3D TRANSFORMATION & VIEWING

Outline

- 3D Translation
- 3D Rotation
- 3D Scaling
- Other Transformation
- Viewing Pipeline
- Viewing Co-ordinates
- Projections
- View Volume and General Projection Transformation

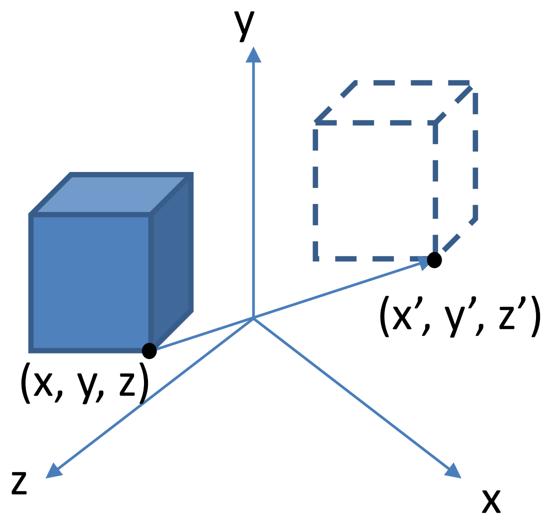
3D Translation

- Similar to 2D translation, which used 3×3 matrices, 3D translation use 4×4 matrices (x, y, z, h).
- In 3D translation point (x, y, z) is to be translated by amount t_x, t_y and t_z to location (x', y', z').

  $$x' = x + tx, \qquad y' = y + ty, \qquad z' = z + tz$$

- Matrix equation,

$$P' = T \cdot P = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Example- 3D Translation

- Translate the given point $P(10, 10, 10)$ into 3D space with translation factor $T(10, 20, 5)$.

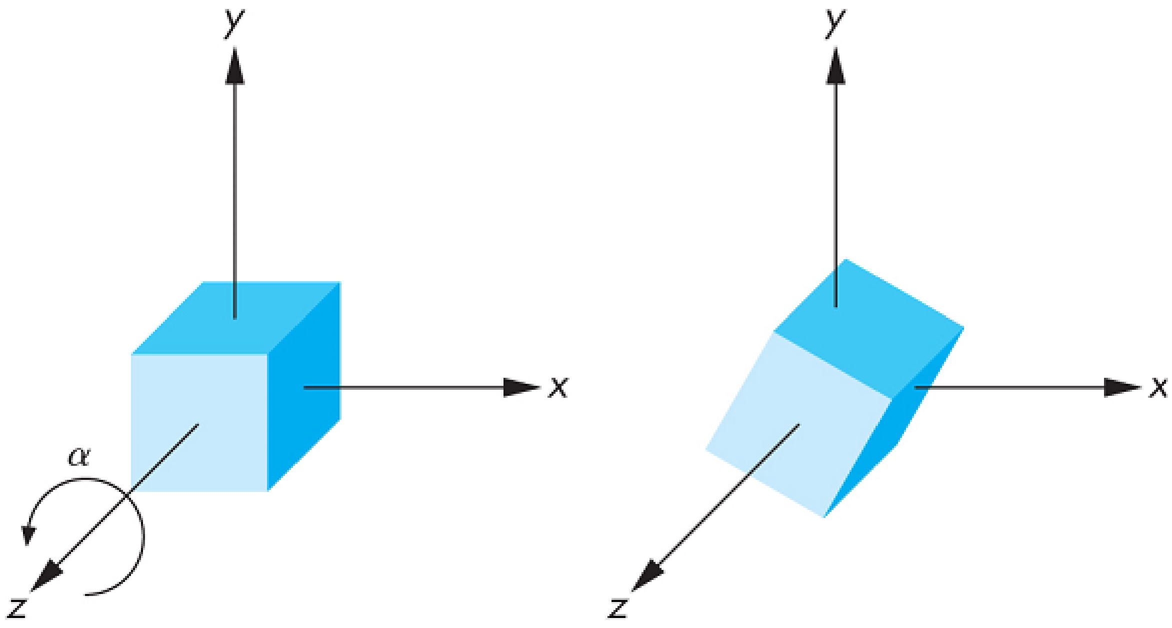
  $P' = T \cdot P$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 10 \\ 0 & 1 & 0 & 20 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 10 \\ 10 \\ 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 20 \\ 30 \\ 15 \\ 1 \end{bmatrix}$$

- Final coordinate after translation is $P'(20, 30, 15)$.

Rotation

- For 3D rotation we need to pick an axis to rotate about.
- The most common choices are the $X - axis$, the $Y - axis$, and the $Z - axis$, it is known as coordinate axis rotation.
- We can also chose other arbitrary axis for rotation.



Source: http://www.c-jump.com

Z-Axis Rotation

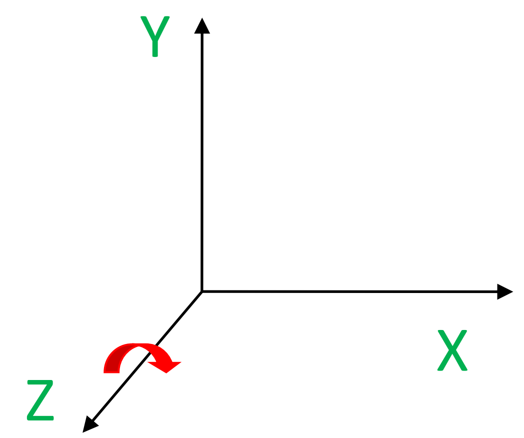
- Two dimension rotation equations can be easily convert into 3D $Z - axis$ rotation equations.
- Rotation about z axis we leave z coordinate unchanged.

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

  $y' = x\sin\theta + y\cos\theta$

  $z' = z$

  where Parameter θ specify rotation angle.



- Matrix equation is written as,

$$P' = R_z(\theta)\cdot P = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

X-Axis Rotation

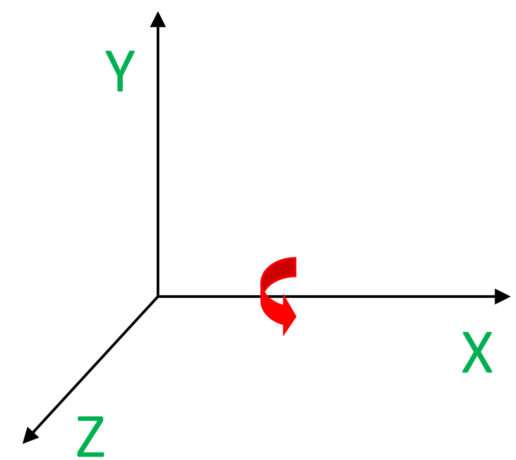
- Transformation equation for $\mathrm{X}-axis$ is obtain from equation of $\mathrm{Z}-axis$ rotation by replacing cyclically as $x \rightarrow y \rightarrow z \rightarrow x$
- Rotation about $X-axis$ we leave x coordinate unchanged.

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

  $z' = y\sin\theta + z\cos\theta$

  $x' = x$

  where Parameter θ specify rotation angle.



- Matrix equation is written as,

$$P' = R_x(\theta)\cdot P = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Y-Axis Rotation

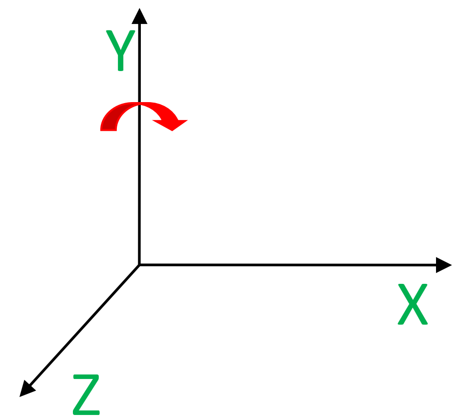
- Transformation equation for $\mathrm{Y}-axis$ is obtain from equation of $X-axis$ rotation by replacing cyclically as $x \rightarrow y \rightarrow z \rightarrow x$
- Rotation about $\mathrm{Y}-axis$ we leave y coordinate unchanged.

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

  $x' = z\sin\theta + x\cos\theta$

  $y' = y$

  where Parameter θ specify rotation angle.



- Matrix equation is written as,

$$P' = R_y(\theta)\cdot P = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Example- Coordinate Axis Rotation

- Rotate the point $P(5,5,5)$ 90^o about Z-axis.

$$P' = R_z(\theta = 90^o) \cdot P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos 90^o & -\sin 90^o & 0 & 0 \\ \sin 90^o & \cos 90^o & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 5 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} -5 \\ 5 \\ 5 \\ 1 \end{bmatrix}$$

- Final coordinate after rotation is $P'(-5,5,5)$.

General 3D Rotations

<u>When rotation axis is parallel to one of the standard axis.</u>

- Three steps require to complete such rotation these are,
    1. Translate the object so that the rotation axis coincides with the parallel coordinate axis.
    2. Perform the specified rotation about that axis.
    3. Translate the object so that the rotation axis is moved back to its original position.
- This can be represented in equation form as,

$$P' = T^{-1} \cdot R(\theta) \cdot T \cdot P$$

General 3D Rotations

When rotation axis is inclined in arbitrary direction.

- First we need rotations to align the axis with a selected coordinate axis and to bring the axis back to its original orientation.
- Five steps require to complete such rotation these are,
  1. Translate the object so that the rotation axis passes through the coordinate origin.
  2. Rotate the object so that the axis of rotation coincides with one of the coordinate axes.
  3. Perform the specified rotation about that coordinate axis.
  4. Apply inverse rotations to bring the rotation axis back to its original orientation.
  5. Apply the inverse translation to bring the rotation axis back to its original position.

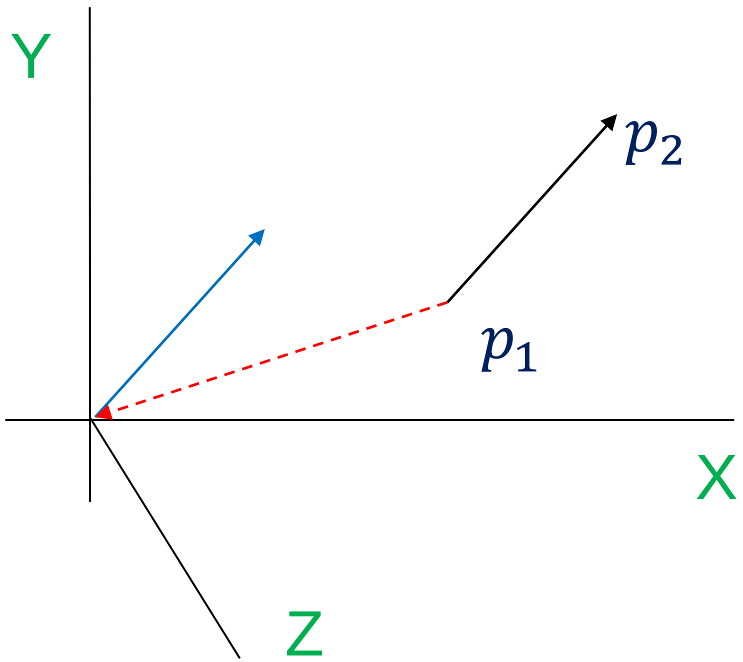
Contd.

- We can transform rotation axis onto any of the three coordinate axes. The $Z - axis$ is a reasonable choice.
- We are given line in the form of two end points $p_1\ (x_1, y_1, z_1)$, and $p_2\ (x_2, y_2, z_2)$.
- Let's discuss procedure step by step.

1. Translate the Object so that the Rotation Axis Passes Through the Coordinate Origin

- For translation of step one we will bring first end point at origin and transformation matrix for the same is as below

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



2. Rotate the Object so that the Axis of Rotation Coincides with One of the Coordinate Axes

- This task can be completed by two rotations first rotation about $X - axis$ and second rotation about $Y - axis$.
- But here we do not know rotation angle so we will use dot product and vector product.
- Vector notation for rotation axis is,

  $V = P_2 - P_1 = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$
- Unit vector along rotation axis is obtained by dividing vector by its magnitude.

$$u = \frac{V}{|V|} = \left(\frac{x_2 - x_1}{|V|}, \frac{y_2 - y_1}{|V|}, \frac{z_2 - z_1}{|V|}\right) = (a, b, c)$$

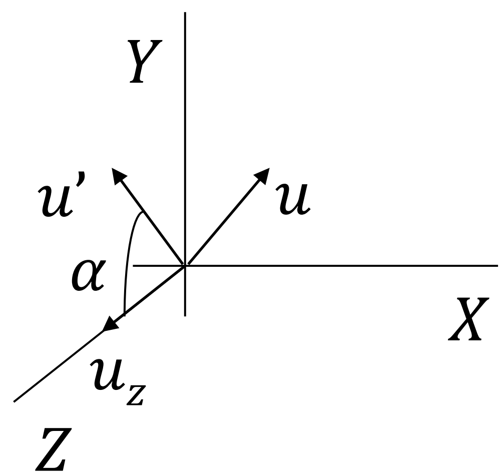
Contd.

- Now we need cosine and sine value of angle between unit vector u and XZ plane.
- For that take projection of u on $YZ - plane$ say u'.
- Find dot product and cross product of u' and u_z .
- Coordinate of u' is $(0, b, c)$ as we will take projection on $YZ - plane$ x value is zero.
- Dot product,

$$u' \cdot u_z = |u'||u_z| \, cos\, \alpha$$

$$cos\, \alpha = \frac{u' \cdot u_z}{|u'||u_z|} = \frac{(0, b, c)(0,0,1)}{\sqrt{b^2 + c^2}} = \frac{c}{d}$$

$$where \;\; d = \sqrt{b^2 + c^2}$$



Contd.

- Cross product,

$$u' \times u_z = u_x |u'||u_z| \sin\alpha \;\ldots(1)$$

$$u' \times u_z = u_x \cdot b \;\ldots(2)$$

u' (0, b, c)

u_z (0, 0, 1)

|u' X u_z |= (b*1-c*0, c*0-0*1, 0*0-0*b)

= (b, 0, 0)

| u' X u_z |=b

- From (1) and (2),

$$u_x |u'||u_z| \sin\alpha = u_x \cdot b$$

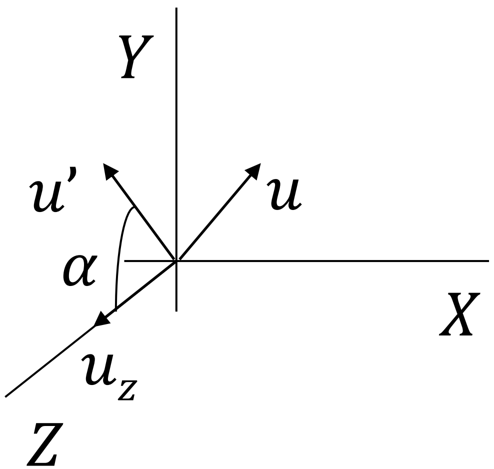
- Comparing magnitude

$$|u'||u_z| \sin\alpha = b$$

$$\sqrt{b^2 + c^2} \cdot (1) \sin\alpha = b$$

$$d \sin\alpha = b$$

$$\sin\alpha = \frac{b}{d}$$



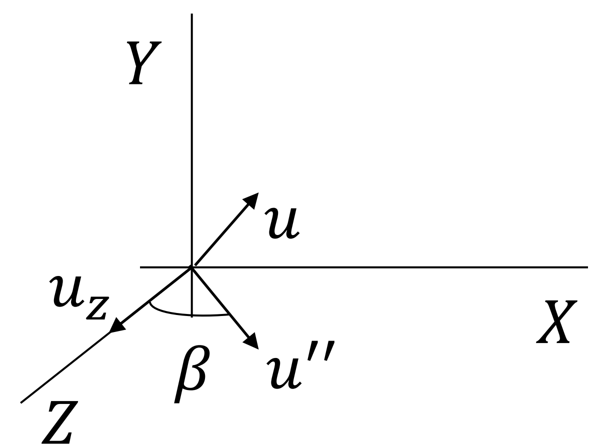
Contd.

- Now we have $sin\,\alpha$ and $cos\,\alpha$ so we will write matrix for rotation about X-axis.

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\,\alpha & -sin\,\alpha & 0 \\ 0 & sin\,\alpha & cos\,\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{b}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Contd.

- After performing above rotation u will rotated into u'' in $XZ-plane$ with coordinates $(a,\ 0,\ \sqrt{(b^2+c^2)})$.
- As we know rotation about x axis will leave x coordinate unchanged.
- u'' is in $XZ-plane$ so y coordinate is zero, and z component is same as magnitude of u'.
- Now rotate u'' about $Y-axis$ so that it coincides with $Z-axis$.



Contd.

- For that we repeat above procedure between u'' and u_z to find matrix for rotation about $Y-axis$.
- Dot product,

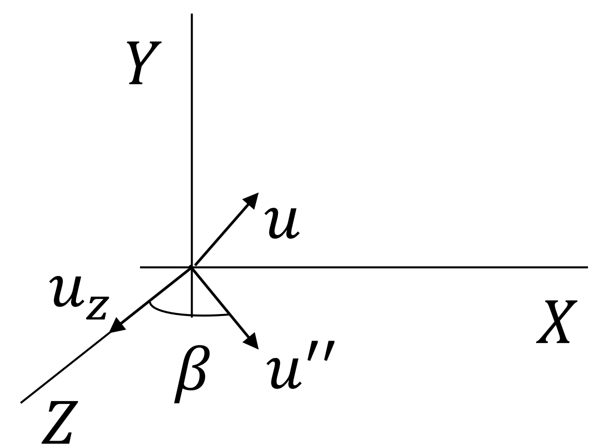
$$u'' \cdot u_z = |u''||u_z| \cos\beta$$

$$\cos\beta = \frac{u'' \cdot u_z}{|u''||u_z|}$$

$$\cos\beta = \frac{\left(a, 0, \sqrt{b^2+c^2}\right)(0,0,1)}{1}$$

$$\cos\beta = \sqrt{b^2+c^2} = d$$

$$where\ \ d = \sqrt{b^2+c^2}$$



Contd.

- Cross product,

  $u'' \times u_z = u_y |u''||u_z| \, sin \, \beta \; \ldots (1)$

  $u'' \times u_z = u_y \cdot (-a) \ldots (2)$

- From (1) and (2),

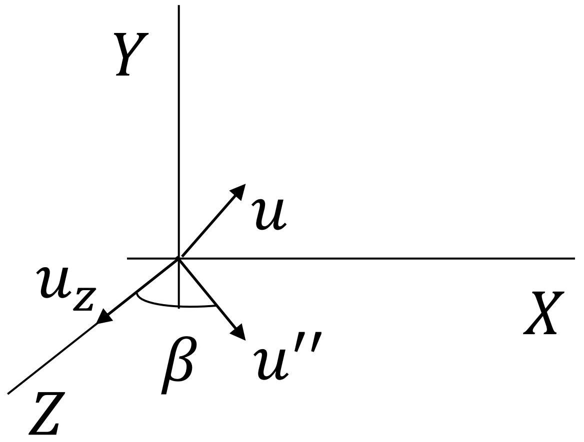
  $u_y |u''||u_z| \, sin \, \beta = u_y \cdot (-a)$

- Comparing magnitude

  $|u''||u_z| \, sin \, \beta = (-a)$

  $(1) \, sin \, \beta = - \, a$

  $sin \, \beta = - \, a$



Contd.

- Now we have $sin\,\beta$ and $cos\,\beta$ so we will write matrix for rotation about $Y - axis$.

$$R_y(\beta) = \begin{bmatrix} cos\,\beta & 0 & sin\,\beta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\,\beta & 0 & cos\,\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Now by combining both rotation we can coincides rotation axis with Z-axis

$$R_y(\beta) \cdot R_x(\alpha) = \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{b}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Perform the Specified Rotation About that Coordinate Axis

- As we align rotation axis with $Z - axis$ so now matrix for rotation about $Z - axis$,

$$R_z(\theta) = \begin{bmatrix} cos\,\theta & -\,sin\,\theta & 0 & 0 \\ sin\,\theta & cos\,\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$where\ \theta\ is\ specified\ rotation\ angle$

4. Apply Inverse Rotations to Bring the Rotation Axis Back to it's Original Orientation

- This step is inverse of step number 2,

  $R_x^{-1}(\alpha) \cdot R_y^{-1}(\beta)$

5. Apply the Inverse Translation to Bring the Rotation Axis Back to it's Original Position

- This step is inverse of step number 1,

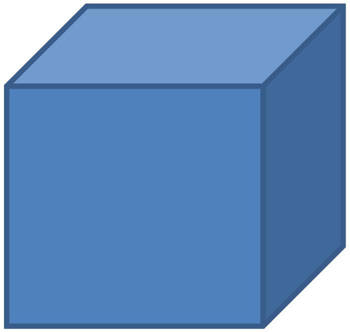
$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **So finally sequence of transformation for general 3D rotation is**

$$P' = T^{-1} \cdot R_x{}^{-1}(\alpha) \cdot R_y{}^{-1}(\beta) \cdot R_z(\theta) \cdot R_y(\beta) \cdot R_x(\alpha) \cdot T \cdot P$$

Scaling

- It is used to resize the object in 3D space.
- We can apply uniform as well as non uniform scaling by selecting proper scaling factor.
- Scaling in 3D is similar to scaling in 2D. Only one extra coordinate need to consider into it.

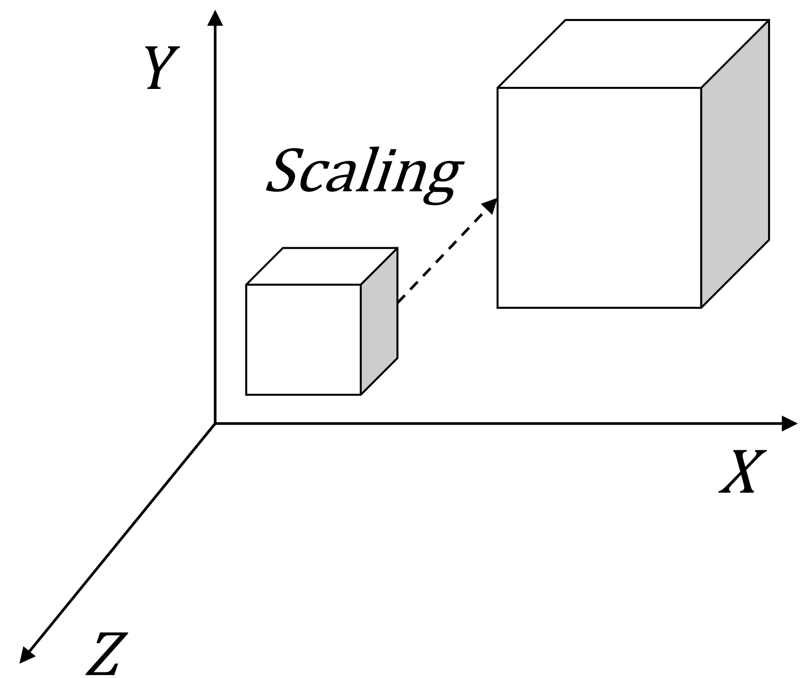


Coordinate Axes Scaling

- Simple coordinate axis scaling can be performed as below,

$$P' = S \cdot P$$

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



Example-Coordinate Axes Scaling

- Example: - Scale the line AB with coordinates $(10,20,10)$ and $(20,30,30)$ respectively with scale factor $S(3,2,4)$.

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

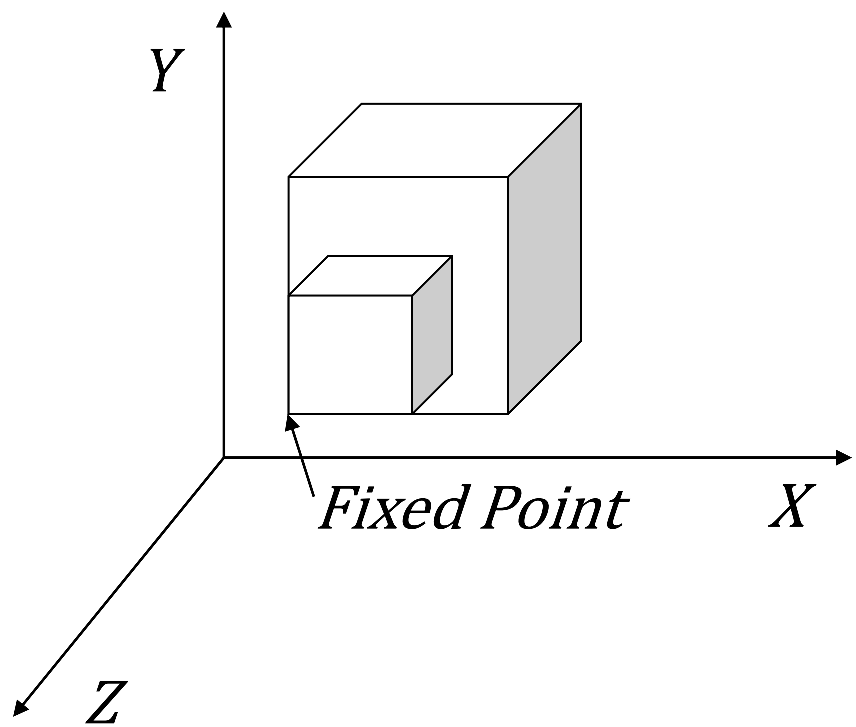
$$\begin{bmatrix} A_x' & B_x' \\ A_y' & B_y' \\ A_z' & B_z' \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 10 & 20 \\ 20 & 30 \\ 10 & 30 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 30 & 60 \\ 40 & 60 \\ 40 & 120 \\ 1 & 1 \end{bmatrix}$$

- Final coordinates after scaling are,

$$A'\ (30, 40, 40)\ and\ B'\ (60, 60, 120)$$

Fixed Point Scaling

- Fixed point scaling is used when we require scaling of object but particular point must be at its original position.
- Three steps require to complete such fixed point scaling these are,
  1. Translate the fixed point to the origin.
  2. Scale the object relative to the coordinate origin using coordinate axes scaling.
  3. Translate the fixed point back to its original position.



Contd.

- Matrix equation

$$P' = T(x_f, y_f, z_f) \cdot S(s_x, s_y, s_z) \cdot T(-x_f, -y_f, -z_f) \cdot P$$

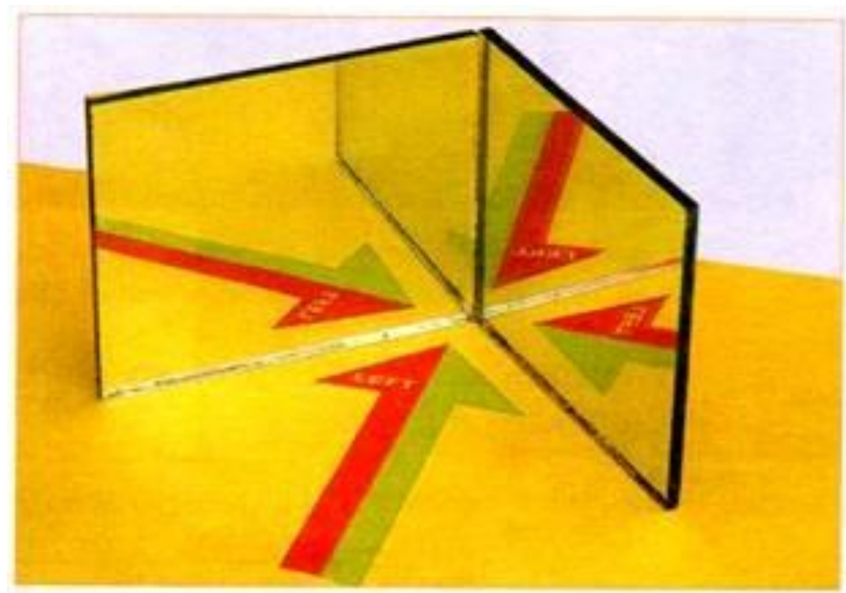
$$P' = \begin{bmatrix} 1 & 0 & 0 & x_f \\ 0 & 1 & 0 & y_f \\ 0 & 0 & 1 & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -x_f \\ 0 & 1 & 0 & -y_f \\ 0 & 0 & 1 & -z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot P$$

$$P' = \begin{bmatrix} s_x & 0 & 0 & (1 - s_x)x_f \\ 0 & s_y & 0 & (1 - s_y)y_f \\ 0 & 0 & s_z & (1 - s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot P$$

Other Transformations- Reflection

- Reflection means mirror image produced when mirror is placed at require position.
- When mirror is placed in XY-plane we obtain coordinates of image by just changing the sign of z coordinate.
- Transformation matrix for reflection about XY-plane is given below,

$$RF_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Source: http://www.yourarticlelibrary.com

Contd.

- Similarly Transformation matrix for reflection about YZ-plane is,

$$RF_x = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

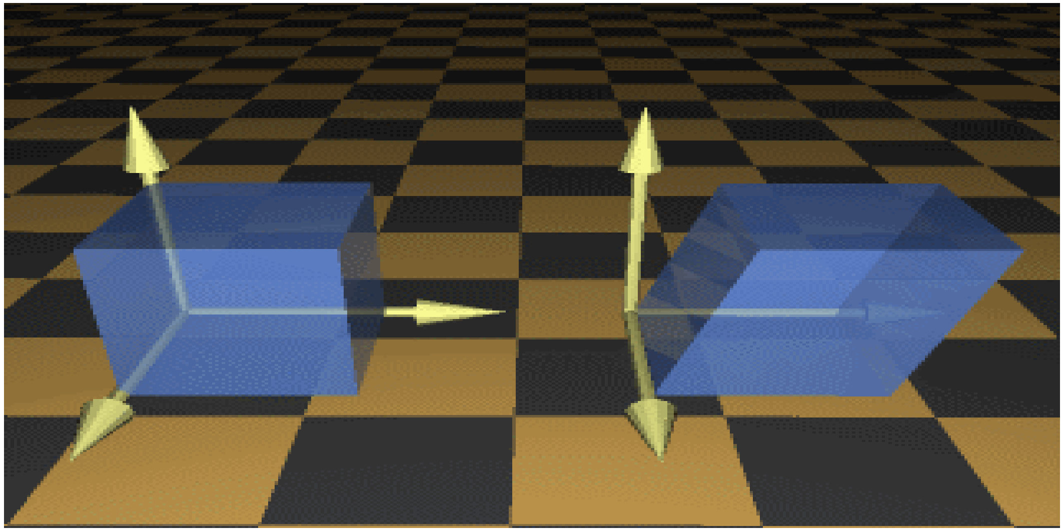
- Similarly Transformation matrix for reflection about XZ-plane is,

$$RF_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Other Transformations-Shear

- Shearing transformation can be used to modify object shapes.
- They are also useful in 3D viewing for obtaining general projection transformations.
- Here we use shear parameter ‘a’ and ‘b’
- Shear matrix for Z-axis is given below,

$$SH_z = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Source: http://140.129.20.249/~jmchen/cg

Other Transformations-Shear

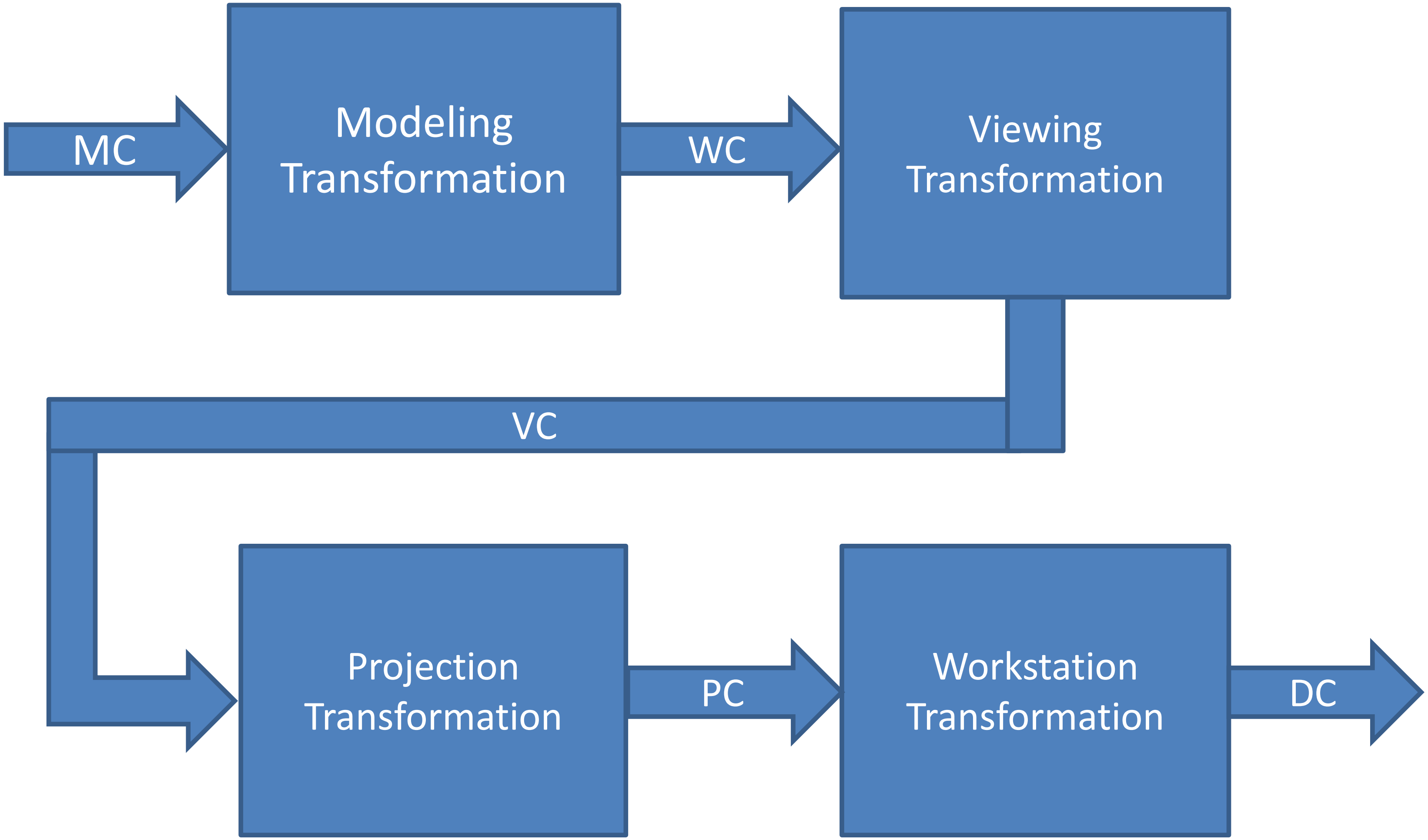
- Similarly Shear matrix for X-axis is,

$$SH_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ b & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Similarly Shear matrix for Y-axis is,

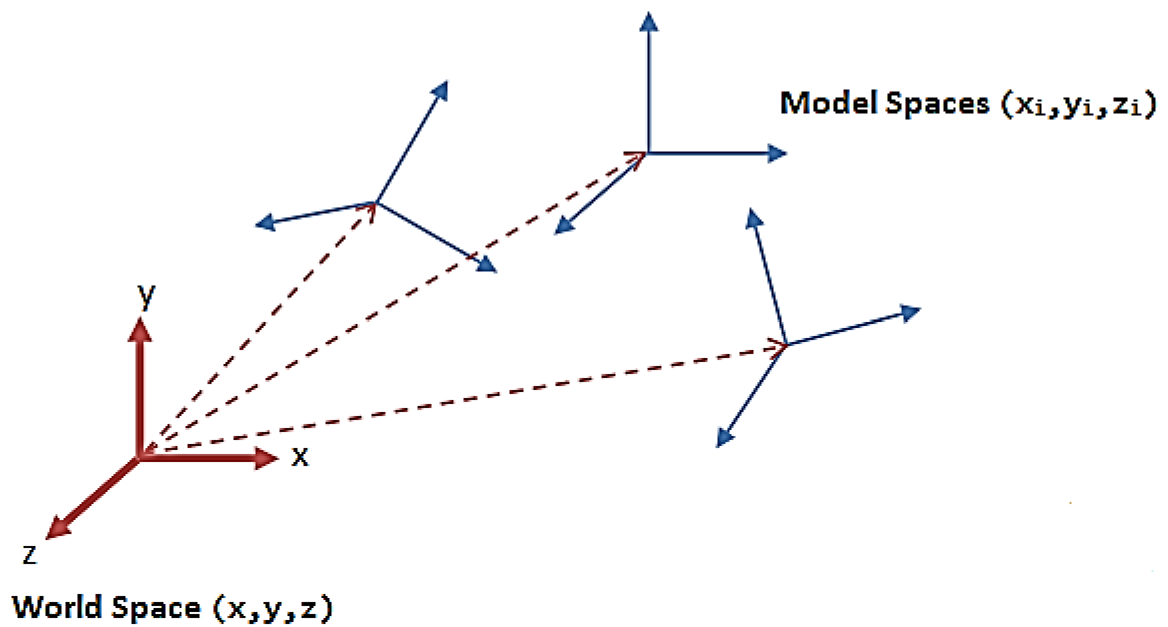
$$SH_y = \begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Viewing Pipeline



Viewing Co-ordinates

- Generating a view of an object is similar to photographing the object.
- We can take photograph from any side with any angle & orientation of camera.
- Similarly we can specify viewing coordinate in ordinary direction.



Specifying the View Plan

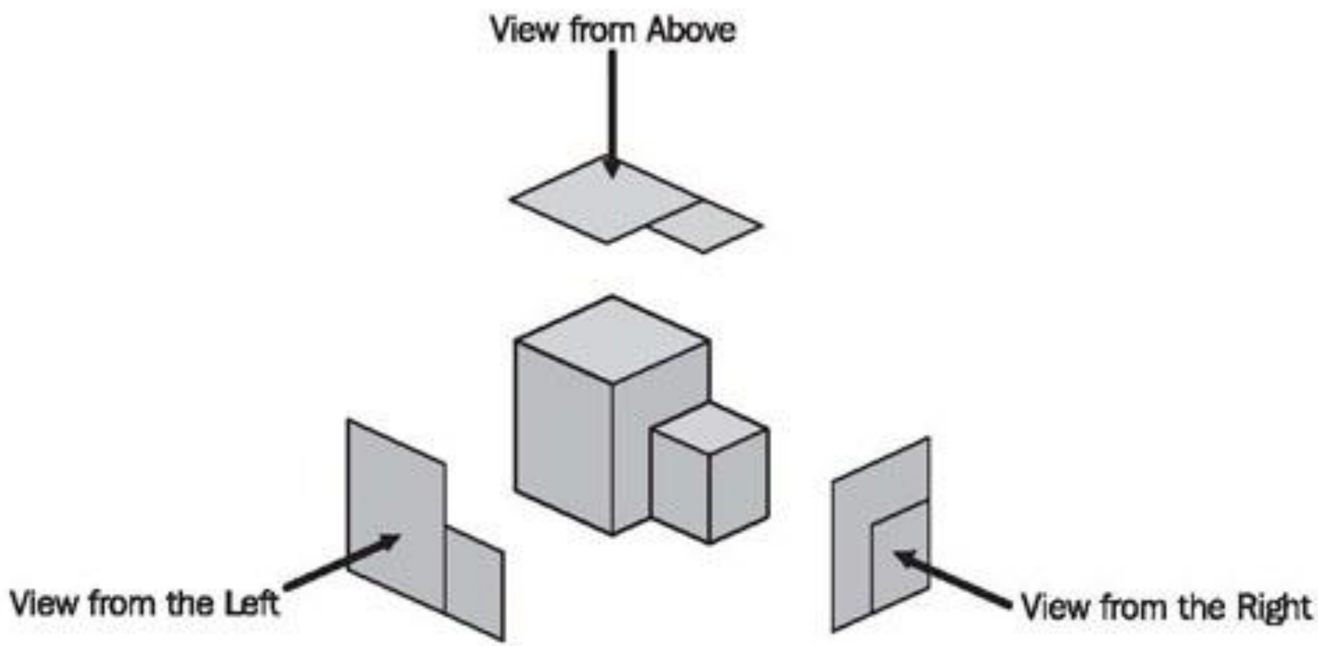
- We decide view for a scene by first establishing viewing coordinate system, also referred as view reference coordinate system.
- Projection plane is setup in perpendicular direction to Z_v axis.
- Projections positions in the scene are transferred to viewing coordinate.
- Then viewing coordinate are projected onto the view plane.
- The origin of our viewing coordinate system is called view reference point.
- View reference point is often chosen to be close to or on the surface as same object scene.
- We can choose other point also.

Contd.

- Next we select positive direction for the viewing Z_v axis and the orientation of the view plane by specifying the view plane normal vector N.
- Finally we choose the up direction for the view by specifying a vector V called the view up vector. Which specify orientation of camera.
- View up vector is generally selected perpendicular to normal vector but we can select any angle between V & N.

Contd.

- By fixing view reference point and changing direction of normal vector N we get different views of same object.



Source: https://sourcemaking.com

World to Viewing Coordinates Transformation

- Before taking projection of view plane object description is need to transfer from world to viewing coordinate.
- It is same as transformation that superimposes viewing coordinate system to world coordinate system.
- It requires following basic transformation.
  1. Translate view reference point to the origin of the world coordinate system.
  2. Apply rotation to align.

Contd.

- Consider view reference point in world coordinate system is at position (x_0, y_0, z_0).
- For align view reference point to world origin we perform translation with matrix,

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

- Now we require rotation sequence up-to three coordinate axis rotations depending upon direction we choose for N.
- In general case N is at arbitrary direction then we can align it with word coordinate axes by rotation sequence $Rz \cdot Ry \cdot Rx$.

Contd.

- Another method for generating the rotation transformation matrix is to calculate $u, v \,\&\, n$ unit vectors and from the composite rotation matrix directly,

$$n = \frac{N}{|N|} = (n_1, n_2, n_3)$$

$$u = \frac{V \times N}{|V \times N|} = (u_1, u_2, u_3)$$

$$v = n \times u = (v_1, v_2, v_3)$$

- This method also automatically adjusts the direction for u so that v is perpendicular to n.

Contd.

- Than composite rotation matrix for the viewing transformation is,

$$R = \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

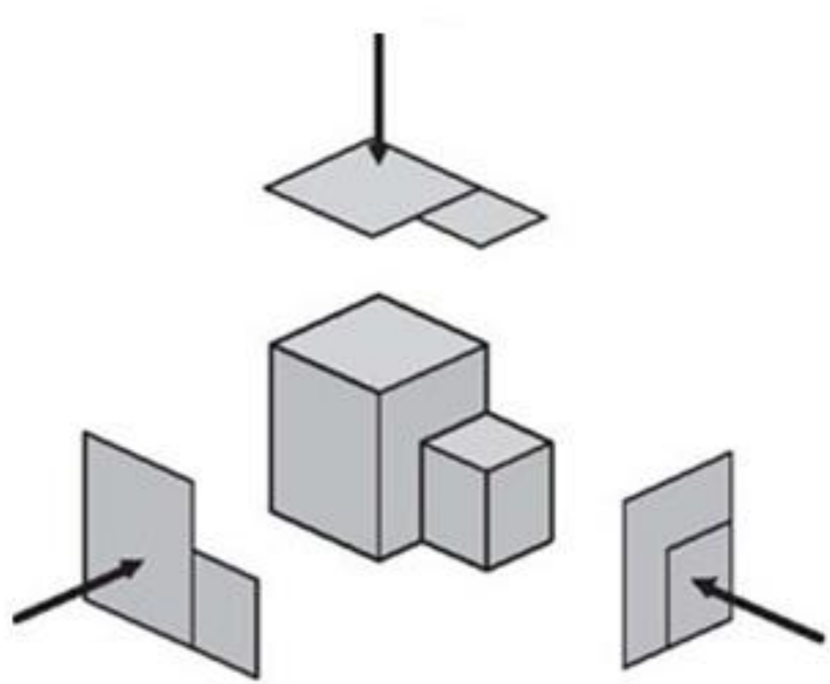
- This aligns u to Xw axis, v to Yw axis and n to Zw axis.
- Finally composite matrix for world to viewing coordinate transformation is given by,

  $$M_{wc,vc} = R \cdot T$$

- This transformation is applied to object's coordinate to transfer them to the viewing reference frame.

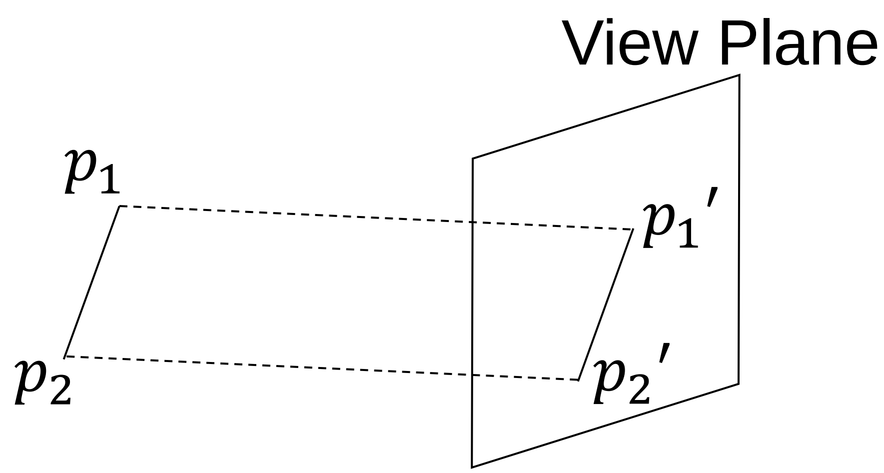
Projections

- Process of converting three-dimensional coordinates into two-dimensional scene is known as **projection**.
- There are two projection methods namely,
  1. Parallel Projection.
  2. Perspective Projection.



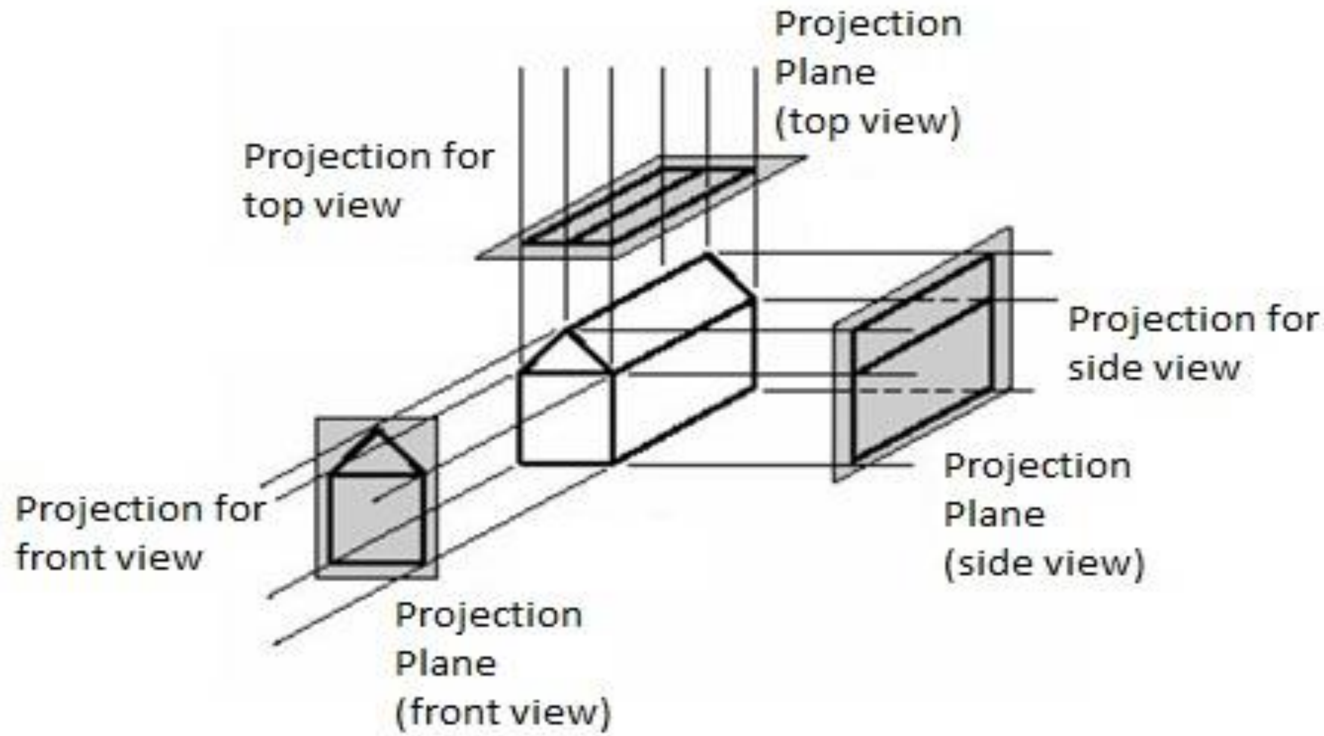
Parallel Projections

- In a parallel projection, coordinate positions are transformed to the view plane along parallel lines.
- We can specify a parallel projection with a projection vector that defines the direction for the projection lines.
- It is further divide into two types,
  1. Orthographic parallel projection.
  2. Oblique parallel projection.



Orthographic Parallel Projection

- When the projection lines are perpendicular to the view plane, we have an orthographic parallel projection.
- Orthographic projections are most often used to produce the front, side, and top views of an object.



Contd.

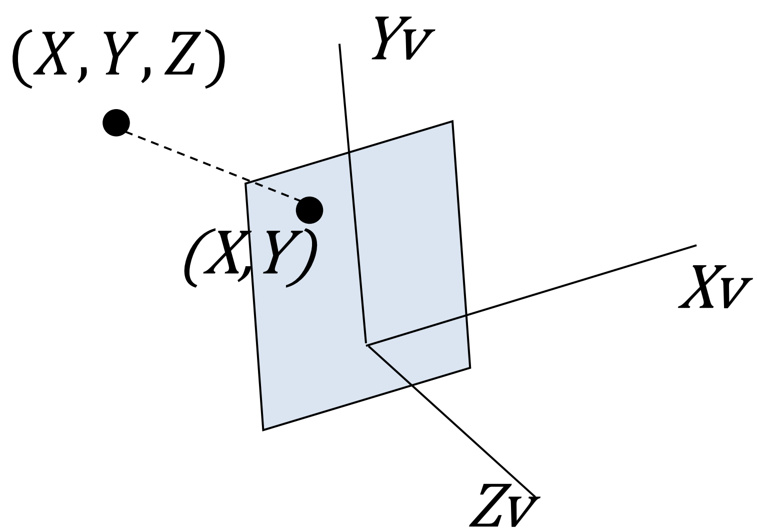
- Engineering and architectural drawings commonly use orthographic projections.
- We can also form orthographic projections that display more than one face of an object.
- Such view are called **axonometric orthographic projections**. Very good example of it is **Isometric** projection.
- Transformation equations for an orthographic parallel projection are straight forward.

Contd.

- If the view plane is placed at position z_{vp} along the z_v axis.
- Then any point (x, y, z) in viewing coordinates is transformed to projection coordinates as,

  $x_p = x, \qquad y_p = y$

- Original z-coordinate value is preserved for the depth information.

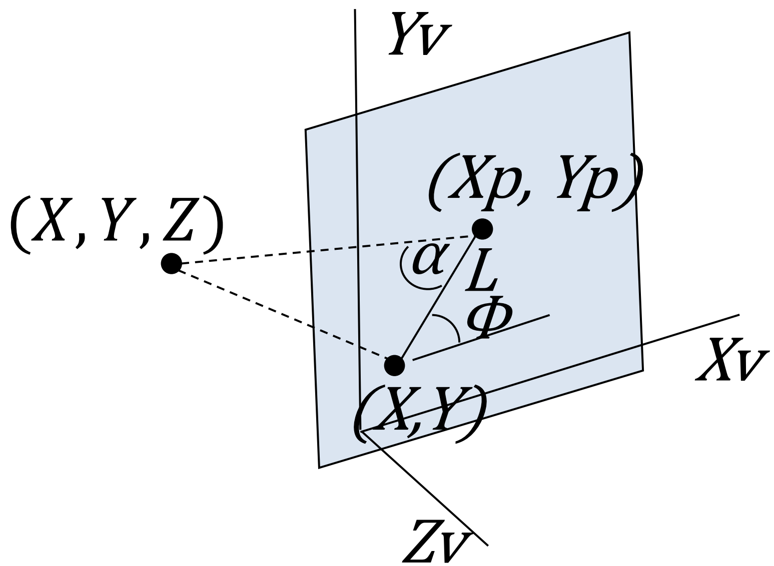


Oblique Parallel Projection

- An oblique projection is obtained by projecting points along parallel lines that are not perpendicular to the projection plane.
- (X, Y, Z) is a point of which we are taking oblique projection (Xp, Yp) on the view plane and point (X, Y) on view plane is orthographic projection of (X, Y, Z).
- Now from figure using trigonometric rules we can write,

$$x_p = x + L\,cos\,\emptyset$$

$$y_p = y + L\,sin\,\emptyset$$



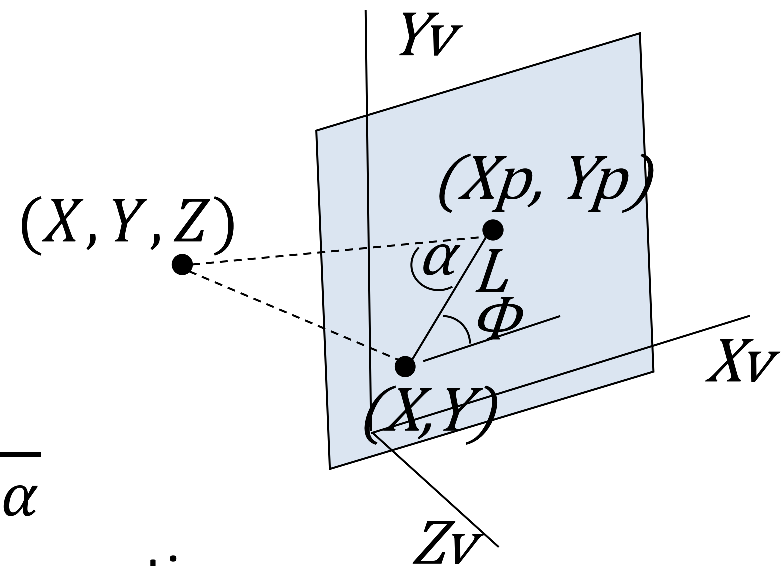
Contd.

- Length L depends on the angle α and the z coordinate of the point to be projected,

$$tan\,\alpha = \frac{Z}{L}$$

$$L = \frac{Z}{tan\,\alpha}$$

$$L = ZL_1\,, \qquad Where\ L_1 = \frac{1}{tan\,\alpha}$$



- Now put the value of L in projection equation.

$$x_p = x + ZL_1\ cos\,\emptyset$$

$$y_p = y + ZL_1\ sin\,\emptyset$$

Contd.

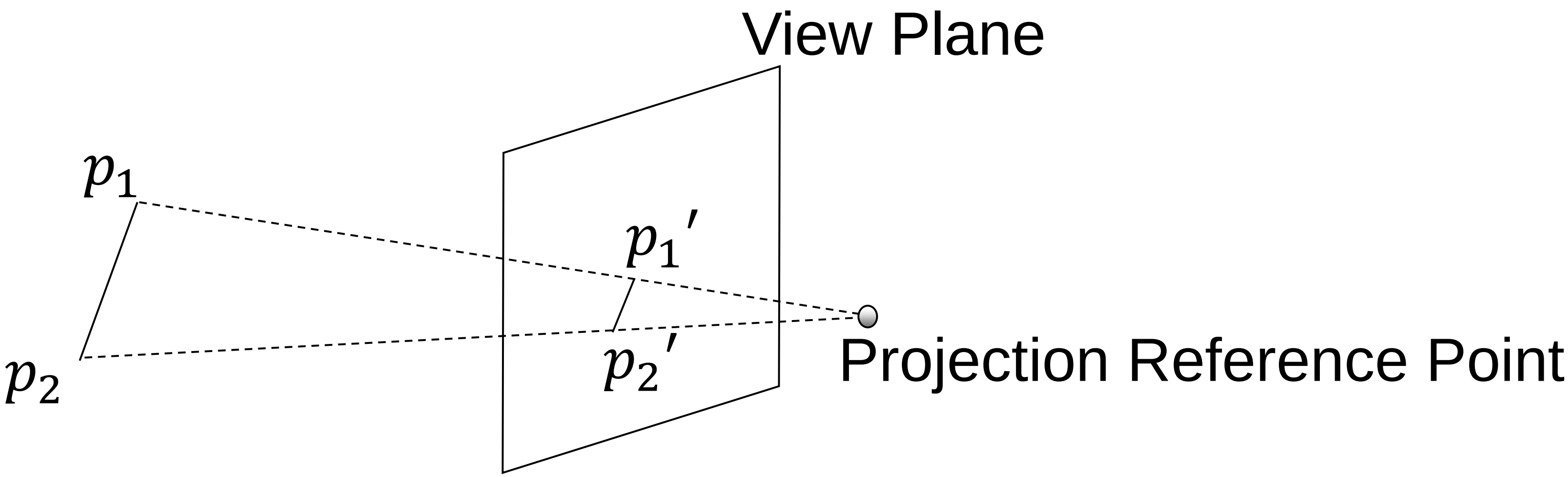
- Transformation matrix for this equation,

$$M_{parallel} = \begin{bmatrix} 1 & 0 & L_1\ cos\,\emptyset & 0 \\ 0 & 1 & L_1\ sin\,\emptyset & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- This equation can be used for any parallel projection.
- For orthographic projection L_1=0 and so whole term which is multiply with z component is zero.
- When value of $\mathbf{tan}\,\boldsymbol{\alpha} = \mathbf{1}$ projection is known as **Cavalier projection**.
- When value of $\mathbf{tan}\,\boldsymbol{\alpha} = \mathbf{2}$ projection is known as **Cabinet projection**.

Perspective Projection

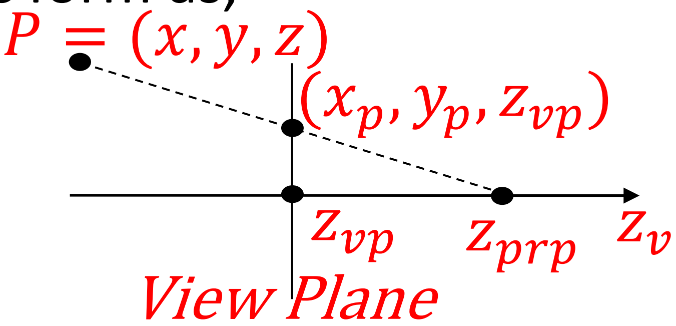
- In perspective projection object positions are transformed to the view plane along lines that converge to a point called the **projection reference point** (or **center of projection** or **vanishing point**).



Contd.

- Suppose we set the projection reference point at position z_{prp} along the z_v axis.
- We place the view plane at z_{vp} as shown in figure.
- We can write equations describing coordinate positions along this perspective projection line in parametric form as,

$$x' = x - xu$$
$$y' = y - yu$$
$$z' = z - (z - z_{prp})u$$



- Here parameter u takes the value from $0\ to\ 1$, which is depends on the position of object, view plane, and projection reference point.

Contd.

- For obtaining value of u we will put $z' = z_{vp}$ and solve equation of z'.

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

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

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

- Now substituting value of u in equation of x' and y' we will obtain,

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

$$y_p = y\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) = y\left(\frac{d_p}{z_{prp} - z}\right), Where\ d_p = z_{prp} - z_{vp}$$

Contd.

- Using 3D homogeneous-coordinate representations, we can write the perspective projection transformation matrix form as,

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{vp}/d_p & z_{vp}\left(z_{prp}/d_p\right) \\ 0 & 0 & -1/d_p & z_{prp}/d_p \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- In this representation, the homogeneous factor is,

$$h = \frac{z_{prp} - z}{d_p} \; and$$

$$x_p = x_h/h \;\; and \; y_p = y_h/h$$

Contd.

- There are number of special cases for the perspective transformation equations.
- If view plane is taken to be uv plane, then $\boldsymbol{z_{vp}} = \mathbf{0}$ and the projection coordinates are,

$$x_p = x\left(\frac{z_{prp}}{z_{prp} - z}\right) = x\left(\frac{1}{1 - z/z_{prp}}\right)$$

$$y_p = y\left(\frac{z_{prp}}{z_{prp} - z}\right) = y\left(\frac{1}{1 - z/z_{prp}}\right)$$

Contd.

- If we take projection reference point at origin than $\boldsymbol{z_{prp} = 0}$ and the projection coordinates are,

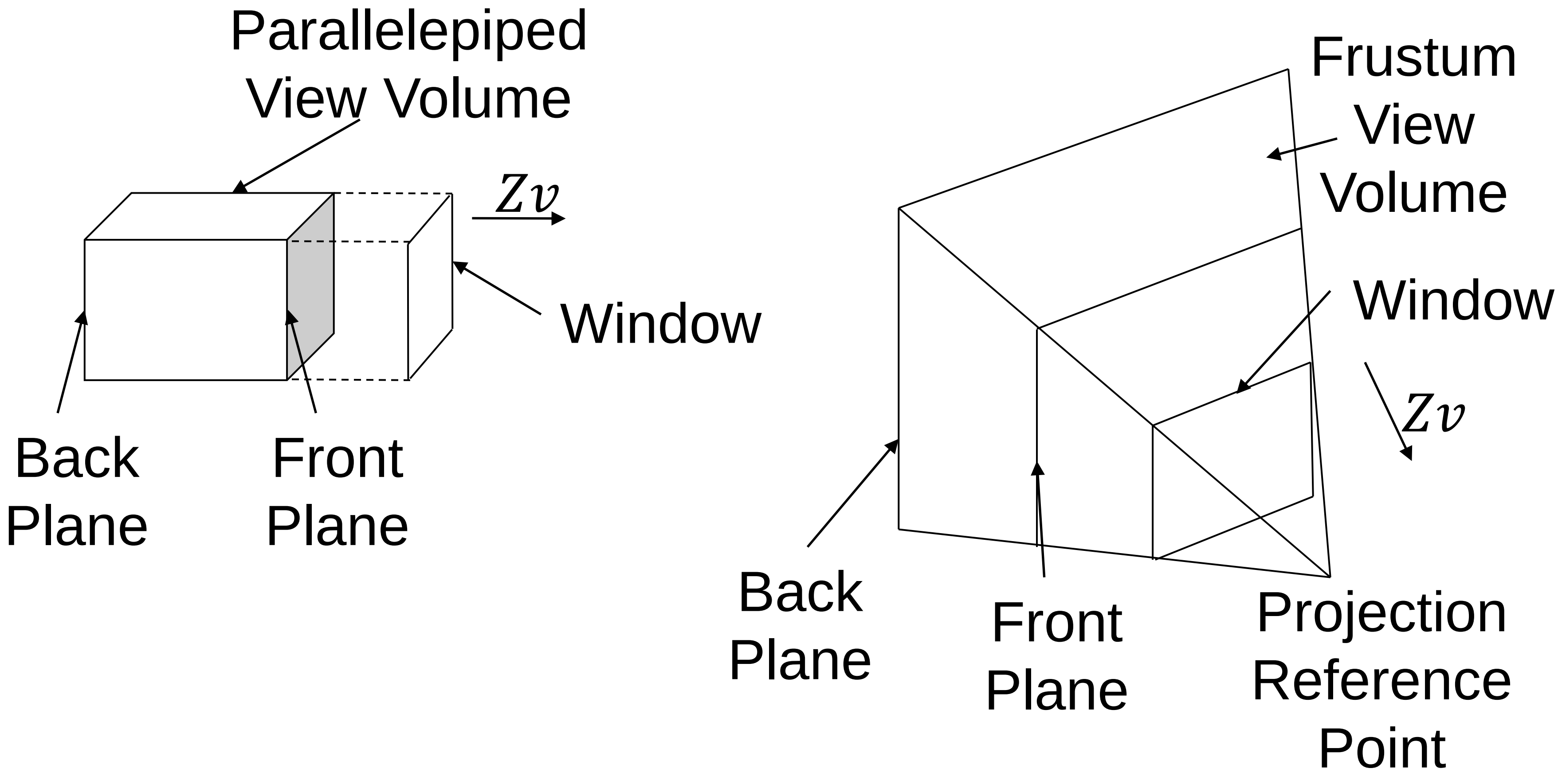
$$x_p = x\left(\frac{z_{vp}}{z}\right) = x\left(\frac{1}{z/z_{vp}}\right)$$

$$y_p = y\left(\frac{z_{vp}}{z}\right) = y\left(\frac{1}{z/z_{vp}}\right)$$

Contd.

- The vanishing point for any set of lines that are parallel to one of the principal axes of an object is referred to as a principal vanishing point.
- With the orientation of the projection plane, and perspective projections are accordingly classified as,
    1. One-point
    2. Two-point
    3. Three-point projections.
- The number of principal vanishing points in a projection is determined by the number of principal axes intersecting the view plane.

View Volumes and General Projection Transformations



(a) Parallel Projection (b) Perspective Projection

Contd.

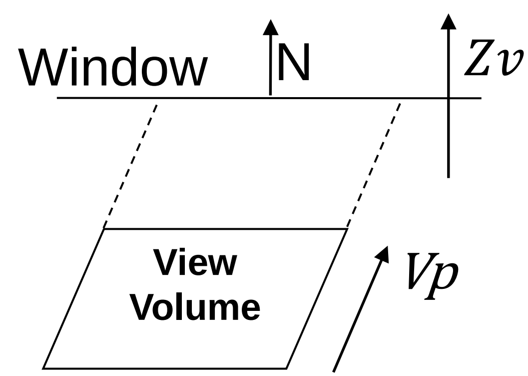
- Based on view window we can generate different image of the same scene.
- Volume which is appears on the display is known as view volume.
- Given the specification of the view window, we can set up a view volume using the window boundaries.
- Only those objects within the view volume will appear in the generated display on an output device, all others are clipped from the display.
- The size of the view volume depends on the size of the window.
- Shape of the view volume depends on the type of projection to be used to generate the display.

Contd.

- A finite view volume is obtained by limiting the extent of the volume in the z_v direction.
- This is done by specifying positions for one or two additional boundary planes.
- These z_v-boundary planes are referred to as the **front plane** and **back plane**, or the **near plane** and the **far plane**, of the viewing volume.
- Orthographic parallel projections are not affected by view-plane positioning.
- Because the projection lines are perpendicular to the view plane regardless of its location.

General Parallel-Projection Transformation

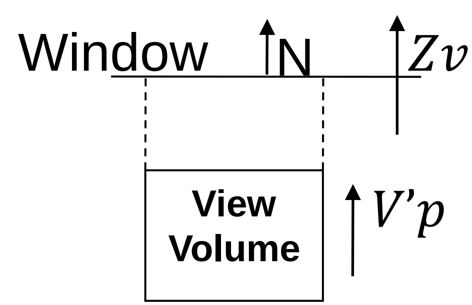
- Oblique projections may be affected by view-plane positioning, depending on how the projection direction is to be specified.
- Obtain transformation matrix for parallel projection which is applicable to both orthographic as well as oblique projection.
- parallel projection is specified with a projection vector from the projection reference point to the view window.



Contd.

- Now we will apply shear transformation
- View volume will convert into regular parallelepiped and projection vector will become parallel to normal vector N.
- Let's consider projection vector $V_p = (p_x, p_y, p_z)$.
- We need to determine the elements of a shear matrix
- That will align the projection vector $\boldsymbol{V_p}$ with the view plane normal vector $\boldsymbol{N}$. This transformation can be expressed as,

$$V_p{}' = M_{parallel} \cdot V_p$$



Contd.

$$V_p{}' = \begin{bmatrix} 0 \\ 0 \\ p_z \\ 1 \end{bmatrix}$$

where $\boldsymbol{M_{parallel}}$ is equivalent to the parallel projection matrix and represents a $z - axis$ shear of the form,

$$M_{parallel} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Contd.

- Now from above equation we can write,

$$\begin{bmatrix} 0 \\ 0 \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

- From matrix we can write,

$$0 = p_x + ap_z$$

$$0 = p_y + bp_z$$

So

$$a = \frac{-p_x}{p_z}, \qquad b = \frac{-p_y}{p_z}$$

Contd.

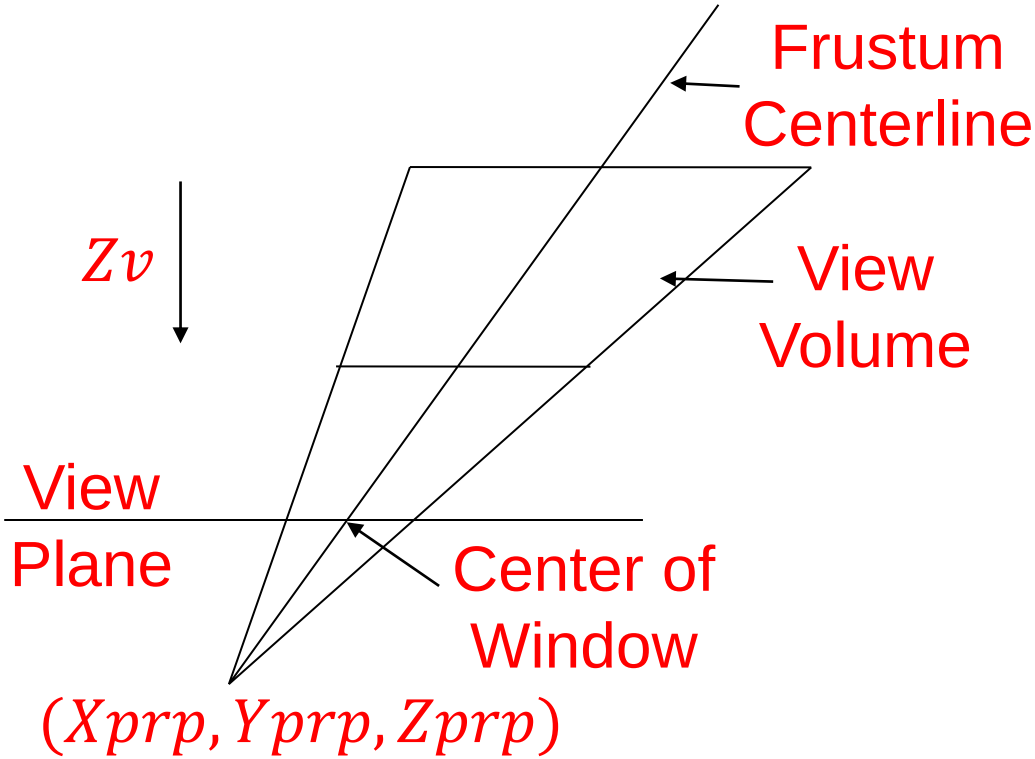
- Thus, we have the general parallel-projection matrix in terms of the elements of the projection vector as,

$$M_{parallel} = \begin{bmatrix} 1 & 0 & \dfrac{-p_x}{p_z} & 0 \\ 0 & 1 & \dfrac{-p_y}{p_z} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- For an orthographic parallel projection, $p_x = p_y = 0$, and is the identity matrix.

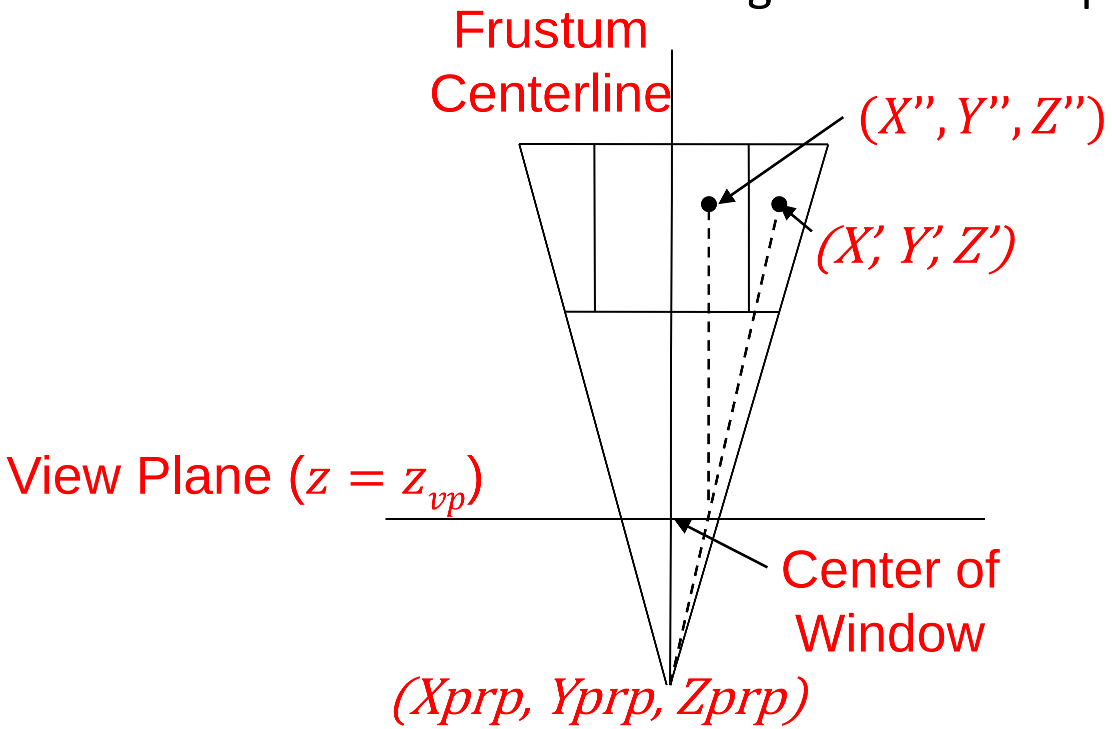
General Perspective-Projection Transformations

- The projection reference point can be located at any position in the viewing system, except on the view plane or between the front and back clipping planes.



Contd.

- We can obtain the general perspective-projection transformation with the following two operations,
  1. Shear the view volume so that the center line of the frustum is perpendicular to the view plane.
  2. Scale the view volume with a scaling factor that depends on $1/z$.



Contd.

- With the projection reference point at a general position $(X_{prp}, Y_{prp}, Z_{prp})$ the transformation involves a combination of $z-axis$ shear and a translation,

$$M_{shear} = \begin{bmatrix} 1 & 0 & a & -az_{prp} \\ 0 & 1 & b & -bz_{prp} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where the shear parameters are,

$$a = -\frac{x_{prp} - \frac{xw_{min} + xw_{max}}{2}}{z_{prp}}, \& \ b = -\frac{y_{prp} - \frac{yw_{min} + yw_{max}}{2}}{z_{prp}}$$

Contd.

- Points within the view volume are transformed by this operation as,

$$x' = x + a(z - z_{prp})$$
$$y' = y + b(z - z_{prp})$$
$$z' = z$$

- After shear we apply scaling operation. Equation for that are,

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

Contd.

- Homogeneous matrix for this transformation is,

$$M_{scale} = \begin{bmatrix} 1 & 0 & \dfrac{-x_{prp}}{z_{prp} - z_{vp}} & \dfrac{x_{prp} z_{vp}}{z_{prp} - z_{vp}} \\ 0 & 1 & \dfrac{-y_{prp}}{z_{prp} - z_{vp}} & \dfrac{y_{prp} z_{vp}}{z_{prp} - z_{vp}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \dfrac{-1}{z_{prp} - z_{vp}} & \dfrac{z_{prp}}{z_{prp} - z_{vp}} \end{bmatrix}$$

- Therefore the general perspective-projection transformation is obtained by equation,

$$M_{perspective} = M_{scale} \cdot M_{shear}$$

Thank You