light stripe only when it is directly below the camera. It is of interest to note that the line scan camera sees only the line on which the two light planes converge, and two-dimensional information can be accumulated as the object moves past the camera.

The directional-lighting approach shown in Fig. 7.6d is useful primarily for inspection of object surfaces. Defects on the surface, such as pits and scratches, can be detected by using a highly directed light beam (e.g., a laser beam) to measure the amount of scatter. For flaw-free surfaces little light is scattered upward to the camera. On the other hand, the presence of a flaw greatly increases the amount of light scattered to the camera, thus facilitating detection of a defect. An example is shown in Fig. 7.11.

7.4 IMAGING GEOMETRY

In the following discussion we consider several important transformations used in imaging, derive a camera model, and treat the stereo imaging problem in some detail. Some of the transformations discussed in the following section were already introduced in Chap. 2 in connection with robot arm kinematics. Here, we consider a similar problem, but from the point of view of imaging.

7.1 Some Basic Transformations

The material in this section deals with the development of a unified representation of problems such as image rotation, scaling, and translation. All transformations
are expressed in a three-dimensional (3D) cartesian coordinate system in which a point has coordinates denoted by \((X, Y, Z)\). In cases involving two-dimensional images, we will adhere to our previous convention of using the lowercase representation \((x, y)\) to denote the coordinates of a pixel. It is common terminology to refer to \((X, Y, Z)\) as the \textit{world coordinates} of a point.

**Translation.** Suppose that we wish 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 following equations:

\[
\begin{align*}
X^* &= X + X_0 \\
Y^* &= Y + Y_0 \\
Z^* &= Z + Z_0
\end{align*}
\]

Equation (7.4-1) can be expressed in matrix form by writing:

\[
\begin{bmatrix}
X^* \\
Y^* \\
Z^*
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 & X_0 \\
0 & 1 & 0 & Y_0 \\
0 & 0 & 1 & Z_0
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

As indicated later in this section, it is often useful to concatenate several transformations into a composite result, such as translation, followed by another transformation.
and \( \mathbf{v}^* \) is a column vector whose components are the transformed coordinates:

\[
\mathbf{v}^*_* = \begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix}
\] (7.4-6)

Using this notation, the matrix used for translation is given by

\[
\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\] (7.4-7)

and the translation process is accomplished by using Eq. (7.4-4), so that \( \mathbf{v}^*_* = \mathbf{T}\mathbf{v} \).

**Scaling.** Scaling by factors \( S_x, S_y, \) and \( S_z \) along the \( X, Y, \) and \( Z \) axes is given by the transformation matrix

\[
\mathbf{S} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\] (7.4-8)

**Rotation.** The transformations used for three-dimensional 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 given point about an arbitrary point in space requires three transformations: The first 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. 7.12, rotation of a point about the \( Z \) coordinate axis by angle \( \theta \) is achieved by using the transformation

\[
\mathbf{R}_\theta = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\] (7.4-9)

The rotation angle \( \theta \) is measured clockwise when looking at the origin from a point on the \( +Z \) axis. It is noted that this transformation affects only the values of \( X \) and \( Y \) coordinates.

In terms of the values of \( X^*, Y^*, \) and \( Z^* \), Eqs. (7.4-2) and (7.4-3) are clearly equivalent.

Throughout this section, we will use the unified matrix representation

\[
\mathbf{v}^*_* = \mathbf{A}\mathbf{v}
\] (7.4-4)

where \( \mathbf{A} \) is a \( 4 \times 4 \) transformation matrix, \( \mathbf{v} \) is a column vector containing the original coordinates:

\[
\mathbf{v} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
\] (7.4-5)
Rotation of a point about the $X$ axis by an angle $\alpha$ is performed by using the transformation
\[
R_\alpha = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha & 0 \\
0 & -\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\] (7.4-10)

Finally, rotation of a point about the $Y$ axis by an angle $\beta$ is achieved by using the transformation
\[
R_\beta = \begin{bmatrix}
\cos \beta & 0 & -\sin \beta & 0 \\
0 & 1 & 0 & 0 \\
\sin \beta & 0 & \cos \beta & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\] (7.4-11)

Concatenation and Inverse Transformations. The application of several transformations can be represented by a single $4 \times 4$ transformation matrix. For example, translation, scaling, and rotation about the $Z$ axis of a point $v$ is given by
\[
v^* = R_\theta S(Tv) = Av
\] (7.4-12)

Although our 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. (7.4-5), let $v_1, v_2, \ldots, v_m$ represent the coordinates of $m$ points. If we form a $4 \times m$ matrix $V$ whose columns are these column vectors, then the simultaneous transformation of all these points by a $4 \times 4$ transformation matrix $A$ is given by
\[
V^* = AV
\] (7.4-13)

The resulting matrix $V^*$ is $4 \times m$. Its $i$th column, $v_i^*$, contains the coordinates of the transformed point corresponding to $v_i$.

Before leaving this section, we point out that 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 given by
\[
T^{-1} = \begin{bmatrix}
1 & 0 & 0 & -X_0 \\
0 & 1 & 0 & -Y_0 \\
0 & 0 & 1 & -Z_0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\] (7.4-14)

Similarly, the inverse rotation matrix $R_\theta^{-1}$ is given by
\[
R_\theta^{-1} = \begin{bmatrix}
\cos (-\theta) & \sin (-\theta) & 0 & 0 \\
-\sin (-\theta) & \cos (-\theta) & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\] (7.4-15)

The inverse of more complex transformation matrices is usually obtained by numerical techniques.

7.4.2 Perspective Transformations

A perspective transformation (also called an imaging transformation) projects 3D 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 three-dimensional world. Although perspective transformations will be expressed later in this section in a $4 \times 4$ matrix form, these transformations are fundamentally different from those discussed in the previous section because they are nonlinear in the sense that they involve division by coordinate values.

A model of the image formation process is shown in Fig. 7.12. We define
It is important to note that these equations are nonlinear because they involve division by the variable $Z$. Although we could use them directly as shown above, it is often convenient to express these equations in matrix form as we did in the previous section for rotation, translation, and scaling. This can be accomplished easily 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

$$w = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \text{(7.4-20)}$$

and its homogeneous counterpart is given by

$$w_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} \quad \text{(7.4-21)}$$

If we define the perspective transformation matrix

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/\lambda & 1 \end{bmatrix} \quad \text{(7.4-22)}$$

Then the product $Pw_h$ yields a vector which we shall denote by $c_h$:

$$c_h = Pw_h = \begin{bmatrix} 1 & 0 & 0 & 0 & kX \\ 0 & 1 & 0 & 0 & kY \\ 0 & 0 & 1 & 0 & kZ \\ 0 & 0 & -1/\lambda & 1 & k \end{bmatrix} \quad \text{(7.4-23)}$$

The elements of $c_h$ are the camera coordinates in homogeneous form. As indi-
cated above, these coordinates can be converted to cartesian form by dividing each of the first three components of \( c_h \) by the fourth. Thus, the cartesian coordinates of any point in the camera coordinate system are given in vector form by

\[
\mathbf{c} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\lambda X}{\lambda - Z} \\ \frac{\lambda Y}{\lambda - Z} \\ \frac{\lambda Z}{\lambda - Z} \end{bmatrix}
\]  
(7.4-24)

The first two components of \( \mathbf{c} \) are the \((x, y)\) coordinates in the image plane of a projected 3D point \((X, Y, Z)\), as shown earlier in Eqs. (7.4-18) and (7.4-19). The third component is of no interest to us in terms of the model in Fig. 7.13. As will be seen below, this component acts as a free variable in the inverse perspective transformation.

The inverse perspective transformation maps an image point back into 3D. Thus, from Eq. (7.4-23),

\[
\mathbf{w}_h = \mathbf{P}^{-1}\mathbf{c}_h
\]  
(7.4-25)

where \( \mathbf{P}^{-1} \) is easily found to be

\[
\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix}
\]  
(7.4-26)

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

\[
\mathbf{c}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ 0 \\ k \end{bmatrix}
\]  
(7.4-27)

Application of Eq. (7.4-25) then yields the homogeneous world coordinate vector

\[
\mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ k \end{bmatrix}
\]  
(7.4-28)

or, in cartesian coordinates,

\[
\mathbf{w} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ 0 \end{bmatrix}
\]  
(7.4-29)

This is obviously not what one would expect since it gives \( Z = 0 \) for any 3D point. The problem here is caused by the fact that mapping a 3D scene onto the image plane is a many-to-one transformation. The image point \((x_0, y_0)\) corresponds to the set of colinear 3D points which lie on the line that passes through \((x_0, y_0, 0)\) and \((0, 0, \lambda)\). The equations of this line in the world coordinate system are obtained from Eqs. (7.4-18) and (7.4-19); that is,

\[
X = \frac{x_0}{\lambda} (\lambda - Z)
\]  
(7.4-30)

and

\[
Y = \frac{y_0}{\lambda} (\lambda - Z)
\]  
(7.4-31)

These equations show that, unless we know something about the 3D point which generated a given image point (for example, its \( Z \) coordinate), we cannot completely recover the 3D point from its image. This observation, which is certainly not unexpected, can be used as a way to formulate the inverse perspective transformation simply by using the \( z \) component of \( \mathbf{c}_h \) as a free variable instead of 0. Thus, letting

\[
\mathbf{c}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \\ k \end{bmatrix}
\]  
(7.4-32)

we now have from Eq. (7.4-25) that

\[
\mathbf{w}_h = \begin{bmatrix} kx_0 \\ ky_0 \\ kz \end{bmatrix}
\]  
(7.4-33)
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 established above that transformation angles are positive when the camera rotates in a counterclockwise manner when 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. 7.15. The camera is shown in normal position in Fig. 7.16a, and displaced from the origin in Fig. 7.16b. It is important to 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 7.16c 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 $\theta$ a positive angle. Figure 7.16d shows a view after pan, along the x axis of the camera to establish tilt. The rotation about this axis is counterclockwise, which makes $\alpha$ a positive angle. The world coordinate axes are shown dashed in the latter two figures to emphasize the fact that their only use is to establish the zero reference for the pan and tilt angles. We do not show in this figure the final step of displacing the image plane from the center of the gimbal.

The following parameter values apply to the problem:

\[
X_0 = 0 \text{ m} \\
Y_0 = 0 \text{ m} \\
Z_0 = 1 \text{ m} \\
\alpha = 135^\circ \\
\theta = 135^\circ \\
r_1 = 0.03 \text{ m} \\
r_2 = r_3 = 0.02 \text{ m} \\
\lambda = 35 \text{ mm} = 0.035 \text{ m}
\]

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 above parameter values into Eqs. (7.4-42) and (7.4-43); that is,

\[
x = \lambda \frac{-0.03}{-1.53 + \lambda}
\]

and

\[
y = \lambda \frac{-0.42}{-1.53 + \lambda}
\]

Substituting $\lambda = 0.035$ yields the image coordinates

\[
x = 0.0007 \text{ m}
\]

and

\[
y = 0.009 \text{ m}
\]

It is of interest to note that these coordinates are well within a 1 x 1 inch (0.025 x 0.025 m) imaging plane. If, for example, we had used a lens with a 200-mm focal length, it is easily verified from the above results that the corner of the block would have been imaged outside the boundary of a plane with these dimensions (i.e., it would have been outside the effective field of view of the camera).

Finally, we point out that all coordinates obtained via the use of Eqs. (7.4-42) and (7.4-43) are with respect to the center of the image plane. A change of coordinates would be required to use the convention established earlier, in which the origin of an image is at its top left corner.

### 7.4.4 Camera Calibration

In Sec. 7.4.3 we obtained explicit equations for the image coordinates $(x, y)$ of a world point $w$. As shown in Eqs. (7.4-42) and (7.4-43), implementation of these equations requires knowledge of the focal length, camera offsets, and angles of pan and tilt. While these parameters could be measured directly, it is often more convenient (e.g., when the camera moves frequently) to determine one or more of the
From the discussion in the previous two sections we know that the camera coordinates in cartesian form are given by

\[ x = \frac{c_{h1}}{c_{h4}} \]  

(7.4-45)

and

\[ y = \frac{c_{h2}}{c_{h4}} \]  

(7.4-46)

Substituting \( c_{h1} = xc_{h4} \) and \( c_{h2} = yc_{h4} \) in Eq. (7.6-44) and expanding the matrix product yields

\[ xc_{h4} = a_{11}X + a_{12}Y + a_{13}Z + a_{14} \]  

(7.4-47)

\[ yc_{h4} = a_{21}X + a_{22}Y + a_{23}Z + a_{24} \]

\[ c_{h4} = a_{31}X + a_{32}Y + a_{33}Z + a_{34} \]

where expansion of \( c_{h3} \) has been ignored because it is related to \( z \).

Substitution of \( c_{h4} \) in the first two equations of (7.4-47) yields two equations with twelve unknown coefficients:

\[ a_{11}X + a_{12}Y + a_{13}Z - a_{41}X - a_{42}Y - a_{43}Z - a_{44}X + a_{14} = 0 \]  

(7.4-48)

\[ a_{21}X + a_{22}Y + a_{23}Z - a_{41}Y - a_{42}Y - a_{43}Y - a_{44}Y + a_{24} = 0 \]  

(7.4-49)

The calibration procedure then consists of (1) obtaining \( m \geq 6 \) world points with known coordinates \((X_i, Y_i, Z_i)\), \( i = 1, 2, \ldots, m \) (there are two equations involving the coordinates of these points, so at least six points are needed), (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. (7.4-48) and (7.4-49) to solve for the unknown coefficients. There are many numerical techniques for finding an optimal solution to a linear system of equations such as (7.4-48) and (7.4-49) (see, for example, Noble [1969]).

### 7.4.5 Stereo Imaging

It was noted in Sec. 7.4.2 that mapping a 3D 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. It is shown in this section that the missing depth information can be obtained by using stereoscopic (stereo for short) imaging techniques.

As shown in Fig. 7.17, stereo-imaging involves obtaining two separate image views of an object of interest (e.g., a world point \( w \)). The distance between the cameras is \( d \) and the objective is to find the coordinates of \( w \) and the distance \( d \) between the cameras from these images.