CS475m - Computer Graphics

Lecture 5 : 3D Transformations

3D Transformations

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S^{-1}(s_x, s_y, s_z) = S\left(\frac{1}{s_x}, \frac{1}{s_y}, \frac{1}{s_z}\right)$$

Scaling

glScalef(s_x, s_y, s_z)

3D Transformations

$$T(l,m,n)=\begin{bmatrix}1 & 0 & 0 & l\\ 0 & 1 & 0 & m\\ 0 & 0 & 1 & n\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$T^{-1}(l,m,n)=T(-l,-m,-n)$$

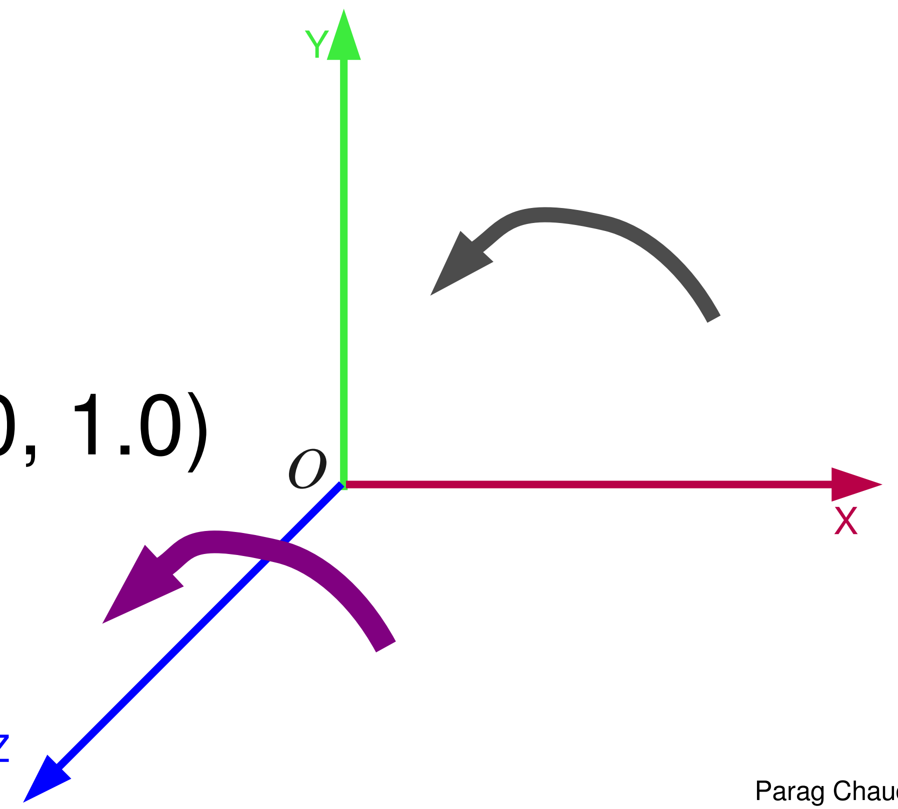
Translation

glTranslatef(l, m, n)

3D Transformations

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

$$R_z^{-1}(\theta) = R_z(-\theta) = R_z^T$$



Rotation about Z axis

glRotatef(angle, 0.0, 0.0, 1.0)

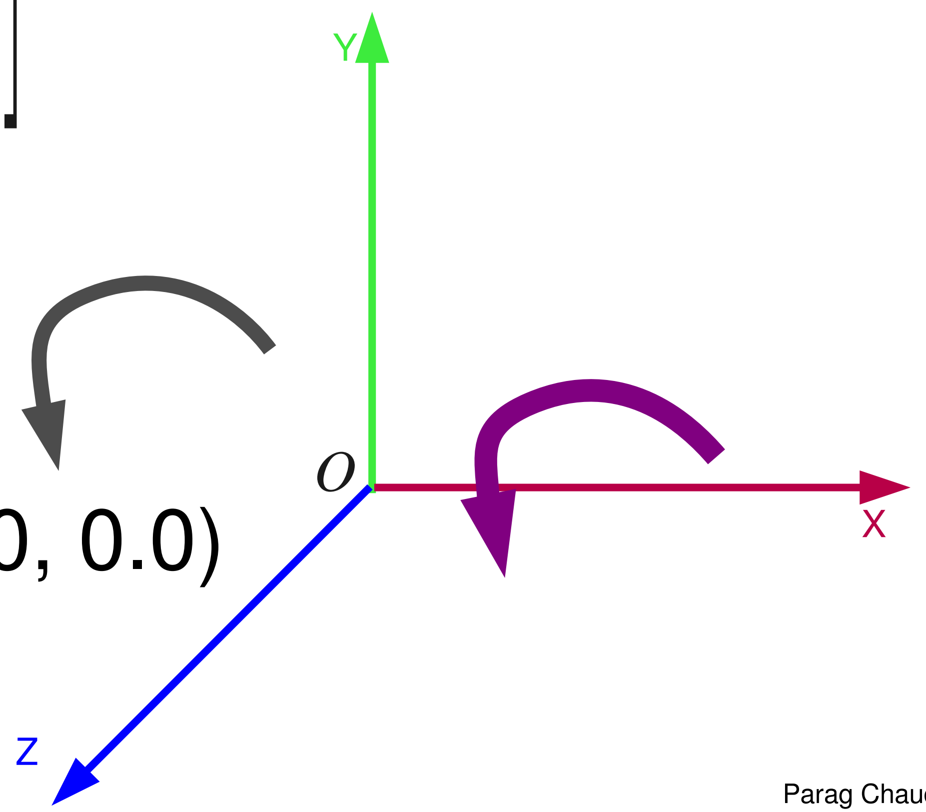
3D Transformations

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

$$R_x^{-1}(\theta) = R_x(-\theta) = R_x^T$$

Rotation about X axis

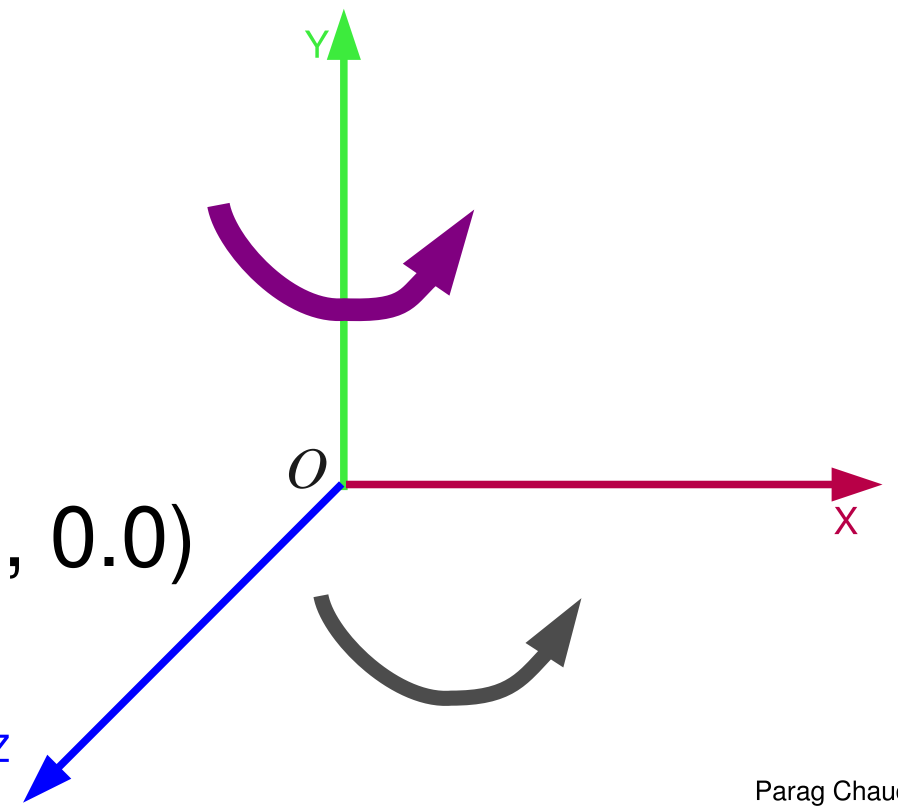
glRotatef(angle, 1.0, 0.0, 0.0)



3D Transformations

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

$$R_y^{-1}(\theta)=R_y(-\theta)=R_y^T$$



Rotation about Y axis

glRotatef(angle, 0.0, 1.0, 0.0)

3D Transformations

In particular for Rotations

$$R^T_{axis}(\theta).R_{axis}(\theta)=R_{axis}(\theta).R^T_{axis}(\theta)=I$$

Rotations are orthogonal matrices.

$$det(R_{axis}(\theta))=1$$

3D Transformations

Shear

$$Sh = \begin{bmatrix} 1 & d & g & 0 \\ b & 1 & h & 0 \\ c & f & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

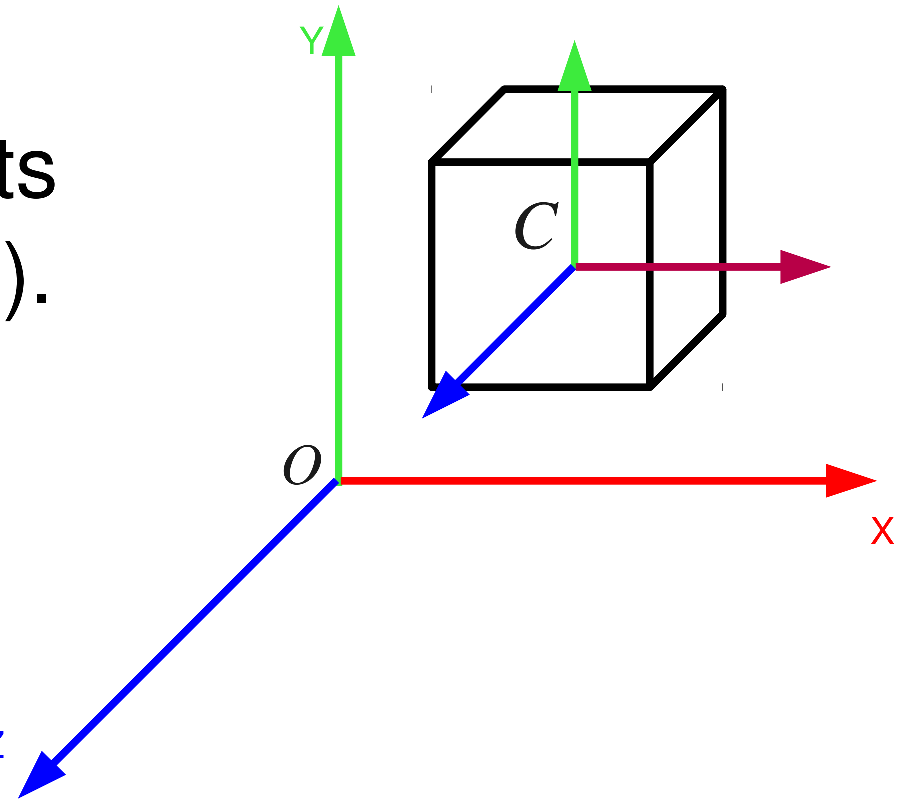
glLoadMatrixf(const Gldouble *m)

glMultMatrixf(const Gldouble *m)

m contains points to 16 consecutive values that are used as the elements of a 4×4 **column-major** matrix.

3D Transformations

Rotating a cube about its center (about the z axis).

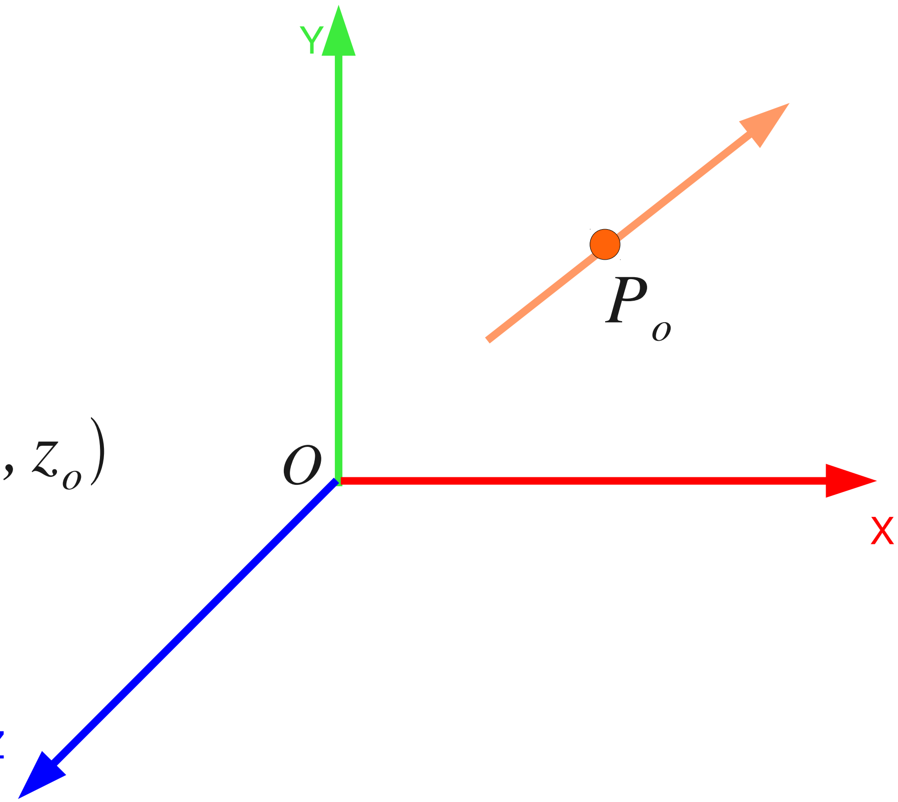


$$P' = T(C).R_z(\theta).T(-C).P$$

3D Transformations

Rotating about
an arbitrary axis.

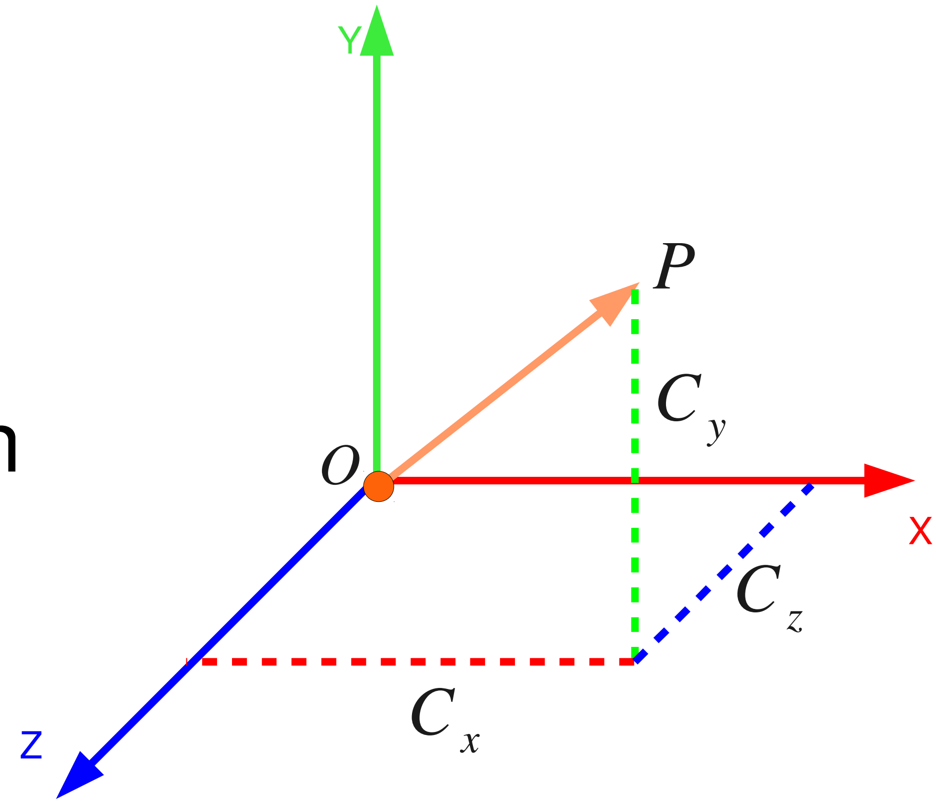
Passing through $P_o(x_o, y_o, z_o)$
and
with direction cosines as
C_x, C_y, C_z
by an angle δ



3D Transformations

Rotating about
an arbitrary axis.

Translate P_o to the origin
using $T(-x_o, -y_o, -z_o)$



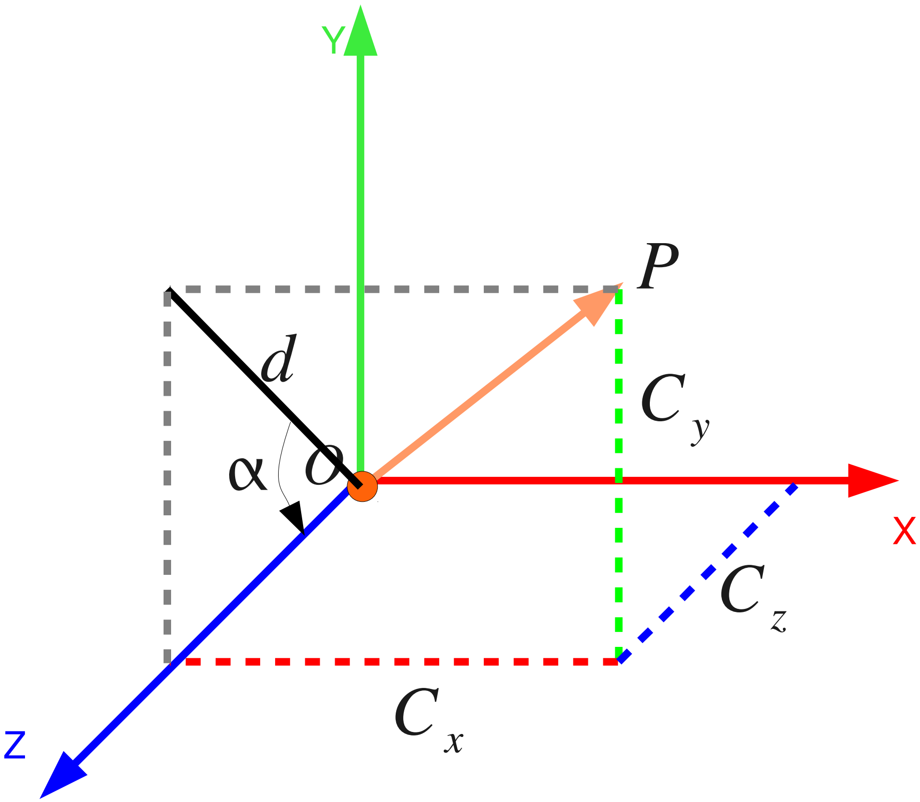
Consider the unit vector
in the direction C_x, C_y, C_z

3D Transformations

Rotating about an arbitrary axis.

Align the vector $\vec{OP}$ to the z axis

Rotate $\vec{OP}$ such that it lies on the XZ plane
i.e., rotate about the X axis by α

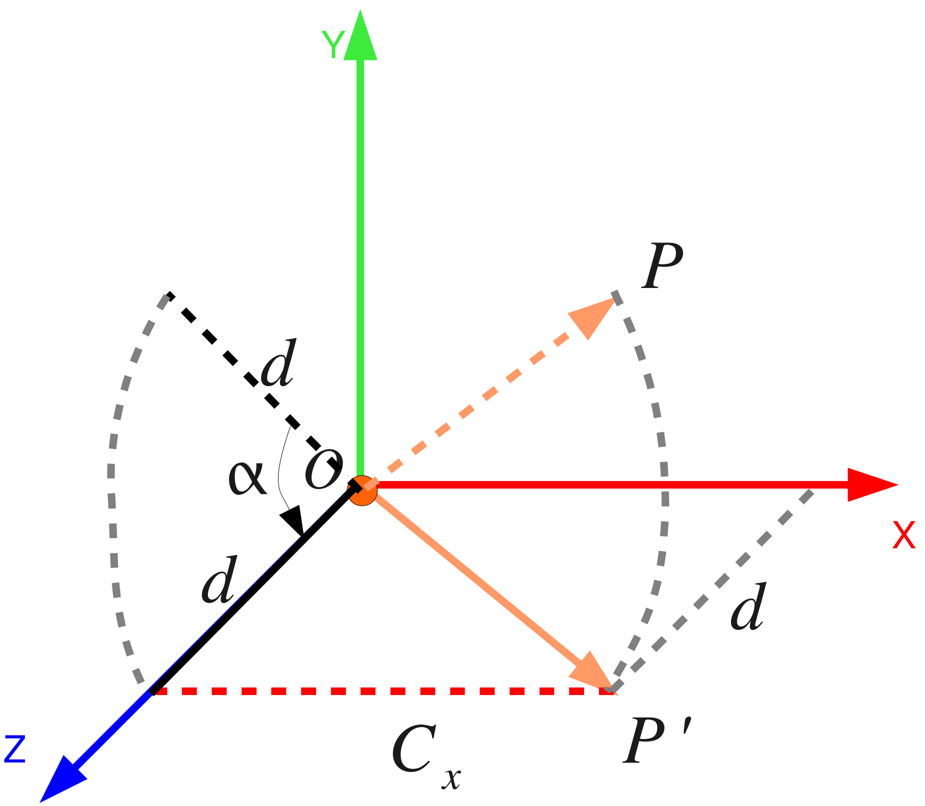


$$d = \sqrt{C_y^2 + C_z^2} \qquad \cos\alpha = \frac{C_z}{d} \qquad \sin\alpha = \frac{C_y}{d} \qquad R_x(\alpha)$$

3D Transformations

Rotating about
an arbitrary axis.

Align the vector $\vec{OP}$
to the z axis

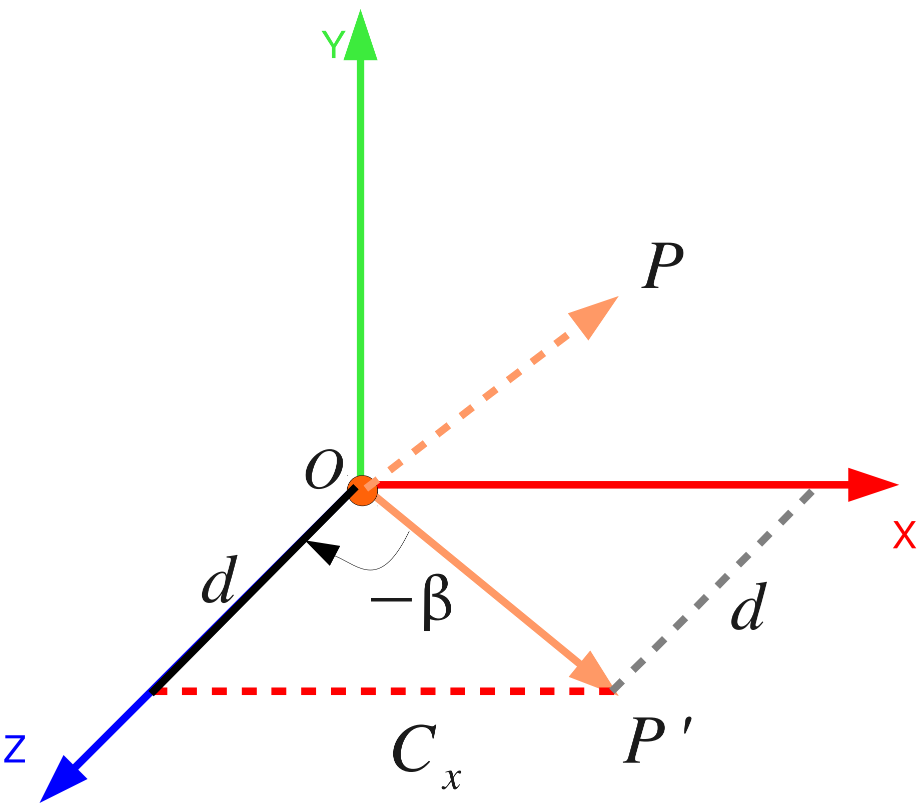


Rotate $\vec{OP}$ such that
it lies on the XZ plane
i.e., rotate about the X axis by α

$$d = \sqrt{C_y^2 + C_z^2} \qquad \cos\alpha = \frac{C_z}{d} \qquad \sin\alpha = \frac{C_y}{d} \qquad R_x(\alpha)$$

3D Transformations

Rotating about an arbitrary axis.



Align the vector $\vec{OP}$ to the z axis

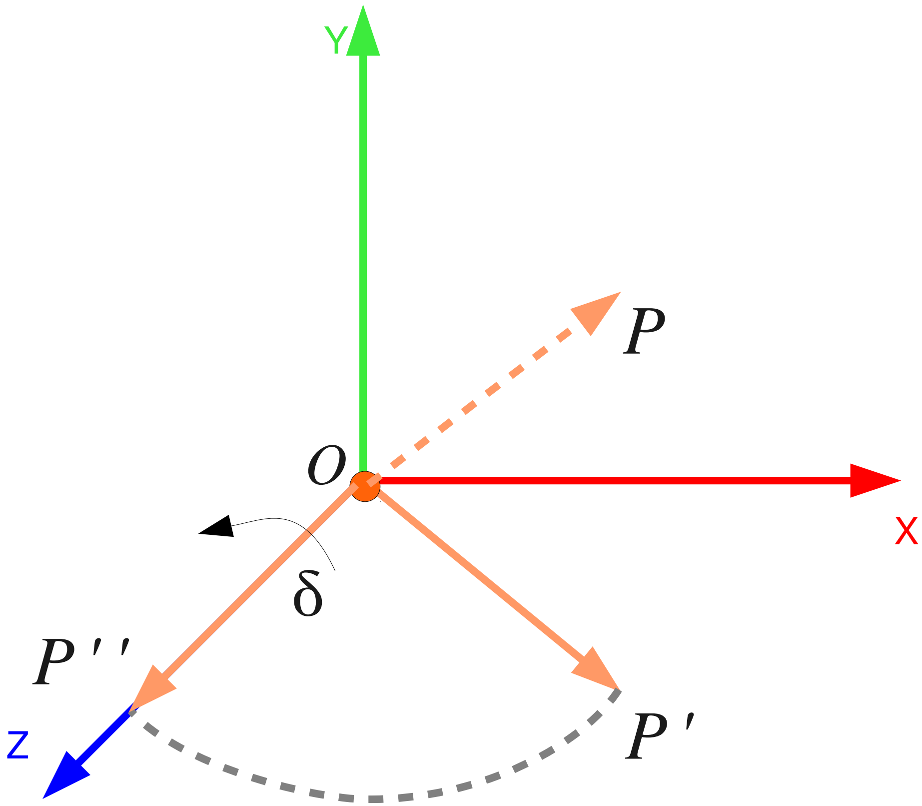
Rotate $\vec{OP}$ around Y axis by $-\beta$

$$d = \sqrt{C_y^2 + C_z^2} \qquad \cos(-\beta) = d \qquad \sin(-\beta) = C_x \qquad R_y(-\beta)$$

3D Transformations

Rotating about
an arbitrary axis.

Now rotate about the
to the z axis by δ



$$R_z(delta)$$

3D Transformations

Rotating about
an arbitrary axis.

Now do the inverse transforms in reverse order to get the composite transformation matrix as:

$$M = T(P_o) . R_x(-\alpha) . R_y(\beta) . R_z(\delta) . R_y(-\beta) . R_x(\alpha) . T(-P_o)$$

3D Transformations

General 3D transformation:

$$T = \left[\begin{array}{ccc|c} a & d & g & l \\ b & e & h & m \\ c & f & i & n \\ \hline p & q & r & s \end{array}\right]$$

Transformations in OpenGL

- When we specify a matrix transformation it is **post multiplied** with the current transformation matrix.

glLoadIdentity() — Current matrix $C_1 = I$

glRotatef(Ө, x, y, z) — Current matrix $C_2 = C_1 . R$

glTranslatef(tx, ty, tz) — Current matrix $C_3 = C_2 . T$

```
glBegin(...)
    glVertex3f(45.6, 34.9, .....)
    glVertex3f(45.6, 34.9, .....)
    :          :
    :          :
glEnd()
```

$$v_{transformed} = C_3 . v$$
$$= R . T . v$$

Parag Chaudhuri

Transformations in OpenGL

- OpenGL maintains a stack of matrices with the current matrix always on top of the stack.
- glPushMatrix copies the matrix on top of the stack and pushes it into the stack.
- glPopMatrix pops the matrix on top of the stack.

glLoadIdentity()
glRotatef(θ_A, 0, 0, 1)
drawShoulder(...)
glPushMatrix()
    glRotatef(θ_B, 0, 0, 1)
    drawElbow(...)
glPopMatrix()