CHAPTER 17

Visualization and Rigid Body Dynamics

We study affine transformations in order to visualize objects in three dimensions. We then solve Euler's equation of motion for rigid body dynamics using the quaternion representation of rotations.

17.1 ■ TWO-DIMENSIONAL TRANSFORMATIONS

Physicists frequently use transformations to convert from one system of coordinates to another. A very common transformation is an *affine transformation*, which has the ability to rotate, scale, stretch, skew, and translate an object. Such a transformation maps straight lines to straight lines. They are often represented using matrices and are manipulated using the tools of linear algebra such as matrix multiplication and matrix inversion. The Java 2D API defines a set of classes designed to create high quality graphics using image composition, image processing, antialiasing, and text layout. Because linear algebra and affine transformations are used extensively in imaging and drawing APIs, we begin our study of two- and three-dimensional visualization techniques by studying the properties of transformations.

It is straightforward to rotate a point (x, y) about the origin by an angle θ (see Figure 17.1) or scale the distance from the origin by (s_x, s_y) using matrices:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{17.1}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{17.2}$$

Performing several transformations corresponds to multiplying matrices. However, the translation of the point (x, y) by (d_x, d_y) is treated as an addition and not as a multiplication and must be written differently:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}. \tag{17.3}$$

This inconsistency in the type of mathematical operation is easily overcome if points are expressed in terms of *homogeneous coordinates* by adding a third coordinate w. Homogeneous coordinates are used extensively in computer graphics to treat all transformations consistently. Instead of representing a point in two dimensions by a pair of numbers (x, y), each point is represented by a triple (x, y, w). Because two homogeneous coordinates represent the same point if one is a multiple of the other, we usually *homogenize* the point

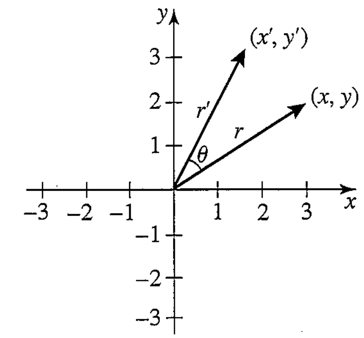

Figure 17.1 A two-dimensional rotation of a point (x, y) produces a point with new coordinates (x', y') as computed according to (17.1).

by dividing the xy-coordinates by w and write the coordinates in the form $(x, y, 1)$. (The w-coordinate can be used to add perspective (see Foley et al.).) By using homogeneous coordinates, an arbitrary affine transformation can be written as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{17.4}$$

A translation, for example, can be expressed as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{17.5}$$

Exercise 17.1 Homogeneous coordinates

(a) How are the rotation and scaling transformations expressed in matrix notation using homogeneous coordinates? Sketch the transformation matrices for a 30° clockwise rotation and for a scaling along the x-axis by a factor of two and then write the transformation matrices. Do these matrices commute?

(b) Describe the effect of the affine transformation

$$\begin{bmatrix} 1 & 0.2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{17.6}$$

■

Exercise 17.1 shows that a coordinate transformation can be broken into parts using a *block matrix* format:

$$\begin{bmatrix} \mathcal{A} & \mathbf{d}^T \\ \mathbf{0} & 1 \end{bmatrix}. \tag{17.7}$$