Table 2.3 ALU Operations

Operations on A	*Operations on B*
—	Multiply by w_5
Shift right	—
—	Add w_4*A
Shift down	—
—	Add w_1*A
Shift left	—
—	Add w_2*A
Shift left	—
—	Add w_3*A
Shift up	—
—	Add w_6*A
Shift up	—
—	Add w_9*A
Shift right	—
—	Add w_8*A
Shift right	—
—	Add w_7*A
Shift left	—
Shift down	—

happens in a single pixel of B by considering how a mask would have to be shifted in order to produce the result of Eq. (2.4-5) in that location. The first operation on B produces w_5 multiplied by the pixel value at that location. Since that value is z_5, we have w_5z_5 after this operation. The first shift to the right brings the neighbor with value z_4 (see Fig. 2.15a) over that location. The next operation multiplies z_4 by w_4 and adds the result to the location of the first step. So at this point the result is $w_4z_4 + w_5z_5$ at the location in question. The next shift on A and ALU operation on B produce $w_1z_1 + w_4z_4 + w_5z_5$ at that location, and so on. The operations are done in parallel for all locations in B, so this procedure takes place simultaneously at the other locations in that frame buffer. In most ALUs, the operation of multiplying an image by a constant (say, $w_i * A$) followed by an ADD is done in one frame time. Thus the ALU implementation of Eq. (2.4-5) for an entire image takes on the order of nine frame times (9/30 sec). For an $n \times m$ mask it would take on the order of nm frame times.

2.5 IMAGING GEOMETRY

In the following discussion we present several important transformations used in imaging, derive a camera model, and treat the stereo imaging problem in some detail.

2.5.1 Some Basic Transformations

The material in this section deals with development of a unified representation for problems such as image rotation, scaling, and translation. All transformations are expressed in a three-dimensional (3-D) Cartesian coordinate system in which a point has coordinates denoted (X, Y, Z). In cases involving 2-D images, we adhere to our previous convention of lowercase representation (x, y) to denote the coordinates of a pixel. Referring to (X, Y, Z) as the *world coordinates* of a point is common terminology.

Translation

Suppose that the task is to translate a point with coordinates (X, Y, Z) to a new location by using displacements (X_0, Y_0, Z_0). The translation is easily accomplished by using the equations:

$$\begin{aligned} X^* &= X + X_0 \\ Y^* &= Y + Y_0 \\ Z^* &= Z + Z_0 \end{aligned} \tag{2.5-1}$$

where (X^*, Y^*, Z^*) are the coordinates of the new point. Equation (2.5-1) may be expressed in matrix form by writing

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.5-2}$$

It is often useful to concatenate several transformations to produce a composite result, such as translation, followed by scaling and then rotation. The use of square matrices simplifies the notational representation of this process considerably. With this in mind, Eq. (2.5-2) can be written as follows:

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.5-3}$$

In terms of the values of X^*, Y^*, and Z^*, Eqs. (2.5-2) and (2.5-3) are equivalent.

Throughout this section, we use the unified matrix representation

$$\mathbf{v}^* = \mathbf{A}\mathbf{v} \tag{2.5-4}$$

where $\mathbf{A}$ is a 4×4 transformation matrix, $\mathbf{v}$ is the column vector containing the original coordinates,

$$\mathbf{v} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.5-5}$$

and $\mathbf{v}^*$ is a column vector whose components are the transformed coordinates

$$\mathbf{v}^* = \begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} \tag{2.5-6}$$

With this notation, the matrix used for translation is

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-7}$$

and the translation process is accomplished by using Eq. (2.5-4), so that $\mathbf{v}^* = \mathbf{Tv}$.

Scaling

Scaling by factors S_x, S_y, and S_z along the X, Y, and Z axes is given by the transformation matrix

$$\mathbf{S} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-8}$$

Rotation

The transformations used for 3-D rotation are inherently more complex than the transformations discussed thus far. The simplest form of these transformations is for rotation of a point about the coordinate axes. To rotate a point about another arbitrary point in space requires three transformations: the first

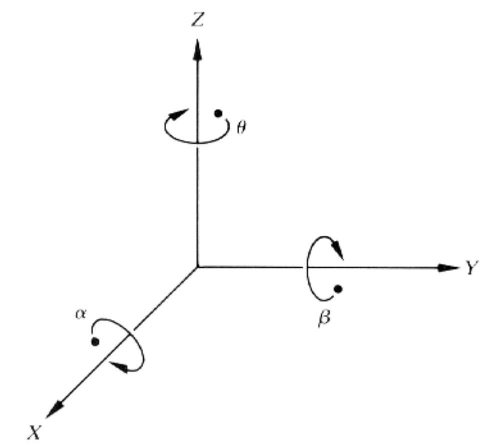

Figure 2.16 *Rotation of a point about each of the coordinate axes. Angles are measured clockwise when looking along the rotation axis toward the origin.*

translates the arbitrary point to the origin, the second performs the rotation, and the third translates the point back to its original position.

With reference to Fig. 2.16, rotation of a point about the Z coordinate axis by an angle θ is achieved by using the transformation

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-9}$$

The rotation angle θ is measured clockwise when looking at the origin from a point on the $+Z$ axis. This transformation affects only the values of X and Y coordinates.

Rotation of a point about the X axis by an angle α is performed by using the transformation

$$\mathbf{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-10}$$

Finally, rotation of a point about the Y axis by an angle β is achieved by using the transformation

$$\mathbf{R}_\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-11}$$

Concatenation and inverse transformations

The application of several transformations can be represented by a single 4 × 4 transformation matrix. For example, translation, scaling, and rotation about the Z axis of a point $\mathbf{v}$ is given by

$$\begin{aligned} \mathbf{v}^* &= \mathbf{R}_\theta(\mathbf{S}(\mathbf{T}\mathbf{v})) \\ &= \mathbf{A}\mathbf{v} \end{aligned} \tag{2.5-12}$$

where $\mathbf{A}$ is the 4 × 4 matrix $\mathbf{A} = \mathbf{R}_\theta\mathbf{S}\mathbf{T}$. These matrices generally do not commute, so the order of application is important.

Although the discussion thus far has been limited to transformations of a single point, the same ideas extend to transforming a set of m points simultaneously by using a single transformation. With reference to Eq. (2.5-5), let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ represent the coordinates of m points. For a 4 × m matrix $\mathbf{V}$ whose columns are these column vectors, the simultaneous transformation of all these points by a 4 × 4 transformation matrix $\mathbf{A}$ is given by

$$\mathbf{V}^* = \mathbf{A}\mathbf{V}. \tag{2.5-13}$$

The resulting matrix $\mathbf{V}^*$ is 4 × m. Its ith column, $\mathbf{v}_i^*$, contains the coordinates of the transformed point corresponding to $\mathbf{v}_i$.

Many of the transformations discussed above have inverse matrices that perform the opposite transformation and can be obtained by inspection. For example, the inverse translation matrix is

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-14}$$

Similarly, the inverse rotation matrix $\mathbf{R}_\theta^{-1}$ is

$$\mathbf{R}_\theta^{-1} = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) & 0 & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-15}$$

The inverses of more complex transformation matrices are usually obtained by numerical techniques.

2.5.2 Perspective Transformations

A perspective transformation (also called an imaging transformation) projects 3-D points onto a plane. Perspective transformations play a central role in image processing because they provide an approximation to the manner in which an image is formed by viewing a 3-D world. These transformations are fundamentally different from those discussed in Section 2.5.1 because they are nonlinear in that they involve division by coordinate values.

Figure 2.17 shows a model of the image formation process. The camera coordinate system (x, y, z) has the image plane coincident with the xy plane and the optical axis (established by the center of the lens) along the z axis. Thus the center of the image plane is at the origin, and the center of the lens is at coordinates $(0, 0, \lambda)$. If the camera is in focus for distant objects, λ is the *focal length* of the lens. Here the assumption is that the camera coordinate system is aligned with the world coordinate system (X, Y, Z). We remove this restriction in Section 2.5.3.

Let (X, Y, Z) be the world coordinates of any point in a 3-D scene, as shown in Fig. 2.17. We assume throughout the following discussion that $Z > \lambda$; that is, all points of interest lie in front of the lens. The first step is to obtain a relationship that gives the coordinates (x, y) of the projection of the point (X, Y, Z) onto the image plane. This is easily accomplished by the use of similar triangles. With reference to Fig. 2.17,

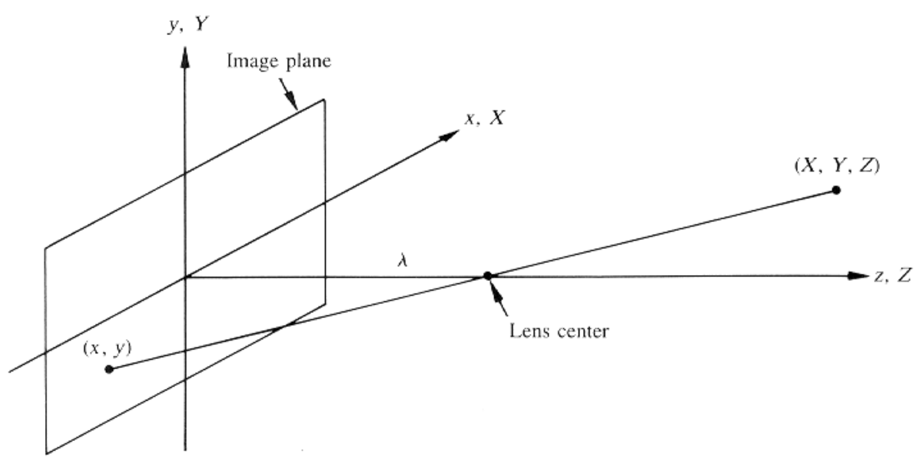

Figure 2.17 *Basic model of the imaging process. The camera coordinate system (x, y, z) is aligned with the world coordinate system (X, Y, Z).*

of similar triangles. With reference to Fig. 2.17,

$$\frac{x}{\lambda} = -\frac{X}{Z - \lambda} = \frac{X}{\lambda - Z} \tag{2.5-16}$$

and

$$\frac{y}{\lambda} = -\frac{Y}{Z - \lambda} = \frac{Y}{\lambda - Z} \tag{2.5-17}$$

where the negative signs in front of X and Y indicate that image points are actually inverted, as the geometry of Fig. 2.17 shows.

The image-plane coordinates of the projected 3-D point follow directly from Eqs. (2.5-16) and (2.5-17):

$$x = \frac{\lambda X}{\lambda - Z} \tag{2.5-18}$$

and

$$y = \frac{\lambda Y}{\lambda - Z} \tag{2.5-19}$$

These equations are nonlinear because they involve division by the variable Z. Although we could use them directly as shown, it is often convenient to express them in linear matrix form, as in Section 2.5.1 for rotation, translation, and scaling. This is easily accomplished by using homogeneous coordinates.

The homogeneous coordinates of a point with Cartesian coordinates (X, Y, Z) are defined as (kX, kY, kZ, k), where k is an arbitrary, nonzero constant. Clearly, conversion of homogeneous coordinates back to Cartesian coordinates is accomplished by dividing the first three homogeneous coordinates by the fourth. A point in the Cartesian world coordinate system may be expressed in vector form as

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.5-20}$$

and its homogeneous counterpart is given by

$$\mathbf{w}_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} \tag{2.5-21}$$

If we define the *perspective transformation matrix* as

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \tag{2.5-22}$$

the product $\mathbf{Pw}_h$ yields a vector denoted $\mathbf{c}_h$:

$$\begin{aligned} \mathbf{c}_h &= \mathbf{Pw}_h \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} \\ &= \begin{bmatrix} kX \\ kY \\ kZ \\ \frac{-kZ}{\lambda} + k \end{bmatrix} \end{aligned} \tag{2.5-23}$$

The elements of $\mathbf{c}_h$ are the camera coordinates in homogeneous form. As indicated, these coordinates can be converted to Cartesian form by dividing each of the first three components of $\mathbf{c}_h$ by the fourth. Thus the Cartesian coordinates of any point in the camera coordinate system are given in vector form by

$$\mathbf{c} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \dfrac{\lambda X}{\lambda - Z} \\ \dfrac{\lambda Y}{\lambda - Z} \\ \dfrac{\lambda Z}{\lambda - Z} \end{bmatrix} \tag{2.5-24}$$

The first two components of $\mathbf{c}$ are the (x, y) coordinates in the image plane of a projected 3-D point (X, Y, Z), as shown earlier in Eqs. (2.5-18) and (2.5-19). The third component is of no interest in terms of the model in Fig. 2.17. As shown next, this component acts as a free variable in the inverse perspective transformation.

The inverse perspective transformation maps an image point back into 3-D. Thus from Eq. (2.5-23),

$$\mathbf{w}_h = \mathbf{P}^{-1}\mathbf{c}_h \tag{2.5-25}$$

where $\mathbf{P}^{-1}$ is

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix} \tag{2.5-26}$$

Suppose that an image point has coordinates $(x_0, y_0, 0)$, where the 0 in the z location simply indicates that the image plane is located at $z = 0$. This point may be expressed in homogeneous vector form as

$$\mathbf{c}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} \tag{2.5-27}$$

Application of Eq. (2.5-25) then yields the homogeneous world coordinate vector

$$\mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix} \tag{2.5-28}$$

or, in Cartesian coordinates,

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix} \tag{2.5-29}$$

This result obviously is unexpected because it gives $Z = 0$ for *any* 3-D point. The problem here is caused by mapping a 3-D scene onto the image plane, which is a many-to-one transformation. The image point (x_0, y_0) corresponds to the set of collinear 3-D points that lie on the line passing through $(x_0, y_0, 0)$

and $(0, 0, \lambda)$. The equations of this line in the world coordinate system come from Eqs. (2.5-18) and (2.5-19); that is,

$$X = \frac{x_0}{\lambda}(\lambda - Z) \tag{2.5-30}$$

and

$$Y = \frac{y_0}{\lambda}(\lambda - Z). \tag{2.5-31}$$

Equations (2.5-30) and (2.5-31) show that unless something is known about the 3-D point that generated an image point (for example, its Z coordinate), it is not possible to completely recover the 3-D point from its image. This observation, which certainly is not unexpected, can be used to formulate the inverse perspective transformation by using the z component of $\mathbf{c}_h$ as a free variable, instead of 0. Thus, by letting

$$\mathbf{c}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ k \end{bmatrix} \tag{2.5-32}$$

it follows from Eq. (2.5-25) that

$$\mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ \dfrac{kz}{\lambda} + \mathrm{k} \end{bmatrix} \tag{2.5-33}$$

which, upon conversion to Cartesian coordinates gives

$$\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \dfrac{\lambda x_0}{\lambda + z} \\ \dfrac{\lambda y_0}{\lambda + z} \\ \dfrac{\lambda z}{\lambda + z} \end{bmatrix} \tag{2.5-34}$$

In other words, treating z as a free variable yields the equations

$$X = \frac{\lambda x_0}{\lambda + z}$$
$$Y = \frac{\lambda y_0}{\lambda + z} \qquad (2.5\text{-}35)$$
$$Z = \frac{\lambda z}{\lambda + z}$$

Solving for z in terms of Z in the last equation and substituting in the first two expressions yields

$$X = \frac{x_0}{\lambda}(\lambda - Z) \qquad (2.5\text{-}36)$$

and

$$Y = \frac{y_0}{\lambda}(\lambda - Z) \qquad (2.5\text{-}37)$$

which agrees with the observation that recovering a 3-D point from its image by means of the inverse perspective transformation requires knowledge of at least one of the world coordinates of the point. We address this problem again in Section 2.5.5.

2.5.3 Camera Model

Equations (2.5-23) and (2.5-24) characterize the formation of an image by projection of 3-D points onto an image plane. These two equations thus constitute a basic mathematical model of an imaging camera. This model is based on the assumption that the camera and world coordinate systems are coincident. In this section we consider a more general problem in which the two coordinate systems are allowed to be separate. However, the basic objective of obtaining the image-plane coordinates of any particular world point remains the same.

Figure 2.18 shows a world coordinate system (X, Y, Z) used to locate both the camera and 3-D points (denoted by $\mathbf{w}$). Figure 2.18 also shows the camera coordinate system (x, y, z) and image points (denoted by $\mathbf{c}$). The assumption is that the camera is mounted on a gimbal, which allows pan through an angle θ and tilt through an angle α. Here, *pan* is the angle between the x and X axes, and *tilt* is the angle between the z and Z axes. The offset of the center of the gimbal from the origin of the world coordinate system is denoted by $\mathbf{w}_0$, and the offset of the center of the imaging plane with respect to the gimbal center is denoted by vector $\mathbf{r}$, with components (r_1, r_2, r_3).

The concepts developed in Sections 2.5.1 and 2.5.2 provide all the necessary tools to derive a camera model based on the geometric arrangement of Fig. 2.18. The approach is to bring the camera and world coordinate systems into

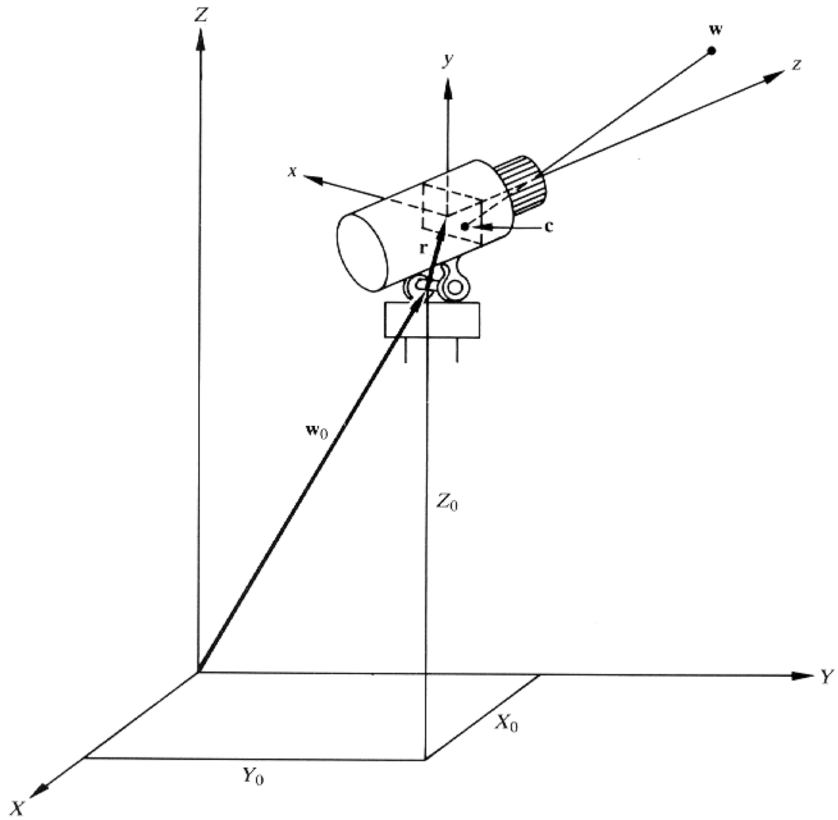

Figure 2.18 Imaging geometry with two coordinate systems. (From Fu, Gonzalez, and Lee [1987].)

alignment by applying a set of transformations. After doing so, we simply apply the perspective transformation of Eq. (2.5-22) to obtain the image-plane coordinates for any world point. In other words, we first reduce the problem to the geometric arrangement shown in Fig. 2.17 before applying the perspective transformation.

Suppose that, initially, the camera was in *normal position*, in the sense that the gimbal center and origin of the image plane were at the origin of the world coordinate system, and all axes were aligned. The geometric arrangement of Fig. 2.18 may then be achieved in several ways. Let us assume the following sequence of steps: (1) displacement of the gimbal center from the origin, (2) pan of the x axis, (3) tilt of the z axis, and (4) displacement of the image plane with respect to the gimbal center.

Obviously, the sequence of these mechanical steps does not affect the world points because the set of points seen by the camera after it was moved from normal position is quite different. However, applying exactly the same sequence of steps to all world points can achieve normal position again. A camera in normal position satisfies the arrangement of Fig. 2.17 for application of the perspective transformation. Thus the problem is reduced to applying to every world point a set of transformations that correspond to the steps listed earlier.

Translation of the origin of the world coordinate system to the location of the gimbal center is accomplished by using the transformation matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.5-38}$$

In other words, a homogeneous world point $\mathbf{w}_h$ that was at coordinates (X_0, Y_0, Z_0) is at the origin of the new coordinate system after the transformation $\mathbf{Gw}_h$.

As indicated earlier, the pan angle is measured between the x and X axes. In normal position, these two axes are aligned. In order to pan the x axis through the desired angle, we simply rotate it by θ. The rotation is with respect to the z axis and is accomplished by using the transformation matrix $\mathbf{R}_\theta$ of Eq. (2.5-9). In other words, application of this matrix to all points (including the point $\mathbf{Gw}_h$) effectively rotates the x axis to the desired location. When using Eq. (2.5-9) it is important to keep clearly in mind the convention established in Fig. 2.16. That is, angles are considered positive when points are rotated clockwise, which implies a counterclockwise rotation of the camera about the z axis. The unrotated (0°) position corresponds to the case when the x and X axes are aligned.

At this point the z and Z axes are still aligned. Since tilt is the angle between these two axes, we tilt the camera an angle α by rotating the z axis by α. The rotation is with respect to the x axis and is accomplished by applying the transformation matrix $\mathbf{R}_\alpha$ of Eq. (2.5-10) to all points (including the point $\mathbf{R}_\theta\mathbf{Gw}_h$). Again, a counterclockwise rotation of the camera implies positive angles, and the 0° mark is when the z and Z axes are aligned.[†]

According to the discussion in Section 2.5.4, the two rotation matrices can be concatenated into a single matrix, $\mathbf{R} = \mathbf{R}_\alpha \mathbf{R}_\theta$. Then, from Eqs. (2.5-9) and

---

[†] A useful way to visualize these transformations is to construct an axis system (for example, with pipe cleaners), label the axes x, y, and z, and perform the rotations manually, one axis at a time.

(2.5-10),

$$\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta\cos\alpha & \cos\theta\cos\alpha & \sin\alpha & 0 \\ \sin\theta\sin\alpha & -\cos\theta\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-39}$$

Finally, displacement of the origin of the image plane by vector $\mathbf{r}$ is achieved by the transformation matrix

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & -r_1 \\ 0 & 1 & 0 & -r_2 \\ 0 & 0 & 1 & -r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5-40}$$

Thus applying to $\mathbf{w}_h$ the series of transformations $\mathbf{CRGw}_h$ brings the world and camera coordinate systems into coincidence. The image-plane coordinates of a point $\mathbf{w}_h$ are finally obtained by using Eq. (2.5-23). In other words, a homogeneous world point that is being viewed by a camera satisfying the geometric arrangement shown in Fig. 2.18 has the following homogeneous representation in the camera coordinate system:

$$\mathbf{c}_h = \mathbf{PCRGw}_h \tag{2.5-41}$$

Equation (2.5-41) represents a perspective transformation involving two coordinate systems.

As indicated in Section 2.5.2, we obtain the Cartesian coordinates (x, y) of the imaged point by dividing the first and second components of $\mathbf{c}_h$ by the fourth. Expanding Eq. (2.5-41) and converting to Cartesian coordinates yields

$$x = \lambda \frac{(X - X_0)\cos\theta + (Y - Y_0)\sin\theta - r_1}{-(X - X_0)\sin\theta\sin\alpha + (Y - Y_0)\cos\theta\sin\alpha - (Z - Z_0)\cos\alpha + r_3 + \lambda} \tag{2.5-42}$$

and

$$y = \lambda \frac{-(X - X_0)\sin\theta\cos\alpha + (Y - Y_0)\cos\theta\cos\alpha + (Z - Z_0)\sin\alpha - r_2}{-(X - X_0)\sin\theta\sin\alpha + (Y - Y_0)\cos\theta\sin\alpha - (Z - Z_0)\cos\alpha + r_3 + \lambda} \tag{2.5-43}$$

which are the image coordinates of a point $\mathbf{w}$ whose world coordinates are $(X,$

Y, Z). These equations reduce to Eqs. (2.5-18) and (2.5-19) when $X_0 = Y_0 = Z_0 = 0$, $r_1 = r_2 = r_3 = 0$, and $\alpha = \theta = 0°$.

Example: As an illustration of the concepts just discussed, suppose that we want to find the image coordinates of the corner of the block shown in Fig. 2.19. The camera is offset from the origin and is viewing the scene with a pan of 135° and a tilt of 135°. We will follow the convention that transformation angles are positive when the camera rotates counterclockwise, viewing the origin along the axis of rotation.

Let us examine in detail the steps required to move the camera from normal position to the geometry shown in Fig. 2.19. The camera is in normal position in Fig. 2.20(a) and displaced from the origin in Fig. 2.20(b). Note that, after this step, the world coordinate axes are used only to establish angle references. That is, after displacement of the world coordinate origin, all rotations take place about the new (camera) axes. Figure 2.20(c) shows a view along the z axis of the camera to establish pan. In this case the rotation of the camera about the z axis is counterclockwise, so world points are rotated about this axis in the opposite direction, which makes θ a positive angle. Figure 2.20(d) shows a view, after pan, along the x axis of the camera to establish tilt. The rotation about this axis is counterclockwise, which makes α a positive angle. The world coordinate axes are shown as dashed lines in the latter two figures to emphasize that their only use is to establish the zero reference for the pan and tilt angles. We do not show the final step of displacing the image plane from the center of the gimbal.

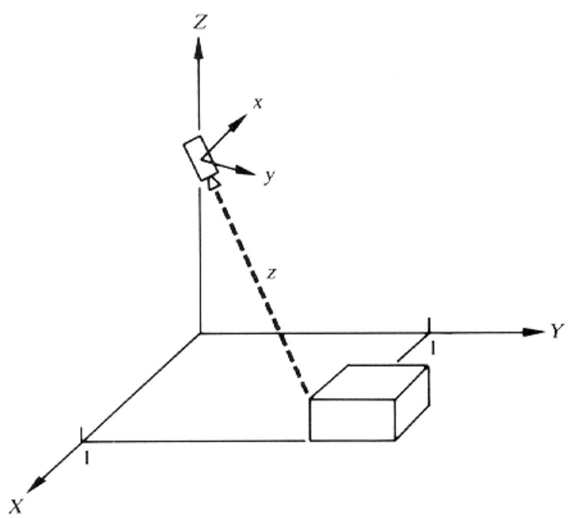

Figure 2.19 *Camera viewing a 3-D scene. (From Fu, Gonzalez, and Lee [1987].)*

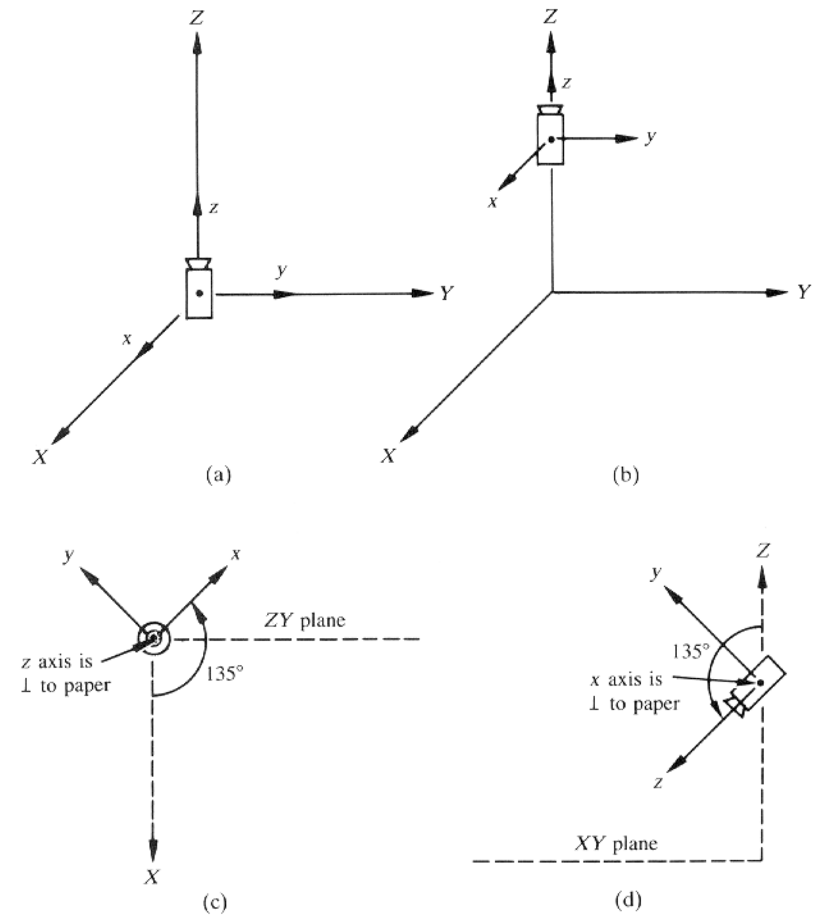

Figure 2.20 *(a) Camera in normal position; (b) gimbal center displaced from origin; (c) observer view of rotation about z axis to determine pan angle; (d) observer view of rotation about x axis for tilt. (From Fu, Gonzalez, and Lee [1987].)*

The following parameter values apply to this problem:

$$X_0 = 0 \text{ m} \qquad Y_0 = 0 \text{ m} \qquad Z_0 = 1 \text{ m};$$

$$\alpha = 135^\circ \qquad \theta = 135^\circ;$$

$$r_1 = 0.03 \text{ m} \qquad r_2 = r_3 = 0.02 \text{ m} \qquad \lambda = 35 \text{ mm} = 0.035 \text{ m}$$

The corner in question is at coordinates $(X, Y, Z) = (1, 1, 0.2)$.

To compute the image coordinates of the block corner, we simply substitute the parameter values into Eqs. (2.5-42) and (2.5-43); that is,

$$x = \lambda \frac{-0.03}{-1.53 + \lambda}$$

Similarly,

$$y = \lambda \frac{+0.42}{-1.53 + \lambda}$$

Substituting $\lambda = 0.035$ yields the image coordinates

$$x = 0.0007 \text{ m} \quad \text{and} \quad y = -0.009 \text{ m}.$$

Note that these coordinates are well within a 1×1 in. (0.025×0.025 m) imaging plane. It is easily verified that use of a lens with a 200-mm focal length, for example, would have imaged the corner of the block outside the boundary of a plane with these dimensions (that is, outside the effective field of view of the camera).

Finally, note that all coordinates obtained with Eqs. (2.5-42) and (2.5-43) are with respect to the center of the image plane. A change of coordinates is required to use the convention established earlier that the origin of an image is at its top left corner. ❑

2.5.4 Camera Calibration

In Section 2.5.3 we developed explicit equations for the image coordinates (x, y), of a world point $\mathbf{w}$. As shown in Eqs. (2.5-42) and (2.5-43), implementation of these equations requires knowledge of the focal length, offsets, and angles of pan and tilt. Although these parameters could be measured directly, determining one or more of the parameters using the camera itself as a measuring device often is more convenient (especially when the camera moves frequently). This requires a set of image points whose world coordinates are known, and the computational procedure used to obtain the camera parameters using these known points often is referred to as *camera calibration*.

With reference to Eq. (2.5-41), let $\mathbf{A} = \mathbf{PCRG}$. The elements of $\mathbf{A}$ contain all the camera parameters and, from Eq. (2.5-41), $\mathbf{c}_h = \mathbf{A}\mathbf{w}_h$. Letting $k = 1$ in the homogeneous representation yields

$$\begin{bmatrix} c_{h1} \\ c_{h2} \\ c_{h3} \\ c_{h4} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.5-44}$$

Based on the discussion in Sections 2.5.2 and 2.5.3, the camera coordinates in

Cartesian form are

$$x = c_{h1}/c_{h4} \tag{2.5-45}$$

and

$$y = c_{h2}/c_{h4}. \tag{2.5-46}$$

Substituting $c_{h1} = xc_{h4}$ and $c_{h2} = yc_{h4}$ in Eq. (2.5-44) and expanding the matrix product yields

$$\begin{aligned} xc_{h4} &= a_{11}X + a_{12}Y + a_{13}Z + a_{14} \\ yc_{h4} &= a_{21}X + a_{22}Y + a_{23}Z + a_{24} \\ c_{h4} &= a_{41}X + a_{42}Y + a_{43}Z + a_{44}, \end{aligned} \tag{2.5-47}$$

where expansion of c_{h3} was ignored because it is related to z.

Substitution of c_{h4} in the first two equations of (2.5-47) yields two equations with 12 unknown coefficients:

$$a_{11}X + a_{12}Y + a_{13}Z - a_{41}xX - a_{42}xY - a_{43}xZ - a_{44}x + a_{14} = 0 \tag{2.5-48}$$

$$a_{21}X + a_{22}Y + a_{23}Z - a_{41}yX - a_{42}yY - a_{43}yZ - a_{44}y + a_{24} = 0. \tag{2.5-49}$$

The calibration procedure then consists of (1) obtaining $m \geqslant 6$ world points (there are *two* equations) with known coordinates (X_i, Y_i, Z_i), $i = 1, 2, \ldots,$ m; (2) imaging these points with the camera in a given position to obtain the corresponding image points (x_i, y_i), $i = 1, 2, \ldots, m$; and (3) using these results in Eqs. (2.5-48) and (2.5-49) to solve for the unknown coefficients. Many numerical techniques exist for finding an optimal solution to a linear system of equations, such as the one given by these equations (see, for example, Noble [1969]).

2.5.5 Stereo Imaging

Recall that mapping a 3-D scene onto an image plane is a many-to-one transformation. That is, an image point does not uniquely determine the location of a corresponding world point. However, the missing *depth* information can be obtained by using stereoscopic (*stereo* for short) imaging techniques.

As Fig. 2.21 shows, stereo imaging involves obtaining two separate image views of an object (a single world point $\mathbf{w}$ in this discussion). The distance between the centers of the two lenses is called the *baseline*, and the objective is to find the coordinates (X, Y, Z) of the point $\mathbf{w}$ having image points (x_1, y_1) and (x_2, y_2). The assumption is that the cameras are identical and that the coordinate systems of both cameras are perfectly aligned, differing only in the location of their origins, a condition usually met in practice. Recall that, after

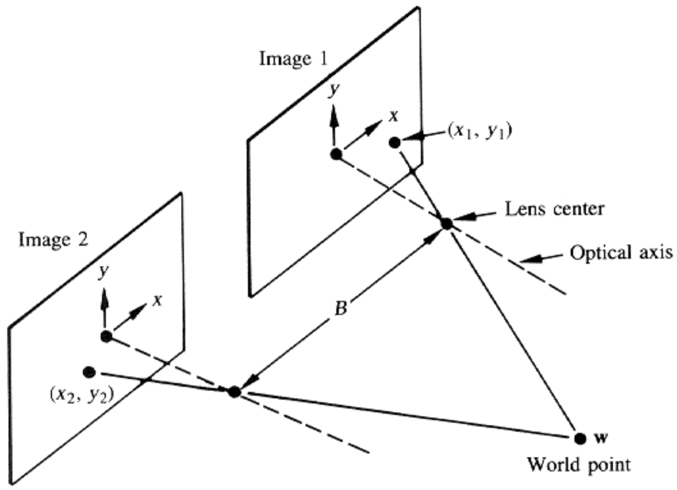

Figure 2.21 *Model of the stereo imaging process. (From Fu, Gonzalez, and Lee [1987].)*

the camera and world coordinate systems have been brought into coincidence, the xy plane of the image is aligned with the XY plane of the world coordinate system. Then, under the above assumption, the Z coordinate of $\mathbf{w}$ is exactly the same for both camera coordinate systems.

Let us bring the first camera into coincidence with the world coordinate system, as shown in Fig. 2.22. Then, from Eq. (2.5-30), $\mathbf{w}$ lies on the line with (partial) coordinates

$$X_1 = \frac{x_1}{\lambda}(\lambda - Z_1) \tag{2.5-50}$$

where the subscripts on X and Z indicate that the first camera was moved to the origin of the world coordinate system, with the second camera and $\mathbf{w}$ following, but keeping the relative arrangement shown in Fig. 2.21. If, instead, the second camera is brought to the origin of the world coordinate system, $\mathbf{w}$ lies on the line with (partial) coordinates

$$X_2 = \frac{x_2}{\lambda}(\lambda - Z_2). \tag{2.5-51}$$

However, because of the separation between cameras and because the Z coordinate of $\mathbf{w}$ is the same for both camera coordinate systems, it follows that

$$X_2 = X_1 + B \tag{2.5-52}$$

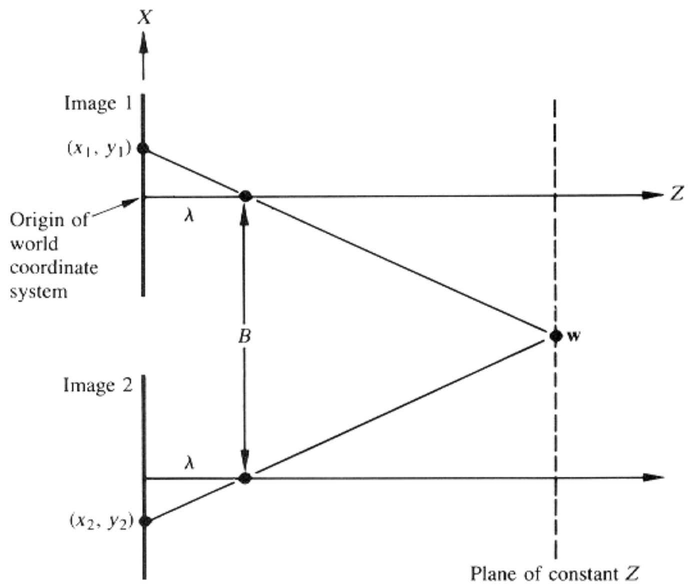

Figure 2.22 *Top view of Fig. 2.21 with the first camera brought into coincidence with the world coordinate system. (From Fu, Gonzalez, and Lee [1987].)*

and

$$Z_2 = Z_1 = Z \tag{2.5-53}$$

where B is the baseline distance.

Substituting Eqs. (2.5-52) and (2.5-53) into Eq. (2.5-50) and (2.5-51) gives

$$X_1 = \frac{x_1}{\lambda}(\lambda - Z) \tag{2.5-54}$$

and

$$X_1 + B = \frac{x_2}{\lambda}(\lambda - Z). \tag{2.5-55}$$

Subtracting Eq. (2.5-54) from (2.5-55) and solving for Z yields

$$Z = \lambda - \frac{\lambda B}{x_2 - x_1} \tag{2.5-56}$$

which indicates that if the difference between the corresponding image coordinates x_2 and x_1 can be determined, and the baseline and focal length are known, calculating the Z coordinate of $\mathbf{w}$ is a simple matter. The X and Y

world coordinates then follow directly from Eqs. (2.5-30) and (2.5-31) using either (x_1, y_1) or (x_2, y_2).

The most difficult task in using Eq. (2.5-56) to obtain Z is to actually find two corresponding points in different images of the same scene. As these points generally are in the same vicinity, a frequently used approach is to select a point within a small region in one of the image views and then attempt to find the best matching region in the other view by using correlation techniques, as discussed in Chapter 9. When the scene contains distinct features, such as prominent corners, a feature-matching approach generally yields a faster solution for establishing correspondence. The calibration procedure developed in Section 2.5.4 is directly applicable to stereo imaging by simply treating the cameras independently.