# Linear Transformations

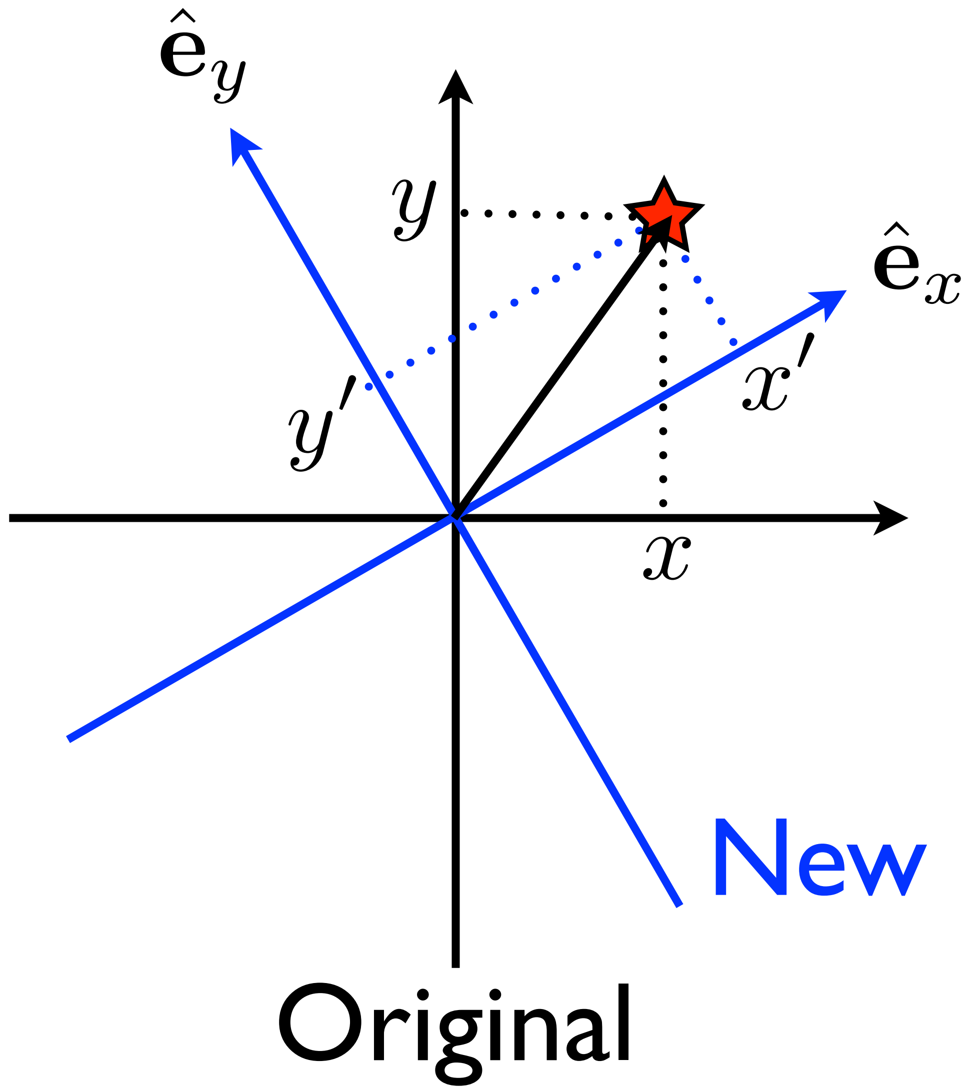
CS 355: Interactive Graphics and Image Processing

# Change of Coordinates

- To compute coordinates in the rotated system, just *project to each of the new axis directions*

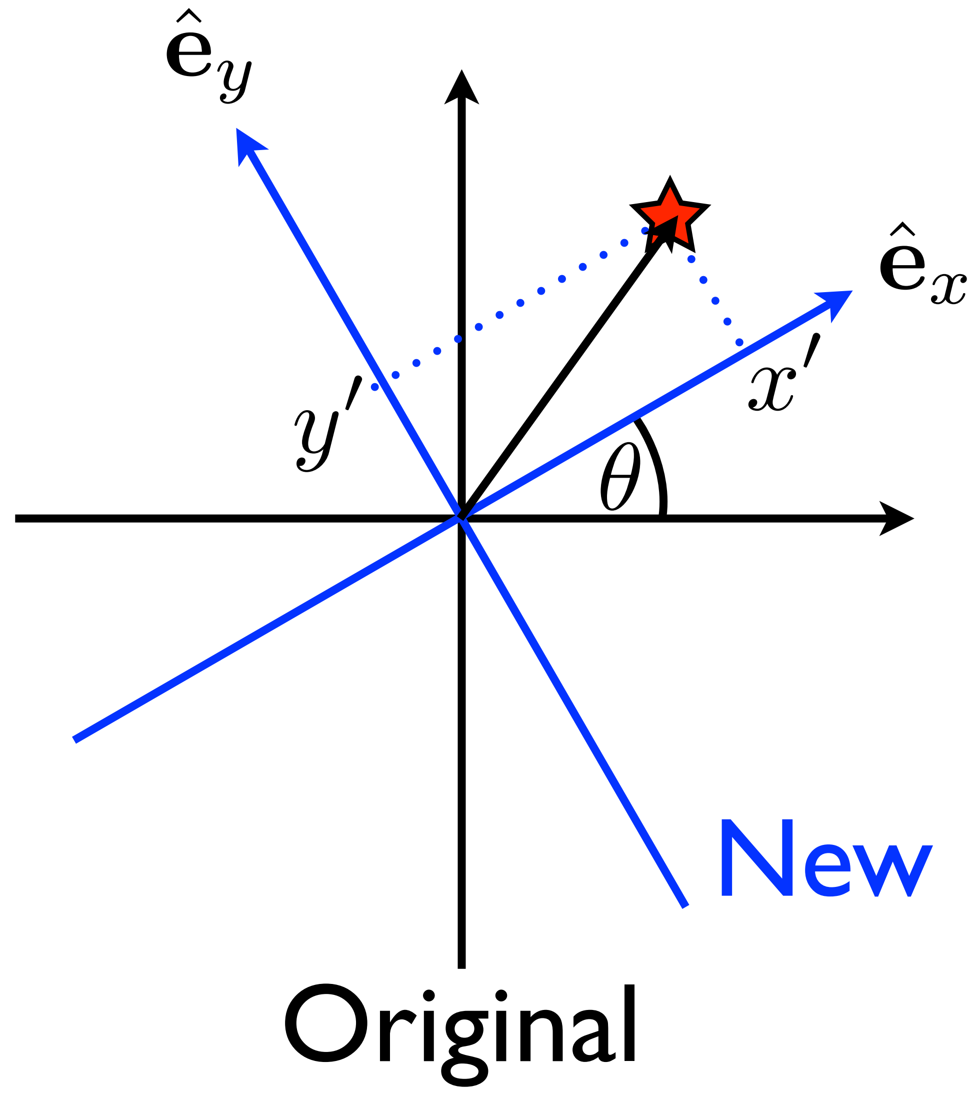
- Use a matrix to do the multiple dot products:

$$\mathbf{p}' = \begin{bmatrix} e_{x1} & e_{x2} \\ e_{y1} & e_{y2} \end{bmatrix} \mathbf{p}$$



# Rotation

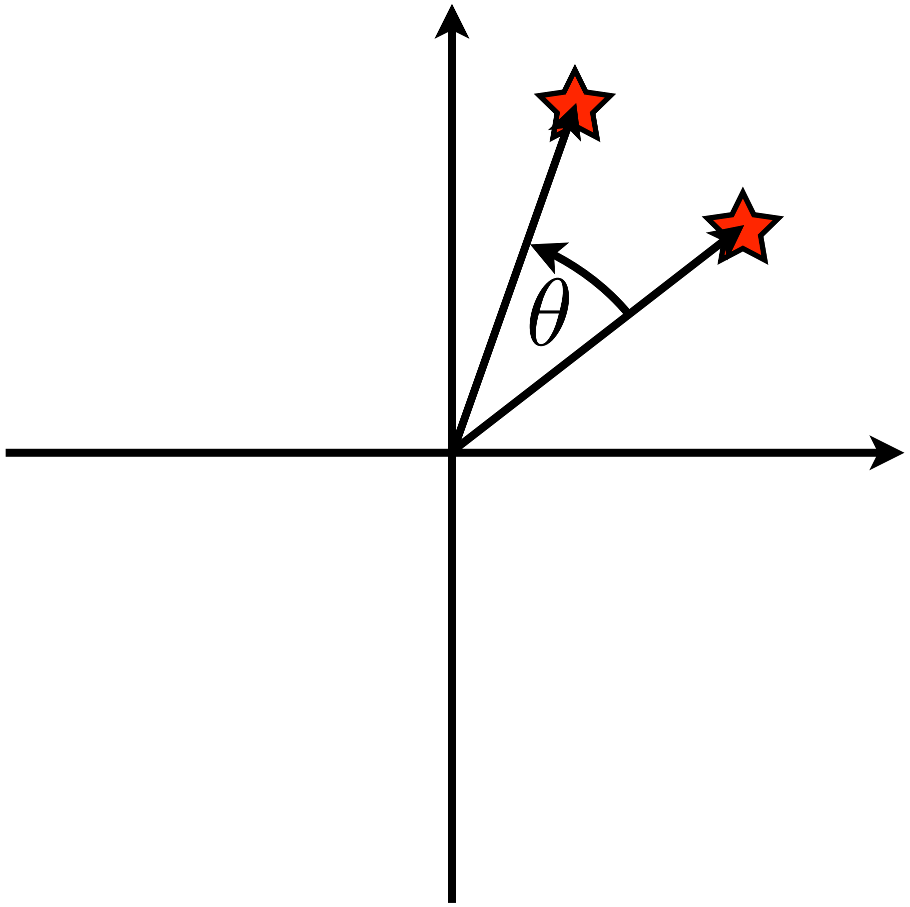
- Rotating the coordinate system one way rotates the point/object the other way
- From here on we'll talk in terms of rotating the point or object



# Rotation

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$\mathbf{R}^{-1}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$



# Rotation and Translation

- Object to world: $$\mathbf{p}_w = \mathbf{R}(\theta)\ \mathbf{p}_o + \mathbf{c}$$
  - Rotate first to new orientation
  - Translate to new position

# Inverse

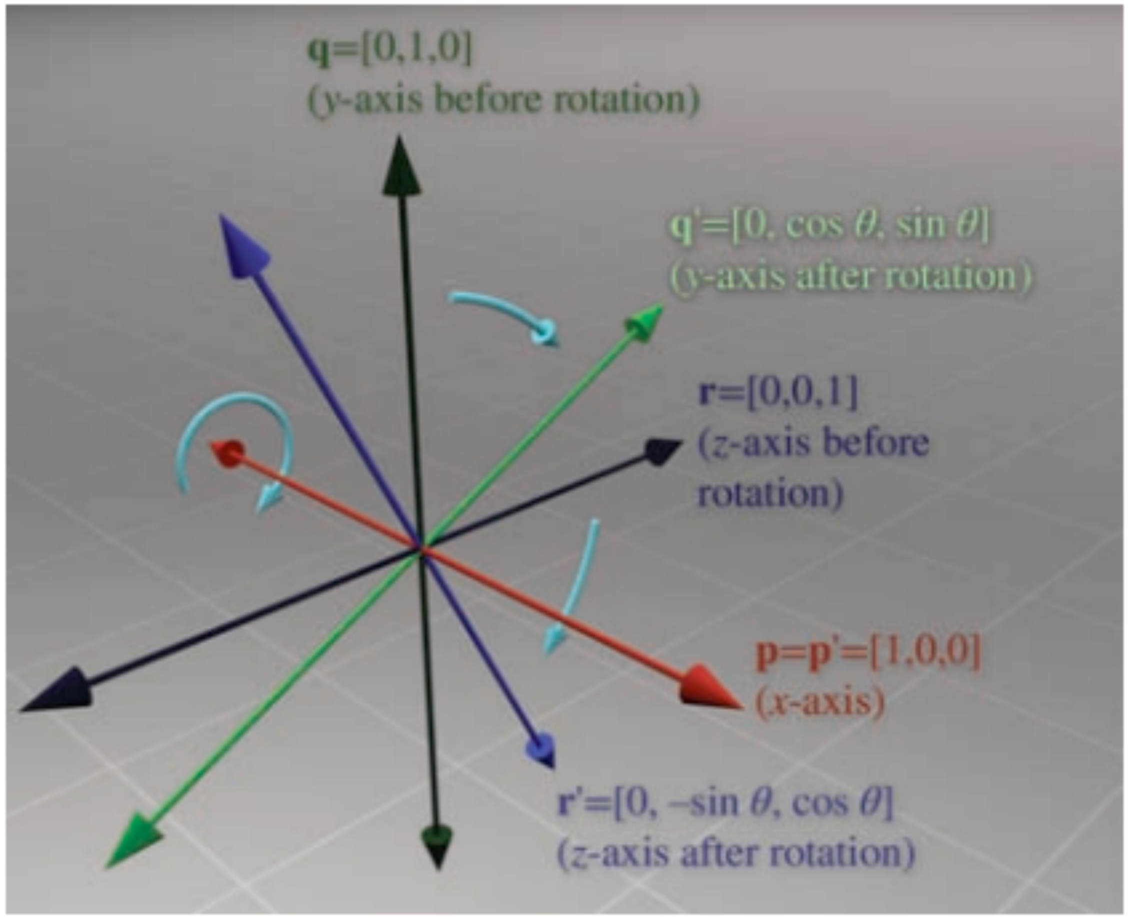
- World to object: $$\mathbf{p}_o = \mathbf{R}^{-1}(\theta)\ (\mathbf{p}_w - \mathbf{c})$$
  - Translate back to origin
  - Rotate back to upright

# Rotation in 3D

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

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

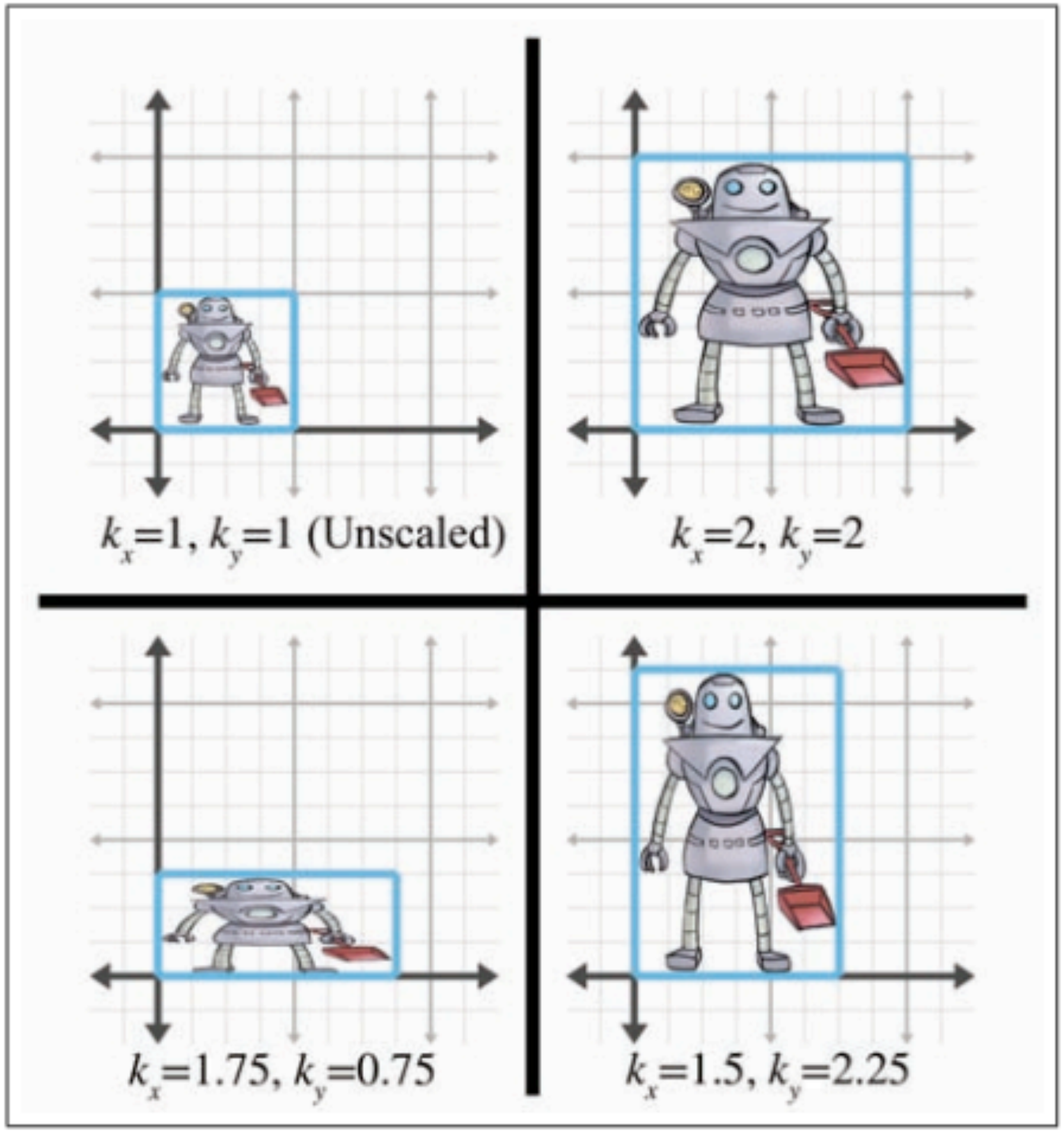
$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



# Scaling

$$\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} sx \\ sy \end{bmatrix}$$

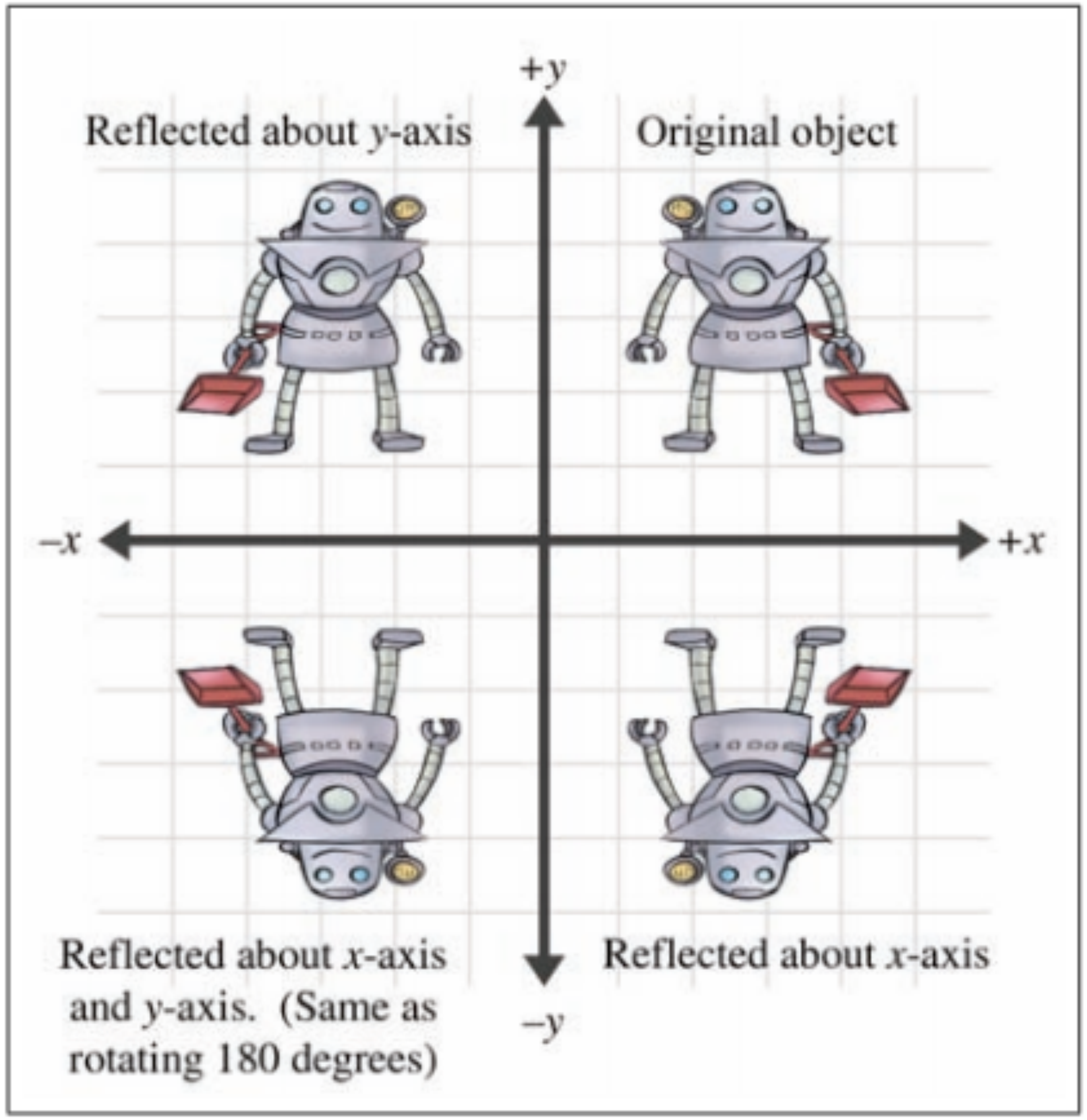
$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x\, x \\ s_y\, y \end{bmatrix}$$



# Reflection

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

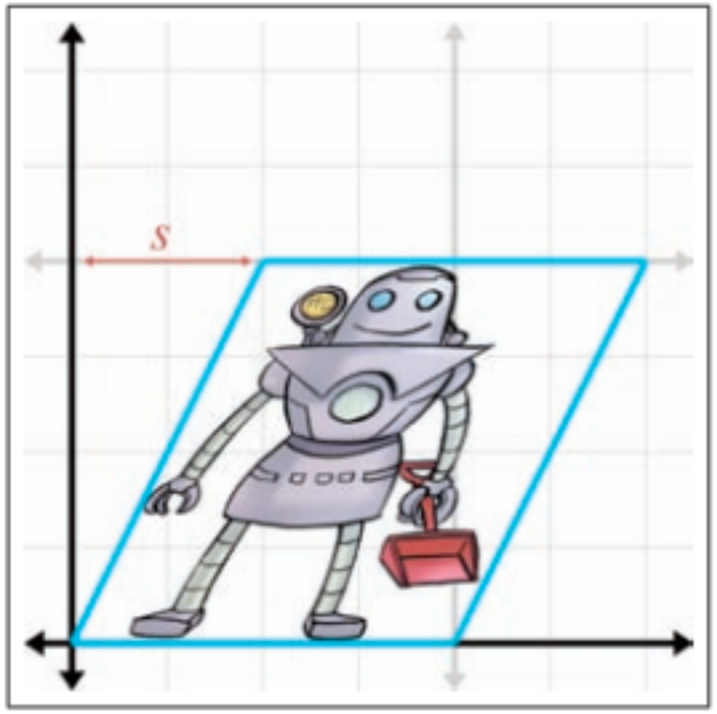
$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ -y \end{bmatrix}$$



# Shearing

$$\begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + sy \\ y \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ sx + y \end{bmatrix}$$




Sometimes called a *skew* transform

# Composition

$$\mathbf{M}_2 = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \qquad \mathbf{M}_1 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$\mathbf{M}_{12} = \mathbf{M}_2\mathbf{M}_1$$

$$\mathbf{M}_{12}\mathbf{p} = (\mathbf{M}_2\mathbf{M}_1)\mathbf{p} = \mathbf{M}_2(\mathbf{M}_1\mathbf{p})$$

# But...

- Composition of transformations (matrix multiplication) is really useful
- But translation (adding a vector offset) is a separate operation
- Is there a way to include translation as a linear transformation?

# Homogeneous Coordinates

- Can include translation as a linear transformation using *homogeneous coordinates*
- Also useful for perspective (covered later)

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ \color{red}{1} \end{bmatrix}$$

Add additional "homogeneous" element

# Homogeneous Matrices

- To make a matrix homogeneous, add an extra row and column

- To do nothing else, make these [0 0 1]

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

$$\begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Translation

$$\begin{bmatrix} 1 & 0 & \boxed{\begin{matrix} t_x \\ t_y \end{matrix}} \\ 0 & 1 & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

(General note: for row vectors, transpose this)

# Combining With Translation

Normal operation

$$\begin{bmatrix} m_{11} & m_{12} & t_x \\ m_{21} & m_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}x + m_{12}y + t_x \\ m_{21}x + m_{22}y + t_y \\ 1 \end{bmatrix}$$

Always [0 ... 0 1]
(for now)

Translation

Normal operation <u>then</u> translation

# Rotation <u>then</u> Translation

Rotation Translation

$$\begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$p' = \mathbf{R}(\theta)\ \mathbf{p} + \mathbf{t}$$

# Rotation <u>then</u> Translation

Rotation        Translation

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \qquad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

# Object to World

Move to location                Rotate

$$\mathbf{O}_i = \begin{bmatrix} 1 & 0 & c_i[x] \\ 0 & 1 & c_i[y] \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & c_i[x] \\ \sin\theta_i & \cos\theta_i & c_i[y] \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{p}_w = \mathbf{O}_i \; \mathbf{p}_o$$

$$\mathbf{p}_w = \underline{\mathbf{R}(\theta)} \; \mathbf{p}_o + \underline{\mathbf{c}}$$

# World to Object

Rotate back Move back to origin

$$\mathbf{O}_i^{-1} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -c_i[x] \\ 0 & 1 & -c_i[y] \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_i & \sin\theta_i & -\cos\theta\ c_i[x] - \sin\theta\ c_i[y] \\ -\sin\theta_i & \cos\theta_i & \sin\theta\ c_i[x] - \cos\theta\ c_i[x] \\ 0 & 0 & 1 \end{bmatrix}$$

Rotate first then translate by *rotated* offset back to center

$$\mathbf{p}_o = \mathbf{O}_i^{-1}\ \mathbf{p}_w$$

$$\mathbf{p}_o = \underline{\mathbf{R}^{-1}(\theta)}\ (\mathbf{p}_w \underline{- \mathbf{c}})$$

$$= \underline{\mathbf{R}^{-1}(\theta)}\ \mathbf{p}_w + \underline{\mathbf{R}^{-1}(\theta)\ (-\mathbf{c})}$$

# Points vs. Vectors

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ {\color{red}1} \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} x \\ y \\ {\color{red}0} \end{bmatrix}$$

allows translation

ignores translation

Remember: points have a position and can be translated, but vectors have only a displacement

# Classes of Transformations

- Rigid body (preserves lengths):
  Rotation and translation
- Euclidean (preserves angles):
  rotation, translation, scale
- Affine:
  Arbitrary linear combinations

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

General form of an
affine transformation

# Coming up...

- A few more matrix things...