CS 378: Computer Game Technology

3D Engines and Scene Graphs
Spring 2012

Representation

- We can represent a **point**, **p** = (x,y), in the plane
  - as a column vector $$\begin{bmatrix} x \\ y \end{bmatrix}$$
  - as a row vector $$\begin{bmatrix} x & y \end{bmatrix}$$

Representation, cont.

- We can represent a **2-D transformation M** by a matrix

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

- If **p** is a column vector, *M* goes on the left: $\mathbf{p}' = \mathbf{M}\mathbf{p}$

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

- If **p** is a row vector, M^T goes on the right: $\mathbf{p}' = \mathbf{p}\mathbf{M}^{\mathrm{T}}$

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

- We will use **column vectors**.

Two-dimensional transformations

- Here's all you get with a 2 x 2 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

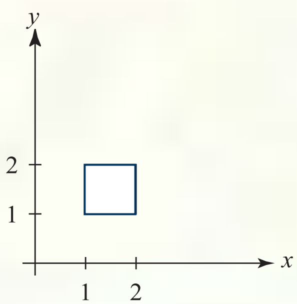
- Suppose we choose $a=d=1,\ b=c=0$:
  - Gives the **identity** matrix:

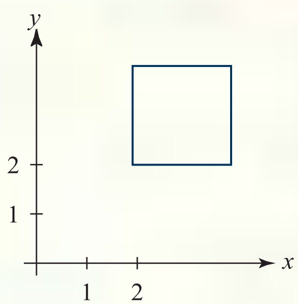
$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

  - Doesn't move the points at all

Scaling

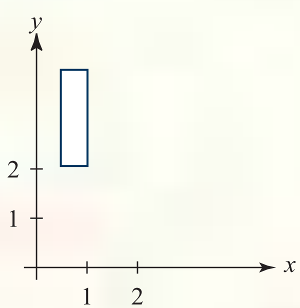
- Suppose *b=c=0*, but let *a* and *d* take on any *positive* value:
  - Gives a **scaling** matrix: $\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}$
  - Provides **differential (non-uniform) scaling** in *x* and *y*:





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

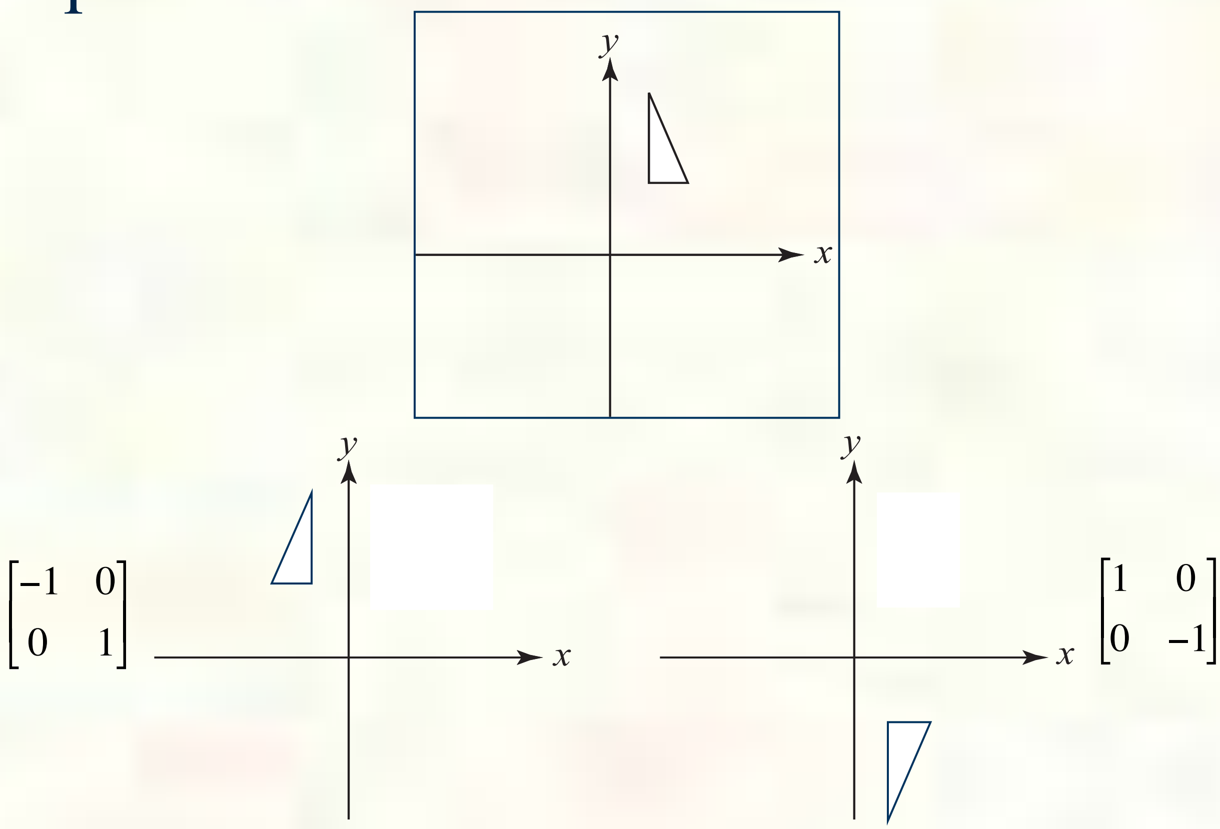
$$x' = ax$$
$$y' = dy$$



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

Reflection

- Suppose $b=c=0$, but let either a or d go negative.
- Examples:



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

Shear

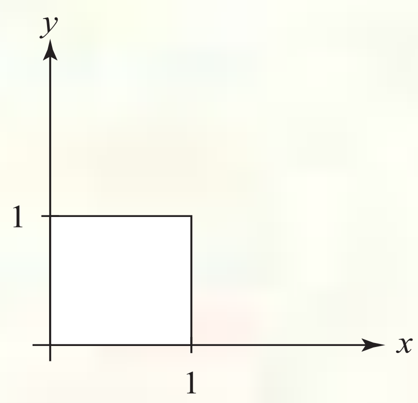
- Now leave *a=d=1* and experiment with *b*
- The matrix

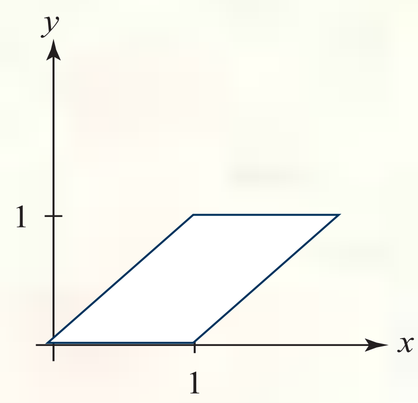
$$\begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}$$

gives:

$$x' = x + by$$

$$y' = y$$





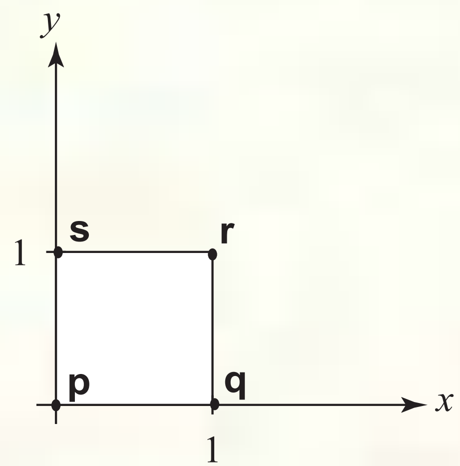
$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

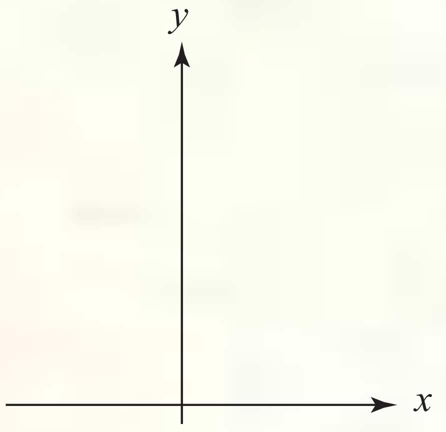
Effect on unit square

- Let's see how a general 2 x 2 transformation **M** affects the unit square:

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

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



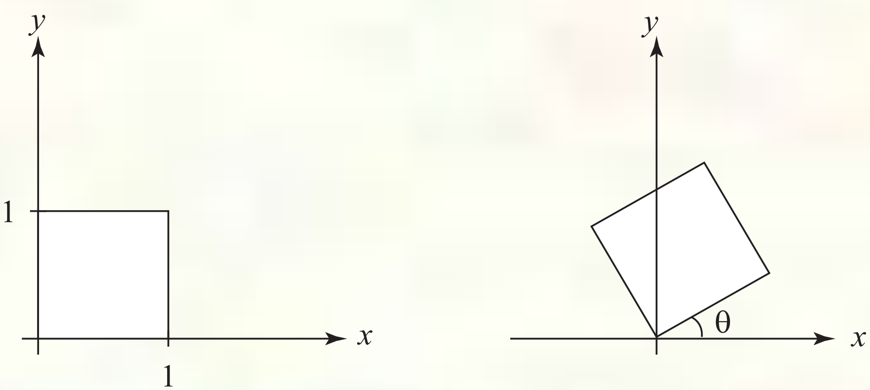


Effect on unit square, cont.

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

Rotation

- From our observations of the effect on the unit square, it should be easy to write down a matrix for "rotation about the origin":



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

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

Thus

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

Linear transformations

- The unit square observations also tell us the 2x2 matrix transformation implies that we are representing a point in a new coordinate system:

$$\begin{aligned}\mathbf{p}' &= \mathbf{M}\mathbf{p} \\ &= \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= x \cdot \mathbf{u} + y \cdot \mathbf{v}\end{aligned}$$

- where $\mathbf{u}=[a\ c]^T$ and $\mathbf{v}=[b\ d]^T$ are vectors that define a new **basis** for a **linear space**.
- The transformation to this new basis (a.k.a., change of basis) is a **linear transformation**.

Limitations of the 2 x 2 matrix

- A 2 x 2 linear transformation matrix allows
  - Scaling
  - Rotation
  - Reflection
  - Shearing
- **Q**: What important operation does that leave out?

Affine transformations

- In order to incorporate the idea that both the basis and the origin can change, we augment the linear space **u**, **v** with an origin **t**.
- Note that while **u** and **v** are **basis vectors**, the origin **t** is a **point**.
- We call **u**, **v**, and **t** (basis and origin) a **frame** for an **affine space**.
- Then, we can represent a change of frame as:

$$\mathbf{p}' = x \cdot \mathbf{u} + y \cdot \mathbf{v} + \mathbf{t}$$

- This change of frame is also known as an **affine transformation**.
- How do we write an affine transformation with matrices?

Homogeneous Coordinates

- To represent transformations among affine frames, we can loft the problem up into 3-space, adding a third component to every point:

$$
\begin{aligned}
\mathbf{p}' &= \mathbf{M}\mathbf{p} \\
&= \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{t} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= x \cdot \mathbf{u} + y \cdot \mathbf{v} + 1 \cdot \mathbf{t}
\end{aligned}
$$

- Note that $[a\ c\ 0]^T$ and $[b\ d\ 0]^T$ represent vectors and $[t_x\ t_y\ 1]^T$, $[x\ y\ 1]^T$ and $[x'\ y'\ 1]^T$ represent points.

Homogeneous coordinates

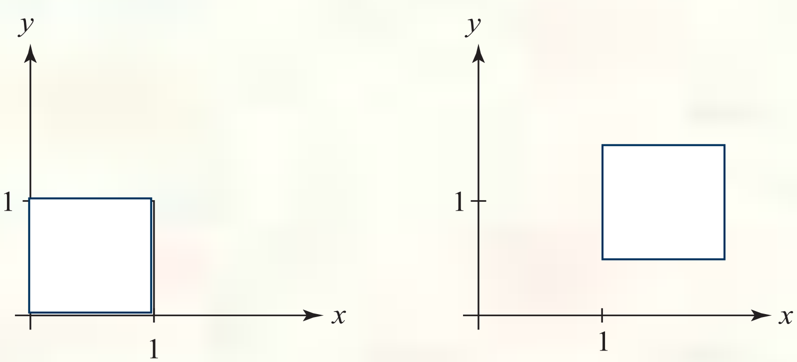
This allows us to perform translation as well as the linear transformations as a matrix operation:

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

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

$$x' = x + t_x$$

$$y' = y + t_y$$

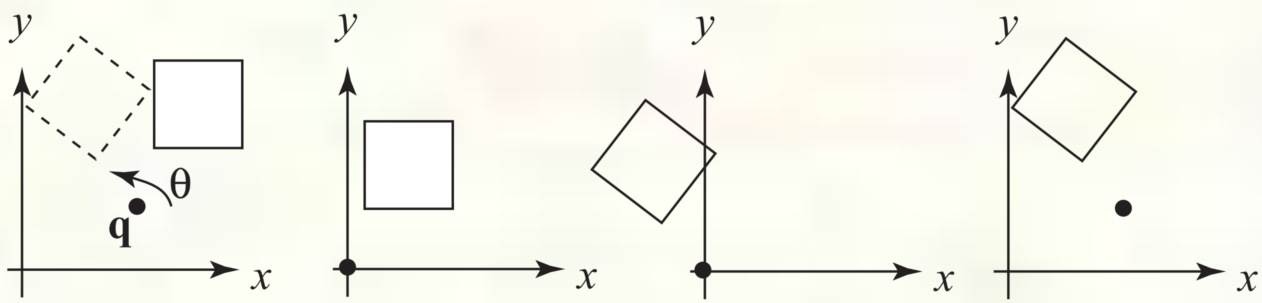


$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1/2 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation about arbitrary points

Until now, we have only considered rotation about the origin.

With homogeneous coordinates, you can specify a rotation, $\mathbf{R}_\mathbf{q}$, about any point $\mathbf{q} = [q_x\ q_y\ 1]^T$ with a matrix:



1. Translate **q** to origin
2. Rotate
3. Translate back

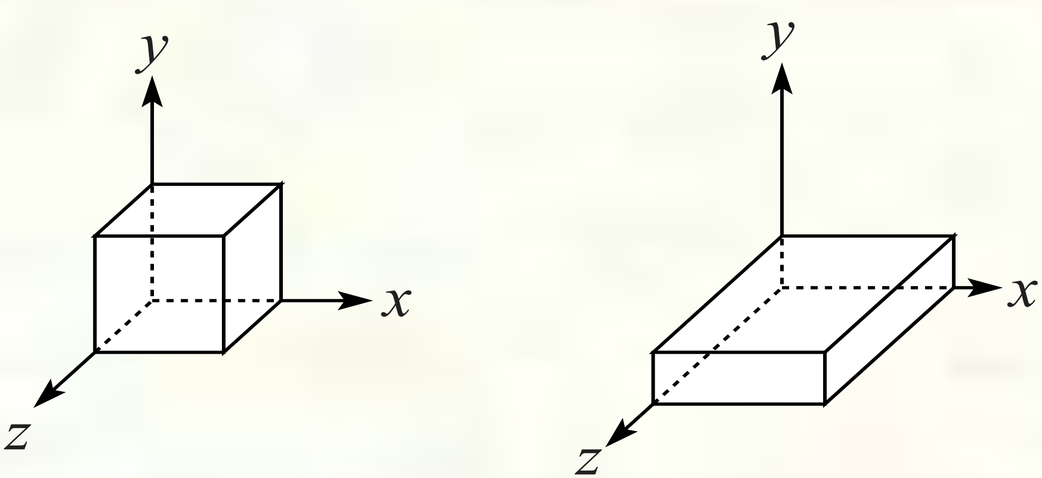
Line up the matrices for these step in right to left order and multiply.

Note: Transformation order is important!!

Basic 3-D transformations: scaling

Some of the 3-D transformations are just like the 2-D ones.

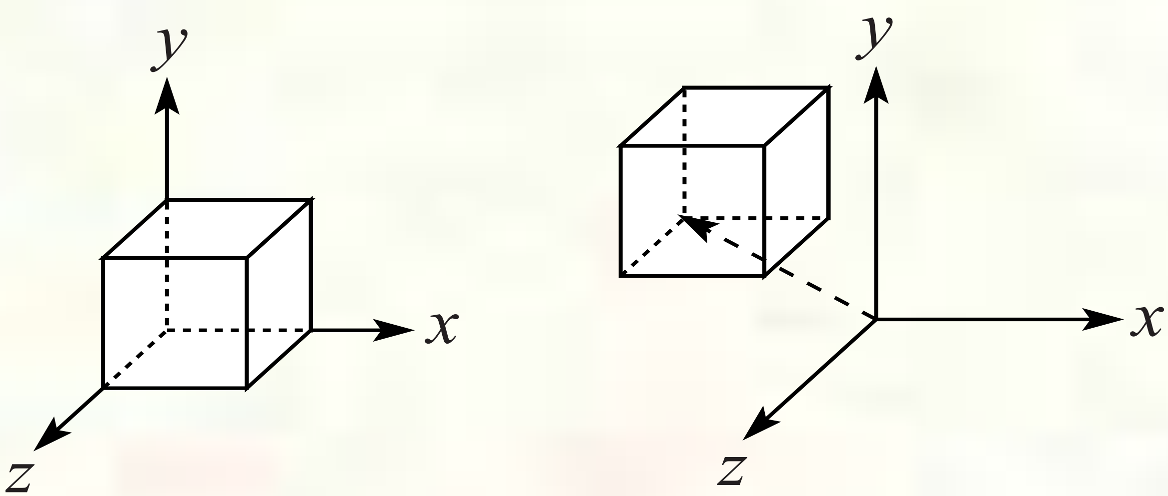
For example, scaling:



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

Translation in 3D

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



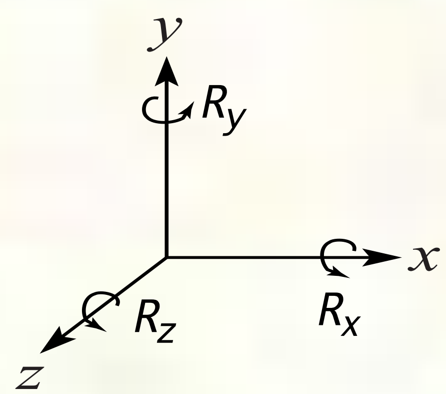
Rotation in 3D

Rotation now has more possibilities in 3D:

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

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

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

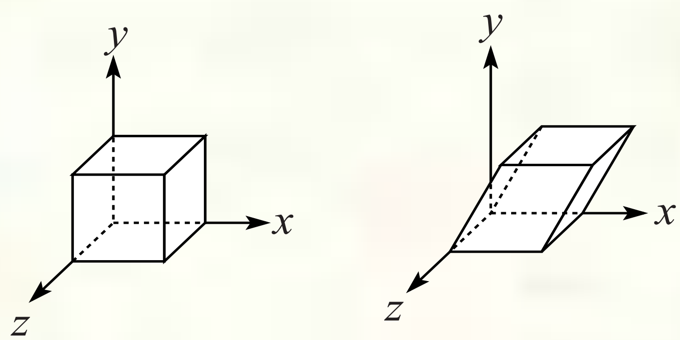


Use right hand rule

Shearing in 3D

- Shearing is also more complicated. Here is one example:

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



- We call this a shear with respect to the x-z plane.