Affine transformations

Reading

Required:

- Shirley, Sec. 2.4, 2.7
- Shirley, Ch. 5.1-5.3
- Shirley, Ch. 6

Further reading:

- 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 tranformations 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.

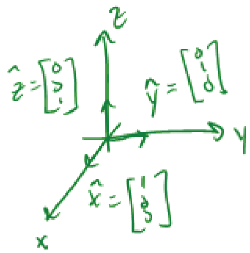
We'll start in 2D...

Representation

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

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

Canonical axes



right-handed coord. systems

Vector length and dot products

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

$= u_x v_x + u_y v_y + u_z v_z$

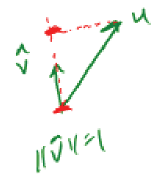
$\|v\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$

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

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

$u \cdot v = 0 \Rightarrow \perp$

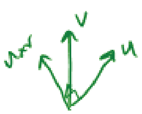
$\hat{u} = \frac{u}{\|u\|} \Rightarrow \|\hat{u}\| = 1$ $u = \underbrace{\|u\|}_{\text{length}} \underbrace{\hat{u}}_{\text{direction}}$



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

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

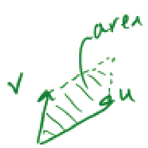
Vector cross products



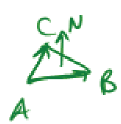
$u \times v = \begin{vmatrix}\hat{x} & \hat{y} & \hat{z}\\ u_x & u_y & u_z\\ v_x & v_y & v_z\end{vmatrix} = (u_y v_z - u_z v_y)\hat{x}$
$+ (u_z v_x - u_x v_z)\hat{y}$
$+ (u_x v_y - u_y v_x)\hat{z}$

$(u \times v) = -(v \times u)$

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



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



$N \sim (B - A) \times (C - A)$

Representation, cont.

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:

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

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

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

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

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

We will use **column vectors**.

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

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

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

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

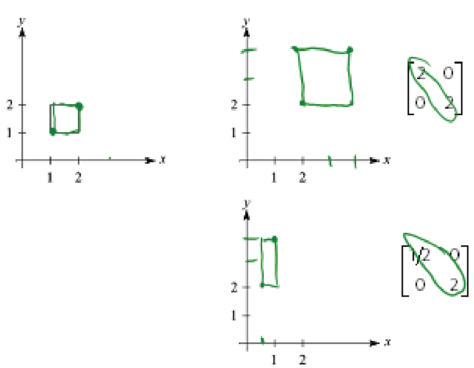
Suppose we set $b=c=0$, but let a and d take on any *positive* value:

- Gives a **scaling** matrix:

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

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

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



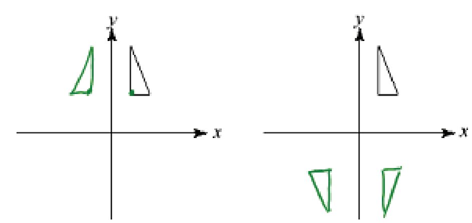
Reflection

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

Examples:

rotation by 180°

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



Shear

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

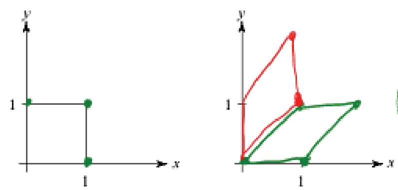
The matrix

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

gives:

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

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



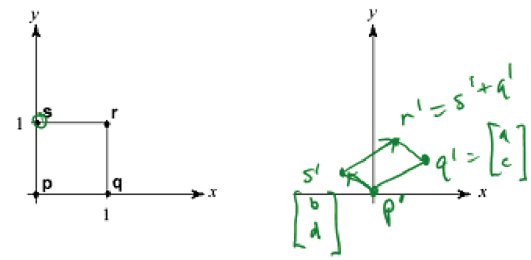
$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x+y \\ y \end{bmatrix}$$

Effect on unit square

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

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

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



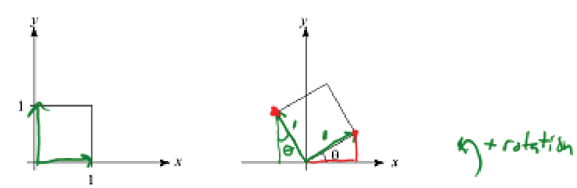
Effect on unit square, cont.

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,

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

Degrees of freedom

For any transformation, we can count its **degrees of freedom** – the number of independent (though not necessarily unique) parameters needed to specify the transformation.

One way to count them is to add up all the apparently free variables and subtract the number of equations that constrain them.

How many degrees of freedom does an arbitrary 2X2 transformation have? 4

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

How many degrees of freedom does a 2D rotation have?

$$\begin{bmatrix} u & v \end{bmatrix} \qquad \left.\begin{matrix} u\cdot u = 1 \quad u\cdot v = 0 \\ v\cdot v = 1 \end{matrix}\right\} \text{3 constraints}$$

$$\Rightarrow 4 - 3 = 1 \text{ DOF}$$

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:

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

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

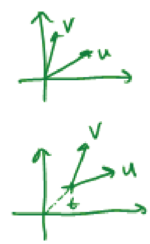
A 2 x 2 linear transformation matrix allows

- Scaling
- Rotation
- Reflection
- Shearing

Q: What important operation does that leave out?

Translation

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:

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

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

How do we write an affine transformation with matrices?

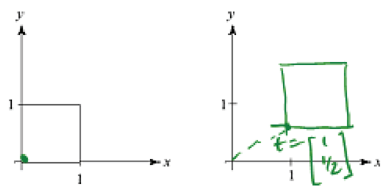
Homogeneous coordinates

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

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

And then transform with a 3 x 3 matrix:

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



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

. . . gives **translation**!

Anatomy of an affine matrix

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

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

A

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:

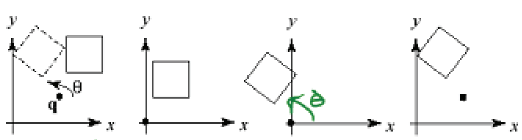
$$M\mathbf{p}_{\text{aff}} = \begin{bmatrix} A\mathbf{p}_{\text{lin}} + \mathbf{t} \\ 1 \end{bmatrix}$$

Rotation about arbitrary points

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

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

$$\begin{matrix} \cos & \sin & 0 \\ -\sin & \cos & 0 \\ 0 & 0 & 1 \end{matrix}$$



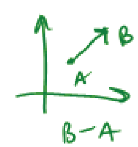
$M \neq T(-q) \cdot R(\theta) \cdot T(q)$

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

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

<u>Note</u>: 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.

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

$$\begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} = \begin{bmatrix} a \ldots \\ \\ \end{bmatrix}$$ no t_x, t_y

⇒ translation invariant

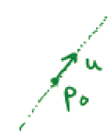
These representations reflect some of the rules of affine operations on points and vectors:

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

One useful combination of affine operations is:

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

Q: What does this describe?

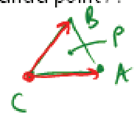


line

$t \geq 0 \Rightarrow$ half the ray

Barycentric coordinates

A set of points can be used to create an affine frame. Consider a triangle ABC and a point P:



We can form a frame with an origin C and the vectors from C to the other vertices:

$$\mathbf{u} = A - C \qquad \mathbf{v} = B - C \qquad \mathbf{t} = C$$

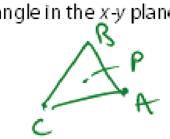
We can then write P in this coordinate frame:

$$P = \alpha\mathbf{u} + \beta\mathbf{v} + \mathbf{t}$$
$$= \alpha(A-C) + \beta(B-C) + C$$
$$= \alpha A + \beta B + (1-\alpha-\beta)C$$
$$= \alpha A + \beta B + \gamma C \qquad \text{where } \gamma = 1-\alpha-\beta$$
$$\alpha+\beta+\gamma = 1$$

The coordinates (α, β, γ) are called the **barycentric coordinates** of P relative to A, B, and C.

Computing barycentric coordinates

Starting from a triangle in the x-y plane



we can compute the barycentric coordinates of P:

$$\alpha A + \beta B + \gamma C = \begin{bmatrix} A_x & B_x & C_x \\ A_y & B_y & C_y \\ 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ 1 \end{bmatrix}$$

Simple matrix analysis gives the solution:

$$\alpha = \frac{\begin{Vmatrix} P_x & B_x & C_x \\ P_y & B_y & C_y \\ 1 & 1 & 1 \end{Vmatrix}}{\begin{Vmatrix} A_x & B_x & C_x \\ A_y & B_y & C_y \\ 1 & 1 & 1 \end{Vmatrix}} \quad \beta = \frac{\begin{Vmatrix} A_x & P_x & C_x \\ A_y & P_y & C_y \\ 1 & 1 & 1 \end{Vmatrix}}{\begin{Vmatrix} A_x & B_x & C_x \\ A_y & B_y & C_y \\ 1 & 1 & 1 \end{Vmatrix}} \quad \gamma = \frac{\begin{Vmatrix} A_x & B_x & P_x \\ A_y & B_y & P_y \\ 1 & 1 & 1 \end{Vmatrix}}{\begin{Vmatrix} A_x & B_x & C_x \\ A_y & B_y & C_y \\ 1 & 1 & 1 \end{Vmatrix}}$$

Computing the determinant of the denominator gives:

$$B_xC_y - B_yC_x + A_yC_x - A_xC_y + A_xB_y - A_yB_x$$

Cross products in 2D

Recall the 3D cross product:

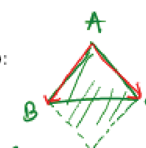
$$\mathbf{u}\times\mathbf{v} = \begin{Vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_x & u_y & u_z \\ v_x & v_y & v_z \end{Vmatrix}$$
$$= (u_yv_z - u_zv_y)\mathbf{i} + (u_zv_x - u_xv_z)\mathbf{j} + (u_xv_y - u_yv_x)\mathbf{k}$$
$$= \begin{bmatrix} u_yv_z - u_zv_y \\ u_zv_x - u_xv_z \\ u_xv_y - u_yv_x \end{bmatrix}$$

What happens when $\mathbf{u}$ and $\mathbf{v}$ lie in the x-y plane?

$$\begin{bmatrix} 0 \\ 0 \\ u_xv_y - u_yv_x \end{bmatrix}$$

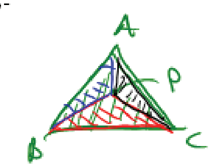
Barycentric coords from area ratios

Now, let's rearrange the equation from two slides ago:



$$B_xC_y - B_yC_x + A_yC_x - A_xC_y + A_xB_y - A_yB_x =$$
$$\underbrace{(B_x - A_x)(C_y - A_y) - (B_y - A_y)(C_x - A_x)}$$

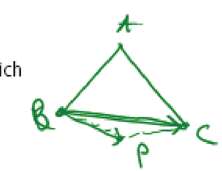
The determinant is then just the z-component of (B-A) x (C-A), which is two times the area of triangle ABC!



Thus, we find:

$$\alpha = \frac{\text{SArea}(PBC)}{\text{SArea}(ABC)} \quad \beta = \frac{\text{SArea}(APC)}{\text{SArea}(ABC)} \quad \gamma = \frac{\text{SArea}(ABP)}{\text{SArea}(ABC)}$$

Where SArea(RST) is the signed area of a triangle, which can be computed with cross-products.



What does it mean for a barycentric coordinate to be negative?

P not inside ΔABC

Affine, vector, and convex combinations

Note that we seem to have constructed a point by adding points together, which we said was illegal, but as long as they have coefficients that sum to one, it's ok.

More generally:

$$P = \alpha_1 A_1 + \cdots + \alpha_n A_n$$

is an **affine combination** if:

$$\sum_{i=1}^{n} \alpha_i = 1$$

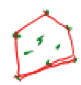
It is a **vector** combination if:

$$\sum_{i=1}^{n} \alpha_i = 0$$

And it is a **convex combination** if:

$$\sum_{i=1}^{n} \alpha_i = 1 \quad \text{and} \quad \alpha_i \geq 0$$

Q: Why is it called a convex combination?



m_i

$$\frac{\sum m_i P_i}{\sum m_i}$$

$$\sum \frac{m_i}{\sum m_i} P_i$$

$\uparrow$

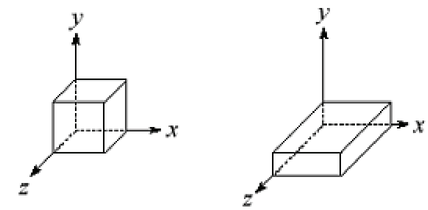
α_i

Basic 3-D transformations: scaling

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

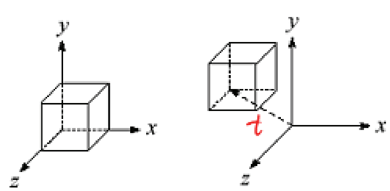
For example, scaling:

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



Translation in 3D

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



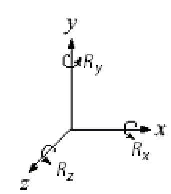
Rotation in 3D

Rotation now has more possibilities in 3D:

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

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

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



Use right hand rule

Rotation in 3D (cont'd)

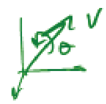
How many degrees of freedom are there in an arbitrary rotation?

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

9 vars.

$$\left.\begin{matrix} u \cdot u = 1 & u \cdot v = 0 \\ v \cdot v = 1 & u \cdot w = 0 \\ w \cdot w = 1 & v \cdot w = 0 \end{matrix}\right\} \text{6 constr.}$$

$$\Rightarrow 9 - 6 = 3 \text{ DOF}$$



How else might you specify a 3D rotation?

v, θ

↳ $v \cdot v = 1$

Quaternions

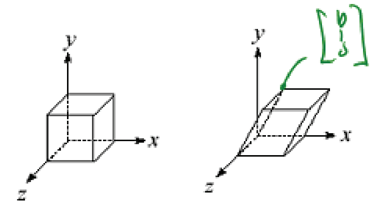
$q = a\hat{i} + b\hat{j} + c\hat{k} + d\hat{e}$

$q \cdot q = 1$

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.

Preservation of affine combinations

A transformation F is an affine transformation if it preserves affine combinations:

$$F(\alpha_1 A_1 + \cdots + \alpha_n A_n) = \alpha_1 F(A_1) + \cdots + \alpha_n F(A_n)$$

where the A_i are points, and:

$$\sum_{i=1}^{n} \alpha_i = 1$$

Clearly, the matrix form of F has this property.

One special example is a matrix that drops a dimension. For example:

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

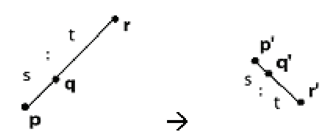
This transformation, known as an orthographic projection is an affine transformation.

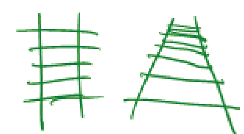
We'll use this fact later...

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()` $\mathbf{M} \leftarrow \mathbf{I}$
  - set **M** to identity
- `glTranslatef(tx, ty, tz)` $\mathbf{M} \leftarrow \mathbf{MT}$
  - 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(sx, sy, sz)` $\mathbf{M} \leftarrow \mathbf{MS}$
  - scale by (s_x, s_y, s_z)

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