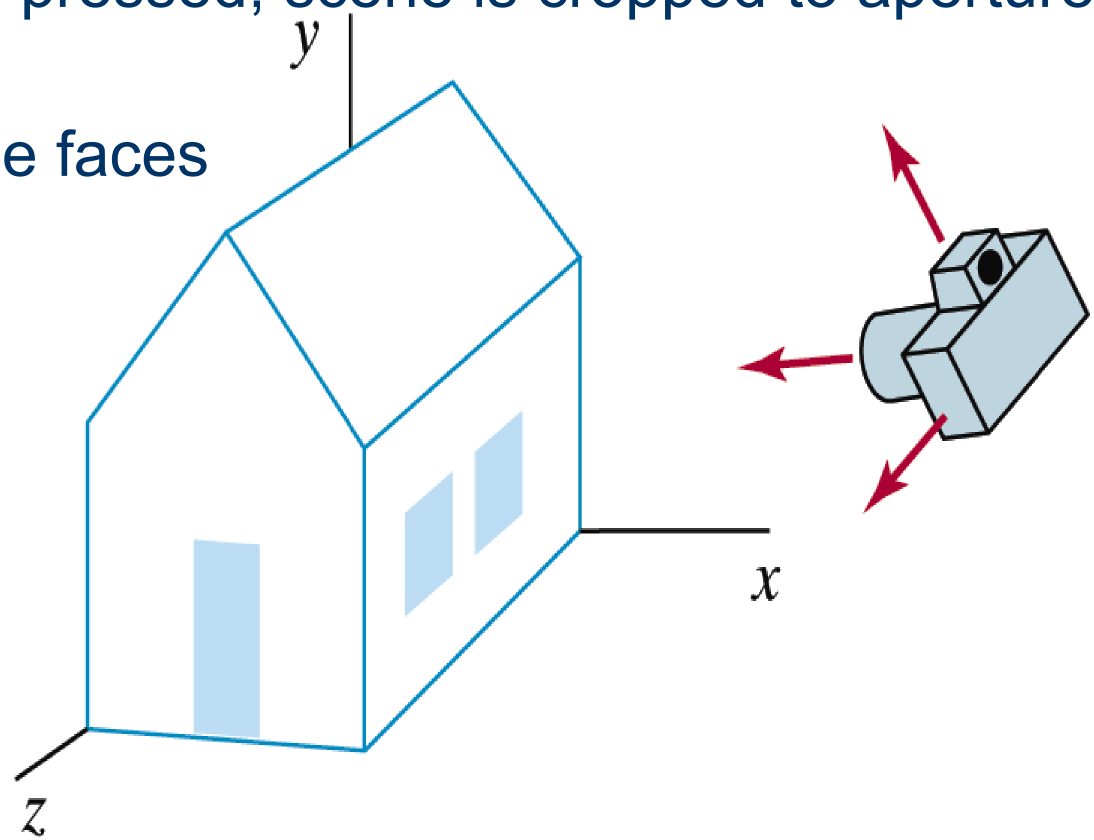
3D Viewing Transform

Amartya Kundu Durjoy
Lecturer
CSE (UGV)

Remember the big thing – Camera Analogy

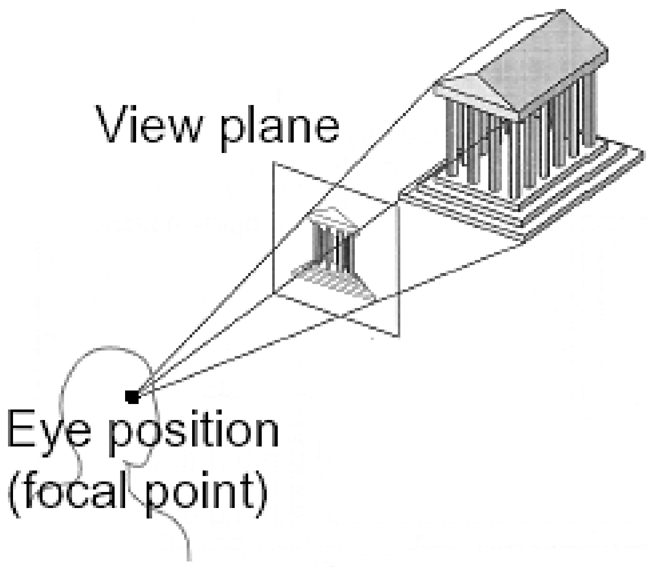
- 3D view of a scene is analogous to photographing
  - Need to position the camera
  - Need to decide camera orientation
  - When shutter is pressed, scene is cropped to aperture (window) size
  - Light from visible faces

projected into film



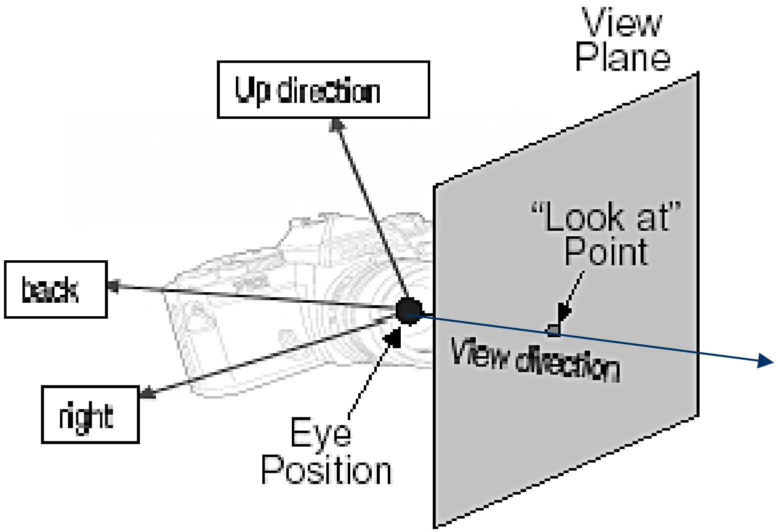
Camera Models

- The most common model is **pin-hole camera**
  - All captured light rays arrive along paths toward focal point without lens distortion (everything is in focus)
  - Sensor response proportional to radiance
  - Other models consider…
    - Depth of field
    - Motion blur
    - Lens distortion



Viewing Parameters

- Position
  - Eye position(px, py, pz)
- Orientation
  - View direction(dx, dy, dz)
  - Up direction(ux, uy, uz)
- Aperture
  - Field of view(xfov, yfov)
- Film plane
  - “look at” point
  - View plane normal



Transformations

P(x, y, z)

↓ 3D Object Coordinate

Model Transformation

↓ 3D World Coordinate

Viewing Transformation

↓ 3D Viewing Coordinate

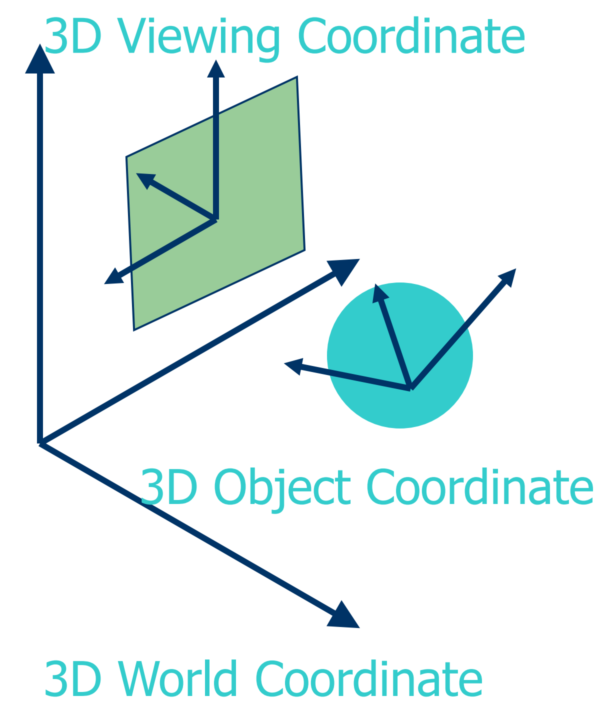
Projection Transformation

↓ 2D Projection Coordinate

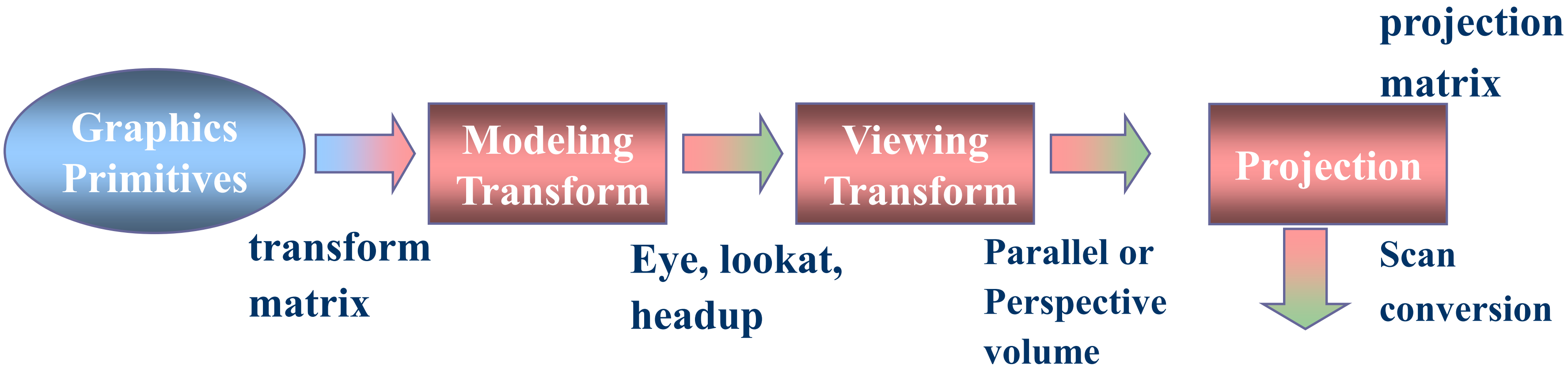
Viewport Transformation

↓ 2D Device Coordinate

p(x', y')

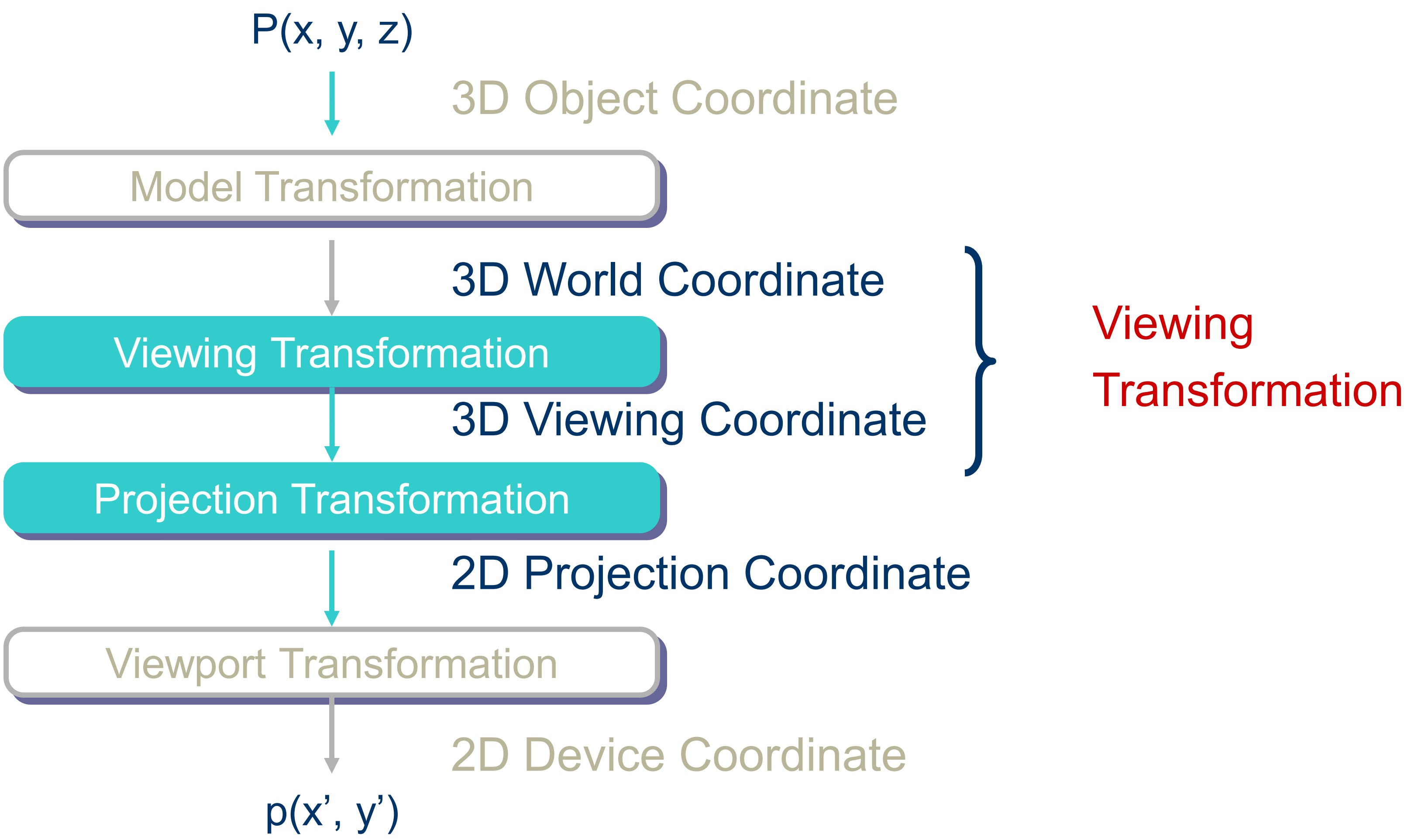


Graphics Pipeline



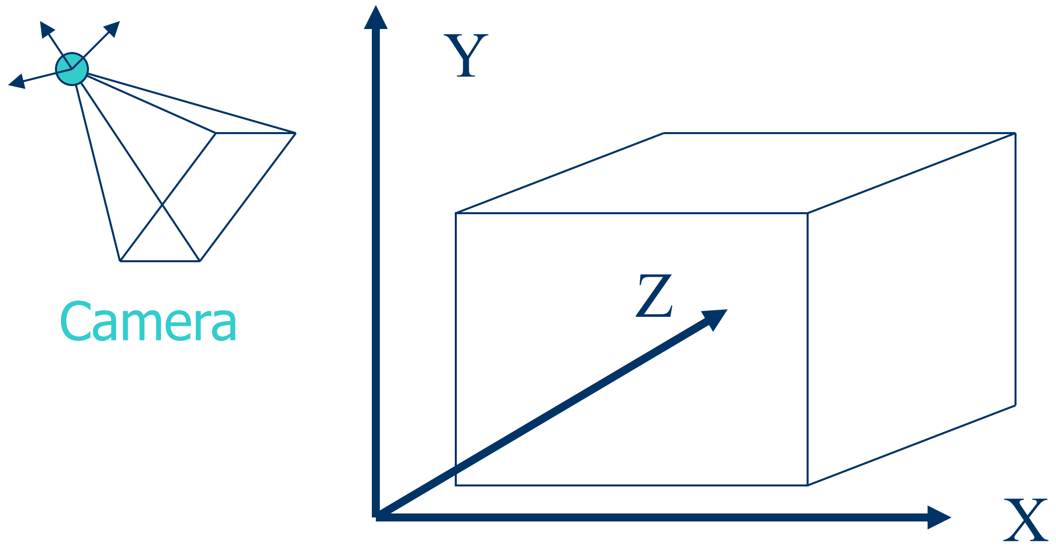
- Once scene has been modeled, world coordinates are converted to viewing coordinates
- Viewing coordinate system is used in graphics package for specifying viewer position and position of projection plane (film)

Viewing Transformation



Viewing Transformation

- Mapping from world to Viewing coordinates
  - Origin moves to eye position
  - Up vector maps to Y axis
  - Right vector maps to X axis



Viewing Coordinate system

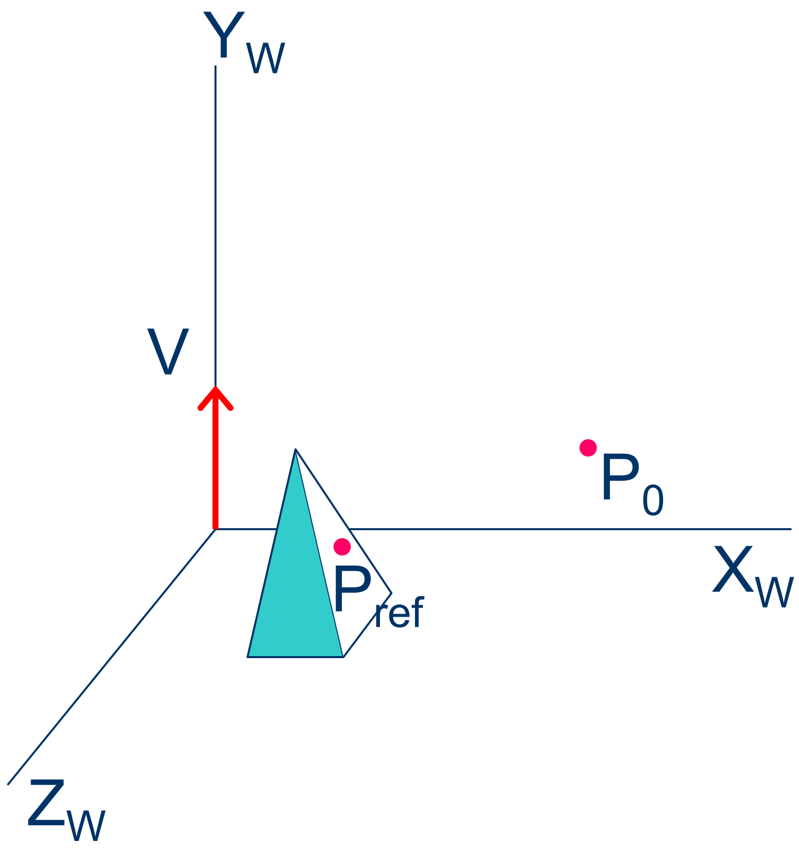
- Particular view of a scene is chosen by
  - Establishing Viewing Coordinate system
  - And then, a View plane (projection plane) is setup perpendicular to the viewing z_V axis.
- To establish Viewing Coordinate system (VCS)
  - First, a world coordinate position (**P_0**) is picked as **view reference point,** it serves as **origin** of VCS.
  - Then, positive direction of z_V axis is specified by specifying **view plane normal vector (N)**
    - Another world position **(P)** is picked – *look at*

Viewing Coordinate system

- And finally, up direction of view is specified by **view up vector (V).**
  - Generally specified by a world coordinate vector
  - It gives an appx. direction of y_V axis
  - Exact value of y_V axis then calculated by graphics package
- Using N and V, graphics package compute third vector U perpendicular to both N and V, to denote x_V axis
  - $U = V \times N$
- Vector denoting y_V axis can be computed by $N \times U$

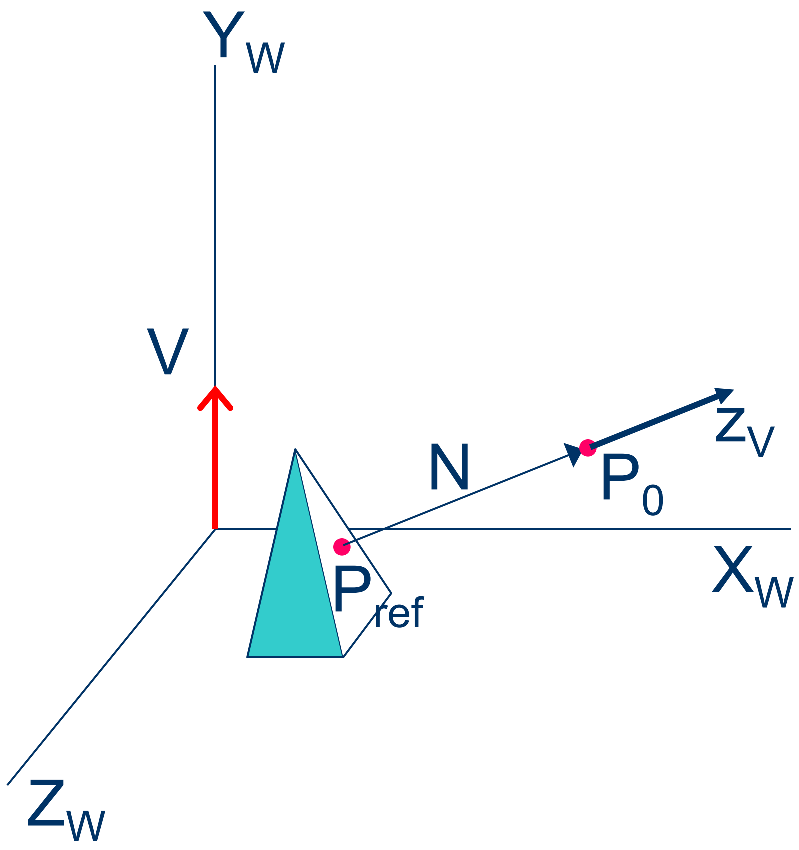
Viewing Coordinate system

- Specification of VCS :
  - **P_0** : *View* or *eye point*
  - **P_{ref}** : *Center* or *look-at point*
  - **V**: *View-up vector plane*



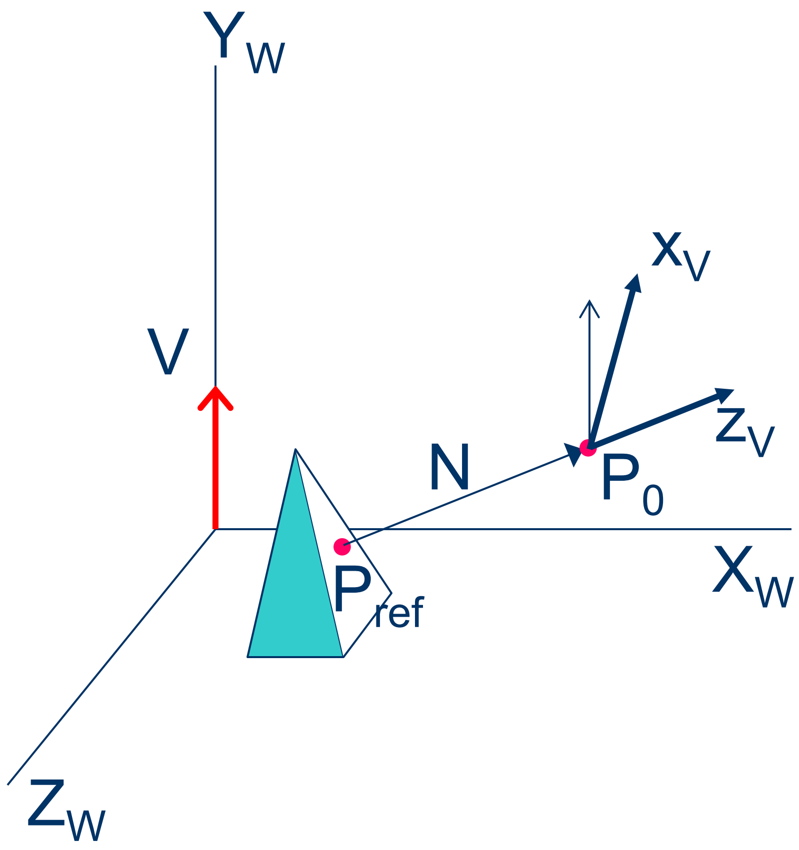
Viewing Coordinate system

- computation of VCS :
  - $z_V = N = P_0 - P$



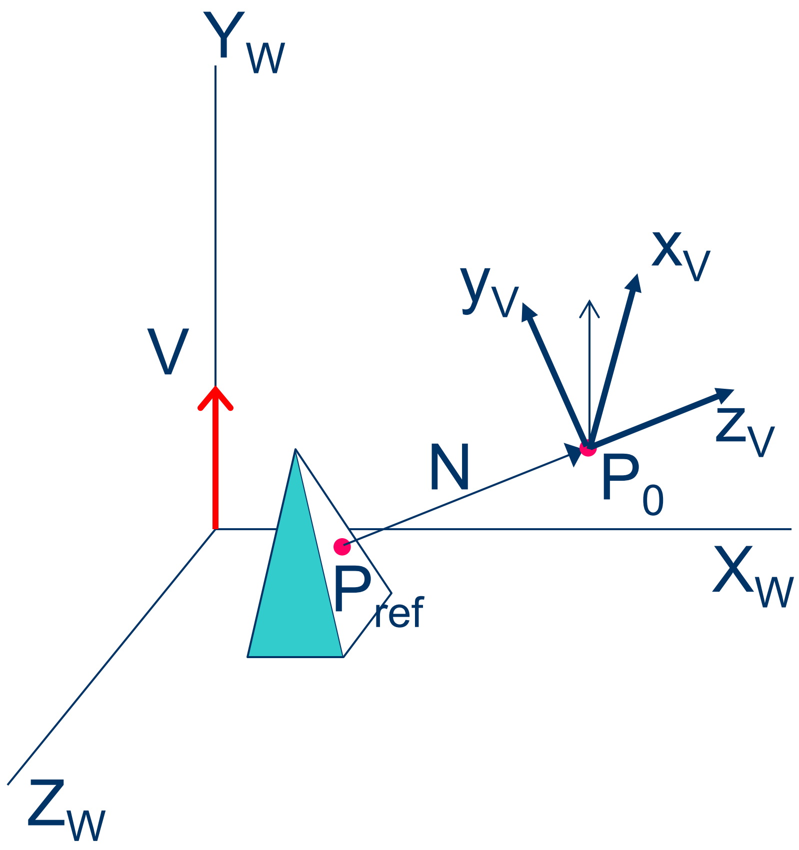
Viewing Coordinate system

- computation of VCS :
  - $z_V = N = P_0 - P$
  - $x_V = V \times N = U$



Viewing Coordinate system

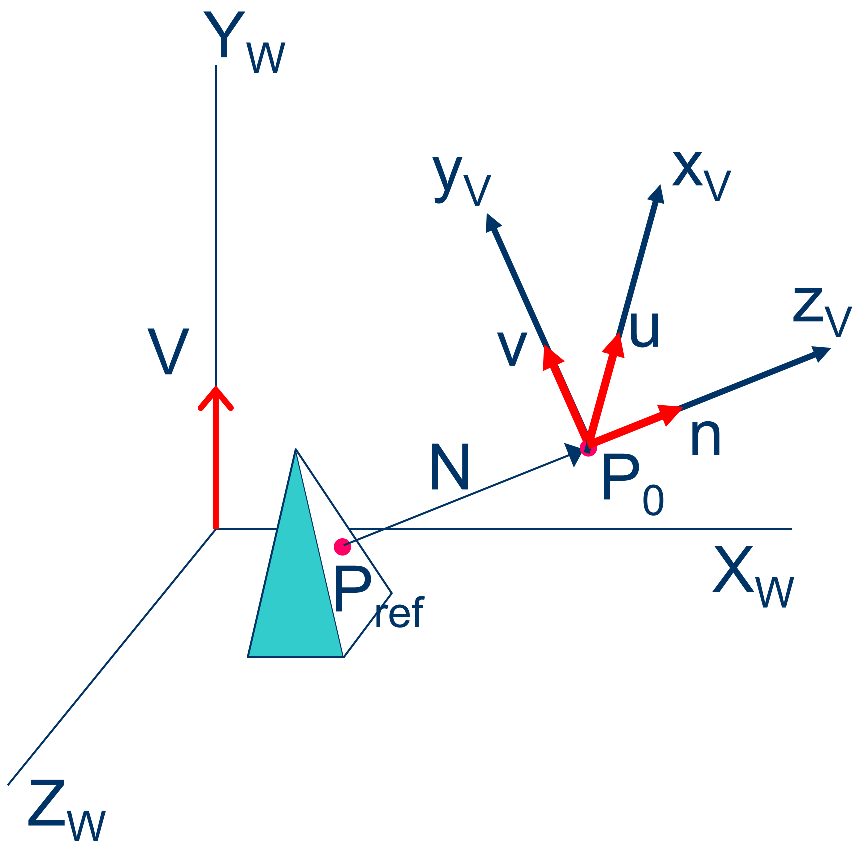
- computation of VCS :
  - $z_V = N = P_0 - P$
  - $x_V = V \times N = U$
  - $y_V = N \times U$



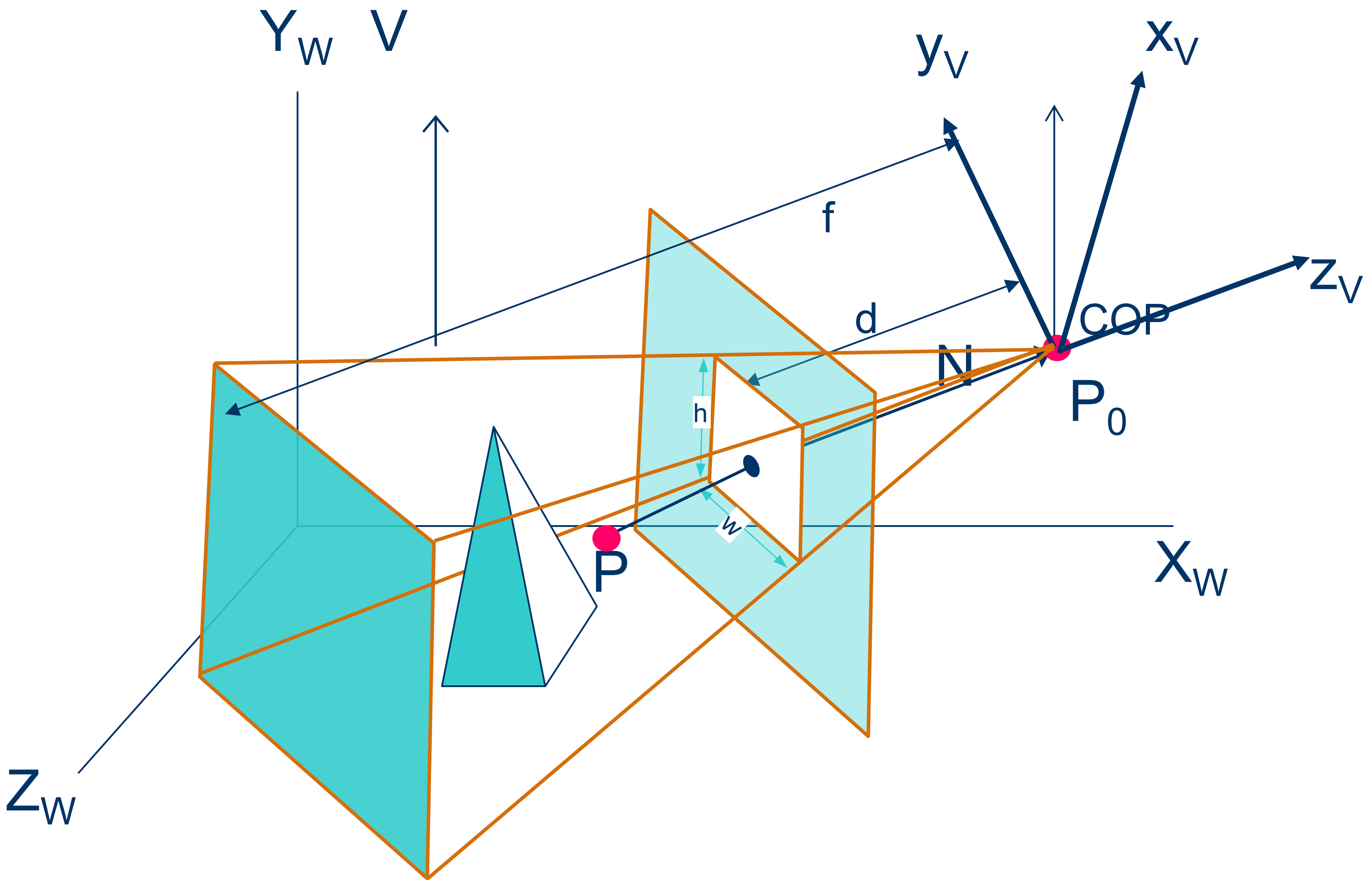
Viewing Coordinate system

- $z_V = N = P_0 - P$
- $x_V = V \times N = U$
- $y_V = N \times U$

- All these computations are actually carried out with unit vectors **u**, **v**, **n**
- And the system is called **uvn** system
- Graphics package allow user to choose view plane position along **n** axis
  - By specifying **view plane distance** from **viewing origin**



WCS, VCS and projection



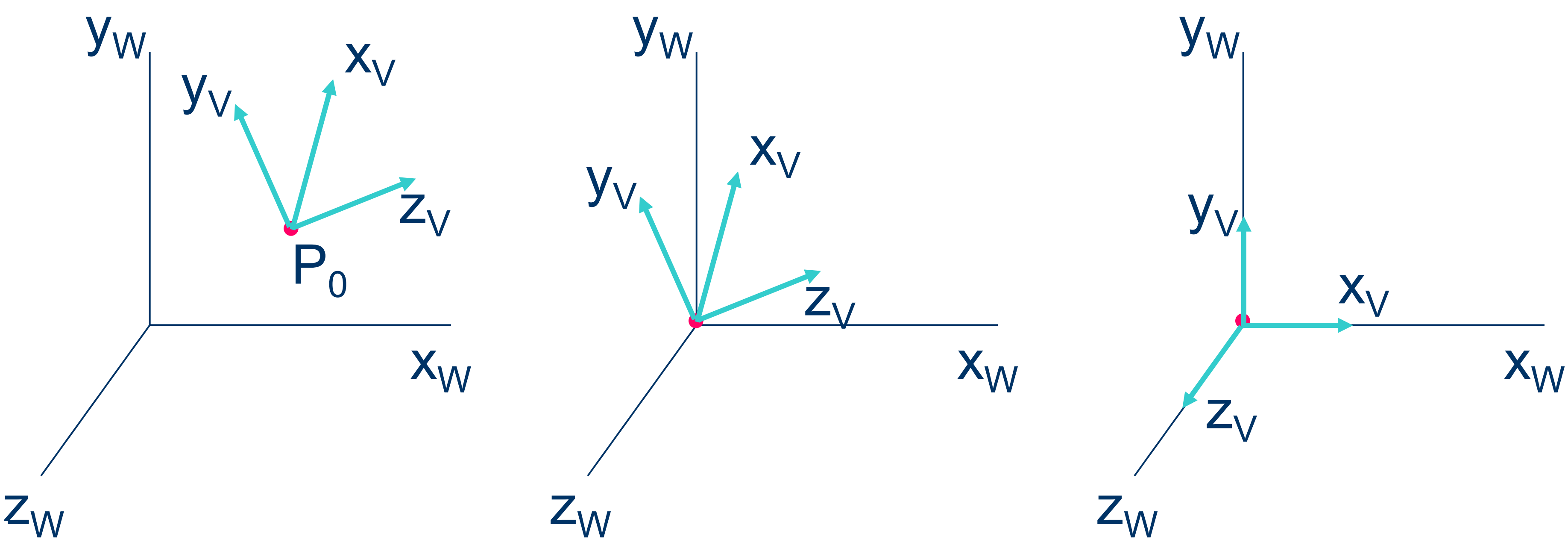
WC to VC transformation

- Before projection, object description must be converted from world to viewing system
- Conversion is eqv. to a transformation that superimposes the viewing system to world system
- This transformation can be achieved by basic transformations
  1. Translate view reference point $P_0(x_0,y_0,z_0)$ to world origin
  2. Apply rotation to align x_V, y_V, z_V axes with x_W, y_W, z_W axes, respectively

WC to VC transformation

- The translation matrix is

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



Translation from world to viewing coordinates

- Rotation matrix **R** can be found by up to three world coordinate axis rotations $R_z . R_y . R_x = R$
  - R_X places N or z_V to $x_W z_W$ plane
  - R_y aligns z_V to z_W axis
  - R_z aligns y_V to y_W axis

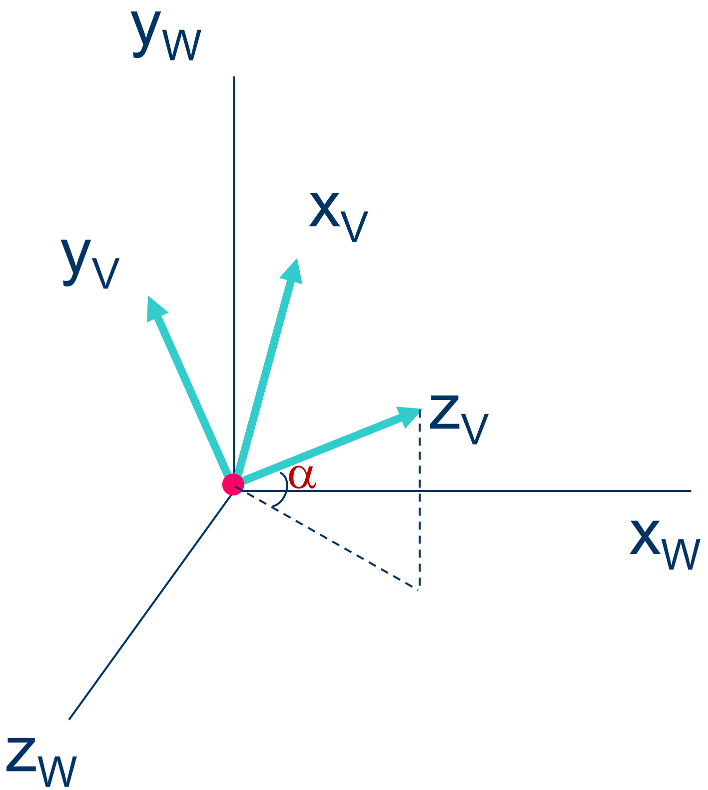
$$n = \frac{N}{|N|} = (n_x, n_y, n_z)$$

$$u = \frac{V \times N}{|V \times N|} = (u_x, u_y, u_z)$$

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

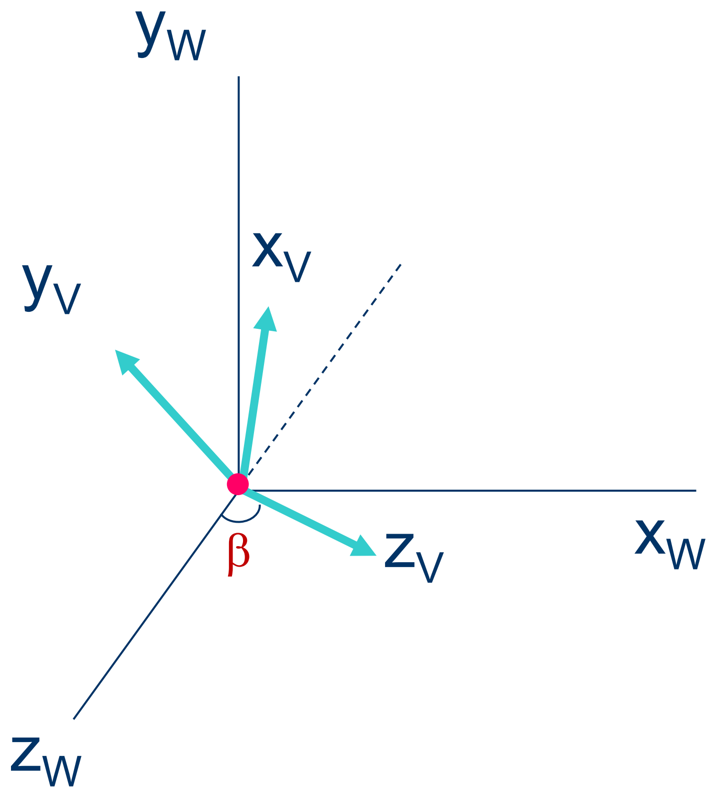
- $M_{WC,VC} = R.T$

Composite R



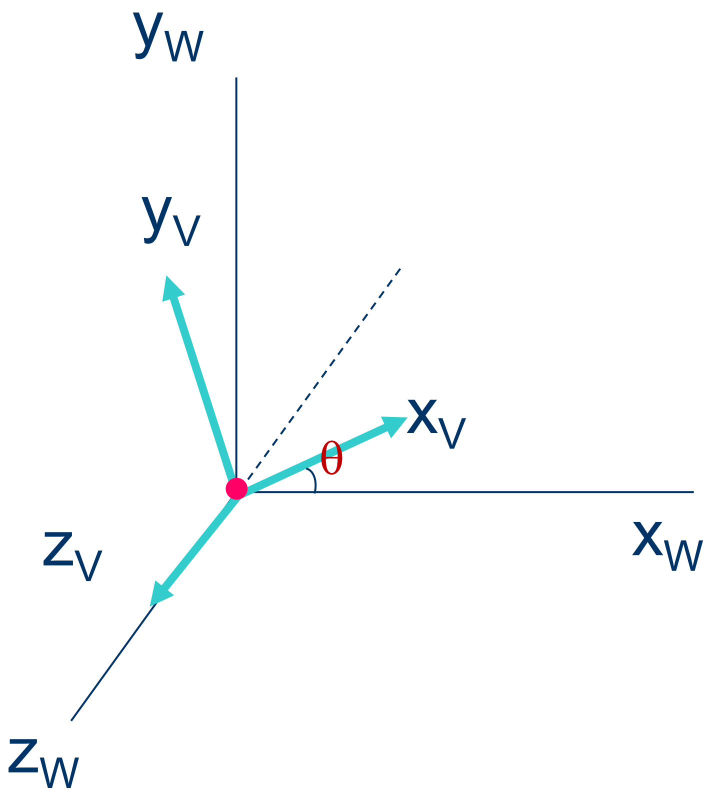
- Rotate α about X_W, R_i

Composite R



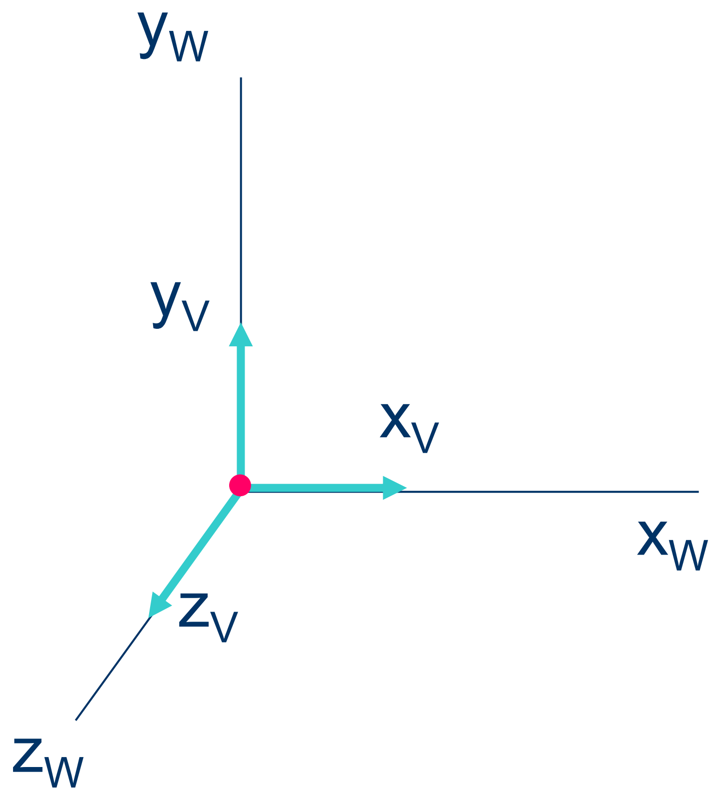
- Rotate α about X_W, R_i
  - Places Z_v in XZ plane
- Rotate β about Y_W, R_j

Composite R



- Rotate α about X_W, R_i
  - Places Z_v in XZ plane
- Rotate β about Y_W, R_j
  - Aligns Z_v with Z_w
  - X_v , Y_v are in XY plane
- Rotate θ about Z_W, R_k

Composite R



- Rotate α about X_W, R_i
  - Places Z_v in XZ plane
- Rotate β about Y_W, R_j
  - Aligns Z_v with Z_w
  - X_v , Y_v are in XY plane
- Rotate θ about Z_W, R_k

Finding composite R

$$R = R_k R_j R_i = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta\cos\beta & -\sin\theta & \cos\theta\sin\beta & 0 \\ \sin\theta\cos\beta & \cos\theta & \sin\theta\sin\beta & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta\cos\beta & -\sin\theta\cos\alpha + \sin\alpha\cos\theta\sin\beta & \sin\alpha\sin\theta + \cos\alpha\cos\theta\sin\beta & 0 \\ \sin\theta\cos\beta & \cos\theta\cos\alpha + \sin\alpha\sin\theta\sin\beta & -\sin\alpha\cos\theta + \cos\alpha\sin\theta\sin\beta & 0 \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

R is Rigid−body Transform – i.e. orthogonal matrix

i) *rows and columns* are unit vectors

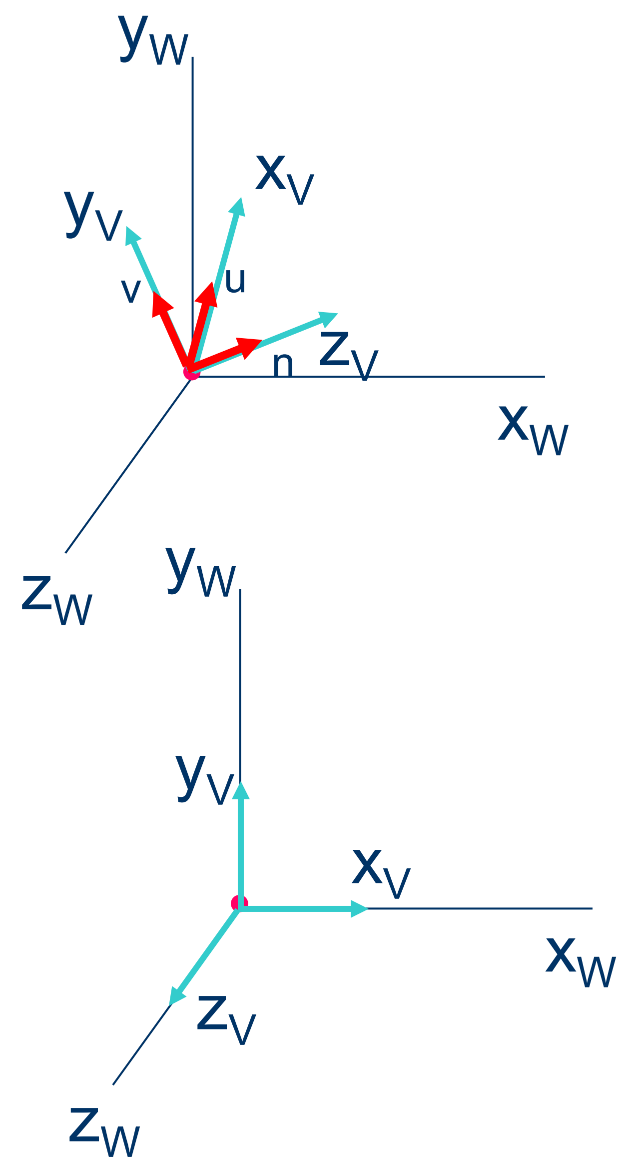
ii) *vectors* are perpendicular *to each other*

Finding composite R (another way)

Let R be

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} R_{1x} & R_{1y} & R_{1z} \\ R_{2x} & R_{2y} & R_{2z} \\ R_{3x} & R_{3y} & R_{3z} \end{bmatrix}$$

Note: $R_{1x} \Rightarrow x$ component of vextor $\overrightarrow{R_1}$



R aligns unit vectors **u, v, n** to **i, j, k** respectably

Finding R

we can write

$$\Rightarrow R \cdot u = \hat{i} \qquad \Rightarrow R^T \cdot \hat{i} = \mathrm{u} \quad \left[\because R^{-1} = R^T\right]$$

$$\Rightarrow \begin{bmatrix} R_{1x} & R_{2x} & R_{3x} \\ R_{1y} & R_{2y} & R_{3y} \\ R_{1z} & R_{2z} & R_{3z} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathrm{u}$$

$$\Rightarrow \begin{bmatrix} R_{1x} \\ R_{1y} \\ R_{1z} \end{bmatrix} = \mathrm{u} = \overrightarrow{R_1}$$

similarly we can find that

$$\overrightarrow{R_2} = \hat{v} \text{ and } \overrightarrow{R_3} = n$$

WC to VC transformation

- Rotation matrix **R** can be found by
  - Calculate unit vectors u,v,n
  - Form composite R directly

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

$$u = \frac{V \times N}{|V \times N|} = (u_x, u_y, u_z)$$

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

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

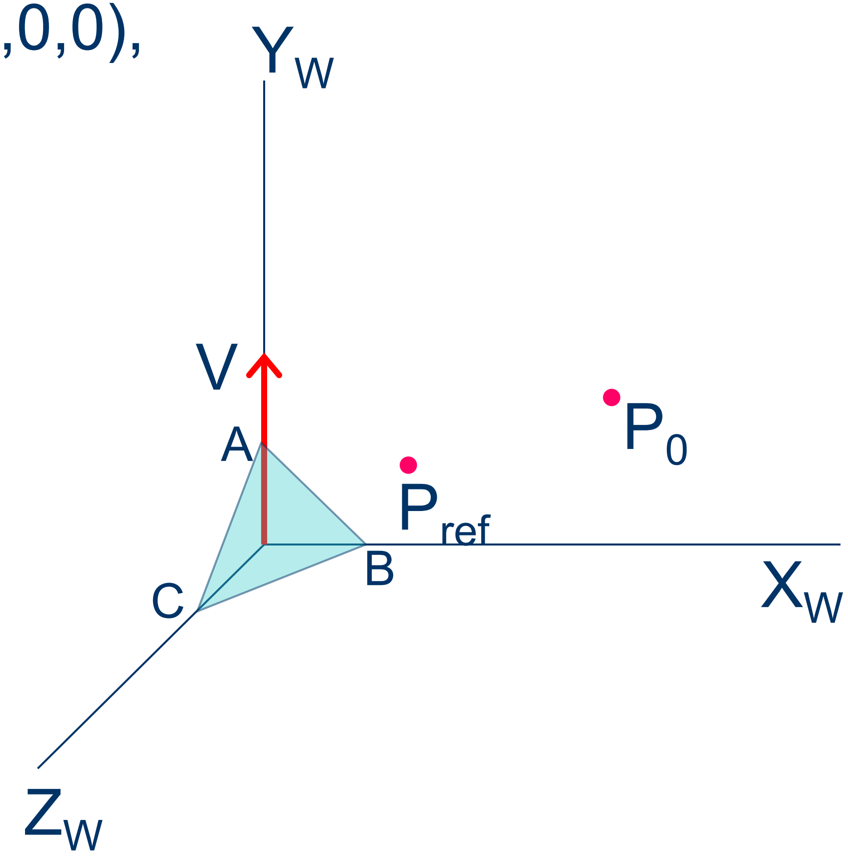
- $M_{WC,VC}$ =R.T

$$= \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z & dx \\ v_x & v_y & v_z & dy \\ n_x & n_y & n_z & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{dx = -P_0.u} \qquad \boldsymbol{dy = -P_0.v} \qquad \boldsymbol{dx = -P_0.n}$$

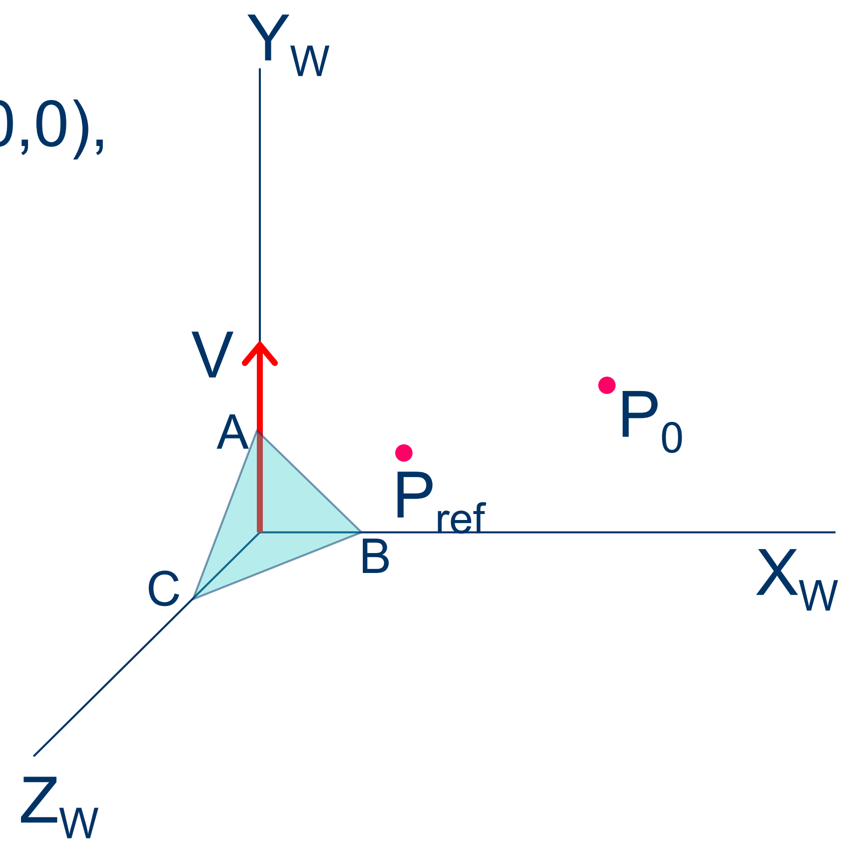
example

- Find the VCS values for the
  - Object: triangle A(0,1,0), B(1,0,0), C(0,0,1)
- when
  - Eye: (3,2,3)
  - Look at point: (2,1,2)
  - Up vector (0,1,0)



example

- Find the VCS values for the
  - Object: triangle A(0,1,0), B(1,0,0), C(0,0,1)
- when
  - Eye: (3,2,3)
  - Look at point: (2,1,2)
  - Up vector (0,1,0)



$N = P_0 - P_{ref} = (1,1,1)$

$$V \times N = (1, 0, -1)$$

$$v = n \times u$$

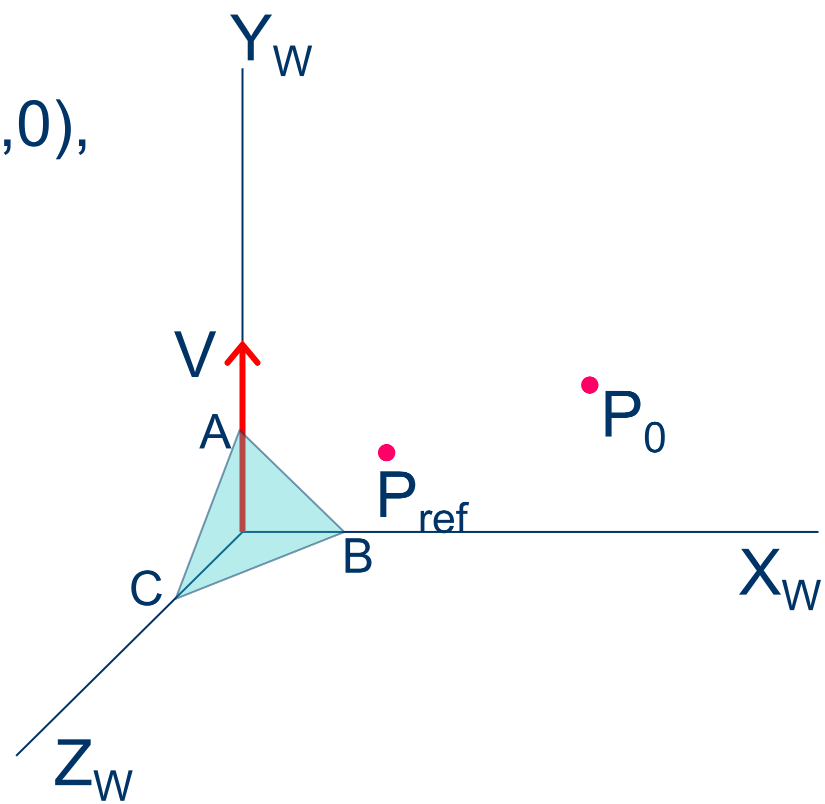
$$n = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right)$$

$$u = \left(\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}} \right)$$

$$= \left(-\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}, -\frac{1}{\sqrt{6}} \right)$$

example

- Find the VCS values for the
  - Object: triangle A(0,1,0), B(1,0,0), C(0,0,1)
- when
  - Eye: (3,2,3)
  - Look at point: (2,1,2)
  - Up vector (0,1,0)



$$u = \left(\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}}\right)$$

$$v = \left(-\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}, -\frac{1}{\sqrt{6}}\right)$$

$$n = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$$

$$\boldsymbol{dx = -P_0.u = 0}$$

$$\boldsymbol{dy = -P_0.v} = \frac{2}{\sqrt{6}}$$

$$\boldsymbol{dx = -P_0.n} = \frac{-8}{\sqrt{3}}$$

DIY

- Find the VCS values for the
  - Point: A(1,0.5,1)
- when
  - Eye: (2,2,3)+last 3 digit of your roll
  - Look at point: (2,1,1)
  - Up vector (0,1,0)

Acknowledgements
cgvr.korea.ac.kr

DIY

10. Write down the 4×4 rotation matrix that takes the orthonormal 3D vectors $\mathbf{u} = (x_u, y_u, z_u)$, $\mathbf{v} = (x_v, y_v, z_v)$, and $\mathbf{w} = (x_w, y_w, z_w)$, to orthonormal 3D vectors $\mathbf{a} = (x_a, y_a, z_a)$, $\mathbf{b} = (x_b, y_b, z_b)$, and $\mathbf{c} = (x_c, y_c, z_c)$, So $M\mathbf{u} = \mathbf{a}$, $M\mathbf{v} = \mathbf{b}$, and $M\mathbf{w} = \mathbf{c}$.