Coordinate Systems and Transformations

Graphic Libraries

Hardware dependent program

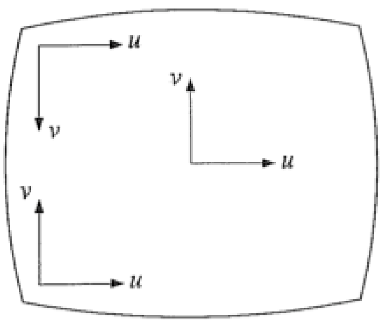
Hardware independent program

GL Examples:

Core, GKS, PEX, OpenGL

Coordinate Systems

- **Device Coordinate System:** identifies locations on the display

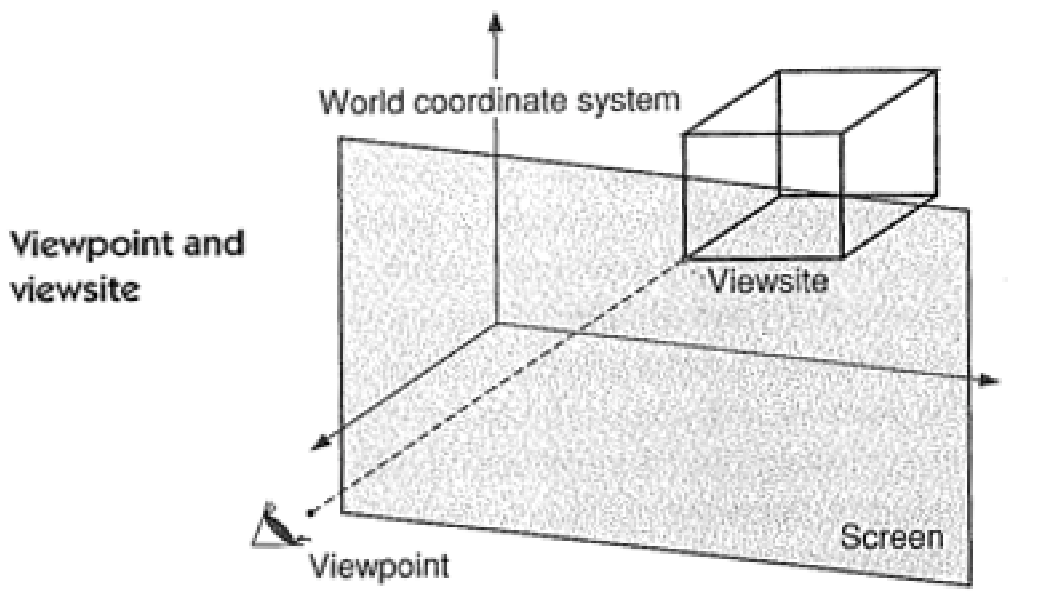


=>Virtual Device Coordinate System:
Usually the origin is at the low left corner and *u,v* range from 0 to 1.

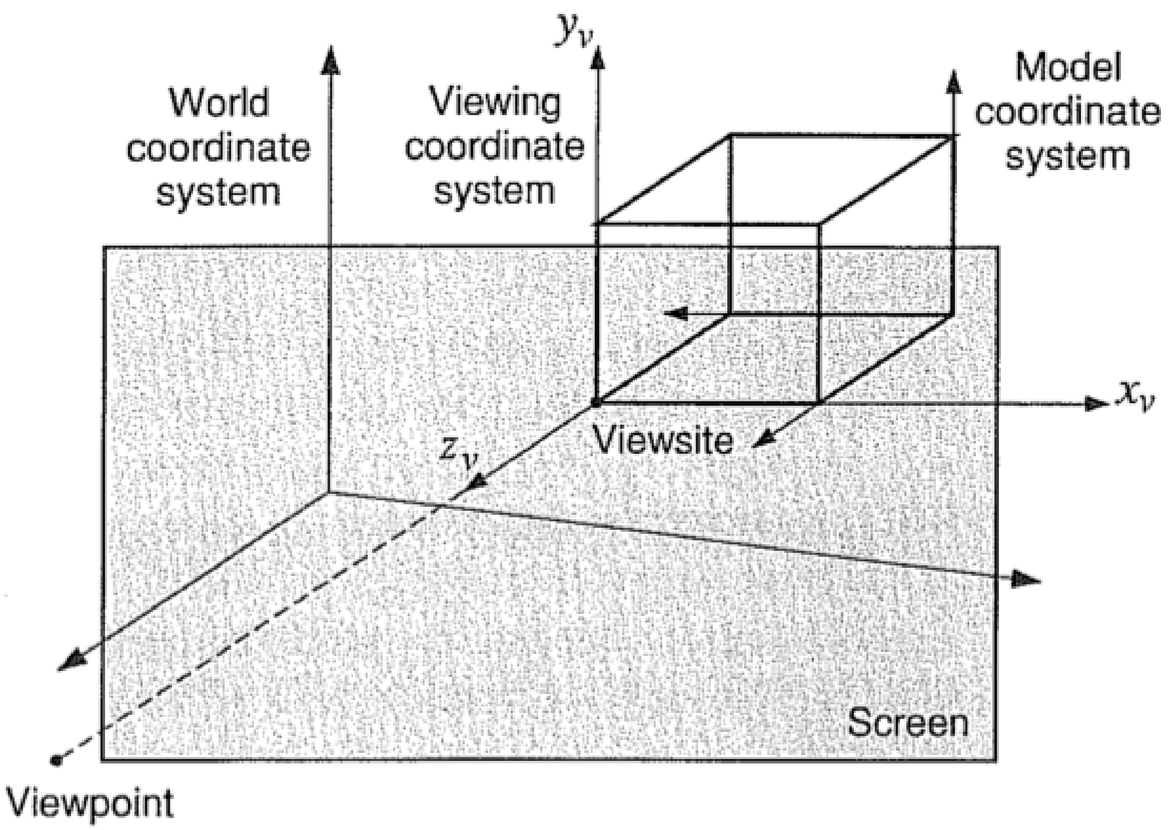
Coordinate Systems

- **Model Coordinate System(MCS):** identifies the shapes of object and it is attached to the object. Therefore the MCS moves with the object in the WCS
- **World Coordinate System (WCS):** identifies locations of objects in the world in the application.
- **Viewing Coordinate System (VCS):** Defined by the viewpoint and viewsite

Coordinate Systems
viewpoint and viewsite



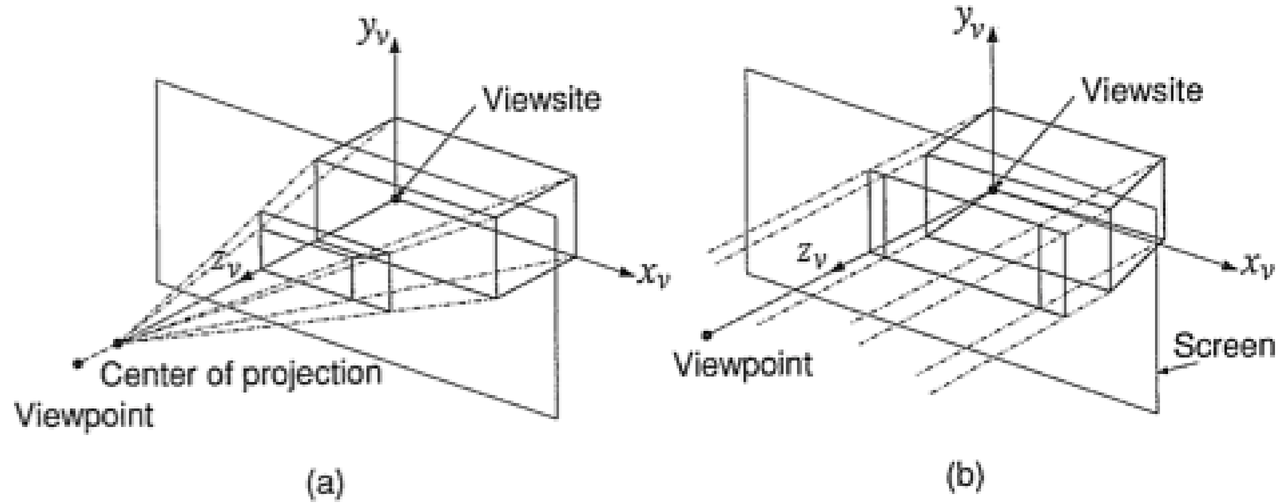
Coordinate Systems



Projection on Screen

- Parallel Projection
- Perspective Projection

Two types of projection: (a) perspective projection and (b) parallel projection



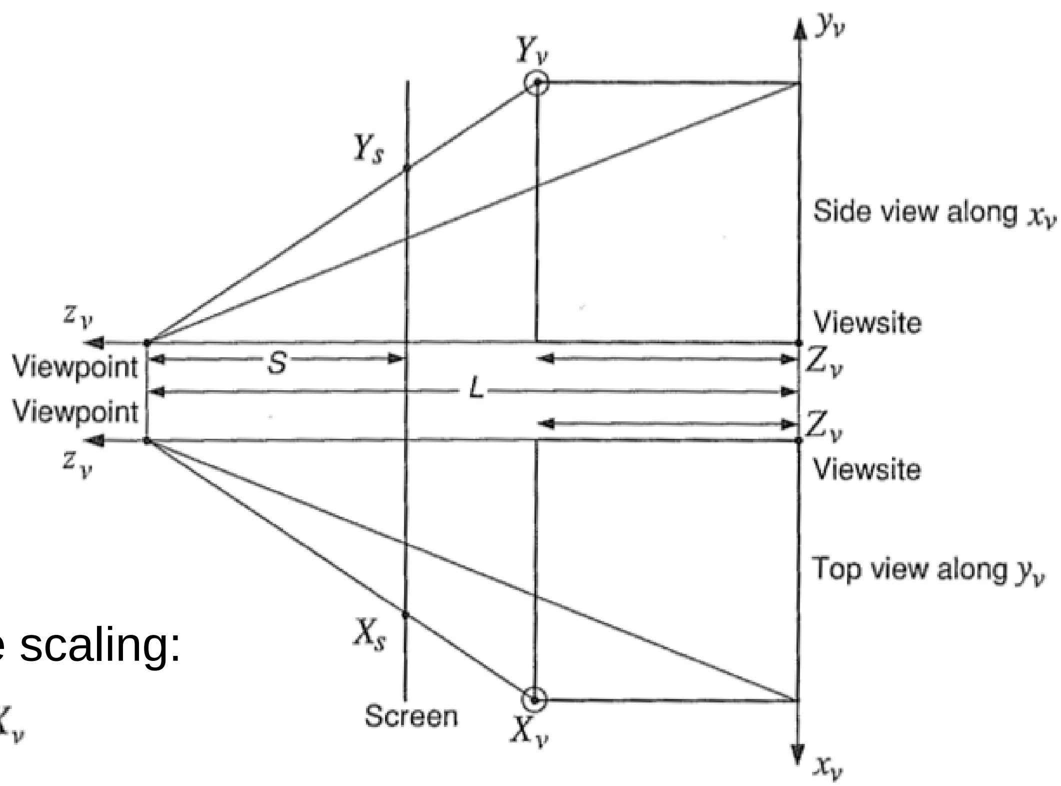
Parallel Projection

- **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**

Perspective Projection

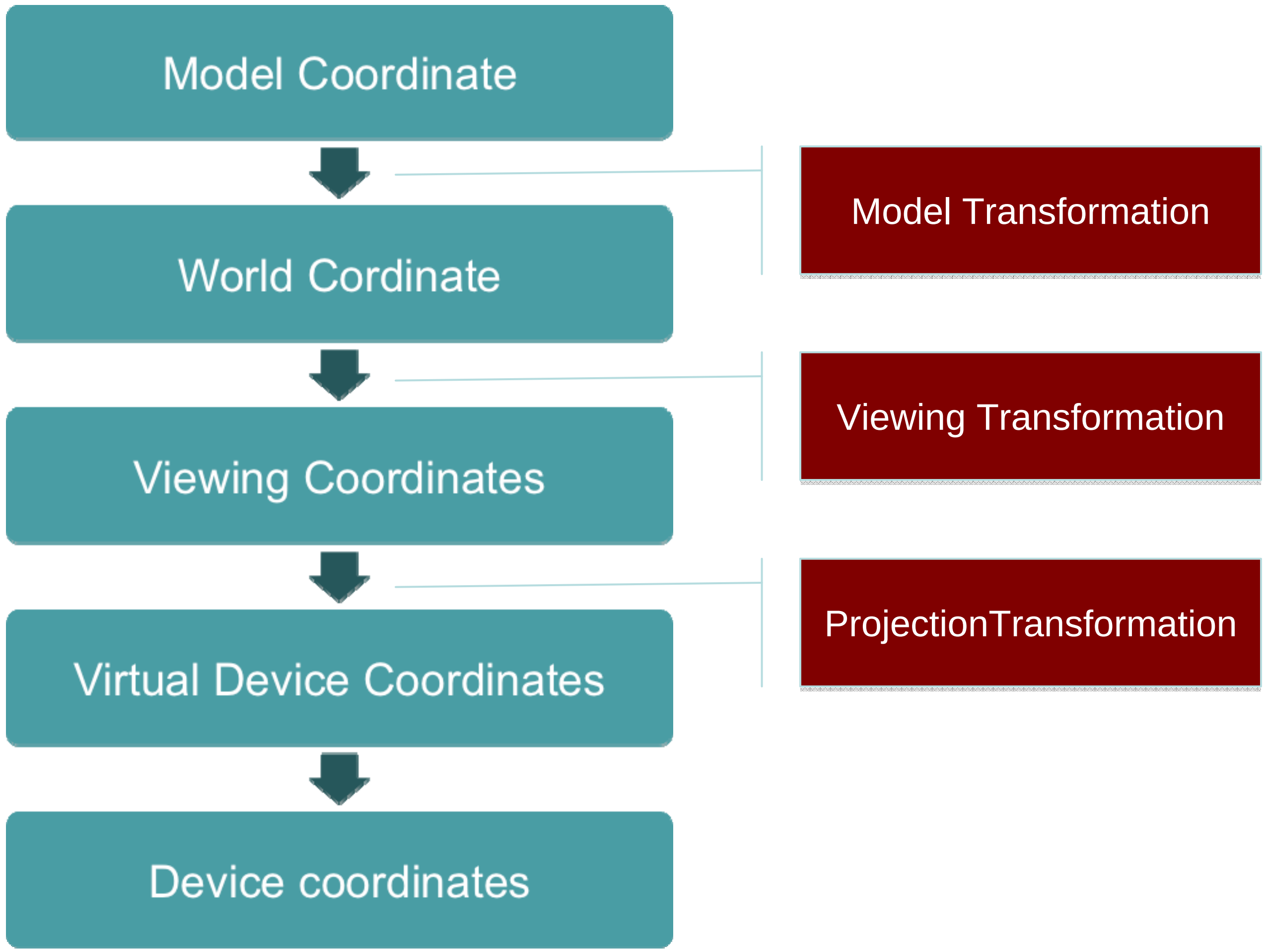


Coordinate scaling:

$$X_s = \frac{S}{L - Z_v} X_v$$

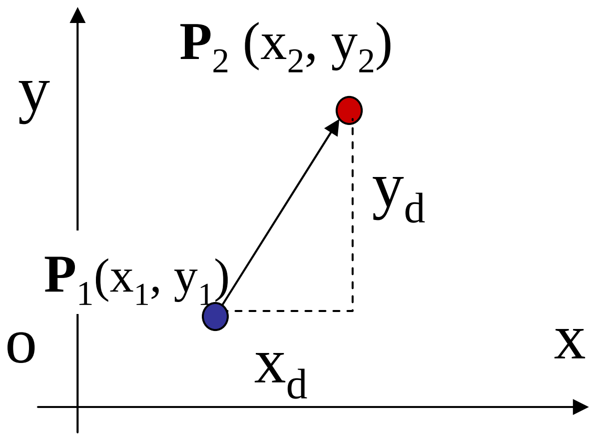
$$Y_s = \frac{S}{L - Z_v} Y_v$$

Transformations between Coordinate Systems

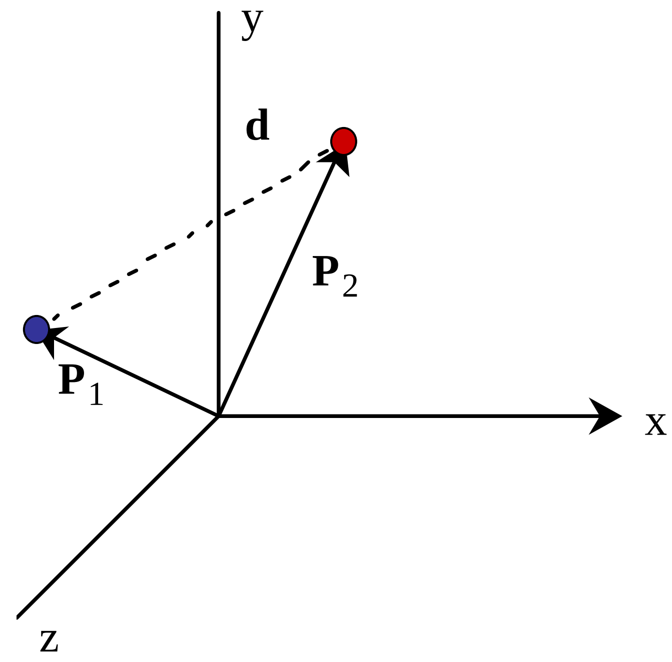


Translation

2D



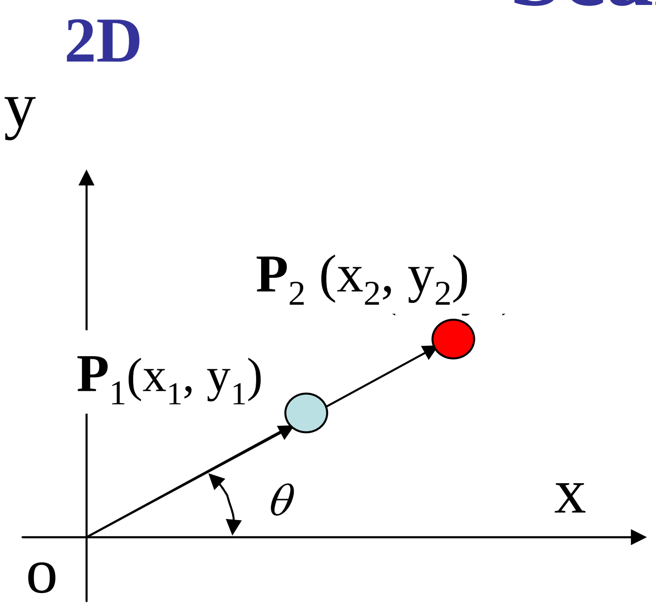
3D

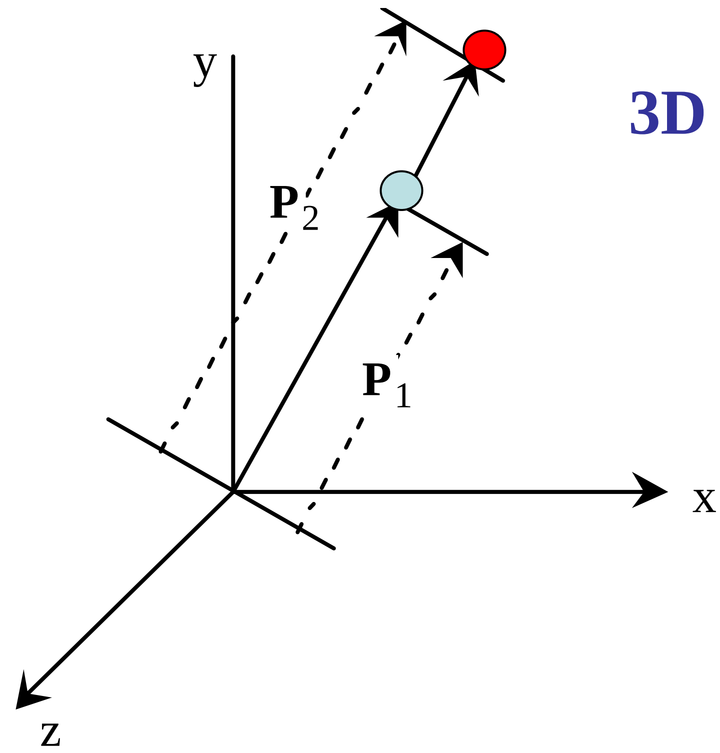


$$\mathbf{P}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

$$\mathbf{P}_2 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = \begin{bmatrix} x_1 + x_d \\ y_1 + y_d \\ z_1 + z_d \end{bmatrix} = \mathbf{P}_1 + \mathbf{d}$$

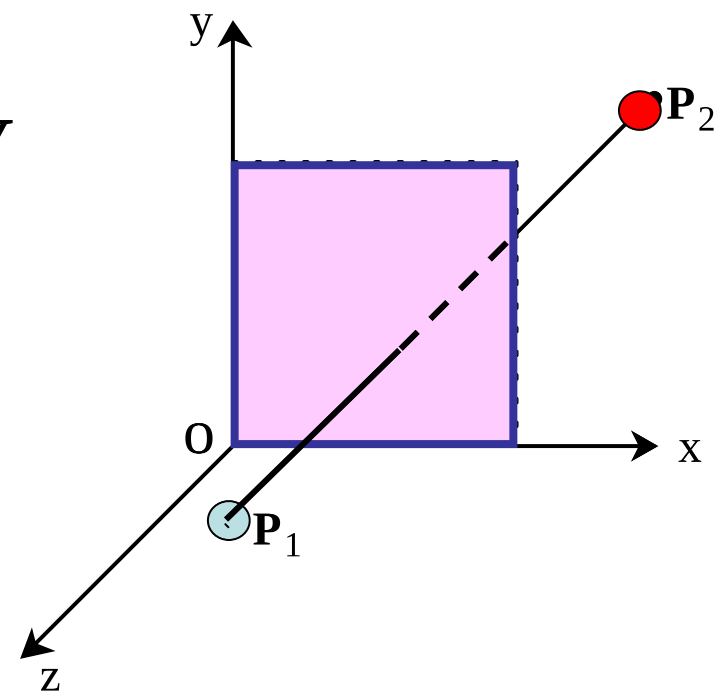
Scaling





$$\mathbf{P}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} sx_1 \\ sy_1 \\ sz_1 \end{bmatrix} \qquad \mathbf{P}_2 = s\mathbf{P}_1$$

Reflection (about XOY Plane)



$$\mathbf{P}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} x_1 \\ y_1 \\ -z_1 \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{P}_1$$

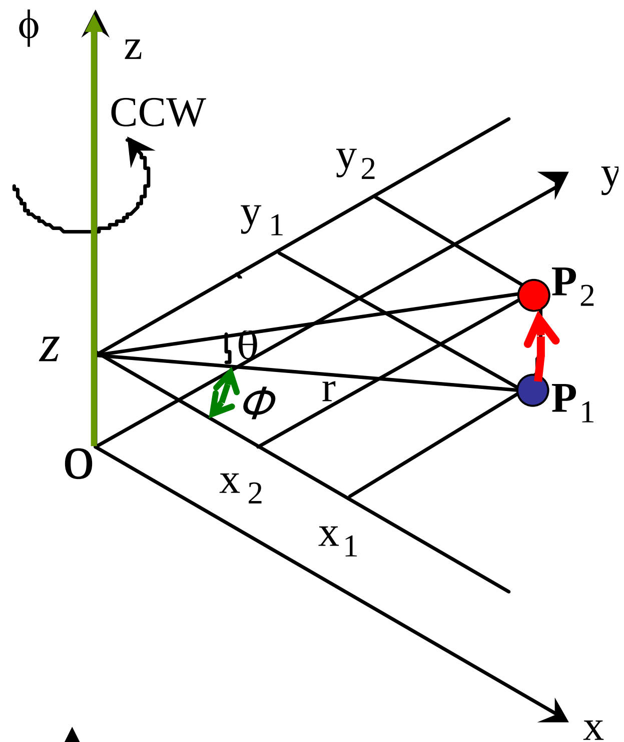
Rotation about Z Axis - CCW by θ

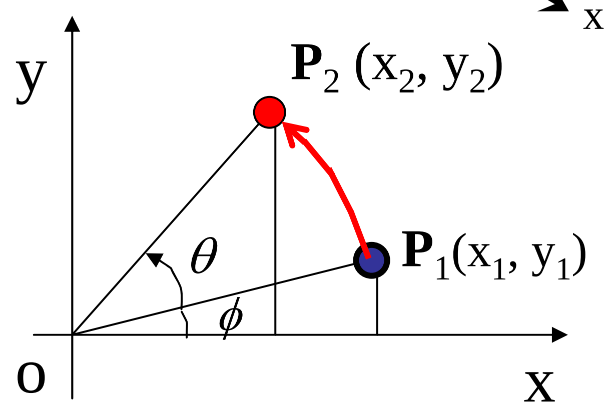
$$\mathbf{P}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} r\cos\varphi \\ r\sin\varphi \\ z \end{bmatrix}$$

$$\mathbf{P}_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} r\cos(\varphi+\theta) \\ r\sin(\varphi+\theta) \\ z \end{bmatrix}$$

$$= \begin{bmatrix} r\cos\varphi\cos\theta - r\sin\varphi\sin\theta \\ r\cos\varphi\sin\theta + r\sin\varphi\cos\theta \\ z \end{bmatrix}$$

$$= \begin{bmatrix} x_1\cos\theta - y_1\sin\theta \\ x_1\sin\theta + y_1\cos\theta \\ z_1 \end{bmatrix}$$



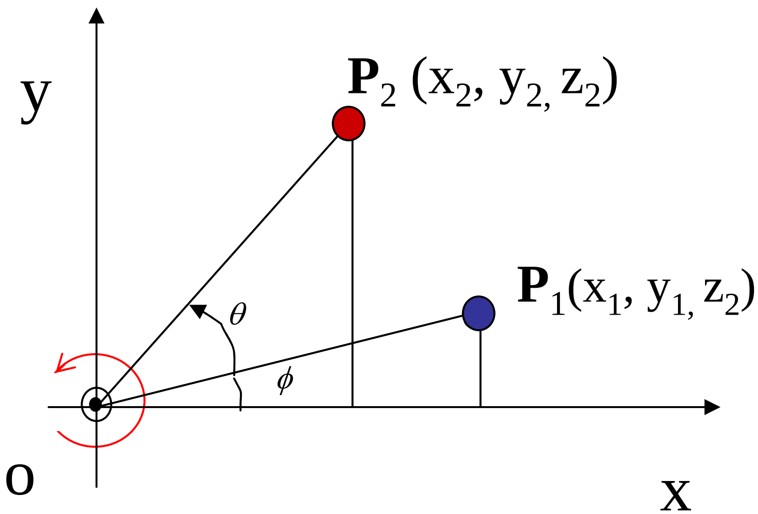


Rotation about Z Axis - CCW by θ

$$x_2 = x_1 \cos\theta - y_1 \sin\theta$$

$$y_2 = x_1 \sin\theta + y_1 \cos\theta$$

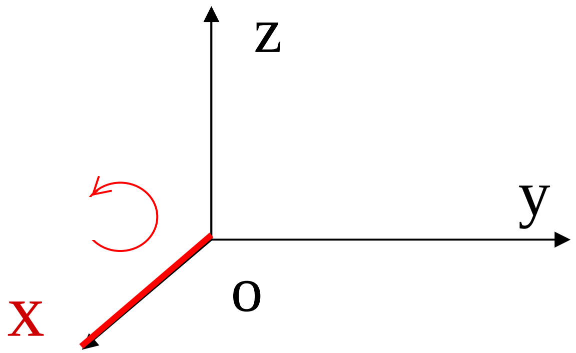
$$z_2 = z_1$$

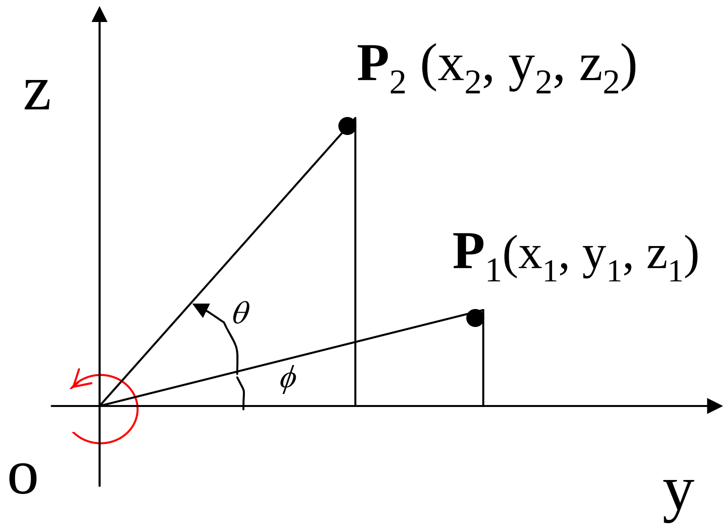


$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

Counter Clockwise, CCW

Rotation about X Axis - CCW by θ





$$x_2 = x$$

$$y_2 = y_1 \cos\theta - z_1 \sin\theta$$

$$z_2 = y_1 \sin\theta + z_1 \cos\theta$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

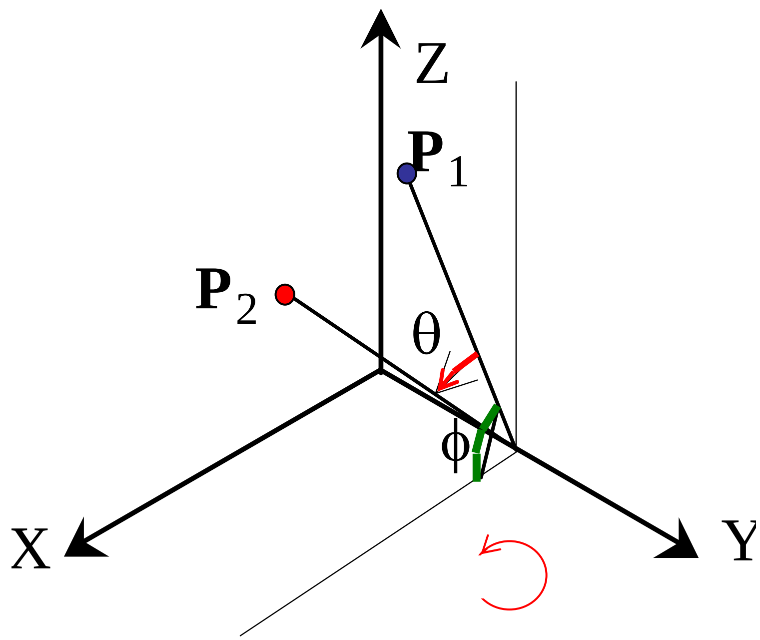
Rotation about Y Axis - CCW by θ

$$\mathbf{P}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} r\cos\varphi \\ y_1 \\ r\sin\varphi \end{bmatrix}$$

$$\mathbf{P}_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} r\cos(\varphi - \theta) \\ y_1 \\ r\sin(\varphi - \theta) \end{bmatrix} = \begin{bmatrix} r\cos\varphi\cos\theta + r\sin\varphi\sin\theta \\ y_1 \\ r\sin\varphi\cos\theta - r\cos\varphi\sin\theta \end{bmatrix}$$

$$= \begin{bmatrix} x_1\cos\theta + z_1\sin\theta \\ y_1 \\ -x_1\sin\theta + z_1\cos\theta \end{bmatrix}$$

$$\mathbf{P}_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = R_{[y]}\mathbf{P}_1$$



Homogeneous Representation

The representation is introduced to express all geometric transformations in the from of matrix multiplication for the convenience of manipulation.

Translation

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}$$

Dummy (n+1)th coordinate to facilitate multiplication

Homogeneous Representations

Scaling

$$[H] = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation

$$\mathbf{P}_2 = [R_z]\mathbf{P}_1 \quad and \quad [R_z] = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Representations

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad [R_y] = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Reflection

$$\mathbf{P}_2 = [M]\mathbf{P}_1 \quad and \quad [M] = \begin{bmatrix} \pm 1 & 0 & 0 & 0 \\ 0 & \pm 1 & 0 & 0 \\ 0 & 0 & \pm 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Composition of Transformation
(Concatenation of Transformation)

via Pre-Multiplication

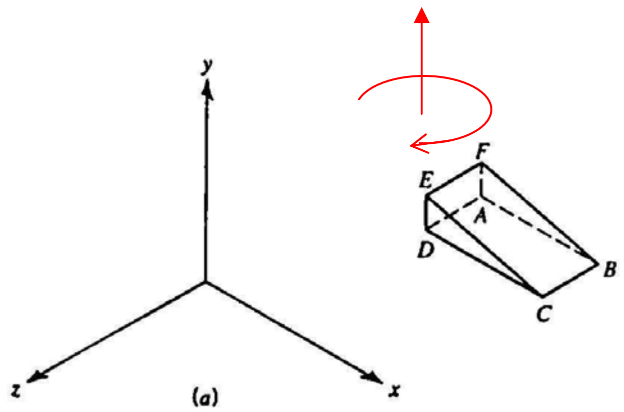
$$P_2 = [H_n][H_{n-1}]\cdots[H_1]P_1$$

An Example

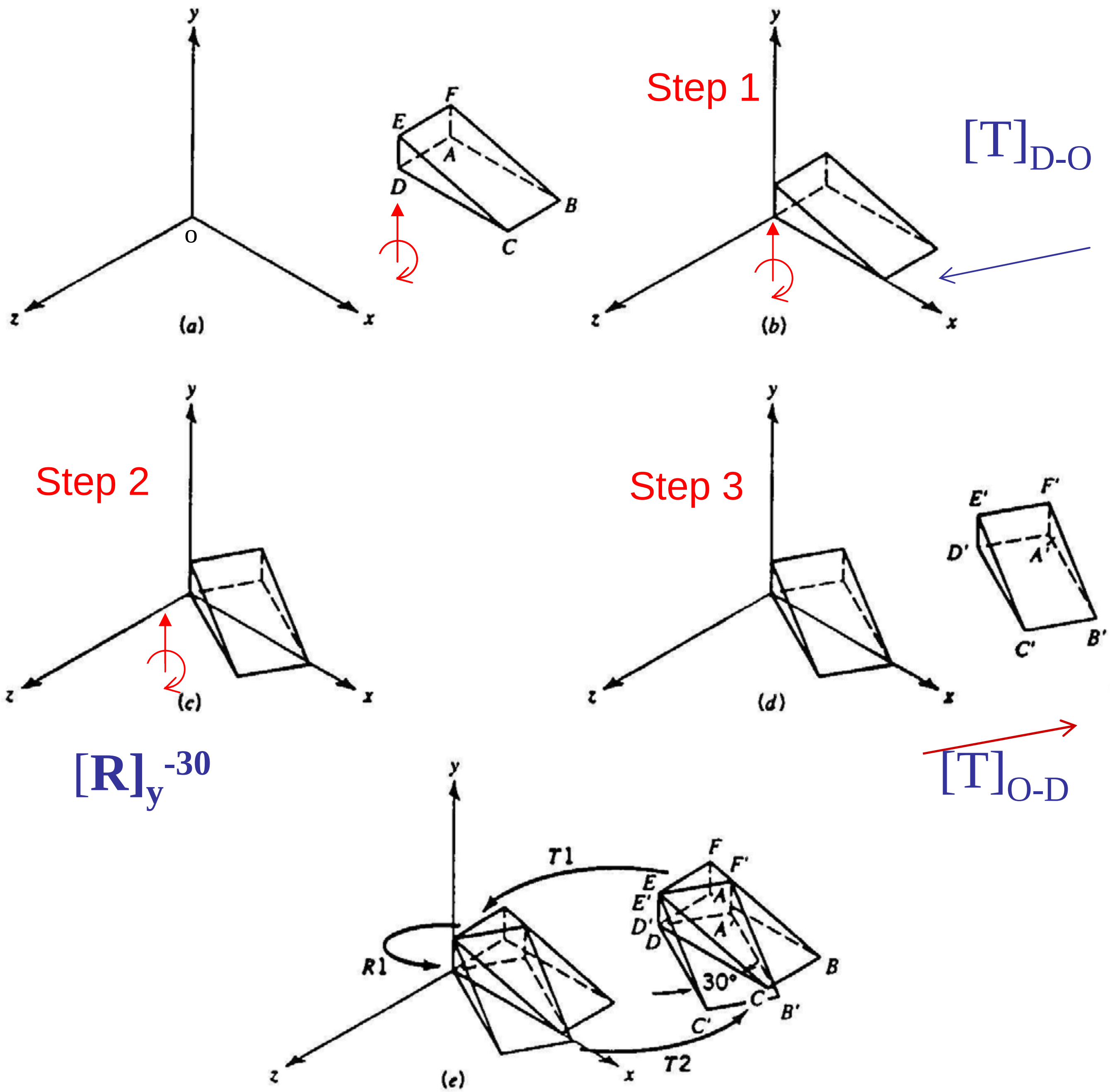
Consider a 3D object. The coordinates of the vertices are given as follows:

A=[3, 5, 3] B=[7, 5, 3] C=[7, 5, 5]

D=[3, 5, 5] E=[3, 6, 5] F=[3, 6, 3]



Rotate the 3D object by 30 degree in clockwise (CW) direction at point D about the Y-axis.



y
o
z
x
(a)
E
F
A
D
B
C
Step 1
[T]D-O
(b)
Step 2
(c)
Step 3
E'
F'
D'
A'
C'
B'
(d)
[R]y -30
[T]O-D
T1
R1
30°
T2
(e)

Procedure:

- First we translate ($[T]_{D\text{-}O}$) the object at the refernece point D to the origin O.
- Then we rotate ($[R]_Y^{-30}$) about the Y-axis
- Finally, we translate ($[T]_{O\text{-}D}$) the point D from the origin **back to** its original position.

$$[T]_{D-O} = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad [R]_Y^{-30} = \begin{bmatrix} \cos(-30) & 0 & \sin(-30) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-30) & 0 & \cos(-30) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[T]_{O-D} = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

[A B C D E F]

$$P_1 = \begin{bmatrix} 3 & 7 & 7 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 & 6 & 6 \\ 3 & 3 & 5 & 5 & 5 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

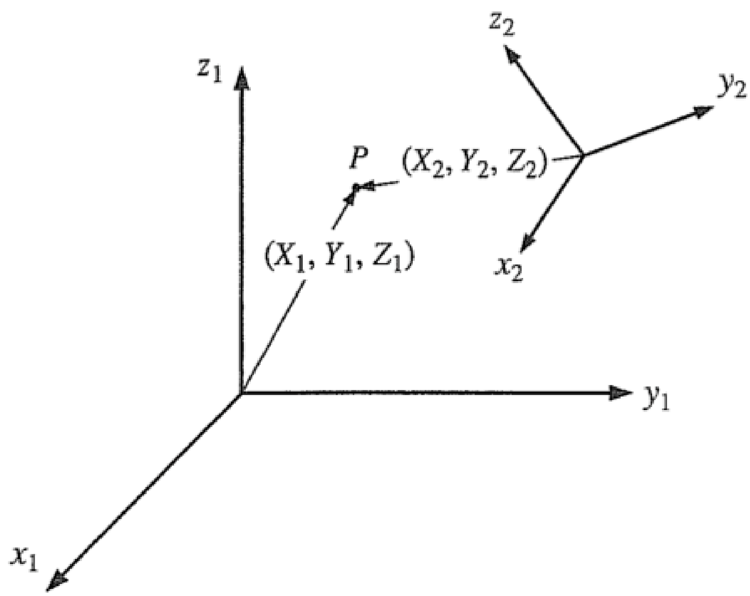
<u>The definition of the point matrix in the homogeneous representation.</u>

$$P_2 = [T]_{O-D}[R]_Z^{-30}[T]_{D-O}P_1$$

$$P_2 = \begin{bmatrix} 4.00 & 7.46 & 6.46 & 3.00 & 3.00 & 4.00 \\ 5.00 & 5.00 & 5.00 & 5.00 & 6.00 & 6.00 \\ 3.27 & 5.27 & 7.00 & 5.00 & 5.00 & 3.27 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \end{bmatrix}$$

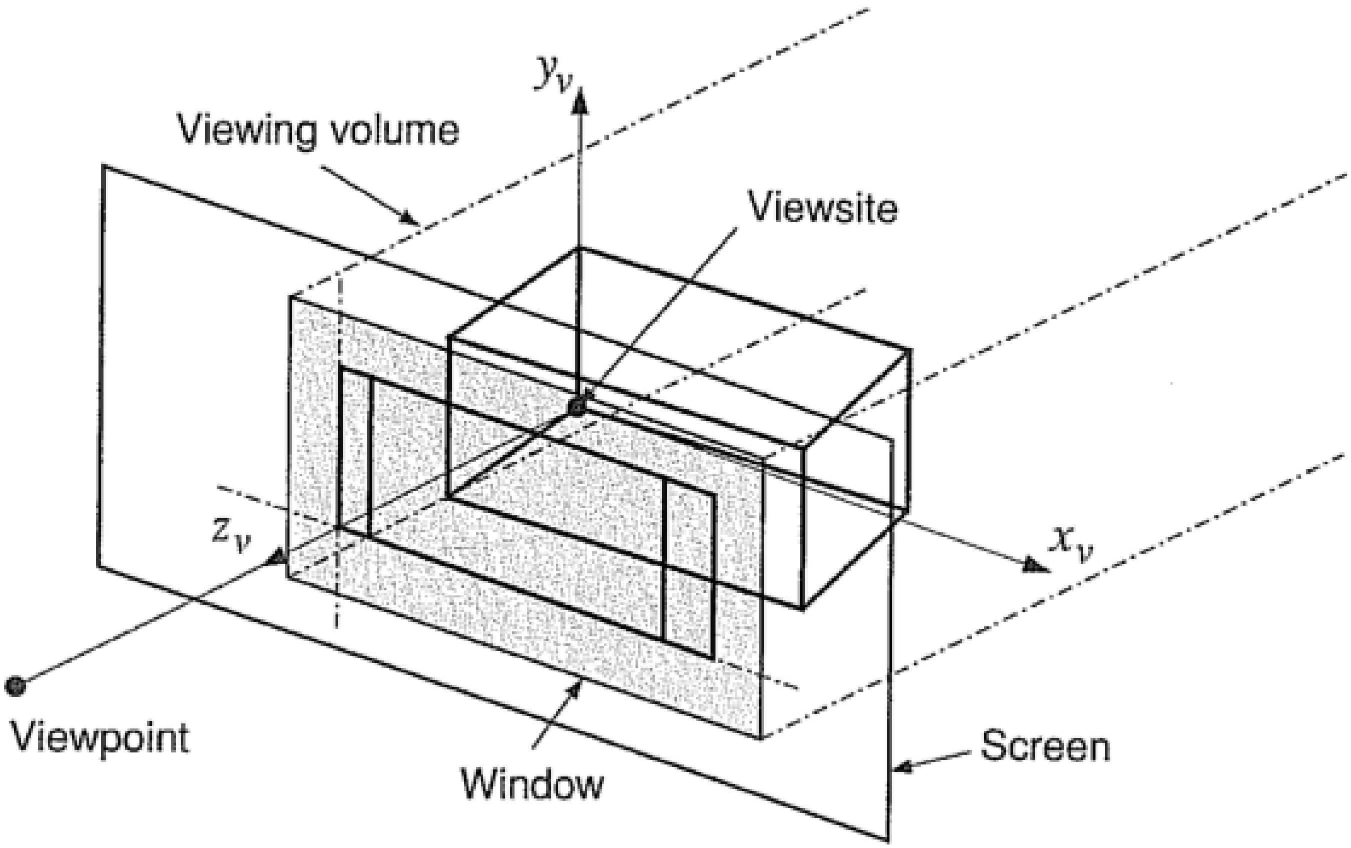
Mapping and Parallel Projections

- Mapping involves calculating the coordinates of a point known with rispect to a CS to a new CS.



About Transformations

- Transformations are applied for both changing WCS in respect to MCS, but also for projections



Mapping and Parallel Projections

(X_2, Y_2, Z_2) are assumed to be calculated by applying the transformation matrix T_{1-2} to (X_1, Y_1, Z_1) as follows:

$$[X_2 \ \ Y_2 \ \ Z_2 \ \ 1]^T = T_{1-2} \cdot [X_1 \ \ Y_1 \ \ Z_1 \ \ 1]^T \tag{3.11}$$

Replacing T_{1-2} with its elements allows Equation (3.11) to be expressed

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} \tag{3.12}$$

Where:

$P_x,p_y,p_z=x_2,y_2,z_2$ components required to translate (X_2,Y_2,Z_2) to (X_1,Y_1,Z_1), the following are evaluated as the translation as been performed:

$n_x,n_y,n_z=x_2,y_2,z_2$ components of a unit vector along x_1

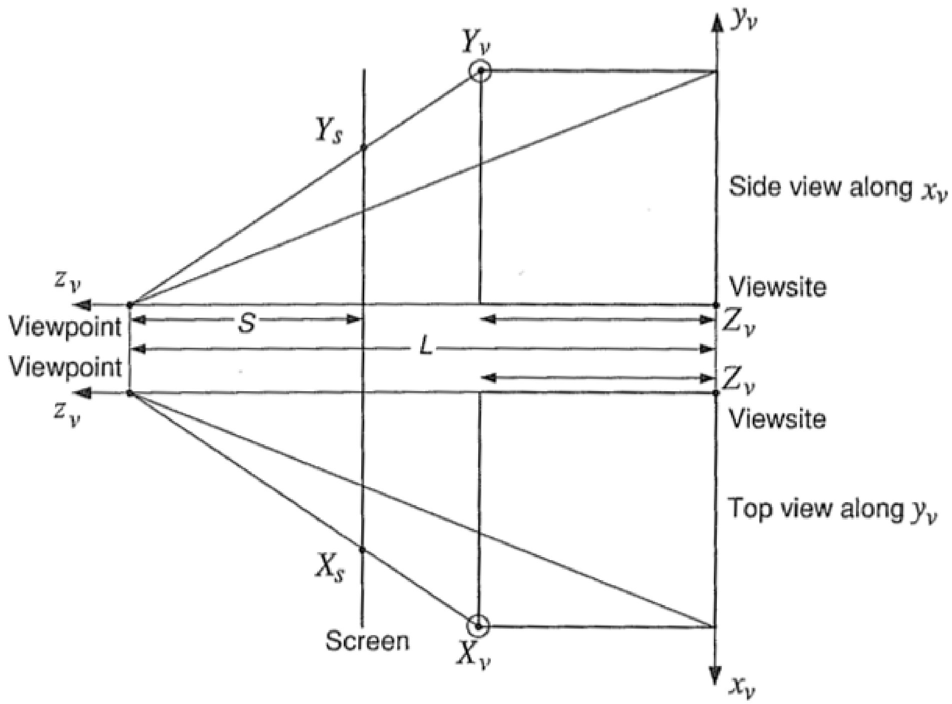
$ox,o_y,o_z=x_2,y_2,z_2$ components of a unit vector along y_1

$a_x,a_y,a_z=x_2,y_2,z_2$ components of a unit vector along y_1

Once mapping has been performed the Z coordinate is dropped

Mapping and Perspective Projections

Once mapping has been performed the z_v coordinate are not dropped yet and the x_v, y_v coordinates are scaled following the similarity rule of triangles (*L*,*S* needed). Then z_v is dropped.



Generating Parallel Projection

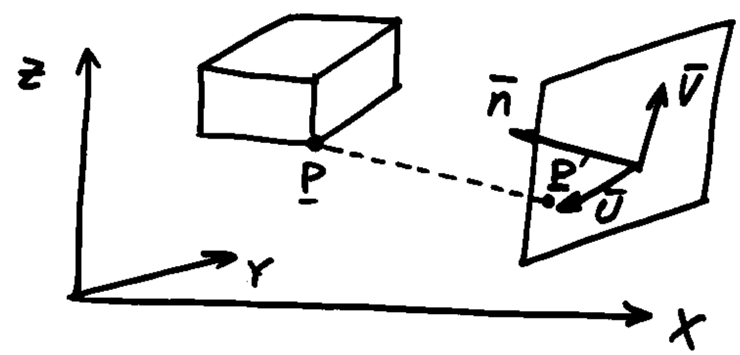
Drop the n (z_v) coordinate

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

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

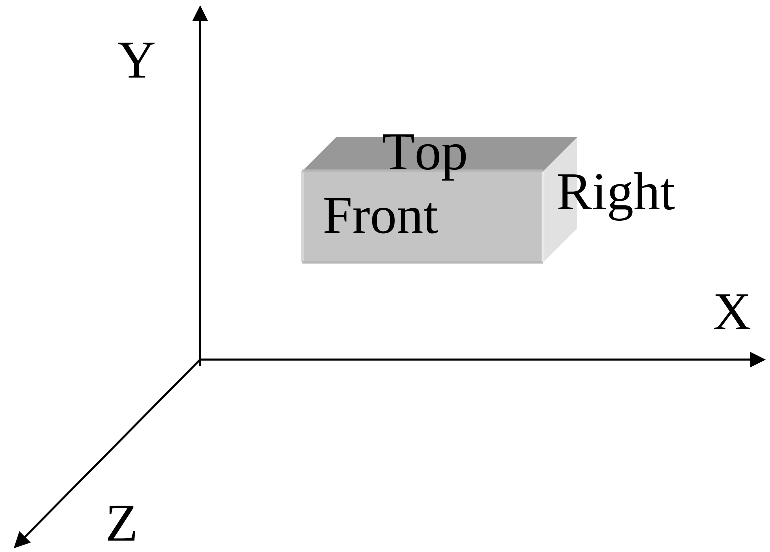
$$[T] = [D_n][R_z][R_y][R_x][D_{o_v,o}]$$

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



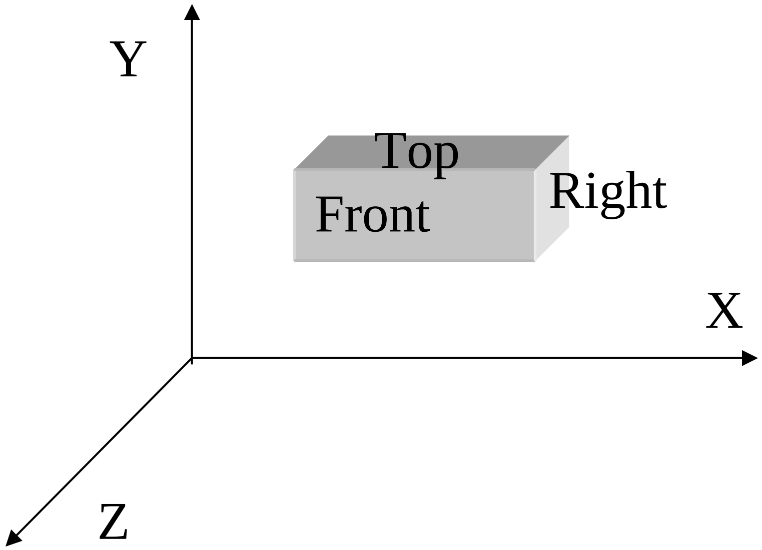
Note: The inverse transforms are not needed! We don't want to go back to x - y - z coordinates.

Orthographic Projection

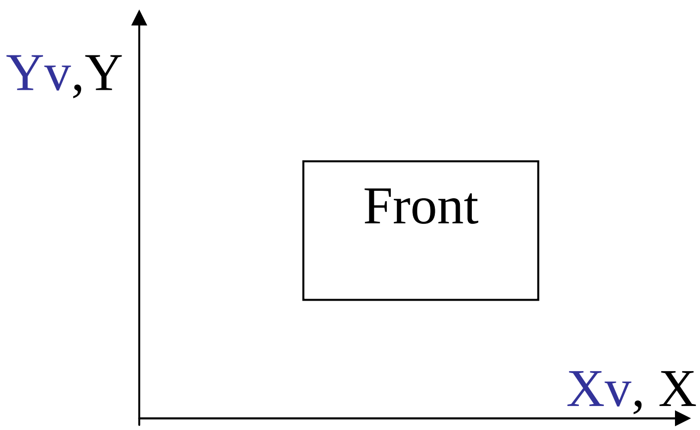


- Projection planes (Viewing planes) are perpendicular to the principal axes of the MCS of the model
- The projection direction (viewing direction) coincides with one of the MCS axes

Geometric Transformations for Generating Orthographic Projection (Front View)

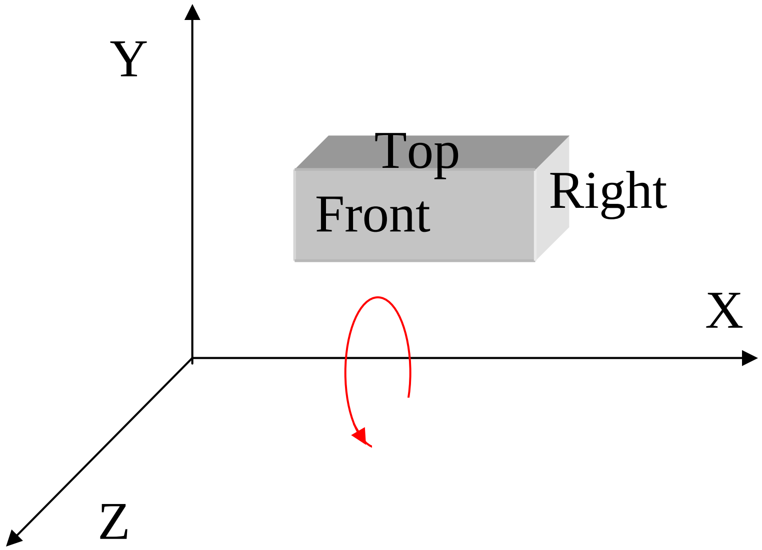


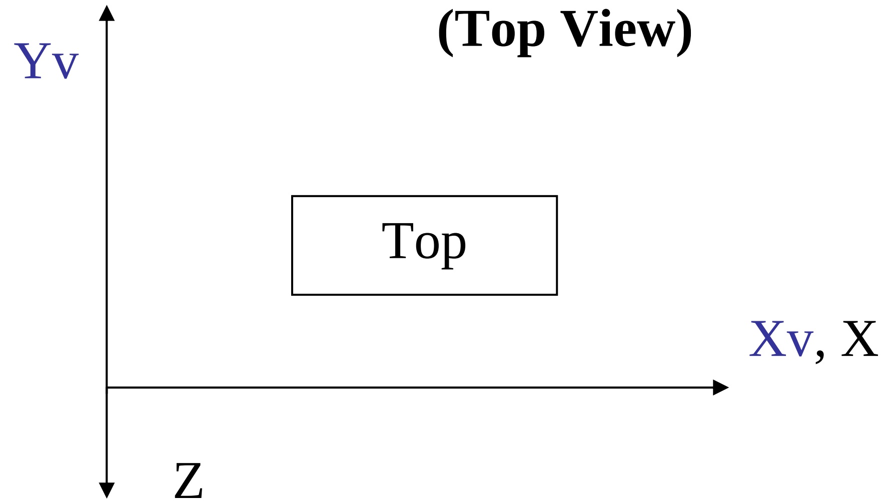
$$Pv = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P$$



Drop Z

Geometric Transformations for Generating Orthographic Projection (Top View)

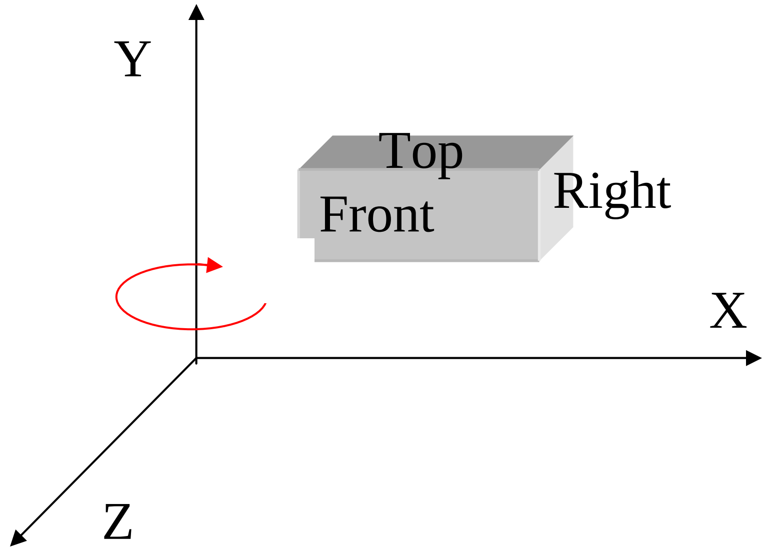


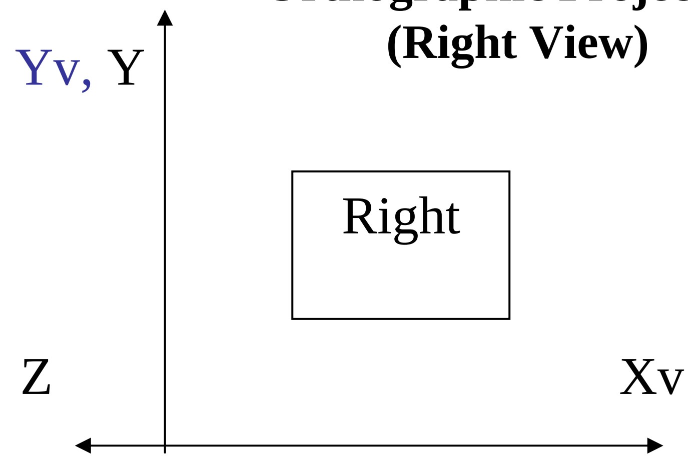


$$P_v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(90^\circ) & -\sin(90^\circ) & 0 \\ 0 & \sin(90^\circ) & \cos(90^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 &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

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

Geometric Transformations for Generating Orthographic Projection (Right View)

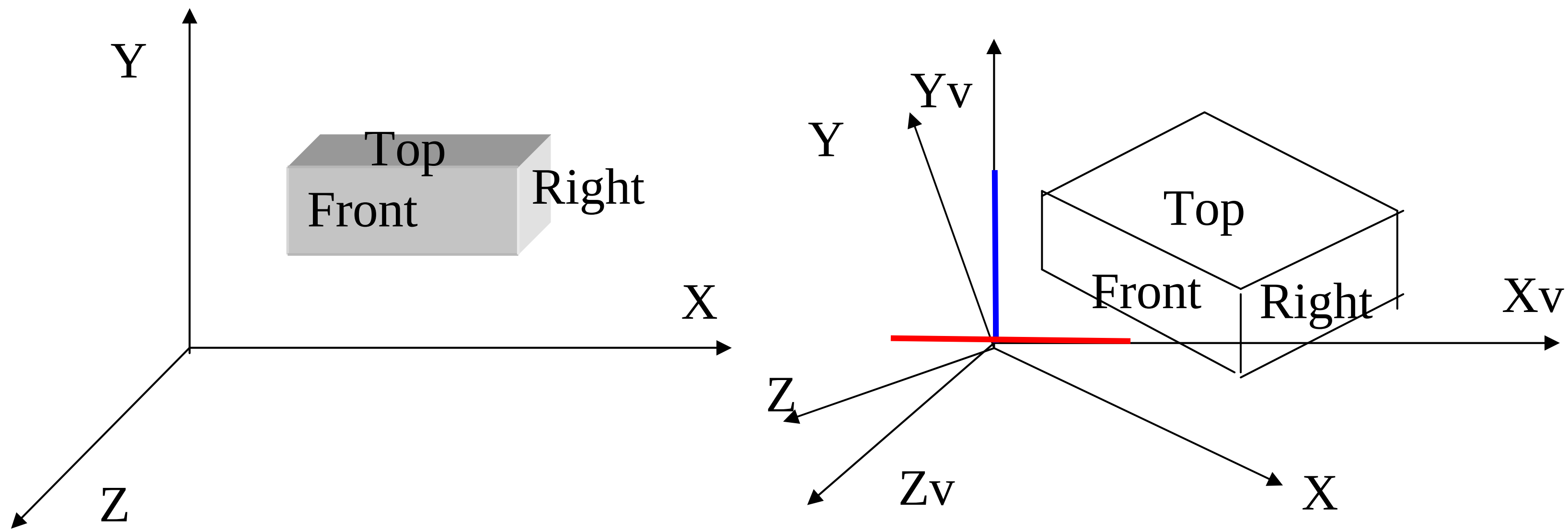




$$P_v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-90^\circ) & 0 & \sin(-90^\circ) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\left(-90^\circ\right) & 0 & \cos(-90^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P$$

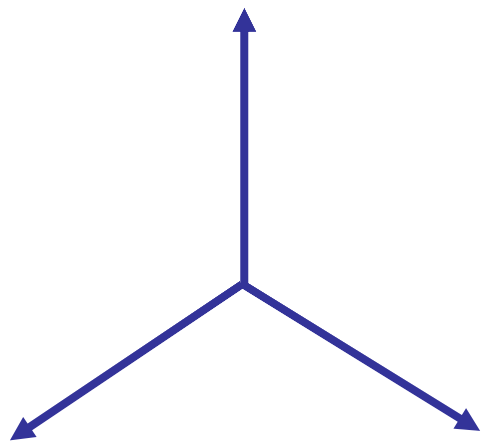
Drop Z $\qquad [R]_y^{-90}$

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*

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