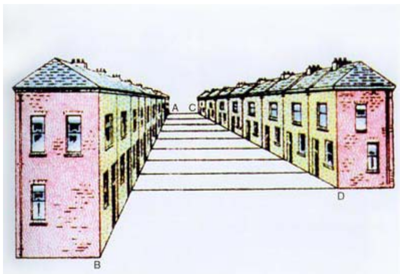
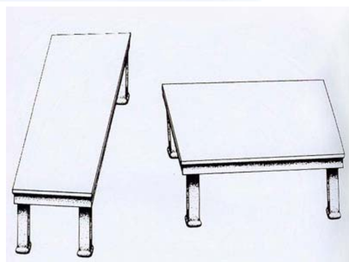
Projection Views





Content

- Coordinate systems
- Orthographic projection
- (Engineering Drawings)

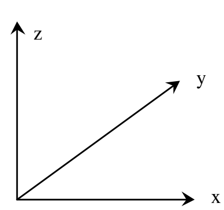
Graphical Coordinator Systems

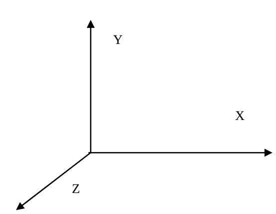
A coordinate system is needed to input, store and display model geometry and graphics.

Four different types of <u>coordinate systems</u> are used in a CAD system at different stages of geometric modeling and for different tasks.

Model (or World, Database) Coordinator System

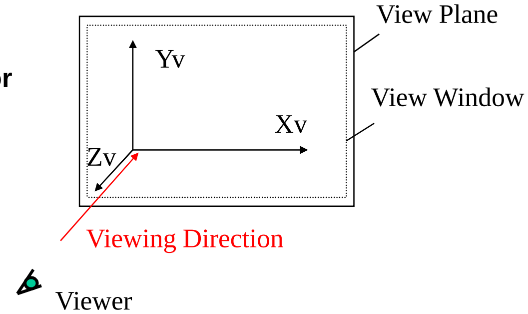
- **The reference space of the model with respect to which all of the geometrical data is stored.**
- **It is a Cartesian system which forms the default coordinate system used by a software system.**



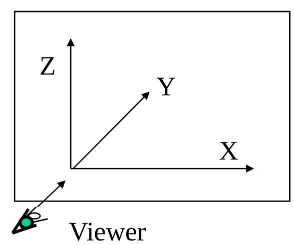


Viewing Coordinate System

Textbook setup - for Parallel Projection, the viewer is at infinity.

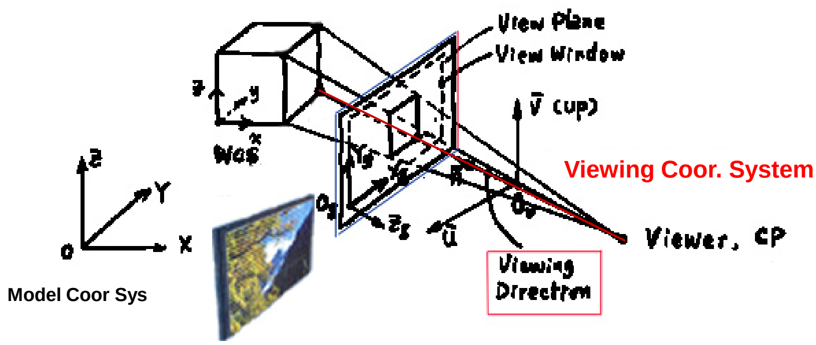


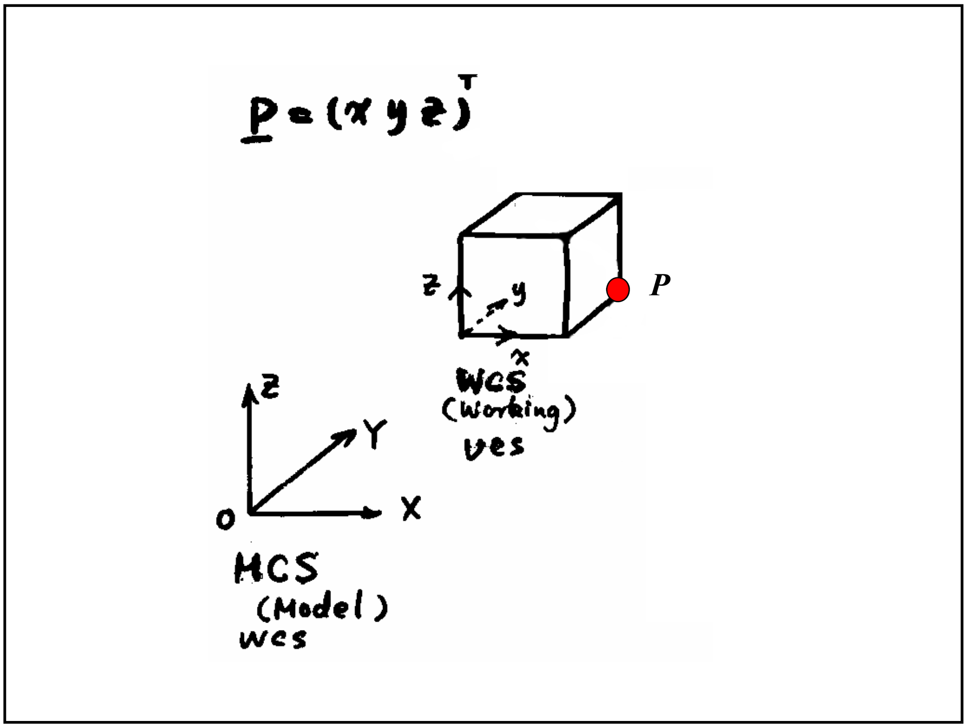
AutoCAD Default Setup

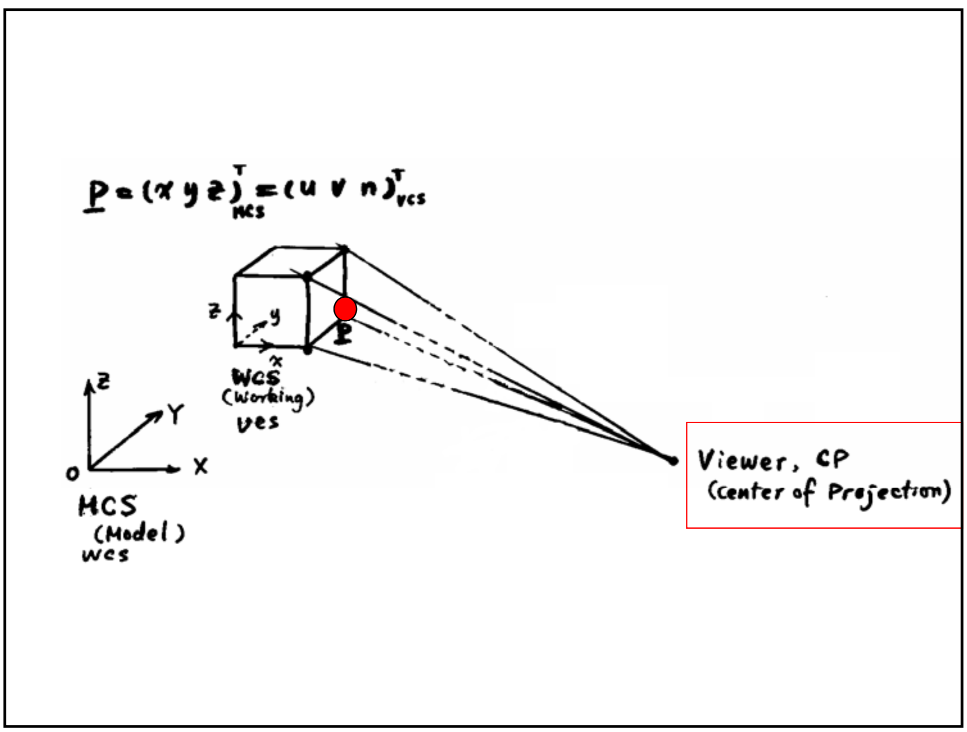


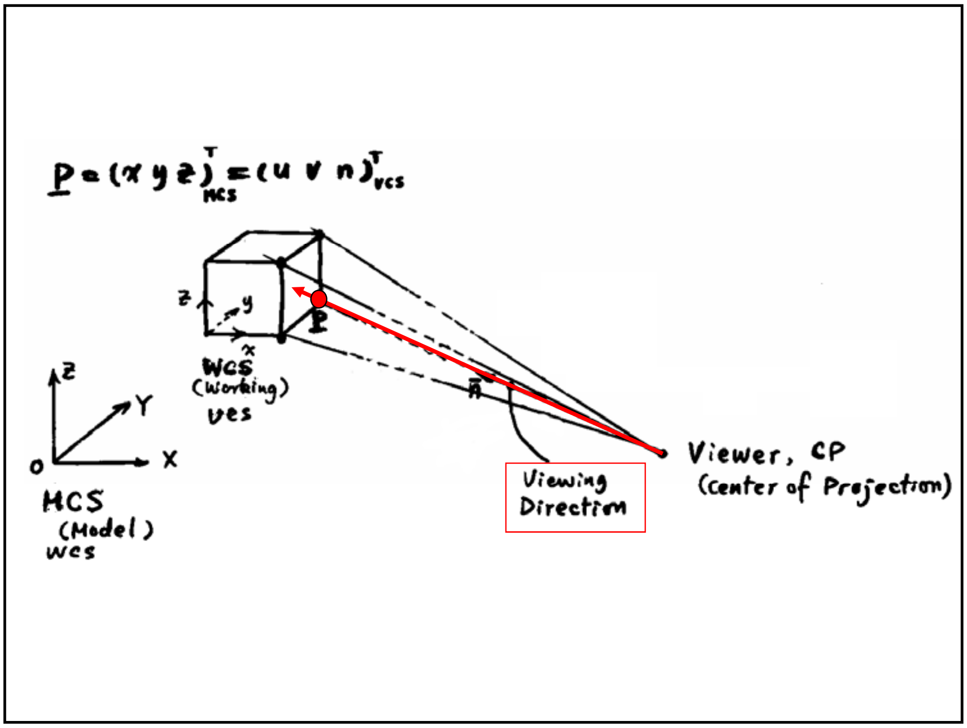
Viewing Coordinate System (VCS)

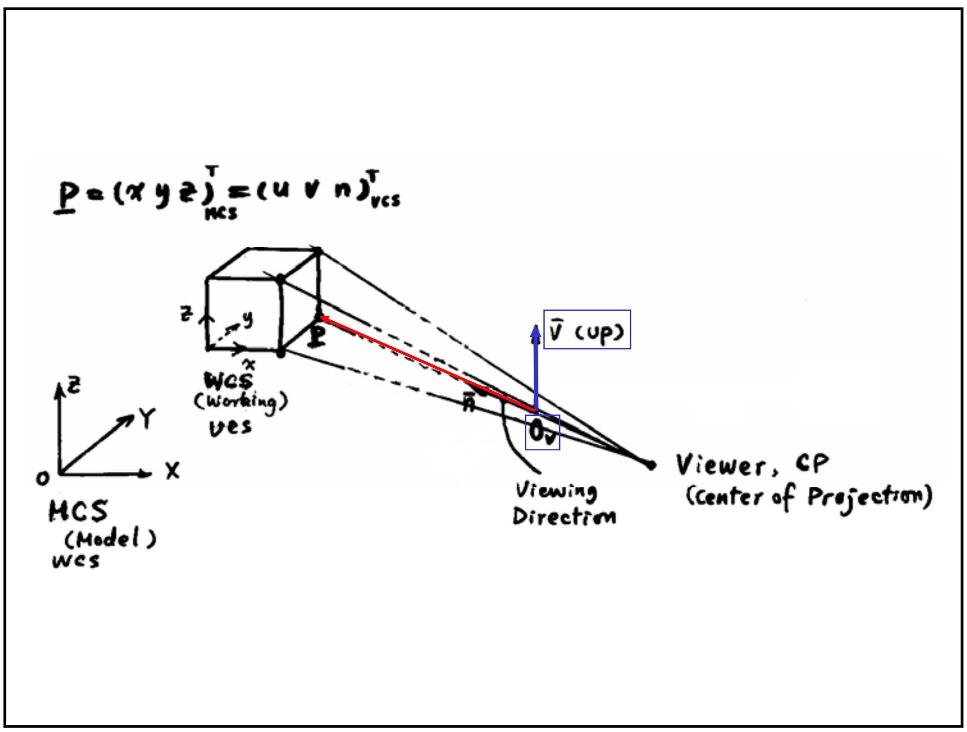
A 3-D Cartesian coordinate system (right hand of left hand) in which a projection of the modeled object is formed. VSC will be discussed in detail under Perspective or Parallel Projections.

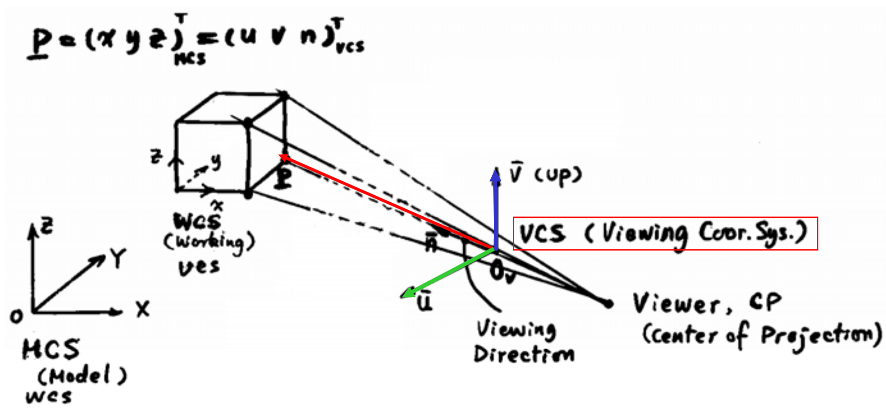


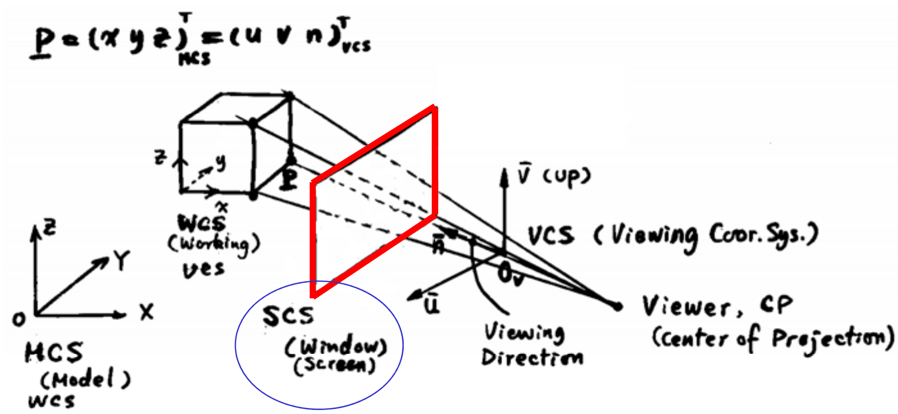

P = (x y z)^T
z
y
x
P
WCS
(Working)
VCS
Z
Y
X
O
MCS
(Model)
WCS

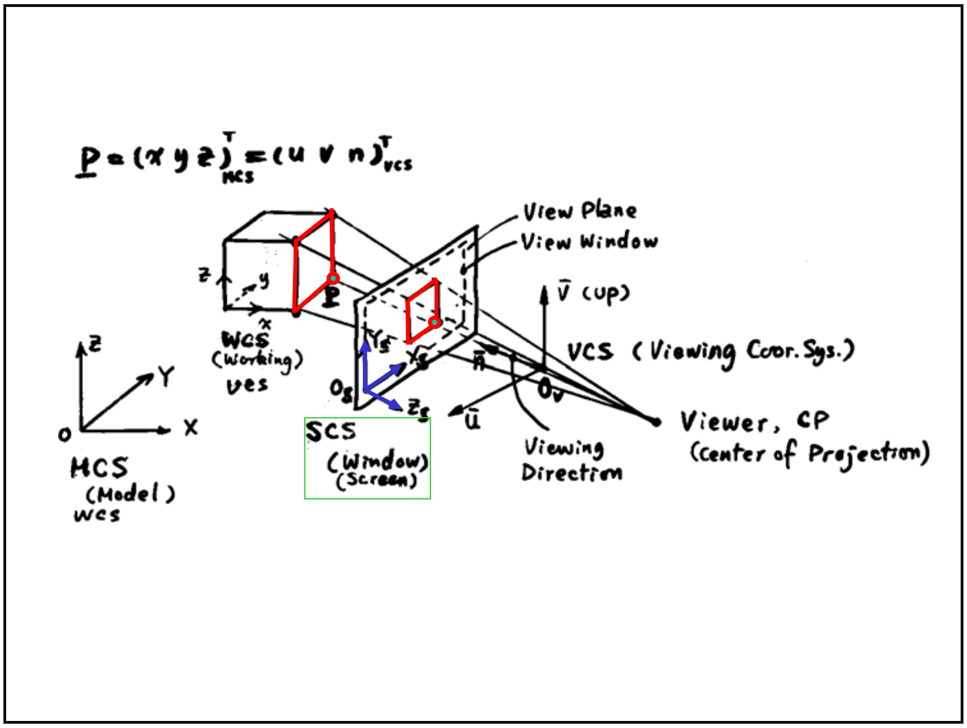

P = (x y z)^T_MCS = (u v n)^T_VCS
z
y
x
P
WCS
(Working)
VCS
Z
Y
X
O
MCS
(Model)
WCS
Viewer, CP
(center of Projection)


P = (x y z)^T_HCS = (u v n)^T_VCS
z
y
x
WCS
(Working)
VCS
Z
Y
X
O
HCS
(Model)
WCS
P
n
Viewing Direction
Viewer, CP
(Center of Projection)

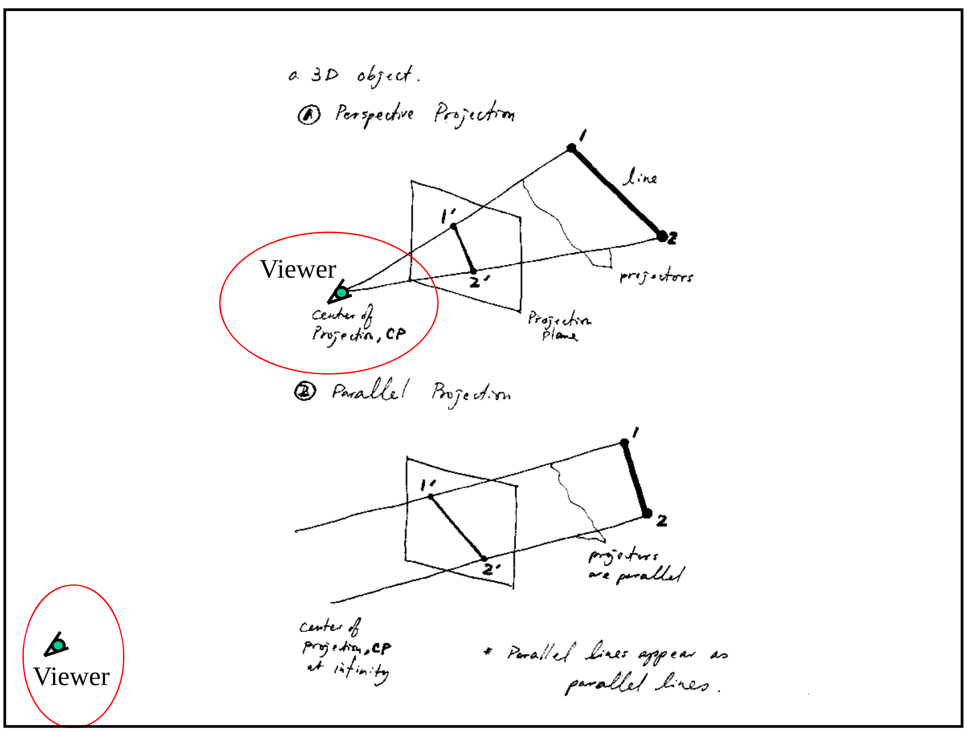

P = (x y z)^T_HCS = (u v n)^T_VCS
z
y
x
WCS
(Working)
VCS
Z
Y
X
O
HCS
(Model)
WCS
P
V (UP)
n
O_V
Viewing Direction
Viewer, CP
(Center of Projection)


P = (x y z)^T_HCS = (u v n)^T_VCS
v̄ (up)
VCS (Viewing Coor. Sys.)
WCS
(Working)
VCS
HCS
(Model)
WCS
n̄
O_v
ū
Viewing Direction
Viewer, CP
(Center of Projection)


P = (x y z)^T_HCS = (u v n)^T_VCS
v̄ (up)
VCS (Viewing Coor. Sys.)
WCS
(Working)
VCS
HCS
(Model)
WCS
SCS
(Window)
(Screen)
n̄
O_v
ū
Viewing Direction
Viewer, CP
(Center of Projection)



View Plane
View Window
WCS
(Working)
VCS (Viewing Coor. Sys.)
Viewer, CP
(center of Projection)
SCS
(Window)
(Screen)
MCS
(Model)
Viewing
Direction



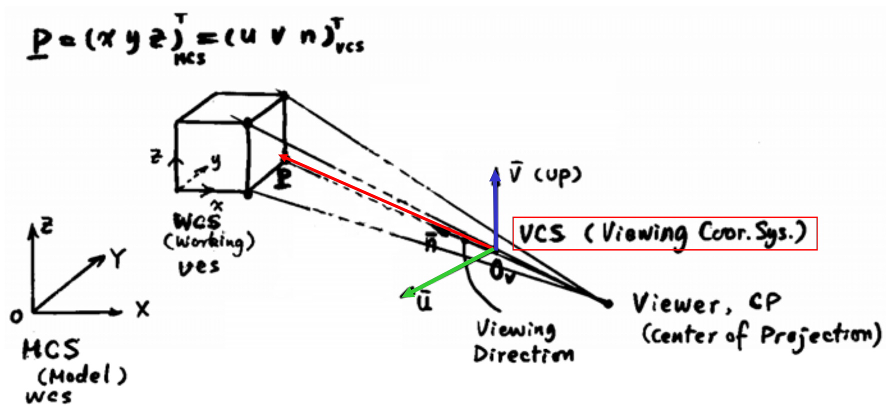
a 3D object.
Perspective Projection
line
Viewer
projectors
Center of
Projection, CP
Projection
Plane
Parallel Projection
projectors
are parallel
center of
projection, CP
at infinity
* Parallel lines appear as
parallel lines.
Viewer

Parallel Projection

- **Preserve actual dimensions and shapes of objects**
- **Preserve parallelism**
- **Angles preserved only on faces parallel to the projection plane**
- **Orthographic projection is one type of parallel projection**

Perspective Projection

- **Doesn't preserve parallelism**
- **Doesn't preserve actual dimensions and angles of objects, therefore shapes deformed**
- **Popular in art (classic painting); architectural design and civil engineering.**
- **Not commonly used in mechanical engineering**



Geometric Transformations for Generating Projection View

Set Up the Viewing Coordinate System (VCS)

i) Define the view reference point $\mathbf{P} = (P_x,\ P_y,\ P_z)^T$

ii) Define the line of the sight vector $\vec{n}$ **(normalized)**

$$\vec{n} = (N_x,\ N_y,\ N_z)^T \quad \text{and} \quad N_x^2 + N_y^2 + N_z^2 = 1$$

iii) Define the "up" direction

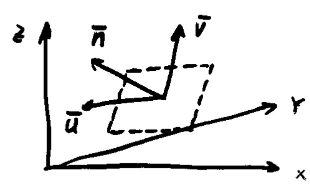
$$\vec{V} = (V_x,\ V_y,\ V_z)^T \perp \vec{n}, \qquad \vec{V} \cdot \vec{n} = 0$$

This also defines an orthogonal vector, $\vec{u} = \vec{V} \times \vec{n}$

$(\vec{u},\ \vec{V},\ \vec{n})$ **forms the viewing coordinates**

Define the View Window in

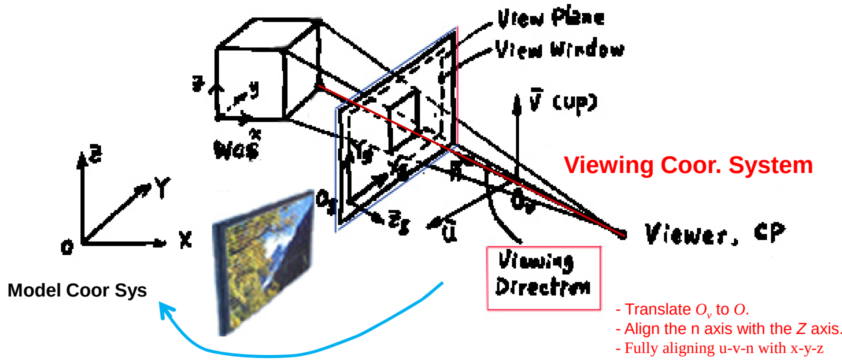
$\bar{u} - \bar{v} - \bar{n}$ **coordinates**



Generating Parallel Projection

Problem: for a given computer model, we know its x-y-z coordinates in MCS; and we need to find its u-v-n coordinates in VCS and X_s-Y_s in WCS.

Getting the u-v-n coordinates of the objects by transforming the objects and u-v-n coordinate system together to fully align u-v-n with x-y-z axes, then drop the n (the depth) component to get X_s and Y_s



Generating Parallel Projection (1)

First transform coordinates of objects into the *u-v-n* coordinates (VCS), then drop the n component. (*n* is the depth)

i.e. Overlapping $u - v - n$ **with** $x - y - z$

i) Translate O_v **to** O**.**

$$[D] = \begin{bmatrix} 1 & 0 & 0 & -0_{vx} \\ 0 & 1 & 0 & -0_{vy} \\ 0 & 0 & 1 & -0_{vz} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

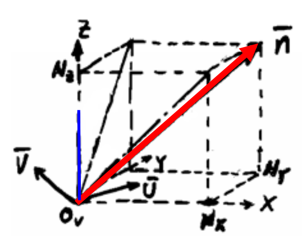
ii) Align the $\vec{n}$ **axis with the** Z **axis.**

The procedure is identical to the transformations used to prepare for the rotation about an axis.

$$A = N_x, \quad B = N_y, \quad C = N_z$$

$$L = \sqrt{N_x^2 + N_y^2 + N_z^2}$$

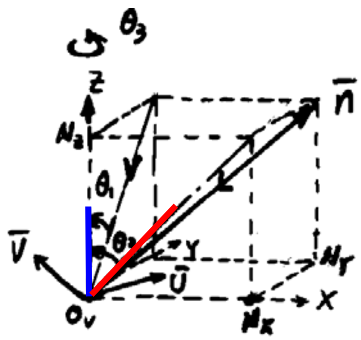
$$V = \sqrt{N_y^2 + N_z^2}$$



Generating Parallel Projection (2)

Rotating θ_1 about *X*: $[R]_x$; and Rotating θ_2 about *Y*: $[R]_y$

Fully aligning u-v-n with x-y-z



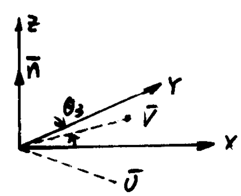
Then, rotate θ_3 about the *Z* axis to align $\bar{u}$ with *X* and $\bar{v}$ with *Y*

Generating Parallel Projection (3)

Rotate θ_3 about the *Z* axis to align $\bar{u}$ with *X* and $\bar{v}$ with *Y*

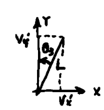
At this point, $\bar{V}$ is given by $(V'_x,\ V'_y,\ 0)^T$ where

$$\begin{pmatrix} V'_x \\ V'_y \\ 0 \\ 1 \end{pmatrix} = [R_y][R_x][Do_v, o]\begin{pmatrix} V_x \\ V_y \\ V_z \\ 1 \end{pmatrix}$$



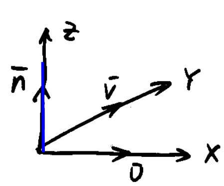
We need to rotate by an angle θ_3 about the *Z* axis

$$L = \sqrt{V'^2_x + V'^2_y}, \quad \sin\theta_3 = \frac{V'_x}{L}, \quad \cos\theta_3 = \frac{V'_y}{L}$$



$$[R_z] = \begin{bmatrix} V'_y/L & -V'_x/L & 0 & 0 \\ V'_x/L & V'_y/L & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Result:



Generating Parallel Projection (4)

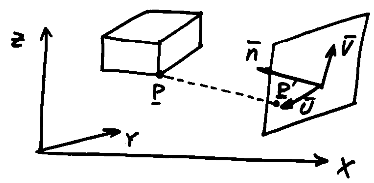
Drop the n coordinate

$$[D_n]=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & \mathbf{0} & 0\\ 0 & 0 & 0 & 1\end{bmatrix}, \quad \begin{pmatrix}u\\ V\\ \mathbf{0}\\ 1\end{pmatrix} = [D_n]\begin{pmatrix}u\\ V\\ n\\ 1\end{pmatrix}$$

In summary, to project a view of an object on the UV plane, one needs to transform each point on the object by:

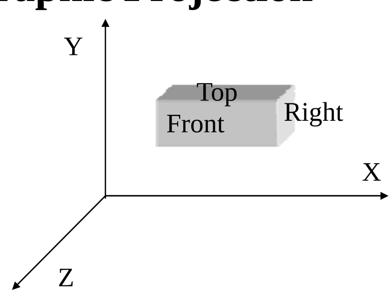
$$[T]=[D_n][R_z][R_y][R_x][D_{o_v,o}]$$

$$\mathbf{P}' = \begin{pmatrix}u\\ V\\ 0\\ 1\end{pmatrix} = [T]\mathbf{P} = [T]\begin{pmatrix}x\\ y\\ z\\ 1\end{pmatrix}$$



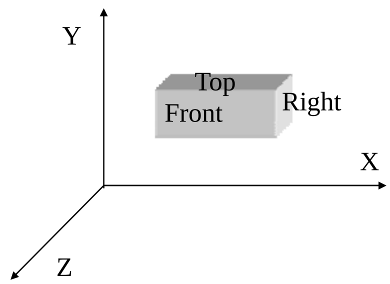
Note: The inverse transforms are not needed! We don't want to go back to x - y - z coordinates.

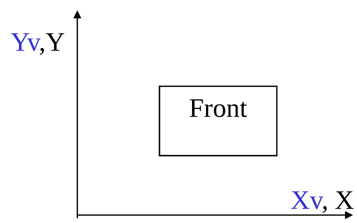
Orthographic Projection



- Projection planes (Viewing planes) are perpendicular to the principal axes of the MCS of the model
- The projection direction (viewing direction) coincides with one of the MCS axes

Geometric Transformations for Generating **Orthographic Projection (Front View)**

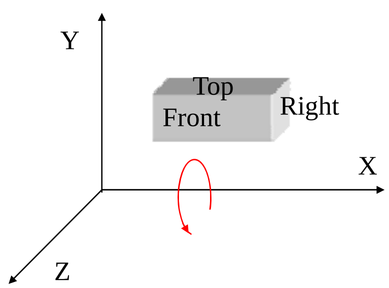


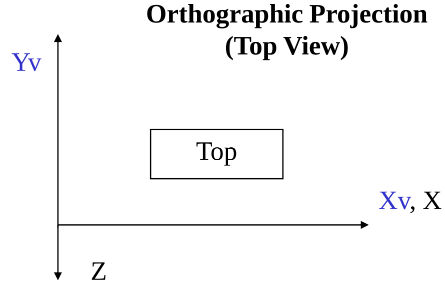


$$Pv = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P$$

Drop Z

Geometric Transformations for Generating **Orthographic Projection (Top View)**

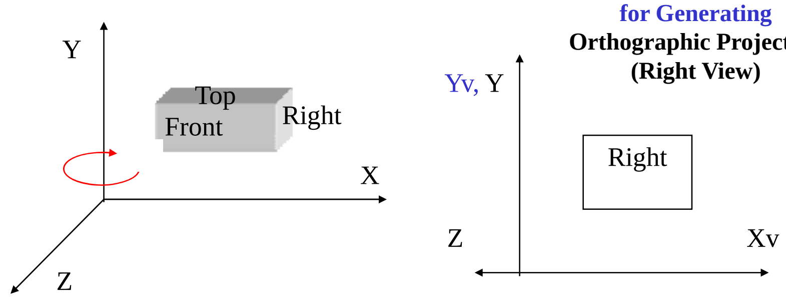




$$P_v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(90°) & -\sin(90°) & 0 \\ 0 & \sin(90°) & \cos(90°) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P$$

Drop Z $[R]_x^{90}$

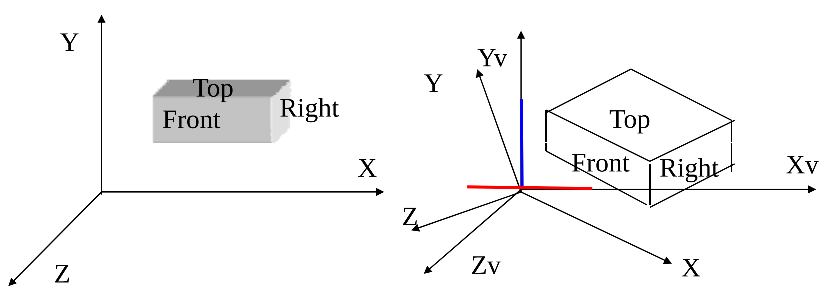
Geometric Transformations for Generating Orthographic Projection (Right View)



$$P_v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-90°) & 0 & \sin(-90°) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-90°) & 0 & \cos(-90°) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P$$

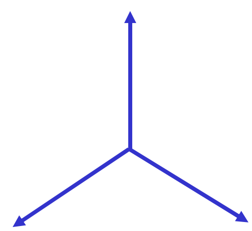
Drop Z $[R]_y^{-90}$

Rotations Needed for Generating Isometric Projection



$$P_v = [R]_x^{\phi}[R]_y^{\theta} P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P$$

Isometric Projection: **Equally foreshorten** **the three main axes**



$$\theta = \pm 45°,\ \phi = \pm 35.26°$$

Other Possible Rotation Paths

- *Rx --> Ry*

$$r_x = \pm 45°,\ r_y = \pm 35.26°$$

- *Rz --> Ry(Rx)*

$$r_z = \pm 45°, r_{y(x)} = \pm 54.74°$$

- *Rx(Ry) --> Rz*

$$r_{y(x)} = \pm 45°, r_z = ANY\ \ ANGLE$$