Spatial Transformations

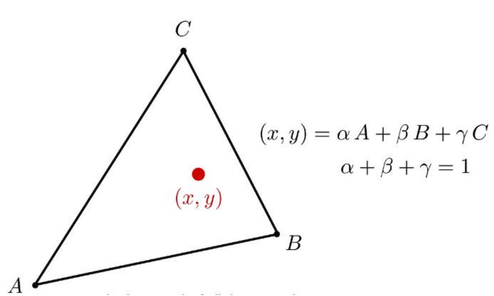
Points versus **vectors**

- Both are represented as a list of coordinates
- The distinction enables checking validity of geometric operations.

Point

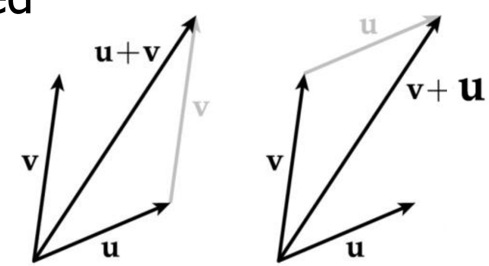
- A location in space
- **P**+**Q** is not defined
- α**P** is not defined
- $\sum \alpha_i \mathbf{P}_i$ is defined if

$$\sum \alpha_i = 1$$



Vector 

- Has direction and magnitude
- Free to move anywhere
- **v** + **u** is defined

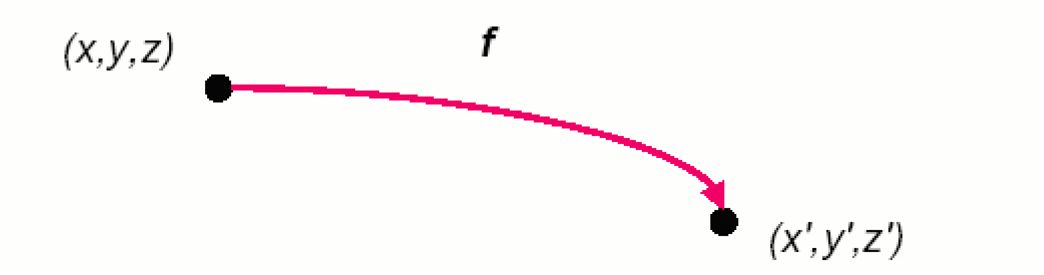


- α**u** is defined

Spatial transformations

Any function that assigns each point a new location:

$$f(x, y, z) = (x', y', z')$$



These transformations can be

- very simple, such as scaling each coordinate
- complex, such as non-linear twists and bends.

We'll focus on transformations that can be represented with **matrix operations**.

Representation

Represent a 2D **point** as

- a column vector $\begin{bmatrix} x \\ y \end{bmatrix}$

- or, a row vector $\begin{bmatrix} x & y \end{bmatrix}$

In graphics, we use **column vectors**.

Representation

Represent a 2D **transformation** M by a **matrix**

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Let p be a **column** vector. Then M goes on the **left**:

$$\mathbf{p}' = M\mathbf{p}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Transformations

Here's all you get with a **2x2** transformation matrix M:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

So:

$$x' = ax + by$$
$$y' = cx + dy$$

We will develop some intimacy with the elements a, b, c, d …

Identity

Suppose we choose a=d=1, b=c=0:

- Gives the **identity** matrix:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

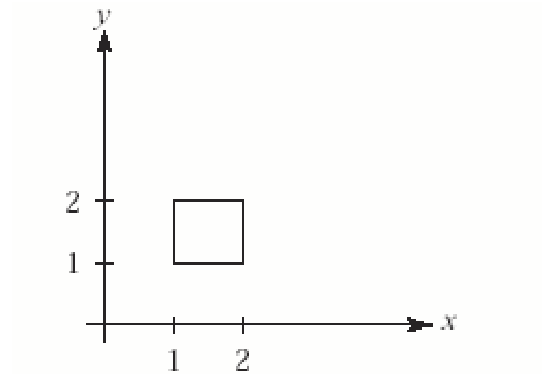
- Doesn't move the points at all

Scaling

Suppose we set b=c=0, but let a and d take on any positive value:

- Gives a **scaling** matrix:

$$\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}$$

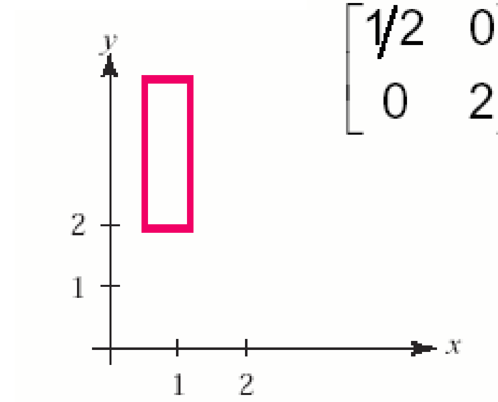


- Provides **non-uniform scaling** in x and y:

$$x' = ax$$

$$y' = dy$$

$$\begin{bmatrix} 1/2 & 0 \\ 0 & 2 \end{bmatrix}$$

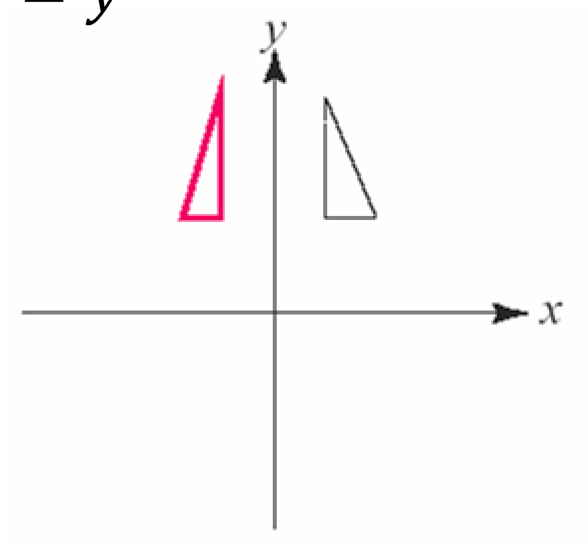


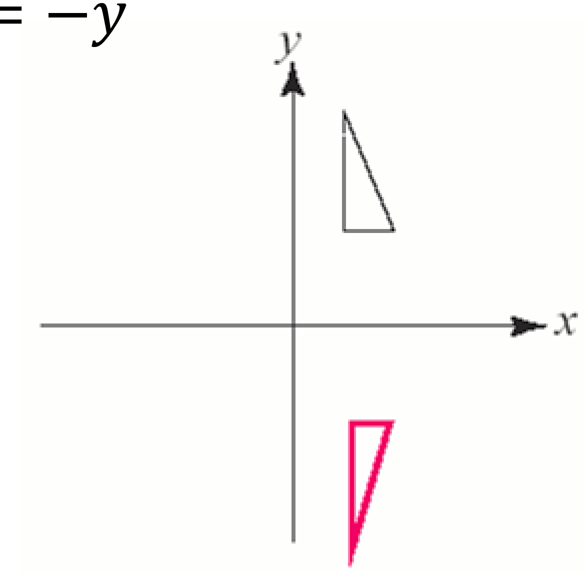
Negative Scaling

Suppose $a = -1$ or $d = -1$:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$x' = -x \qquad x' = x$$
$$y' = y \qquad y' = -y$$





Each reflection reverses orientation

Negative Scaling

- If $a = d = -1$, we can think of the scaling as a sequence of reflections
- In 2D

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

- Since each reflection reverse orientation, two reflections preserve orientation

Shearing

Now let a=d=1 and experiment with b ….

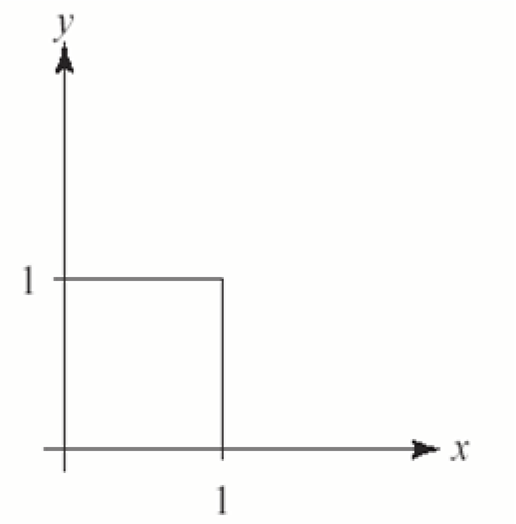
The matrix $\begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}$

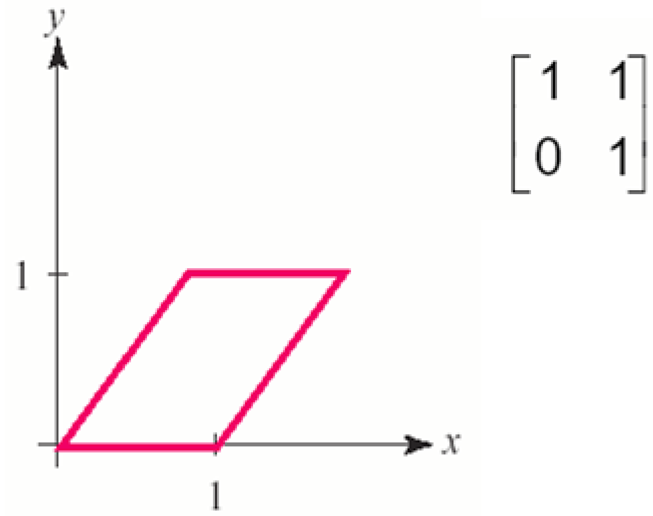
gives:

$$x' = x + by$$

$$y' = y$$

displaces x coordinate according to the distance along y direction



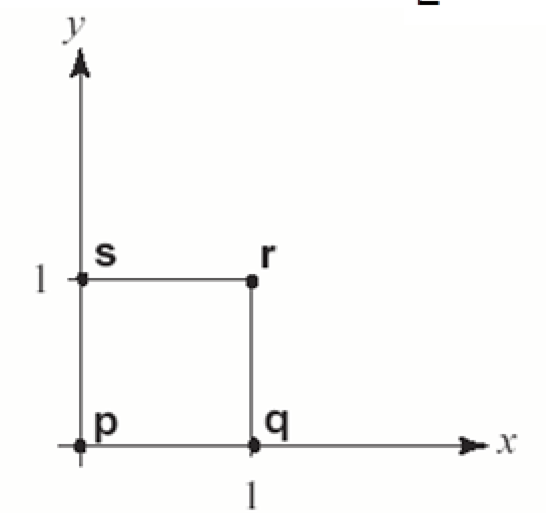


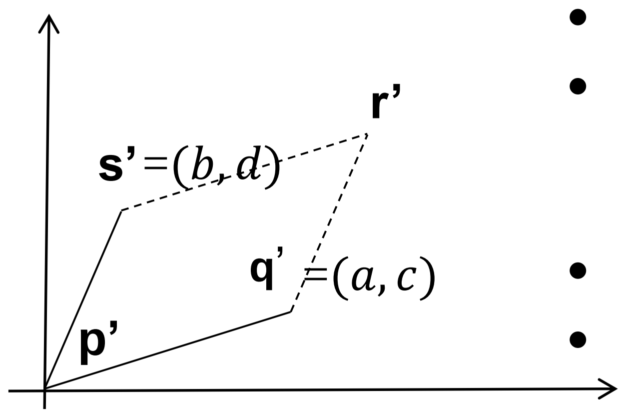
Effect on unit square

Let’s see how a general 2x2 transformation M affects the **unit square**:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \mathbf{p} & \mathbf{q} & \mathbf{r} & \mathbf{s} \end{bmatrix} = \begin{bmatrix} \mathbf{p}' & \mathbf{q}' & \mathbf{r}' & \mathbf{s}' \end{bmatrix}$$

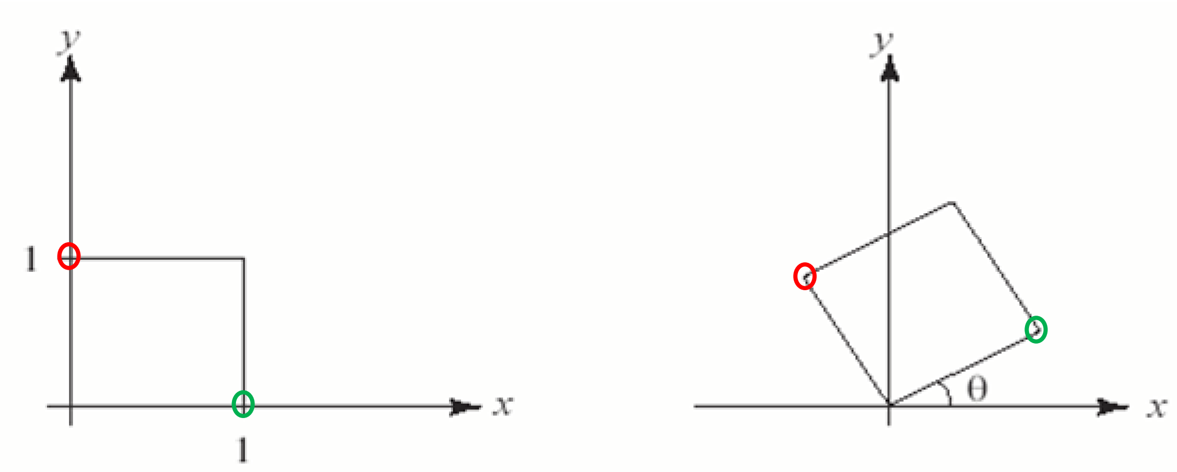
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & \boxed{\begin{matrix} a \\ c \end{matrix}} & \begin{matrix} a+b \\ c+d \end{matrix} & \boxed{\begin{matrix} b \\ d \end{matrix}} \end{bmatrix}$$





- Origin invariant under M
- M can be determined just by knowing where (1,0) and (0,1) are mapped to
- a and d give x- and y-**scaling**
- b and c give x- and y-**shearing**

Rotation



$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix}$$

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

Linear transformations

We have seen a **2x2 matrix** allows

- Scaling
- Rotation
- Reflection
- Shearing

- They are called **linear transformations**
  - take lines to lines
  - keep the origin fixed
  - Preserves midpoints (in general, preserves ratios)
  - not necessarily preserve Euclidean distances and angles

- What about translation?

Affine transformations

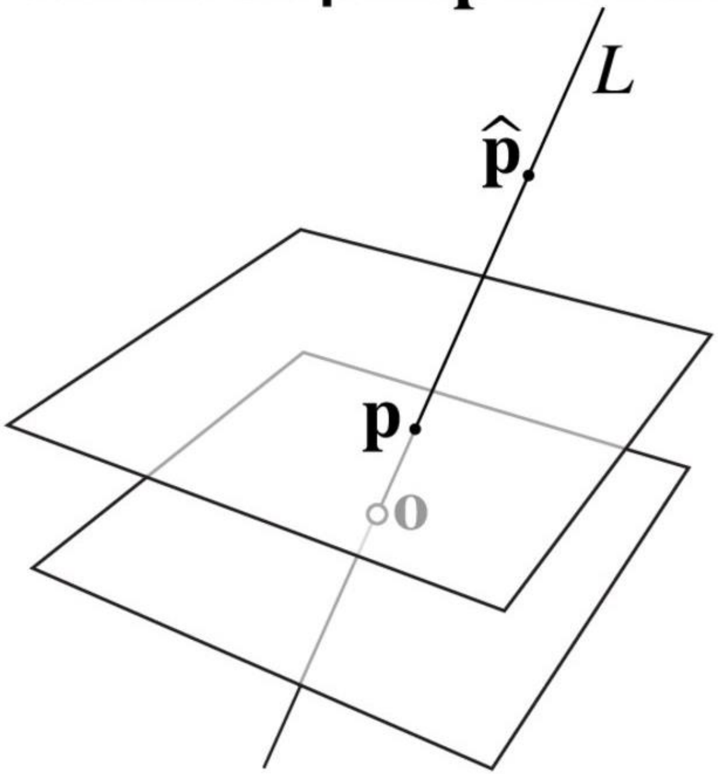
- 2D translation cannot be represented by a 2x2 matrix.
- Translation is an **affine transformation** (unlike linear transformations, it need not preserve origin)
- An important computer graphics magic trick to turn affine transformations to linear transformations is to use **homogeneous coordinates**

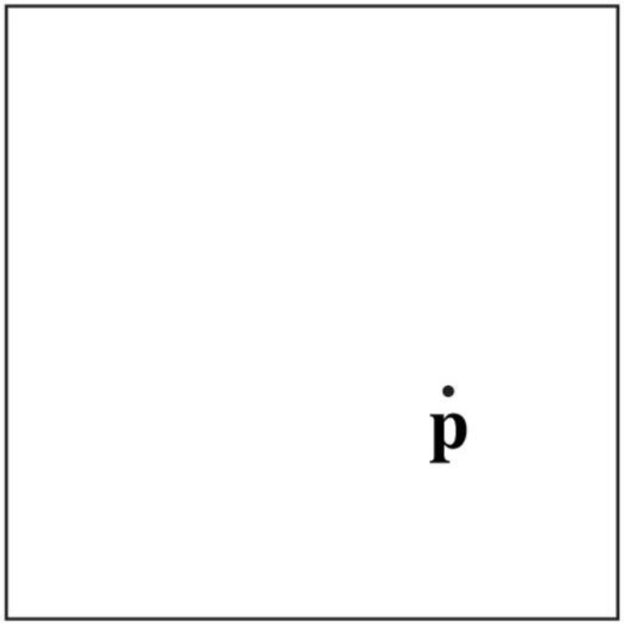
Homogeneous Coordinates

- Homogeneous coordinates show up naturally in a surprising large number of places in computer graphics
  - 3D transformations
  - Perspective projection
  - Clipping
  - Directional lights
  - Rational B-splines, NURBS
  - Quadric error simplification
  - .....

Homogeneous Coordinates—Basic Idea

- Consider any 2D plane that does not pass through the origin $\mathbf{o}$ in 3D
- Every <u>line</u> through the origin in 3D corresponds to a <u>point</u> in the 2D plane
  - Just find the point $\mathbf{p}$ where the line L pierces the plane

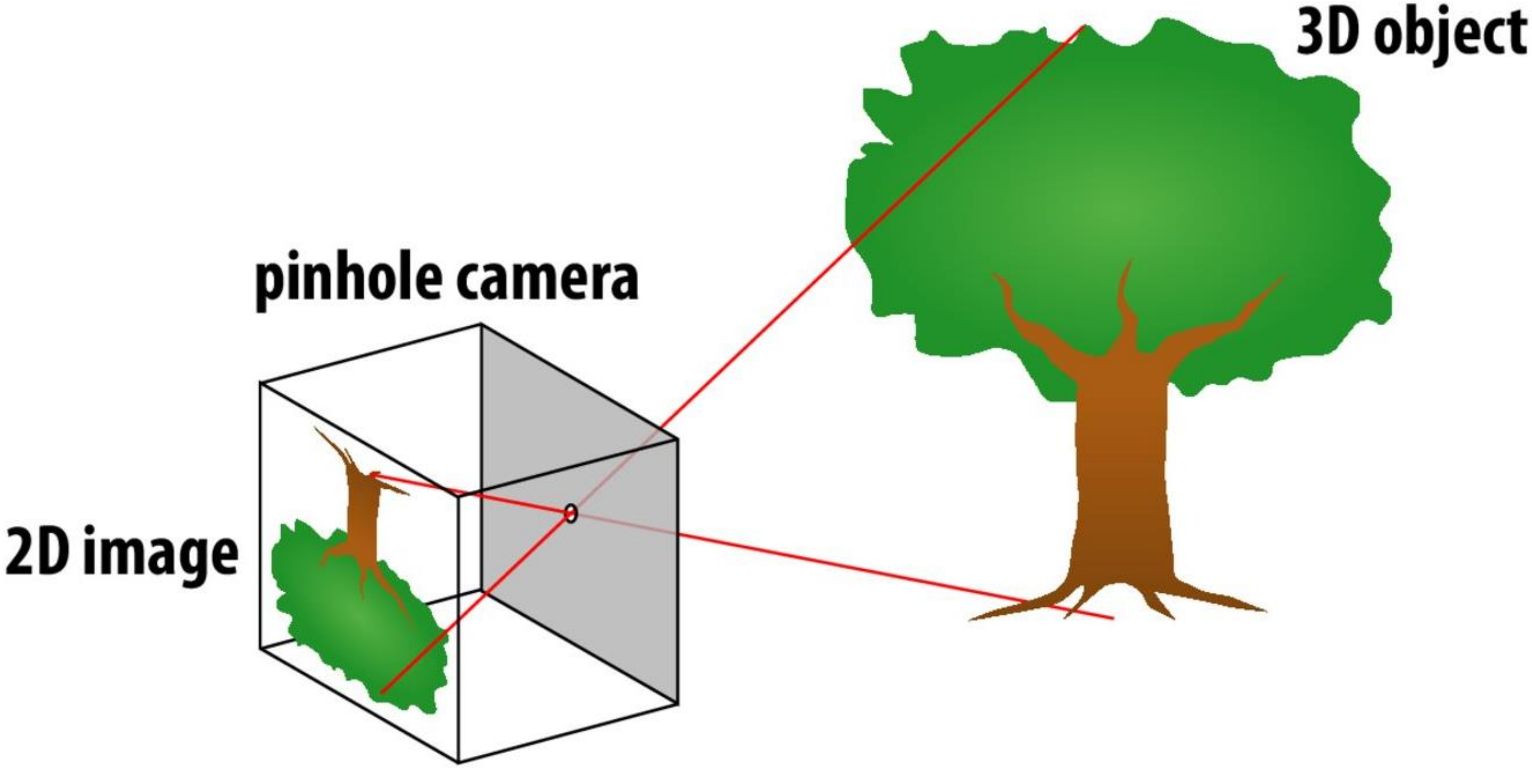




Hence, <u>any</u> point $\widehat{\mathbf{p}}$ on the line L can be used to represent the point $\mathbf{p}$.

Review: Perspective projection

- Hopefully it reminds you of our "pinhole camera"
- Objects along the same line project to the same point

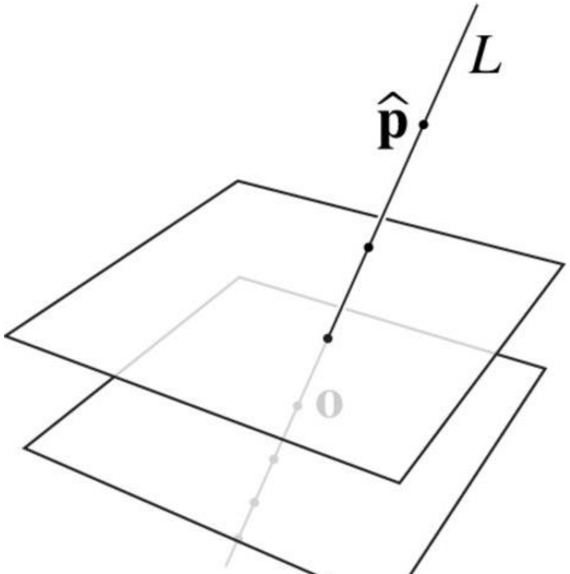


If you have an image of a single dot, can't know where it is!
Only which line it belongs to.

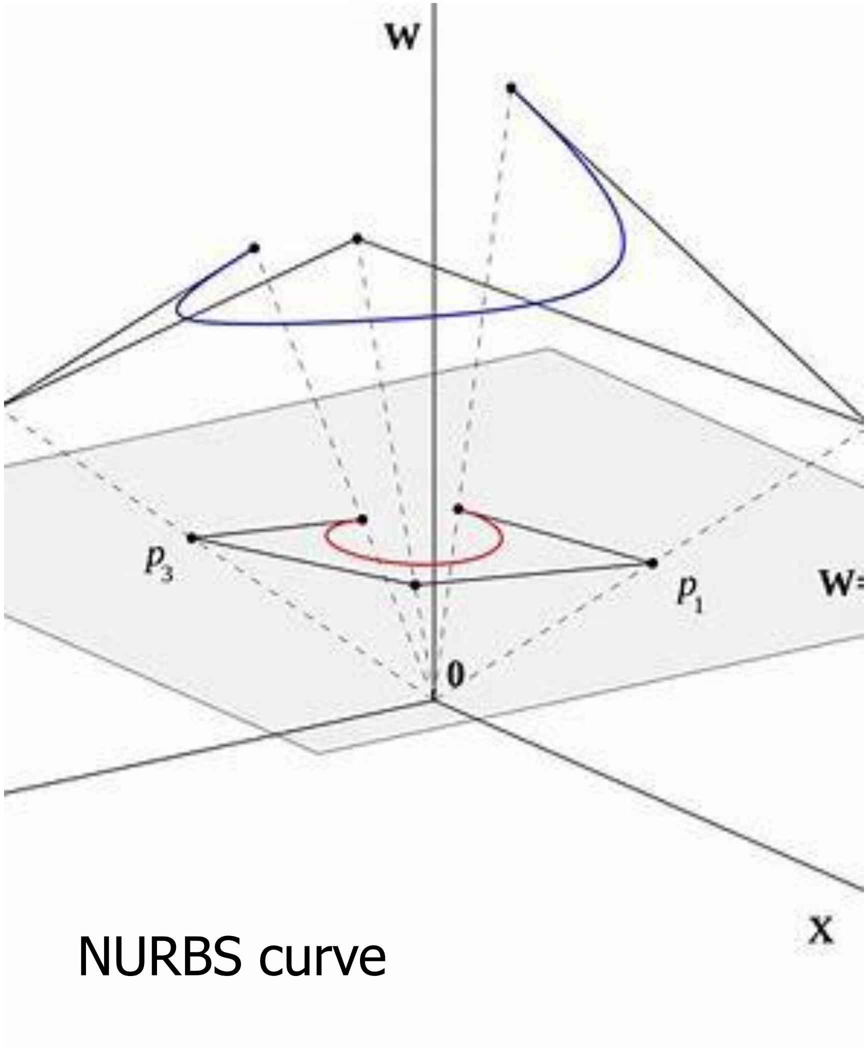
Homogeneous Coordinates (2D)

- More explicitly, consider a point $\mathbf{p} = (x, y)$, and the plane $z = 1$ in 3D

- Any three numbers $\widehat{\mathbf{p}} = (a, b, c)$ such that $(a/c, b/c) = (x, y)$ are <u>homogeneous coordinates</u> for $\mathbf{p}$
  - E.g., $(x, y, 1)$
  - In general: (cx, cy, c) for $c \neq 0$



Homogeneous coordinates



NURBS curve

Translation with Homogeneous coordinates

Idea is to loft the problem up into 3-space

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

And then transform with a **3x3 matrix**:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = x + t_x$$
$$y' = y + t_y$$
$$w' = 1$$

Why care about representing transformations as matrices?

- Composition of transformations is matrix product
- Product of **many** matrices is a **single** matrix
- Gives uniform representation of transformations
- Simplifies graphics algorithms, systems (e.g., GPUs & APIs)

$$\underbrace{\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\textit{rotation}} \underbrace{\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}}_{\textit{scale}} \underbrace{\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}}_{\textit{rotation}} \cdots = \underbrace{\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}}_{\textit{composite transformation}}$$

Rotation around arbitrary point

With homogeneous coordinates, we can compose transformations including translations

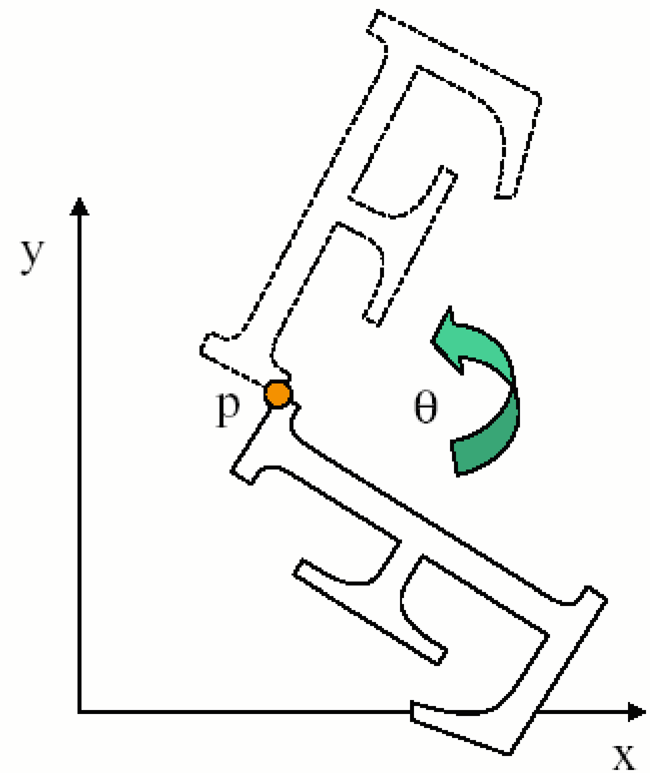
E.g., allow **rotations about arbitrary point** to be specified as a matrix

1. Translate object so that pivot is at origin: M1
2. Rotate object: M2
3. Translate so that pivot is back to original position $M1^{-1}$

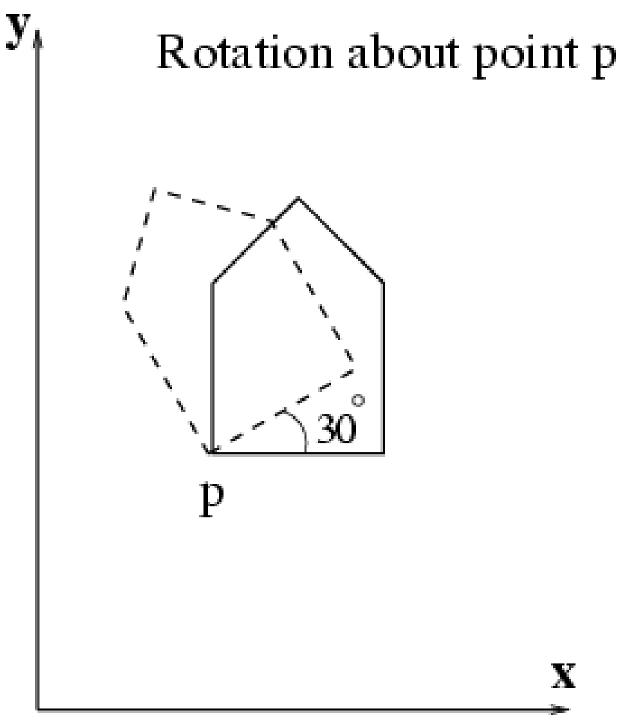
p' = $M1^{-1}$ M2 M1 **p**

Order is important!

Read from right to left



Compositing multiple transformations



Example:

3 steps

1) T(-2, -3)

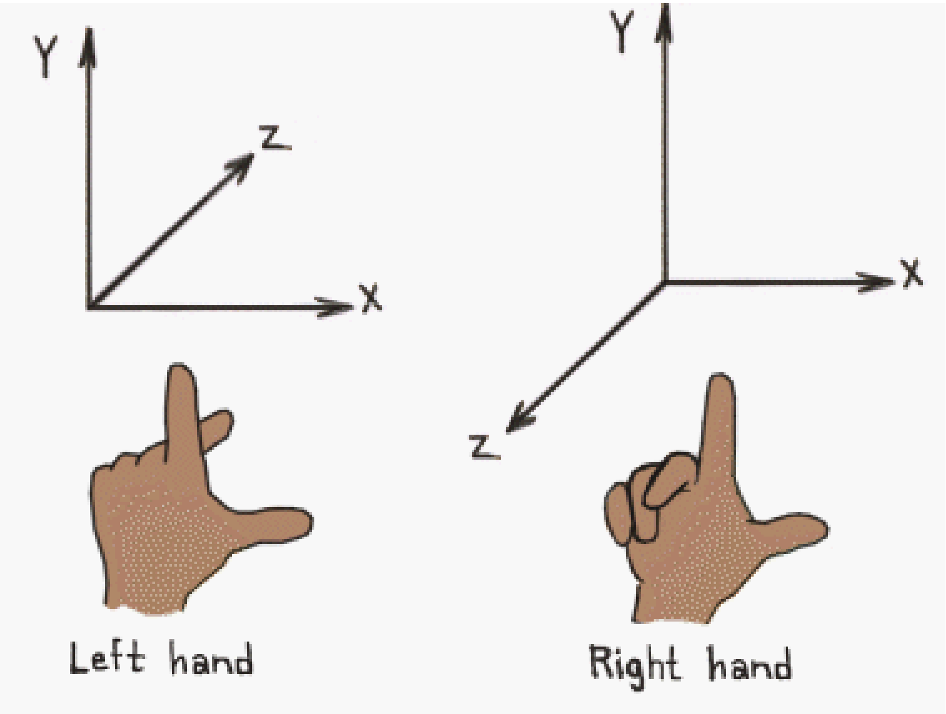
2) R(30)

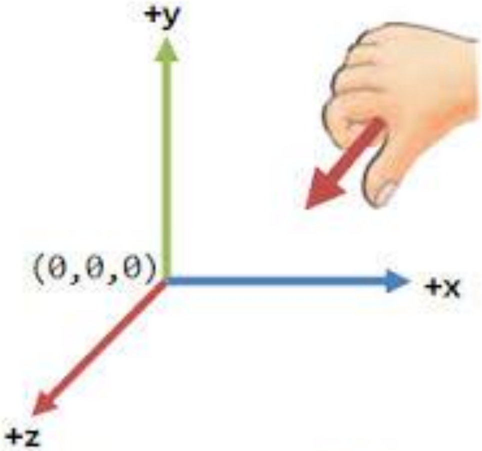
3) T(2, 3)

q'=T(2,3) R(30) T(-2,-3) q

3D Transformations

Right-handed or left-handed coordinate system

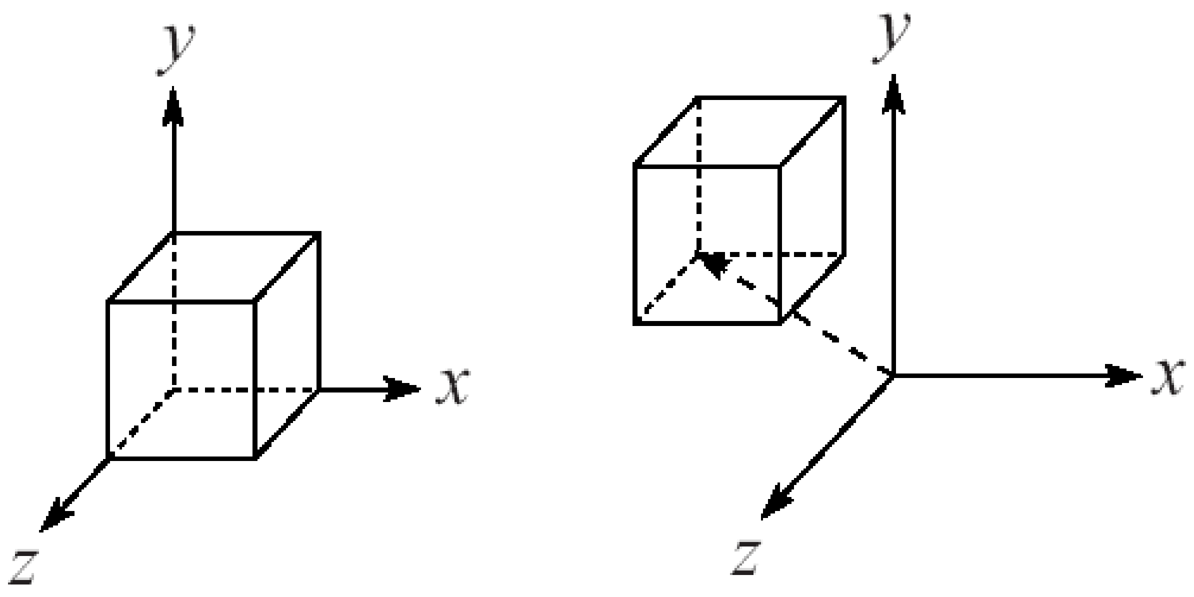




Right-Hand System (RHS) or
Counter-Clockwise (CCW) System

Translation in 3D

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Rotation in 3D

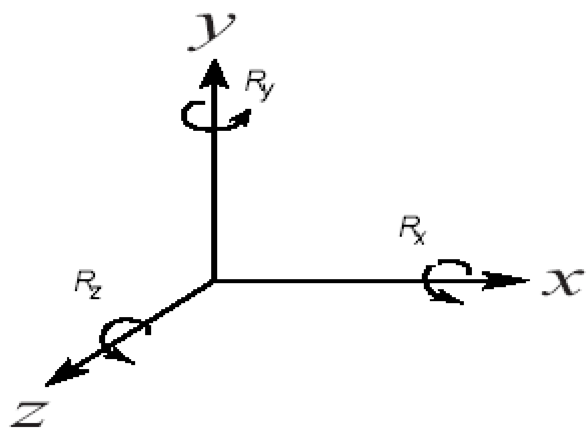
Rotation now has more possibilities in 3D:

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

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

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

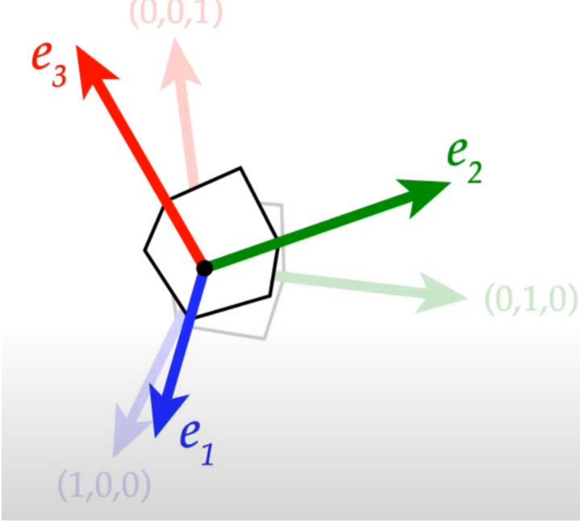
Rotation about x axis: apply rotation to y and z coordinates, x coordinate fixed



Use right hand rule

Rotations – transpose as inverse

Rotation will map standard basis to orthonormal basis e_1, e_2, e_3:



$$\begin{array}{cc} R^\top & R \\ \begin{bmatrix} - & e_1^T & - \\ - & e_2^T & - \\ - & e_3^T & - \end{bmatrix} & \begin{bmatrix} | & | & | \\ e_1 & e_2 & e_3 \\ | & | & | \end{bmatrix} \end{array}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Hence, $R^\top R = I$**, or equivalently,** $R^\top = R^{-1}$.

Reflections

- Q: Does <u>every</u> matrix $Q^\mathsf{T}Q = I$ describe a rotation?
  - Remember that rotations must preserve the <u>origin</u>, preserve <u>distances</u>, and preserve <u>orientation</u>
  - Consider for instance this matrix:

$$Q = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q^\mathsf{T}Q = \begin{bmatrix} (-1)^2 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Q: Does this matrix represent a rotation?

(If not, which invariant does it fail to preserve?)

A: No! It represents a reflection across the y-axis

(and hence fails to preserve <u>orientation</u>)

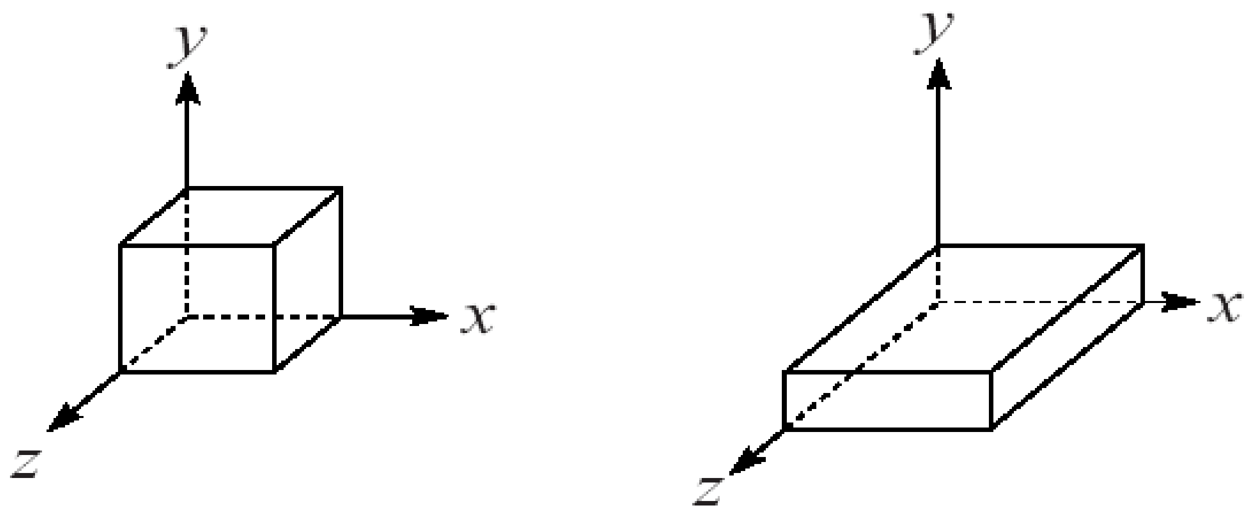
Orthogonal Transformations

- When $Q^{\mathsf{T}}Q = I$, the transformation is called an **orthogonal transformation**
- They preserve distances and the origin, but not necessarily orientation
  - **Rotations additionally <u>preserve</u> orientation:** $\det(Q) > 0$
  - **Reflections <u>reverse</u> orientation:** $\det(Q) < 0$

Non-uniform scaling in 3D

Scale each axis by a different amount

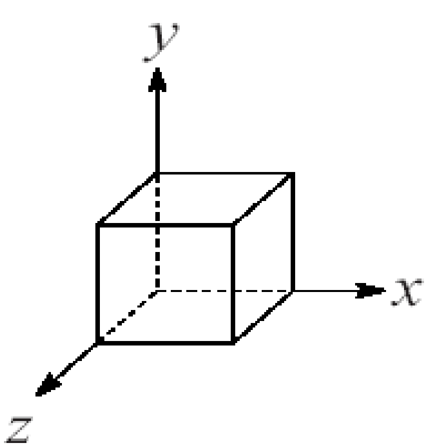
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

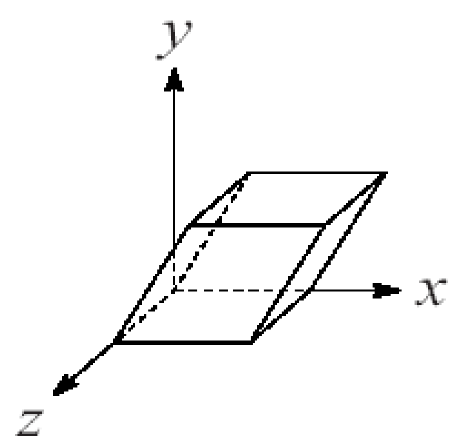


Shearing in 3D

- displaces the x-coordinate of each point according to its distance along y-axis

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





Decomposition of Linear Transformations

- In general, no **unique** way to write a given linear transformation as a composition of basic transformations!

- However, there are *many* useful decompositions:
  - singular value decomposition (good for signal processing)
  - LU factorization (good for solving linear systems)
  - polar decomposition (good for spatial transformations)
  - ...