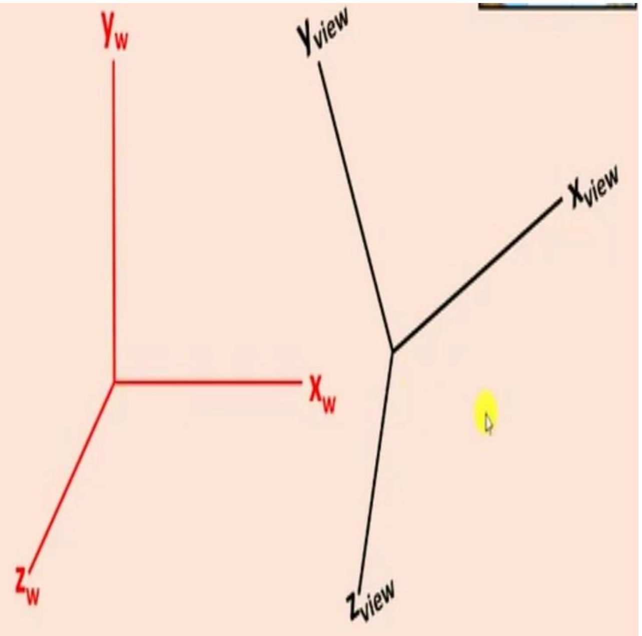


FIGURE 14
Viewing an object from different directions using a fixed reference point.

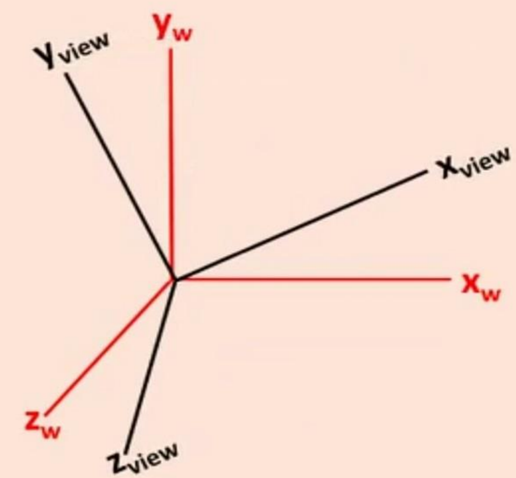
World Coordinates as Reference



Transformation from World to Viewing Coordinates

World Coordinates as Reference

Step-1 : Translate

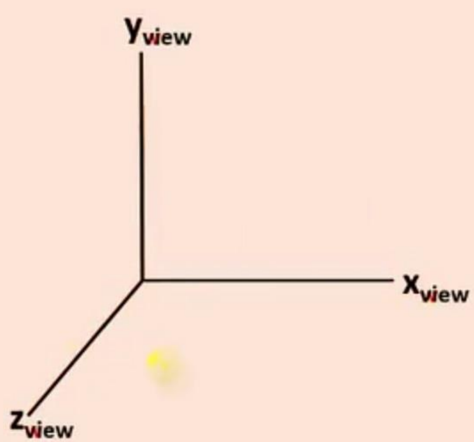


Transformation from World to Viewing Coordinates

World Coordinates as Reference

Step-1 : Translate

Step-2 : Rotate



Transformation from World to Viewing Coordinates

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

World Coordinates to Viewing Coordinates

$$\mathbf{M}_{WC,\,VC} = \mathbf{R} \cdot \mathbf{T} = \begin{bmatrix} u_x & u_y & u_z & -\mathbf{u} \cdot \mathbf{P}_0 \\ v_x & v_y & v_z & -\mathbf{v} \cdot \mathbf{P}_0 \\ n_x & n_y & n_z & -\mathbf{n} \cdot \mathbf{P}_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1. Translate the viewing-coordinate origin to the origin of the world-coordinate system.
2. Apply rotations to align the x_{view}, y_{view}, and z_{view} axes with the world x_w, y_w, and z_w axes, respectively.

The viewing-coordinate origin is at world position $\mathbf{P_0} = (x_0, y_0, z_0)$. Therefore, the matrix for translating the viewing origin to the world origin is

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

For the rotation transformation, we can use the unit vectors **u**, **v**, and **n** to form the composite rotation matrix that superimposes the viewing axes onto the world frame. This transformation matrix is

$$\mathbf{R} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where the elements of matrix **R** are the components of the **uvn** axis vectors.

The coordinate transformation matrix is then obtained as the product of the preceding translation and rotation matrices:

$$\begin{aligned} \mathbf{M}_{WC,\,VC} &= \mathbf{R} \cdot \mathbf{T} \\ &= \begin{bmatrix} u_x & u_y & u_z & -\mathbf{u} \cdot \mathbf{P}_0 \\ v_x & v_y & v_z & -\mathbf{v} \cdot \mathbf{P}_0 \\ n_x & n_y & n_z & -\mathbf{n} \cdot \mathbf{P}_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4}$$

Transformation of window port to viewport

Translation factors in this matrix are calculated as the vector dot product of each of the **u**, **v**, and **n** unit vectors with $\mathbf{P}_0$, which represents a vector from the world origin to the viewing origin. In other words, the translation factors are the negative projections of $\mathbf{P}_0$ on each of the viewing-coordinate axes (the negative components of $\mathbf{P}_0$ in viewing coordinates). These matrix elements are evaluated as

$$
\begin{aligned}
-\mathbf{u} \cdot \mathbf{P}_0 &= -x_0 u_x - y_0 u_y - z_0 u_z \\
-\mathbf{v} \cdot \mathbf{P}_0 &= -x_0 v_x - y_0 v_y - z_0 v_z \\
-\mathbf{n} \cdot \mathbf{P}_0 &= -x_0 n_x - y_0 n_y - z_0 n_z
\end{aligned}
\tag{5}
$$

Matrix 4 transfers world-coordinate object descriptions to the viewing reference frame.

Thank You