School of Computing
National University of Singapore
CS5240 Theoretical Foundations in Multimedia

Geometric Transformations

In computer graphics and vision, we often need to manipulate objects. Typically, we rotate, scale, and translate an object. These operations can be described using vectors and matrices. Given a 2D point $\mathbf{x} = [x \quad y]^\top$, such geometric transformations may be achieved via a simple matrix multiplication:

$$\mathbf{u} = \mathbf{M}\mathbf{x} \tag{1}$$

where $\mathbf{M}$ is the appropriate matrix that performs the transformation, and $\mathbf{u}$ is the coordinates of $\mathbf{x}$ after the transformation.

1 Scaling

$$\begin{bmatrix} \alpha x \\ \alpha y \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

This scales both axes by the same amount α. In general, we can scale the x and y axes differently:

$$\mathbf{M} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \tag{3}$$

2 Rotation

Let's work out how to rotate a 2D point around the origin by an angle θ. By convention, θ is positive if you rotate anti- (counter) clockwise, and negative if clockwise. Consider the point $(1, 0)$, which ends up at $(\cos\theta, \sin\theta)$, as shown in Figure 1. Also consider the point $(0, 1)$, which is rotated to $(-\sin\theta, \cos\theta)$. Using Equations (4) and (5), can you determine what the rotation matrix is?

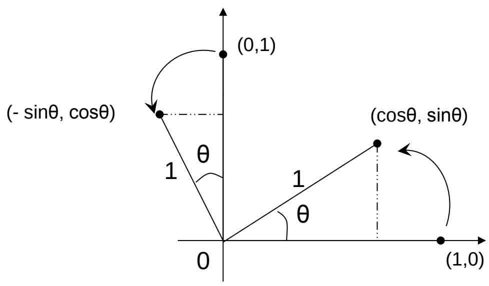

Figure 1: Rotating the point $(1, 0)$ by θ.

$$\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5}$$

In general, a rotation matrix is given by: $\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

2.1 Properties

$\det(\mathbf{R}) = \cos^2\theta + \sin^2\theta = 1$

Rotate clockwise $\mathbf{R}_{-\theta} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \mathbf{R}_\theta^\top$

Note: $\mathbf{R}^\top\mathbf{R} = \mathbf{I} \Rightarrow \mathbf{R}$ is orthogonal.

3 Translation

Translating a point $\mathbf{x}$ by an amount $\mathbf{t} = [t_x \quad t_y]^\top$ may be written as:

$$\begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{6}$$

But this breaks the simple multiplcation form that we like. For example, if you transform a point with $\mathbf{M}$, and then translate it by $\mathbf{t}$, you will need to write: $\mathbf{Mx} + \mathbf{t}$. This is messy! It is better to rewrite:

$$\begin{bmatrix} \mathbf{M} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{Mx} + \mathbf{t} \\ 1 \end{bmatrix} \tag{7}$$

This leads naturally to the concept of Homogeneous Coordinates.

4 Homogeneous Coordinates

The idea here is to add another dimension to your 2D vector:

$$\begin{bmatrix} x \\ y \end{bmatrix} \to \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}, \quad \text{where } x = x'/z', y = y'/z', z' \neq 0$$

Note: $\begin{bmatrix} kx' \\ ky' \\ kz' \end{bmatrix} \to \begin{bmatrix} x \\ y \end{bmatrix}, \quad k \neq 0$ represent the same point.

In homogeneous coordinates, all geometric transformations can be written as matrix multiplication, without addition.

$$\mathbf{u} = \mathbf{Mx}, \quad \text{or} \quad \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

To get back to the usual 2D coords from homogeneous coords:

$$u = u'/w', \; v = v'/w', \; w' \neq 0$$

If $w' = 0$ then the point is at infinity.

4.1 Composition

Applying transformation $\mathbf{T}_1$ to a point $\mathbf{x}$, followed by transformation $\mathbf{T}_2$ may be written as:

$$\mathbf{u} = \mathbf{T}_2\mathbf{T}_1\mathbf{x}$$

In general, composition is not commutative, i.e. $\mathbf{T}_2\mathbf{T}_1 \neq \mathbf{T}_1\mathbf{T}_2$

Note: $k\mathbf{u} = (k\mathbf{M})\mathbf{x}$
This means that a scalar multiple of a transformation represents the same transformation.

4.2 General Transformation

From now on, we will use homogenous coordinations, unless stated otherwise. A general transformation is now a 3×3 matrix:

$$\mathbf{M} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \tag{8}$$

This has 9 entries, but really only 8 "degrees of freedom". It is usual to let $i = 1$. Note that a transformation is unique "up to scale", i.e. the scalar multiple is unknown, and need not be known. The matrix in Equation (8) is also called a *perspective projection*. Some special cases are worth noting:

Affine transformation:

$$\mathbf{M} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Euclidean transformation:

$$\mathbf{M} = \begin{bmatrix} s \cdot \cos\theta & -s \cdot \sin\theta & t_x \\ s \cdot \sin\theta & s \cdot \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Similarity transformation:

$$\mathbf{M} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

If both $\mathbf{u}$ and $\mathbf{x}$ lie on plane, then M is called a "homography",i.e. a transformation from one plane to another.

Example: What matrix will rotate $\mathbf{x}$ around a point $\mathbf{p} = [p_x \; p_y]^\top$ to get to $\mathbf{u}$?

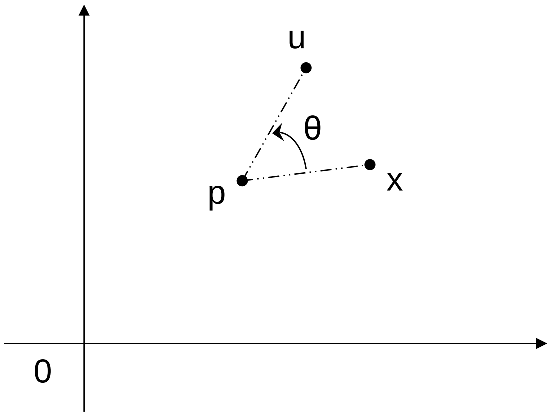

Figure 2: Rotation around a point $\mathbf{p}$

Answer:

$$\mathbf{u} = \underbrace{\begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}^{-1}} \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R}} \underbrace{\begin{bmatrix} 1 & 0 & -p_x \\ 0 & 1 & -p_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \mathbf{x}$$

4.3 Transformations in 3D

We may generalize geometric transformations to 3D. The key idea is again to use homogenous coordinates. For a 3D point $\mathbf{x}$, its homogeneous coords is a 4D vector:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}$$

where $x = x'/w',\ y = y'/w',\ z = z'/w'$, and $w' \neq 0$.

Rotation in 3D is more complicated, since you need to specify which axis around which to perform the rotation. We will not cover this. Please refer to standard computer graphics textbooks instead.