

Transformations



Oregon State
University



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](#)

Mike Bailey

mjb@cs.oregonstate.edu

$$\begin{array}{l}
 \text{x consuming column} \\
 \text{y consuming column} \\
 \text{z consuming column} \\
 \text{constant column}
 \end{array}
 \begin{array}{c}
 \downarrow \\
 \left[\begin{array}{cccc} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ 0 & 0 & 0 & 1 \end{array} \right] \cdot \left[\begin{array}{c} x \\ y \\ z \\ 1 \end{array} \right]
 \end{array}$$

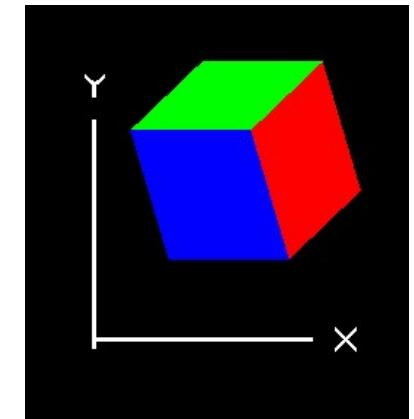
x' producing row
 y' producing row
 z' producing row



Oregon State
University

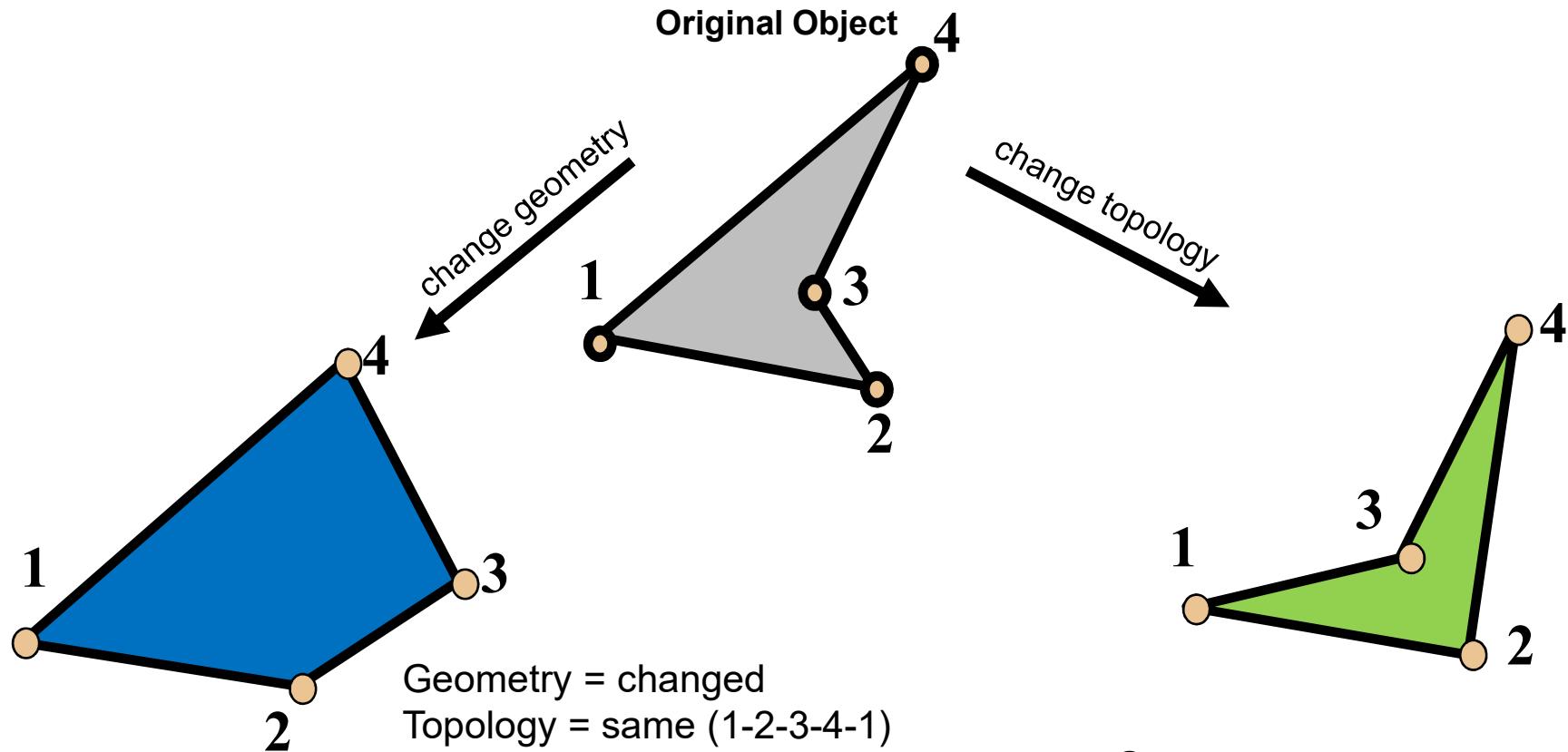
Computer Graphics

transformations.pptx



mjb – August 25, 2022

Geometry vs. Topology



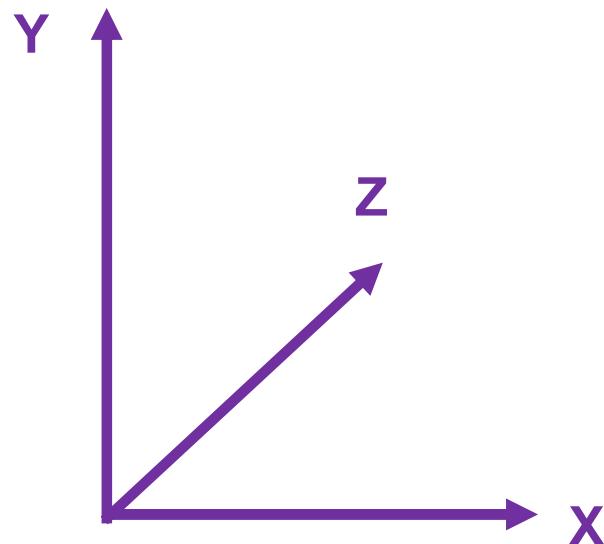
Geometry:

Where things are (e.g., coordinates)

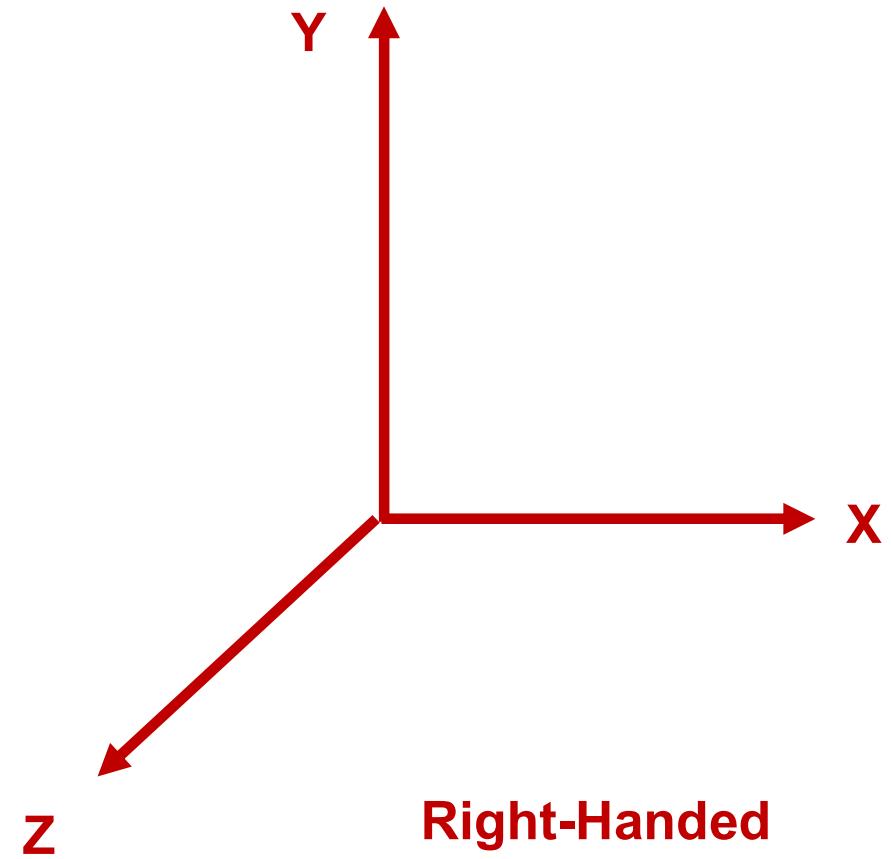
Topology:

How things are connected

3D Coordinate Systems



Left-Handed

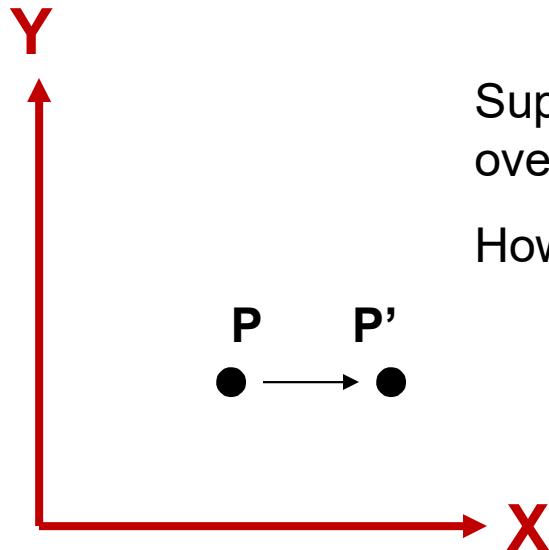


Right-Handed



Oregon State
University
Computer Graphics

Transformations



Suppose you have a point P and you want to move it over by 2 units in X.

How would you change P's coordinates?

$$P' = (P'_x, P'_y) = (P_x + 2., P_y)$$

This is known as a coordinate *transformation*

General Form of 3D Linear Transformations

$$x' = Ax + By + Cz + D$$

$$y' = Ex + Fy + Gz + H$$

$$z' = Ix + Jy + Kz + L$$

It's called a "Linear Transformation" because all of the coordinates are raised to the 1st power, that is, there are no x^2 , x^3 , etc. terms.

Transform the geometry – leave the topology as is



Oregon State
University

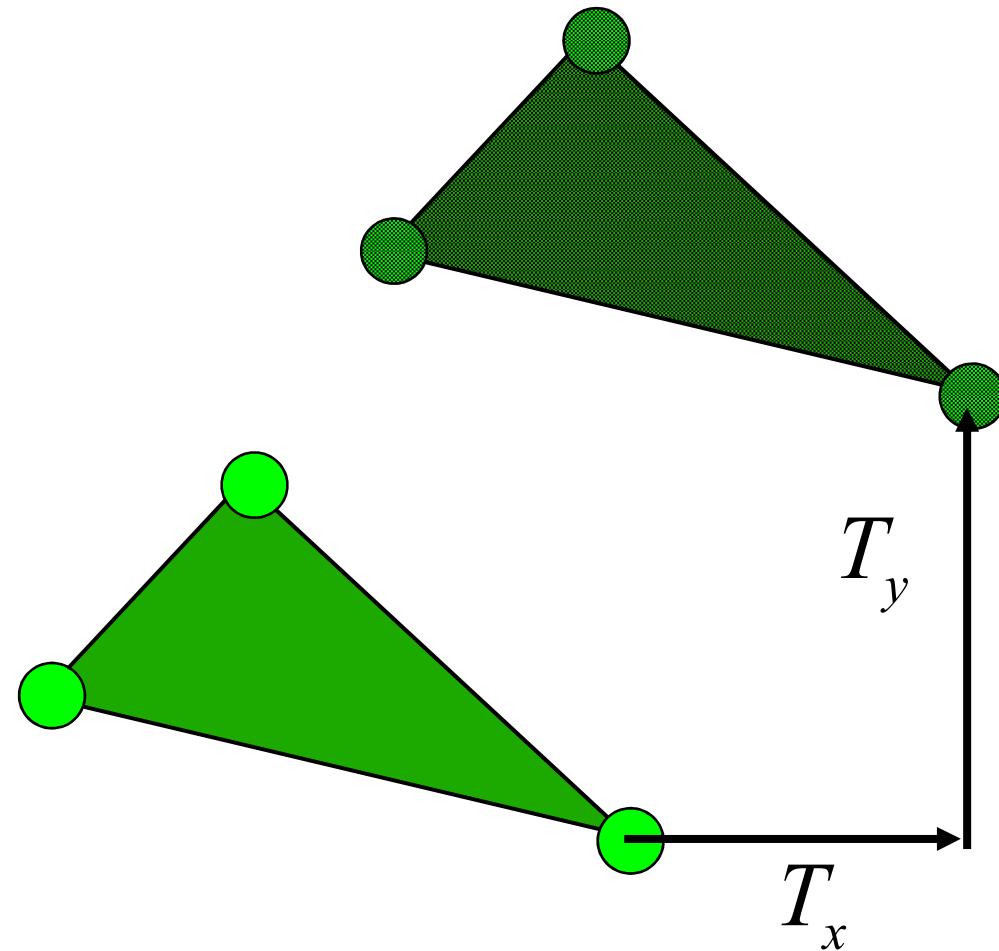
Computer Graphics

Translation Equations

$$x' = x + T_x$$

$$y' = y + T_y$$

$$z' = z + T_z$$



Oregon State

University

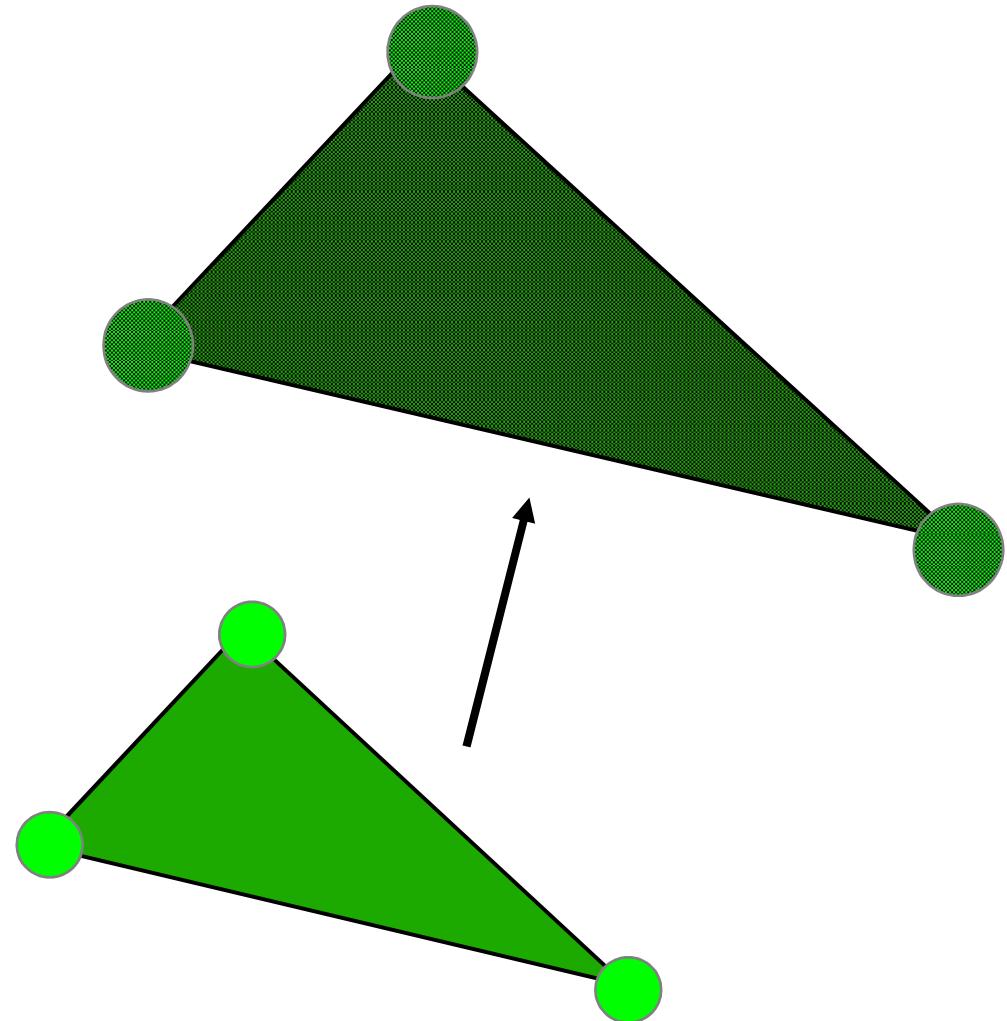
Computer Graphics

Scaling About the Origin

$$x' = x \cdot S_x$$

$$y' = y \cdot S_y$$

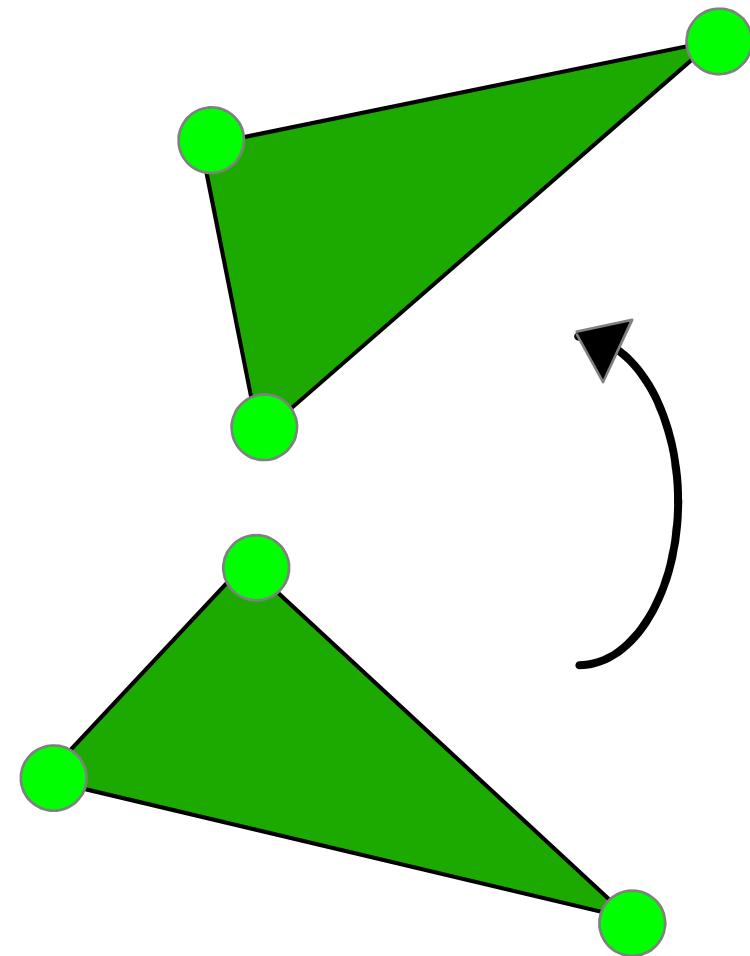
$$z' = z \cdot S_z$$



2D Rotation About the Origin

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

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



Linear Equations in Matrix Form

$$x' = Ax + By + Cz + D$$

$$y' = Ex + Fy + Gz + H$$

$$z' = Ix + Jy + Kz + L$$

x' producing row

y' producing row

z' producing row

x consuming column

y consuming column

z consuming column

constant column

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Oregon State
University

Computer Graphics

Identity Matrix ([I])

$$\begin{aligned}x' &= Ax + By + Cz + D \\y' &= Ex + Fy + Gz + H \\z' &= Ix + Jy + Kz + L\end{aligned}$$

x consuming column
 y consuming column
 z consuming column
 constant column

$$\begin{array}{l} \xrightarrow{\text{x' producing row}} \left\{ \begin{matrix} x' \\ y' \\ z' \\ 1 \end{matrix} \right\} = \left[\begin{matrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ 0 & 0 & 0 & 1 \end{matrix} \right] \cdot \left\{ \begin{matrix} x \\ y \\ z \\ 1 \end{matrix} \right\} \\ \xrightarrow{\text{y' producing row}} \\ \xrightarrow{\text{z' producing row}} \end{array}$$

$$x' = x$$

$$y' = y$$

$$z' = z$$

$$\left\{ \begin{matrix} x' \\ y' \\ z' \\ 1 \end{matrix} \right\} = \left[\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \right] \cdot \left\{ \begin{matrix} x \\ y \\ z \\ 1 \end{matrix} \right\}$$



Oregon State
University

Computer Graphics

[I] signifies that “Nothing has changed”

Matrix Inverse

$$[M] \bullet [M]^{-1} = [I]$$

$$[M] \bullet [M]^{-1} = \text{“Nothing has changed”}$$

“Whatever $[M]$ does, $[M]^{-1}$ undoes”



Oregon State

University

Computer Graphics

Translation Matrix

$$x' = Ax + By + Cz + D$$

$$y' = Ex + Fy + Gz + H$$

$$z' = Ix + Jy + Kz + L$$

$$\begin{array}{c} \text{x consuming column} \\ \text{y consuming column} \\ \text{z consuming column} \\ \text{constant column} \end{array} \quad \begin{array}{l} \text{x' producing row} \\ \text{y' producing row} \\ \text{z' producing row} \end{array}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Oregon State

University

Computer Graphics

Quick! What is the inverse of this matrix?

Scaling Matrix

$$x' = Ax + By + Cz + D$$

$$y' = Ex + Fy + Gz + H$$

$$z' = Ix + Jy + Kz + L$$

$$\begin{array}{l} \text{x consuming column} \\ \text{y consuming column} \\ \text{z consuming column} \\ \text{constant column} \end{array} \quad \begin{array}{c} \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \end{array} \quad \begin{array}{l} \text{x' producing row} \\ \text{y' producing row} \\ \text{z' producing row} \\ \text{1} \end{array} \quad \left[\begin{array}{cccc} x' & A & B & C & D \\ y' & E & F & G & H \\ z' & I & J & K & L \\ 1 & 0 & 0 & 0 & 1 \end{array} \right] \cdot \left\{ \begin{array}{l} x \\ y \\ z \\ 1 \end{array} \right\}$$

$$\left\{ \begin{array}{l} x' \\ y' \\ z' \\ 1 \end{array} \right\} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \left\{ \begin{array}{l} x \\ y \\ z \\ 1 \end{array} \right\}$$

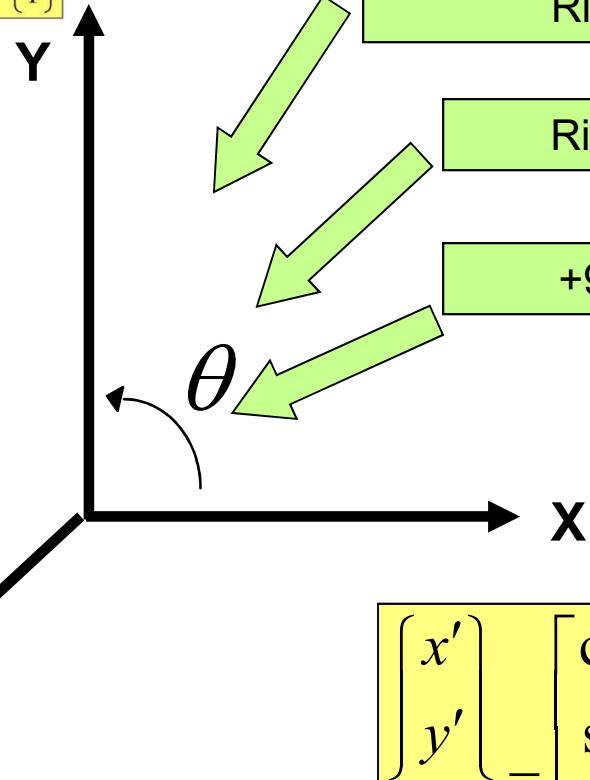


3D Rotation Matrix About Z

$$\begin{aligned}x' &= Ax + By + Cz + D \\y' &= Ex + Fy + Gz + H \\z' &= Ix + Jy + Kz + L\end{aligned}$$

x' y' z' 1	$\left[\begin{array}{c} x' \\ y' \\ z' \\ 1 \end{array} \right] = \left[\begin{array}{cccc} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ 0 & 0 & 0 & 1 \end{array} \right] \cdot \left[\begin{array}{c} x \\ y \\ z \\ 1 \end{array} \right]$
---------------------------	--

x consuming column
y consuming column
z consuming column
constant column



Right-handed coordinates

Right-handed positive rotation rule

+90° rotation gives: $y' = x$

$$\left[\begin{array}{c} x' \\ y' \\ z' \\ 1 \end{array} \right] = \left[\begin{array}{cccc} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \cdot \left[\begin{array}{c} x \\ y \\ z \\ 1 \end{array} \right]$$



Oregon State
University

Computer Graphics

3D Rotation Matrix About Y

$$x' = Ax + By + Cz + D$$

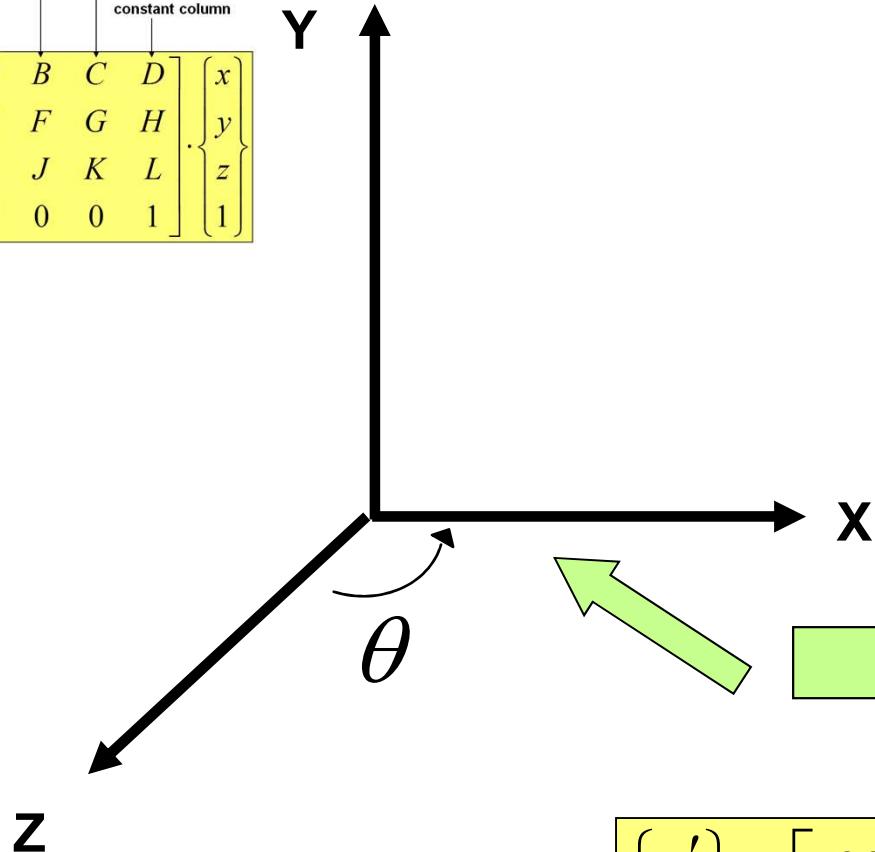
$$y' = Ex + Fy + Gz + H$$

$$z' = Ix + Jy + Kz + L$$

x consuming column
 y consuming column
 z consuming column
 constant column

$$\begin{matrix}
 \begin{array}{c} x' \\ y' \\ z' \\ 1 \end{array} & = & \begin{bmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{c} x \\ y \\ z \\ 1 \end{array}
 \end{matrix}$$

x' producing row
 y' producing row
 z' producing row



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



Oregon State
University

Computer Graphics

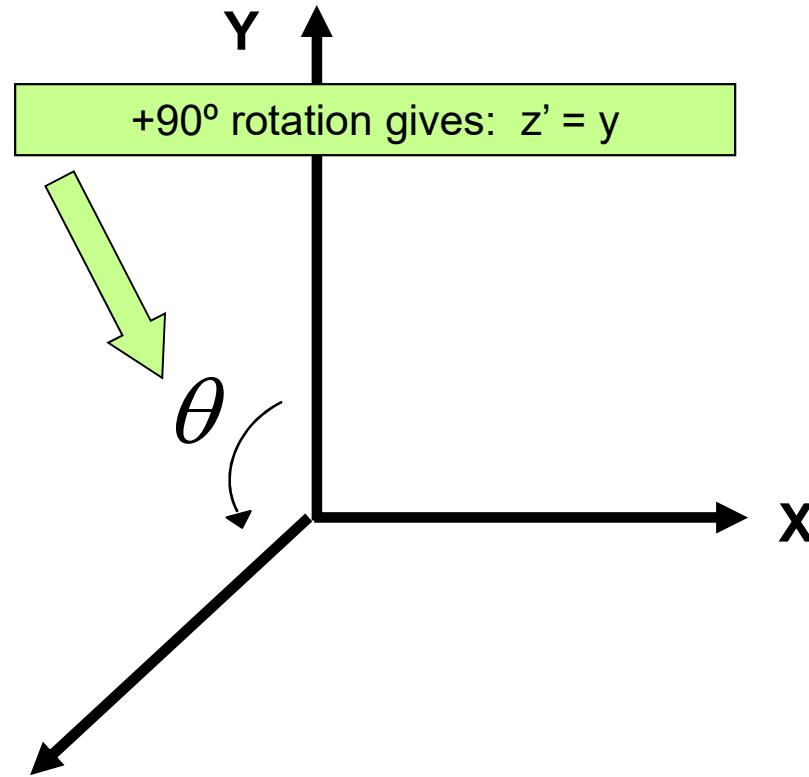
3D Rotation Matrix About X

$$\begin{aligned}x' &= Ax + By + Cz + D \\y' &= Ex + Fy + Gz + H \\z' &= Ix + Jy + Kz + L\end{aligned}$$

x consuming column
y consuming column
z consuming column
constant column

$$\begin{array}{l} \begin{array}{l} x' \\ y' \\ z' \\ 1 \end{array} = \begin{array}{c|c|c|c} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ 0 & 0 & 0 & 1 \end{array} \cdot \begin{array}{l} x \\ y \\ z \\ 1 \end{array} \end{array}$$

x' producing row
y' producing row
z' producing row



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



Oregon State
University

Computer Graphics

How it Really Works :-)

$$\begin{bmatrix} \cos 90^\circ & \sin 90^\circ \\ -\sin 90^\circ & \cos 90^\circ \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

<http://xkcd.com>

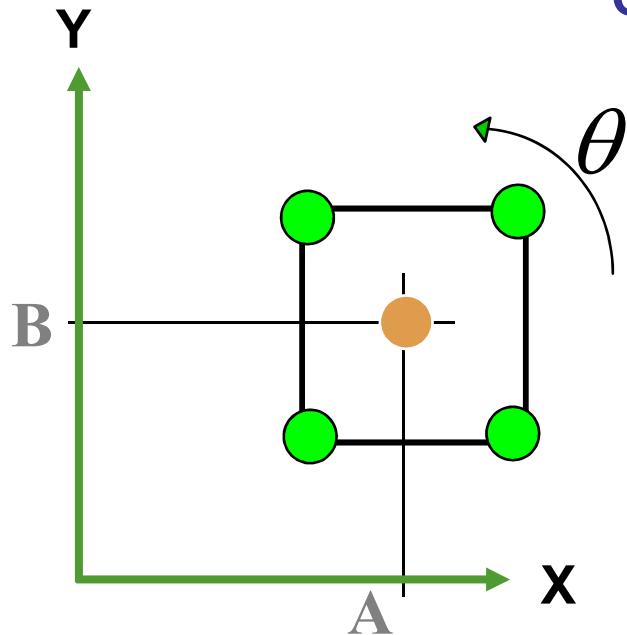


Oregon State

University

Computer Graphics

Compound Transformations



Q: Our rotation matrices only work around the origin. What if we want to rotate about an arbitrary point (A,B)?

A: Use more than one matrix.

Write it

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \left[T_{+A,+B} \right] \cdot \left[R_\theta \right] \cdot \left[T_{-A,-B} \right] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

3 2 1

Say it

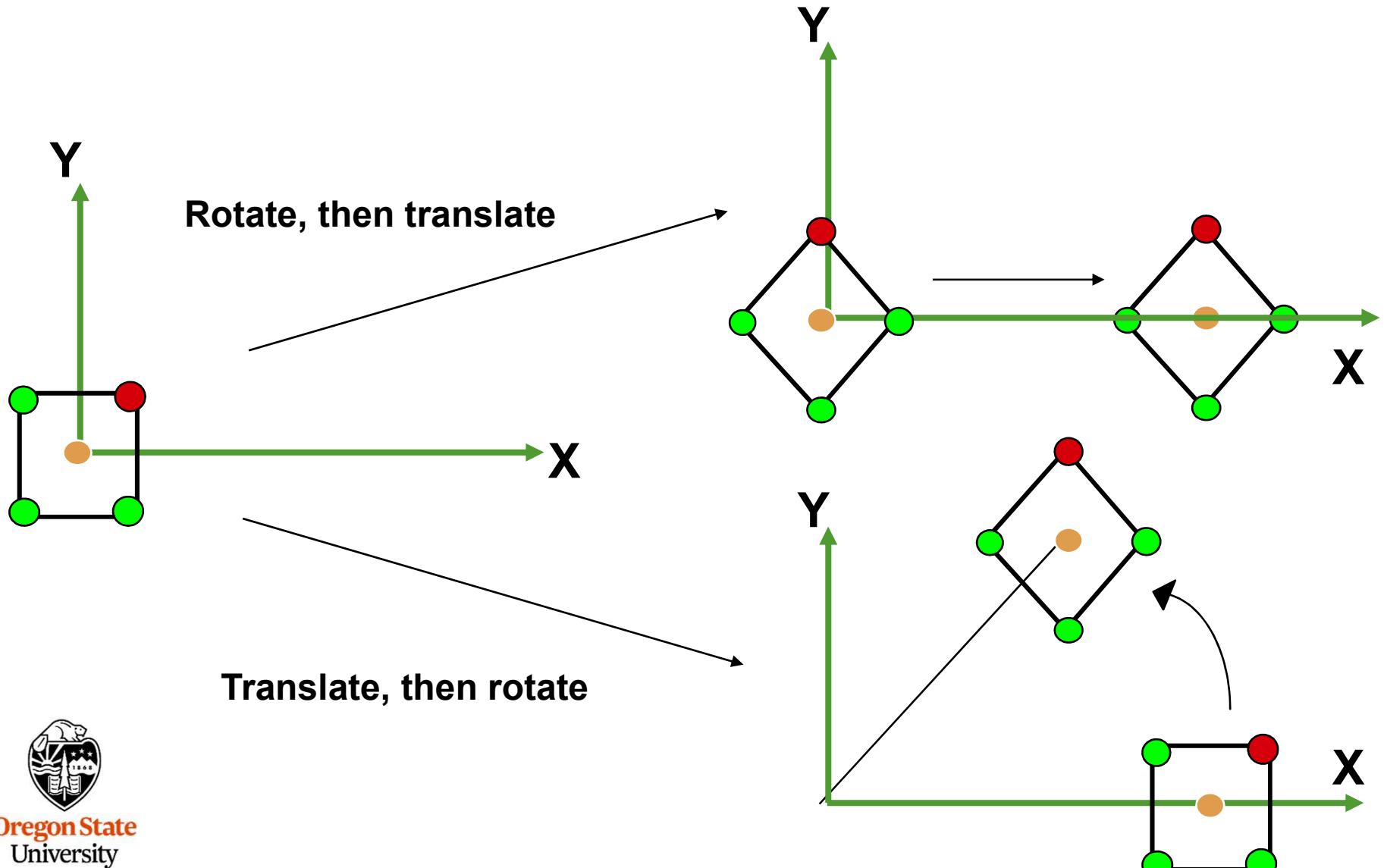


Oregon State

University

Computer Graphics

Matrix Multiplication *is not* Commutative



Matrix Multiplication *is* Associative

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \left[T_{+A,+B} \right] \cdot \left[R_\theta \right] \cdot \left[T_{-A,-B} \right] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \underbrace{\left(\left[T_{+A,+B} \right] \cdot \left[R_\theta \right] \cdot \left[T_{-A,-B} \right] \right)}_{\text{One matrix --}} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

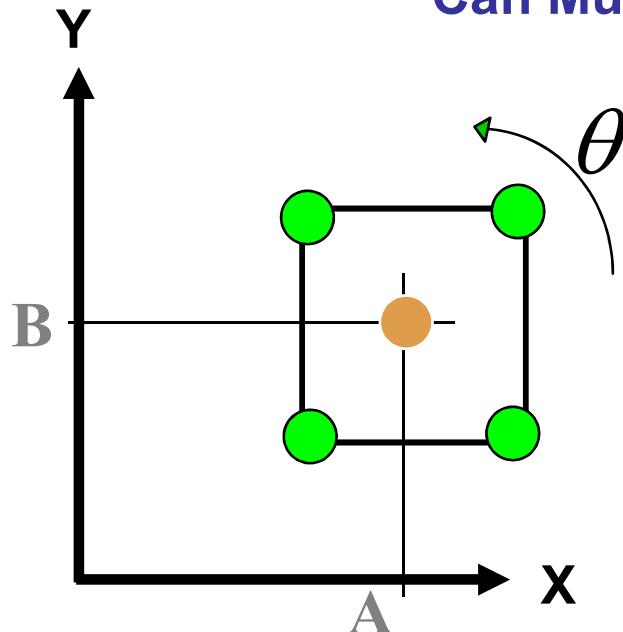
the Current Transformation Matrix, or CTM



Oregon State
University

Computer Graphics

Can Multiply All Geometry by One Matrix !



$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = [M] \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



Oregon State
University

Computer Graphics

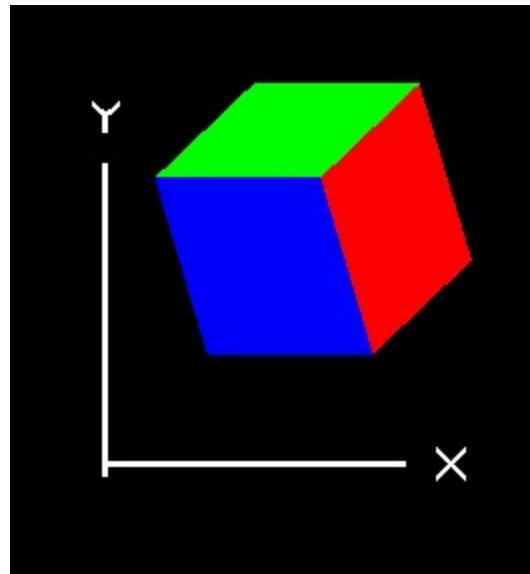
Graphics hardware can do this very quickly!

OpenGL Will Do the Transformation Compounding for You !

22

```
glTranslatef( A, B, C );  
  
glRotatef( (GLfloat)Yrot, 0., 1., 0. );  
glRotatef( (GLfloat)Xrot, 1., 0., 0. );  
  
glScalef( (GLfloat)Scale, (GLfloat)Scale, (GLfloat)Scale );  
  
glCallList( BoxList );
```

Typically objects are modeled around their own local origin, so the
glTranslate(-A, -B, -C)
step is unnecessary.



```
glTranslatef( A, B, C );  
glRotatef( (GLfloat)Yrot, 0., 1., 0. );  
glRotatef( (GLfloat)Xrot, 1., 0., 0. );  
glScalef( (GLfloat)Scale, (GLfloat)Scale, (GLfloat)Scale );  
glCallList( BoxList );
```

OpenGL Will Do the Transformation Compounding for You !

```
for( ; ; )
{
    << Turn mouse position into Xrot and Yrot >>

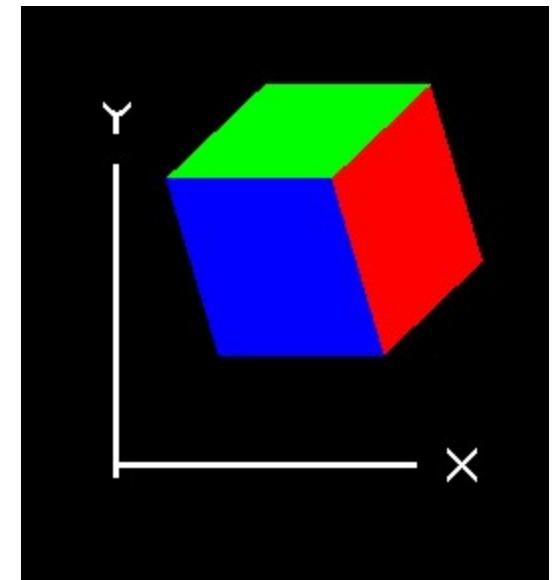
    glLoadIdentity();

    glTranslatef( A, B, C );

    glRotatef( (GLfloat)Yrot, 0., 1., 0. );
    glRotatef( (GLfloat)Xrot, 1., 0., 0. );

    glScalef( (GLfloat)Scale, (GLfloat)Scale, (GLfloat)Scale );

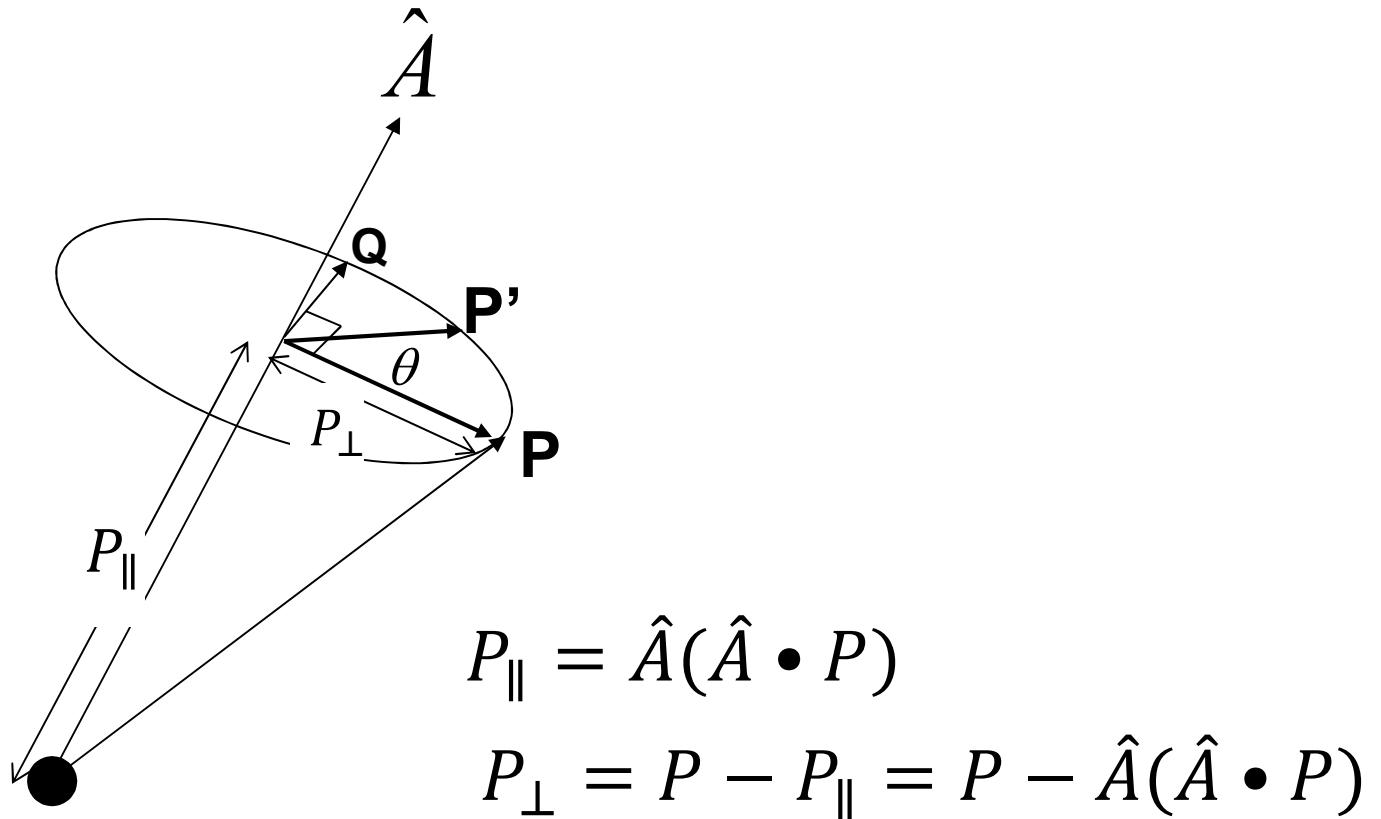
    glCallList( BoxList );
}
```



Oregon State
University

Computer Graphics

The Funky Rotation Matrix for an Arbitrary Axis and Angle



$$Q = \hat{A} \times P_{\perp} = \hat{A} \times P_{\perp} + 0 = \hat{A} \times P_{\perp} + \hat{A} \times P_{\parallel} = \hat{A} \times (P_{\perp} + P_{\parallel}) = \hat{A} \times P$$



Oregon State

University

Computer Graphics

The Funky Rotation Matrix for an Arbitrary Axis and Angle

$$\begin{aligned} P_{\parallel}' &= P_{\parallel} \\ P_{\perp}' &= P_{\perp} \cos \theta + Q \sin \theta \\ P' &= P'_{\parallel} + P'_{\perp} \\ P' &= [\hat{A}(\hat{A} \bullet P)] + \cos \theta [P - \hat{A}(\hat{A} \bullet P)] + \sin \theta [\hat{A} \times P] \end{aligned}$$



Oregon State

University

Computer Graphics

The Funky Rotation Matrix for an Arbitrary Axis and Angle

$$\hat{A}(\hat{A} \bullet P) = \begin{bmatrix} A_x A_x & A_x A_y & A_x A_z \\ A_y A_x & A_y A_y & A_y A_z \\ A_z A_x & A_z A_y & A_z A_z \end{bmatrix} \{P\}$$

$$\hat{A} \times P = \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} \{P\}$$



Oregon State
University

Computer Graphics

The Funky Rotation Matrix for an Arbitrary Axis and Angle

$$P' = [\hat{A}(\hat{A} \bullet P)] + \cos \theta [P - \hat{A}(\hat{A} \bullet P)] + \sin \theta [\hat{A} \times P]$$

$$P' = \left(\begin{bmatrix} A_x A_x & A_x A_y & A_x A_z \\ A_y A_x & A_y A_y & A_y A_z \\ A_z A_x & A_z A_y & A_z A_z \end{bmatrix} + \cos \theta \left[[I] - \begin{bmatrix} A_x A_x & A_x A_y & A_x A_z \\ A_y A_x & A_y A_y & A_y A_z \\ A_z A_x & A_z A_y & A_z A_z \end{bmatrix} \right] + \sin \theta \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} \right) \{P\}$$

$$[M] = \left(\begin{bmatrix} A_x A_x & A_x A_y & A_x A_z \\ A_y A_x & A_y A_y & A_y A_z \\ A_z A_x & A_z A_y & A_z A_z \end{bmatrix} + \cos \theta \begin{bmatrix} 1 - A_x A_x & -A_x A_y & -A_x A_z \\ -A_y A_x & 1 - A_y A_y & -A_y A_z \\ -A_z A_x & -A_z A_y & 1 - A_z A_z \end{bmatrix} + \sin \theta \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} \right)$$

The Funky Rotation Matrix for an Arbitrary Axis and Angle

$$[M] = \begin{pmatrix} A_x A_x & A_x A_y & A_x A_z \\ A_y A_x & A_y A_y & A_y A_z \\ A_z A_x & A_z A_y & A_z A_z \end{pmatrix} + \cos \theta \begin{pmatrix} 1 - A_x A_x & -A_x A_y & -A_x A_z \\ -A_y A_x & 1 - A_y A_y & -A_y A_z \\ -A_z A_x & -A_z A_y & 1 - A_z A_z \end{pmatrix} + \sin \theta \begin{pmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{pmatrix}$$

$$[M] = \begin{bmatrix} A_x A_x + \cos \theta (1 - A_x A_x) & A_x A_y - \cos \theta (A_x A_y) - \sin \theta A_z & A_x A_z - \cos \theta (A_x A_z) + \sin \theta A_y \\ A_y A_x - \cos \theta (A_y A_x) + \sin \theta A_z & A_y A_y + \cos \theta (1 - A_y A_y) & A_y A_z - \cos \theta (A_y A_z) - \sin \theta A_x \\ A_z A_x - \cos \theta (A_z A_x) - \sin \theta A_y & A_z A_y - \cos \theta (A_z A_y) + \sin \theta A_x & A_z A_z + \cos \theta (1 - A_z A_z) \end{bmatrix}$$

For this to work, A must be a unit vector

