Geometric Transformation In Three Dimensional Space

Three Dimensional Translation

A position $\mathbf{P} = (x, y, z)$ in three-dimensional space is translated to a location $\mathbf{P}' = (x', y', z')$ by adding translation distances t_x, t_y, and t_z to the Cartesian coordinates of $\mathbf{P}$:

$$x' = x + t_x, \qquad y' = y + t_y, \qquad z' = z + t_z \tag{5-70}$$

Figure 5-34 illustrates three-dimensional point translation.

We can express these three-dimensional translation operations in matrix form as in Eq. 5-17. But now the coordinate positions, $\mathbf{P}$ and $\mathbf{P}'$, are represented in homogeneous coordinates with four-element column matrices, and the translation operator $\mathbf{T}$ is a 4 by 4 matrix:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5-71}$$

or

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{P} \tag{5-72}$$

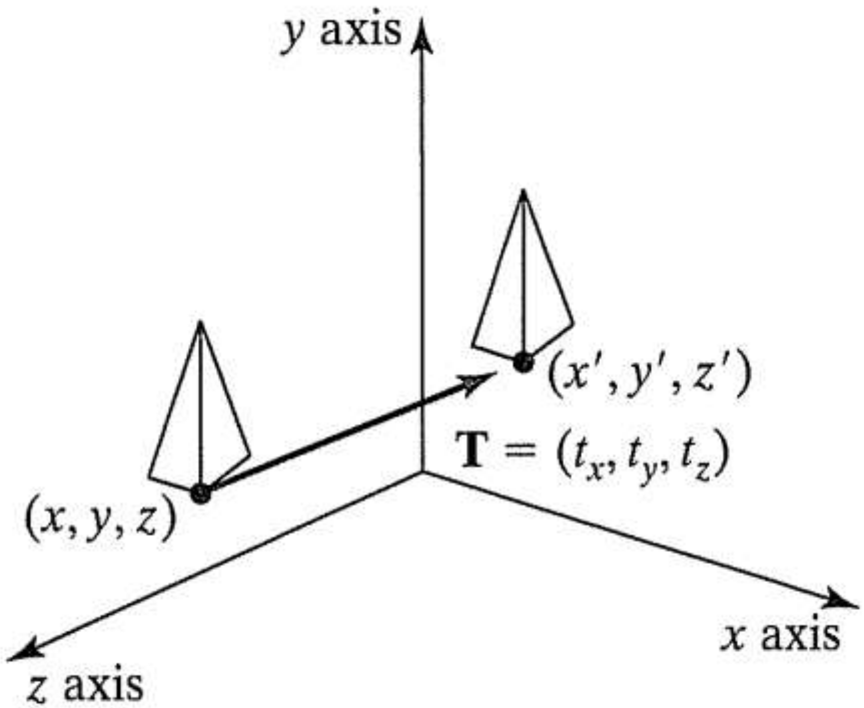


FIGURE 5-35 Shifting the position of a three-dimensional object using translation vector **T**.

Three Dimensional Rotation

We can rotate an object about any axis in space, but the easiest rotation axes to handle are those that are parallel to the Cartesian-coordinate axes. Also, we can use combinations of coordinate-axis rotations (along with appropriate translations) to specify a rotation about any other line in space. Therefore, we first consider the operations involved in coordinate-axis rotations, then we discuss the calculations needed for other rotation axes.

By convention, positive rotation angles produce counterclockwise rotations about a coordinate axis, assuming that we are looking in the negative direction along that coordinate axis (Fig. 5-36). This agrees with our earlier discussion of rotations in two dimensions, where positive rotations in the xy plane are counterclockwise about a pivot point (an axis that is parallel to the z axis).

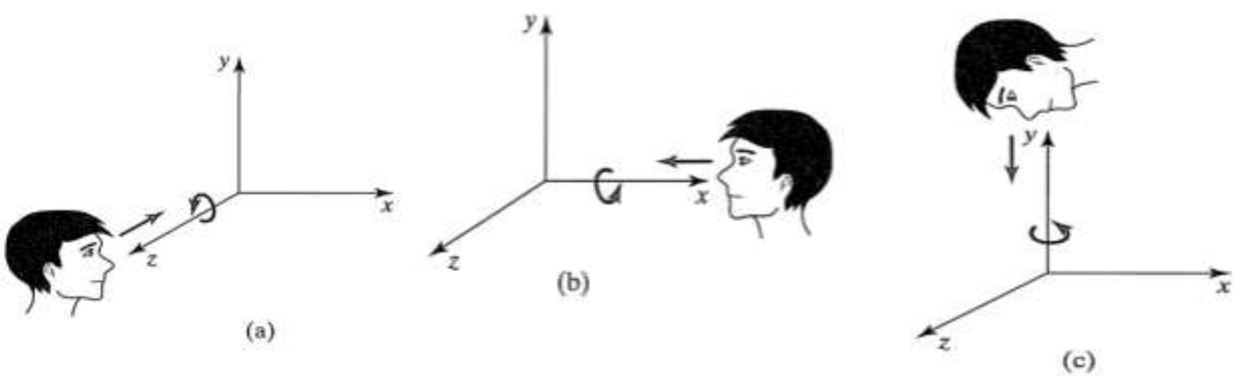


FIGURE 5-36 Positive rotations about a coordinate axis are counterclockwise, when looking along the positive half of the axis toward the origin.

Three-Dimensional Coordinate-Axis Rotations

The two-dimensional **z-axis rotation** equations are easily extended to three dimensions:

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta \qquad (5\text{-}73)$$
$$z' = z$$

Parameter θ specifies the rotation angle about the z axis, and z-coordinate values are unchanged by this transformation. In homogeneous-coordinate form, the three-dimensional z-axis rotation equations are

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (5\text{-}74)$$

which we can write more compactly as

$$\mathbf{P}' = \mathbf{R}_z(\theta) \cdot \mathbf{P} \qquad (5\text{-}75)$$

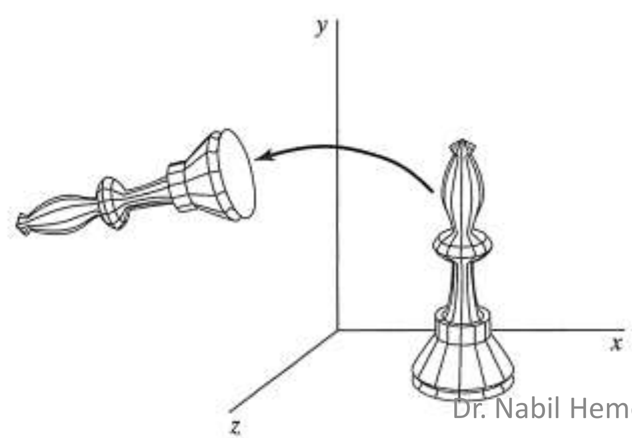


FIGURE 5-37 Rotation of an object about the z axis.

Transformation equations for rotations about the other two coordinate axes can be obtained with a cyclic permutation of the coordinate parameters x, y, and z in Eqs. 5-73:

$$x \rightarrow y \rightarrow z \rightarrow x \tag{5-76}$$

Thus, to obtain the x-axis and y-axis rotation transformations, we cyclically replace x with y, y with z, and z with x, as illustrated in Fig. 5-38.

Substituting permutations 5-76 into Eqs. 5-73, we get the equations for an **x-axis rotation**:

$$\begin{aligned} y' &= y\cos\theta - z\sin\theta \\ z' &= y\sin\theta + z\cos\theta \\ x' &= x \end{aligned} \tag{5-77}$$

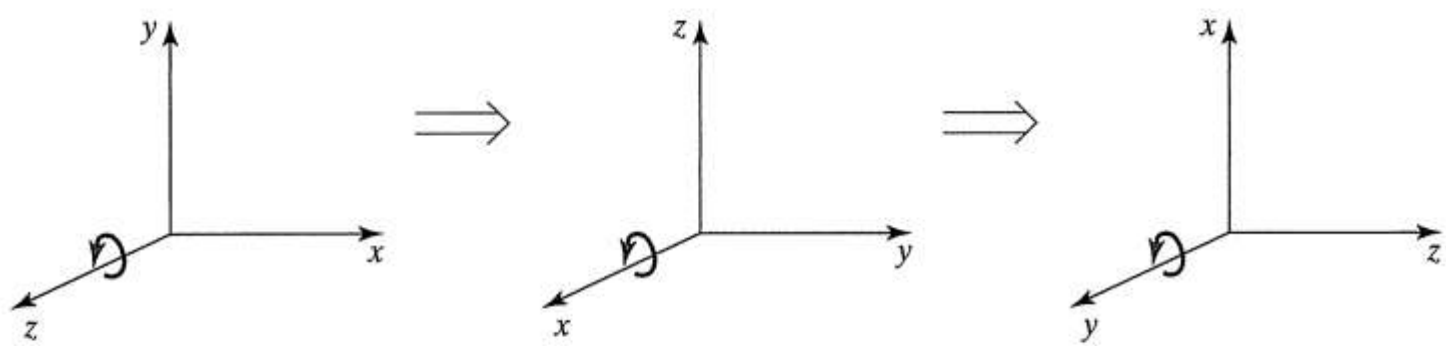


FIGURE 5-38 Cyclic permutation of the Cartesian-coordinate axes to produce the three sets of coordinate-axis rotation equations.

A cyclic permutation of coordinates in Eqs. 5-77 gives us the transformation equations for a ***y*-axis rotation:**

$$z' = z\cos\theta - x\sin\theta$$
$$x' = z\sin\theta + x\cos\theta$$
$$y' = y$$

An example of y-axis rotation is shown in Fig. 5-40.

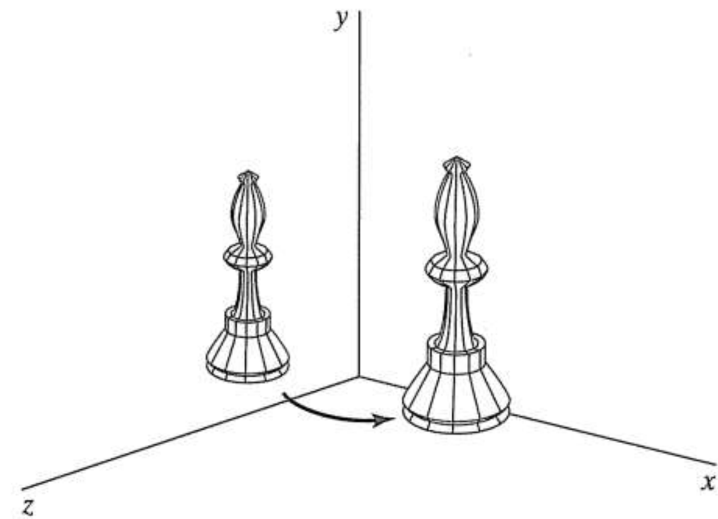


FIGURE 5-40 Rotation of an object about the y axis.

An inverse three-dimensional rotation matrix is obtained in the same way as the inverse rotations in two dimensions. We just replace the angle θ with $-\theta$. Negative values for rotation angles generate rotations in a clockwise direction, and the identity matrix is produced when we multiply any rotation matrix 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 forming its transpose ($\mathbf{R}^{-1} = \mathbf{R}^{T}$).

General Three-dimensional Rotation

A rotation matrix for any axis that does not coincide with a coordinate axis can be set up as a composite transformation involving combinations of translations and the coordinate-axis rotations. We first move the designated rotation axis onto one of the coordinate axes. Then we apply the appropriate rotation matrix for that coordinate axis. The last step in the transformation sequence is to return the rotation axis to its original position.

In the special case where an object is to be rotated about an axis that is parallel to one of the coordinate axes, we attain the desired rotation with the following transformation sequence.

(1) Translate the object so that the rotation axis coincides with the parallel coordinate axis.

(2) Perform the specified rotation about that axis.

(3) Translate the object so that the rotation axis is moved back to its original position.

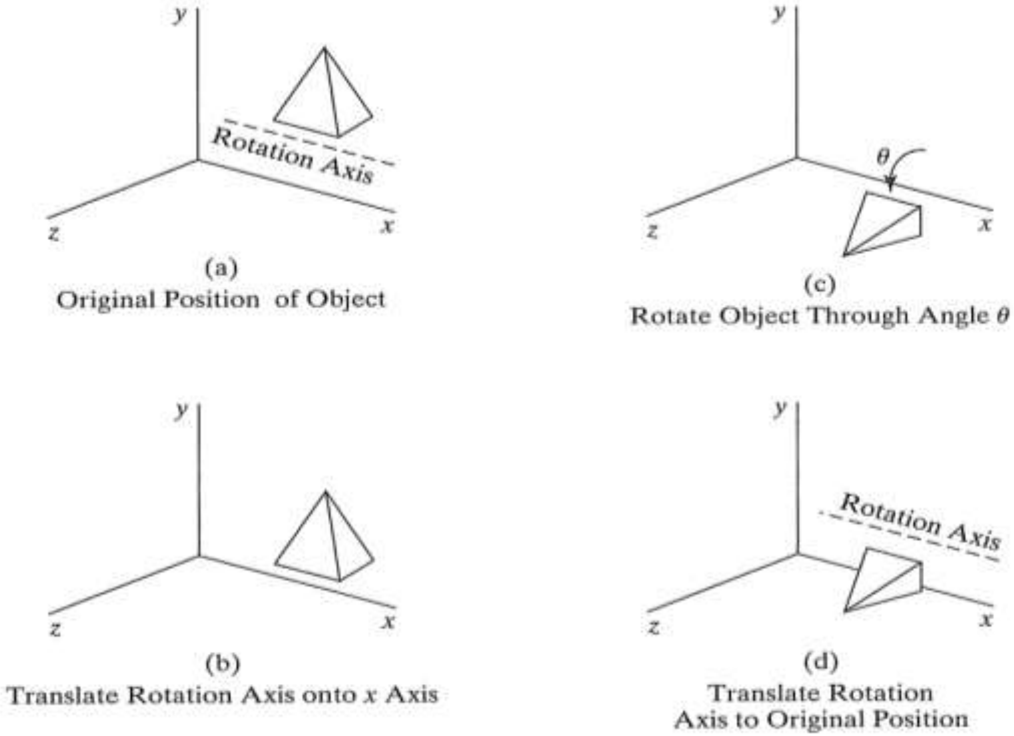


FIGURE 5-41 Sequence of transformations for rotating an object about an axis that is parallel to the x axis.

The steps in this sequence are illustrated in Fig. 5-41. A coordinate position **P** is transformed with the sequence shown in this figure as

$$\mathbf{P}' = \mathbf{T}^{-1} \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T} \cdot \mathbf{P} \tag{5-79}$$

where the composite rotation matrix for the transformation is

$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R}_x(\theta) \cdot \mathbf{T} \tag{5-80}$$

This composite matrix is of the same form as the two-dimensional transformation sequence for rotation about an axis that is parallel to the z axis (a pivot point that is not at the coordinate origin).

When an object is to be rotated about an axis that is not parallel to one of the coordinate axes, we must perform some additional transformations. In this case, we also need rotations to align the rotation axis with a selected coordinate axis and then to bring the rotation axis back to its original orientation. Given the specifications for the rotation axis and the rotation angle, we can accomplish the required rotation in five steps:

(1) Translate the object so that the rotation axis passes through the coordinate origin.

(2) Rotate the object so that the axis of rotation coincides with one of the coordinate axes.

(3) Perform the specified rotation about the selected coordinate axis.

(4) Apply inverse rotations to bring the rotation axis back to its original orientation.

(5) Apply the inverse translation to bring the rotation axis back to its original spatial position.

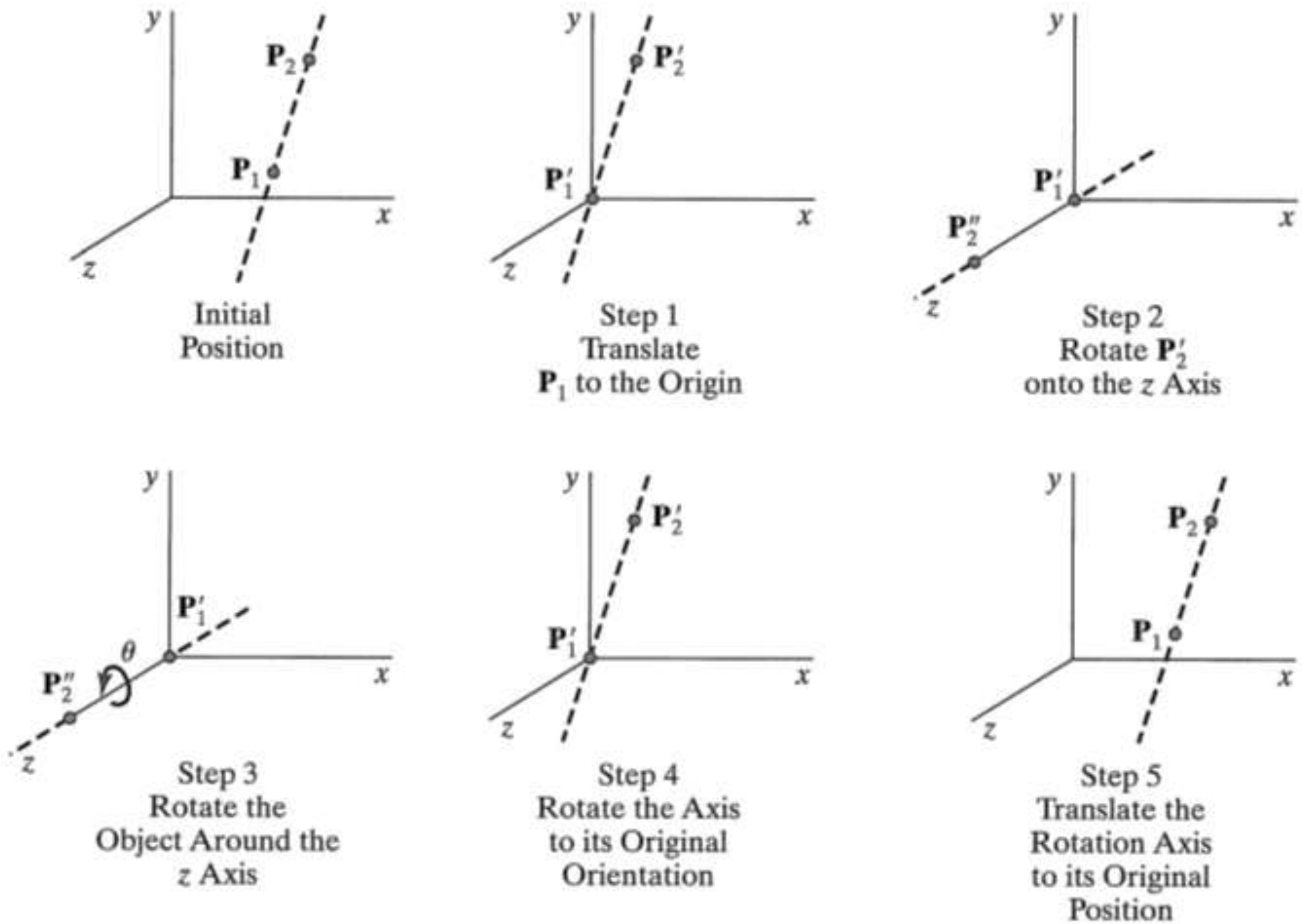


FIGURE 5-42 Five transformation steps for obtaining a composite matrix for rotation about an arbitrary axis, with the rotation axis projected onto the z axis.

Three dimensional Scaling

The matrix expression for the three-dimensional scaling transformation of a position $\mathbf{P} = (x, y, z)$ relative to the coordinate origin is a simple extension of two-dimensional scaling. We just include the parameter for z-coordinate scaling in the

transformation matrix:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (5\text{-}110)$$

The three-dimensional scaling transformation for a point position can be represented as

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P} \qquad (5\text{-}111)$$

where scaling parameters s_x, s_y, and s_z are assigned any positive values. Explicit expressions for the scaling transformation relative to the origin are

$$x' = x \cdot s_x, \qquad y' = y \cdot s_y, \qquad z' = z \cdot s_z \qquad (5\text{-}112)$$

Scaling an object with transformation 5-110 changes the position of the object relative to the coordinate origin. A parameter value greater than 1 moves a point farther from the origin in the corresponding coordinate direction. Similarly, a parameter value less than 1 moves a point closer to the origin in that coordinate direction. Also, if the scaling parameters are not all equal, relative dimensions of a transformed object are changed. We preserve the original shape of an object with a *uniform scaling*: $s_x = s_y = s_z$. The result of scaling an object uniformly with each scaling parameter set to 2 is illustrated in Fig. 5-50.

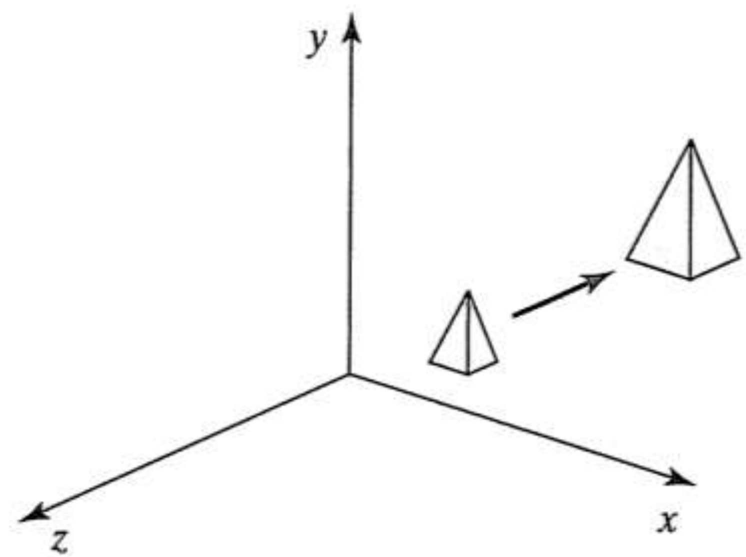


FIGURE 5-50 Doubling the size of an object with transformation 5-110 also moves the object farther from the origin.

Since some graphics packages provide only a routine that scales relative to the coordinate origin, we can always construct a scaling transformation with respect to any selected *fixed position* (x_f, y_f, z_f) using the following transformation sequence.

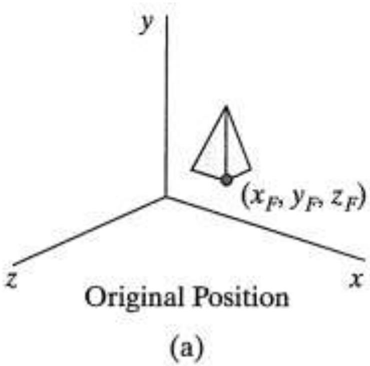
(1) Translate the fixed point to the origin.

(2) Apply the scaling transformation relative to the coordinate origin using Eq. 5-110.

(3) Translate the fixed point back to its original position.

This sequence of transformations is demonstrated in Fig. 5-51. The matrix representation for an arbitrary fixed-point scaling can then be expressed as the concatenation of these translate-scale-translate transformations:

$$\mathbf{T}(x_f, y_f, z_f)\cdot\mathbf{S}(s_x, s_y, s_z)\cdot\mathbf{T}(-x_f, -y_f, -z_f) = \begin{bmatrix} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5\text{-}113)$$



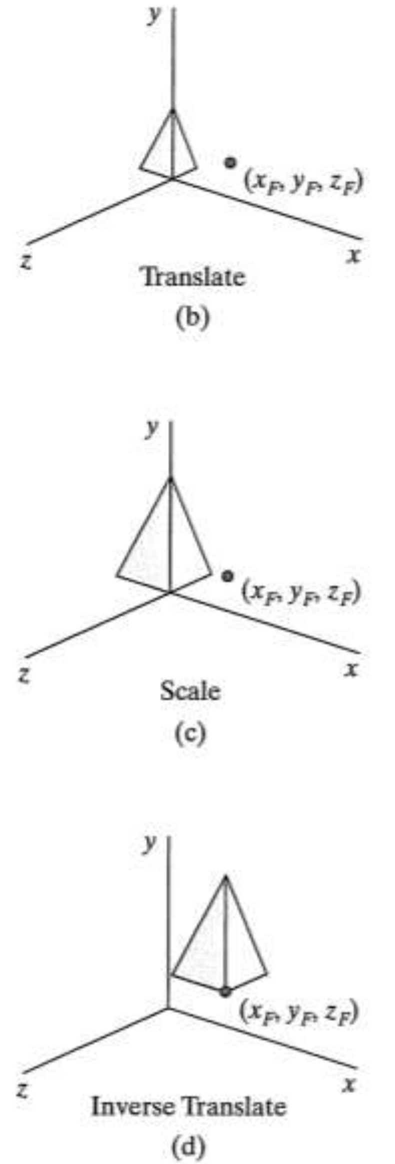


FIGURE 5-51 A sequence of transformations for scaling an object relative to a selected fixed point, using Eq. 5-110.

Three-Dimensional Reflections

A reflection in a three-dimensional space can be performed relative to a selected *reflection axis* or with respect to a *reflection plane*. In general, three-dimensional reflection matrices are set up similarly to those for two dimensions. Reflections relative to a given axis are equivalent to 180° rotations about that axis. Reflections with respect to a plane are equivalent to 180° rotations in four-dimensional space. When the reflection plane is a coordinate plane (xy, xz, or yz), we can think of the transformation as a conversion between a left-handed frame and a right-handed frame (Appendix A).

An example of a reflection that converts coordinate specifications from a right-handed system to a left-handed system (or vice versa) is shown in Fig. 5-52. This transformation changes the sign of z coordinates, leaving the values for the x and y coordinates unchanged. The matrix representation for this reflection relative to the xy plane is

$$M_{\text{zreflect}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5\text{-}114)$$

Transformation matrices for inverting x coordinates or y coordinates are defined similarly, as reflections relative to the yz plane or to the xz plane, respectively. Reflections about other planes can be obtained as a combination of rotations and coordinate-plane reflections.

5-16 AFFINE TRANSFORMATIONS

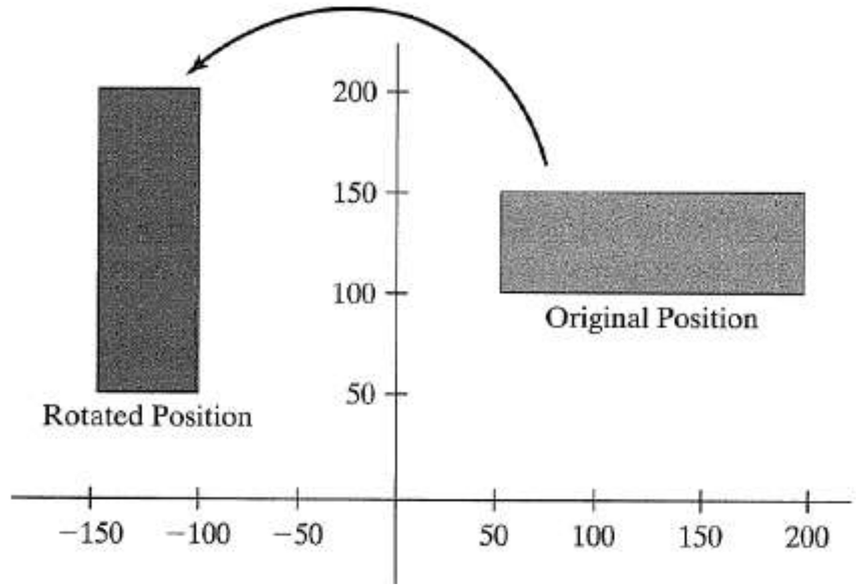
A coordinate transformation of the form

$$\begin{aligned} x' &= a_{xx}x + a_{xy}y + a_{xz}z + b_x \\ y' &= a_{yx}x + a_{yy}y + a_{yz}z + b_y \\ z' &= a_{zx}x + a_{zy}y + a_{zz}z + b_z \end{aligned} \tag{5-117}$$

is called an **affine transformation.** Each of the transformed coordinates x', y', and z' is a linear function of the original coordinates x, y, and z, and parameters a_{ij} and b_k are constants determined by the transformation type. Affine transformations (in two dimensions, three dimensions, or higher dimensions) have the general properties that parallel lines are transformed into parallel lines and finite points map to finite points.

Translation, rotation, scaling, reflection, and shear are examples of affine transformations. We can always express any affine transformation as some composition of these five transformations. Another example of an affine transformation is the conversion of coordinate descriptions for a scene from one reference system to another, since this transformation can be described as a combination of translation and rotation. An affine transformation involving only translation, rotation, and reflection preserves angles and lengths, as well as parallel lines. For each of these three transformations, line lengths and the angle between any two lines remain the same after the transformation.

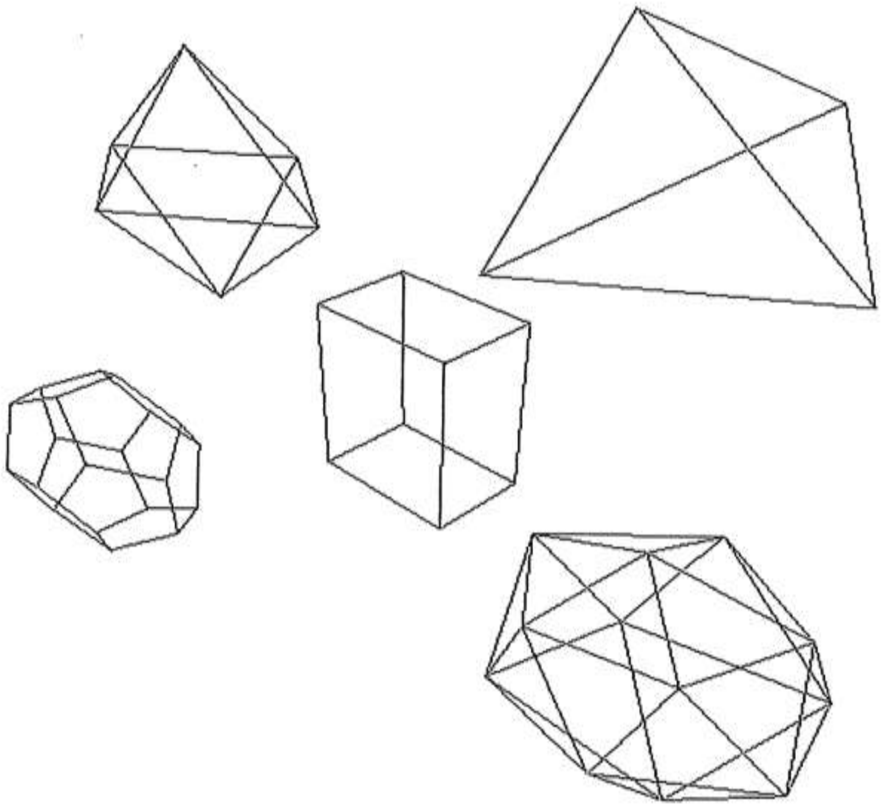
Find the transformation matrix for the rectangle shown below:



Three-Dimensional Object Representation

Representation schemes for solid objects are often divided into two broad categories, although not all representations fall neatly into one or the other of these two categories. **Boundary representations (B-reps)** describe a three-dimensional object as a set of surfaces that separate the object interior from the environment. Typical examples of boundary representations are polygon facets and spline patches. **Space-partitioning representations** are used to describe interior properties, by partitioning the spatial region containing an object into a set of small, nonoverlapping, contiguous solids (usually cubes).

Boundary representation



POLYHEDRA

The most commonly used boundary representation for a three-dimensional graphics object is a set of surface polygons that enclose the object interior. Many graphics systems store all object descriptions as sets of surface polygons. This simplifies and speeds up the surface rendering and display of objects, since all surfaces are described with linear equations. For this reason, polygon descriptions are often referred to as *standard graphics objects*. In some cases, a polygonal representation is the only one available, but many packages also allow object surfaces to be described with other schemes, such as spline surfaces, which are usually converted to polygonal representations for processing through the viewing pipeline.

To describe an object as a set of polygon facets, we give the list of vertex coordinates for each polygon section over the object surface. The vertex coordinates and edge information for the surface sections are then stored in tables (Section 3-15), along with other information such as the surface normal vector for each polygon. Some graphics packages provide routines for generating a polygon-surface mesh as a set of triangles or quadrilaterals. This allows us to describe a large section of

8-4 QUADRIC SURFACES

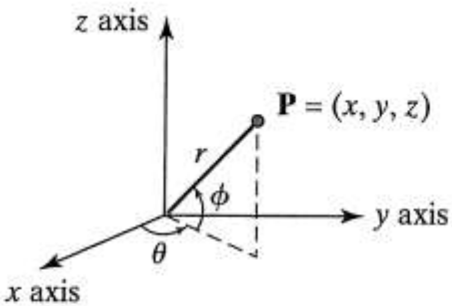


FIGURE 8-2 Parametric coordinate position (r, θ, ϕ) on the surface of a sphere with radius r.

A frequently used class of objects are the *quadric surfaces*, which are described with second-degree equations (quadratics). They include spheres, ellipsoids, tori, paraboloids, and hyperboloids. Quadric surfaces, particularly spheres and ellipsoids, are common elements of graphics scenes, and routines for generating these surfaces are often available in graphics packages. Also, quadric surfaces can be produced with rational spline representations.

Sphere

In Cartesian coordinates, a spherical surface with radius r centered on the coordinate origin is defined as the set of points (x, y, z) that satisfy the equation:

$$x^2 + y^2 + z^2 = r^2 \tag{8-1}$$

We can also describe the spherical surface in parametric form, using latitude and longitude angles (Fig. 8-2):

$$\begin{aligned} x &= r\cos\phi\cos\theta, & -\pi/2 \le \phi \le \pi/2 \\ y &= r\cos\phi\sin\theta, & -\pi \le \theta \le \pi \\ z &= r\sin\phi \end{aligned} \tag{8-2}$$

Ellipsoid

An ellipsoidal surface can be described as an extension of a spherical surface, where the radii in three mutually perpendicular directions can have different values (Fig. 8-4). The Cartesian representation for points over the surface of an ellipsoid centered on the origin is

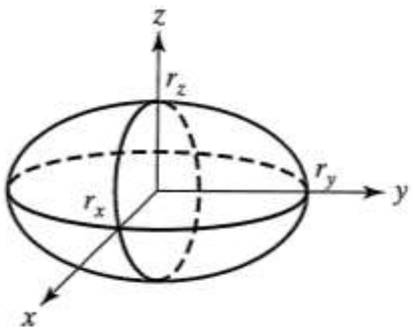


FIGURE 8-4 An ellipsoid with radii r_x, r_y, and r_z, centered on the coordinate origin.

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1 \tag{8-3}$$

And a parametric representation for the ellipsoid in terms of the latitude angle ϕ and the longitude angle θ in Fig. 8-2 is

$$\begin{aligned} x &= r_x \cos\phi \cos\theta, \qquad -\pi/2 \le \phi \le \pi/2 \\ y &= r_y \cos\phi \sin\theta, \qquad -\pi \le \theta \le \pi \\ z &= r_z \sin\phi \end{aligned} \tag{8-4}$$

END OF PRESENTATION