CMSC 435
Introductory Computer Graphics

Penny Rheingans

UMBC

Translation

- Offset from position

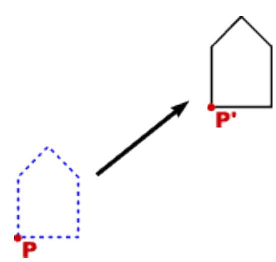
  $x' = x + t_x$

  $y' = y + t_y$

- In matrix form

  $P' = P + T$

  where $T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$



- Repeat transformation for each point in figure (rigid body transformation)

Scaling

- Resize relative to position

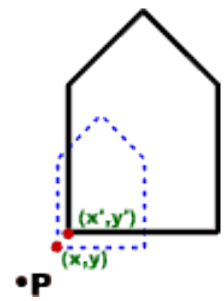
  $x' = x * s_x$

  $y' = y * s_y$

- In matrix form

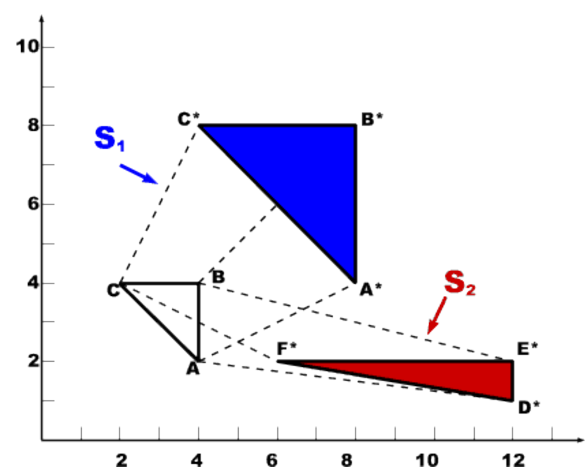
  $P' = P * S$

  where T = $\begin{bmatrix} s_x \\ s_y \end{bmatrix}$



- Repeat transformation for each point in figure (rigid body transformation)

Non-uniform Scaling

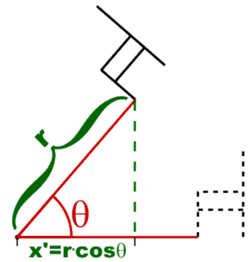


Rotation

- Turn about a point

  x' = x cosθ - y sinθ

  y' = x sinθ + y cosθ



- In matrix form

  P' = R•P

where R = $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

Homogenous Coordinates

- Want to do all standard transformations as matrix multiplications

$$P = \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x/h \\ y/h \\ 1 \end{bmatrix} or \begin{bmatrix} x \\ y \\ h \end{bmatrix}$$

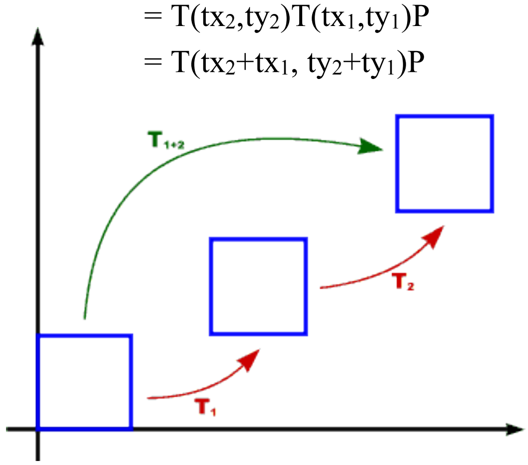
Composite Transformations

- Translation

  $P' = T_2 T_1 P$

  $= T(tx_2, ty_2) T(tx_1, ty_1) P$

  $= T(tx_2+tx_1,\ ty_2+ty_1) P$

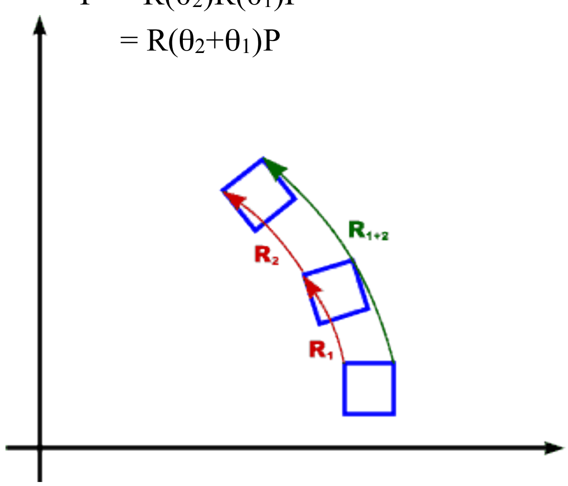


Composite Transformations

- Rotation

  $P' = R(\theta_2) R(\theta_1) P$

  $= R(\theta_2+\theta_1) P$



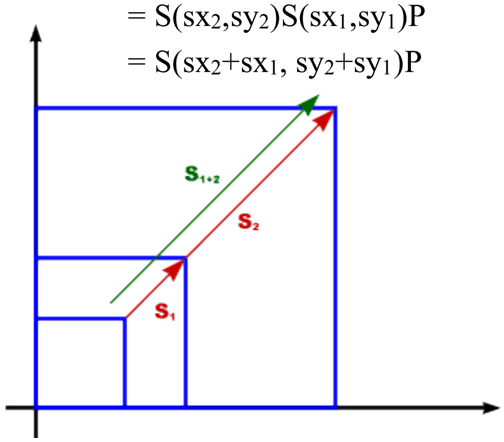
Composite Transformations

- Scaling

  $P' = S_2 S_1 P$

  $= S(sx_2, sy_2) S(sx_1, sy_1) P$

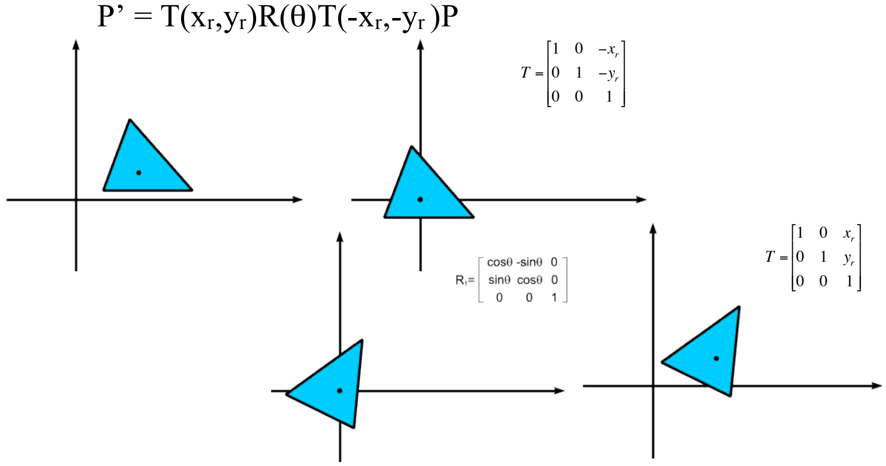
  $= S(sx_2 + sx_1,\ sy_2 + sy_1) P$



General Pivot Point Rotation

- A transformation sequence for rotating an object about a specified pivot point

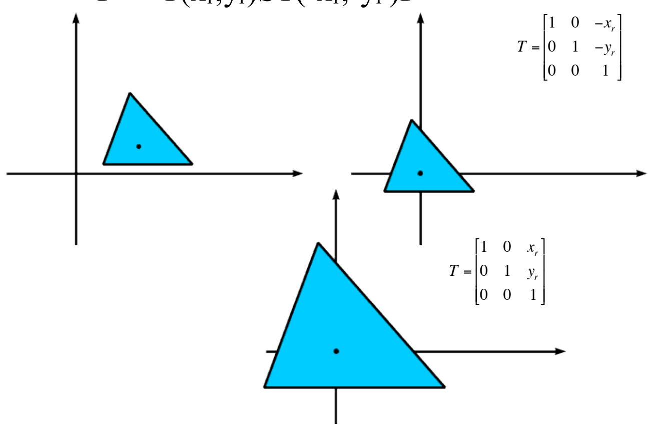
  $P' = T(x_r, y_r) R(\theta) T(-x_r, -y_r) P$



General Pivot Point Scaling

- A transformation sequence for scaling an object about a specified pivot point

  $P' = T(x_r, y_r) S T(-x_r, -y_r) P$



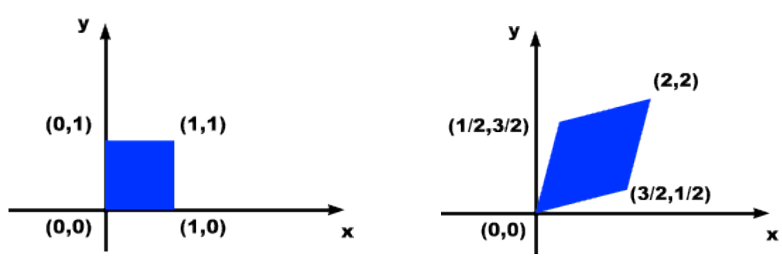
$$T = \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix}$$

General Scaling Direction

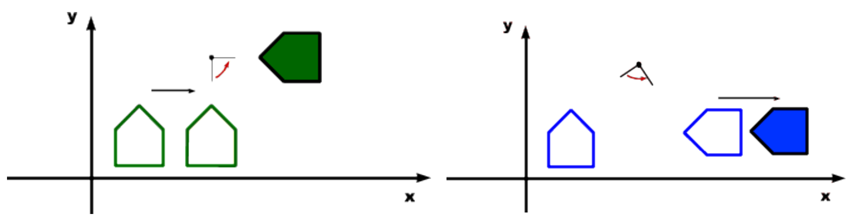
- A transformation sequence for scaling an object along a specified direction
  - Rotate scaling direction to axis
  - Scale
  - Rotate back

  $P' = R(\theta) S R(-\theta) P$

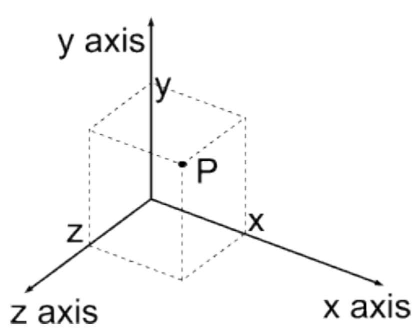


Order of Transformations

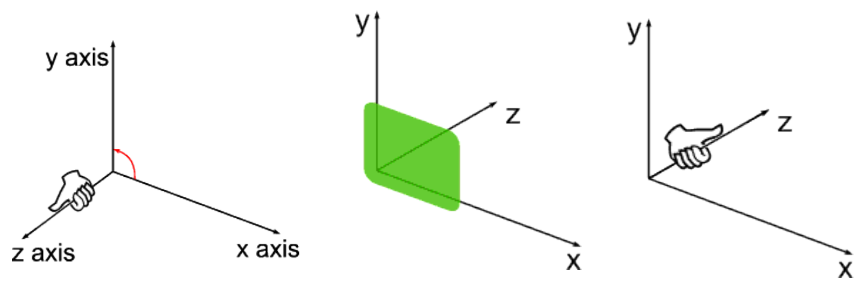
- RTP != TRP



3D Cartesian Coordinate Systems



Handedness

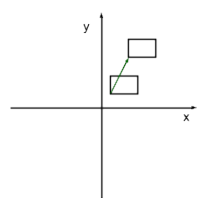


Translation

- 2D Translation

  $x' = x + t_x$

  $y' = y + t_y$



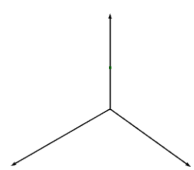
$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

- 3D Translation

  $x' = x + t_x$

  $y' = y + t_y$

  $z' = z + t_z$



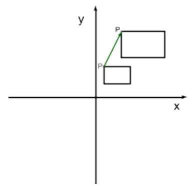
$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling

- 2D Translation

  $x' = x * s_x$

  $y' = y * s_y$



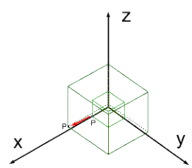
$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- 3D Translation

  $x' = x * s_x$

  $y' = y * s_y$

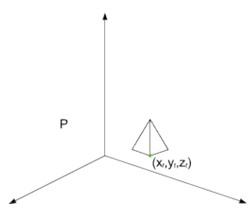
  $z' = z * s_z$

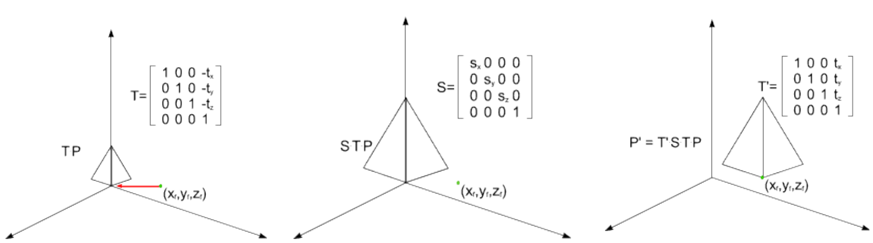


$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

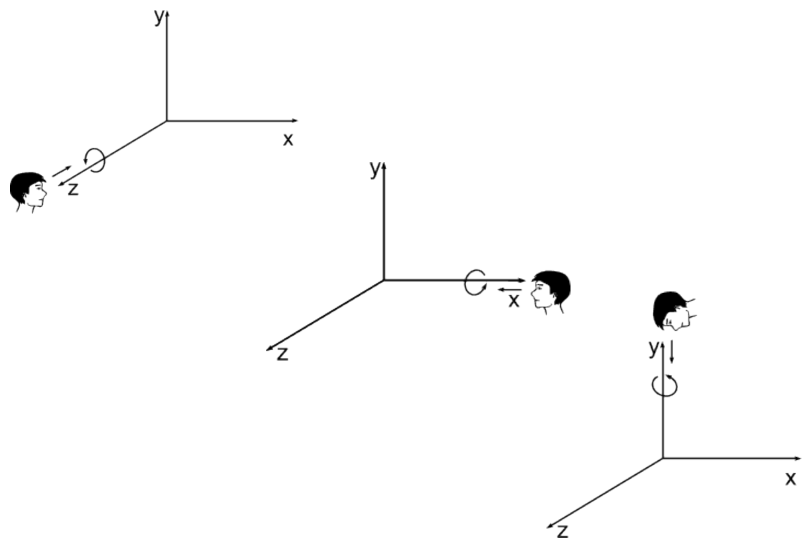
Scale with Fixed Point

- Translate fixed point to origin
- Scale
- Translate back

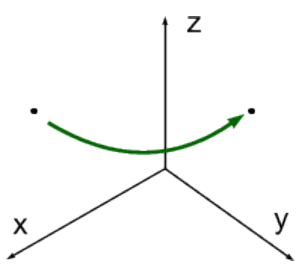




Positive 3D Rotation



Rotation about Z axis



- Like 2D

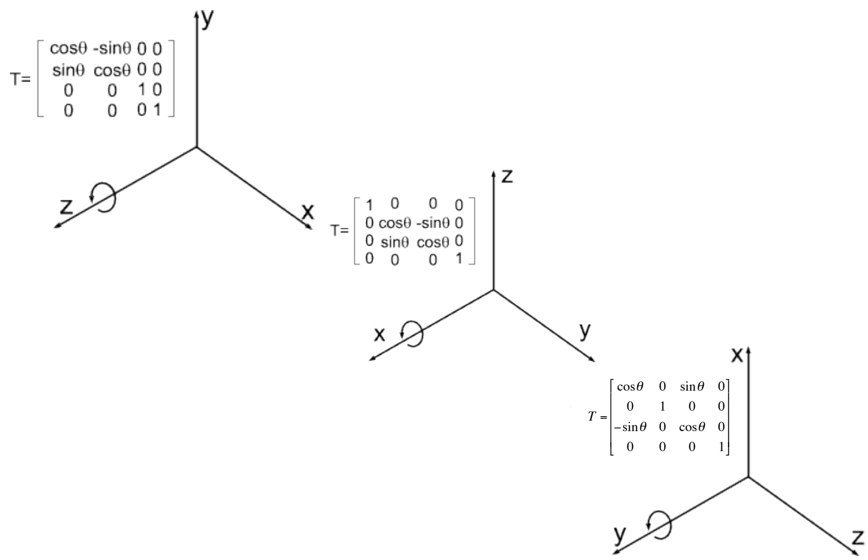
  $x' = x \cos\theta - y \sin\theta$

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

  $z' = z$

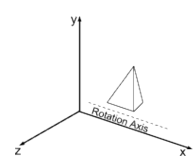
$$T = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

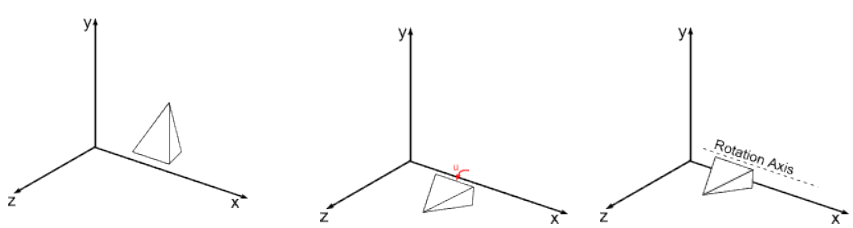
Rotation about Coordinate Axes



Rotation about Axis Parallel to X

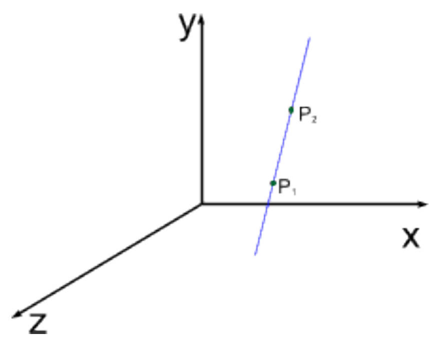
- Translate rotation axis to X
- Rotate
- Translate back





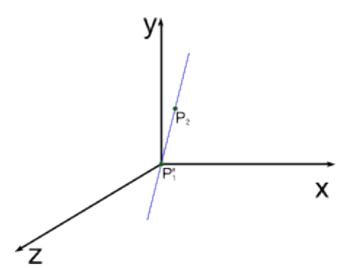
Rotation about Arbitrary Axis

- To rotate about axis through P_1P_2

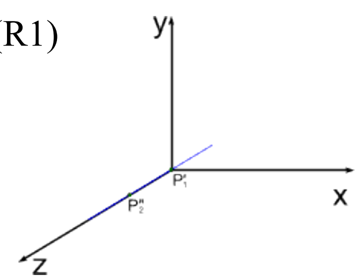


Rotation about Arbitrary Axis (2)

- Translate P_1 to origin (T)

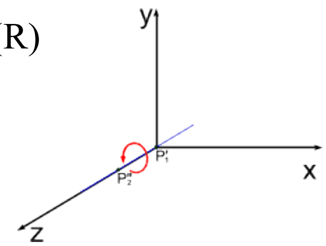


- Rotate until P_2 lies on z axis (R1)

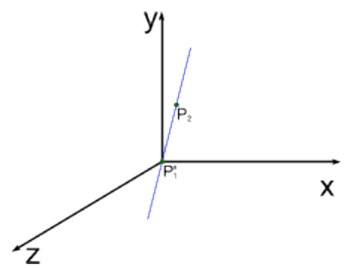


Rotation about Arbitrary Axis (3)

- Perform desired rotation (R)

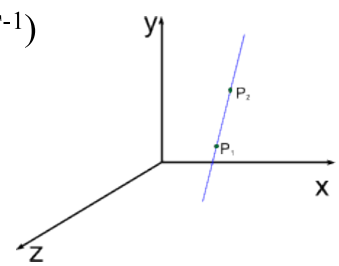


- Rotate axis back (R_1^{-1})



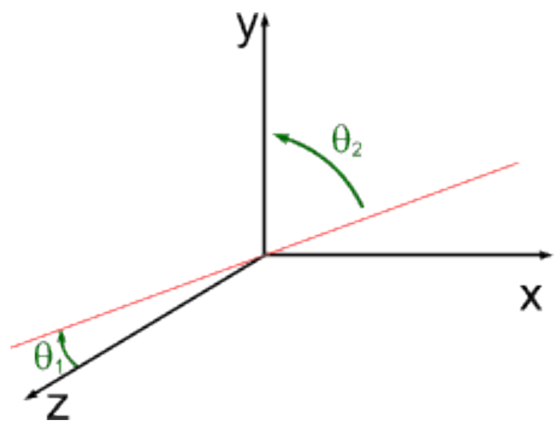
Rotation about Arbitrary Axis (4)

- Translate axis back (T^{-1})



Rotation Axis Specification

1. Point and two rotation angles



Rotation Axis Specification

2. Two points

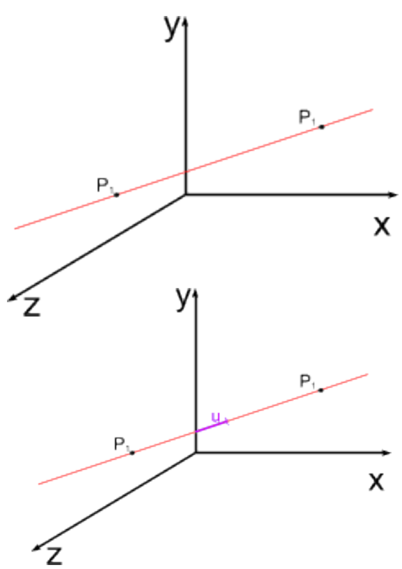
$V = P_2 - P_1$

$U = V / |V| = (a,b,c)$

$a = (x_2 - x_1) / |V|$

$b = (y_2 - y_1) / |V|$

$c = (z_2 - z_1) / |V|$



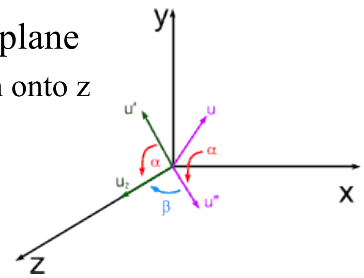
Rotating Axis onto Coordinate Axis

1. Rotate about x-axis into xz plane

Same as rotation of yz projection onto z

u’ = (0,b,c)

$u_2 = (0,0,d)$



Rotating Axis onto Coordinate Axis

1. Rotate about x-axis into xz plane

Same as rotation of yz projection onto z

u’ = (0,b,c)

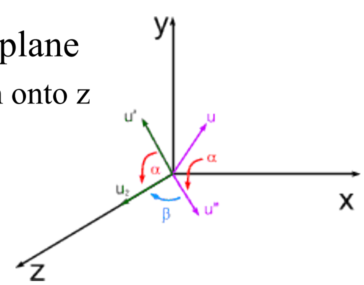
$u_2 = (0,0,d)$

$\cos \alpha = (u' \bullet u_2)/(|u'||u_2|$

$= cd/dd$

$= c/d$

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



Rotating Axis onto Coordinate Axis

1. Rotate about x-axis into xz plane

Same as rotation of yz projection onto z

$u' = (0,b,c)$

$u_2 = (0,0,d)$

$\cos\alpha = (u' \bullet u_2)/(|u'||u_2|$

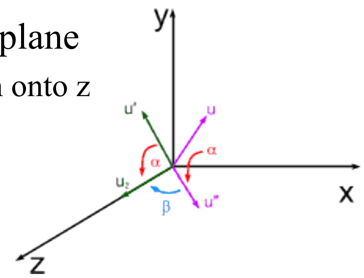
$= cd/dd$

$= c/d$

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

$\sin\alpha = b/d$

$u'' = (a,0,d)$



$$R_x(x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotating Axis onto Coordinate Axis

1. Rotate about x-axis into xz plane
2. Rotate about y onto x axis

$$R_x(x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(x) = \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- composite rotation $R_1 = R_y(\beta)R_x(\alpha)$

Transforming Normals

- Transforming by same matrix as points doesn't necessarily work
- Can calculate correct matrix using relationship to tangent vector (which does transform correctly)
  - $N = (M^{-1})^T$

Coordinate Transformations

- Can move coordinate frame, rather than points
  - $\mathbf{p} + u\mathbf{u} + v\mathbf{v} + w\mathbf{w}$
  - origin **p**, basis vectors **u**,**v**,**w**
- Frame-to-canonical conversion

$$\mathbf{p}_{\mathbf{xy}z} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{p}_{\mathbf{uv}w}$$

$$\mathbf{p}_{uvw} = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \mathbf{p}_{xyz}$$