Geometric Transformation

Basic Two Dimensional Geometric Transformations

The geometric-transformation functions that are available in all graphics packages are those for translation, rotation, and scaling. Other useful transformation routines that are sometimes included in a package are reflection and shearing operations. To introduce the general concepts associated with geometric transformations, we first consider operations in two dimensions, then we discuss how these basic ideas can be extended to three-dimensional scenes. Once we understand the basic concepts, we can easily write routines to perform geometric transformations on objects in a two-dimensional scene.

Two Dimensional Translation

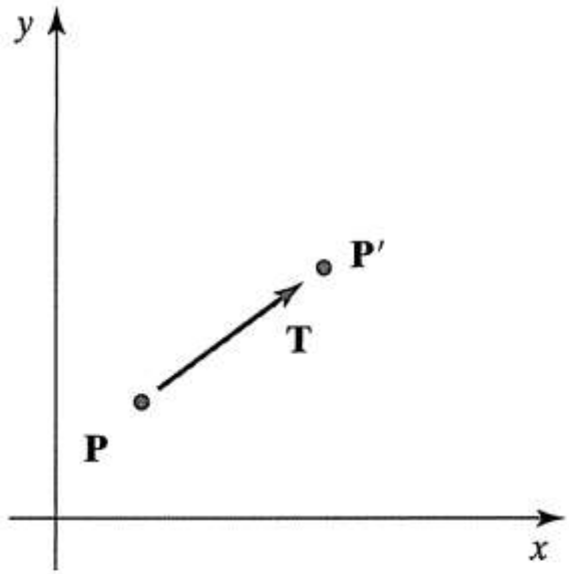


FIGURE 5-1 Translating a point from position **P** to position **P′** using a translation vector **T**.

Two Dimensional Translation (Continued)

To translate a two-dimensional position, we add **translation distances** t_x and t_y to the original coordinates (x, y) to obtain the new coordinate position (x', y') as shown in Fig. 5-1.

$$x' = x + t_x, \qquad y' = y + t_y \tag{5-1}$$

The translation distance pair (t_x, t_y) is called a **translation vector** or **shift vector.**

We can express the translation equations 5-1 as a single matrix equation by using the following column vectors to represent coordinate positions and the translation vector.

$$\mathbf{P} = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \qquad \mathbf{T} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{5-2}$$

This allows us to write the two-dimensional translation equations in the matrix form

$$\mathbf{P}' = \mathbf{P} + \mathbf{T} \tag{5-3}$$

Translation is a *rigid-body transformation* that moves objects without deformation. That is, every point on the object is translated by the same amount. A straight-line segment is translated by applying the transformation equation 5-3 to each of the two line endpoints and redrawing the line between the new endpoint positions. A polygon is translated similarly. We add a translation vector to the coordinate position of each vertex and then regenerate the polygon using the new set of vertex coordinates. Figure 5-2 illustrates the application of a specified translation vector to move an object from one position to another.

How to Move a circle from on position to another?

Example:

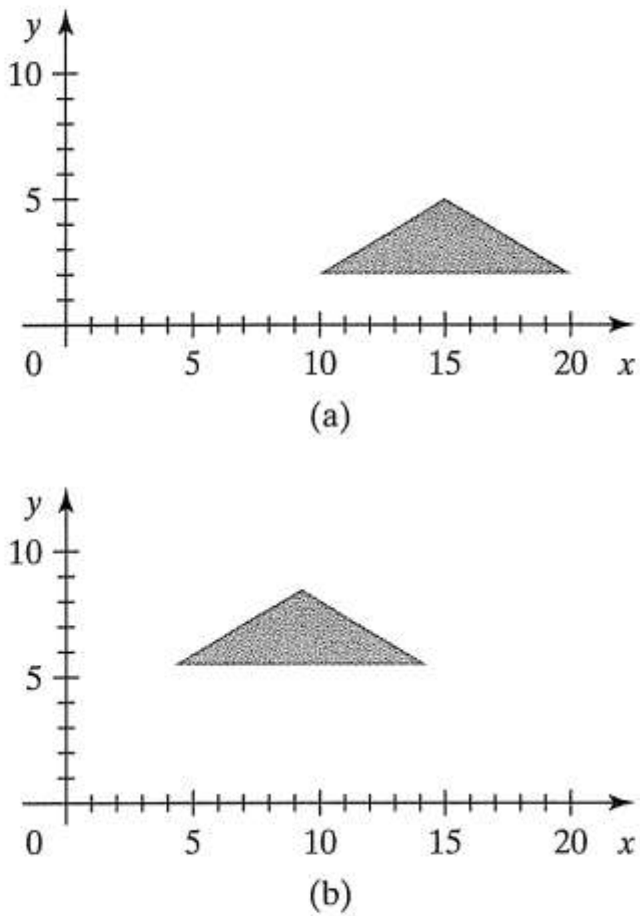


FIGURE 5-2 Moving a polygon from position (a) to position (b) with the translation vector (–5.50, 3.75).

Two –Dimensional Rotation

We generate a **rotation** transformation of an object by specifying a rotation axis and a rotation angle. All points of the object are then transformed to new positions by rotating the points through the specified angle about the rotation axis.

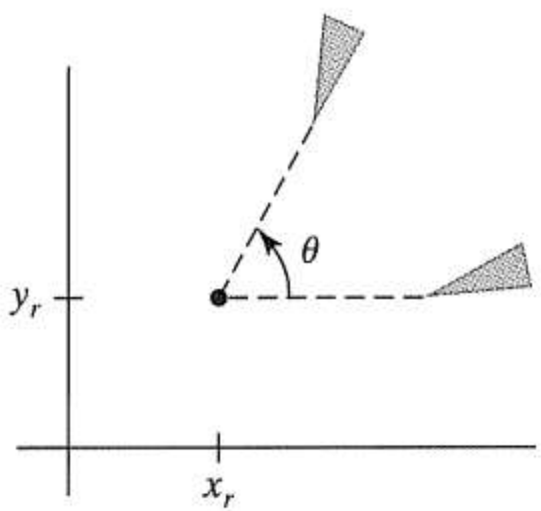


FIGURE 5-3 Rotation of an object through angle θ about the pivot point (x_r, y_r).

Two –Dimensional Rotation

A two-dimensional rotation of an object is obtained by repositioning the object along a circular path in the *xy* plane. In this case, we are rotating the object about a rotation axis that is perpendicular to the *xy* plane (parallel to the coordinate z axis). Parameters for the two-dimensional rotation are the rotation angle θ and a position (x_r, y_r), called the **rotation point** (or **pivot point**), about which the object is to be rotated (Fig. 5-3). The pivot point is the intersection position of the rotation axis with the *xy* plane. A positive value for the angle θ defines a counterclockwise rotation about the pivot point, as in Fig. 5-3, and a negative value rotates objects in the clockwise direction.

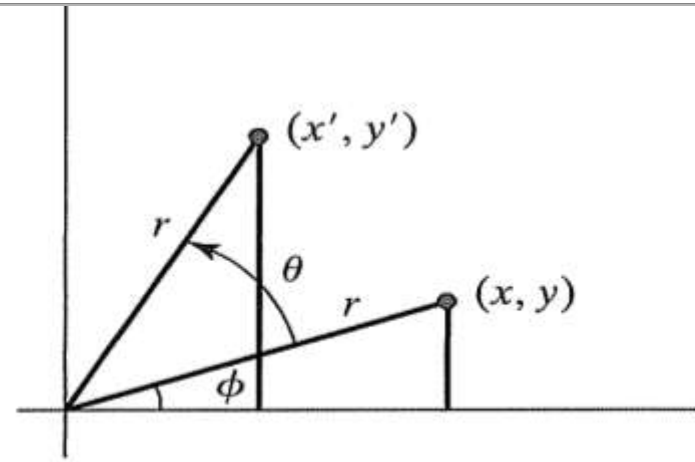


FIGURE 5–4 Rotation of a point from position (x, y) to position (x', y') through an angle θ relative to the coordinate origin. The original angular displacement of the point from the x axis is ϕ.

To simplify the explanation of the basic method, we first determine the transformation equations for rotation of a point position **P** when the pivot point is at the coordinate origin. The angular and coordinate relationships of the original and transformed point positions are shown in Fig. 5-4. In this figure, r is the constant distance of the point from the origin, angle ϕ is the original angular position of the point from the horizontal, and θ is the rotation angle. Using standard trigonometric identities, we can express the transformed coordinates in terms of angles θ and ϕ as

$$\begin{aligned} x' &= r\cos(\phi + \theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta \\ y' &= r\sin(\phi + \theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta \end{aligned} \tag{5-4}$$

The original coordinates of the point in polar coordinates are

$$x = r \cos \phi, \qquad y = r \sin \phi \tag{5-5}$$

Substituting expressions 5-5 into 5-4, we obtain the transformation equations for rotating a point at position (x, y) through an angle θ about the origin:

$$\begin{aligned} x' &= x \cos \theta - y \sin \theta \\ y' &= x \sin \theta + y \cos \theta \end{aligned} \tag{5-6}$$

With the column-vector representations 5-2 for coordinate positions, we can write the rotation equations in the matrix form

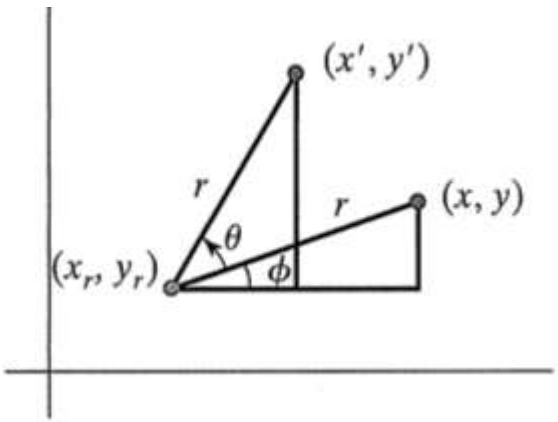
$$\mathbf{P}' = \mathbf{R} \cdot \mathbf{P} \tag{5-7}$$

where the rotation matrix is

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin\theta \\ \sin \theta & \cos \theta \end{bmatrix} \tag{5-8}$$

Rotation of a point about an arbitrary pivot position is illustrated in Fig. 5-5. Using the trigonometric relationships indicated by the two right triangles in this figure, we can generalize Eqs. 5-6 to obtain the transformation equations

for rotation of a point about any specified rotation position (x_r, y_r):



$$x' = x_r + (x - x_r)\cos\theta - (y - y_r)\sin\theta$$
$$y' = y_r + (x - x_r)\sin\theta + (y - y_r)\cos\theta \qquad (5\text{-}9)$$

FIGURE 5-5 Rotating a point from position (x, y) to position (x', y') through an angle θ about rotation point (x_r, y_r).

Two –Dimensional Scaling

To alter the size of an object, we apply a **scaling** transformation. A simple two-dimensional scaling operation is performed by multiplying object positions (x, y) by **scaling factors** s_x and s_y to produce the transformed coordinates (x', y'):

$$x' = x \cdot s_x, \qquad y' = y \cdot s_y \tag{5-10}$$

Scaling factor s_x scales an object in the x direction, while s_y scales in the y direction. The basic two-dimensional scaling equations 5-10 can also be written in the following matrix form.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{5-11}$$

or

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P} \tag{5-12}$$

where **S** is the 2 by 2 scaling matrix in Eq. 5-11.

Any positive values can be assigned to the scaling factors s_x and s_y. Values less than 1 reduce the size of objects; values greater than 1 produce enlargements. Specifying a value of 1 for both s_x and s_y leaves the size of objects unchanged. When s_x and s_y are assigned the same value, a **uniform scaling** is produced which maintains relative object proportions. Unequal values for s_x and s_y result in a **differential scaling** that is often used in design applications, where pictures are constructed from a few basic shapes that can be adjusted by scaling and positioning transformations (Fig. 5-6). In some systems, negative values can also be specified for the scaling parameters. This not only resizes an object, it reflects it about one or more of the coordinate axes.

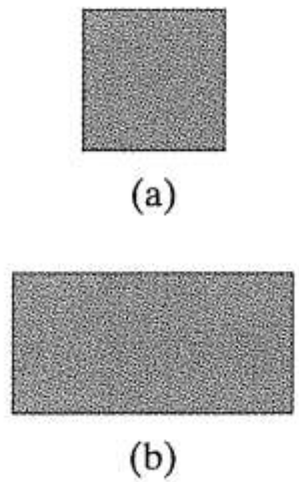


FIGURE 5-6 Turning a square (a) into a rectangle (b) with scaling factors $s_x = 2$ and $s_y = 1$.

Objects transformed with Eq. 5-11 are both scaled and repositioned. Scaling factors with absolute values less than 1 move objects closer to the coordinate origin, while absolute values greater than 1 move coordinate positions farther from the origin. Figure 5-7 illustrates scaling of a line by assigning the value 0.5 to both s_x and s_y in Eq. 5-11. Both the line length and the distance from the origin are reduced by a factor of $\frac{1}{2}$.

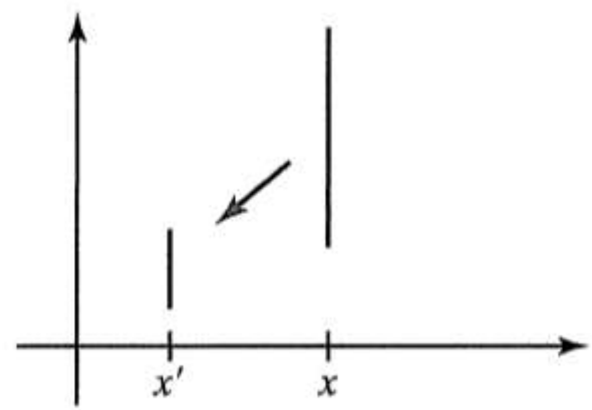


FIGURE 5-7 A line scaled with Eq. 5-12 using $s_x = s_y = 0.5$ is reduced in size and moved closer to the coordinate origin.

[redacted]

We can control the location of a scaled object by choosing a position, called the fixed point, that is to remain unchanged after the scaling transformation. Coordinates for the fixed point, (x_f, y_f), are often chosen at some object position, such as its centroid (Appendix A), but any other spatial position can be selected. Objects are now resized by scaling the distances between object points and the fixed point (Fig. 5-8). For a coordinate position (x, y), the scaled coordinates (x', y') are then calculated from the following relationships.

$$x' - x_f = (x - x_f)s_x, \qquad y' - y_f = (y - y_f)s_y \tag{5-13}$$

We can rewrite Eqs. 5-13 to separate the multiplicative and additive terms as

$$\begin{aligned} x' &= x \cdot s_x + x_f(1 - s_x) \\ y' &= y \cdot s_y + y_f(1 - s_y) \end{aligned} \tag{5-14}$$

where the additive terms $x_f(1 - s_x)$ and $y_f(1 - s_y)$ are constants for all points in the object.

P1=(7,7)
P2 = (10,1)
P3= (5,2)
Fp=(4,5)

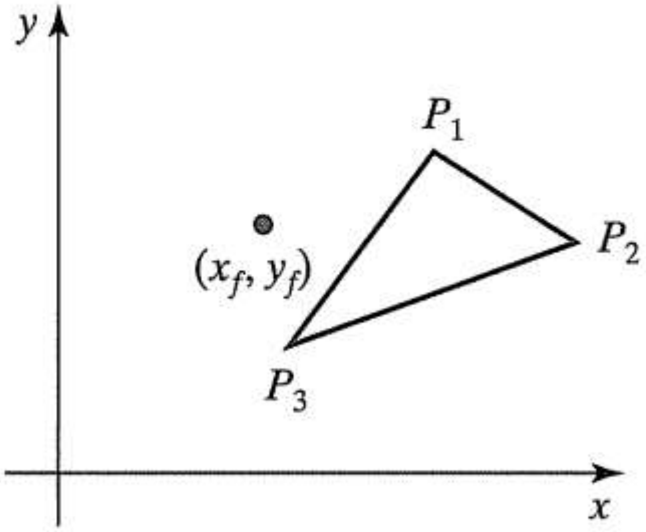


FIGURE 5-8 Scaling relative to a chosen fixed point (x_f, y_f). The distance from each polygon vertex to the fixed point is scaled by transformation equations 5-13.

Polygons are scaled by applying transformations 5-14 to each vertex, then regenerating the polygon using the transformed vertices. For other objects, we apply the scaling transformation equations to the parameters defining the objects. To change the size of a circle, we can scale its radius and calculate the new coordinate positions around the circumference.

End Of Presentation