Affine transformations

CSE 457
Winter 2014

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

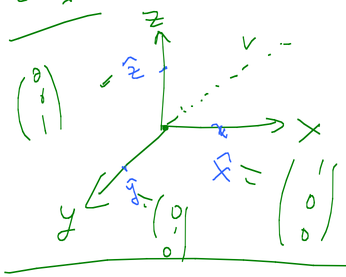
- as column vectors $\begin{bmatrix} x \\ y \end{bmatrix}$ $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$

  2D 3D

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

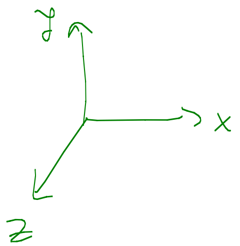
3D

Canonical axes



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

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



right hand rule

- coord. system
- rotations
- normal direction (triangles)

Vector length and dot products

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

$u = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$

$\|v\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$ — norm, magnitude

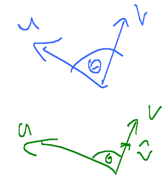
dot product $u \cdot v = u_x v_x + u_y v_y + u_z v_z = \boxed{v^T u}$

$= \sum_i u_i v_i$

$= [v_x \; v_y \; v_z] \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ row, column

$u \cdot v \overset{?}{=} v \cdot u$

$\|v\|^2 = v \cdot v$



$\boxed{u \cdot v = \|u\| \, \|v\| \cos\theta}$

$u \cdot v = 0 \Rightarrow \theta = 90°, -90°$ orthogonal

$\|u\| = 0$

$\|v\| = 0$

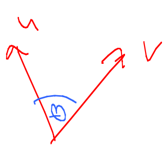
unit length

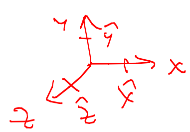
$\hat{v} = \frac{v}{\|v\|}$ $\quad \hat{u} = \frac{u}{\|u\|}$

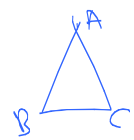
$\boxed{\hat{u} \cdot \hat{v} = \cos\theta}$

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

Vector cross products







$u = C - B$

$v = B - A$

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

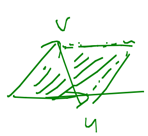
2x2

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

$= \begin{bmatrix}1\\0\\0\end{bmatrix} \underbrace{(u_y v_z - u_z v_y)} - \begin{bmatrix}0\\1\\0\end{bmatrix} (u_x v_z - v_x u_z) + \begin{bmatrix}0\\0\\1\end{bmatrix} (u_x v_y - v_x u_y)$

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

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



Area

$= \|u \times v\|$

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

Area

$= \frac{1}{2} \|u \times v\|$

$(AB)^T =$

$= B^T A^T$

$(AB)^{-1}$

~~$A^{-1}B^{-1}$~~ $\quad B^{-1}A^{-1}$

$A^{-1}A = I$

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

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

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

Representation, cont.

2x2

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

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

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

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

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

2x1 $\quad$ 2x2 $\quad$ 2x1

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

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

$$[x' \;\; y'] = [x \;\; y] \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}^T$$

1x2 $\quad$ 1x2 $\quad$ 2x2

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:

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

2×2

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

3×3

- Doesn't move the points at all

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

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

Scaling

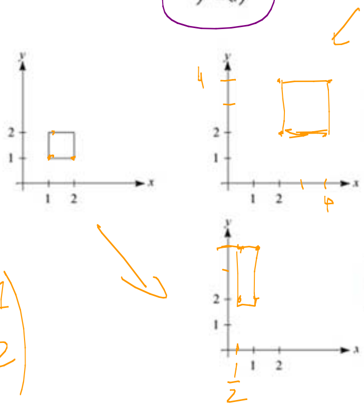
Suppose we set $b=c=0$, but let a and d take on any *positive* value:

- Gives a **scaling** matrix:

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

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

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



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

① uniform

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

② non-uniform

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

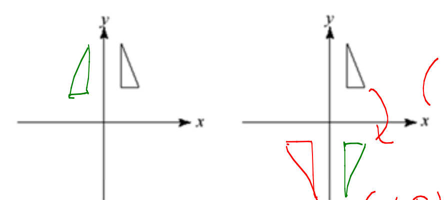
Reflection, Mirroring

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

Examples:

① $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ reflect across Y

② $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ reflect across X



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

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

rotation by 180°

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

Shear

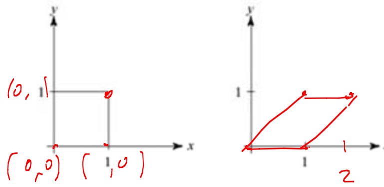
Now let's leave $a=d=1$ and experiment with b....

The matrix

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

gives:

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



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

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

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

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

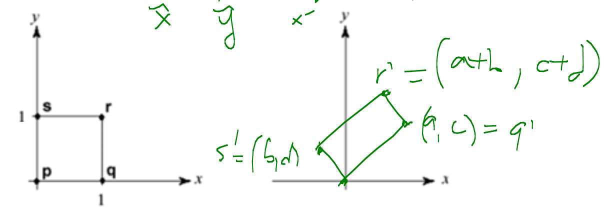
Effect on unit square

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

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

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

x y x-axis y-axis



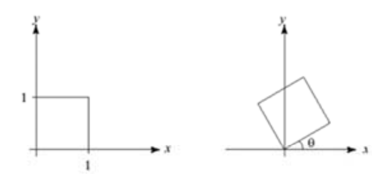
Effect on unit square, cont.

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":



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

1 dof

Thus,

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

2x2 rotation

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**.

We call **u**, **v**, and **t** (basis and origin) a **frame** for an **affine space**.

Then, we can represent a change of frame as:

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

2x1

This change of frame is also known as an **affine transformation**.

How do we write an affine transformation with matrices?

Homogeneous coordinates

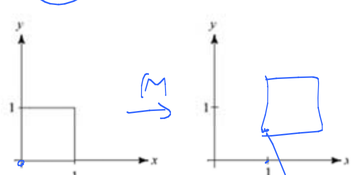
Idea is to loft the problem up into 3-space, adding a third component to every point:

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

And then transform with a 3 x 3 matrix:

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

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



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

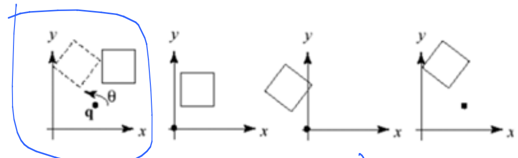
$\left(1, \frac{1}{2}\right)$

... gives **translation**!

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:



$M \neq T(-q) \cdot R(\theta) \cdot T(q) \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

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

$M \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

Note: Transformation order is important!!

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

Points and vectors

$P = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

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

origin-free

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

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

$$\begin{bmatrix} A_{2\times 2} & t_{2\times 1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} Av \\ 0 \end{bmatrix}$$

3×1

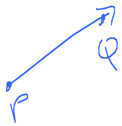
These representations reflect some of the rules of affine operations on points and vectors:

vector + vector	→	vector
~~scalar · vector~~	→	vector
point - point	→	vector
point + vector	→	point
point + point	→	nonsense

$\alpha P + \beta Q = R$

R point $\quad \alpha + \beta = 1$

R vector $\quad \alpha + \beta = 0$



$Q - P = \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} - \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}$

One useful combination of affine operations is:

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

Q: What does this describe?

line $\quad t \in (-\infty, \infty)$

half-line / ray $\quad t \in [0, \infty)$

$\alpha P - \alpha Q$

$\alpha (P - Q)$

$\sum \alpha_i P_i = R$

$\sum \alpha_i = 1$

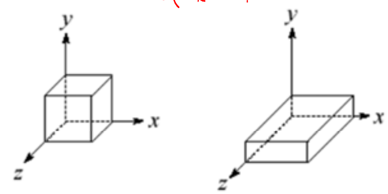
Basic 3-D transformations: scaling

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

For example, scaling:

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

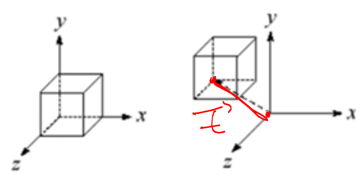
$4\times 4 \quad 4\times 1$



Translation in 3D

$3\times 3 \quad 3\times 1$

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



Rotation in 3D

$u^T v = 0 \quad v^T v = 1$

$u^T w = 0 \quad u^T u = 1$

$v^T w = 0 \quad w^T w = 1$

3 dof

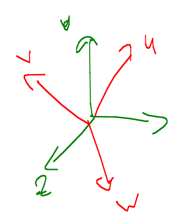
How many degrees of freedom are there in an arbitrary 3D rotation?

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

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

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

$R_{3D} = R_x \cdot R_y \cdot R_z$



$R = \begin{bmatrix} u & v & w \end{bmatrix}$

$R^T = \begin{bmatrix} u^T \\ v^T \\ w^T \end{bmatrix}$

$R^T R = \begin{bmatrix} u^T u & u^T v & u^T w \\ v^T u & v^T v & v^T w \\ w^T u & w^T v & w^T w \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

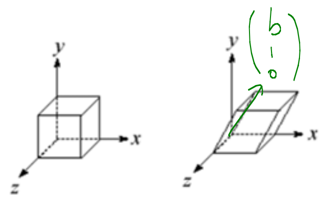
3 dof

$\boxed{R^T = R^{-1}}$

Shearing in 3D

Shearing is also more complicated. Here is one example:

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

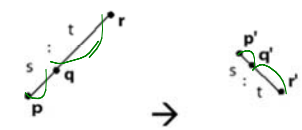


We call this a shear with respect to the x-z plane.

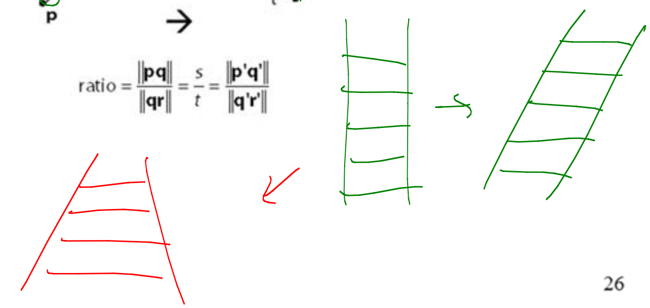
Properties of affine transformations

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)



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



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(`t_x, t_y, t_z`)` **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(`s_x, s_y, s_z`)` **M** ← **MS**
  - scale by (s_x, s_y, s_z)

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