Affine Transformations

CSE 457

Winter 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: $(x', y', z') = \boldsymbol{f}(x, y, z)$.

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

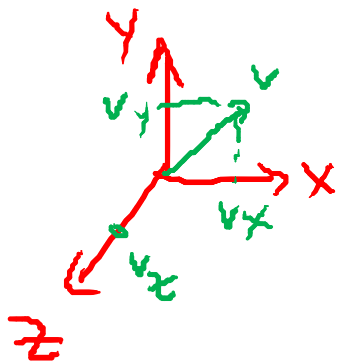
We can represent a **point**, **p** = (x,y), in the plane or **p**=(x,y,z) in 3D space

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

2D 3D

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

Canonical axes



right hand rule

$$v = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$$

$\hat{x}, \hat{y}, \hat{z}$ $\quad \hat{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \hat{y} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \hat{z} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

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

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} v$$

Vector length and dot products

$v = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$ $\|v\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$ $v^T = (v_x \; v_y \; v_z)$

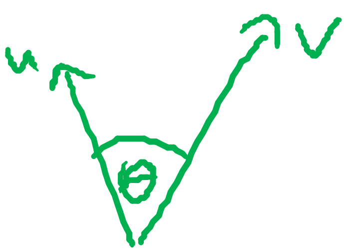
$\hat{v}$ - unit vector

$u = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$ $\hat{v} = \dfrac{v}{\|v\|}$ $\|\hat{v}\| = 1$

scalar

$u \cdot v = u_x v_x + u_y v_y + u_z v_z = u^T v$

$u \cdot u = \|u\|^2$



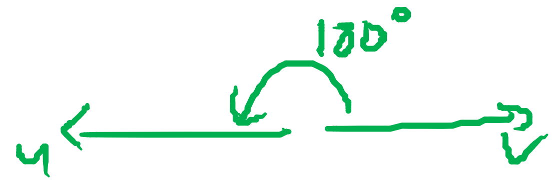
$u \cdot v = \|u\| \, \|v\| \cos\theta$ geometric

$\theta = 0^\circ \Rightarrow$ parallel

$\theta = 90^\circ \Rightarrow u \cdot v = 0$ $u \perp v$

$u \cdot v = v \cdot u$ yes.

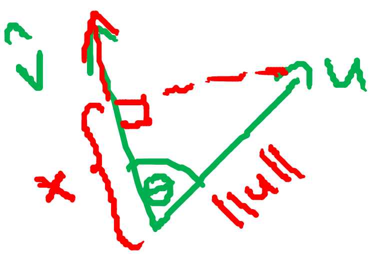
$\theta = 180°$ $\quad \hat{u} \cdot \hat{v} = -1$



---

projection of u onto $\hat{v}$

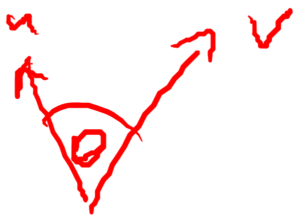
$(u \cdot \hat{v})\hat{v}$



$$\cos\theta = \frac{x}{\|u\|}$$

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

Vector cross products



$$\hat{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

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

$$= \hat{x} \begin{vmatrix} u_y & u_z \\ v_y & v_z \end{vmatrix} - \hat{y} \begin{vmatrix} u_x & u_z \\ v_x & v_z \end{vmatrix} + \hat{z} \begin{vmatrix} u_x & u_y \\ v_x & v_y \end{vmatrix}$$

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

$$\underline{\underline{u \times v = -v \times u}}$$

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

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

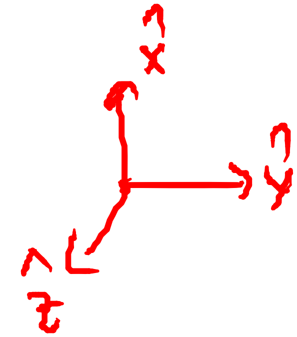
$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

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

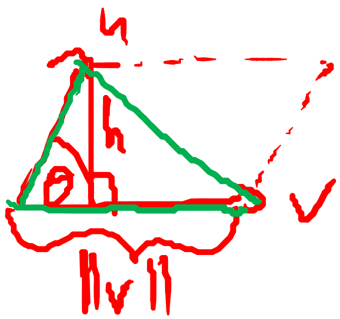
$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} \ \end{vmatrix} - b \begin{vmatrix} \ \end{vmatrix} + c \begin{vmatrix} \ \end{vmatrix}$$

$\hat{x}, \hat{y}, \hat{z}$

$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$



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



$$A_{\square} = \|u \times v\|$$

$$A_{\square} = h \cdot \|v\| = \|v\|\|u\| \sin\theta$$

$$A_{\triangle} = \frac{\|u \times v\|}{2}$$

Representation, cont.

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

$A^{-1} A = I$

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

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

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

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

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

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

assuming A, B invertible

If **p** is a column vector, *M* goes on the left:

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

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

$M^T = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$

If **p** is a row vector, M^T goes on the right:

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

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

$(x' \;\; y') = (x \;\; y)\, M^T$

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*…

Identity

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

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

- Gives the **identity** matrix:

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

- Doesn't move the points at all

$$x' = x$$
$$y' = y$$

Scaling

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

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

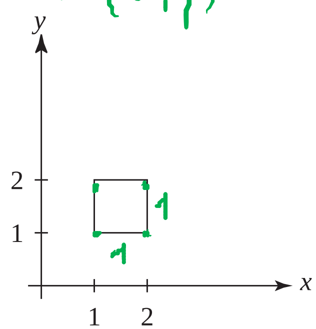
- Gives a **scaling** matrix:

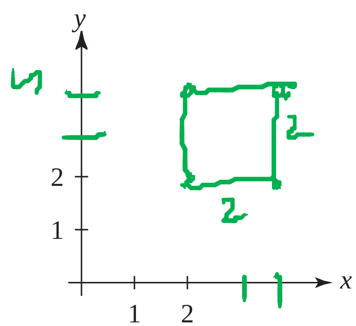
$$\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix}$$

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

$$\boxed{\begin{aligned} x' &= ax \\ y' &= dy \end{aligned}}$$

Input: (x, y)





M

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

$x' = 2x$

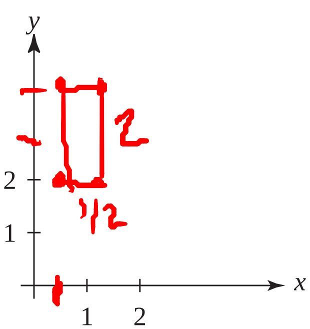
$y' = 2y$

$(1,1) \to (2,2)$

$(2,1) \to (4,2)$

$(1,2) \to (2,4)$

$(2,2) \to (4,4)$



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

$x' = \frac{1}{2}x$

$y' = 2y$

$(1,1) \to (\frac{1}{2}, 2)$

$(2,1) \to (1, 2)$

$(1,2) \to (\frac{1}{2}, 4)$

$(2,2) \to (1, 4)$

Reflection / Mirroring

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

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

Examples:

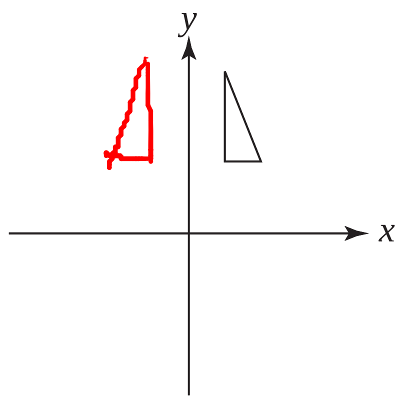
$$M_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

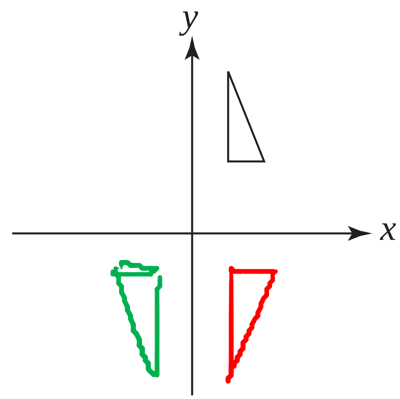
$x' = x$

$y' = -y$

$x' = -x$

$y' = y$





det Rot = 1

det Refl = −1

Rotation

$$= M_1 M_2 \begin{pmatrix} x \\ y \end{pmatrix}$$

shearing

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

Now let's leave *a*=*d*=*1* and experiment with *b*. . . .

The matrix

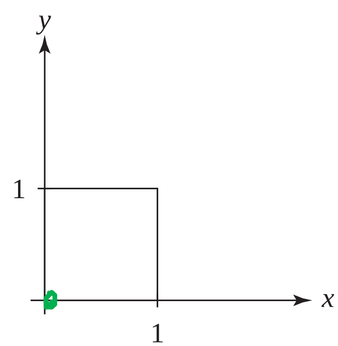
$$M = \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}$$

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

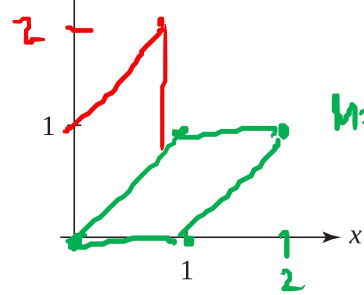
gives:

$$x' = x + by$$
$$y' = y$$

input



output



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

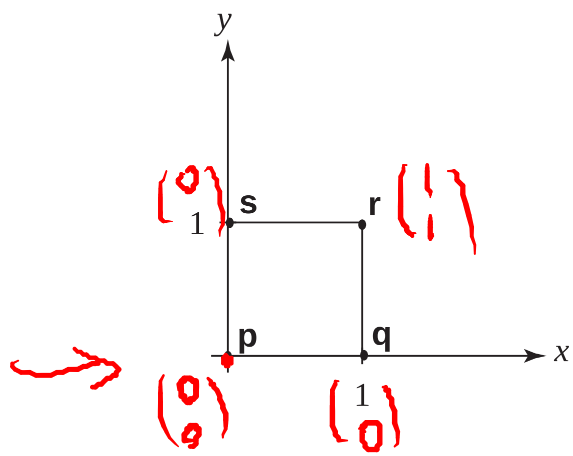
$(0,0) \to (0,0)$
$(1,0) \to (1,0)$
$(0,1) \to (1,1)$
$(1,1) \to (2,1)$

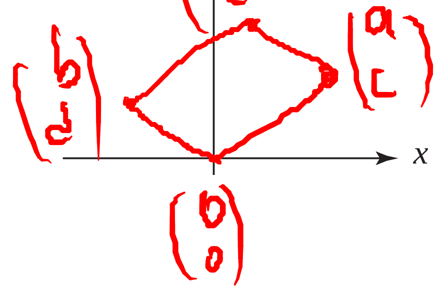
Effect on unit square

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

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

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





$$\begin{pmatrix}a+b\\c+d\end{pmatrix} = \begin{pmatrix}a\\c\end{pmatrix} + \begin{pmatrix}b\\d\end{pmatrix}$$

Effect on unit square, cont.

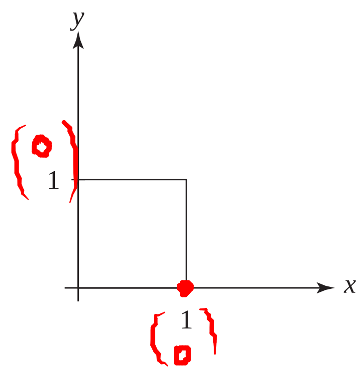
$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

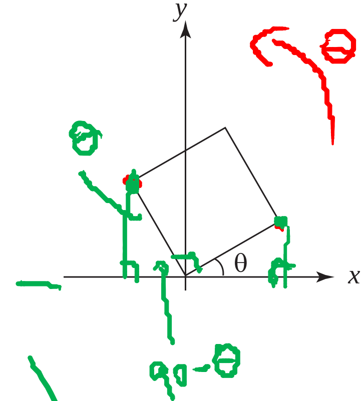
Observe:

- 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":





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

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

$$\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \rightarrow \begin{pmatrix} & \\ & \end{pmatrix}$$

Thus,

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

2D rotation by θ

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

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

2x2 translation

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ 0 \end{pmatrix}$$

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

And then transform with a 3 x 3 matrix:

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

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

. . . 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:

$$M = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{A} & \mathbf{t} \\ \hline 0 & 0 & 1 \end{array}\right]$$

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:

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

Applying an affine transformation gives another affine point:

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

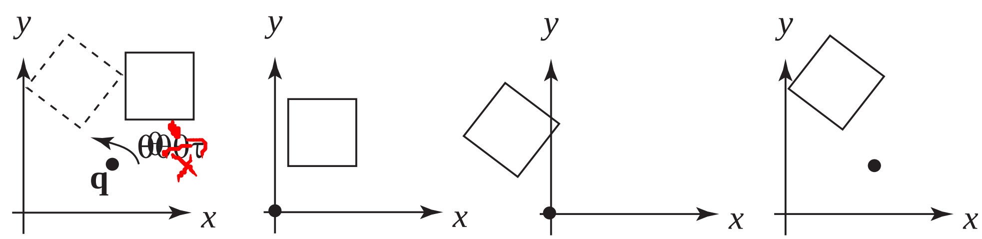
Rotation about arbitrary points

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

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

$$R(x-t)+t = x'$$

$$Rx - Rt + t = x'$$



$$\begin{pmatrix} R & -Rt+t \\ 0\quad 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

$$\underbrace{T(+q)\,R(\theta)\,T(-q)}_{M}\begin{pmatrix} x \\ y \end{pmatrix}$$

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

<u>Note</u>: Transformation order is important!!

$p = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \vec{v} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$

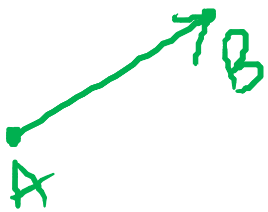
$p + p \longrightarrow$ chaos

$aP + bQ =$

$= \begin{pmatrix} \vdots \\ a+b \end{pmatrix}$

$\boxed{a+b=1}$

$$\begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \begin{pmatrix} ax+by \\ cx+dy \\ 0 \end{pmatrix}$$



$\vec{v} = B - A$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} - \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{pmatrix} = \begin{pmatrix} x - \tilde{x} \\ y - \tilde{y} \\ 0 \end{pmatrix}$$

$aP + bQ + cR$

$a + b + c = 1$

$v + v \longrightarrow$ vector

scalar $\cdot\, v \longrightarrow$ vector

$p - p \longrightarrow$ vector

$p + v \longrightarrow$ point

$\begin{pmatrix} \vdots \\ 1 \end{pmatrix} + \begin{pmatrix} \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \vdots \\ 1 \end{pmatrix}$

line $\quad p(t) = p_0 + t\vec{u} \quad t \in [-\infty, \infty]$

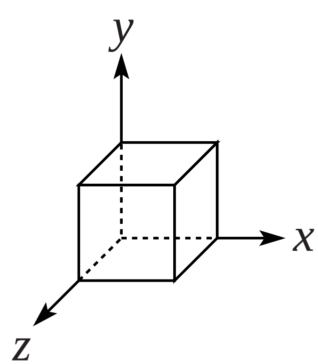
Ray: $\quad t \in (0, \infty)$

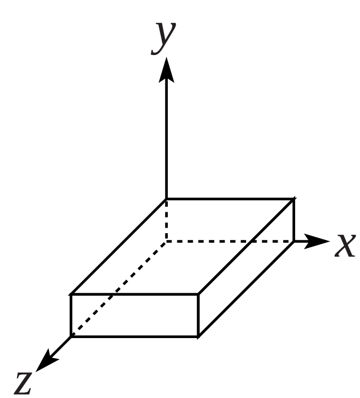
Basic 3-D transformations: scaling

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

For example, <u>scaling</u>:

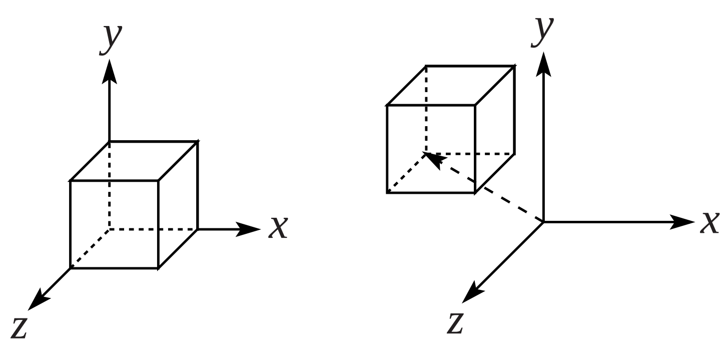
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$





Translation in 3D

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



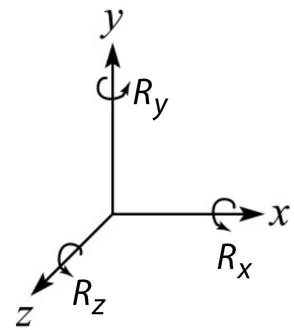
Rotation in 3D (cont'd)

These are the rotations about the canonical axes:

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

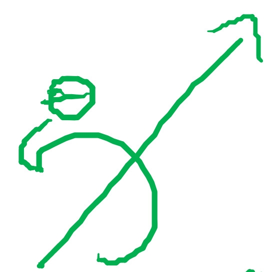
$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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



Use right hand rule

3D vector



only care about direction

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

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

$R = R_x(\alpha) R_y(\beta) R_z(\gamma)$

Shearing in 3D

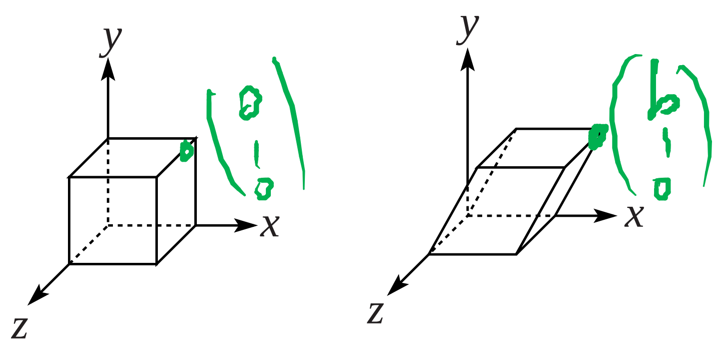
Shearing is also more complicated. Here is one example:

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

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

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

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \to \begin{pmatrix} b \\ 1 \\ 0 \end{pmatrix}$$



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

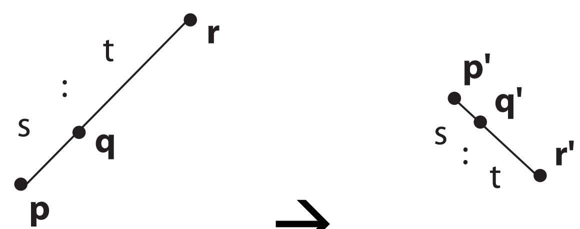
Properties of affine transformations

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)

$L_1: \; p_0 + t\vec{u}$

$L_2: \; p_1 + t\vec{u}$



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

$$A(p_0 + t\vec{u}) = Ap_0 + \underbrace{At\vec{u}}$$

$$A(p_1 + t\vec{u}) = Ap_1 + At\vec{u}$$

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()` — $\mathbf{M} \leftarrow \mathbf{I}$
  - set **M** to identity
- `glTranslatef(`t_x`, `t_y`, `t_z`)` — $\mathbf{M} \leftarrow \mathbf{MT}$
  - translate by (t_x, t_y, t_z)
- `glRotatef(θ, x, y, z)` — $\mathbf{M} \leftarrow \mathbf{MR}$
  - rotate by angle θ about axis (x, y, z)
- `glScalef(`s_x`, `s_y`, `s_z`)` — $\mathbf{M} \leftarrow \mathbf{MS}$
  - 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 x 2 transformation matrix do and how these generalize to 3 x 3 transformations.
- What homogeneous coordinates are and how they work for affine transformations.
- How to concatenate transformations.
- The mathematical properties of affine transformations.