Geometric Transformations

Lecturer: Carol O'Sullivan
Professor of Visual Computing
Carol.OSullivan@cs.tcd.ie

Course www: http://isg.cs.tcd.ie/cosulliv/

Geometric Transformations

- Many geometric transformations are *linear* and can be represented as a matrix multiplication.
- Function *f* is linear iff:

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

- Implications:
  - to transform a line we transform the *end-points*. Points between are *affine combinations* of the transformed endpoints.
  - Given line defined by points *P* and *Q*, points along transformed line are affine combinations of transformed P' and Q'

$$L(t) = P + t(Q - P)$$
$$L'(t) = P' + t(Q' - P')$$

Homogeneous Co-ordinates

- Basis of the homogeneous co-ordinate system is the set of *n* basis vectors and the origin position:

$$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \text{ and } P_o$$

- All points and vectors are therefore compactly represented using their ordinates:

$$\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ a_o \end{bmatrix} \text{or more usually} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Homogeneous Co-ordinates

- Vectors have no positional information and are represented using $a_o = 0$ whereas points are represented with $a_o = 1$:

$$\vec{v} = a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n + 0$$

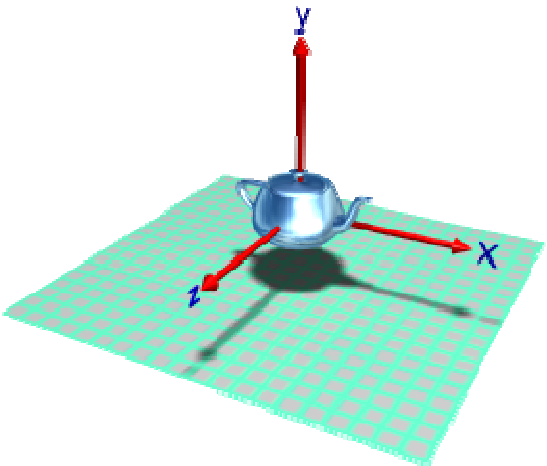
$$P = a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n + P_o$$

- Examples: $\begin{bmatrix} 0.2 \\ 1.3 \\ 2.2 \\ 1 \end{bmatrix} \begin{bmatrix} 1.0 \\ 1.0 \\ 0.0 \\ 1 \end{bmatrix}$ $\quad$ $\begin{bmatrix} 0.2 \\ 1.3 \\ 2.2 \\ 0 \end{bmatrix} \begin{bmatrix} 1.0 \\ 1.0 \\ 0.0 \\ 0 \end{bmatrix}$

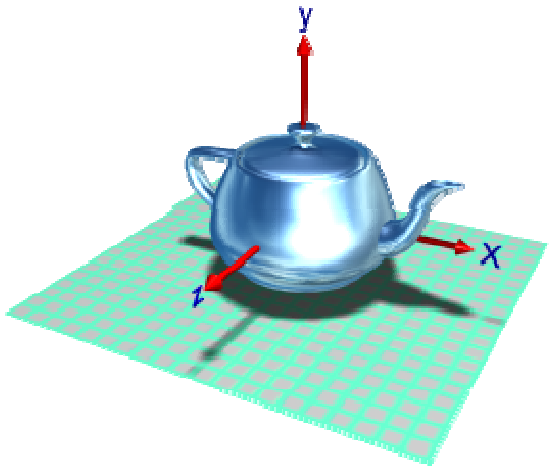
Points $\qquad$ Associated vectors

Scale

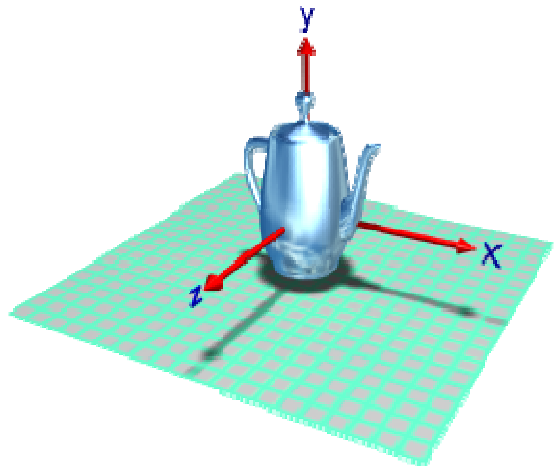
- all vectors are scaled from the origin:



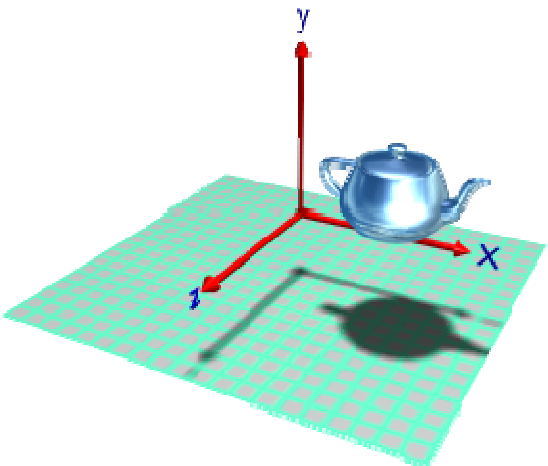
Original



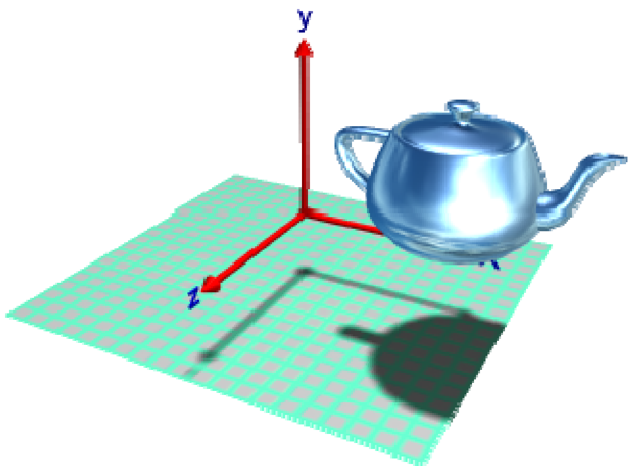
scale all axes



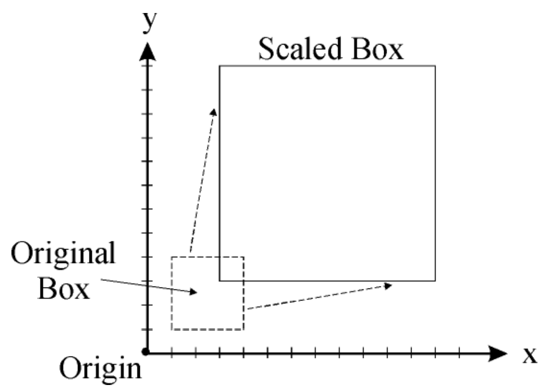
scale Y axis



offset from origin



distance from origin also scales



Scale

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ s_z z \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \mathbf{v}' = \mathbf{S}\mathbf{v}$$

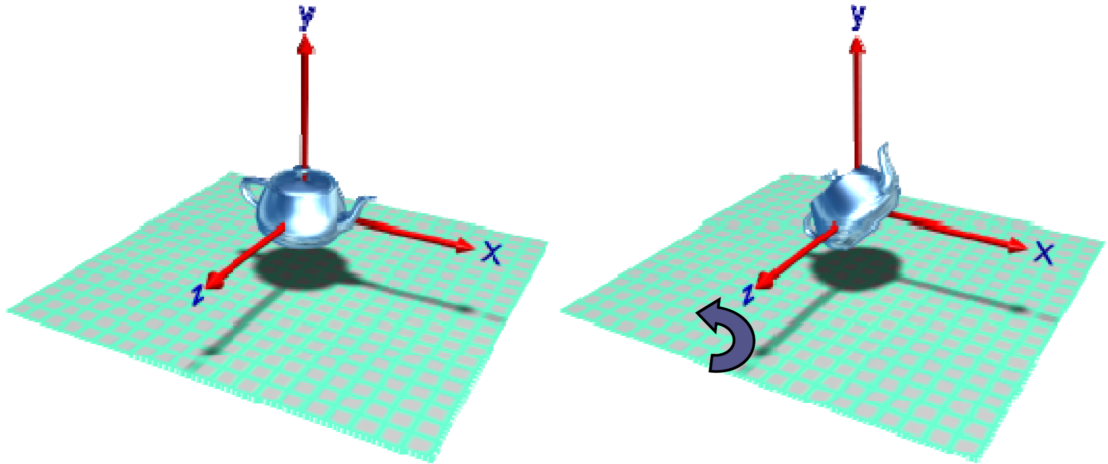
We would also like to scale points thus we need a *homogeneous transformation* for consistency:

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

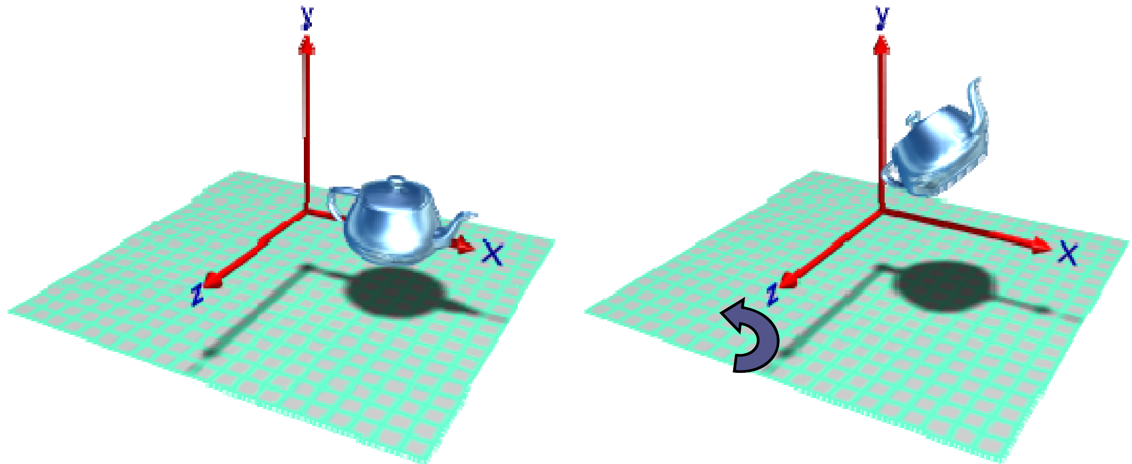
$$\mathbf{S}^{-1} = \begin{bmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 1/s_y & 0 & 0 \\ 0 & 0 & 1/s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation

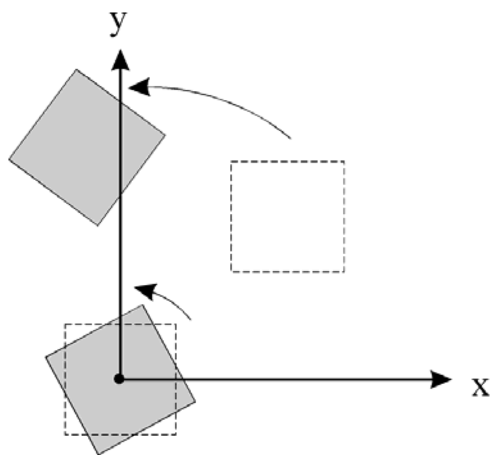
- Rotations are *anti-clockwise* about the *origin:*



rotation of 45^{o} about the Z axis

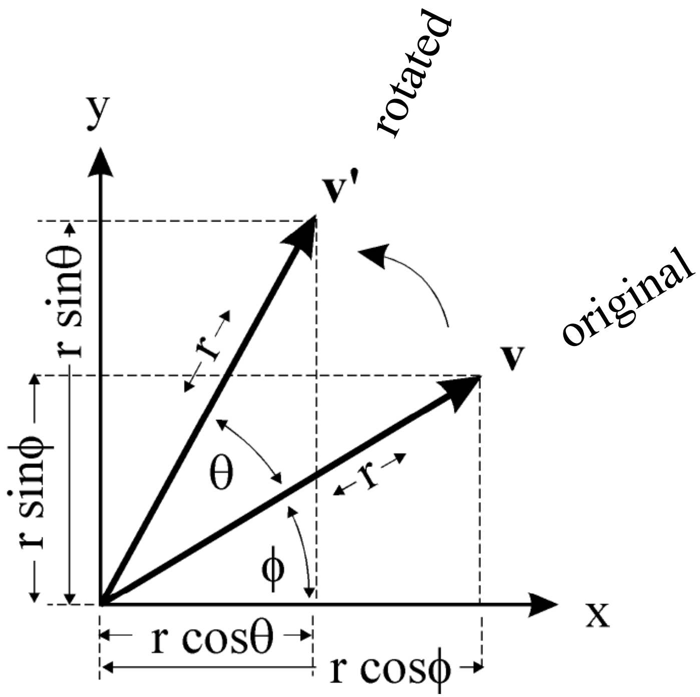


offset from origin rotation



Rotation

$$\mathbf{v} = \begin{bmatrix} r\cos\phi \\ r\sin\phi \end{bmatrix} \quad \mathbf{v}' = \begin{bmatrix} r\cos(\phi+\theta) \\ r\sin(\phi+\theta) \end{bmatrix}$$



$$\text{expand } (\phi+\theta) \Rightarrow \begin{cases} x' = r\cos\phi\cos\theta - r\sin\phi\sin\theta \\ y' = r\cos\phi\sin\theta + r\sin\phi\cos\theta \end{cases}$$

$$\text{but} \quad \begin{matrix} x = r\cos\phi \\ y = r\sin\theta \end{matrix} \Rightarrow \begin{matrix} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta + y\cos\theta \end{matrix}$$

Rotation

- 2D rotation of θ about origin: $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
- 3D homogeneous rotations:

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{R}_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{R}_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note:

$$\begin{matrix} \cos(-\theta) = \cos\theta \\ \sin(-\theta) = -\sin\theta \end{matrix} \Rightarrow \mathbf{R}^{-1}(\theta) = \mathbf{R}(-\theta) = \mathbf{R}^{T}(\theta)$$

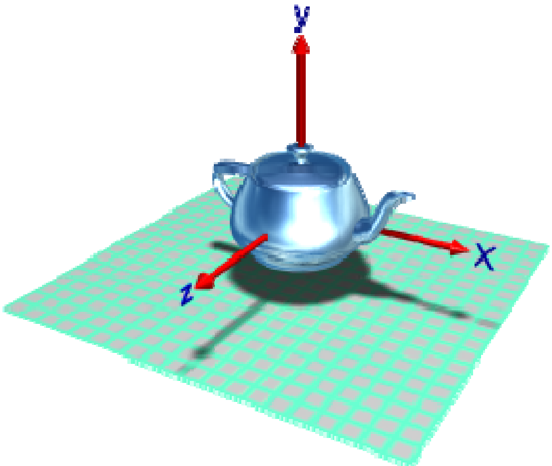
- If $\mathbf{M}^{-1} = \mathbf{M}^{T}$ then **M** is *orthonormal*. All orthonormal matrices are rotations about the origin.

Translation

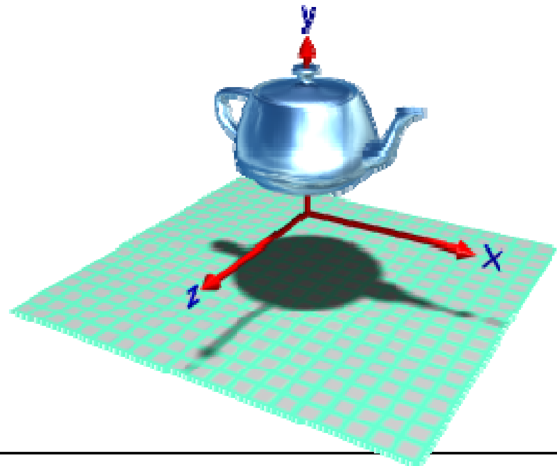
- Translation *only applies to points*, we never translate vectors.
- Remember: points have homogeneous co-ordinate $w = 1$

$$x' = x + a$$
$$y' = y + b$$
$$z' = z + c$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} x+a \\ y+b \\ z+c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

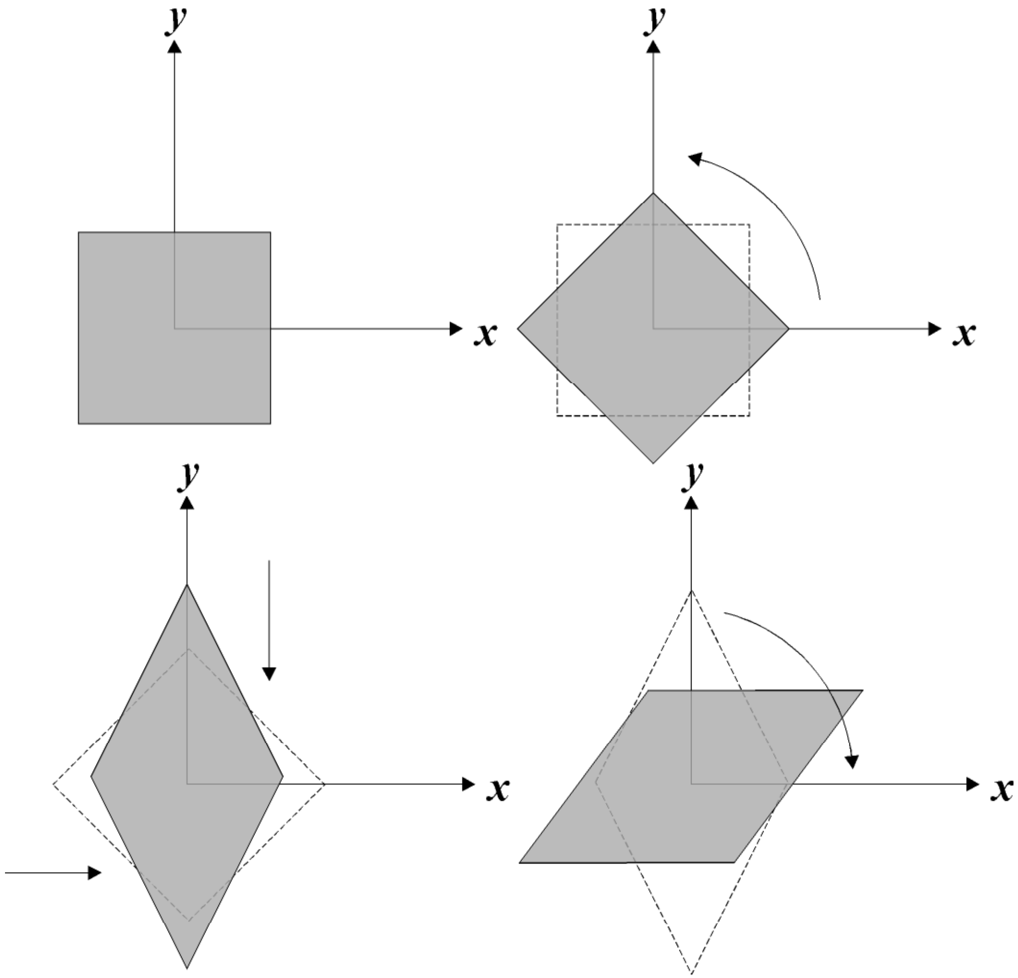


translate along ***y***



Affine Transformations

- All *affine transformations* are combinations of rotations, scaling and translations.



Transformation Composition

- More complex transformations can be created by *concatenating* or *composing* individual transformations together.

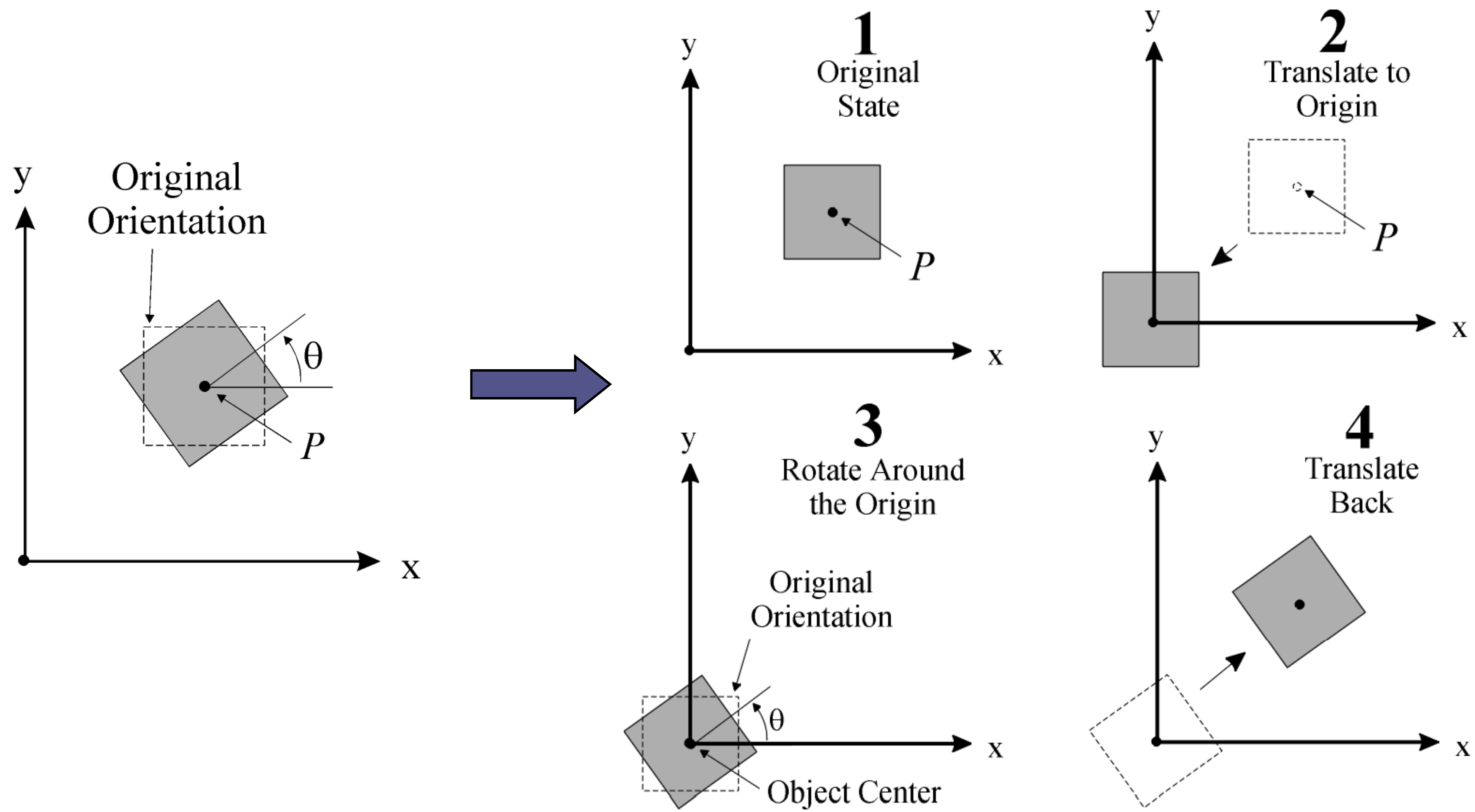
$$\mathbf{M} = \mathbf{T} \circ \mathbf{R} \circ \mathbf{S} \circ \mathbf{T} = \mathbf{TRST} \qquad \mathbf{v}' = \mathbf{T}[\mathbf{R}[\mathbf{S}[\mathbf{Tv}]]] = \mathbf{Mv}$$

- Matrix multiplication is *non-commutative* $\Rightarrow$ **order is vital**
- We can create an affine transformation representing rotation about a point P_R:

$$\mathbf{M} = \mathbf{T}(P_R)\mathbf{R}(\theta)\mathbf{T}(-P_R)$$

= *translate to origin, rotate about origin, translate back to original location*

Transformation Composition



Transformation Composition

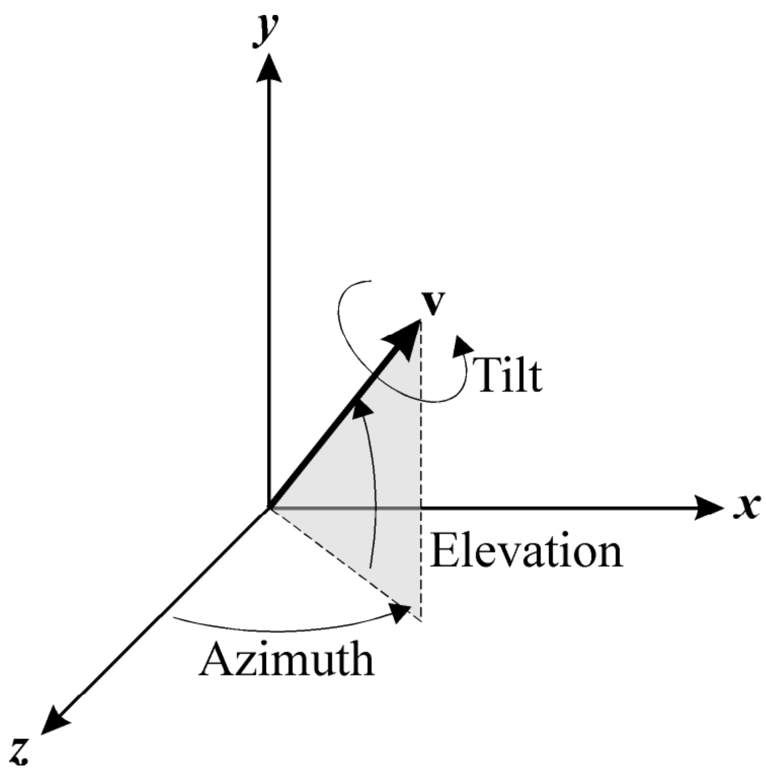
Rotation in **XY** plane by *q* degrees anti-clockwise about point *P*

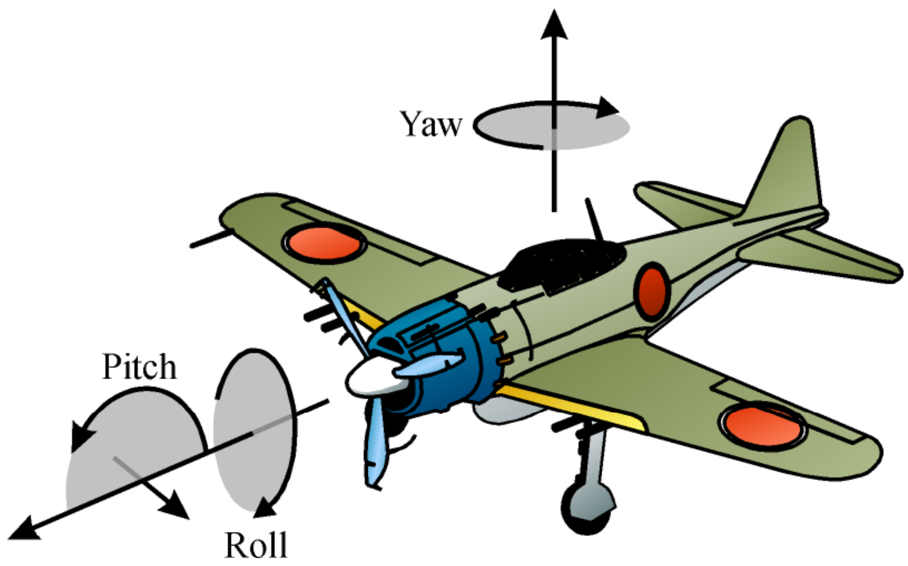
$$
\begin{aligned}
\mathbf{M} &= \mathbf{T}(P)\mathbf{R}(\theta)\mathbf{T}(-P) \\
&= \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & -P_x \\ 0 & 1 & 0 & -P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta & -\sin\theta & 0 & P_x - P_x\cos\theta + P_y\sin\theta \\ \sin\theta & \cos\theta & 0 & P_y - P_x\sin\theta - P_y\cos\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Euler Angles

- *Euler angles* represent the angles of rotation about the co-ordinate axes required to achieve a given orientation (θ_x, θ_y, θ_z)
- The resulting matrix is: $\mathbf{M} = \mathbf{R}(\theta_x)\mathbf{R}(\theta_y)\mathbf{R}(\theta_z)$
- Any required rotation may be described (*though not uniquely*) as a composition of 3 rotations about the coordinate axes.
- Remember rotation *does not commute* $\Rightarrow$ order is important

Rotational DOF



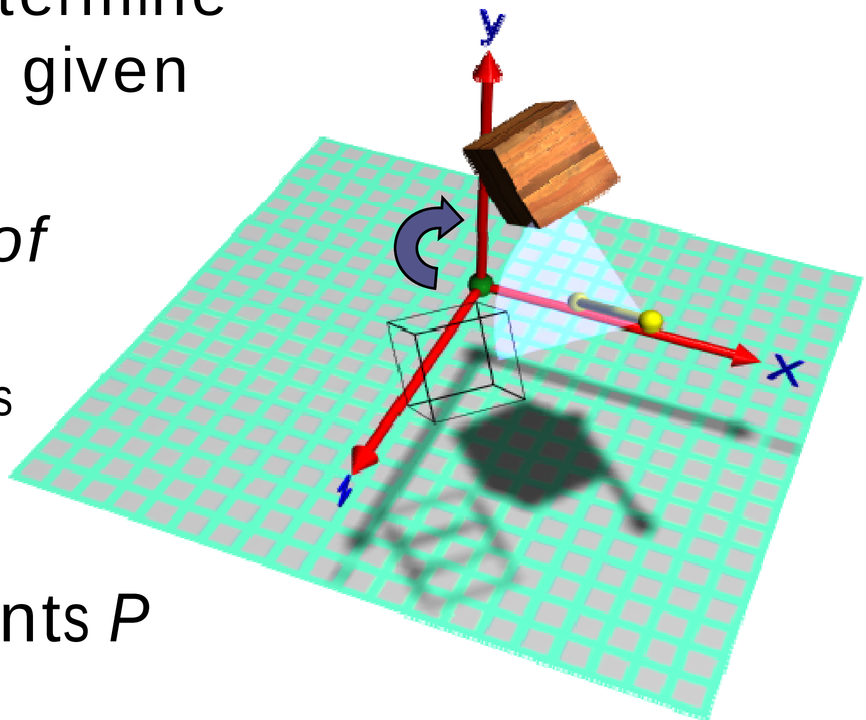


Sometimes known as *roll*, *pitch* and *yaw*

Rotation about an axis

- A frequent requirement is to determine the matrix for rotation about an given axis.
- Such rotations have 3 *degrees of freedom* (DOF):
  - 2 for spherical angles specifying axis orientation
  - 1 for twist about the rotation axis
- Assume axis is defined by points *P* and *Q*
- Pivot point is *P* and rotation axis vector is:

$$\mathbf{v} = \frac{P - Q}{|P - Q|}$$



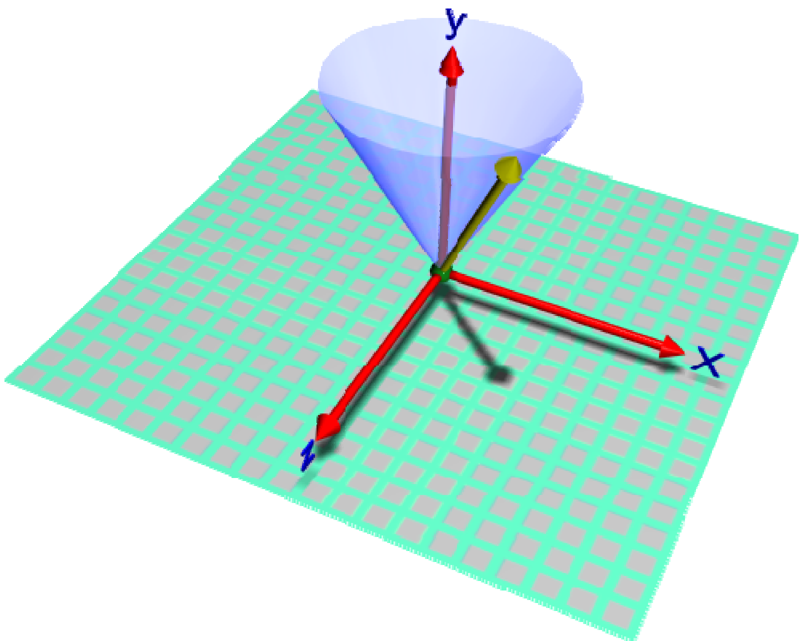
Rotation about an axis

1. Translate the pivot point to the origin $\Rightarrow$ **T**$(-P)$
2. Determine a series of rotations to achieve the required rotation about the desired vector.
3. Rotate the axis and object so that the axis lines up with **z** say $\Rightarrow$ **R**(θ_y)**R**(θ_x)
4. Rotate about **z** by the required angle $\theta \Rightarrow$ **R**(θ)
5. Undo the first 2 rotations to bring us back to the original orientation $\Rightarrow$ **R**$(-\theta_x)$**R**$(-\theta_y)$
6. Translate back to the original position $\Rightarrow$ **T**(P)
7. The final rotation matrix is:

$$\mathbf{M} = \mathbf{T}(P)\mathbf{R}(-\theta_y)\mathbf{R}(-\theta_x)\mathbf{R}(\theta)\mathbf{R}(\theta_x)\mathbf{R}(\theta_y)\mathbf{T}(-P)$$

Rotation about an axis

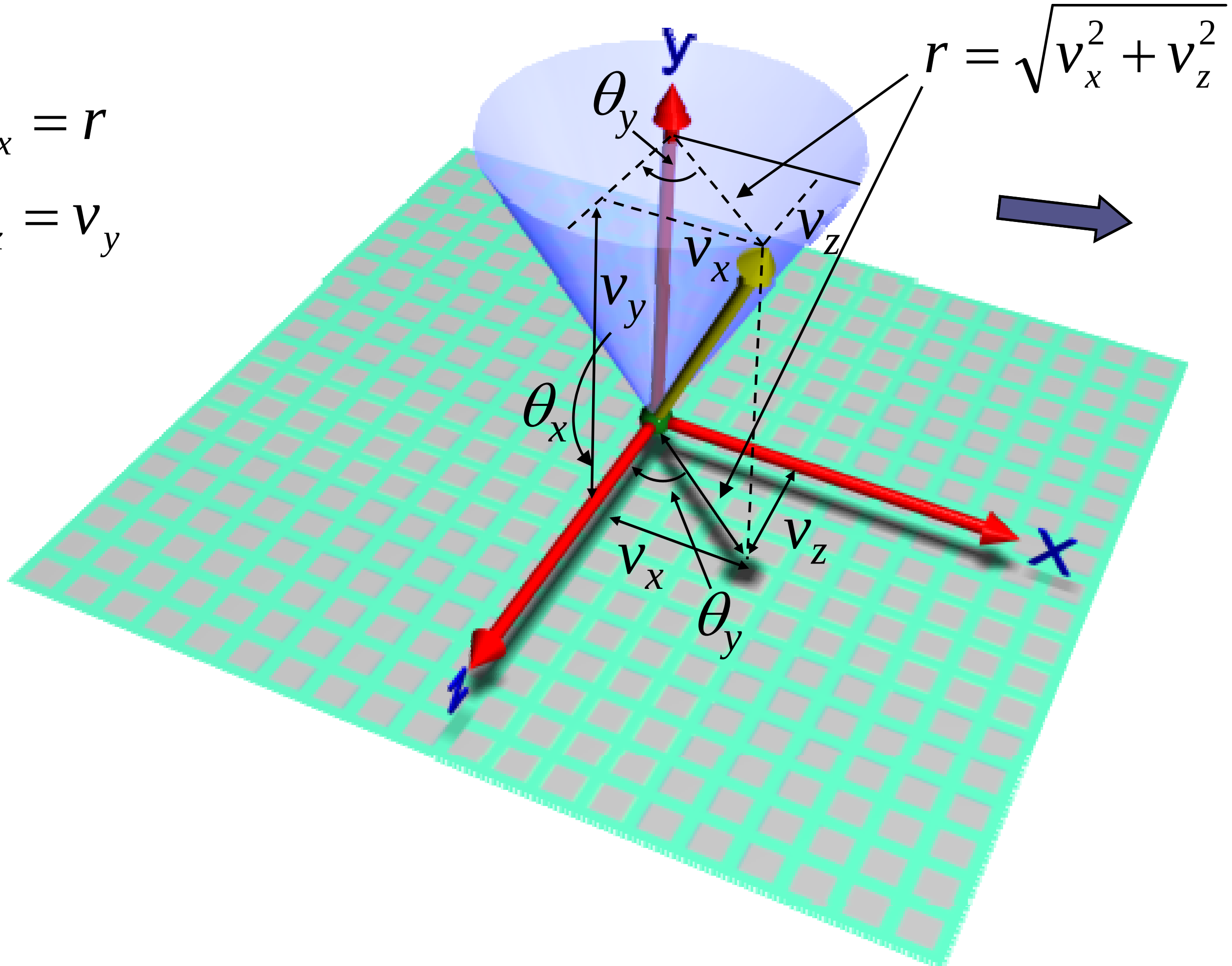
- We need the Euler angles θ_x and θ_y which will orient the rotation axis along the **z** axis.
- We determine these using simple trigonometry.

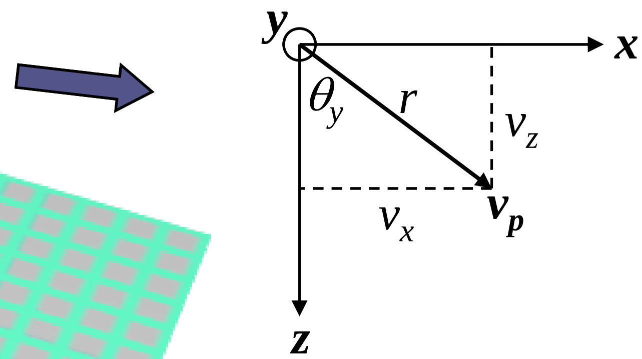


Aligning axis with z

$\cos\theta_x = r$

$\sin\theta_x = v_y$





$$\cos\theta_y = \frac{v_z}{r}$$

$$\sin\theta_y = \frac{v_x}{r}$$

Aligning axis with z

- Note that as shown the rotation about the **x** axis is anti-clockwise but the **y** axis rotation is *clockwise*.
- Therefore the required **y** axis rotation is $-\theta_y \Rightarrow$

$$\mathbf{R}(\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & r & -v_y & 0 \\ 0 & v_y & r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \mathbf{R}(-\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & r & v_y & 0 \\ 0 & -v_y & r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}(\theta_y) = \begin{bmatrix} {}^{v_z}\!/_{r} & 0 & {}^{v_x}\!/_{r} & 0 \\ 0 & 1 & 0 & 0 \\ -{}^{v_x}\!/_{r} & 0 & {}^{v_z}\!/_{r} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \mathbf{R}(-\theta_y) = \begin{bmatrix} {}^{v_z}\!/_{r} & 0 & -{}^{v_x}\!/_{r} & 0 \\ 0 & 1 & 0 & 0 \\ {}^{v_x}\!/_{r} & 0 & {}^{v_z}\!/_{r} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{M} = \mathbf{T}(P)\mathbf{R}(\theta_y)\mathbf{R}(-\theta_x)\mathbf{R}(\theta)\mathbf{R}(\theta_x)\mathbf{R}(-\theta_y)\mathbf{T}(-P)$$