COS 426 Lecture 5:

Geometric Transformations

Reading:

- Angel, Sections 4.5–4.8, 5.1, 5.3, 5.8, 7.2
- Foley *et al.*, Chapter 5 (optional)
- David F. Rogers and J. Alan Adams, *Mathematical Elements for Computer Graphics*, Second edition, McGraw-Hill, New York, 1990, Chapters 2–3. (optional)

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

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

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

$$p' = M\,p$$

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

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

$$p' = p\,T$$

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

We will use column vectors.

Two-dimensional transformations

Here's all you get with a 2×2 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 $a, b, c, d \ldots$

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 scaling in x and y:

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

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

Examples:

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

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

The matrix

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

gives:

$$x' = x$$
$$y' = cx + y$$

Effect is called a "_____."

Effect on unit square

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

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

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":

- $\begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow$
- $\begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow$

Can also be viewed as a simultaneous "squash and shear."

Limitations of the 2×2 matrix

A 2×2 matrix allows

- Scaling
- Rotation
- Reflection
- Shearing

Q: What important operation does that leave out?

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 3×3 matrix:

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

. . . Gives translation!

Rotation about arbitrary points

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

With homogeneous coordinates, you can specify rotations about any point q with a matrix:

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

Note: Transformation order is important!

Basic 3-D transformations

The basic 3-D transformations are just like the 2-D ones:

Translation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Basic 3-D transformations, cont.

Rotation

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

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

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

Mathematical properties of affine transformations

All of the transformations we've looked at so far are examples 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)

Mathematical properties of projective transformations

The perspective projection is an example of a "projective transformation."

Here are some properties of projective transformations:

- Lines map to lines
- Parallel lines do not necessarily remain parallel
- Ratios are not preserved (but "cross-ratios" are):

Cross-ratios (Farin, 4th ed, fig 13.2)

A typical eye space

- "Eye"
  - Acts as the COP
  - Placed at the origin
  - Looks down the z-axis (gives left-handed coordinate system)
- "Screen"
  - Lies in the PP
  - Perpendicular to z-axis
  - At distance D from the eye
  - Centered on z-axis, with radius S

Q: Which objects are visible?

Projecting eye space into screen space

Q: How do we perform the perspective projection from eye space into screen space?

Using similar triangles gives:

Projecting eye space into screen space, cont.

We can write this transformation in matrix form:

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

General perspective projection

Now, at last, we can see what the "last row" does.

In general, the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & q & r & s \end{bmatrix}$$

performs a perspective projection into the plane $px + qy + rz + s = 1$.

Q: Suppose we have a cube C whose edges are aligned with the principal axes. Which matrices give drawings of C with

- one-point perspective
- two-point perspective
- three-point perspective?

Perspective depth

Q: What did our perspective projection do to z?

Often, it's useful to have a z around — e.g., for hidden surface calculations.

Hither and yon planes

In order to preserve depth, we set up two planes:

- The "hither" plane $z_e = N$
- The "yon" plane $z_e = F$

We'll map:

$$z_e = N \;\Rightarrow\; z_s = 0$$

$$z_e = F \;\Rightarrow\; z_s = 1$$

Exercise: Derive the matrix to do this projection.