# 3D TRANSFORMATIONS

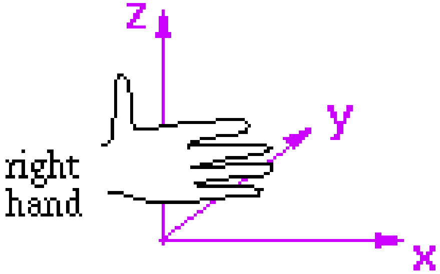
# 3D transformations

- Instead of creating every new object from scratch, we may transform some existing objects
- We also need to transform objects from one space to another, e.g., world space to view space/camera space
- Instead of performing each transformation alone, we accumulate the transformations in a matrix before applying them to objects

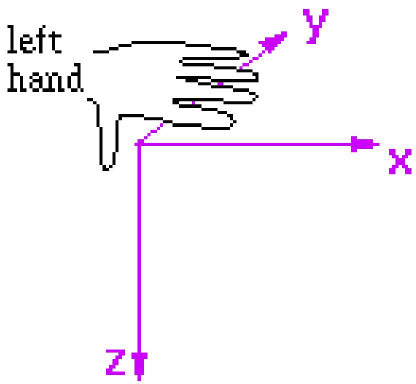
# 3D transformations

- Transformations are measured in a coordinate matrix
- There are two possible ways of specifying the Z-axis, which gives rise to a **left-handed(**suitable for screens) or a **right-handed** system(consisent with math)

RHCS




LHCS



# 3D Point

- We will consider points as column vectors. Thus, a typical point with coordinates (x, y, z) is represented as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

# 3D Point Homogenous Coordinate

- A 3D point **P** is represented in homogeneous coordinates by a 4-dim. Vect:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

# 2D Transformation

- Scaling
  - X' = aX
  - Y' = aY
- Translation
  - X'=x+b
  - Y'=y+b
- Rotation

$x' = r\cos(\theta + \phi) =$

$\underline{r\cos\theta}\cos\phi - \underline{r\sin\theta}\sin\phi$

$y' = r\sin(\theta + \phi) =$

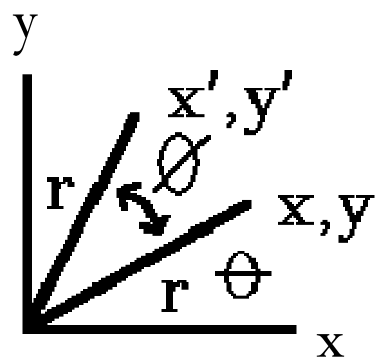
$\underline{r\sin\theta}\cos\phi + \underline{r\cos\theta}\sin\phi$

so

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

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

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \cos\varphi & \sin\varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$




$x = r\cos\theta$

$y = r\sin\theta$

# Homogeneous Coordinates

- In 2D, three elements can be used to represent a point, e.g., (x,y,1) to represent (x,y) or (xh,yh,h) h≠0 to represent (x,y)
- (x,y,h) is a homogeneous coordinate representing the point (x/h, y/h).
- The 3D homogenous representation is given by

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

# 2D Transformation Matrices

- Translation matrix:

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Scaling matrix:

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

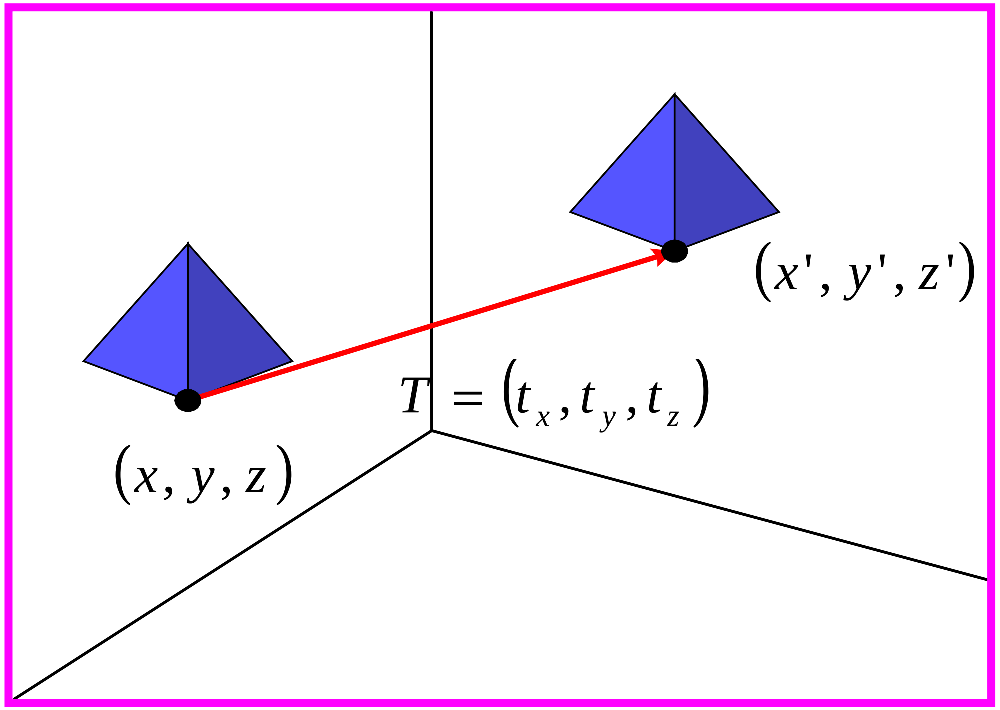
- Rotation matrix:

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

# 3D Translation

- **P** is **translated** to **P'** by:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

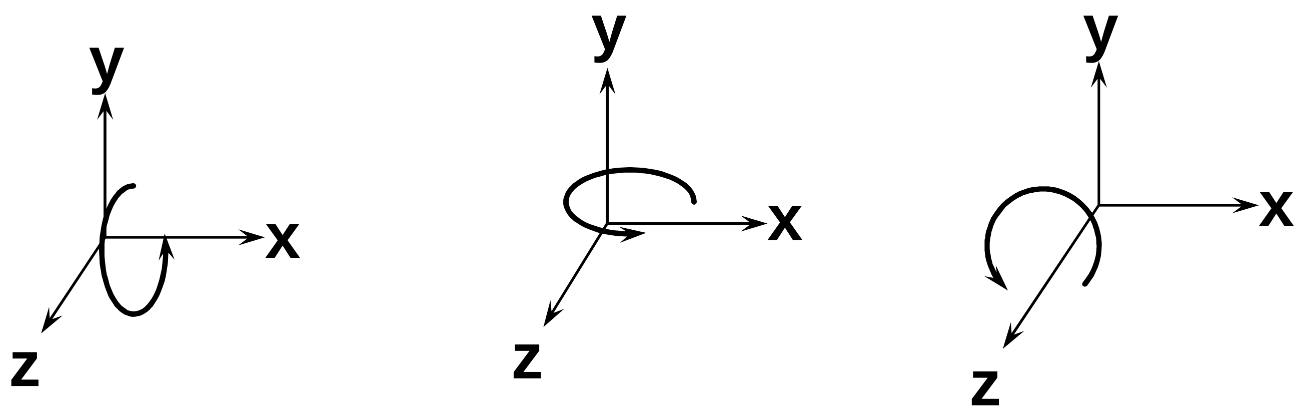


# 3D Transformation- Translation

- The matrix representation is equal to three equations:
- $x' = x + t_x$ $y' = y + t_y$ $z' = z + t_z$
- An object is translated in three dimensions by transforming each of the defining points of the object.
- For polygon surfaces, translate each vertex of the surface and redraw the polygon facets in new position

# 3D Rotation

- In 2D, rotation is about a point
- In 3D, rotation is about a vector, which can be done through rotations about x, y or z axes
- Positive rotations are anti-clockwise, negative rotations are clockwise, when looking down a positive axis towards the origin

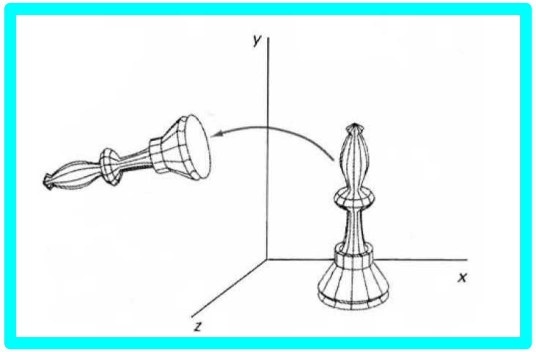


# Coordinate Axis Rotations

- **Z-axis rotation:** For z axis same as 2D rotation:

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

$$\mathbf{P}' = \mathbf{R}_z(\theta) \cdot \mathbf{P}$$

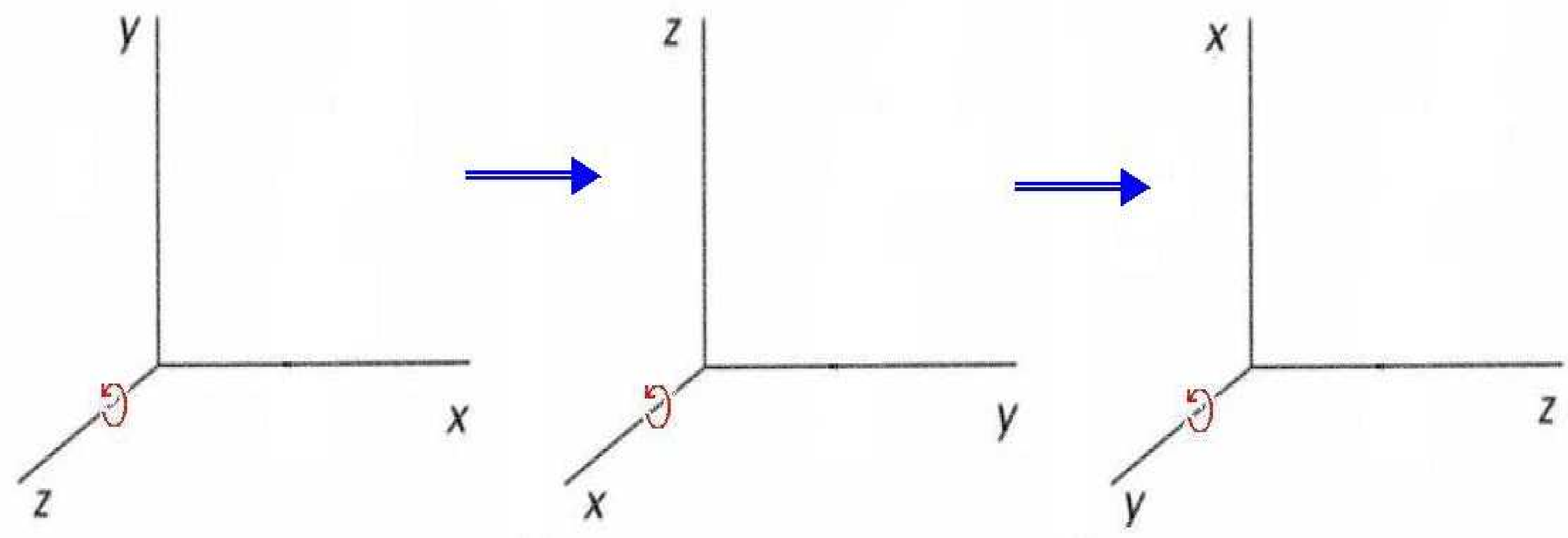


# 3D Rotation

- The 2D equations are easily extended to 3D z-axis rotation
  - $x' = x \cos \Theta - y \sin \Theta$
  - $y' = x \sin \Theta + y \cos \Theta$
  - $z' = z$
  - $P' = R_z(\Theta) . P$
- Transformations equations for rotations about other two axes can be obtained with cyclic permutation of the coordinate parameters x ,y and z
- $x \rightarrow y \rightarrow z \rightarrow x$

# Coordinate Axis Rotations

- Obtain rotations around other axes through cyclic permutation of coordinate parameters:

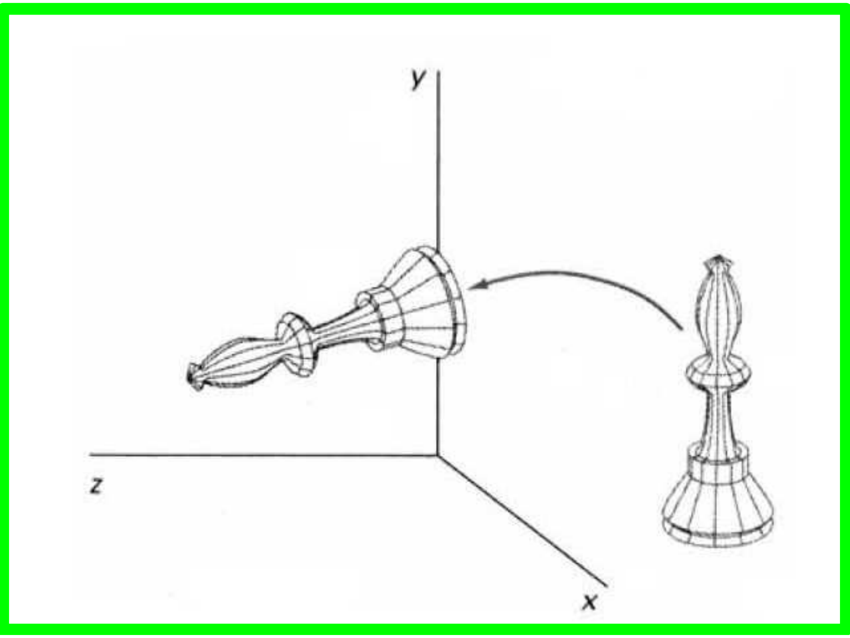


# Coordinate Axis Rotations

- **X-axis rotation:**

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

$$\mathbf{P}' = \mathbf{R}_x(\theta) \cdot \mathbf{P}$$

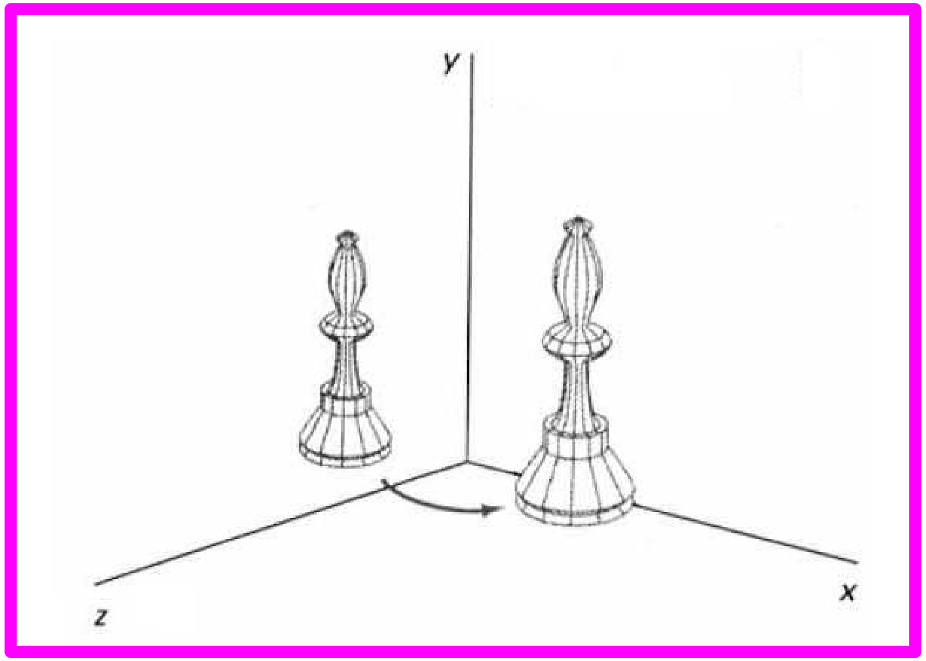


# Coordinate Axis Rotations

- **Y-axis rotation:**

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

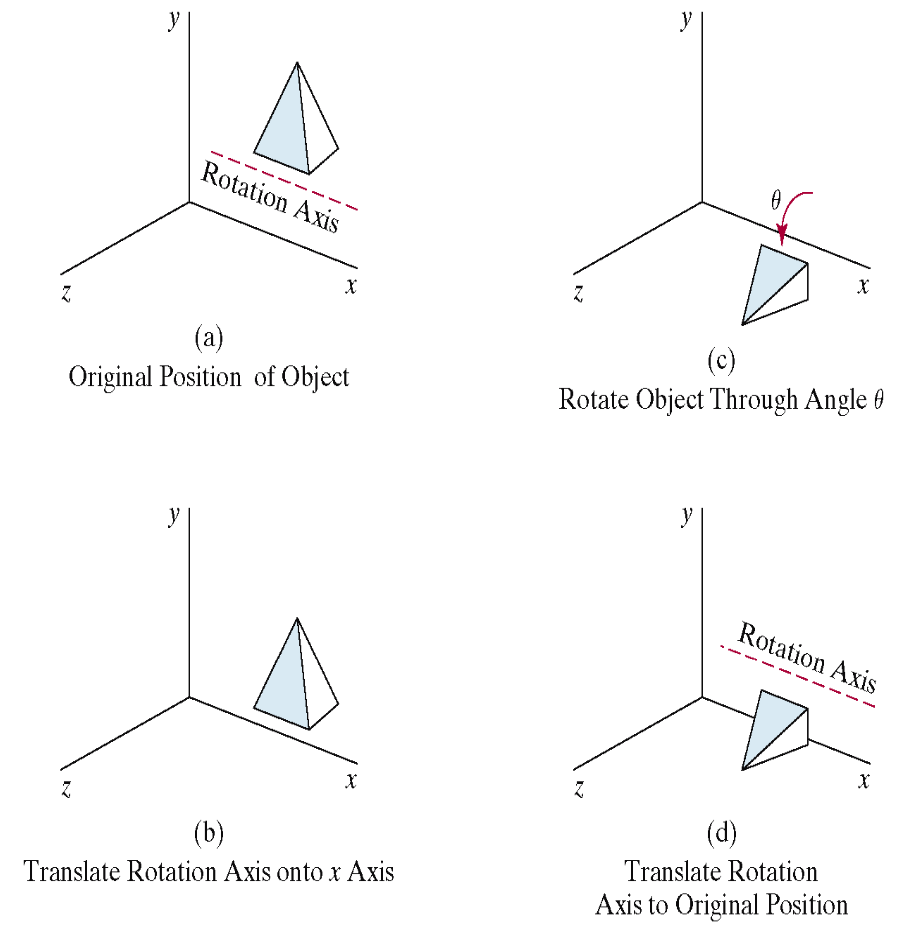
$$\mathbf{P}' = \mathbf{R}_y(\theta) \cdot \mathbf{P}$$



# General Three Dimensional Rotations

# General Three Dimensional Rotations

**Rotation axis parallel with coordinate axis (Example x axis):**




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

# General Three Dimensional Rotations

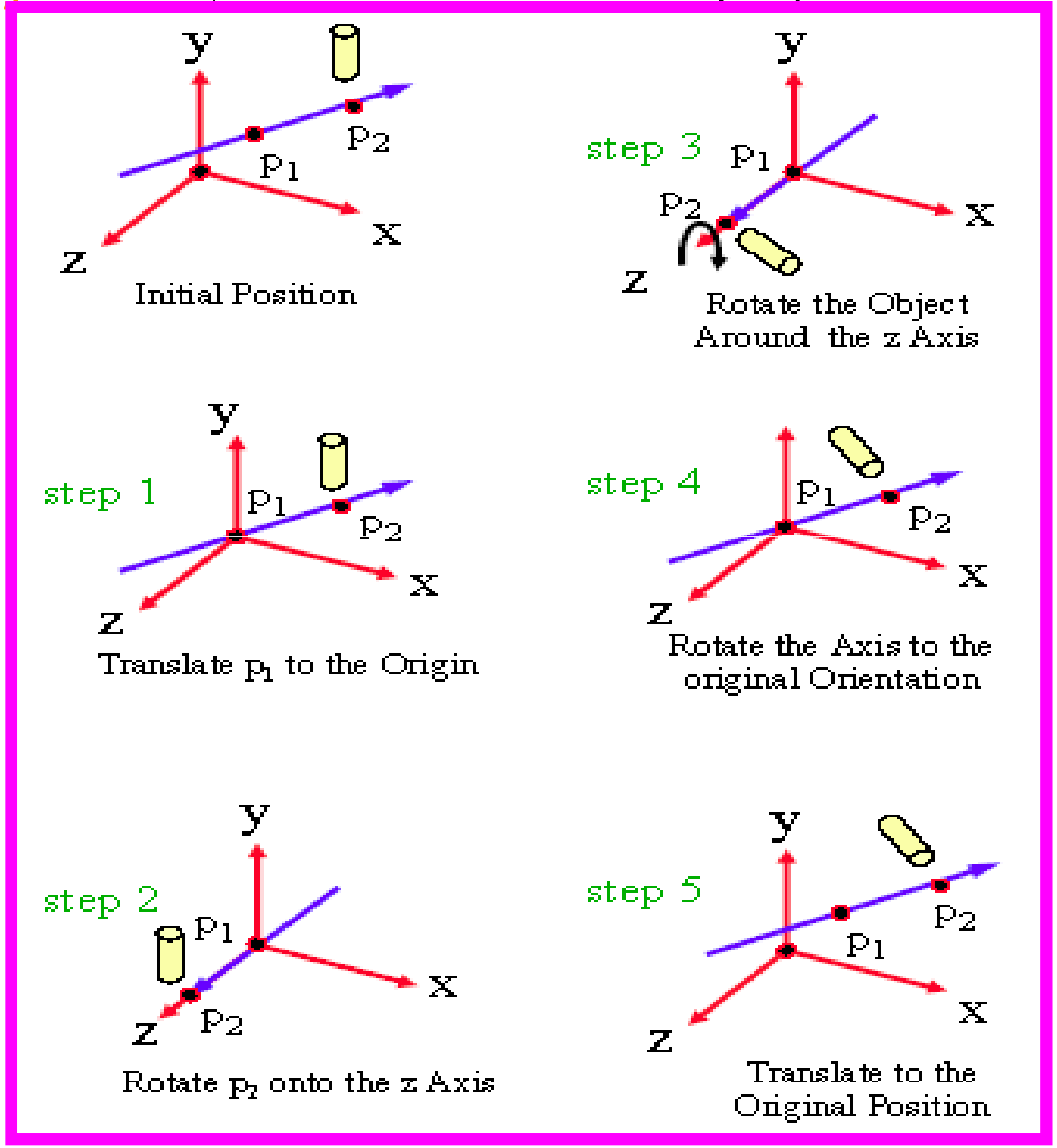
$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R}_x^{-1}(\alpha) \cdot \mathbf{R}_y^{-1}(\beta) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\alpha) \cdot \mathbf{T}$$

- A rotation matrix for any axis that does not coincide with a coordinate axis can be set up as a composite transformation involving combination of translations and the coordinate-axes rotations:

1. Translate the object so that the rotation axis passes through the coordinate origin
2. Rotate the object so that the axis rotation coincides with one of the coordinate axes
3. Perform the specified rotation about that coordinate axis
4. Apply inverse rotation axis back to its original orientation
5. Apply the inverse translation to bring the rotation axis back to its original position

# General Three Dimensional Rotations

**An arbitrary axis** (with the rotation axis projected onto the Z axis):




$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R}_x^{-1}(\alpha) \cdot \mathbf{R}_y^{-1}(\beta) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\alpha) \cdot \mathbf{T}$$

# *Rotations for an arbitrary axis*

- Steps:
- 1. Normalize vector $u$
- 2. Compute $\alpha$
- 3. Compute $\beta$
- 4. Create rotation matrix

# *Rotations for an arbitrary axis*

The rotation axis is defined by two points, then the direction of rotation is to be counterclockwise along the axis from p2 to p1

An axis vector defined by the two points

$$\mathbf{V} = \mathbf{P}_2 - \mathbf{P}_1 = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

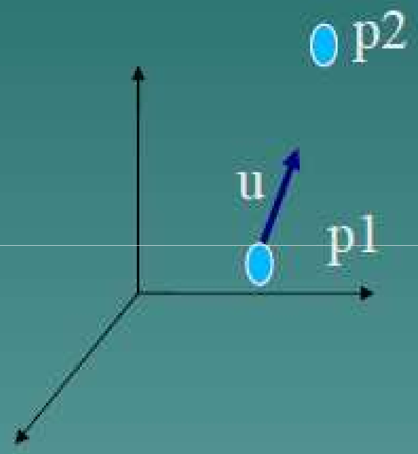
$\mathbf{u}$ is the unit vector along $\mathbf{V}$:

$$\mathbf{u} = \frac{\mathbf{V}}{|\mathbf{V}|} = (a, b, c)$$

# Rotations for an arbitrary axis

- First step: Translate $\mathbf{P}_1$ to origin:

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




- Next step: Align $\mathbf{u}$ with the $z$ axis

we need two rotations:

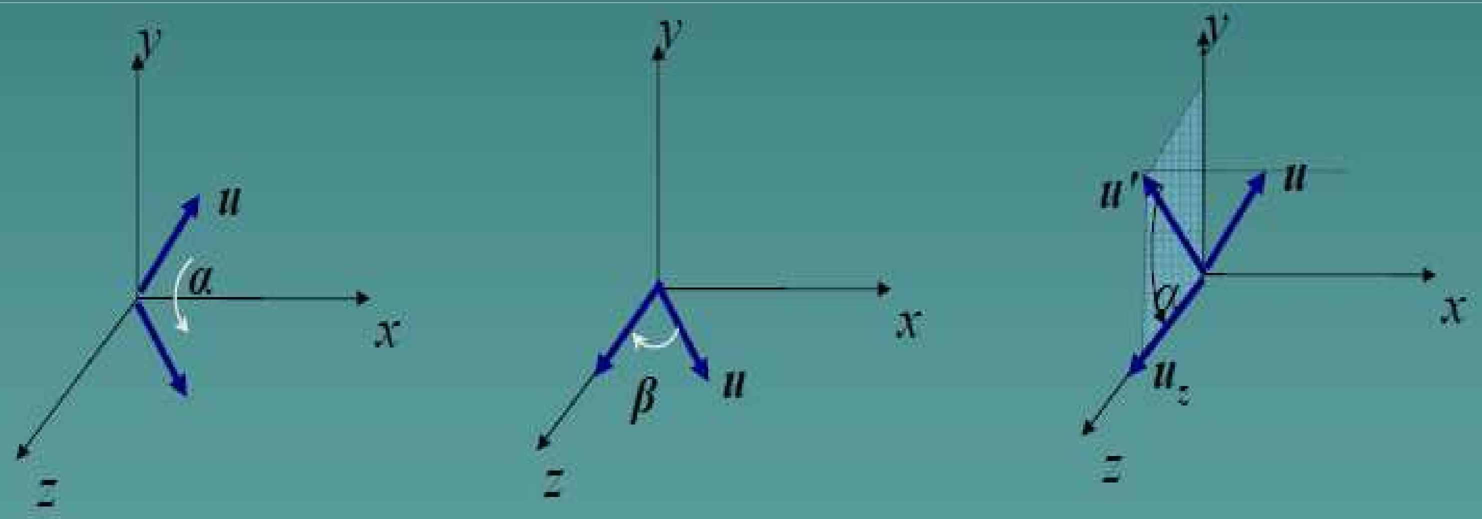
- rotate around $x$ axis to get $\mathbf{u}$ onto the $xz$ plane,
- rotate around $y$ axis to get $\mathbf{u}$ aligned with $z$ axis.

# *Rotations for an arbitrary axis*

Align **u** with the $z$ axis

1) rotate around $x$ axis to get **u** into the $xz$ plane,

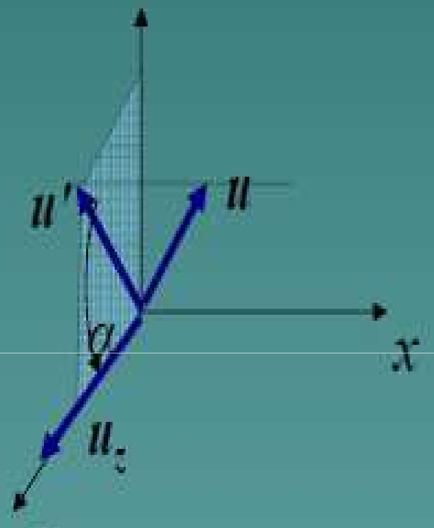
2) rotate around $y$ axis to get **u** aligned with the $z$ axis



# Dot product and cross product

- Since rotation calculations involves sine and cosine functions standard vector operations are used to obtain the elements of rotation matrices
- The dot product allows to determine cosine terms:
  - $V \cdot U = |V|\ |U| \cos \Theta$
- The cross product determine sine term
- $|v| * |u| * \sin(a),$

# Rotations for an arbitrary axis




$$\mathbf{u}' = (0, b, c)$$

Projection of **u** on $yz$ plane

We need cosine and sine of α for rotation

$$\cos\alpha = \frac{\mathbf{u}' \cdot \mathbf{u}_z}{|\mathbf{u}'||\mathbf{u}_z|} = \frac{c}{d} \qquad d = \sqrt{b^2 + c^2}$$

$$\mathbf{u}' \times \mathbf{u}_z = \mathbf{u}_x |\mathbf{u}'||\mathbf{u}_z| \sin\alpha = \mathbf{u}_x b$$

$$b = d \sin\alpha$$

$$\cos\alpha = \frac{c}{d} \qquad \sin\alpha = \frac{b}{d}$$

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{d} & -\frac{b}{d} & 0 \\ 0 & \frac{b}{d} & \frac{c}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Rotations for an arbitrary axis

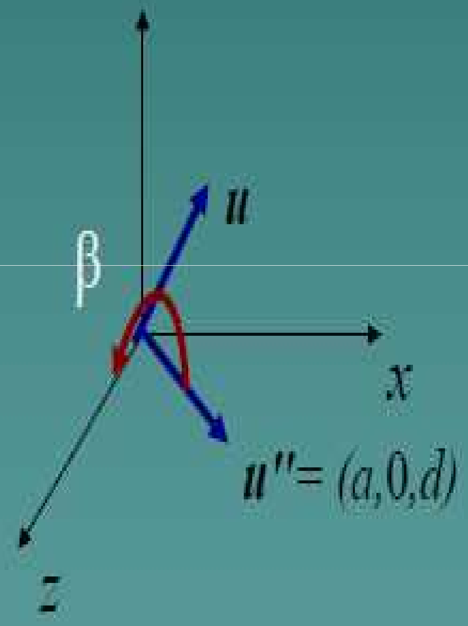
2) rotate around $y$ axis to get $\mathbf{u}$ aligned with the $z$ axis

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

$$\mathbf{u}'' \times \mathbf{u}_z = \mathbf{u}_y |\mathbf{u}''| |\mathbf{u}_z| \sin\beta = \mathbf{u}_y \cdot (-a)$$

$$\cos\beta = d \qquad \sin\beta = -a$$

$$\mathbf{R}_y(\beta) = \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



# *Rotations for an arbitrary axis*

- Aligned the rotation axis with positive z axis.
- Specified rotation angle given by

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

$$\mathbf{R}(\theta) = \mathbf{T}(x_1, y_1, z_1) \cdot \mathbf{R}_x(-\alpha) \cdot \mathbf{R}_y(-\beta) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\alpha) \cdot \mathbf{T}(-x_1, -y_1, -z_1)$$

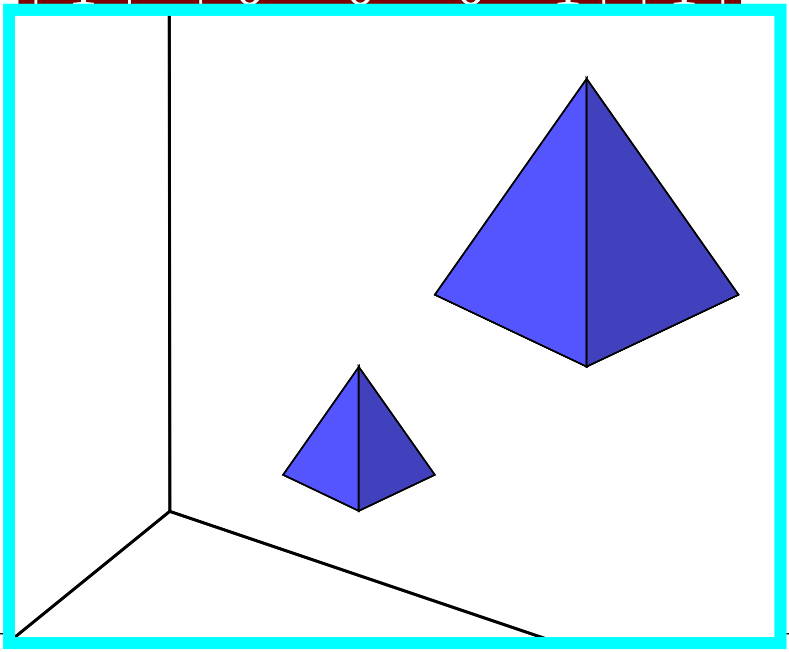
# 3D Scaling

## 3D Scaling

- **About origin:** Changes the size of the object and repositions the object relative to the coordinate origin.

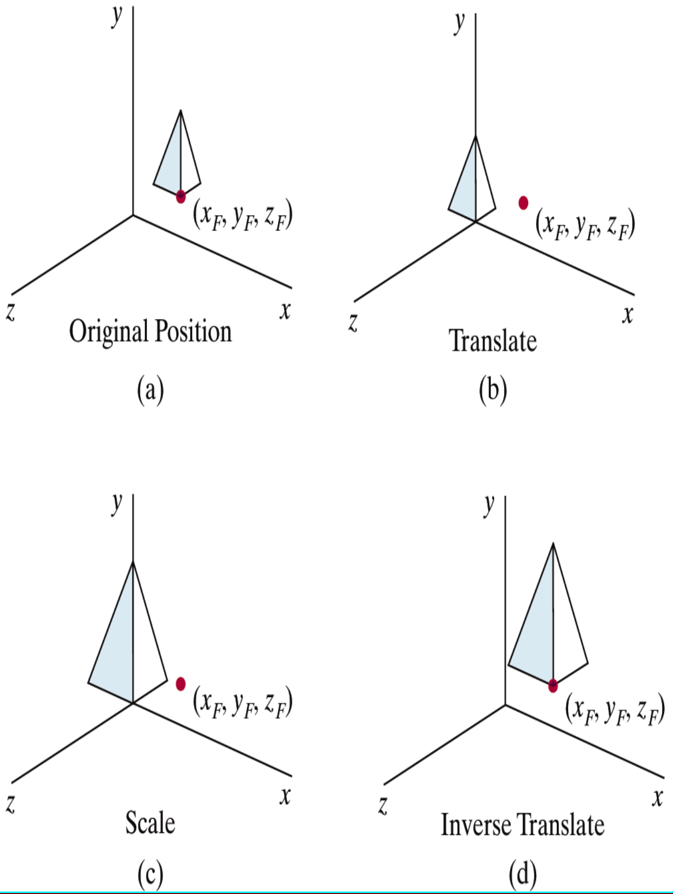
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$$



# 3D Scaling

- **About any fixed point:**




$$\mathbf{T}(x_f, y_f, z_f)\cdot\mathbf{S}(s_x, s_y, s_z)\cdot\mathbf{T}(-x_f, -y_f, -z_f) = \begin{bmatrix} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Composite
# 3D Transformations

# Composite 3D Transformations

- **Same way as in two dimensions:**
  - Multiply matrices
  - Rightmost term in matrix product is the first transformation to be applied

# 3D Reflections

## 3D Reflections

- **About an axis:** equivalent to $180^\circ$ rotation about that axis

# 3D Reflections

## About a plane:

- A reflection through the **xy** plane:

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

- A reflections through the **xz** and the **yz** planes are defined similarly.

# 3D Shearing

# 3D Shearing

- **Modify object shapes**
- **Useful for perspective projections:**
  - E.g. draw a cube (3D) on a screen (2D)
  - Alter the values for **x** and **y** by an amount proportional to the distance from $z_{ref}$

# 3D Shearing

$$M_{zshear} = \begin{bmatrix} 1 & 0 & sh_{zx} & -sh_{zx} \cdot z_{ref} \\ 0 & 1 & sh_{zy} & -sh_{zy} \cdot z_{ref} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$