CS4815 Week04 Tutorial

1. The rotation matrix R for rotating points around the origin through a positive (anti-clockwise) angle θ is

$$R = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

   That is, the point $p = (x, y)$ gets rotated to $q = Rp$.

   What is the matrix for rotating the opposite (clockwise) way?

   Answer:
   To rotate in the opposite direction

$$R^{-} = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2. You would expect that if you rotated a point anti-clockwise through some angle, θ, first and then rotated the result by the same angle but in the opposite (clockwise) direction, $-\theta$ that you would get back to where you started.

   Show that this is true by taking a point, say, $p = (1, 1)$, rotating it through θ and then rotate the result through $-\theta$. That is, compute

$$p_1 = R_\theta p$$

   and then

$$p_2 = R_{-\theta} p_1.$$

   How does this compare to multiplying the two *matrices* together first of all and then multiplying the point by the result as in

$$q = (R_{-\theta} R_\theta)p$$

   Why might the second approach be better in general cases where one rotation doesn't cancel the other out?

   Answer:
   The advantage of the second approach is that when several transformations need to be applied in succession to all of the points in a scene once the *compound* matrix is found then every point can be simply transformed to its destination by one multiplication. Otherwise we are looking at having to do each individual transformation on each point – far more matrix multiplications.

3. By multiplying together a translation matrix, a rotation matrix and an un-translation matrix, build a matrix for rotating around the point $(1, 1)$.

   Answer:

   We can only ever perform rotations **around the origin** using matrix multiplications. This means that if we wish to rotate around some *other* point we have to a) shift the entire scene to the origin, b) perform the required rotation through θ and, c) un-shift the scene back to where it was.

   Step a) can be achieved in this case by multiplying every 2-D point by the translation matrix

$$T_1 = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}.$$

   Following rotation the translation matrix to return to the original co-ordinates is

$$T_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

   In summary, a point $(x, y)^T$ will get transformed to

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = T_2 \times R_\theta \times T_1 \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = M \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

   where

$$M = T_2 \times R_\theta \times T_1$$

   Note the order of multiplication here. Very important!

   Suppose, for simplicity, $\theta = 90°$ ($=\pi/2$ radians) and we wish to rotate our world around $(1, 1)$ by this angle. (Convince yourself that the point $(2, 1)$ will get rotated to $(1, 2)$ by this transformation. Go on, draw a picture.)

   The rotation matrix will be

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

   since $\cos \pi/2 = 0$ and $\sin \pi/2 = 1$.

   This gives the composite matrix M as

$$M = T_2 \times R_\theta \times T_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 2 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

   When we hit the vector corresponding to $(2, 1)$ with M you should verify that we get $(1, 2)$, as expected.

   One final question: what would the point $(1, 1)$ get rotated to?

4. The *determinant* of a matrix gives valuable information about how that matrix transforms a polygon. We will not justify this but it is a fact that the determinant of a matrix tells how the area of the polygon changes when transformed by applying the matrix transformation on every point that comprises the polygon. That is, the determinant of a matrix is the ratio of after-to-before areas.

   Clearly, rotating a square or a triangle or any polygon will leave its area intact.

   (a) By taking the determinant of a rotation matrix verify the above property.

   Answer:
   One way to calculate the determinant of M is

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - ceg - bdi - afh$$

   For a rotation matrix R

$$\begin{vmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{vmatrix} = \cos^2\theta + 0 + 0 - 0 - (-\sin^2\theta) - 0 = \cos^2\theta + \sin^2\theta = 1$$

   (b) What is the matrix that achieves the *flip* transformation? You can either flip about the x-axis or the y axis.

   Answer:
   Flipping around the x-axis can be achieved with

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

   (c) How should the area be affected by a flip transformation?

   Answer:
   Like with a rotation flipping should leave the area of the polygon unchanged.

   (d) Verify this by taking the determinant of your matrix. Anything mildly unsettling about your answer?

   Answer:
   The determinant of F above is -1. So in absolute value terms the areas will remain the same. The reason for the sign change is that the *ordering* of the points is reversed: tracing around the outline of the polygon after the flip results in the opposite ordering of the points to before the flip and this accounts for the negative sign. (This doesn't happen with a rotation.)