Transformations

CS116A
Chris Pollett
Oct 13, 2004.

Outline

- Two Dimensional Composite Transformations
- Other Two-dimensional Transformations
- Raster Methods for Geometric Transformations

Two Dimensional Composite Transformations

- One can perform a sequence of transformations using a composite transformation matrix.
- Basically, this is this a matrix that results from taking the product of the individual matrices:

  P' = M_2 M_1 P

  P' = M P

Composite 2D-Translations

- Let T(t_{1x}, t_{1y}) and T(t_{2x}, t_{2y}) be two translation matrices.
- To calculate the result of both transformations could do:

newP = T(t_{2x}, t_{2y}) [T(t_{1x}, t_{1y}) P]

= [T(t_{2x}, t_{2y}) T(t_{1x}, t_{1y})] P

$$= \begin{bmatrix} 0 & 0 & t\_2x \\ 0 & 1 & t\_2y \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & t\_1x \\ 0 & 1 & t\_1y \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & t\_2x+t\_1x \\ 0 & 1 & t\_2y+t\_1y \\ 0 & 1 & 1 \end{bmatrix}$$

= T(t_2x}+t_(1x), t_{2y}+ t_{1y}) P.

Composite 2D-Rotations

- A similar thing happens as a result of two rotations:

$$\mathbf{newP} = \mathbf{R(\theta\_2)\ (R(\theta\_1)\ P)}$$
$$= \mathbf{(R(\theta\_2)\ R(\theta\_1))\ P}$$
$$= \mathbf{R(\theta\_2+\theta\_1)P}$$

Composite 2D Scalings

- For scalings something slightly different happens:

$$\begin{bmatrix} s\_2x & 0 & 0 \\ 0 & s\_2y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s\_1x & 0 & 0 \\ 0 & s\_1y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s\_1x*s\_2x & 0 & 0 \\ 0 & s\_1y*s\_2y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- So **S(s_{2x},s_{2y}) S(s_{1x},s_{1y}) = S(s_{2x}*s_{1x}, s_{2y}*s_{1y})**

General 2D Pivot-Point Rotation

- Suppose we want to rotate by an angle θ about some point (x,y). How do we do it?
- First, do $\mathbf{T(-x, -y)= T^{-1}(x,y)}$ to move (x,y) to the origin
- Then do a rotation $\mathbf{R(\theta)}$
- Finally ,undo our translation using $\mathbf{T(x,y)}$
- So have $\mathbf{R(x,y,\theta) = T(x,y)\ R(\theta)T(-x,-y) =}$

$$\begin{bmatrix} \cos\theta & -\sin\theta & x(1-\cos\theta)+y\sin\theta \\ \sin\theta & \cos\theta & y(1-\cos\theta)+x\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

General 2D Fixed Point Scaling

- We might similarly want to do a scaling with respect to some fixed point:

S(x, y, s_x, s_y) = T(x,y) S(x,y) T(-x,-y)

$$= \begin{bmatrix} s\_x & 0 & x(1- s\_x) \\ 0 & s\_y & y(1 - s\_y) \\ 0 & 0 & 1 \end{bmatrix}$$

General 2D Scaling Directions

- We might also want to scale with respect to some direction other than the x and y axis.
- To do this we rotate to the direction we want to scale in **R(θ)**
- Then we scale **S(s_1,s_2)**
- Then we rotate back **R^{-1}(θ) = R(-θ).**
- This give **R^{-1}(θ) S(s_1,s_2) R(θ)**
- You should work out the matrix.

Matrix Concatenation Properties

- In the previous slides we have been using the following useful property of matrices:

  M_3 M_2 M_1 = (M_3 M_2) M_1
  = M_3 (M_2 M_1)
- This property is called associativety and holds even if the matrices are not square.
- In fact, in previous slides one matrix (the column vector for the point) was not square.
- Note: in general, **M_2 M_1 =\= M_1 M_2**

General 2D Composite Transformations and Computational Efficiency

- A 2D Transformation representing any combination of rotations/ scaling/ translations can be written as:

$$\begin{bmatrix} rs\_xx & rs\_xy & trs\_x \\ rs\_yx & rs\_yy & trs\_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Can show need a maximum of 4 mults, 4 adds / coordinate to do any such transformation
- Note if have an angle θ, we can calculate sin, cos once and rs's and trs's once and then use these coefficients over and over

2D Rigid-Body Transformations

- A rigid body transformation consists of only rotations and translations.
- Matrix looks like:

$$\begin{bmatrix} r\_xx & r\_xy & tr\_x \\ r\_yx & r\_yy & tr\_y \\ 0 & 0 & 1 \end{bmatrix}$$

- For an object, all of its edge lengths and angles will be preserved by such a transformation

Other 2D Transformations

- Some graphics packages support additional kinds of 2D transformations;
- For example:
  - Reflection
  - Shear

Reflections

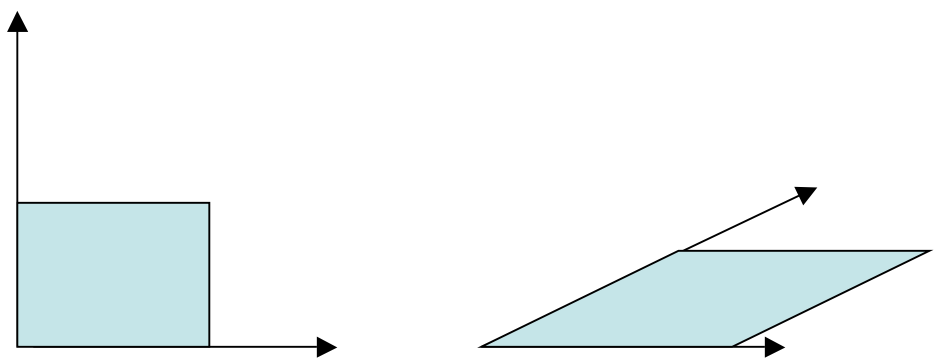
- Reflections about either x or y axis.
  - x-axis: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
  - y-axis: $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
  - Both axes together: $\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

More Reflections

- Reflections about an arbitrary line can be achieved using reflections combined with translations and rotations.

Shear

- A **shear** causes an effect like the following:



- That is, we take a system of coordinates and tilt over one of the axes. Matrix for x-axis shear looks like:

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

Raster Methods for Geometric Transformations

- Many of the transformations we have considered can be carried out rapidly by raster systems without having to multiply each point by a matrix.
- One common useful raster operation is a block transfer (aka bitblt or pixblt).
- This allows us to move a rectangular block of pixel values from one position to another in the frame. It can be used to do translations rapidly.
- Rotations by 90 or 180 degrees for rectangular regions can also be calculated rapidly. Can generalize to other angles.
- Similarly, there are tricks for scaling.