

Affine transformations

**Brian Curless
CSE 457
Spring 2015**

Reading

Required:

- ♦ Angel 3.1, 3.7-3.11

Further reading:

- ♦ Angel, the rest of Chapter 3
- ♦ Foley, et al, Chapter 5.1-5.5.
- ♦ David F. Rogers and J. Alan Adams, *Mathematical Elements for Computer Graphics*, 2nd Ed., McGraw-Hill, New York, 1990, Chapter 2.

Geometric transformations

Geometric transformations will map points in one space to points in another: $(x', y', z') = \mathbf{f}(x, y, z)$.

These transformations can be very simple, such as scaling each coordinate, or complex, such as non-linear twists and bends.

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

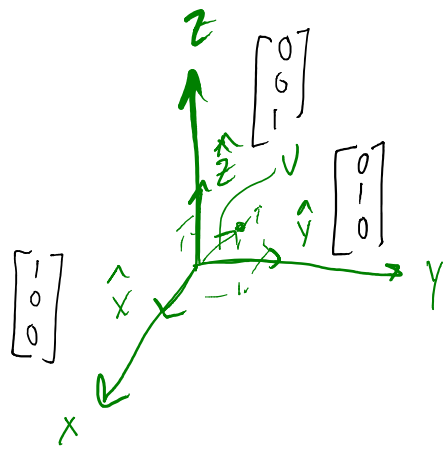
Vector representation

We can represent a **point**, $\mathbf{p} = (x, y)$, in the plane or $\mathbf{p} = (x, y, z)$ in 3D space

◆ as column vectors $\begin{bmatrix} x \\ y \end{bmatrix}$ $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$

~~◆ as row vectors $\begin{bmatrix} x & y \end{bmatrix}$
 $\begin{bmatrix} x & y & z \end{bmatrix}$~~

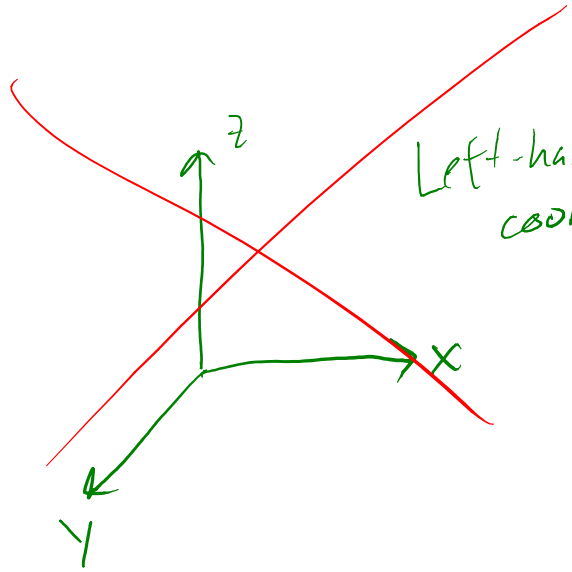
Canonical axes



Right-handed
coord. system

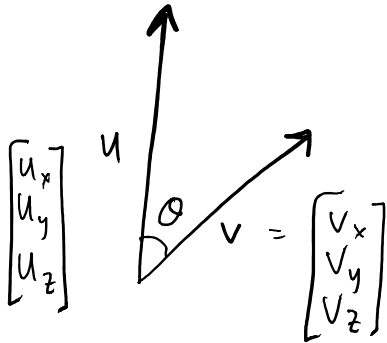
$$\|\hat{x}\| = \|\hat{y}\| = \|\hat{z}\| = 1$$

$$v = v_x \hat{x} + v_y \hat{y} + v_z \hat{z}$$



Left-handed
coord. sys.

Vector length and dot products $v^T = [v_x \ v_y \ v_z]$



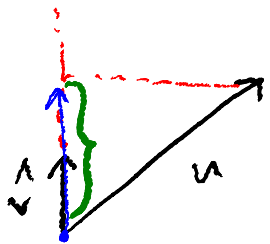
$$\|v\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$$

$$u \cdot v \equiv u_x v_x + u_y v_y + u_z v_z \quad \leftarrow \text{scalar}$$

$$u \cdot v = v \cdot u? \quad \text{Yes}$$

$$u^T v = [u_x \ u_y \ u_z] \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = u \cdot v$$

$$(u \cdot \hat{v}) \hat{v}$$



$$u \cdot \hat{v} = \|u\| \cos \theta$$

$$u \cdot v = \|u\| \|v\| \cos \theta$$

When is $u \cdot v = 0$?

$$\rightarrow u \perp v$$

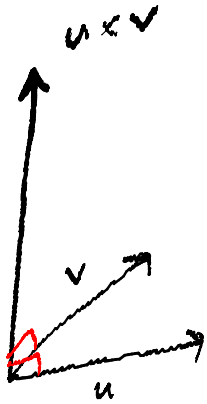
$$\rightarrow \|u\| \text{ or } \|v\| = 0$$

$$\frac{u}{\|u\|} = \hat{u}$$

$$\|\hat{u}\| = 1$$

$$\hat{u} \cdot \hat{v} = \cos \theta$$

Vector cross products



$$u \times v \equiv \det \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \\ u_x & u_y & u_z \\ v_x & v_y & v_z \end{bmatrix}$$

$$= \hat{x} (u_y v_z - u_z v_y)$$

$$- \hat{y} (u_x v_z - u_z v_x)$$

$$+ \hat{z} (u_x v_y - u_y v_x)$$

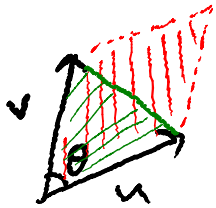
$$= \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

$$\hat{x} \times \hat{y} = \hat{z}$$

$$(u \times v) \cdot u = 0$$

$$(u \times v) \cdot v = 0$$

$$u \times v = -v \times u$$



$$\|u \times v\| = \|u\| \|v\| \sin \theta$$

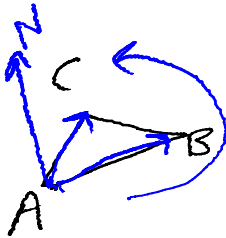
$$= \text{area}(\text{parallelogram}_{u,v})$$

$$\text{area}(\Delta_{u,v}) = \frac{\|u \times v\|}{2}$$



$$u = \alpha v$$

$$u \times v = 0$$



$$N(\Delta_{ABC}) \propto (B-A) \times (C-A)$$

$$(AB)^T = B^T A^T$$

$$M^{-1}M = I$$

$$(AB)^{-1}AB = I$$

$$(AB)^{-1}AB B^{-1} = \cancel{B} B^{-1}$$

$$(AB)^{-1}AA^{-1} = B^{-1}A^{-1}$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

Representation, cont.

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

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

If \mathbf{p} is a column vector, M goes on the left:

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

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$

If \mathbf{p} is a row vector, M^T goes on the right:

$$\mathbf{p}' = \mathbf{p}M^T$$

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

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

$$(\mathbf{p}')^T = (M\mathbf{p})^T$$

$$(\mathbf{p}')^T = \mathbf{p}^T M^T$$

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 $a, b, c, d \dots$

Identity

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

- ◆ Gives the **identity** matrix:

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

- ◆ Doesn't move the points at all

Scaling

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

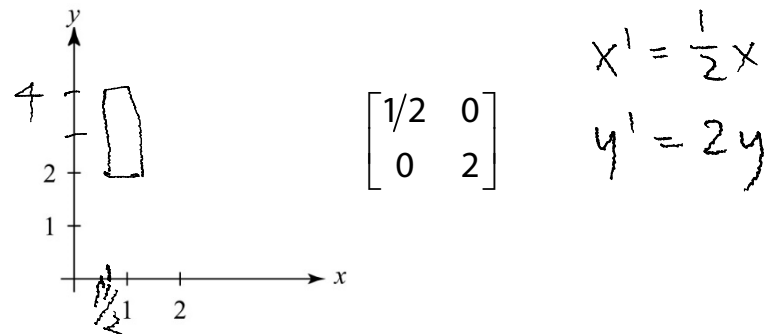
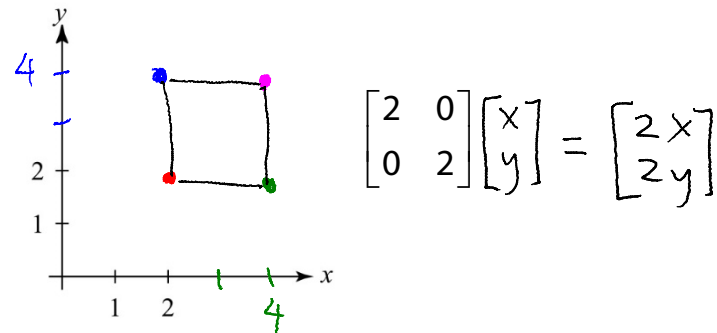
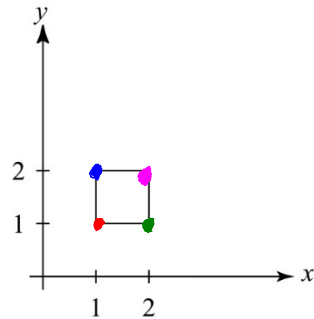
- ◆ Gives a **scaling** matrix:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax \\ dy \end{bmatrix}$$

- ◆ Provides **differential (non-uniform) scaling** in x and y :

$$x' = ax$$

$$y' = dy$$

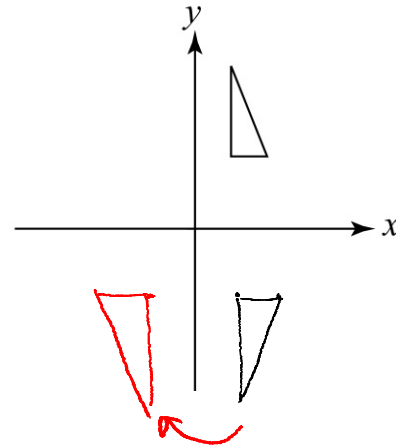
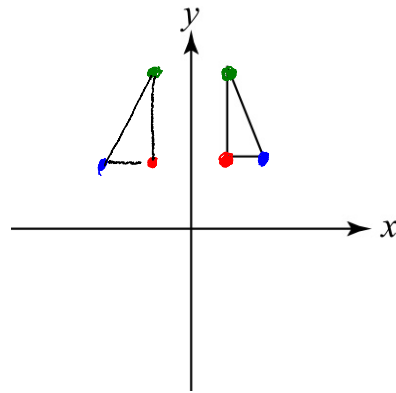


Mirror / Reflection

Suppose we keep $b=c=0$, but let either a or d go negative.

Examples:

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



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

Shear

Now let's leave $a=d=1$ and experiment with $b \dots$

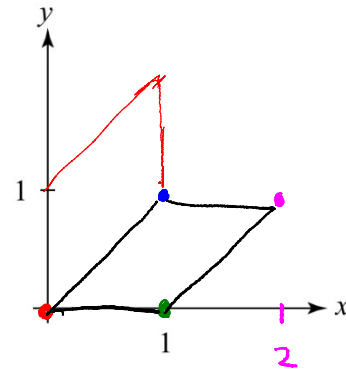
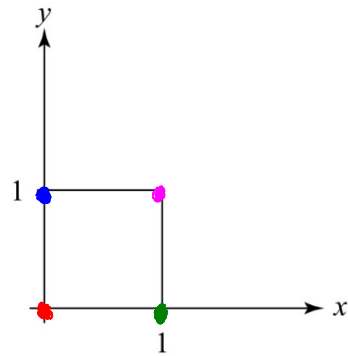
The matrix

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

gives:

$$x' = x + by$$

$$y' = y$$



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

$$x' = x + y$$

$$y' = y$$

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

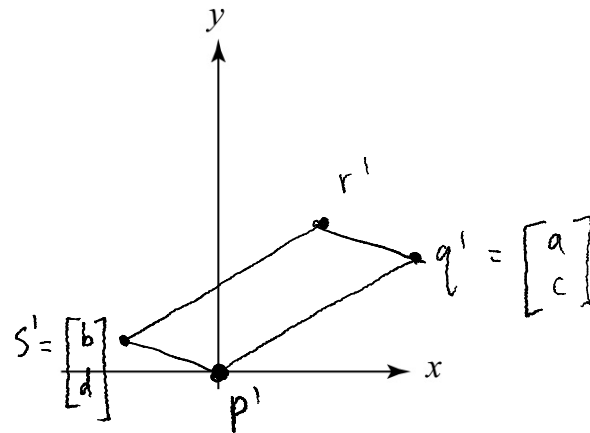
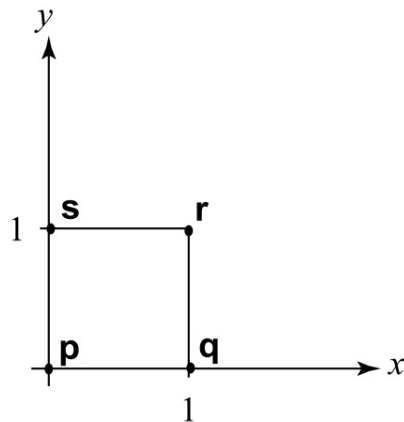
Effect on unit square

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

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

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

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



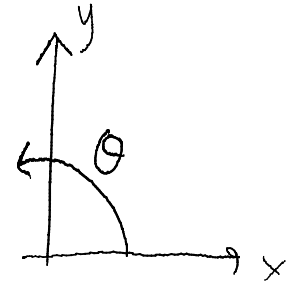
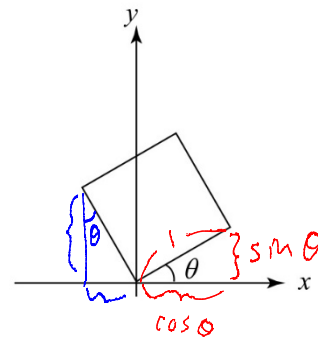
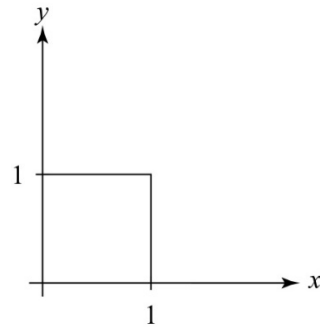
Effect on unit square, cont.

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,

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

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?

Translation

Homogeneous coordinates

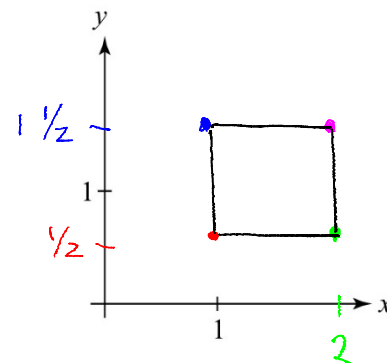
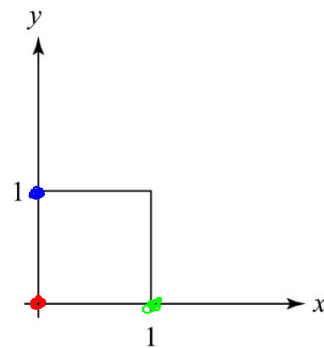
Idea is to lift the problem up into 3-space, adding a third component to every point:

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

Adding the third “w” component puts us in **homogenous coordinates**.

And then transform with a 3 x 3 matrix:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = T(\mathbf{t}) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + 1 \\ y + 1/2 \\ 1 \end{bmatrix}$$

... gives **translation!**

Anatomy of an affine matrix

The addition of translation to linear transformations gives us **affine transformations**.

In matrix form, 2D affine transformations always look like this:

$$M = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} = \left[\begin{array}{cc|c} \mathbf{A} & & \mathbf{t} \\ \hline 0 & 0 & 1 \end{array} \right]$$

2D affine transformations always have a bottom row of [0 0 1].

An "affine point" is a "linear point" with an added w-coordinate which is always 1:

$$\mathbf{p}_{\text{aff}} = \begin{bmatrix} \mathbf{p}_{\text{lin}} \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \end{bmatrix}$$

Applying an affine transformation gives another affine point:

$$M\mathbf{p}_{\text{aff}} = \begin{bmatrix} \mathbf{A}\mathbf{p}_{\text{lin}} + \mathbf{t} \\ 1 \end{bmatrix}$$

$$M \quad \mathbf{p}_{\text{aff}} \\ \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} ax + by + t_x \\ cx + dy + t_y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} ax + by \\ cx + dy \\ 1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ 0 \end{bmatrix}$$

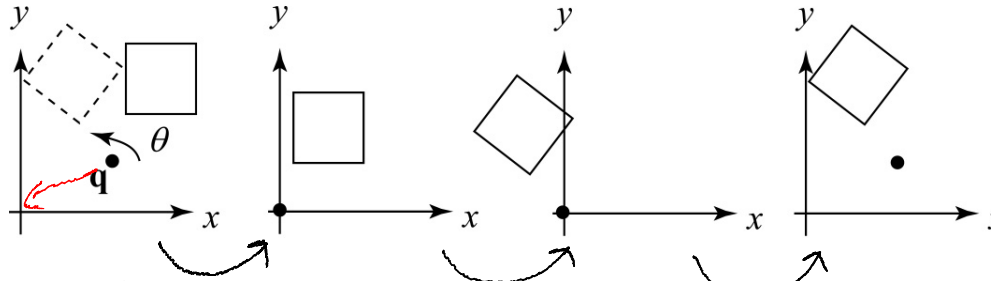
$$= \begin{bmatrix} \mathbf{A} \begin{bmatrix} x \\ y \end{bmatrix} \\ 1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ 0 \end{bmatrix}$$

Rotation about arbitrary points

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

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

$T(\cdot)$ = translation
 $R(\cdot)$ = rotation.



$$M_p \neq T(-q) \circ R(\theta) \cdot T(q) \cdot p$$

1. Translate \mathbf{q} to origin
2. Rotate
3. Translate back

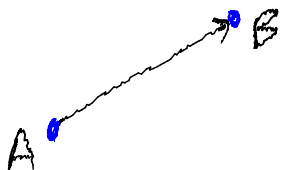
$$M = T(q)R(\theta)T(-q)$$

Note: Transformation order is important!!

Points and vectors

Vectors have an additional coordinate of $w = 0$. Thus, a change of origin has no effect on vectors.

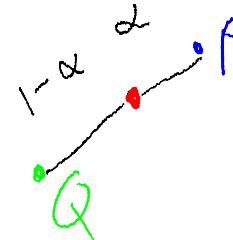
$$\begin{aligned} & \sum m_i P_i \\ & \sum m_i \\ & \sum \frac{m_i}{\sum m_i} P_i \end{aligned}$$



$$\begin{bmatrix} B_x \\ B_y \\ 1 \end{bmatrix} - \begin{bmatrix} A_x \\ A_y \\ 1 \end{bmatrix} = \begin{bmatrix} B_x - A_x \\ B_y - A_y \\ 0 \end{bmatrix}$$

Q: What happens if we multiply a vector by an affine matrix?

$$\begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} = \begin{bmatrix} av_x + bv_y \\ cv_x + dv_y \\ 0 \end{bmatrix}$$



These representations reflect some of the rules of affine operations on points and vectors:

- vector + vector → vector
- scalar · vector → vector
- point - point → vector
- point + vector → point
- point + point → chaos!

$$aP + bQ = \text{point} \quad \text{if } a + b = 1$$

$$aP + bQ = \text{vector} \quad \text{if } a + b = 0$$

One useful combination of affine operations is:

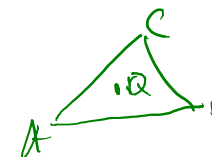
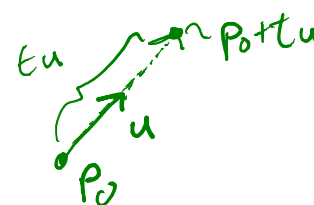
$$\mathbf{p}(t) = \mathbf{p}_0 + t\mathbf{u}$$

/ Point \quad \backslash vector

Q: What does this describe?

$$t \in [-\infty, \infty] \Rightarrow \text{line}$$

$$t \in [0, \infty] \Rightarrow \text{half-line, ray}$$

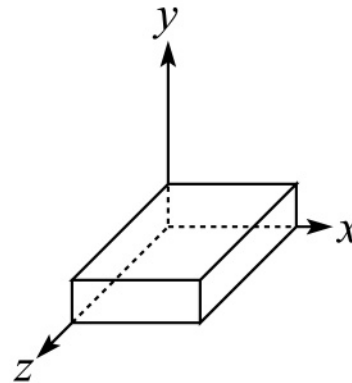
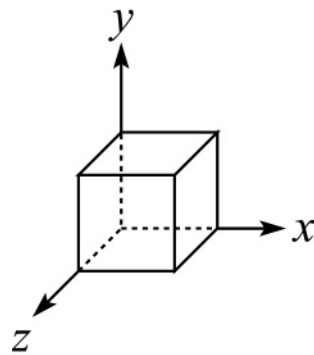


Basic 3-D transformations: scaling

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

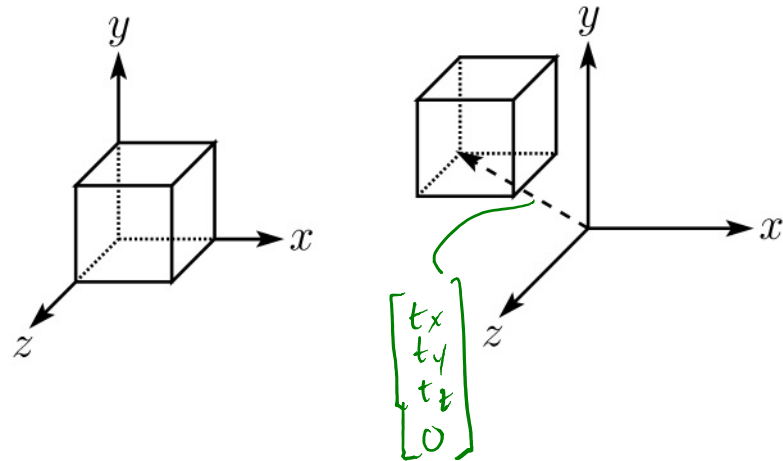
For example, scaling:

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



Translation in 3D

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



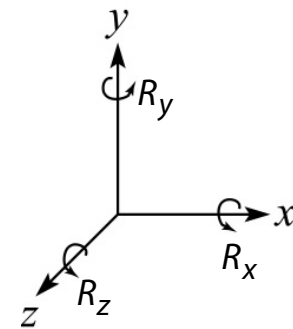
Rotation in 3D (cont'd)

These are the rotations about the canonical axes:

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

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

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



Use right hand rule

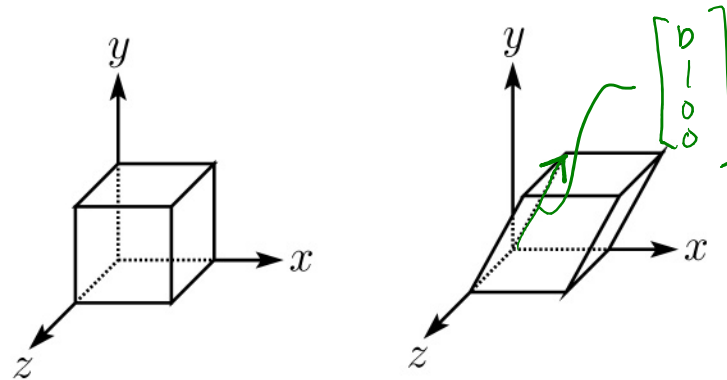
A general rotation can be specified in terms of a product of these three matrices. How else might you specify a rotation?

Quaternion \approx rotation about a direction
amount of rotation $-\theta$
direction $-\hat{v}$ (normalized)

Shearing in 3D

Shearing is also more complicated. Here is one example:

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

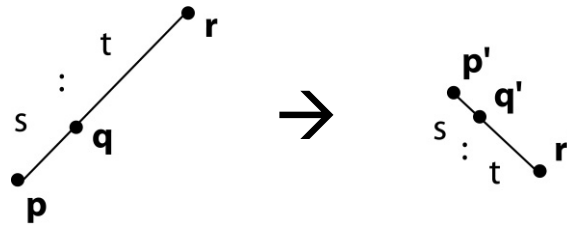
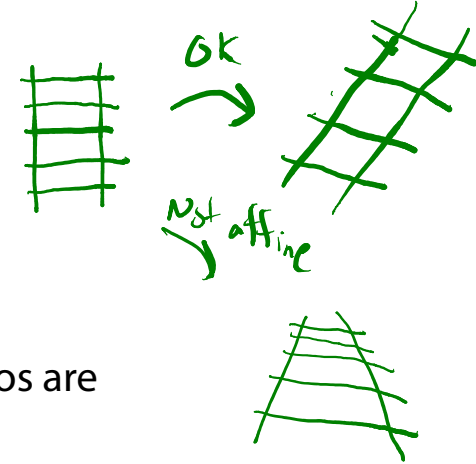


We call this a shear with respect to the x - z plane.

Properties of affine transformations

Here are some useful properties of affine transformations:

- ◆ Lines map to lines
- ◆ Parallel lines remain parallel
- ◆ Midpoints map to midpoints (in fact, ratios are always preserved)



$$\text{ratio} = \frac{\|pq\|}{\|qr\|} = \frac{s}{t} = \frac{\|p'q'\|}{\|q'r'\|}$$

Affine transformations in OpenGL

OpenGL maintains a “modelview” matrix that holds the current transformation **M**.

The modelview matrix is applied to points (usually vertices of polygons) before drawing.

It is modified by commands including:

- ◆ `glLoadIdentity()` **M** ← **I**
– set **M** to identity
- ◆ `glTranslatef(tx, ty, tz)` **M** ← **MT**
– translate by (*t_x*, *t_y*, *t_z*)
- ◆ `glRotatef(θ, x, y, z)` **M** ← **MR**
– rotate by angle *θ* about axis (*x*, *y*, *z*)
- ◆ `glScalef(sx, sy, sz)` **M** ← **MS**
– scale by (*s_x*, *s_y*, *s_z*)

Note that OpenGL adds transformations by *postmultiplication* of the modelview matrix.

Summary

What to take away from this lecture:

- ◆ All the names in boldface.
- ◆ How points and transformations are represented.
- ◆ How to compute lengths, dot products, and cross products of vectors, and what their geometrical meanings are.
- ◆ What all the elements of a 2×2 transformation matrix do and how these generalize to 3×3 transformations.
- ◆ What homogeneous coordinates are and how they work for affine transformations.
- ◆ How to concatenate transformations.
- ◆ The mathematical properties of affine transformations.