Matrix Representation and Homogeneous Coordinates

We have seen in Section 5-1 that each of the three basic two-dimensional transformations (translation, rotation, and scaling) can be expressed in the general matrix form

$$\mathbf{P}' = \mathbf{M}_1 \cdot \mathbf{P} + \mathbf{M}_2 \qquad (5\text{-}15)$$

with coordinate positions $\mathbf{P}$ and $\mathbf{P}'$ represented as column vectors. Matrix $\mathbf{M}_1$ is a 2 by 2 array containing multiplicative factors, and $\mathbf{M}_2$ is a two-element column matrix containing translational terms. For translation, $\mathbf{M}_1$ is the identity matrix. For rotation or scaling, $\mathbf{M}_2$ contains the translational terms associated with the pivot point or scaling fixed point. To produce a sequence of transformations with these equations, such as scaling followed by rotation then translation, we could calculate the transformed coordinates one step at a time. First, coordinate positions are scaled, then these scaled coordinates are rotated, and finally the rotated coordinates are translated. A more efficient approach, however, is to combine the transformations so that the final coordinate positions are obtained directly from the initial coordinates, without calculating intermediate coordinate values. We can do this by reformulating Eq. 5-15 to eliminate the matrix addition operation.

Homogeneous Coordinates

Multiplicative and translational terms for a two-dimensional geometric transformation can be combined into a single matrix if we expand the representations to 3 by 3 matrices. Then we can use the third column of a transformation matrix for the translation terms, and all transformation equations can be expressed as matrix multiplications. But to do so, we also need to expand the matrix representation for a two-dimensional coordinate position to a three-element column matrix. A standard technique for accomplishing this is to expand each two-dimensional coordinate-position representation (x, y) to a three-element representation (x_h, y_h, h), called homogeneous coordinates, where the **homogeneous parameter** h is a nonzero value such that

$$x = \frac{x_h}{h}, \qquad y = \frac{y_h}{h} \tag{5-16}$$

Therefore, a general two-dimensional homogeneous coordinate representation could also be written as $(h \cdot x, h \cdot y, h)$. For geometric transformations, we can choose the homogeneous parameter h to be any nonzero value. Thus, there are an infinite number of equivalent homogeneous representations for each coordinate point (x, y). A convenient choice is simply to set $h = 1$. Each two-dimensional position is then represented with homogeneous coordinates $(x, y, 1)$. Other values for parameter h are needed, for example, in matrix formulations of three-dimensional viewing transformations.

Two-Dimensional Translation Matrix

Using a homogeneous-coordinate approach, we can represent the equations for a two-dimensional translation of a coordinate position using the following matrix multiplication.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (5\text{-}17)$$

This translation operation can be written in the abbreviated form

$$\mathbf{P}' = \mathbf{T}(t_x, t_y) \cdot \mathbf{P} \qquad (5\text{-}18)$$

with $\mathbf{T}(t_x, t_y)$ as the 3 by 3 translation matrix in Eq. 5-17. In situations where there is no ambiguity about the translation parameters, we can simply represent the translation matrix as $\mathbf{T}$.

Tow-dimensional Rotation matrix

Similarly, two-dimensional rotation transformation equations about the coordinate origin can be expressed in the matrix form

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5-19}$$

or as

$$\mathbf{P}' = \mathbf{R}(\theta) \cdot \mathbf{P} \tag{5-20}$$

The rotation transformation operator $\mathbf{R}(\theta)$ is the 3 by 3 matrix in Eq. 5-19 with rotation parameter θ. We can also write this rotation matrix simply as $\mathbf{R}$.

Two-Dimensional Scaling Matrix

Finally, a scaling transformation relative to the coordinate origin can now be expressed as the matrix multiplication

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5-21}$$

or

$$\mathbf{P}' = \mathbf{S}(s_x, s_y) \cdot \mathbf{P} \tag{5-22}$$

The scaling operator $\mathbf{S}(s_x, s_y)$ is the 3 by 3 matrix in Eq. 5-21 with parameters s_x and s_y. And, in most cases, we can represent the scaling matrix simply as $\mathbf{S}$.

Inverse Transformations
Inverse Translation

For translation, we obtain the inverse matrix by negating the translation distances. Thus, if we have two-dimensional translation distances t_x and t_y, the inverse translation matrix is

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5-23}$$

This produces a translation in the opposite direction, and the product of a translation matrix and its inverse produces the identity matrix.

Inverse rotation

An inverse rotation is accomplished by replacing the rotation angle by its negative. For example, a two-dimensional rotation through an angle θ about the coordinate origin has the inverse transformation matrix

$$\mathbf{R}^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5-24}$$

Negative values for rotation angles generate rotations in a clockwise direction, so the identity matrix is produced when any rotation matrix is multiplied by its inverse. Since only the sine function is affected by the change in sign of the rotation angle, the inverse matrix can also be obtained by interchanging rows and columns. That is, we can calculate the inverse of any rotation matrix $\mathbf{R}$ by evaluating its transpose ($\mathbf{R}^{-1} = \mathbf{R}^T$).

Inverse scaling

We form the inverse matrix for any scaling transformation by replacing the scaling parameters with their reciprocals. For two-dimensional scaling with parameters s_x and s_y applied relative to the coordinate origin, the inverse transformation matrix is

$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (5\text{-}25)$$

The inverse matrix generates an opposite scaling transformation, so the multiplication of any scaling matrix with its inverse produces the identity matrix.

Two Dimensional composite transformations

Using matrix representations, we can set up a sequence of transformations as a **composite transformation matrix** by calculating the product of the individual transformations. Forming products of transformation matrices is often referred to as a **concatenation,** or **composition,** of matrices. Since a coordinate position is represented with a homogeneous column matrix, we must premultiply the column matrix by the matrices representing any transformation sequence. And, since many positions in a scene are typically transformed by the same sequence, it is more efficient to first multiply the transformation matrices to form a single composite matrix. Thus, if we want to apply two transformations to point position $\mathbf{P}$, the transformed location would be calculated as

$$\begin{aligned} \mathbf{P}' &= \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{P} \\ &= \mathbf{M} \cdot \mathbf{P} \end{aligned} \tag{5-26}$$

The coordinate position is transformed using the composite matrix $\mathbf{M}$, rather than applying the individual transformations $\mathbf{M}_1$ and then $\mathbf{M}_2$.

Composite Two- dimensional Translations

If two successive translation vectors (t_{1x}, t_{1y}) and (t_{2x}, t_{2y}) are applied to a two-dimensional coordinate position $\mathbf{P}$, the final transformed location $\mathbf{P}'$ is calculated as

$$\begin{aligned} \mathbf{P}' &= \mathbf{T}(t_{2x}, t_{2y}) \cdot \{\mathbf{T}(t_{1x}, t_{1y}) \cdot \mathbf{P}\} \\ &= \{\mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y})\} \cdot \mathbf{P} \end{aligned} \tag{5-27}$$

where $\mathbf{P}$ and $\mathbf{P}'$ are represented as three-element, homogeneous-coordinate column vectors. We can verify this result by calculating the matrix product for the two associative groupings. Also, the composite transformation matrix for this sequence of translations is

$$\begin{bmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{1x} + t_{2x} \\ 0 & 1 & t_{1y} + t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \tag{5-28}$$

or

$$\mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y}) = \mathbf{T}(t_{1x} + t_{2x}, t_{1y} + t_{2y}) \tag{5-29}$$

which demonstrates that two successive translations are additive.

Composite Two Dimensional Rotations

Two successive rotations applied to a point **P** produce the transformed position

$$\begin{aligned} \mathbf{P}' &= \mathbf{R}(\theta_2) \cdot \{\mathbf{R}(\theta_1) \cdot \mathbf{P}\} \\ &= \{\mathbf{R}(\theta_2) \cdot \mathbf{R}(\theta_1)\} \cdot \mathbf{P} \end{aligned} \qquad (5\text{-}30)$$

By multiplying the two rotation matrices, we can verify that two successive rotations are additive:

$$\mathbf{R}(\theta_2) \cdot \mathbf{R}(\theta_1) = \mathbf{R}(\theta_1 + \theta_2) \qquad (5\text{-}31)$$

so that the final rotated coordinates of a point can be calculated with the composite rotation matrix as

$$\mathbf{P}' = \mathbf{R}(\theta_1 + \theta_2) \cdot \mathbf{P} \qquad (5\text{-}32)$$

Composite two-dimensional scaling

Concatenating transformation matrices for two successive scaling operations in two dimensions produces the following composite scaling matrix.

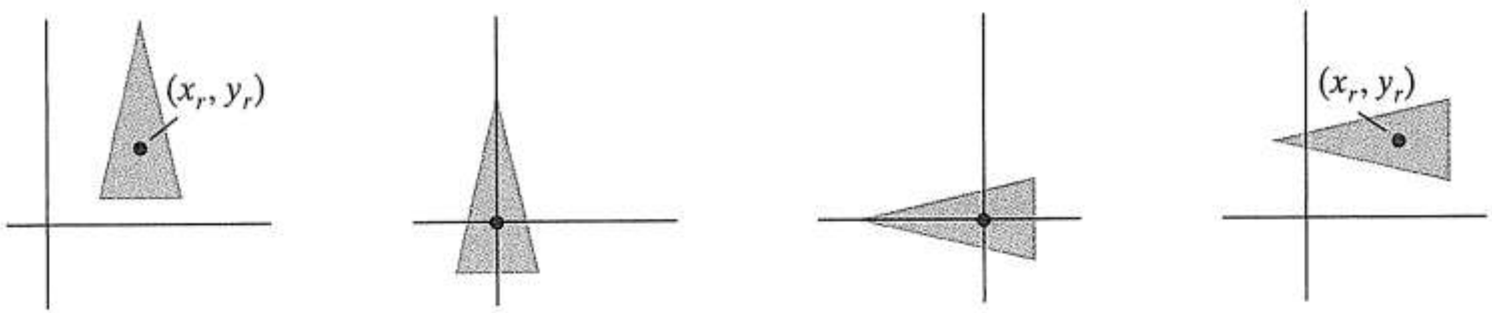
$$\begin{bmatrix} s_{2x} & 0 & 0 \\ 0 & s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{1x} & 0 & 0 \\ 0 & s_{1y} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{1x} \cdot s_{2x} & 0 & 0 \\ 0 & s_{1y} \cdot s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5\text{-}33)$$

or

$$\mathbf{S}(s_{2x}, s_{2y}) \cdot \mathbf{S}(s_{1x}, s_{1y}) = \mathbf{S}(s_{1x} \cdot s_{2x}, \ \ s_{1y} \cdot s_{2y}) \quad (5\text{-}34)$$

The resulting matrix in this case indicates that successive scaling operations are multiplicative. That is, if we were to triple the size of an object twice in succession, the final size would be nine times that of the original.

General 2-dimensional pivot-point rotation



(a) Original Position of Object and Pivot Point

(b) Translation of Object so that Pivot Point (x_r, y_r) is at Origin

(c) Rotation about Origin

(d) Translation of Object so that the Pivot Point is Returned to Position (x_r, y_r)

General Two-Dimensional Pivot-Point Rotation

When a graphics package provides only a rotate function with respect to the coordinate origin, we can generate a two-dimensional rotation about any other pivot point (x_r, y_r) by performing the following sequence of translate-rotate-translate operations.

(1) Translate the object so that the pivot-point position is moved to the coordinate origin.

(2) Rotate the object about the coordinate origin.

(3) Translate the object so that the pivot point is returned to its original position.

This transformation sequence is illustrated in Fig. 5-9. The composite transformation matrix for this sequence is obtained with the concatenation

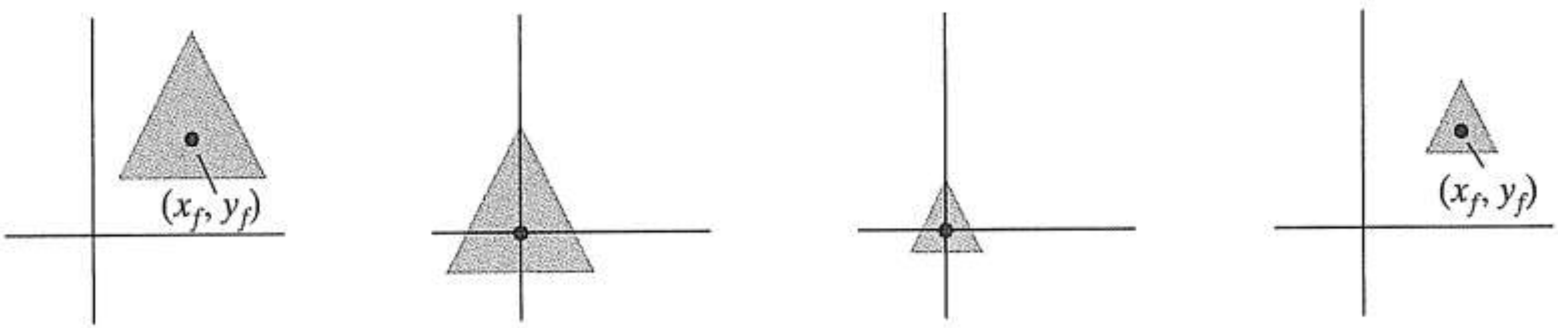
$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta) + y_r \sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta) - x_r \sin\theta \\ 0 & 0 & 1 \end{bmatrix} \tag{5-35}$$

which can be expressed in the form

$$\mathbf{T}(x_r, y_r) \cdot \mathbf{R}(\theta) \cdot \mathbf{T}(-x_r, -y_r) = \mathbf{R}(x_r, y_r, \theta) \tag{5-36}$$

General Two-dimensional Fixed point Scaling



(a)
Original Position of Object and Fixed Point

(b)
Translate Object so that Fixed Point (x_f, y_f) is at Origin

(c)
Scale Object with Respect to Origin

(d)
Translate Object so that the Fixed Point is Returned to Position (x_f, y_f)

General Two-dimensional Fixed point Scaling

Figure 5-10 illustrates a transformation sequence to produce a two-dimensional scaling with respect to a selected fixed position (x_f, y_f), when we have a function that can scale relative to the coordinate origin only. This sequence is

(1) Translate the object so that the fixed point coincides with the coordinate origin.

(2) Scale the object with respect to the coordinate origin.

(3) Use the inverse of the translation in step (1) to return the object to its original position.

Concatenating the matrices for these three operations produces the required scaling matrix:

$$\begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & x_f(1-s_x) \\ 0 & s_y & y_f(1-s_y) \\ 0 & 0 & 1 \end{bmatrix} \qquad (5\text{-}37)$$

or

$$\mathbf{T}(x_f, y_f) \cdot \mathbf{S}(s_x, s_y) \cdot \mathbf{T}(-x_f, -y_f) = \mathbf{S}(x_f, y_f, s_x, s_y) \qquad (5\text{-}38)$$

Matrix Concatenation Properties

Multiplication of matrices is associative. For any three matrices, $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$, the matrix product $\mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1$ can be performed by first multiplying $\mathbf{M}_3$ and $\mathbf{M}_2$ or by first multiplying $\mathbf{M}_2$ and $\mathbf{M}_1$:

$$\mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 = (\mathbf{M}_3 \cdot \mathbf{M}_2) \cdot \mathbf{M}_1 = \mathbf{M}_3 \cdot (\mathbf{M}_2 \cdot \mathbf{M}_1) \tag{5-40}$$

Therefore, depending upon the order in which the transformations are specified, we can construct a composite matrix either by multiplying from left-to-right

(premultiplying) or by multiplying from right-to-left (postmultiplying). Some graphics packages require that transformations be specified in the order in which they are to be applied. In that case, we would first invoke transformation $\mathbf{M}_1$, then $\mathbf{M}_2$, then $\mathbf{M}_3$. As each successive transformation routine is called, its matrix is concatenated on the left of the previous matrix product. Other graphics systems, however, postmultiply matrices, so that this transformation sequence would have to be invoked in the reverse order: the last transformation invoked (which is $\mathbf{M}_1$ for this example) is the first to be applied, and the first transformation that is called ($\mathbf{M}_3$ in this case) is the last to be applied.

Transformation products, on the other hand, may not be commutative. The matrix product $\mathbf{M}_2 \cdot \mathbf{M}_1$ is not equal to $\mathbf{M}_1 \cdot \mathbf{M}_2$, in general. This means that if we want to translate and rotate an object, we must be careful about the order in which the composite matrix is evaluated (Fig. 5-13). For some special cases—such as a sequence of transformations that are all of the same kind—the multiplication of transformation matrices is commutative. As an example, two successive rotations could be performed in either order and the final position would be the same. This commutative property holds also for two successive translations or two successive scalings. Another commutative pair of operations is rotation and uniform scaling $(s_x = s_y)$.

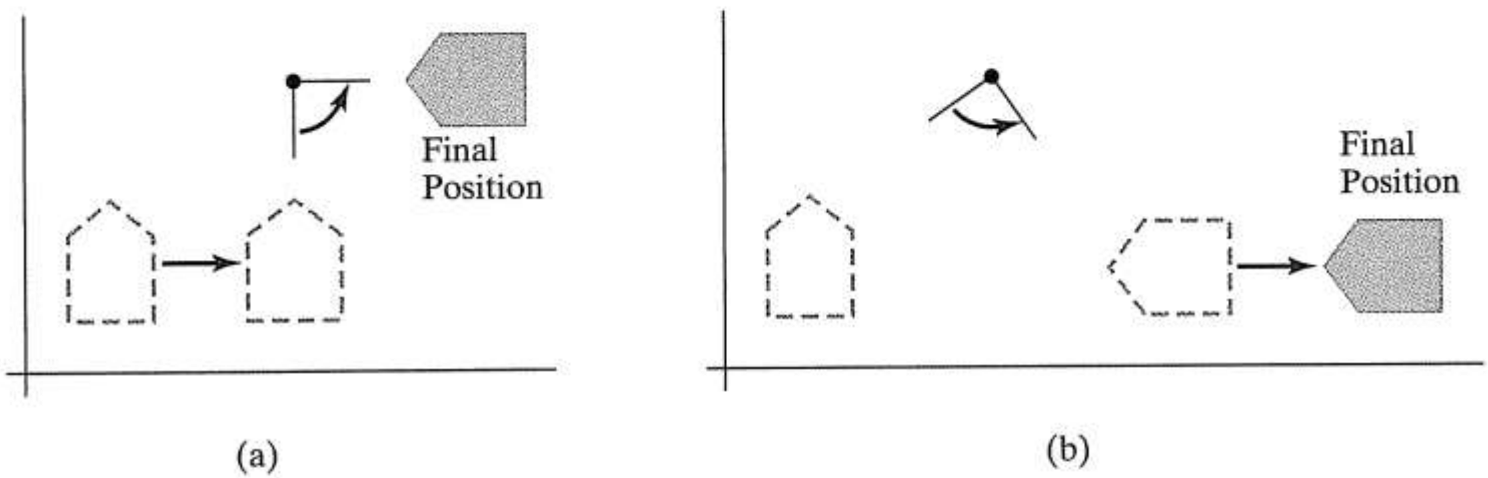


FIGURE 5-13 Reversing the order in which a sequence of transformations is performed may affect the transformed position of an object. In (a), an object is first translated in the x direction, then rotated counterclockwise through an angle of 45°. In (b), the object is first rotated 45° counterclockwise, then translated in the x direction.

Reflection

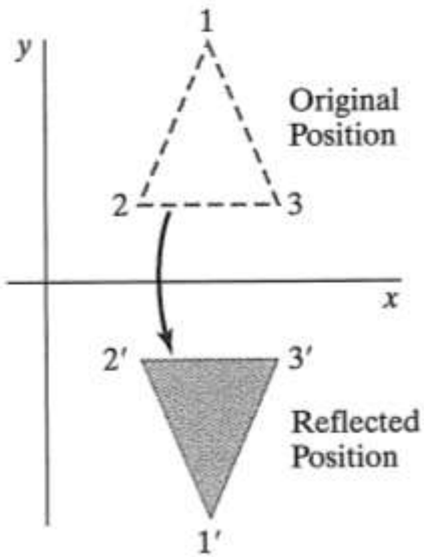


FIGURE 5-16 Reflection of an object about the x axis.

A transformation that produces a mirror image of an object is called a **reflection.** For a two-dimensional reflection, this image is generated relative to an **axis of reflection** by rotating the object 180° about the reflection axis. We can choose an axis of reflection in the xy plane or perpendicular to the xy plane. When the reflection axis is a line in the xy plane, the rotation path about this axis is in a plane perpendicular to the xy plane. For reflection axes that are perpendicular to the xy plane, the rotation path is in the xy plane. Following are examples of some common reflections.

Reflection about the line $y = 0$ (the x axis) is accomplished with the transformation matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5-52}$$

This transformation retains x values, but "flips" the y values of coordinate positions. The resulting orientation of an object after it has been reflected about the x axis is shown in Fig. 5-16. To envision the rotation transformation path for this reflection, we can think of the flat object moving out of the xy plane and rotating 180° through three-dimensional space about the x axis and back into the xy plane on the other side of the x axis.

A reflection about the line $x = 0$ (the y axis) flips x coordinates while keeping y coordinates the same. The matrix for this transformation is

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5-53}$$

Figure 5-17 illustrates the change in position of an object that has been reflected about the line $x = 0$. The equivalent rotation in this case is 180° through three-dimensional space about the y axis.

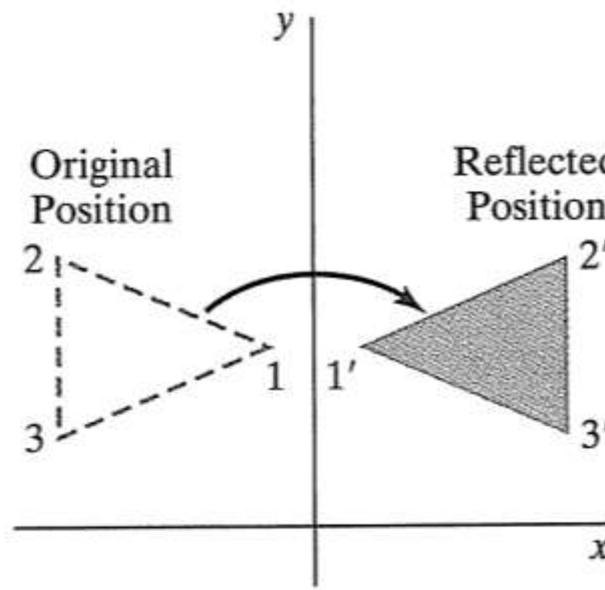


FIGURE 5-17 Reflection of an object about the y axis.

We flip both the x and y coordinates of a point by reflecting relative to an axis that is perpendicular to the xy plane and that passes through the coordinate origin. This reflection is sometimes referred to as a reflection relative to the coordinate origin, and it is equivalent to reflecting with respect to both coordinate axes. The matrix representation for this reflection is

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (5\text{-}54)$$

An example of reflection about the origin is shown in Fig. 5-18. The reflection matrix 5-54 is the same as the rotation matrix $\mathbf{R}(\theta)$ with $\theta = 180°$. We are simply rotating the object in the xy plane half a revolution about the origin.

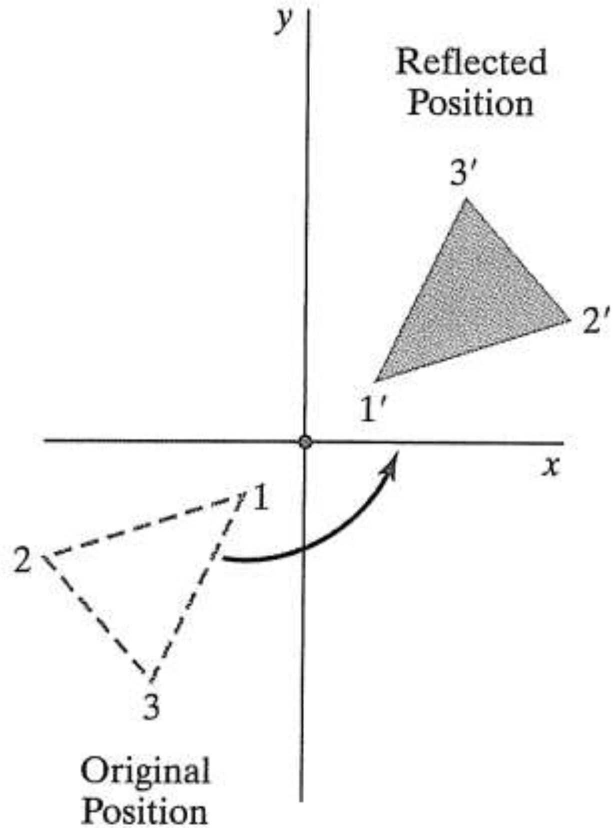


FIGURE 5-18 Reflection of an object relative to the coordinate origin. This transformation can be accomplished with a rotation in the xy plane about the coordinate origin.

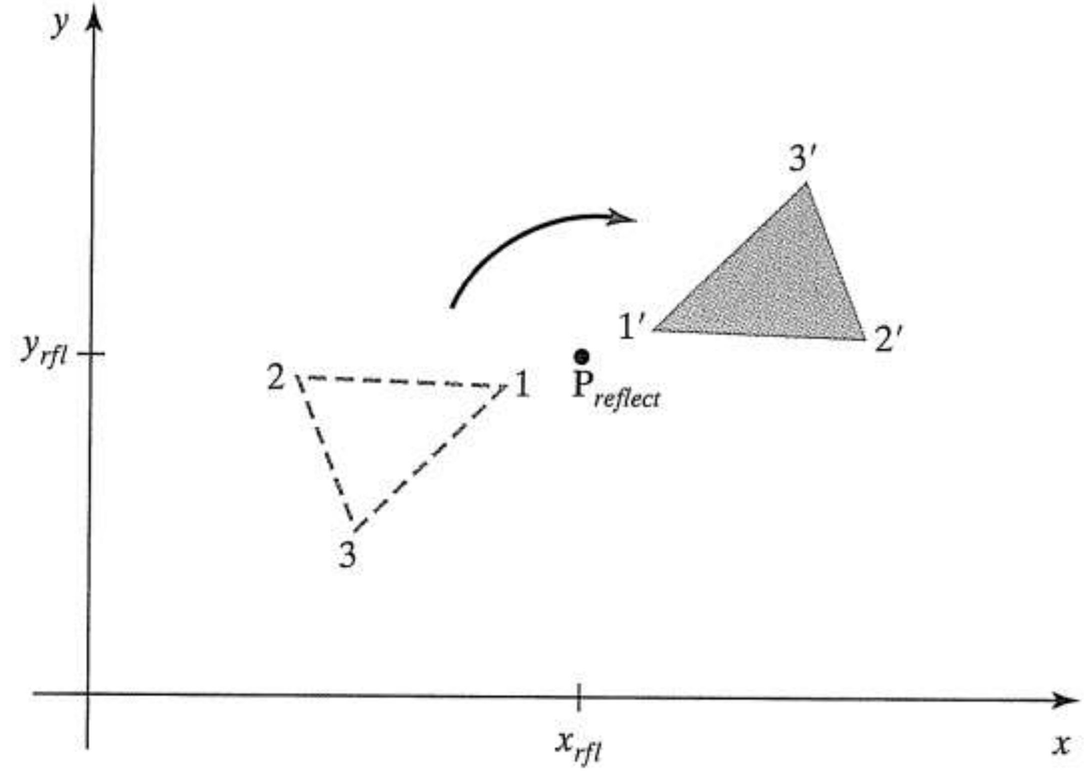


FIGURE 5-19 Reflection of an object relative to an axis perpendicular to the xy plane and passing through point $\mathbf{P}_{\text{reflect}}$.

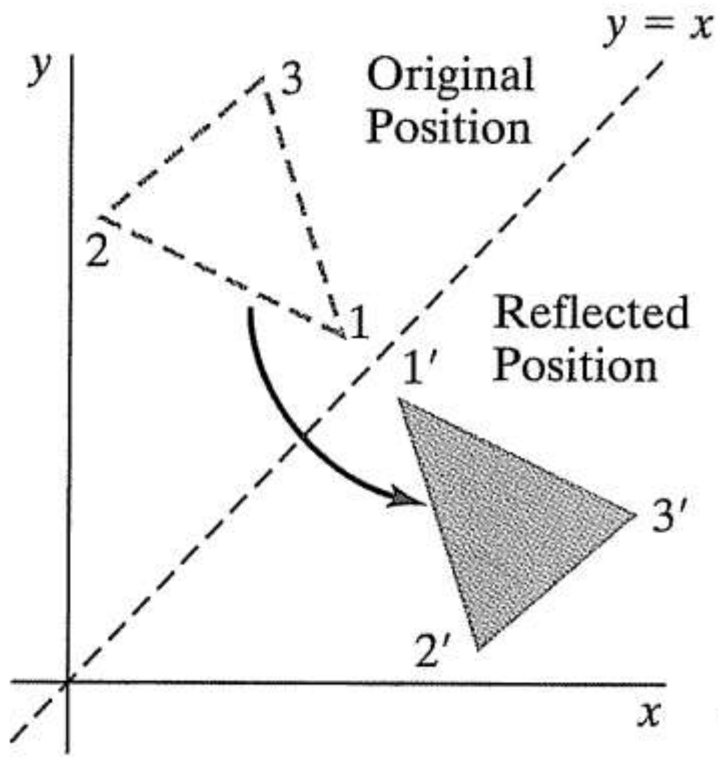


FIGURE 5-20 Reflection of an object with respect to the line $y = x$.

To obtain a transformation matrix for reflection about the diagonal $y = -x$, we could concatenate matrices for the transformation sequence: (1) clockwise rotation by 45°, (2) reflection about the y axis, and (3) counterclockwise rotation by 45°. The resulting transformation matrix is

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5-56}$$

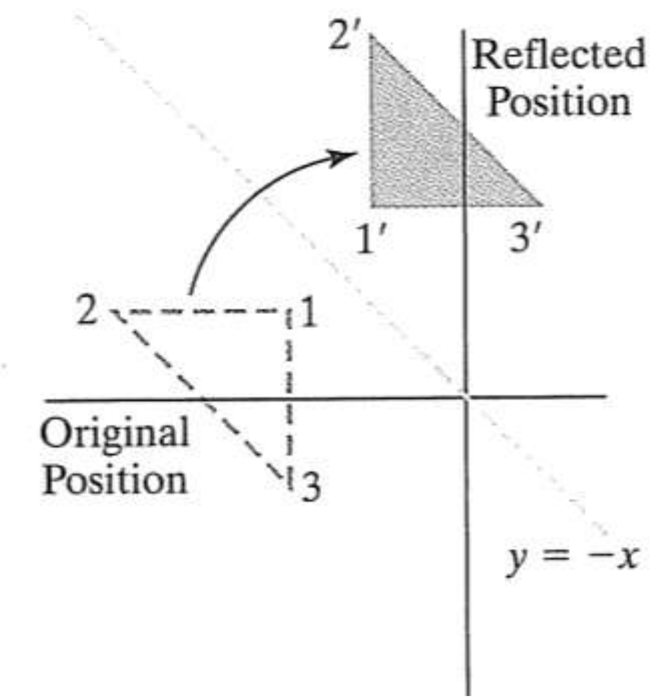


FIGURE 5-22 Reflection with respect to the line $y = -x$.

Shear

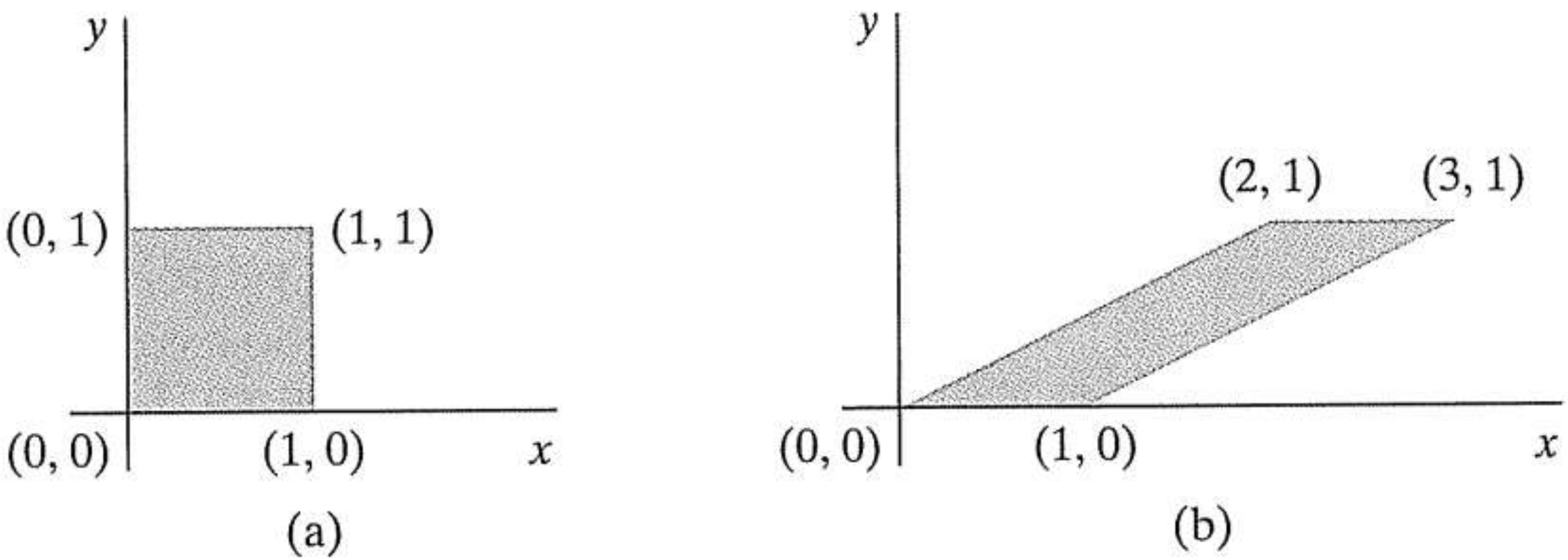


FIGURE 5-23 A unit square (a) is converted to a parallelogram (b) using the x-direction shear matrix 5-57 with $sh_x = 2$.

A transformation that distorts the shape of an object such that the transformed shape appears as if the object were composed of internal layers that had been caused to slide over each other is called a **shear.** Two common shearing transformations are those that shift coordinate x values and those that shift y values.

An x-direction shear relative to the x axis is produced with the transformation matrix

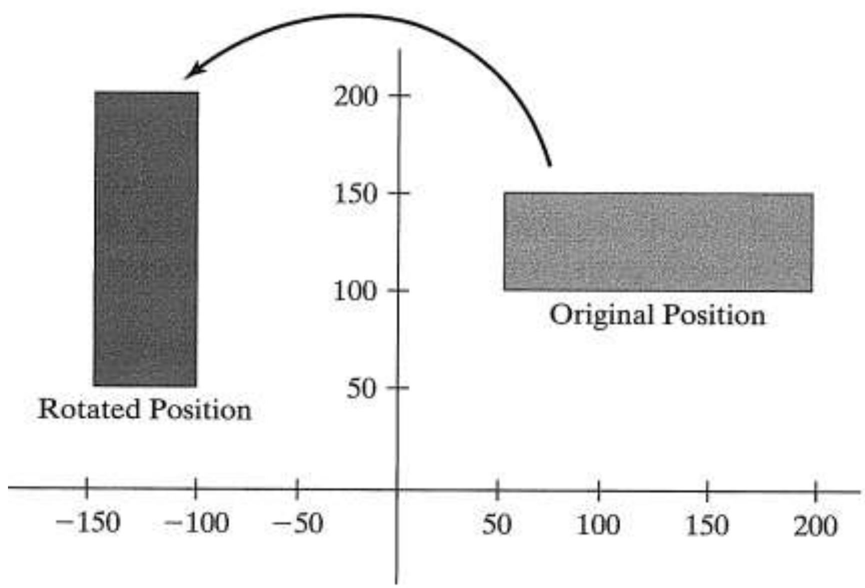
$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5-57}$$

which transforms coordinate positions as

$$x' = x + sh_x \cdot y, \qquad y' = y \tag{5-58}$$

Exercise:

Find the transformation matrix applied to the rectangle shown below (original position); what are the coordinates of the rotation point?



End Of Presentation