Chapter 8 Camera Calibration

In mathematical terms, a camera maps all points on a 3D target object surface to a collection of 2D points on the image plane; a *camera model* thus relates the image coordinates to the physical locations of the object points in the FOV. *Camera calibration* refers to the process of deriving the internal (*intrinsic*) and external (*extrinsic*) parameters of the camera model and image-capture hardware. Intrinsic parameters embody the characteristics of the optical system and its geometric relationship with the image sensor, while extrinsic parameters relate the location and orientation of the camera with respect to the 3D object (Euclidean) space. The 3D object space is used for measuring the collection of object point coordinates $[\Re\{(x_o, y_o, z_o)\}]$ in physical units that make up the target scene in the FOV. Extrinsic parameters are derived as a set of rigid body transformation matrices: three rotations about the x, y, and z axes, and three translations along these axes (block 1 in Fig. 8.1). The two sets of output from the extrinsic calibration process $\Re(x_i = x_i^{ud}, y_i = y_i^{ud})$ are fed into the intrinsic calibration process to make up the complete camera calibration model. In cinematography, rotations about the x, y, and z axes are called *pan*, *tilt*, and *roll*, respectively, and the movement along the z axis is called *zooming*.

The first stage of intrinsic transformation is to convert each 3D camera point (x_c, y_c, z_c) into an ideal image point (x_i, y_i) on the 2D image plane through perspective transformation (block 2 in Fig. 8.1). Intrinsic transformations in the two subsequent stages involve the spatial conversion of each ideal (undistorted) image point (x_i^{ud}, y_i^{ud}) into an actual (distorted) optical image point (x_i^d, y_i^d) on the continuous image plane, and its conversion into an image pixel at $(x_i^d/\Delta_{sx}, y_i^d/\Delta_{sy})$ on the 2D image sensor. The collection of these sensor pixels $\Re\{(x_i^d/\Delta_{sx}, y_i^d/\Delta_{sy})\}$ is captured by the framegrabber to make up one stored image frame $\Re\{(x_{fi}, y_{fi})\}$. The intrinsic parameters include the focal length, lens geometric errors, image sensor parameters, and spatial scaling introduced during image data transfer and storage onto the frame store.

While the focal length and the optical parameters of the lens are listed in the system hardware specifications, the location of the target object with respect to the image plane (image sensor) is not known *a priori*. These parameters, along with the generally unspecified parameters of lens distortion and image transfer, are candidates for camera calibration. Since *calibration parameters* are used for extracting the physical dimensions of the 3D object from its 2D image,

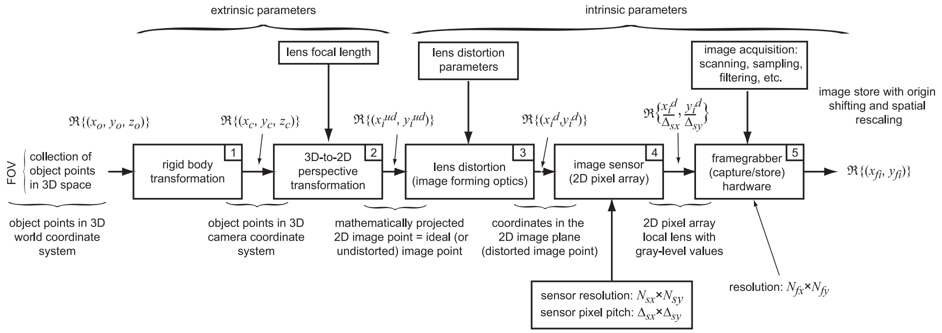

extrinsic parameters
intrinsic parameters
lens focal length
lens distortion parameters
image acquisition: scanning, sampling, filtering, etc.
image store with origin shifting and spatial rescaling
$\Re\{(x_o, y_o, z_o)\}$
$\Re\{(x_c, y_c, z_c)\}$
$\Re\{(x_i^{ud}, y_i^{ud})\}$
$\Re\{(x_i^d, y_i^d)\}$
$\Re\{\frac{x_i^d}{\Delta_{sx}}, \frac{y_i^d}{\Delta_{sy}}\}$
FOV
collection of object points in 3D space
rigid body transformation
1
3D-to-2D perspective transformation
2
lens distortion (image forming optics)
3
image sensor (2D pixel array)
4
framegrabber (capture/store) hardware
5
$\Re\{(x_{fi}, y_{fi})\}$
object points in 3D world coordinate system
object points in 3D camera coordinate system
mathematically projected 2D image point = ideal (or undistorted) image point
coordinates in the 2D image plane (distorted image point)
2D pixel array local lens with gray-level values
sensor resolution: N_{sx}×N_{sy}
sensor pixel pitch: Δ_{sx}×Δ_{sy}
resolution: N_{fx}×N_{fy}

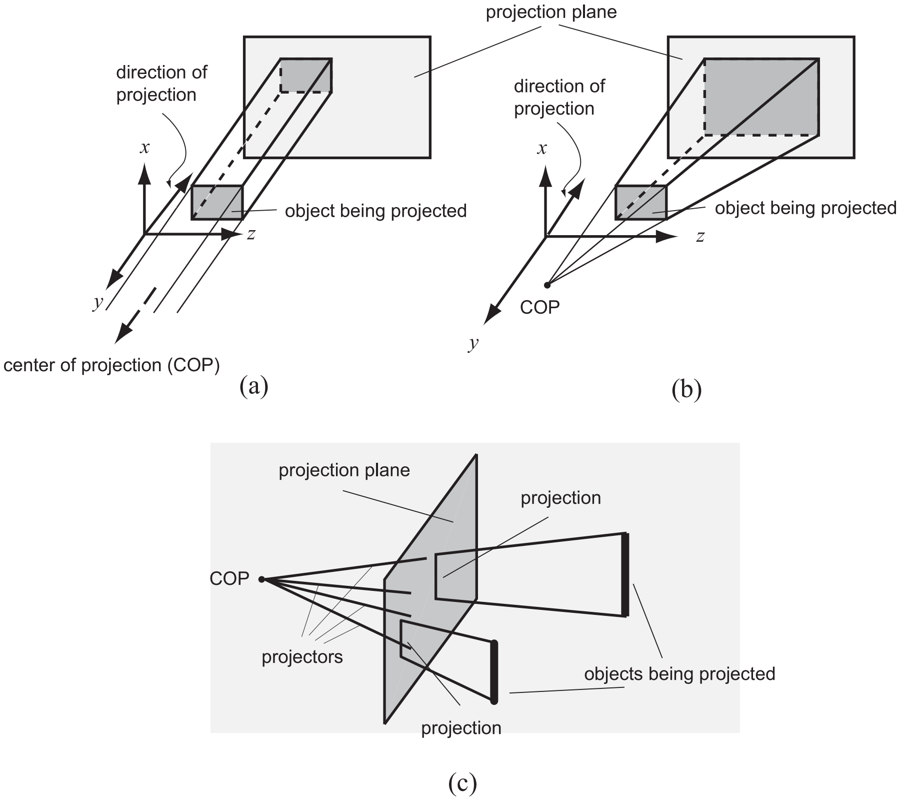
Figure 8.1 Notation and transformations in camera modeling. The transformations in blocks 1 and 2 create an ideal optical image point on the continuous image plane for each object point. The transformation in block 3 converts this ideal image point into an actual image point by introducing lens distortions. Block 4 generates discrete image pixels on the 2D image sensor placed at the focal plane of the lens. Block 5 models the pixel-by-pixel capture of the sensor image output and storage to make up one image frame of the FOV.

camera modeling is critical in an image-based measurement system. Some of the basic projective geometry and modeling concepts in blocks 1 and 2 of Fig. 8.1 are considered in Secs. 8.1 through 8.4, and the Tsai calibration method that includes all blocks is developed in Sec. 8.5. A review of stereo imaging for depth measurement is given in Sec. 8.6, and some of the commonly used concepts of feature matching used in stereo imaging are developed in Sec. 8.7. In some application environments, a relatively simple setup may be preferred at the expense of some accuracy. The monocular inclined camera arrangement presented in Sec. 8.8 may be suitable for some of these applications. The calibration procedure for this geometry is developed in the concluding section.

8.1 Projection

In mathematical terms, a *space* is a collection of points. Any conceptually defined space has a dimension given by the number of parameters (coordinates) required to uniquely identify all points in the space. *Euclidean* space refers to the 3D physical (*real*) space defined by three perpendicular (*orthogonal*) axes. In plane geometry, projections are formed by the intersection of lines (*projectors*) with a plane (*projection plane*); projection lines emanate from a *center of projection* [Figs. 8.2(a) and (b)]. Projection methods are broadly grouped under *parallel projection* and *perspective projection.* There are two types of parallel projection: *orthographic* parallel projection with the direction of projection normal to the projection plane, and *oblique* parallel projection with two tilted directions. Parallel projection preserves the object size after projection with the center of projection at infinity. Objects closer to the view plane appear larger when projected, while distant objects appear smaller after projection [Fig. 8.2(c)].

In perspective projection, parallel lines that are not parallel to the projection plane converge to a vanishing point. The concept of vanishing points is well



(a) (b) (c)

Figure 8.2 (a) Parallel projection and (b) perspective projection with the center of projection (COP) at infinity. (c) Objects nearer the perspective projection plane appear to be larger.

established in perspective drawing (Fig. 8.3); perspective projections are subgrouped by the number of vanishing points. The mathematics of *absolute*, *affine*, and *projective* geometries describe the various mechanics of transforming a set of given coordinate points into another using orthogonal and nonorthogonal axes. Absolute geometry subsumes Euclidean geometry. Affine transformation preserves the collinearity of points and the parallelism of lines, among other properties. Some of these properties are used in Chapter 10.

Most geometric shapes in computer graphics are generated by one or more subsets of parallel and perspective transformation.[1,2] In imaging, *projection* refers to the process of converting the brightness of a 3D object scene into a spatial intensity distribution on a 2D image plane (*projection plane*) through perspective transformation with the center of projection placed at the geometric center of the lens. The optical imaging process is a combination of rigid body transformation (3D world coordinates to 3D camera coordinates of each object point) followed by perspective projection (3D image scene to 2D image). This process is mathematically modeled by

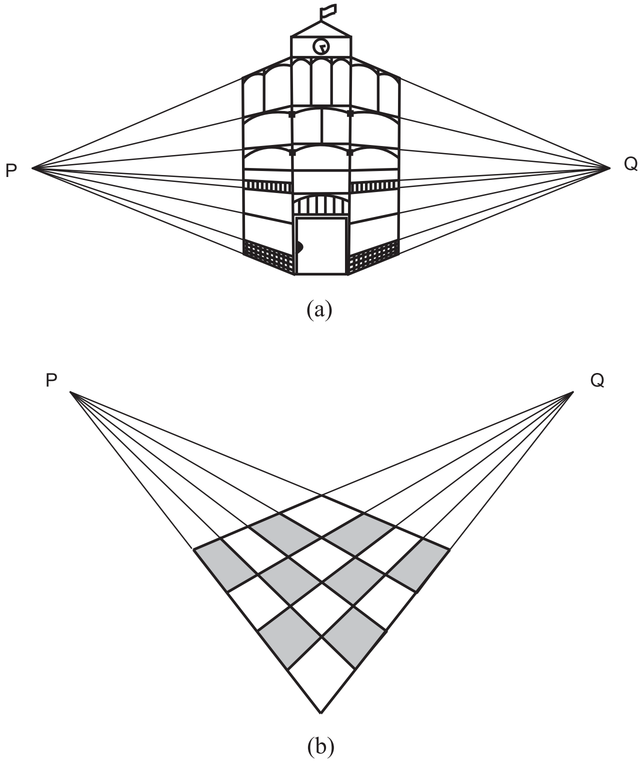


Figure 8.3 Locations of vanishing points P and Q. (a) Use of vanishing points in projective drawings. (b) Parallel lines meeting at a vanishing point after projection.

$$\begin{aligned}
\begin{bmatrix}\text{2D image point } (x_i, y_i)\\ \text{in } \textit{homogeneous}\\ \textit{coordinates}\end{bmatrix}_{3\times 1} &= \begin{bmatrix}\text{Transformation of the image}\\ \text{sensor to create an image frame;}\\ \text{2D image transfer, capture and}\\ \text{storage parameters } (\textit{intrinsic})\end{bmatrix}_{3\times 3}\\
&\times \begin{bmatrix}\text{3D-to-2D perspective transformation}\\ \text{from object space to image plane}\\ \text{using object-to-image projective}\\ \text{transformation } (\textit{intrinsic})\end{bmatrix}_{3\times 4}\\
&\times \begin{bmatrix}\text{transformation to align 3D}\\ \text{object coodinate systems to the}\\ \text{3D camera coordinate systems}\\ (\textit{extrinsic} \text{ transformation})\end{bmatrix}_{4\times 4}\\
&\times \begin{bmatrix}\text{coordinates of the}\\ \text{object point } (x_w, y_w, z_w)\\ \text{in 3D space in}\\ \textit{homogeneous form}\end{bmatrix}_{4\times 1}.
\end{aligned} \tag{8.1}$$

An additional transformation has been added to Eq. (8.1) to account for the parameters related to image transfer and capture. The camera location (the optical center of the lens acting as the center of projection), its orientation (viewing angle), the optical parameters, and the physical dimensions of the 3D object being imaged are all embedded in the intrinsic and extrinsic parameters of a given setup.

Despite its widespread use in modeling, a pinhole is not used in modern cameras. Generally, pinhole images are dark because the pinhole permits only a small number of rays to reach the image plane; a large pinhole causes blurring, and a small pinhole produces diffraction effects that dominate. These limitations can be eliminated by using an objective (a lens that collects all incoming rays from the scene). For general optical modeling, an infinitely thin lens is assumed to be at the optical center to emulate the characteristics of a pinhole [Fig. 8.4(a)]. However, the finite thickness of the lens will add an inherent defocusing error due to the separation between the two principal planes that is usually discounted in a first analysis [Figs. 8.4(b) and (c)]. In optical modeling, the image coordinates' origin is placed at the *principal point* (center of the image plane/sensor). In imaging software, the image points (pixels) are assigned coordinates with respect to the top left corner of the image sensor as viewed from the camera. An origin transfer from the center of the optical image plane is a default setting in image capture/storage hardware.

Since depth information is not explicit in 2D image formation, Eq. (8.1) uses *homogeneous coordinates* to analytically embed the z-coordinate value [Fig. 8.4(a)] in the location of the projection plane (image plane) with respect to the center of projection. Homogeneous coordinates are extensively used in computer graphics for consistency and uniformity in mathematical modeling of rotational and translational motion.[3–6] In the transformation of a point from its physical coordinates to the homogeneous coordinates, its dimension is augmented by introducing the scaling factor w. This procedure follows three conventions:

1. An n-dimensional point in homogenous coordinates has $n - 1$ dimensions in the physical (world) coordinates.
2. For conversion from homogeneous coordinates to physical coordinates, all homogeneous coordinate values are first divided by the arbitrary scaling factor w.
3. Following the conversion to physical coordinates, the last (nth) elements in the homogeneous coordinates are deleted (Table 8.1).

Table 8.1 Relationship between the physical and homogeneous coordinate systems.

Location	Physical coordinates	Homogeneous coordinates
Image point	$p_i \Rightarrow [y_i \quad z_i]$	$p_i^h \Rightarrow [wx_i \quad wy_i \quad w]$
Object point	$p_o \Rightarrow [x_o \quad y_o \quad z_o]$	$p_o^h \Rightarrow [wx_o \quad wy_o \quad wz_o \quad w]$

A key feature of the homogeneous presentation is *projective equivalence*, i.e., if a line projects on several planes, the coordinates of the projection points on these planes are related by the scaling factor w. Thus, if a line $\mathcal{L}$ through an

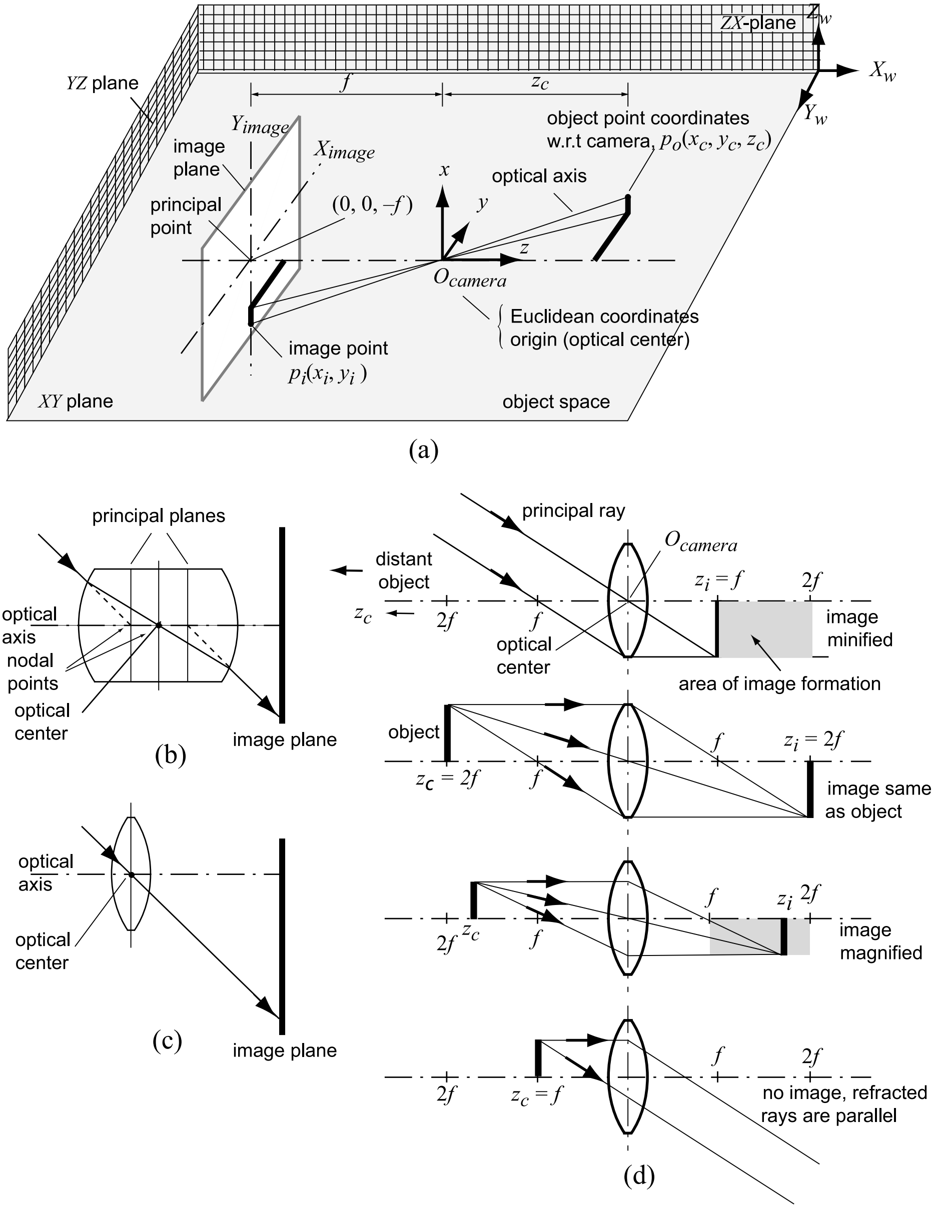


Figure 8.4 (a) Conversion of Euclidean coordinates into image coordinates through perspective projection with a pinhole. This configuration uses the simplest geometry with the optical axis aligned with the z axis of the world coordinate system; the object coordinates $p_o(x_c, y_c, z_c)$ are given with respect to the camera coordinate system, with O_{camera} as the origin. (b) Projection lines through a thick lens. (c) Ideal projection lines through a pinhole (an infinitely thin lens). (d) Images formed by thin converging lenses. For machine vision applications $z_c \gg f$; consequently, the minified image is placed at the focal point of the lens ($z_i = f$). For convenience, only the principal rays are shown.

object point p_o intersects three projection planes Σ_1, Σ_2, and Σ_3 that are defined by three scaling factors w_1, w_2, and w_3, with $p_1^h = [w_1x_1 \;\; w_1y_1 \;\; w_1]^{\mathrm{T}}$ being the homogeneous coordinates of the intersection point on the plane Σ_1, then the homogeneous coordinates of the intersection of $\mathcal{L}$ on the other two planes are $p_2^h = [w_2x_1 \;\; w_2y_1 \;\; w_2]^{\mathrm{T}}$ and $p_3^h = [w_3x_1 \;\; w_3y_1 \;\; w_3]^{\mathrm{T}}$ (shown in Fig. 8.5). In homogeneous coordinate notation, these three points are said to be projectively equivalent with their coordinates derived as $[kx_i \;\; ky_i \;\; k]$, where k can assume any of the three values of $w_i = z_i/f_i$, for $i = 1, 2, 3$. The values of these *weights* or *scaling factors* from the 3D object (or Euclidean) space to the 2D image plane are derived in Sec. 8.2.

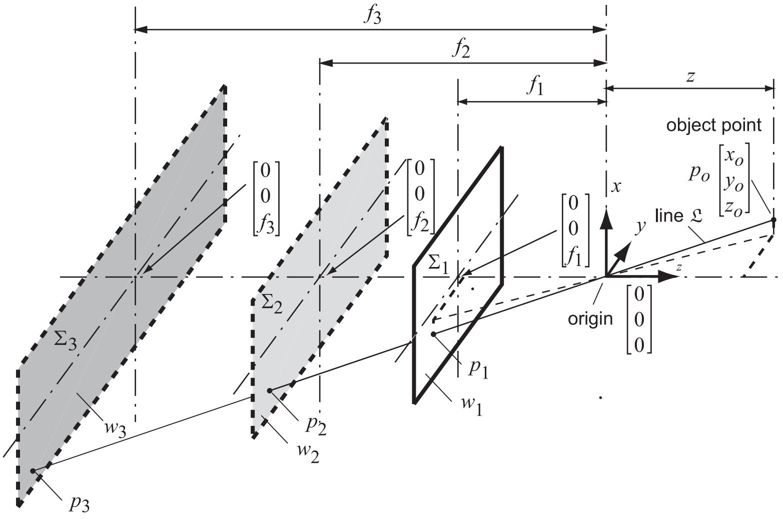

f_3
f_2
f_1
z
object point
p_o
$\begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \\ f_3 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \\ f_2 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \\ f_1 \end{bmatrix}$
x
y
line $\mathfrak{L}$
z
Σ_1
Σ_2
Σ_3
origin
$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
p_1
w_1
p_2
w_2
w_3
p_3

Figure 8.5 Intersection of a line with three projection planes defined by three scaling factors w_1, w_2, and w_3.

8.2 Ideal Intrinsic Model

The ideal intrinsic model is a first-order approximation of the imaging process that projects all object points onto a projection plane along projection lines through a single viewpoint. This viewpoint is placed at the center of the lens (*optical center*, O_{camera}). Using the parameters in Fig. 8.4(a) with $z_c > f$, the Gaussian form of the lens equation along the optical axis is [Sec. 3.3, Eq. (3.6a)]

$$\frac{1}{z_c} + \frac{1}{z_i} = \frac{1}{f}, \tag{8.2a}$$

where s_o and s_i replace z_c and z_i, respectively. By rearranging some terms, the lateral (or transverse) magnification can be derived as

$$M = \frac{z_i}{z_c} = \frac{f}{z_c - f}. \tag{8.2b}$$

In these derivations, magnification is the ratio of the image size to the object size. When $z_c \gg f$, the image is inverted with $|M| < 1$ [Fig. 8.4(d)]. Adding Eq. (8.2b) to the object-to-image magnifications along the x and y axes gives the perspective transformation relation

$$\frac{x_i}{x_c} = \frac{y_i}{y_c} = M = \frac{f}{f - z_c}. \tag{8.2c}$$

If Eq. (8.2c) is expressed in matrix notation, the object point in the Euclidean space gets transformed to the image point, which is also located in the Euclidean space, by

$$\begin{bmatrix} \left(\dfrac{f - z_c}{f}\right) x_i \\ \left(\dfrac{f - z_c}{f}\right) y_i \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix}. \tag{8.2d}$$

In vector notation, with the origin placed at the focal point (optical center) for an arbitrary scaling factor k between the object and the image points, Eq. (8.2d) becomes the following matrix equation:

$$\begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \\ 0 \end{bmatrix} = k \left\{ \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix} \right\} \quad \text{or} \quad \begin{bmatrix} -x_i \\ -y_i \\ f \end{bmatrix} = \begin{bmatrix} kx_o \\ ky_o \\ k(z_o - f) \end{bmatrix}. \tag{8.3a}$$

When a substitution is made for the value of $k = f/z_o - f$, Eq. (8.3a) reverts back to Eq. (8.2b). By using the relations in Table 8.1 and assigning a scaling factor $w = f - z_o/f$, the homogeneous coordinates of the image point can be given by

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} \triangleq \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \equiv \begin{bmatrix} wx_i \\ wy_i \\ w \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix}, \tag{8.3b}$$

where the scaling factor w subsumes the value of z_o, which corresponds to the location of the projection (image) plane from the origin or the depth information. To ensure conformability in matrix operations and make the depth appear in this projective transformation, the homogeneous coordinates of the object and the image points are given in the matrix form

$$\begin{bmatrix} wx_i \\ wy_i \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = P \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}, \tag{8.4}$$

where the 3×4 matrix P is the perspective transformation matrix that converts the 3D object point $p_o(x_c, y_c, z_c)$ into a 2D image point $p_i(x_i, y_i)$ on the image plane in Fig. 8.4(a). In Eq. (8.4), P is the intrinsic camera model within Eq. (8.1) and represents a nonlinear projection of the 3D coordinates of the object point with respect to the camera coordinate system onto a 2D plane. This nonlinear projection is due to division by a coordinate value in the perspective projection relation [Eq. (8.2a)].

Using the notation in Fig. 8.1, the ideal case of no lens distortion corresponds to $(x_i, y_i) \equiv (x_i^{ud}, y_i^{ud}) = (x_i^d, y_i^d)$. Assuming that there is no spatial error during the image transfer from the sensor to the framegrabber, when an image sensor spans the optical image plane, the intensity value g_{xy} at the analog image coordinate (x_i, y_i) corresponds to the sensor pixel output at the discrete location $(x_i/\Delta_{sx}, y_i/\Delta_{sy})$. With a quantized gray level $g_{xy}|_q$, the sensor pixel pitch along the two axes is $\Delta_{sx} \times \Delta_{sx}$. All of these pixel locations and their intensity values are then transferred through the clocking circuitry to the framegrabber and stored as a captured image frame.

Although the pixel pitch and resolution are available from the sensor data sheets, the required object distance z_c from the optical center O_{camera} and the focal length are related to the lens magnification for a specific setup. If the object-to-image distance ($z_c + z_i$ = *sum of conjugate distances*) is known, the focal length for a given magnification may be derived from Eq. (8.5a) [using $M = z_i/z_c$ from Eq. (8.2b)]:

$$z_c + z_i = \left(1 + \frac{1}{M}\right) f + (M + 1) f \tag{8.5a}$$

or

$$f = \frac{z_c + z_i}{\frac{1}{M} + M + 2}. \tag{8.5b}$$

This general relation can be further modified with $z_i = f$ from Fig. 8.4(d) ($z_c > 2f$) to

$$|f| = \frac{z_c}{\frac{1}{M} + M + 1}. \tag{8.5c}$$

This concludes the numerical work related to camera setup for a given lens magnification factor, but some iterations may be necessary to combine the lens magnification and image resolution and meet application-specific requirements. Since the image must fit within the 2D sensor plane, the magnification is dictated by the FOV dimension and image format. For example, with a half-inch-format camera with an image plane of dimensions 6.4 mm × 4.8 mm (Sec. 3.3.3.6, Fig. 3.8), the magnification along the horizontal (x axis) and vertical (y axis) directions are 6.4/horizontal object size (mm) and 4.8/vertical object size (mm), respectively. In contrast, the choice of image sensor is based on the resolution level required of the captured image, which is defined as the ratio of the minimum resolvable object size to the sensor pixel pitch. Although these two parameters are related to the image format (number of pixels × pixel pitch), a tradeoff between the two parameters is often necessary. For instance, given a lens magnification of 0.1 (image to object) with a half-inch-format sensor, the size of the FOV is 64 mm × 48 mm. If the image sensor in the camera has a pixel pitch of 9 μm × 9 μm, then the minimum resolvable object size (resolution) in the captured image is 90 μm × 90 μm. However, if a larger FOV is required such as a factor-of-5 increase to 320 mm × 240 mm, the lens magnification would need to be 0.02. With the same image sensor, the captured image would have a resolution of 450 μm × 450 μm. This reduction in resolution with increase in object size (or FOV) for a given image sensor is referred to in the optical design literature as the *space–bandwidth product*. [The minimum resolvable object size corresponds to the dimension in the target scene spanned by one pixel. For measurement, the feature size is at least two times the resolution (*Nyquist sampling*; see Sec. 6.3.1 and Appendix B).]

The intrinsic model in Eq. (8.4) is a nonlinear projection of 3D camera coordinates onto a 2D image plane and must be added to an extrinsic model to complete Eq. (8.1). In the absence of predefined world (or absolute) coordinates in any FOV, a set of abstract notations for defining the camera center and scene points is required to relate the physically measured object coordinates to their corresponding image plane coordinates. These concepts are introduced in Sec. 8.3.

8.3 Extrinsic Model

The calibration relations from the scene dimensions (in physical units of length) to the image dimension (in number of pixels) in Eq. (8.4) are direct because the optical axis is aligned with the z axis of the object space and the object coordinates are defined with respect to the camera coordinates. Although many imaging systems are configured as shown in Fig. 8.4(a), the resulting camera location and orientation may impose operational constraints or the FOV may not capture the required geometry of the target scene. For this reason, one of the configurations in Fig. 8.6 is more appropriate for general camera modeling. The point p_o is the object point in the 3D object (Euclidean) space with its coordinates given with respect to the world coordinate system, with the origin at O_w. In all subsequent derivations, the coordinates of p_o are marked with subscript w to conform to standard notation of rigid-body modeling. Note that this subscript is not related to the scaling variable w in Eq. (8.4); the variable w is the standard notation for scaling in the homogeneous coordinate system.

Modeling with an arbitrary camera–object geometry follows a three-stage process. In the first stage, the world coordinates are aligned with the camera coordinates through rigid-body rotations and translation movements. This alignment operation models the axial and planar relationships between the two coordinate systems and generates the object coordinates with respect to the camera coordinate system. In the second stage, perspective transformation is applied using Eq. (8.4) to derive the 2D image coordinates. In the final stage, scaling is applied because magnification is related to the image format and sensor resolution. Using the homogeneous coordinates, each of these operations leads to a transformation matrix. These operations are derived in the following subsections.[5–9]

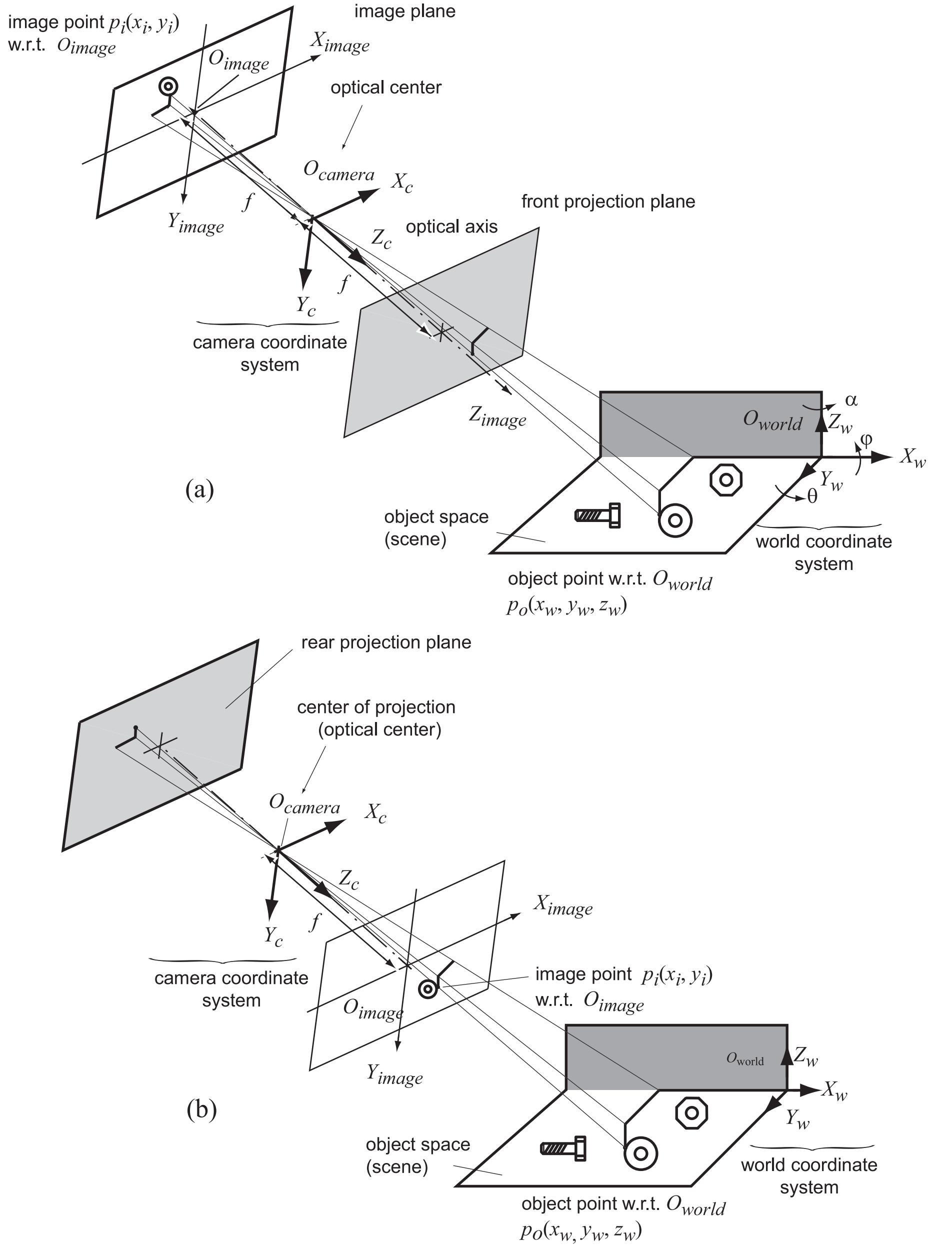


Figure 8.6 Image plane at an arbitrary orientation and axial location with respect to the world coordinates. Configurations for the image on (a) rear and (b) front projection planes. In both cases, the origin of the camera coordinate system is placed at the optical center (the geometric center of the lens). The image is inverted on the rear projection plane and is upright on the front projection plane. The rear projection model is used for extrinsic and general camera modeling in Sections 8.3 and 8.4, while the front projection is used for the Tsai calibration model in Sec. 8.5.

8.3.1 Translation

Translation is a linear movement of a point to align the origin of the world coordinates to the origin of the image plane. If the necessary translation movements are t_x, t_y, and t_z along the X_w, Y_w, and Z_w axes, then the origin of the object space after translation becomes

$$\begin{bmatrix} x_w^t \\ y_w^t \\ z_w^t \end{bmatrix} = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} x_w + t_x \\ y_w + t_y \\ z_w + t_z \end{bmatrix}. \tag{8.6a}$$

For consistency with the homogeneous coordinate system, this transformation is written in matrix form as

$$\begin{bmatrix} x_w^t \\ y_w^t \\ z_w^t \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = T \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{8.6b}$$

where T is the translation matrix.

8.3.2 Rotation

Rotation, a circular motion around a designated axis, is common in robotic modeling. For rotation about the z axis, only x and y coordinate values are affected. Using the notation in Fig. 8.7,

$$\begin{aligned} x_w^{rz} &= r\cos(\gamma + \alpha) = r\cos\gamma\cos\alpha - r\sin\gamma\sin\alpha = x_w\cos\alpha - y_w\sin\alpha \\ y_w^{rz} &= r\sin(\gamma + \alpha) = r\sin\gamma\cos\alpha + r\cos\gamma\sin\alpha = x_w\cos\alpha + x_w\sin\alpha. \end{aligned} \tag{8.7}$$

In homogeneous coordinates, the rotational transformation matrix $R_{z\alpha}$ around the z axis is given by

$$\begin{bmatrix} x_w^{rz} \\ y_w^{rz} \\ z_w^{rz} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = R_{z\alpha} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \tag{8.8a}$$

Similar derivations for rotations about the y and x axes lead to the following transformation matrices $R_{y\theta}$ and $R_{x\varphi}$, respectively:

$$\begin{bmatrix} x_w^{ry} \\ y_w^{ry} \\ z_w^{ry} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = R_{y\theta} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{8.8b}$$

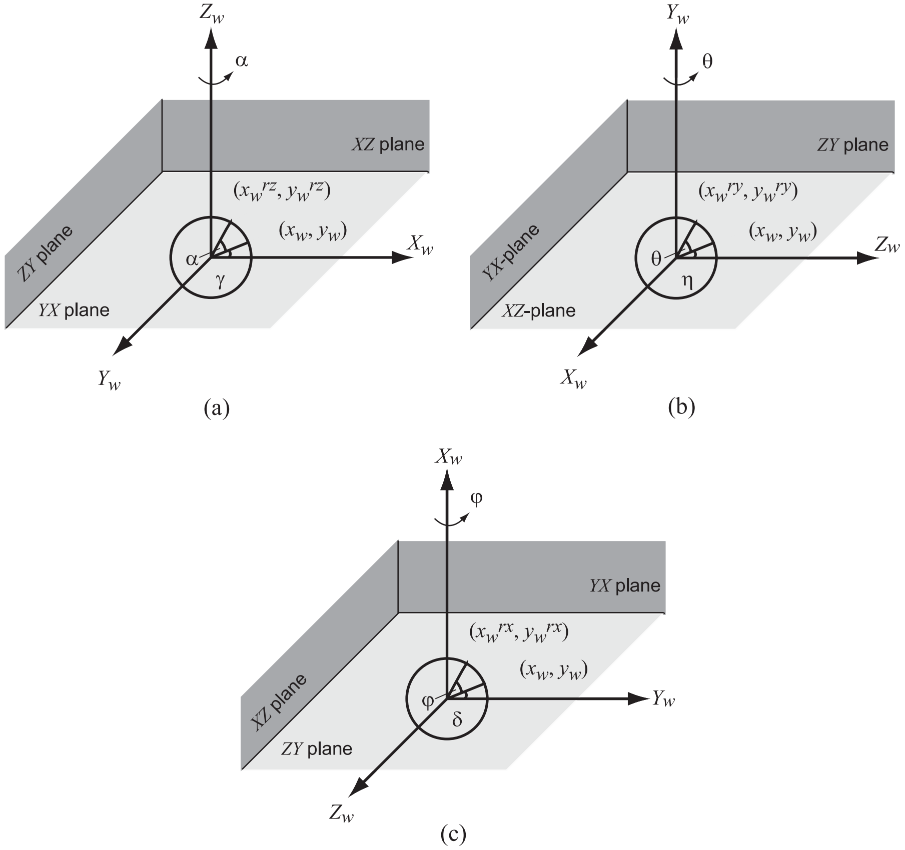


Figure 8.7 Rotation of a point around (a) the Z axis by α, (b) the Y axis by θ, and (c) the X axis by φ. All angles are positive in the counterclockwise rotation.

and

$$\begin{bmatrix} x_w^{rx} \\ y_w^{rx} \\ z_w^{rx} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi & 0 \\ 0 & \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = R_{x\varphi} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \qquad (8.8c)$$

Matrices for the inverse rotation are obtained by changing the signs of the respective angles. Since this is equivalent to a row–column transpose, the rotational matrices are orthogonal. [A matrix R is said to be orthogonal if $RR^{\mathrm{T}} = I$ (identity matrix).]

8.4 General Camera Model

Not including lens distortions, a direct correspondence between the image coordinates and the object coordinates may be obtained by first aligning the object coordinate system with the image coordinates and then taking the perspective projection of the object point on the image plane. The first operation involves a translation [Eq. (8.6)] and three rotations (one, two, or all three rotations, depending on the relative orientations of the two sets of coordinates) from Eq. (8.8) to obtain

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = R_{x\varphi}R_{y\theta}R_{z\alpha}T \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{8.9a}$$

where

$$\left.\begin{aligned} R_{x\varphi}R_{y\theta}R_{z\alpha} &= \begin{bmatrix} \cos\theta\cos\alpha & -\cos\theta\sin\alpha & \sin\theta & 0 \\ \sin\theta\sin\varphi\cos\alpha + \cos\varphi\sin\alpha & -\sin\theta\sin\varphi\sin\alpha + \cos\varphi\cos\alpha & -\cos\theta\sin\varphi & 0 \\ -\sin\theta\cos\varphi\cos\alpha + \sin\varphi\sin\alpha & \sin\theta\cos\varphi\sin\alpha + \sin\varphi\cos\alpha & \cos\theta\cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \text{and } T &= \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}\right\}. \tag{8.9b}$$

The second operation requires a perspective projection [Eq. (8.4)]:

$$\begin{bmatrix} wx_i \\ wy_i \\ w \end{bmatrix} = P \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}. \tag{8.9c}$$

The above operations yield the overall transformation matrix

$$\begin{bmatrix} wx_i \\ wy_i \\ w \end{bmatrix} = PR_{x\varphi}R_{y\theta}R_{z\alpha}T \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{8.9d}$$

where the camera parameters are embedded in $\{a_{ij}\}$. Each of these elements is related to the rotation angles, translation distance, and lens focal length.

The homogeneous coordinate scaling factor and the image coordinates are then derived as

$$\left.\begin{aligned} x_i &= \frac{wx_i}{w} = \frac{a_{11}x_w + a_{12}y_w + a_{13}z_w + a_{14}}{a_{31}x_w + a_{32}y_w + a_{33}z_w + 1} \\ y_i &= \frac{wy_i}{w} = \frac{a_{21}x_w + a_{22}y_w + a_{23}z_w + a_{24}}{a_{31}x_w + a_{32}y_w + a_{33}z_w + 1} \end{aligned}\right\} \tag{8.10a}$$

or

$$\left.\begin{array}{l} x_i = a_{11}x_w + a_{12}y_w + a_{13}z_w + a_{14} - a_{31}x_wx_i - a_{32}y_wx_i - a_{33}z_wx_i \\ y_i = a_{21}x_w + a_{22}y_w + a_{23}z_w + a_{24} - a_{31}x_wy_i - a_{32}y_wy_i - a_{33}z_wy_i \end{array}\right\}. \quad (8.10b)$$

Since the relative locations of the camera (image plane) and the object with respect to the object coordinates (Euclidean space) are specific to a particular camera–object geometry, the elements of $\boldsymbol{a} = \{a_{ij}\}$ must be derived for each camera setup using camera calibration. For numerical solutions, Eq. (8.10b) is expressed in this matrix form:

$$\boldsymbol{qa} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \quad (8.10c)$$

where

$$\boldsymbol{q} = \begin{bmatrix} x_w & y_w & z_w & 1 & 0 & 0 & 0 & 0 & -x_wx_i & -y_wx_i & -z_wx_i \\ 0 & 0 & 0 & 0 & x_w & y_w & z_w & 1 & -x_wy_i & -y_wy_i & -z_wy_i \end{bmatrix} \quad \text{and}$$

$$\boldsymbol{a} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{21} & a_{22} & a_{23} & a_{24} & a_{31} & a_{32} & a_{33} \end{bmatrix}^{\mathrm{T}}.$$

Since the three object coordinates are related to two image coordinates through 11 camera parameters, Eq. (8.10) is overdefined; therefore, no unique solution exists for the elements of $\boldsymbol{a}$. Although these camera parameters may be derived from Eq. (8.10c) for one pair of object–image coordinates, an improved numerical accuracy is obtained by recording the image coordinates for several known object points (*control points*).[10,11] If these control points are assigned by a set of n object–image pairs $[(x_w^k, y_w^k, z_w^k); (x_i^k, y_i^k)]$, $k \in 1, \ldots, n$, then the augmented form of Eq. (8.10c) becomes

$$\boldsymbol{Qa} = \begin{bmatrix} x_i^1 \\ \vdots \\ x_i^n \\ y_i^1 \\ \vdots \\ y_i^n \end{bmatrix} \text{ with } Q = \begin{bmatrix} x_w^1 & y_w^1 & z_w^1 & 1 & 0 & 0 & 0 & 0 & -x_w^1x_i^1 & -y_w^1x_i^1 & -z_w^1x_i^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_w^n & y_w^n & z_w^n & 1 & 0 & 0 & 0 & 0 & -x_w^nx_i^n & -y_w^nx_i^n & -z_w^nx_i^n \\ 0 & 0 & 0 & 0 & x_w^1 & y_w^1 & z_w^1 & 0 & -x_w^1y_i^1 & -y_w^1y_i^1 & -z_w^1y_i^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & x_w^n & y_w^n & z_w^n & 1 & -x_w^ny_i^n & -y_w^ny_i^n & -z_w^ny_i^n \end{bmatrix}. \quad (8.11a)$$

If the z coordinate is excluded from this calibration process, the pixel coordinate matrix $\boldsymbol{q}$ in Eq. (8.10c) has dimensions of 2 × 8. The result of using this reduced-order matrix (*8-point algorithm*) is known as *weak calibration*.

For $Q_{pinv} = [Q^{\mathrm{T}}Q]^{-1}Q^{\mathrm{T}}$, which marks the pseudo-inverse of Q as Q_{pinv}, the least-squared solution of $\boldsymbol{a}$ is derived as [by minimizing the error vector $\boldsymbol{e}$ in $\boldsymbol{Q\hat{a}} + \boldsymbol{e}$ using $d/d\boldsymbol{\hat{a}}(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}) = 0$ and $(d^2/d\boldsymbol{\hat{a}}^2)(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}) > 0$]

$$\hat{\boldsymbol{a}} = Q_{pinv} \begin{bmatrix} x_i^1 \\ \vdots \\ x_i^n \\ y_i^1 \\ \vdots \\ y_i^n \end{bmatrix}. \tag{8.11b}$$

From a numerical standpoint, Q must have full-column rank for the pseudo-inverse to exist. This condition produces two constraints: at least six control points are required, and these control points must be non-coplanar. The procedure for collecting control points with a calibration object is described in Sec. 8.5. It should be noted that this general camera calibration method is algorithmically convenient, but one limitation is that measurements for several control points are required for every calibration process.

8.5 Tsai Calibration[12–14]

Although the mathematical stages for deriving the camera parameters from the composite matrix in Eq. (8.11) are well established, a direct relationship between the explicit and implicit camera parameters does not exist in this transformation. To simplify and add a degree of physical interpretation, the abscissa and ordinate values of the image coordinates are processed separately, giving a two-stage modeling in the Tsai calibration procedure.

8.5.1 Camera model

The Tsai algorithm uses a simplified form with Euler angles being positive for clockwise rotations. The derivations in Sec. 8.3 assumed positive angles for the counterclockwise motion in line with the robotics convention.[7] For consistency with a wide selection of Tsai calibration literature, the rotational matrix in Eq. (8.12) below is restated with a change in signs: $\hat{\theta} = -\theta, \hat{\varphi} = -\varphi$ and $\hat{\alpha} = -\alpha$ (Fig. 8.8). Also, although the homogeneous coordinate models derived earlier are widely used for general modeling of rigid-body motion, camera models can be simplified by excluding the unity scaling factor in the general model. These two modifications, along with the vector form of the translation in Eq. (8.6b), reduce the world-coordinate-to-camera-coordinate transformation to

$$\begin{aligned}
\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\hat{\varphi} & \sin\hat{\varphi} \\ 0 & -\sin\hat{\varphi} & \cos\hat{\varphi} \end{bmatrix} \begin{bmatrix} \cos\hat{\theta} & 0 & -\sin\hat{\theta} \\ 0 & 1 & 0 \\ \sin\hat{\theta} & 0 & \cos\hat{\theta} \end{bmatrix} \begin{bmatrix} \cos\hat{\alpha} & \sin\hat{\alpha} & 0 \\ -\sin\hat{\alpha} & \cos\hat{\alpha} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \\
&= \begin{bmatrix} \cos\hat{\theta}\cos\hat{\alpha} & \cos\hat{\theta}\sin\hat{\alpha} & -\sin\hat{\theta} \\ \sin\hat{\theta}\sin\hat{\varphi}\cos\hat{\alpha} - \cos\hat{\varphi}\sin\hat{\alpha} & \sin\hat{\theta}\sin\hat{\varphi}\sin\hat{\alpha} + \cos\hat{\varphi}\cos\hat{\alpha} & \cos\hat{\theta}\sin\hat{\varphi} \\ \sin\hat{\theta}\cos\hat{\varphi}\cos\hat{\alpha} + \sin\hat{\varphi}\sin\hat{\alpha} & \sin\hat{\theta}\cos\hat{\varphi}\sin\hat{\alpha} - \sin\hat{\varphi}\cos\hat{\alpha} & \cos\hat{\theta}\cos\hat{\varphi} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \\
&\equiv \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = R(\hat{\theta}, \hat{\varphi}, \hat{\alpha}) \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}.
\end{aligned} \tag{8.12}$$

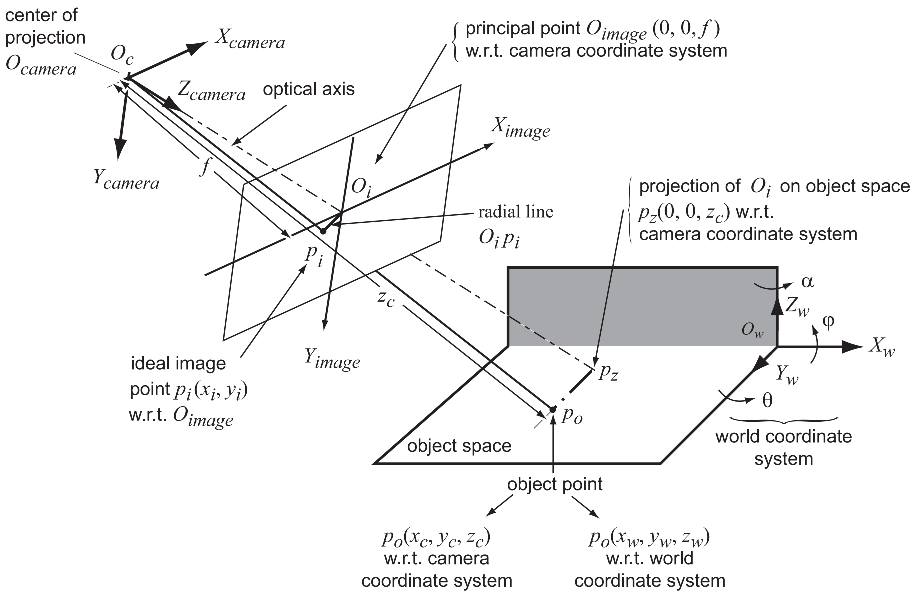

center of projection O_{camera}
O_c
X_{camera}
Z_{camera}
optical axis
Y_{camera}
f
principal point O_{image} (0, 0, f) w.r.t. camera coordinate system
X_{image}
O_i
radial line $O_i p_i$
p_i
z_c
projection of O_i on object space $p_z(0, 0, z_c)$ w.r.t. camera coordinate system
α
Z_w
φ
O_w
X_w
Y_w
θ
p_z
p_o
ideal image point $p_i(x_i, y_i)$ w.r.t. O_{image}
Y_{image}
object space
world coordinate system
object point
$p_o(x_c, y_c, z_c)$ w.r.t. camera coordinate system
$p_o(x_w, y_w, z_w)$ w.r.t. world coordinate system

Figure 8.8 World and camera coordinate systems with front projection and ideal image projection point $p_i(x_i, y_i)$. In Fig. 8.1, this point p_i is marked as the undistorted image point p_i^{ud} with coordinates (x_i^{ud}, y_i^{ud}).

For the front projection configuration in Fig. 8.8 and the camera coordinate origin at the focal point (center of projection), the perspective transformation matrix becomes

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{z_c} & 0 & 0 \\ 0 & \frac{f}{z_c} & 0 \\ 0 & 0 & \frac{1}{z_c} \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}. \tag{8.13}$$

If the above two equations are combined, in the absence of any lens distortion (the effects of lens distortion are considered in Sec. 8.5.5), the projection of the object point $p_o(x_w, y_w, z_w)$ to the image plane gives the following image coordinates:

$$x_i = f\frac{x_c}{z_c} = f\frac{r_1 x_w + r_2 y_w + r_3 z_w + t_x}{r_7 x_w + r_8 y_w + r_9 z_w + t_z} \tag{8.14a}$$

and

$$y_i = f\frac{y_c}{z_c} = f\frac{r_4 x_w + r_5 y_w + r_6 z_w + t_y}{r_7 x_w + r_8 y_w + r_9 z_w + t_z}. \tag{8.14b}$$

Since the z axis of the camera coordinate system (optical axis) is perpendicular to the image plane, the radial line $O_i p_i$ on the image plane is parallel to the vertical projection line $p_z p_o$ from the object point to the z axis of the camera plane, as marked in Fig. 8.8. Consequently, the cross product of the two vectors $\overrightarrow{O_i p_i} \times \overrightarrow{p_z p_o}$ is a null vector. As these two vectors are both perpendicular to the $O_c Z_{camera}$ axis, $[x_i, y_i] \times [x_c, y_c] = x_i y_c - x_c y_i = 0$, or $x_i = y_i \frac{x_c}{y_c}$; when combined with Eq. (8.14), this gives

$$x_i = y_i \frac{r_1 x_w + r_2 y_w + r_3 z_w + t_x}{r_4 x_w + r_5 y_w + r_6 z_w + t_y}. \tag{8.15}$$

This image abscissa forms the geometric basis for the first stage of the calibration procedure.

8.5.2 Scaling and origin transfer

The collection of ideal pixel locations (x_i, y_i) and their image values are transferred from the image sensor to the image acquisition hardware by timing circuits. These circuits clock out the photosite outputs from individual sensor pixel locations to corresponding memory locations in the framegrabber. Due to uncertainties in the image transfer clocking circuitry, not all sensor pixels are likely to maintain geometrically equivalent locations in the framebuffer. Even a minor variation in the high-frequency transfer clock (Sec. 6.3) may introduce recognizable errors during pixel relocation in the frame store. Due to the line-by-line transfer of sensor pixel contents to the image buffer, the spatial properties in the image's vertical pixels are assumed to be preserved with greater certainty in the image buffer; the pixels along the horizontal axis are likely to be more susceptible to clocking variations. Not including optical distortions and with s as a scaling uncertainty factor along the x axis, the pixel at (x_i, y_i) in the image sensor will be located at (sx_i, y_i) in the captured image framebuffer. (For a rigorous analysis, an additional scaling factor must be added to the y coordinate value. This factor has been excluded here for consistency with the Tsai calibration literature.)

With the image sensor located to cover the entire image plane, the coordinates of the optical image point $p_i(x_i, y_i)$ are given with respect to the origin at the center of the image sensor. However, for numerical convenience, the origin in the stored image frame is located at the top left corner. Assuming that the image sensor and the capture hardware have the spatial resolution $N_{sx} \times N_{sy}$, the sensor image at (x_i, y_i) becomes the retrievable and addressable pixel at (x_{fi}, y_{fi}) in the framebuffer. This origin transfer, along with the horizontal uncertainty scaling above, is defined by the spatial transformation

$$\begin{bmatrix} x_{fi} \\ y_{fi} \\ 1 \end{bmatrix} = \begin{bmatrix} s & 0 & \frac{N_{sx}}{2} \\ 0 & 1 & \frac{N_{sy}}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{8.16a}$$

and

$$\begin{aligned}\bar{x} &= x_{fi} - \frac{N_{sx}}{2} = sx_i\\ \bar{y} &= y_{fi} - \frac{N_{sy}}{2} = y_i,\end{aligned} \tag{8.16b}$$

where $(\bar{x}, \bar{y})$ are biased pixel locations in the framebuffer. These locations may be derived directly from the stored image pixel (x_{fi}, y_{fi}) address by a coordinates' bias of $(-N_{sx}/2, -N_{sy}/2)$. By adding the scaling uncertainty and the origin shifting transformation to the sensor pixel coordinates in Eqs. (8.14) and (8.15), the following linear homogeneous equations are derived in terms of the eight calibration parameters:

$$\left.\begin{array}{l}\bar{x} = \bar{y}\dfrac{s(r_1x_w + r_2y_w + r_3z_w + t_x)}{r_4x_w + r_5y_w + r_6z_w + t_y}\\ \text{or}\\ \left(\dfrac{r_4}{t_y}x_w + \dfrac{r_5}{t_y}y_w + \dfrac{r_6}{t_y}z_w + 1\right)\bar{x} = s\left(\dfrac{r_1}{t_y}x_w + \dfrac{r_2}{t_y}y_w + \dfrac{r_3}{t_y}z_w + \dfrac{t_x}{t_y}\right)\bar{y}\end{array}\right\} \tag{8.17a}$$

and

$$\left.\begin{array}{l}\bar{y}_i = f\dfrac{r_4x_w + r_5y_w + r_6z_w + t_y}{r_7x_w + r_8y_w + r_9z_w + t_z}\\ \text{or}\\ (r_4x_w + r_5y_w + r_6z_w + t_y)f - \bar{y}t_z = (r_7x_w + r_8y_w + r_9z_w)\bar{y}\end{array}\right\}. \tag{8.17b}$$

Equation (8.17a) provides a basis to derive the elements of $R(\hat{\theta}, \hat{\varphi}, \hat{\alpha})$ and (t_x, t_y), while Eq. (8.17b) is used for deriving f and t_z.

8.5.3 Stage 1 calibration: Parameters embedded in image abscissa

Since the physical location of the object point p_o is known, the measured values of its world coordinates (x_w, y_w, z_w) yield the biased pixel abscissa $\bar{x}$ in terms of the seven extrinsic parameters and one intrinsic parameter. Defining $b_1 = s\frac{r_1}{t_y}$, $b_2 = s\frac{r_2}{t_y}$, $b_3 = s\frac{r_3}{t_y}$, $b_4 = s\frac{t_x}{t_y}$, $b_5 = \frac{r_4}{t_y}$, $b_6 = \frac{r_5}{t_y}$ and $b_7 = \frac{r_6}{t_y}$ as unknown variables and rearranging the terms, Eq. (8.17a) may be restated in matrix form as

$$\begin{bmatrix}\bar{y}x_w & \bar{y}y_w & \bar{y}z_w & \bar{y} & -\bar{x}x_w & -\bar{x}y_w & -\bar{x}z_w\end{bmatrix}\begin{bmatrix}b_1\\ b_2\\ b_3\\ b_4\\ b_5\\ b_6\\ b_7\end{bmatrix} = \bar{x}. \tag{8.18a}$$

For a collection of *n* control (or *tie*) points in the world coordinate system and their corresponding image point abscissa, Eq. (8.18a) can be augmented to

$$\left.\begin{array}{l}\begin{bmatrix} \bar{y}^1 x_w^1 & \bar{y}^1 y_w^1 & \bar{y}^1 z_w^1 & \bar{y}^1 & -\bar{x}^1 x_w^1 & -\bar{x}^1 y_w^1 & -\bar{x}^1 z_w^1 \\ \bar{y}^2 x_w^2 & \bar{y}^2 y_w^2 & \bar{y}^2 z_w^2 & \bar{y}^2 y_i^2 & -\bar{x}^2 x_w^2 & -\bar{x}^2 y_w^2 & -\bar{x}^2 z_w^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \bar{y}^{n-1} x_w^{n-1} & \bar{y}^{n-1} y_w^{n-1} & \bar{y}^{n-1} z_w^{n-1} & \bar{y}^{n-1} & -\bar{x}^{n-1} x_w^{n-1} & -\bar{x}^{n-1} y_w^{n-1} & -\bar{x}^{n-1} z_w^{n-1} \\ \bar{y}^n x_w^n & \bar{y}^n y_w^n & \bar{y}^n z_w^n & \bar{y}^n & -\bar{x}^n x_w^n & -\bar{x}^n y_w^n & -\bar{x}^n z_w^n \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} \bar{x}^1 \\ \bar{x}^2 \\ \cdot \\ \cdot \\ \cdot \\ \bar{x}^{n-1} \\ \bar{x}^n \end{bmatrix} \\ \text{or} \quad \bar{Q} B_{rT} = \bar{X} \end{array}\right\}, \tag{8.18b}$$

where (x_w^k, y_w^k, z_w^k) and $(\bar{x}^k, \bar{y}^k)$, $k = 1, 2, \ldots, (n-1), n$ make up the measured set of known object points and their corresponding pixels in the framebuffer. With $\bar{X} = \{\bar{x}^k\}$ and the object–image point matrix $\bar{Q} = \{\bar{q}^k\}$, and with its *k*th row given by $\{\bar{q}^k\} = \left[\bar{y}^k x_w^k \;\; \bar{y}^k y_w^k \;\; \bar{y}^k z_w^k \;\; \bar{y}^k \;\; -\bar{x}^k x_w^k \;\; -\bar{x}^k y_w^k \;\; -\bar{x}^k z_w^k\right]$, Eq. (8.18) makes up a set of *n* homogeneous equations in seven unknown elements of $B_{rT} = [b_1 \;\; b_2 \;\; b_3 \;\; b_4 \;\; b_5 \;\; b_6 \;\; b_7]^{\mathrm{T}}$. For $n > 7$, the over-determined Eq. (8.18b) can be solved by the least-squared method. These values are then used to determine the extrinsic parameters with the linear dependence of the row/columns of the rotational matrix $R(\hat{\theta}, \hat{\varphi}, \hat{\alpha})$:

$$R(\hat{\theta}, \hat{\varphi}, \hat{\alpha}) = \begin{bmatrix} r_1 & r_2 & \sqrt{1 - r_1^2 - r_2^2} \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & -\sqrt{1 - r_1^2 - r_2^2} \\ r_4 & r_5 & -sgn\sqrt{1 - r_4^2 - r_5^2} \\ r_7 & r_8 & r_9 \end{bmatrix}, \tag{8.19}$$

with $[r_7 \;\; r_8 \;\; r_9] = [r_1 \;\; r_2 \;\; r_3] \times [r_4 \;\; r_5 \;\; r_6]$, giving $r_7 = r_2 r_6 - r_5 r_3$, $r_8 = r_3 r_4 - r_6 r_1$, $r_9 = r_1 r_5 - r_4 r_2$, where *sgn* represents the sign of the expression $(r_1 r_4 + r_2 r_5)$. A substitution of the elements in B_{rT} derived above in the first row of $R(\hat{\theta}, \hat{\varphi}, \hat{\alpha})$ gives the amount of translation necessary along the *y* axis to align the world coordinate system with the camera coordinate system. This translation parameter along the *y* axis is computed (after completing the square-root operation) by

$$\begin{aligned} |t_y| &= \left[\left(\frac{r_4}{t_y}\right)^2 + \left(\frac{r_5}{t_y}\right)^2 + \left(\frac{r_6}{t_y}\right)^2\right]^{-\frac{1}{2}} = [(b_5)^2 + (b_6)^2 + (b_7)^2]^{-\frac{1}{2}} \\ t_y &= \overline{\mathrm{sgn}}|t_y|, \end{aligned} \tag{8.20}$$

where $\overline{\mathrm{sgn}}$ is +1 or −1 as determined by assessing the quadrant locations of the object point in the camera coordinate system and its image point on the image plane. This value is then substituted to derive

$$\begin{aligned}
&r_1 = \left(\frac{r_1}{t_y}\right) t_y = \frac{b_1 t_y}{s}, \quad r_2 = \left(\frac{r_2}{t_y}\right) t_y = \frac{b_2 t_y}{s}, \quad r_3 = -\sqrt{1 - r_1^2 - r_2^2},\\
&r_4 = \left(\frac{r_4}{t_y}\right) t_y = \frac{b_5 t_y}{s}, \quad r_5 = \left(\frac{r_5}{t_y}\right) t_y = \frac{b_7 t_y}{s}, \quad r_6 = \left(\frac{r_6}{t_y}\right) sgn\sqrt{1 - r_4^2 - r_5^2}.
\end{aligned} \tag{8.21}$$

The scaling parameter and t_x are then derived as

$$s = |t_y| \left[\left(\frac{sr_1}{t_y}\right)^2 + \left(\frac{sr_2}{t_y}\right)^2 + \left(\frac{sr_3}{t_y}\right)^2\right]^{\frac{1}{2}} = |t_y|[(b_1)^2 + (b_2)^2 + (b_3)^2]^{\frac{1}{2}}. \tag{8.22a}$$

$$t_x = \left(\frac{t_x}{t_y}\right) t_y = \frac{b_4 t_y}{s}. \tag{8.22b}$$

Since each projection line from the object point converges at the optical center, as viewed from the camera coordinates with the front projection model shown in Fig. 8.8, the object point p_o and its image point p_i are located in the same quadrants of the projection plane with the object plane being normal to the optical axis. (With back projection, the model object point and its image point will be located in diagonally opposite quadrants.) Since the z-axis coordinate is positive toward the image plane, the x and y coordinate values of p_o and p_i have the same sign. This geometric property provides an iterative basis to assign the polarity of the absolute value of t_y in Eq. (8.20), assuming that the world coordinate system has been aligned along the z axis of the camera coordinate system through rotation. The process begins by computing the rotational parameters in Eq. (8.19) and assuming that the value of t_y in Eq. (8.20) is positive. Once the noncentered tie point $p_{object}^{tie}(x_w^t, y_w^t, y_w^t)$ in the object space and its corresponding image pixel $p_{fi}^{tie}(x_{fi}^{tie}, y_{fi}^{tie})$ in the framebuffer are identified, the coordinates of this measured object point p_{object}^{tie} with respect to the camera coordinates $p_{object}^{tie}(x_{camera}, y_{camera})$ are then computed by

$$\begin{aligned}
x_{camera}^{tie} &= r_1 x_w^t + r_2 y_w^t + t_x,\\
y_{camera}^{tie} &= r_3 x_w^t + r_4 y_w^t + t_y,
\end{aligned} \tag{8.23a}$$

using Eq. (8.12) with $z_w = 0$.

By using the inverse transformation of Eq. (8.16a), the coordinates of the optical image point corresponding to p_{fi}^{tie} are then computed by

$$\begin{aligned}
\bar{x}_{optical}^{tie} &= \frac{1}{s}\left(x_{fi}^{tie} - \frac{N_{sx}}{2}\right),\\
\bar{y}_{optical}^{tie} &= \left(y_{fi}^{tie} - \frac{N_{sy}}{2}\right).
\end{aligned} \tag{8.23b}$$

If $(x_{camer}^{tie}, \bar{y}_{camera}^{tie})$ and $(\bar{x}_{fi}^{tie}, \bar{y}_{fi}^{tie})$ are located in the same quadrant, i.e., the two x-coordinate values have the same sign and the two y-coordinate values have the same sign, then the coordinate transformation and projection are correct and t_y will be positive; otherwise, $\overline{\text{sgn}} = -1$ and the relevant parameters are recomputed with negative t_y.

The algorithmic steps for Stage 1 are listed below:

1. Collect camera calibration data (input object and image points).
   a. Select $n \geq 7$ points in the 3D object space and mark their coordinates in physical units (mm) with respect to a world-coordinate system origin, and mark these as $(x_w^k, y_w^k, z_w^k),\ k = 1, 2, \ldots, (n-1), n$.
   b. Identify the corresponding image points for each of the above object points and record their pixel locations: $(x_p^k, y_p^k), k = 1, 2, \ldots, (n-1), n$.
2. Perform preprocessing with the above input calibration data ($N_{sx} \times N_{sy}$ available in the sensor specifications).
   a. For each image pixel above, derive $\bar{x}^k = x_{fi}^k - \frac{N_{sx}}{2}$ and $\bar{y}^k = y_{fi}^k - \frac{N_{sy}}{2}$.
   b. Construct the column vector $\bar{X} = \{\bar{x}^k\},\ k = 1, 2, \ldots, (n-1), n$.
   c. Construct the $7 \times n$ matrix $\bar{Q}$ with its kth row as

$$\{\bar{q}^k\} = \left[\bar{y}^k x_w^k \vdots\ \bar{y}^k y_w^k \vdots\ \bar{y}^k z_w^k \vdots\ \bar{y}^k \vdots\ -\bar{x}^k x_w^k \vdots\ -\bar{x}^k y_w^k \vdots\ -\bar{x}^k z_w^k\right].$$

   d. Construct its transpose $\bar{Q}^{\mathrm{T}}$ and derive $[\bar{Q}^{\mathrm{T}}\bar{Q}]^{-1}$.
   e. Construct $[\bar{Q}^{\mathrm{T}}\bar{Q}]^{-1}\bar{Q}^{\mathrm{T}}$.
3. Derive the matrix solution of Eq. (8.18b) as $B_{rT} = [\bar{Q}^{\mathrm{T}}\bar{Q}]^{-1}\bar{Q}^{\mathrm{T}}\bar{X}$.
4. Compute $\{r_i\},\ i = 1, 2, \ldots, 8, 9$ using Eq. (8.21).

In Tsai calibration work, a major preparatory task is the production of a target object to generate 3D calibration points. This task is described after the stage 2 steps.

8.5.4 Stage 2 calibration: Parameters related to image ordinate

By rearranging the terms on both sides, Eq. (8.17b) can be expressed in matrix form of two unknowns f and t_z as given by

$$\left[(r_4 x_w + r_5 y_w + r_6 z_w + t_y)\ \ -\bar{y}\right]\begin{bmatrix} f \\ t_z \end{bmatrix} = (r_7 x_w + r_8 y_w + r_9 z_w)\bar{y}. \tag{8.24a}$$

Using the same n tie points used earlier, the augmented matrix in Eq. (8.24b) is constructed to solve for two unknowns f and t_z using a least-squared method:

$$\left.\begin{aligned}&\begin{bmatrix} (r_4 x_w^1 + r_5 y_w^1 + r_6 z_w^1 + t_y) & -\bar{y}^1 \\ (r_4 x_w^2 + r_5 y_w^2 + r_6 z_w^2 + t_y) & -\bar{y}^2 \\ \vdots & \vdots \\ (r_4 x_w^{n-1} + r_5 y_w^{n-1} + r_6 z_w^{n-1} + t_y) & -\bar{y}^{n-1} \\ (r_4 x_w^n + r_5 y_w^n + r_6 z_w^n + t_y) & -\bar{y}^n \end{bmatrix}\begin{bmatrix} f \\ t_z \end{bmatrix} = \begin{bmatrix} (r_7 x_w^1 + r_8 y_w^1 + r_9 z_w^1)\bar{y}^1 \\ (r_7 x_w^2 + r_8 y_w^2 + r_9 z_w^2)\bar{y}^2 \\ \vdots \\ (r_7 x_w^{n-1} + r_8 y_w^{n-1} + r_9 z_w^{n-1})\bar{y}^{n-1} \\ (r_7 x_w^n + r_8 y_w^n + r_9 z_w^n)\bar{y}^n \end{bmatrix} \\ &\text{or}\quad \Gamma\begin{bmatrix} f \\ t_z \end{bmatrix} = D\bar{Y}\end{aligned}\right\}, \tag{8.24b}$$

where

$\Gamma = \{\gamma^k\}$ is an $n \times 2$ matrix with its kth rows given by $\gamma^k = \left[(r_4 x_w^k + r_5 y_w^k + r_6 z_w^k + t_y) \vdots -\bar{y}^k\right]$, $D = \{d^k\}$ is an n-column matrix with its kth column as $d^k = [(r_7 x_w^k + r_8 y_w^k + r_9 z_w^k)\bar{y}^k]$, and $\bar{Y} = \{\bar{y}^k\},\ k = 1, 2, \ldots (n-1), n$ for $n \geq 2$.

The algorithmic steps for the solution of the two unknown variables are as follows:

1. Collect camera calibration data (input object and image points) from stage 1.
   a. Select $n \geq 2$ points in the 3D object space and mark their coordinates in physical units (mm) with respect to a world coordinate system origin. Mark these coordinates as

$$(x_w^k, y_w^k, z_w^k),\ k = 1, 2, \ldots, (n-1), n.$$

   b. Identify the corresponding image points for each of the above object points and record their pixel locations as

$$(x_p^k, y_p^k),\ k = 1, 2, \ldots, (n-1), n.$$

2. Perform the preprocessing with the above input calibration data using the parameters from stage 1 for $k = 1, 2, \ldots, (n-1), n$.
   a. Derive the n rows $\gamma^k = \left[(r_4 x_w^k + r_5 y_w^k + r_6 z_w^k + t_y) \vdots -\bar{y}^k\right]$ and construct the $n \times 2$ matrix $\Gamma = \{\gamma^k\}$.
   b. Derive the n element $d^k = [(r_7 x_w^k + r_8 y_w^k + r_9 z_w^k)\bar{y}^k]$ and construct the column vector $D = \{d^k\}$.
   c. Construct the column vector $\bar{Y} = \{\bar{y}^k\},\ k = 1, 2, \ldots, (n-1), n$.
3. Construct transpose Γ^T and derive $[\Gamma^T\Gamma]^{-1}\Gamma$.
4. Derive the matrix solution of Eq. (8.24b) as $\left[\begin{smallmatrix} f \\ t_z \end{smallmatrix}\right] = [\Gamma^T\Gamma]^{-1}\Gamma^T D\bar{Y}$.

A flat or inclined surface containing a sea of squares with known center coordinates is widely used[15] as a calibration surface [Figs. 8.9(a) and (b)]. Although such coplanar object points simplify the process, they do not estimate the translation parameter in the Z direction t_z or the scaling uncertainty factor s. Multiple sets of calibration input data are required with a coplanar target surface placed at different z-axis locations with respect to the image plane. The general calibration process is more convenient if the object has a 3D calibrated surface [Fig. 8.9(c) and (d)].

8.5.5 Resolution and distortion

The assumption of ideal optics (no lens distortion) and identical pixel resolutions of the image sensor and the framegrabber provides reasonably accurate extrinsic and intrinsic camera parameters, but it may not always be applicable. These two aspects of camera calibration are briefly considered below.

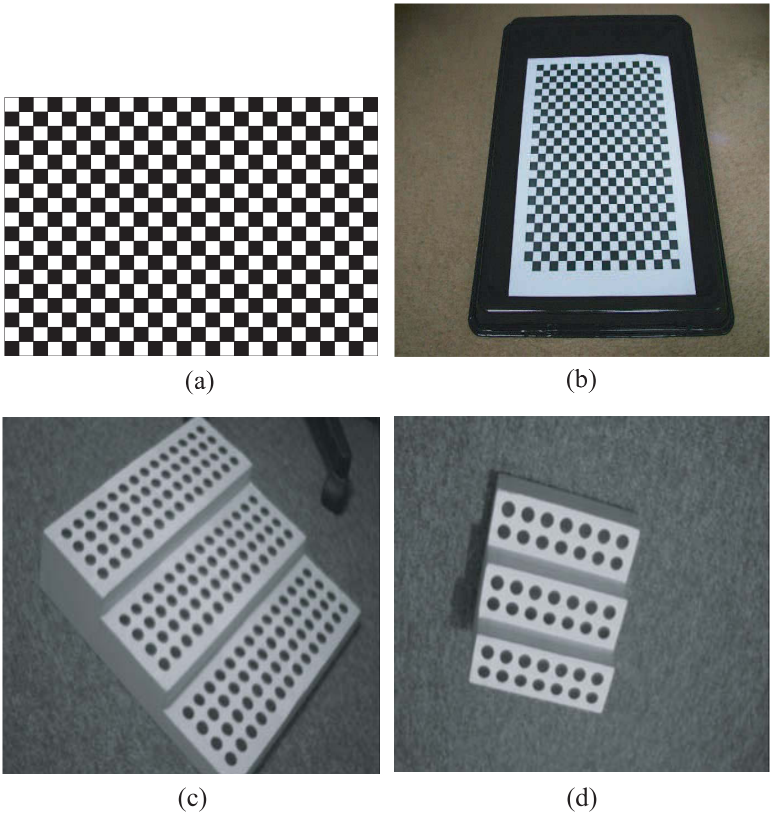


Figure 8.9 Calibration objects commonly used with the Tsai algorithm, including the coplanar *sea of squares* (a) orthogonal to the optical axis and (b) placed over a slanted plane. (c) and (d) Two 3D staircases with black circles. The size of the calibration object is chosen to encompass the camera FOV. The control (tie) points at the centers of the circles (square black spots are also used) are marked with known physically measured (x_w, y_w, z_w) coordinates with respect to a user-chosen world coordinate system (typically, the top left corner).

1. *Difference in resolutions:* Although the resolutions of the image sensor and the framegrabber are usually the same, some applications may require them to have different resolutions. For the discussions here, the sensor is assumed to have a pixel resolution of $N_{sx} \times N_{sy}$ and a pixel pitch of $\Delta_{sx} \times \Delta_{sy}$ with a framegrabber resolution of $N_{fx} \times N_{fy}$. As indicated earlier, in the absence of any lens distortion, the analog image coordinate (x_i, y_i) becomes an image pixel location $(x_i/\Delta_{sx}, y_i/\Delta_{sy})$ with an intensity level g_{xy}, and all sensor pixel locations and their intensity values are transferred and stored in the framegrabber/host processor as an image frame. To preserve the overall dimensional relationship and the aspect ratio between the sensor image and the captured image in the framegrabber, the location and intensity of the corresponding pixel becomes $\left(x_i/\beta_x, y_i/\beta_y, \, g_{xy}\big|_{quantized}\right)$, where $\beta_x = (N_{sx}/N_{fx})\Delta_{sx}$ and $\beta_y = (N_{sy}/N_{fy})\Delta_{sy}$. Using this notation and adding

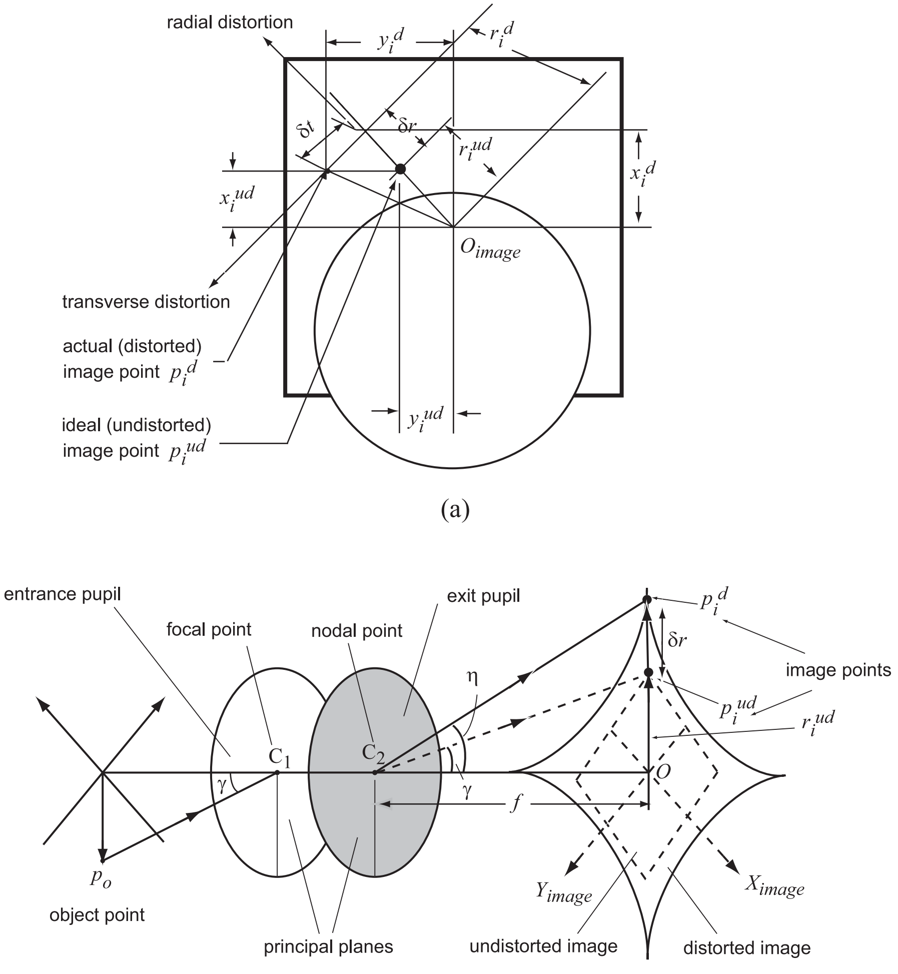
the horizontal scaling factor and origin transfer [Eq. (8.16a)], the pixel coordinate transformation from the sensor input to the stored image frame is given by

$$\begin{bmatrix} x_{fi} \\ y_{fi} \\ 1 \end{bmatrix} = \begin{bmatrix} \dfrac{s}{\beta_{fx}} & 0 & \dfrac{N_{fx}}{2} \\ 0 & \dfrac{1}{\beta_{fy}} & \dfrac{N_{fy}}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \tag{8.25}$$

2. *Lens distortion*: The derivations in stages 1 and 2 assume that there are no optical distortions in the perspective projection process; hence, Eq. (8.13) is the complete intrinsic model of the camera with front-projection geometry. With an ideal optical system, the mathematically projected image point $p_i(x_i, y_i)$ in Fig. 8.10 would become an ideal or *undistorted* image point. For notational convenience, this undistorted image point was marked as $p_i^{ud}(x_i^{ud}, y_i^{ud})$ in Fig. 8.1 (i.e., $x_i^{ud} \equiv x_i$ and $y_i^{ud} \equiv y_i$).

While the assumption of perfect optical components does not affect the general algorithmic work, a calibration process must account for lens distortions if precise measurements are to be made from captured images, especially since most of the commonly used machine vision cameras use lenses with relatively wide manufacturing tolerances. Lens distortions (Sec. 3.7) would cause the object point p_o to not project at the ideal undistorted location p_i^{ud} on the image plane; instead, it would project at a different location referred to as the *distorted* (actual) image point $p_i^d(x_i^d, y_i^d)$. The coordinate difference between these two image points is the vector sum of the optical distortions along the radial δr and transverse δt directions. The commonly used relation between the coordinates of p_i^{ud} and p_i^d is $\overline{r_i^d} = \overline{(r_i^{ud} + \delta r)} + \overline{\delta t}$, as shown in Fig. 8.10(a).

Transverse (or tangential) distortion is caused by lenses being tilted or mounted off the optical axis in the lens assembly. Transverse distortion moves the image point at right angles to the radial lines from the center of the lens toward its edge, while radial distortion places the actual image point away from the true location along the radial line. Radial distortion results when the principal ray enters the entrance pupil at angle γ and leaves the exit pupil at angle η [Fig. 8.10(b)]. The effect of these non-coinciding centers is either pincushion distortion or barrel distortion. Pincushion distortion occurs when an image point moves from its ideal location toward the lens edge ($\eta > \gamma$). Barrel distortion occurs when an image point moves toward the lens center ($\eta < \gamma$). With telephoto and zoom lens assemblies, the optical center in the equivalent pinhole model is at the nodal point rather than the focal point, moving the image plane forward and introducing an error in the projected image geometry. The degree of pincushion or barrel distortion is insignificant within the paraxial area but increases as the image point moves toward the edge of the lens.



(a)

(b)

Figure 8.10 (a) Lens distortion components along the radial and transverse directions [image point locations correspond to pincushion distortion with $\overline{r_i^d} = \overline{r_i^{ud}} + \overline{(\delta_r + \delta_t)}$]. (b) Image distortion due to non-coinciding principal planes.

The cameras used in most machine vision systems have negligible transverse distortion; therefore, the general lens model covers only radial distortion effects. With $\delta t = 0$, the principal point and the distorted and undistorted image points remain collinear. This condition is referred to as the *radial angular constraint*. Consequently, the projection line $p_o p_z$ in Fig. 8.8 remains unchanged, which keeps Eq. (8.15) valid for the distorted image point in Fig. 8.11. The degree of radial distortion is insignificant within the paraxial area but increases as the image point moves toward the lens edge. This varying magnification along the radial direction is modeled[16] in the photogrammetry literature by

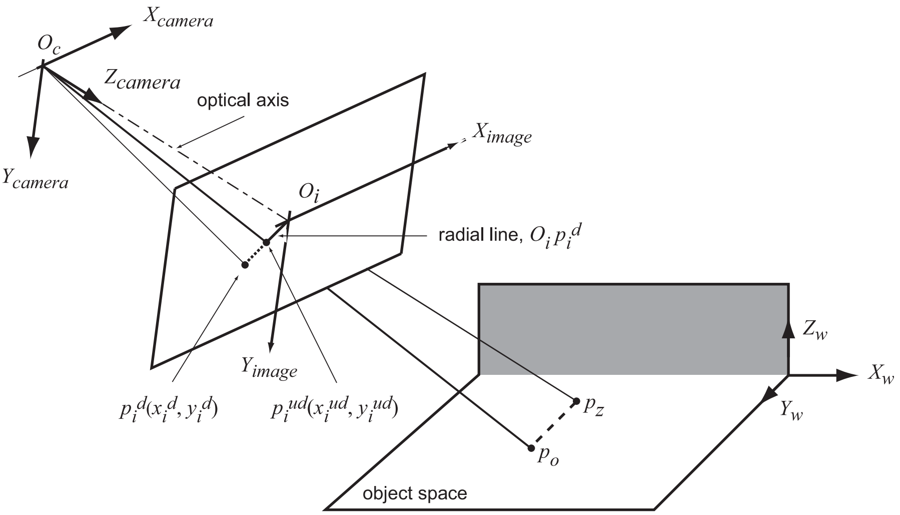


Figure 8.11 Front-projection model with a distorted image point.

$$\left.\begin{aligned}
x_i^{ud} &= x_i^d + \kappa_1 x_i^d (r_i^d)^2 + \kappa_2 x_i^d (r_i^d)^4 + \kappa_3 x_i^d (r_i^d)^6 + p_{ri1}[2(x_i^d)^2 + (r_i^{d2})^2] + 2p_{ri2} x_i^d y_i^d \\
&= x_i^d + \delta x_r \\
y_i^{ud} &= y_i^d + \kappa_1 y_i^d (r_i^d)^2 + \kappa_2 y_i^d (r_i^d)^4 + \kappa_3 y_i^d (r_i^d)^6 + p_{ri2}[2(y_i^d)^2 + (r_i^d)^2] + 2p_{ri1} x_i^d y_i^d \\
&= y_i^d + \delta y_r \\
(r_i^d)^2 &= (x_i^d)^2 + \varepsilon^2 (y_i^d)^2 \\
\overline{\delta r} &= \delta x_r + \delta y_r
\end{aligned}\right\}, \tag{8.26a}$$

where ε is the image aspect ratio (generally assumed to be unity in image sensor geometry), and $\kappa_\bullet$ is the radial distortion coefficient (positive for barrel distortion and negative for pincushion distortion) for a square image frame ($\varepsilon = 1$);

$$\left.\begin{aligned}
x_i^{ud} &= x_i^d + \kappa_1 x_i^d [(x_i^d)^2 + (y_i^d)^2] + O\{(r_i^d)^4\} \\
y_i^{ud} &= y_i^d + \kappa_1 y_i^d [(x_i^d)^2 + (y_i^d)^2] + O\{(r_i^d)^4\}
\end{aligned}\right\}, \tag{8.26b}$$

where $O\{(r_i^d)^4\}$ denotes distortion terms of power 4 and above. Since experimental observations indicate that the first-order correction accounts for nearly 90% of the total distortion effects, contributions from $O\{(r_i^d)^4\}$ are discounted in a first analysis for simplicity. This leads to the generally used distortion estimate model:

$$\left.\begin{aligned}
\tilde{x}_i^{ud} &\approx x_i^d + \kappa_1 x_i^d [(x_i^d)^2 + (y_i^d)^2] \\
\tilde{y}_i^{ud} &\approx y_i^d + \kappa_1 y_i^d [(x_i^d)^2 + (y_i^d)^2]
\end{aligned}\right\}. \tag{8.26c}$$

The commonly used transverse (tangential) distortion model in photogrammetry is (assuming zero distortion along the optical axis)

$$x_i^d = x_i^{ud} - x_i^d(\xi_1 r_d^2 + \xi_2 r_d^4 + \xi_3 r_d^6) \quad \text{and} \quad y_i^d = y_i^{ud} - y_i^d(\xi_1 r_d^2 + \xi_2 r_d^4 + \xi_3 r_d^6). \quad (8.27)$$

Equations (8.25) and (8.26c) provide the transformation matrices for blocks 3 and 4 in Fig. 8.1, where $\Re\{p_{fi}(x_{fi}, y_{fi})\}$ is the captured image frame in the host that embodies imperfections in the lens and the capturing hardware. Consequently, if undistorted pixel coordinates are required, then the inverse transformation of Eqs. (8.26c) and (8.25) is applied to $\Re(x_{fi}, y_{fi})$ to estimate the undistorted (distortion-corrected) pixel locations $\Re\{\tilde{p}_i^{ud}(\tilde{x}_i^{ud}, \tilde{y}_i^{ud})\}$. Figure 8.12 illustrates this correction with $O\{(r_i^d)^4\} = 0$.

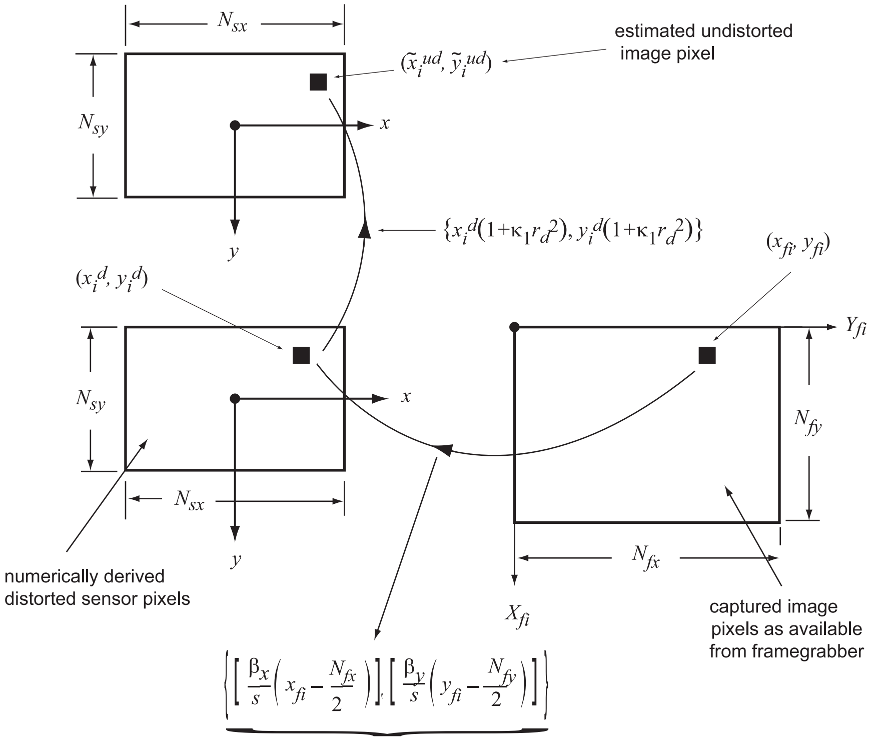

N_{sx}
estimated undistorted image pixel
$(\tilde{x}_i^{ud}, \tilde{y}_i^{ud})$
N_{sy}
x
y
$\{x_i^d(1+\kappa_1 r_d^2), y_i^d(1+\kappa_1 r_d^2)\}$
(x_{fi}, y_{fi})
(x_i^d, y_i^d)
Y_{fi}
N_{sy}
x
N_{fy}
N_{sx}
y
N_{fx}
numerically derived distorted sensor pixels
X_{fi}
captured image pixels as available from framegrabber
$\left\{\left[\frac{\beta_x}{s}\left(x_{fi} - \frac{N_{fx}}{2}\right)\right], \left[\frac{\beta_y}{s}\left(y_{fi} - \frac{N_{fy}}{2}\right)\right]\right\}$
shifting and scaling

Figure 8.12 Distortion correction with the inverse transformation of a captured image frame. Since only the first-order distortion term is taken, the distortion-corrected pixel locations are marked as estimates $(\tilde{x}_i^{ud}, \tilde{y}_i^{ud})$ rather than as the actual values (x_i^{ud}, y_i^{ud}) in Fig. 8.11 due to the assumption that $O\{(r_i^d)^4\} = 0$.

Since the ideal projection relationships are $x_i^{ud} = x_i = f\frac{x_c}{z_c}$ and $y_i^{ud} = y_i = f\frac{y_c}{z_c}$ [from Eq. (8.14) and Fig. 8.8], using the radial distortion model above, Eq. (8.14) may be restated as

$$[1+\kappa_1(\hat{r}_i^d)^2]x_i^d = \tilde{x}_i^{ud} \approx x_i = f\frac{r_1x_w + r_2y_w + r_3z_w + t_x}{r_7x_w + r_8y_w + r_9z_w + t_z} \tag{8.28a}$$

and

$$[1+\kappa_1(\hat{r}_i^d)^2]y_i^d = \tilde{y}_i^{ud} \approx y_i = f\frac{r_4x_w + r_5y_w + r_6z_w + t_y}{r_7x_w + r_8y_w + r_9z_w + t_z}. \tag{8.28b}$$

For precise distortion correction from the framegrabber image pixels, the calibrator must account for all coefficients in $O\{(r_i^d)^4\}$ of Eq. (8.26a). However, the common practice is to use the first-order distortion coefficient κ_1 to estimate the effects of the higher-order coefficients by minimizing the sum-squared error $J = \sum_{k=1}^{n}[(\tilde{x}_i^{udk} - x_i^k)^2 - (\tilde{y}_i^{udk} - y_i^k)^2] \leq J_{\min}$ for the control points collected during calibration in stages 1 and 2. With the value of κ_1 derived by iteration using Fig. 8.13, the coordinates of the undistorted (ideal) image frame $\Re\{(\tilde{x}_i^{ud}, \tilde{y}_i^{ud})\}$ are estimated from the captured image frame $\Re\{(x_{fi}, y_{fi})\}$ by

$$\left.\begin{aligned} x_i^{dk} &= \frac{\beta_{fx}}{s}\left(x_{fi}^k - \frac{N_{fx}}{2}\right)\\ y_i^{dk} &= \frac{\beta_{fy}}{s}\left(y_{fi}^k - \frac{N_{fy}}{2}\right)\\ \tilde{x}_i^{udk} &= [1+(\kappa_0+\Delta\kappa)(\hat{r}_i^d)^2]x_i^{dk}\\ \tilde{y}_i^{udk} &= [1+(\kappa_0+\Delta\kappa)(\hat{r}_i^d)^2]y_i^{dk} \end{aligned}\right\}. \tag{8.29}$$

The Tsai calibration routine uses the Levenberg–Marquart optimization algorithm. The implementation details are documented in the literature,[17] and an account of the error analysis is documented elsewhere.[14] For standard lenses in machine vision systems, κ_1 is typically quoted to be in the range of 10^{-8} to 10^{-6} pixels per mm^2 of sensor area.[16] The horizontal timing error uncertainty scaling factor is related to pixel jitter and typically estimated to be around 1.02 to 1.05. Though this value is not significant, the choice of pixel resolution with sensor format size may need to be assessed in the context of the required measurement accuracy.

8.6 Stereo Imaging

The measurement of the distance between a sensor and a target object (*range* or *depth*) is a common task in robotics and automated assembly processes.[9,18,19] A range sensor uses the geometric property for an arbitrary triangle made up of the optical centers of two cameras and the target (Fig. 8.14):

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}. \tag{8.30a}$$

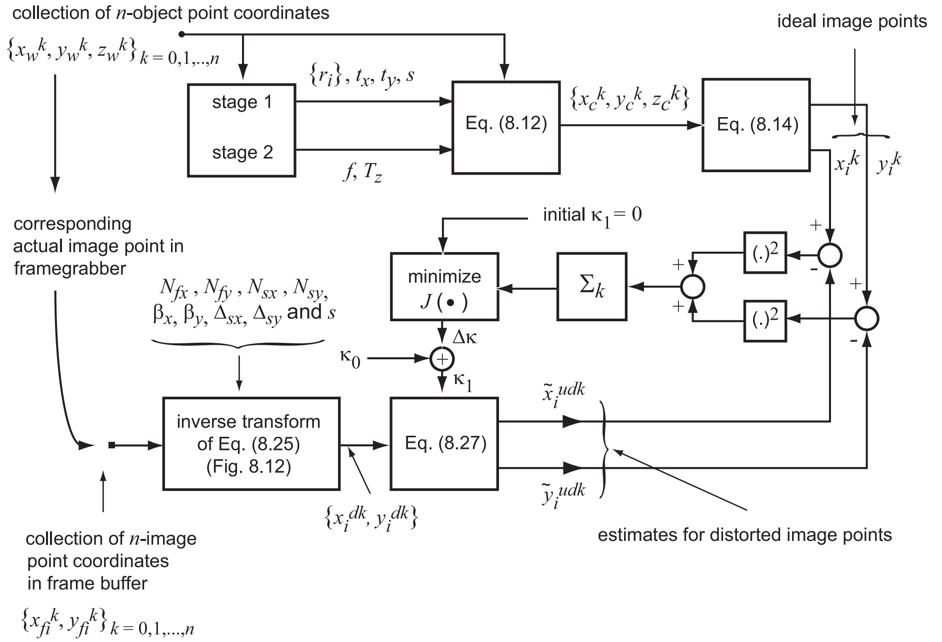

collection of n-object point coordinates
$\{x_w^k, y_w^k, z_w^k\}_{k=0,1,..,n}$
ideal image points
stage 1
stage 2
$\{r_i\}, t_x, t_y, s$
f, T_z
Eq. (8.12)
$\{x_c^k, y_c^k, z_c^k\}$
Eq. (8.14)
x_i^k y_i^k
corresponding actual image point in framegrabber
initial $\kappa_1 = 0$
minimize $J(\bullet)$
Σ_k
$(.)^2$
$(.)^2$
$N_{fx}, N_{fy}, N_{sx}, N_{sy}, \beta_x, \beta_y, \Delta_{sx}, \Delta_{sy}$ and s
$\Delta\kappa$
κ_0
κ_1
inverse transform of Eq. (8.25) (Fig. 8.12)
Eq. (8.27)
$\tilde{x}_i^{udk}$
$\tilde{y}_i^{udk}$
$\{x_i^{dk}, y_i^{dk}\}$
estimates for distorted image points
collection of n-image point coordinates in frame buffer
$\{x_{fi}^k, y_{fi}^k\}_{k=0,1,...,n}$

Figure 8.13 Iterative derivation of the first-order distortion coefficient. The initial value κ_0 is chosen to be zero for the distortion-free image coordinates [Eq. (8.14)].

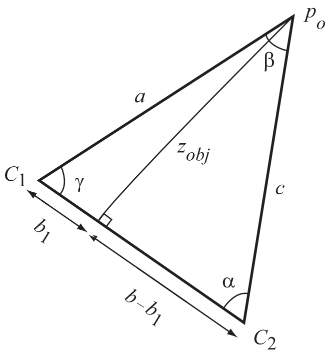


Figure 8.14 Triangulation parameters for distance measurement.

Using triangulation with the parameters in Fig. 8.14, the target object depth from the baseline is given by

$$\begin{aligned} z_{obj} &= b_1 \tan\gamma = (b - b_1)\tan\alpha, \\ b_1 &= b\frac{\cos\gamma \sin\alpha}{\sin(\alpha+\gamma)}, \\ (b - b_1) &= b\frac{\cos\alpha \sin\gamma}{\sin(\alpha+\gamma)}, \end{aligned} \tag{8.30b}$$

and the distances between the target and the two optical centers are derived by

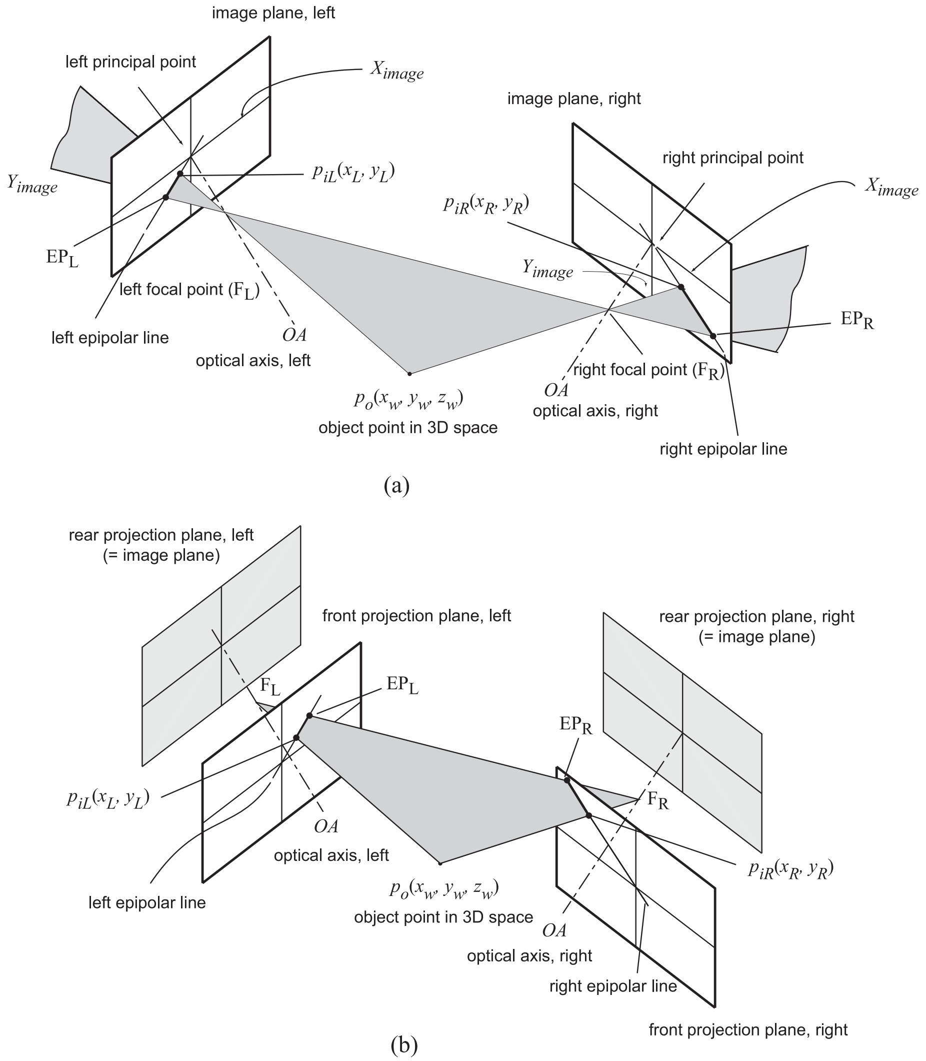
$$\begin{aligned} a &= \frac{b_1}{\cos\gamma} = b\frac{\sin\alpha}{\sin(\alpha+\gamma)}, \\ c &= \frac{b-b_1}{\cos\alpha} = b\frac{\sin\gamma}{\sin(\alpha+\gamma)}. \end{aligned} \tag{8.31}$$

With a known baseline distance b, the input signals α and γ are measured by rotating scanners that sweep the FOV. A device that contains two sensors at C_1 and C_2 is a passive range sensor, while the sensor in an active range sensor measures the two angles in conjunction with a positional control system to move the sensor from C_1 to C_2. Although conventional cameras may be adopted for triangulation, rangefinders based on the geometric property in Eq. (8.31) are better suited for relatively larger distances without any reference to 2D image capture. The more widely used methods of depth measurement from a stereo image pair are described in Sec. 8.6.1.

8.6.1 Epipolar geometry

Figure 8.15 shows a general schematic of a stereoscopic viewing system with converging cameras. Two images of the same target scene are taken by two cameras from two slightly different viewpoints. Because of perspective projection, and depending on the separation between the two optical axes, the object points far from the cameras will appear at almost identical points on the two image planes, while nearer object points will appear at separate locations. (A distance may be considered "far" when compared with the focal lengths of the lenses. For mathematical convenience, the optical parameters of both cameras are assumed to be identical.) This separation between the two sets of image points as a function of the distance (depth) of the object point from the lens focal points is termed *disparity*. Disparity provides the basis for depth recovery of a 3D object point from two 2D images. Stereoscopic depth measurement uses the properties of *epipolar geometry*[1–3] and the concept of *correspondence* between points on the 2D image plane.[18–25]

Since the fundamental assumption of a pinhole camera model is that the incoming rays converge at the focal point (center of projection), the projection lines from $p_o(x_w, y_w, z_w)$ to the left and the right cameras intersect at $\mathrm{F_L}$ and $\mathrm{F_R}$, respectively. After crossing the optical axis at the lens focal point, the 3D projection lines intersect the 2D image planes to create the two image points $p_{iL}(x_L, y_L)$ and $p_{iR}(x_R, y_R)$. For the configurations shown in Figs. 8.15(a) and (b), the 3D lines through the two focal points $\mathrm{F_L}$ and $\mathrm{F_R}$ also intersect the two image planes at *epipolar points* $\mathrm{EP_L}$ and $\mathrm{EP_R}$. The 3D plane defined by the object point and the two focal points is called the *epipolar plane*, and its intersecting lines on the two image planes are *epipolar lines*. For stereoscopic imaging with a pinhole camera model, the epipolar geometric properties may be interpreted as follows:



(a)

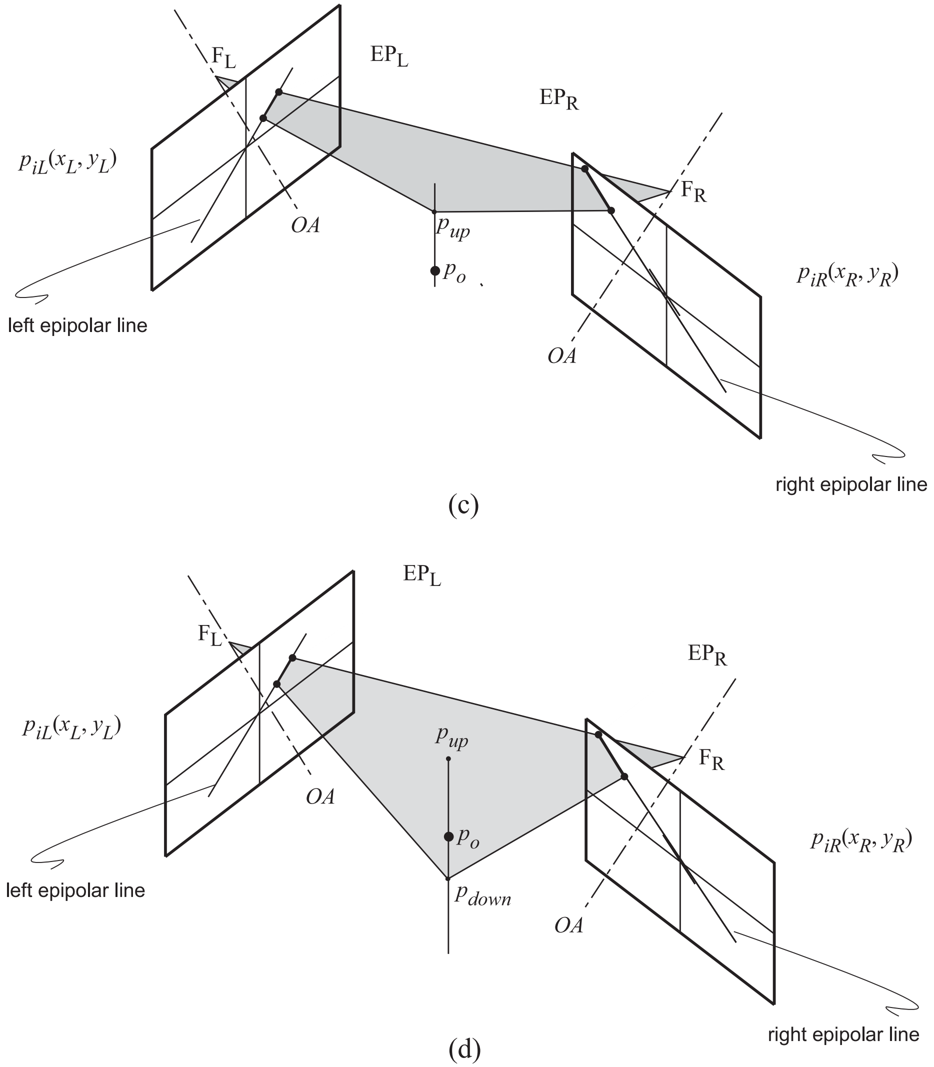
(b)

Figure continued on next page.

Figure 8.15 Converging (or *verging*) camera configuration.[13] (a) An object point captured on two image planes. (b) Front projection of image (a) used for visual interpretation of the epipolar plane (shaded plane). (c) and (d) Front projections showing a pair of optical axes where the epipolar plane hinges about the line through the two centers of projection as the image points move along the y axes of the image planes.

1. For every infinitesimally small point $p_o(x_w, y_w, z_w)$ in the 3D object space, the tuple $[p_o, p_{iL}, p_{iR}]$ defines a unique epipolar plane with its apex at $p_o(\bullet)$.
2. As a consequence of the first property, the left and right epipolar lines are coplanar.

Using these geometric properties, for a given z_w, the epipolar plane hinges with fulcrum line $\mathrm{F_L} - \mathrm{F_R}$ as the object point p_o moves along the Y_w axis [Figs. 8.15(c)



(c)

(d)

Figure 8.15 (*continued*)

and (d)] and stretches horizontally as p_o moves along the X_w axis. Consequently, the epipolar lines are parallel, or each is a projection of itself for all object points with the same depth.

For the general verging camera configuration, an object point $p_o(x_w, y_w, z_w)$ will create left and right image points $p_{iL}(x_L, y_L)$ and $p_{iR}(x_R, y_R)$ with slightly differing coordinate values. Since each object point creates two image points, the pair $\{p_{iL}, p_{iR}\}$ is expected to inherit a set of common image characteristics (*features*) present in the vicinity of the object point p_o. In this context, the *identicalness* between a pair of left and right image points refers to the existence of some of these common features. In formal terms, a point on one image plane is said to *correspond* to a point on the other if they have a common set of predefined features. These two points are referred to as a pair of *corresponding points* or *matching points*. The search for corresponding points is one of the major algorithmic tasks in depth recovery from stereo images. Using the parameters in Fig. 8.16(a),

$$\text{for the left image: } \frac{x_L}{f} = \frac{x_w}{z_w} \text{ and } \frac{y_L}{f} = \frac{y_w}{z_w}; \tag{8.32a}$$

$$\text{and for the right image: } \frac{x_R}{f} = \frac{x_w}{z_w} \text{ and } \frac{y_R}{f} = \frac{y_w}{z_w}, \tag{8.32b}$$

where a and b are the baseline separations between the two optical axes for the pair of images

$$\frac{x_L}{f} = \frac{x_w + \frac{b}{2}}{z_w}, \ \frac{x_R}{f} = \frac{x_w - \frac{b}{2}}{z_w} \tag{8.32c}$$

and

$$\frac{y_L}{f} = \frac{y_w + \frac{a}{2}}{z_w}, \ \frac{y_R}{f} = \frac{y_w - \frac{a}{2}}{z_w} \tag{8.32d}$$

The separations between the two image coordinates $d_x = (x_L - x_R)$ and $d_y = (y_L - y_R)$ are referred to as the *disparity* along the horizontal and vertical axes.

One of the physical properties of any surface that defines a target object is that the contour consists of an infinite number of closely located object points. Consequently, the disparity variations between two neighboring object points making up the contour are small and continuous.[25] This *continuity constraint* imposes the requirement that all 3D object points of interest must be captured by both images and provides a basis to identify the eligible points from a collection of correspondence candidates. It also creates difficulties in cases where the target contains occluding surfaces, as illustrated in Fig. 8.16(b).

The common problem of multiple correspondence candidates for any object point is illustrated in Fig. 8.16(c), and disparity values are listed in Table 8.2. Although it is visually difficult to associate object points and corresponding image points, the continuity of disparity leads to the criterion that two matching neighboring image points will have a minimum disparity change. From the

Table 8.2 Disparity magnitudes of the four points in Fig. 8.16(c).

Object point	**Potential candidates for image points**		**Disparity magnitude**
	Left image	**Right image**	
p_1	x_{L1}	x_{R1}	$\lvert x_{L1} - x_{R1}\rvert$
p_2	x_{L1}	x_{R2}	$\lvert x_{L1} - x_{R2}\rvert$
p_3	x_{L2}	x_{R2}	$\lvert x_{L2} - x_{R2}\rvert$
p_4	x_{L2}	x_{R1}	$\lvert x_{L2} - x_{R1}\rvert$

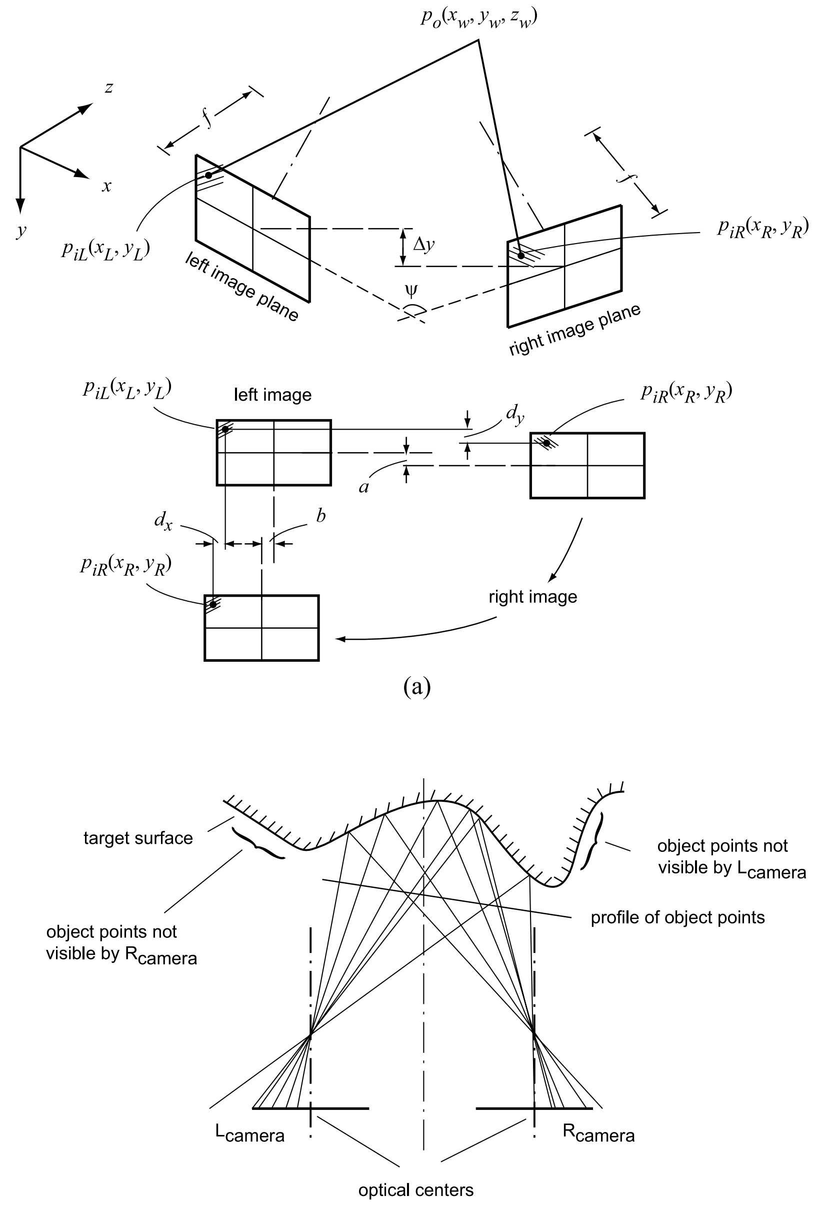
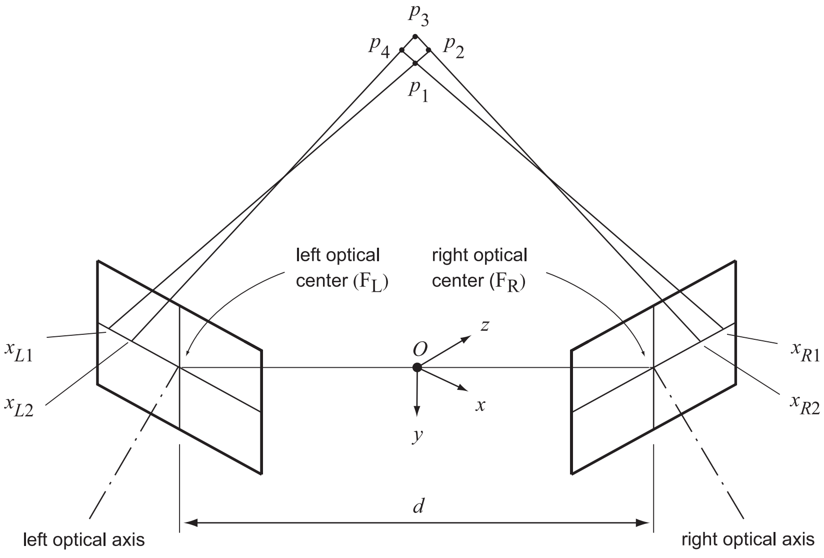


Figure continued on next page.

Figure 8.16 (a) Horizontal and vertical disparity with a verging camera setup. (b) Illustration of discontinuous disparity and occlusion. (c) Ambiguities in correspondence matching.

disparity magnitude values in Table 8.2 and the physical measurements of the points, the locations of the four points are such that points p_2 and p_3 yield the least disparity change and hence qualify as neighboring points in the object space.



(c)

Figure 8.16 (*continued*)

[This difficulty is referred to as the *double-nail illusion* in the human vision literature. A simple experiment consists of placing a very slender object at a 30- to 50-cm reading distance and another very slender object slightly behind the first. When a person's eyes are focused on the objects, these two objects would appear to be on the same level rather than one appearing to be behind the other. See Ref. 26]

For general matching, the feature of each point on the left image point must be compared with the features of all 2D image points on the right image (or vice versa). Consequently, the computational overhead for a general point-to-point feature mapping is high. Because of the difficulties in numerically ascertaining a pixel feature, there is a high degree of uncertainty in any feature-based correspondence. For increased computational efficiency, the epipolar line is used for transforming this feature matching from a 2D search to a 1D (line) search on images captured using the *canonical configuration*. For a 1D feature search, two cameras are mounted in the Euclidean space with their y axes lined up along the same horizontal line (Fig. 8.17).

For *horizontal registration*, $d_y = 0$, and $d_x = (x_L - x_R) = d$ is defined with known values of b and f; the disparity is derived from the coordinates of the object point by

$$x_w = \frac{b(x_L + x_R)}{2d}, \; y_w = \frac{b(y_L + y_R)}{2d}, \text{and } z_w = bf\left(\frac{1}{x_L - x_R}\right) = \frac{bf}{d}. \qquad (8.33)$$

In the verging configuration, the two images must be aligned prior to the correspondence match, which requires translation and rotation using the *essential*

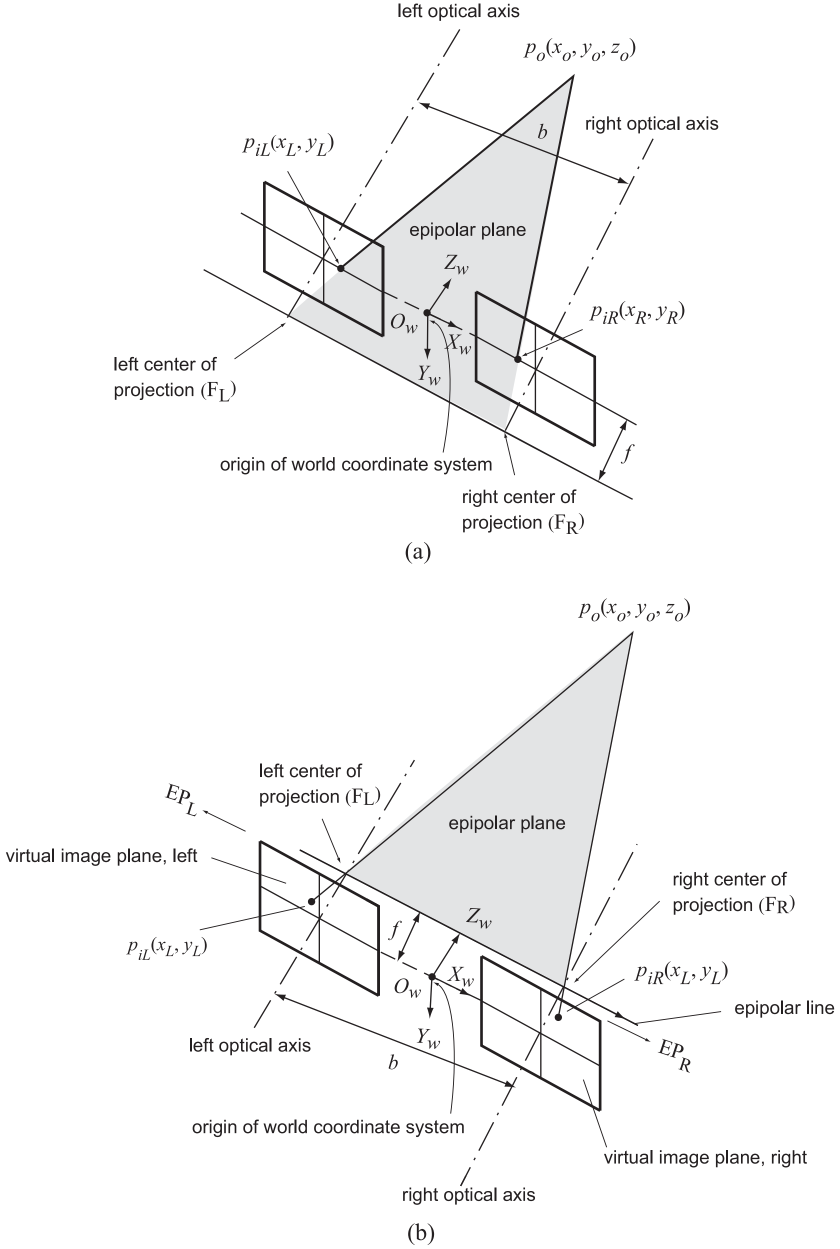


Figure 8.17 Canonical configuration (horizontally registered cameras) for stereo imaging. The object point and two stereo image points with (a) front projection and (b) back projection. The front projection schematic containing virtual image planes is commonly used in the literature for visual convenience to show the epipolar plane.

matrix.[24] The need for image rotation and translation (*composite transformation*) is eliminated by horizontally registering the two cameras.

The primary task in the search for a correspondence match is computation of the disparity value $d = x_L - x_R$ for a given pair of coordinates (x_L, y_L) and (x_R, y_R). Disparity reduces to zero as the depth of the object point becomes infinitely large, making the coordinates of distant object points coincide on the image planes. Also, for a given depth z_w, the disparity may be increased by increasing the focal length or the separation between the two optical axes. Since the focal length choice is somewhat constrained by the relative dimensions of the overall setup, any error in the disparity measurement can be reduced by increasing the baseline distance between the two cameras. However, as the camera baseline separation increases, the similarity between the left and right image points is reduced; in extreme cases, it may not wholly share the same FOV. This increasing dissimilarity in turn makes the task of establishing the identicalness of image points more difficult, requiring a tradeoff between accuracy and the system setup parameters.

Despite its conceptual simplicity, developing a disparity-based depth measurement system requires a considerable attention to details because a perfect match over the entire region of interest is implied. Further, the choice of sensor and lens combination dictates the overall spatial resolution in the computed disparity values. For a pixel resolution of Δ_{pixel} (pixel pitch), the depth measurement resolution Δz_w is given by

$$\Delta z_w = \frac{bf}{d} - \frac{bf}{d \pm \Delta_{pixel}} = \pm bf \frac{\Delta_{pixel}}{d^2} \left(\frac{1}{1 \pm \frac{\Delta_{pixel}}{d}} \right) \simeq \pm bf \frac{\Delta_{pixel}}{d^2}. \tag{8.34}$$

Thus, the resolution of depth derived through the disparity is connected to the parameters of the image-capturing hardware.

With horizontal registration of the two cameras, for a given image point $p_{iL}(x_{iL}, y_{iL})$ and its epipolar line on the left image plane, the search for its corresponding point needs to take place only along the corresponding epipolar line on the right image plane. However, the epiploar plane for an object point p_o is uniquely defined by the locations of the two focal points in the object space, so a precise knowledge of the camera locations is necessary to derive the epipolar line equations. For measurement accuracy and operational simplicity, the two image planes are aligned with the x axis, and their optical axes are made parallel to the Z_w axis of the object (world) coordinate system. This gives the *canonical configuration*, where both focal points have the same Y_w coordinates. Consequently, the line joining the two optical centers becomes parallel to the image planes with two epipoles moving to infinity. For a given depth, the variation in the Y_w coordinate values of the object point then produces parallel epipolar lines, and the variation in the X_w coordinate values of the epipolar lines remains invariant. The advantage of the axial alignment provided by horizontal registration with ideal optics and identical image planes is that an arbitrary left image point (x_{iL}, y_{iL}) has its corresponding point located at coordinates $(x_{iL}, y_{iR} = y_{iL})$ on the right image plane. With predefined feature criteria, the search for the corresponding point for each image point on the left image becomes a process of feature matching with all points located on the same y axis on the right image (and vice versa).

8.6.2 Matching with epipolar constraints

The search for corresponding points on two images is based on the premise that (1) all matching points have the same horizontal location, and (2) intensity profiles in the vicinity of a pair of corresponding points will display a good similarity, subject to errors in the image capture hardware, lens distortions, and a slight difference between the two FOVs. The algorithmic steps for matching along the epipolar lines are described below.

The search for a correspondence match begins with a point of known coordinates on the left image, which serves as the *reference image*. Pixels on the right image are referred to as *candidate pixels* for correspondence matching. The right image then becomes the *target image* or the search field for the corresponding points. With a canonical configuration, the epipolar lines are parallel to the y axes in both image planes. Consequently, each left image point (x_{iL}, y_{iL}) has its corresponding point on the right image with coordinates (x_{iR}, y_{iL}); only x_{iR} is to be determined through feature matching. For any point (x_{iL}, y_{iL}), each point on the corresponding y-axis line $(x_{iR}, y_{iR} = y_{iL})$ in the right image is taken in turn as a candidate pixel for the correspondence match. In the three methods considered here, feature matching refers to similarities between two small neighborhoods (*windows*): one around each pixel in the reference image, and the other around each candidate pixel in the target image. For feature extraction with a given window size (*feature window*), depth measurement can be accomplished in the following discrete steps of (x, y) coordinate values:

1. Capture two images from horizontally registered cameras (canonical configuration). Assign one image as the reference image (the left image below) and the other as the target image (right image).
2. With the top left corner of the reference image as the origin, locate the center of the feature window at $(x_i = x_L, y_j = y_L)$ and derive selected matching reference feature(s).
3. Traverse the feature window along the y_j axis (ordinate) in the right image and compute features at each candidate feature location $x_i,\ i = 1, \ldots, M$ with $y = y_L$, in the right image (image size $M \times M$).
4. Compare the reference feature window in step 2 with M candidate feature windows in step 3. Identify the candidate feature window that gives the closest match with the target feature window and read the abscissa of this matched feature window $(x_{\hat{i}} = x_R)$.
5. Compute the disparity (in units of pixel) as $d_{ij}(x_i,\ y_j) = x_i - x_{\hat{i}} = x_L - x_R,\ y_j = y_L$ for each location of the reference feature window in the left image.
6. Derive the depth $z_{ij}(x_i, y_j) = \frac{bf}{d_{ij}}$ at each (i, j) location of the reference feature window, where $i, j = 1, 2, 3, \ldots, M$.

The results of the above algorithm are presented as either a *depth map* or a *disparity map*. The array of depth values $z_{ij}(x_i, y_j)$ converted to a 2D gray-level image is a depth map. To aid with visual analysis, the depth image is biased and scaled such that the dark areas in the displayed image indicate near and bright areas' distant object points. The disparity, or the positional matching error, in step 5 is converted to a gray-level value (after biasing and scaling) and plotted as a 2D image to create a disparity map. A darker area in the depth map indicates distant object points, and a lighter area indicates closer object points. Although mathematically well defined, feature-based matching generally produces sparse values, necessitating the use of interpolation in disparity and depth plotting. For reference, the disparity map of an experimental system is shown Fig. 8.18.[27]

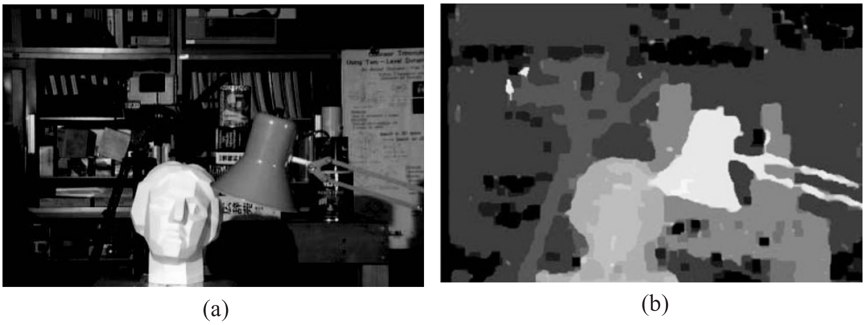

(a) (b)

Figure 8.18 Illustration of a disparity output as an image with (a) a scene image from the left camera and (b) the corresponding disparity map. Closer object points have larger disparity values and therefore a higher intensity. The intensity scale has been normalized in this plot (courtesy of the Pattern Recognition Institute, University of Erlangen-Nuremberg, Germany, © 2002 IEEE).

The 2D plane abscissa and $x_{\hat{i}} = x_i - d_{ij}$ along the ordinate, referred to as the *disparity space*, will ideally have a unity slope for a perfect match. A disparity map with a constant average slope indicates nearly uniform matching within a finite disparity range $\pm d_0$, which gives the uniqueness constraint that only one match is to be expected over $\pm d_0$. Thus, the disparity space provides a measure of continuity in depth calculations. With the epipolar constraint fully met, the choice of reference feature for numerical matching is critical in stereopsis, which is the creation of a physiological sensation of depth from the corresponding images and the position of the eye. In the numerical algorithms, this corresponds to binocular depth measurement to create still stereo images.[28,29]

The disparity and depth maps produced by many widely used correspondence matching algorithms are qualitatively good; however, the derivation of quantitatively reliable depth data for calibration and metrological use requires considerable attention to the physical setup (optics and illumination) and algorithmic refinements to cope with the effects of noise, an uneven finish, and undulation of the target surface.

8.7 Feature Matching

The choice of reference feature(s) as well as matching techniques and several aspects of uniqueness constraints are extensively documented in the literature.[24,25,30–32] In this section, three comparatively easy-to-implement feature-matching techniques with intensity data are described: the difference in intensity values, pattern of intensity distribution, and edge gradients (pattern) in the intensity profile. Since disparity is a local feature, all features are applied to a small neighborhood (window) surrounding each candidate pixel.

Notational conventions: the subscript *i* used in previous sections refers to the image plane as opposed to the framebuffer memory. The index (i, j) used in this section, in line with derivations in other chapters, refers to 2D image coordinates in the host memory map that are available for preprocessing.

8.7.1 Intensity matching[30,33]

If a target scene consists primarily of unrepeated reflectance with the captured images embodying this property in their intensity features and free of noise, one simple way of establishing a feature match is to assume that the intensity separation at any (i, j) location between the two images $e_g(i, j) = g_L(i, j) - g_R(i, j)$ will ideally be zero. The sum of this error $\sum\sum e_g(i, j)$ over the matching $\bar{m} \times \bar{n}$ window ($\bar{m}$ and $\bar{n}$ are odd) may then be assumed to be a small gray-level value, as given by

$$\left.\begin{aligned}\sum\sum e_g(i, j) &= \sum_{m=1}^{\bar{m}}\sum_{n=1}^{\bar{n}}[g_L\{(i - m + (\bar{m} + 1)/2), (j - n + (\bar{n} + 1)/2)\}\\ &\quad - g_R\{(i - m + (\bar{m} + 1)/2), (j - n + (\bar{n} + 1)/2)\}]\\ &\le g_0\end{aligned}\right\}. \tag{8.35}$$

This definition has the limitation that the output is unreliable if the intensity distribution is nearly constant in the two images.

To reduce the effect of local gain variations, this error is normalized by the total intensity value within the right image window $\sum\sum g_R(\bullet\bullet)$, as given by

$$\varepsilon(i, j) = \frac{\sum\limits_{j=1}^{\bar{m}}\sum\limits_{j=1}^{\bar{n}}[g_L\{(i - m + (\bar{m} + 1)/2), (j - n + (\bar{n} + 1)/2)\} - g_R\{(i - m + (\bar{m} + 1)/2), (j - n + (\bar{n} + 1)/2)\}]}{\frac{1}{\bar{m}\bar{n}}\sum\limits_{m=1}^{\bar{m}}\sum\limits_{j=1}^{\bar{n}}[g_R\{(i - m + (\bar{m} + 1)/2), (j - n + (\bar{n} + 1)/2)\}]}. \tag{8.36}$$

Although the result should ideally be zero for a match, a small gray-level threshold $g_0 \ll g_{\max}$ is added to cover uncertainties, where $(0, g_{\max})$ represents the full gray-level scale in the captured image. This addition leads to the following matching criterion:

$$\varepsilon(i, j)\begin{cases}\le g_0 & \text{matched}\\ > g_0 & \text{not matched.}\end{cases} \tag{8.37}$$

In addition to assuming ideal conditions to capture the object scene and the optimum choice of matching window size, this simple area-based method has two limitations: the threshold value may need to be determined after some preliminary assessment for a given pair of images, and multiple matched points may appear for any chosen g_0. One solution is to perform multiple passes of matching with different values of g_0, and to take the first matched point found at each pass as the matching window travels along the right (target) image and mark others as ambiguous points. Some form of statistical analysis on the collection of matched points from all passes may yield a more reliable result. However, with noise-free images, the intensity-matching method provides a more reliable basis to satisfy the disparity-continuity condition and give a *dense* disparity map.

8.7.2 Cross-correlation

An ergodic process refers to a stochastic (random) signal with a constant time-averaged mean. In a wide sense, a time-varying stochastic signal is stationary if its statistics (mean, standard deviation, etc.) do not change with time. Since an image frame is taken as a signal window, an image data frame is treated as a stationary 2D signal. In time-domain signal processing, the cross-correlation between two ergodic processes $f(t)$ and $g(t)$ is defined by[33]

$$R_{fh}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{+T} f(t)h(t+\tau)\,d\tau \equiv \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{+T} h(t)f(t+\tau)\,d\tau = R_{hf}(\tau), \tag{8.38a}$$

where τ is an advance time. If the two signals are given as a collection of finite numbers of sampled points $f(i)$ and $h(i)$, $i = 1, 2, \ldots \bar{m}$, the discrete form of the cross-correlation definition is given by

$$R_{fh}(i) = \frac{1}{\bar{m}} \sum_{m=0}^{\bar{m}-1} f(m)h(i+m) \equiv \frac{1}{\bar{m}} \sum_{m=0}^{\bar{m}-1} h(m)f(i+m) = R_{hf}(i). \tag{8.38b}$$

For the sake of brevity and consistency with the imaging literature, the scaling factor $1/\bar{m}$ is excluded in subsequent derivations. Other index notations are also used, e.g., $R_{fh}(i) = \sum_{m=1}^{M} f(m)h\{m - i + (M+1)/2\}$.

In signal processing, cross-correlation is used for quantifying the degree of similarities between the two signals $f(\bullet)$ and $h(\bullet)$; two extreme values are $R_{fh}(\bullet) = 0$ (for no similarity or *uncorrelated* signals) and $R_{fh}(\bullet) = 1$ (for complete similarity or *correlated* signals). Equation (8.38b) can be extended with the 2D definition of cross-correlation between two discrete signals, where $f(i, j)$ is the target image and $h(i, j)$ is the template (or reference) image, given as

$$R_{fh}(i, j) = \left[\sum_{m=1}^{\bar{m}-1} \sum_{n=1}^{\bar{n}-1} f(m, n)h(i+m, j+n)\right] \equiv \left[\sum_{m=1}^{\bar{m}-1} \sum_{n=1}^{\bar{n}-1} h(m, n)f(i+m, j+n)\right]. \tag{8.38c}$$

This definition forms the basis of image segmentation and object recognition through template matching with the template image chosen to embody the gray-level characteristic features of texture, shape, or size to be identified or detected in the target image.

Although Eq. (8.38c) is based on the analytical concept of signal similarity, its direct use with image data is limited in that the range of $R_{fh}(\bullet\bullet)$ is dependent on the size, shape, orientation, and reflectance of the image feature to be detected, and its range can be relatively large in the presence of a few a bright pixel values. Some of these limitations may be acceptable in certain area-based similarity assessments, but a modified form of cross-correlation is preferable for feature matching. This modification is based on the Euclidean distance measure that subsumes cross-correlation, as defined for any arbitrary location (i, j) of the template center[30] by

$$d^2(i,j) = \sum_{m=0}^{\bar{m}-1}\sum_{n=0}^{\bar{n}-1}[f(i+m, j+n) - h(m,n)]^2 = \bar{f}_{ij}^2 - 2R_{fg}(i,j) + \bar{h}_{ij}^2, \quad (8.39a)$$

where $\bar{f}_{ij}^2$ is the image norm, and $\bar{h}^2$ (a constant value) is the template norm given by

$$\left.\begin{aligned} \bar{h}_{ij}^2 &= \sum_{i=-M}^{i=+M}\sum_{j=-N}^{j=+N}[h(i,j)]^2 \\ \bar{f}_{ij}^2 &= \sum_{i=-M}^{i=+M}\sum_{j=-N}^{j=+N}[f(i+m, j+n)]^2 \end{aligned}\right\}. \quad (8.39b)$$

Since the matrix norm here is the sum of the squares of pixel brightness values, it is also referred to in the image-processing literature as energy.

The direct use of the actual Euclidean distances from Eq. (8.39) is not satisfactory because the image norm is likely to vary across the image, and $|d(i, j)|$ is dependent on the size of the feature template. These issues, coupled with the inevitable presence of noise in the target image, lead to the *normalized cross-correlation* derived below.[20,30]

For a general analysis, the image intensity over any small area with a match may be assumed to have the form

$$f_n(i,j) = a\{h(i+\xi, j+\eta)\} + n(i,j) + b, \quad (8.40)$$

where $n(i, j)$ is the local noise, a is the local gain, and b is the uniform bias (background intensity) in the target image. Thus, any measure of distance, or cross-correlation, will be highly variable across the image and may not always reflect the extent of similarities between the template $h(\bullet\bullet)$ and any image window $f_n(\bullet\bullet)$ or subarea on the target image. One way of overcoming the bias effects is to perform cross-correlation with respect to the arithmetic means, and scale up if necessary, per the following equation:

$$R^{b}_{fh}(i, j) = \sum_{m=0}^{\bar{m}-1}\sum_{n=0}^{\bar{n}-1}\{[f_n(i+m,\ i+n) - \bar{f}_n][h(i,\ j) - \bar{h}]\}, \tag{8.41}$$

where $\bar{f}_n = \sum_{m=0}^{\bar{m}-1}\sum_{n=0}^{\bar{n}-1} f_n(i+m,\ j+n)$ and $\bar{h} = \sum_{m=0}^{\bar{m}-1}\sum_{n=0}^{\bar{n}-1} h(i,\ j)$.

Although the limits of $[h(j,\ k) - \bar{h}]$ are known, $R^{b}_{fh}(i, j)$ may still assume large values due to the wide variations in the local gain a. One way of limiting the range of $R_{fh}(i, j)$ is to normalize Eq. (8.41) with respect to the local energy (norm) as defined by

$$R^{a}_{fh}(i, j) = \frac{\left\{\sum\limits_{m=0}^{\bar{m}-1}\sum\limits_{n=0}^{\bar{n}-1}[f(i+m,\ j+n) - \bar{f}][h(i,\ j) - \bar{h}]\right\}}{\{\bar{f}^2\bar{h}^2\}^{\frac{1}{2}}}, \tag{8.42}$$

where $\bar{f}^2 = \sum_{m=0}^{\bar{m}-1}\sum_{n=0}^{\bar{n}-1}[f(i+m,\ j+n)]^2$ and $\bar{h}^2 = \sum_{m=0}^{\bar{m}-1}\sum_{n=0}^{\bar{n}-1}[h(i,\ j)]^2$.

In some images, normalization with respect to the bias and gain may be required to make a cross-correlation invariant of both the gain and bias. This is achieved by scaling with respect to the local energy variations and the mean of each image window:

$$R^{norm}_{fh}(i, j) = \frac{\left\{\sum\limits_{m=0}^{\bar{m}-1}\sum\limits_{n=0}^{\bar{n}-1}[f(i+m,\ j+n) - \bar{f}][h(i,\ j) - \bar{h}]\right\}}{\{[f(i,\ j) - \bar{f}]^2[h(i,\ j) - \bar{h}]^2\}^{\frac{1}{2}}}, \tag{8.43}$$

where $R^{norm}_{fh}(i, j)$ is referred to as the *normalized cross-correlation coefficient*.[8,30] Although it is numerically simple to generate the values of $R^{norm}_{g_L g_R}(\bullet\bullet)$ with $f(\bullet\bullet) = g_L(\bullet\bullet)$ and $h(\bullet\bullet) = g_R(\bullet\bullet)$, the process has relatively high computing overheads.

A major weakness of Eq. (8.43) is that its maximum value does not have any direct relationship with the disparity continuity constraint. Also, since the disparity is a local feature as in intensity matching, the cross-correlation operation requires small template windows; however, too small a window will generate insufficient pixels for any reliable measurement. Since the two cameras have different viewing angles and hence different projection geometries, spatial scaling as well as some type of transformation of one of the images may be required prior to the correlation computation. Equation (8.43) has been used to extract stereo data,[24] but in the absence of information about the precise orientation of the two cameras plus the need for exhaustive searches for reliable matches, the use of the intensity-based correlation has limited scope in disparity measurement.

8.7.3 Edge feature

In general, a smooth variation in intensity levels in a captured image is an indication of contrast continuity, while a sudden change in intensity (above a preset threshold) between two neighboring pixels is interpreted as a discontinuity or an *edge* (Fig. 8.19). This discontinuity in gray-level values is often used to separate (*segment*) parts of the captured image. Although a complete edge that marks a separation between neighboring objects within an image may be spread across the entire image, part of an edge contour is often a unique feature within a small neighborhood. This condition provides the motivation for correspondence matching using *edge features*.

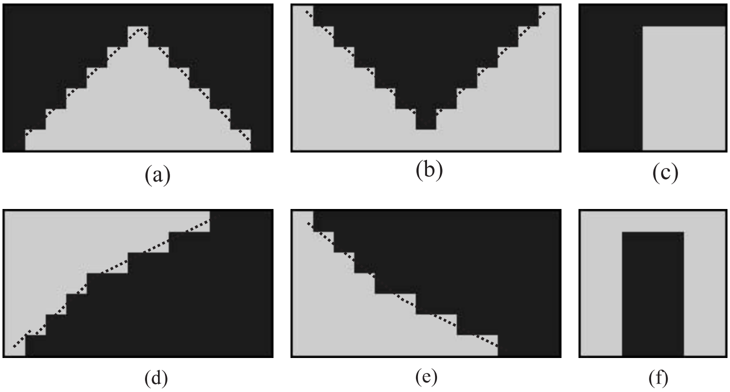


Figure 8.19 Commonly encountered image edges, where the dotted lines mark average edge contours: (a) convex roof, (b) concave roof, (c) step, (d) convex ramp, (e) concave ramp, and (f) bar.[31]

In stereo matching, gradient-based edge detection is commonly used. For a 2D image intensity map $f(x, y)$, this is conveniently derived by the first-order differential operator in

$$\nabla f(x,y) = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}, \tag{8.44a}$$

$$||\nabla f(x,y)|| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}, \tag{8.44b}$$

and

$$\measuredangle \nabla f(x,y) = \tan^{-1}\left(\frac{\partial f(x,y)}{\partial y}\bigg/\frac{\partial f(x,y)}{\partial x}\right). \tag{8.44c}$$

For reference, the derivations of some commonly used edge-detection kernels are given in the appendix at the end of Chapter 11. If captured images are noise-free and illumination is set to exactly replicate the target contrast, the edge strength $||\nabla f(x, y)||$ and edge direction $\measuredangle \nabla f(x, y)$ are adequate for a correspondence match. However, in this first-order gradient operator, the individual partial differentiation terms detect separate gradient values along the x and y axes, and the gradient amplitude is nondirectional.

For reliability in a correspondence match, the locations where gradients change signs are also taken as edge features. Since intensity gradient values are bounded, the second partial derivative of $f(x, y)$ goes through a sign change at the center of an edge, called a *zero-crossing* pixel or point. A zero-crossing point marks the gradient peak in $\nabla f(\bullet\bullet)$ and the change of gradient directions in $\nabla^2 f(\bullet\bullet)$. For numerical work, it is more convenient to detect the zero-crossing points from $\nabla^2 f(\bullet\bullet)$ rather than search from the stored values of the gradient peaks in $\nabla f(\bullet\bullet)$. The numerical values of these zero-crossing points are readily derived by

$$\nabla\{\nabla f(x,y)\} = \frac{\partial}{\partial x}\nabla f(x,y) + \frac{\partial}{\partial y}\nabla f(x,y) = \left[\frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial y^2}\right] f(x,y) = \nabla^2 f(x,y), \quad (8.45)$$

where ∇^2 is the *Laplacian operator*.

In differentiating the spatial intensity distribution, derivative operators generate noise-induced edges due to the inherent noise within the captured image. While the number of such false edges can be reduced by using an appropriate edge threshold, the common practice is to *smooth* the input image prior to differentiation. Smoothing in this case refers to removing high gray-level values that are not consistent within a small neighborhood; thus, a smoothed image has more uniform contrast and reduced edge sharpness. In stereo matching, smoothing is performed by the convolution of each input image with the 2D Gaussian function by

$$\hat{f}(x,y) = f(x,y) \otimes h_G(x,y,\sigma) \triangleq h_G(x,y,\sigma_x,\sigma_y) \otimes f(x,y). \quad (8.46a)$$

Neuropsychological experiments have supported the choice of Gaussian filter for establishing a link between image intensity and its interpretation by the human vision system.[28,29,34]

Due to the symmetrical property of the Gaussian function and for notational convenience, the following simplified form is generally used in subsequent derivations:

$$h_G(x,y,\sigma) = \frac{1}{\sigma_x\sqrt{2\pi}}\frac{1}{\sigma_y\sqrt{2\pi}}\, e^{-\left(\frac{(x-\bar{x})^2}{2\sigma_y^2} + \frac{(y-\bar{y})^2}{2\sigma_y^2}\right)} \equiv \frac{1}{2\pi\sigma^2}\, e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}\Bigg|_{\substack{\text{zero mean: } \bar{x}=\bar{y}=0 \\ \text{equal standard deviation:} \\ \sigma_x=\sigma_y=\sigma.}} \quad (8.46b)$$

Figure 8.20 shows the general shapes of these derivations for an arbitrary value of $\sigma = 6.2225$. In Gaussian filtering, a compromise is achieved between an acceptable level of smoothing effect and excessive blurring (loss of edge sharpness) in the output image $\hat{f}(x, y)$ by assigning suitable values to the standard deviation σ along the two axes. Since the standard deviation represents the spread of the two variates x and y, a larger value of σ implies a smaller height and a larger width in the Gaussian function.[35] Thus, smaller σ values with a narrow $h_G(\bullet)$ introduce a sharper differentiation between neighboring pixels and therefore an accentuated edge lineation in the output image $\hat{f}(\bullet\bullet)$, while larger σ values are more suitable for image smoothing.

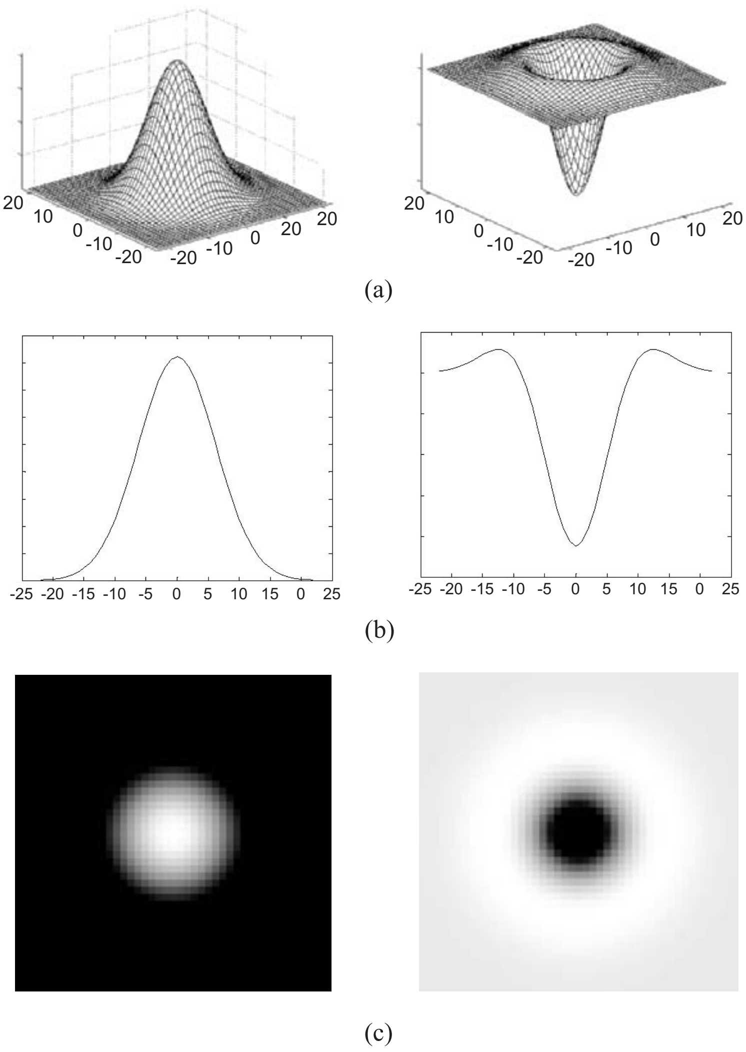


Figure 8.20 (a) General shapes of $h_G(x, y, \sigma)$ on the left and the corresponding Laplacian of the Gaussian (LoG) operator on the right for an arbitrary value of $\sigma = 6.2225$. (b) Corresponding sectional views along the x axis. (c) Corresponding gray-level images.[36]

Since the width of the first negative lobe of $h_G(x, y, \sigma)$ on each side of the origin is $\sqrt{2}\sigma$, the minimum size of the convolution mask (kernel) required to detect two consecutive zero-crossing points in the input image is $w = 2\sqrt{2}\sigma \simeq 3\sigma$. Since the statistical properties of an input image are generally not known, the σ value for edge detection with Gaussian filtering is chosen iteratively. Initially, a large value of σ is chosen to create a wider kernel (window) so that the statistical and edge properties of the subimage within the window can be derived from a large population of pixels (Table 8.3). The size of this window is made smaller iteratively by reducing the value of σ until the desired level of smoothing without

Table 8.3 Standard deviation and probability $[p(x)]$ that a random variate x will fall within the interval $\pm x/\sigma$ and width w of 1D $h_G(x, \sigma)$.[34]

$\frac{x}{\sigma} = \alpha$	$p(x) = \int_{-\alpha}^{+\alpha} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{x^2}{\sigma^2}\right) dx$	$w = 2\sqrt{2}\sigma$	$3w$	**Kernel size for 2D convolution**
0.5	0.383	1.414	4.24	5 × 5
1.0	0.683	2.828	8.48	9 × 9
1.5	0.866	4.243	12.73	13 × 13
2.0	0.954	5.659	16.98	17 × 17
2.5	0.989	7.071	21.21	21 × 21
3.0	0.997	8.485	25.46	25 × 25

losing edge data is attained. Although the spatial spread of the LoG function is infinite, over 99% of the area under the Gaussian curve is contained within the $(\pm x/\sigma, \pm y/\sigma)$ range. For edge detection, the range $3 < \sigma < 4$ and mask size of $3w$ have been found to be satisfactory.[28,33] A list of standard deviation values and the corresponding window sizes are given in Table 8.3.

The analytical form of the filter function presents a significant numerical advantage because the image filtering and derivative operations can be combined by using the associative property of convolution derived below:

$$\begin{aligned}\frac{\partial^2 \hat{f}(x,y)}{\partial x^2} + \frac{\partial^2 \hat{f}(x,y)}{\partial y^2} &= \frac{\partial^2 h_G(x,y,\sigma) \otimes f(x,y)}{\partial x^2} + \frac{\partial^2 h_G(x,y,\sigma) \otimes f(x,y)}{\partial y^2} \\ &= \left\{\frac{\partial^2 h_G(x,y,\sigma)}{\partial x^2} + \frac{\partial^2 h_G(x,y,\sigma)}{\partial y^2}\right\} \otimes f(x,y) \\ &= \left\{\left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right] h_G(x,y,\sigma)\right\} \otimes f(x,y) \\ &= \{[\nabla^2] h_G(x,y,\sigma)\} \otimes f(x,y). \end{aligned} \tag{8.47}$$

Since the location of the zero-crossing (z-c) point is generated by the exponential term in LoG and the signs of the two gradients around it are of primary importance, various scaled versions of the Gaussian function and the corresponding LoG are in common use, including the following example:

$$\tilde{h}_G(x,y,\sigma) = e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \tag{8.48}$$

and

$$[\nabla^2]\tilde{h}_G(x,y,\sigma) = \frac{1}{\sigma^2}\left[\frac{x^2+y^2}{\sigma^2} - 2\right] e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}. \tag{8.49}$$

For numerical convenience, a further simplification is made by replacing the $1/\sigma^2$ factor by a constant gain (typically unity) without any loss of quantitative information about the location of the z-c points or the signs of the gradients.[37]

The various physiological and theoretical aspects of stereo matching that have been studied have produced an extensive amount of literature under the titles of stereopsis, edge detection, and edge matching. Many of the results related to theories of the human vision system and the continuity of disparity are cited as roots for edge-based stereo matching work.[34] The addition of zero-crossing points and gradient directions as edge features led to the formulation of the *Marr–Poggio–Grimson* (MPG) algorithm for correspondence matching.[29,34,36,37] The MPG algorithm's key feature is the generation of a coordinate map of z-c points and direction changes of the intensity gradients at z-c points. These maps are sequentially updated with several passes of reducing window sizes that typically start with 57 × 57 reducing to 13 × 13 or 7 × 7, depending on the edge contents of the two images, along with a suitable threshold to limit false edges. Additional improvements in these plots can be achieved by adding other distinctive neighborhood intensity variations, such as corners, in the feature list.

8.8 Inclined Camera

The general camera model presented in Sec. 8.4 applies to a set of arbitrary coordinate systems. With orthogonal axes, one rotation and two translations are required for angular alignment of the camera and the world coordinate system that provide the basis of the inclined camera configuration shown in Fig. 8.21(a). The primary advantage of this simplified geometry is fewer camera calibration parameters and image–object control points.

This *inclined-camera modeling* requires that the supporting surface of the target object contains the origin of the world coordinates such that the base of the FOV becomes the $X_w Y_w$ plane. With the origin of the image coordinates placed at the center of the image plane, the camera is orientated such that the optical axis $O_i Z_{image}$ points toward the origin of the world coordinates. The obtuse angle of this viewing direction with respect to the azimuth is marked as β in Fig. 8.21(b), with the image and world coordinate axes chosen such that $O_{image} X_{image}$ and $O_w X_w$ are parallel and $O_{image} Y_{image}$ are coplanar with the $Y_w Z_w$ plane.

When the notation in Sec. 8.4 are used with the geometry shown in Fig. 8.21(b), the rotational angles $(\varphi|_{rx_w} = -\beta, \theta|_{ry_w} = 0, \alpha|_{rz_w} = 0)$ align the image and the world coordinate axes, and the translation distances $(t_x = 0,\ -t_y,\ t_z)$ make the two origins coincide. These yield the following simplified 3D-object-space to 2D-image-plane coordinate transformation:

$$\begin{bmatrix} wx_i \\ wy_i \\ w \end{bmatrix} = PR_{x\beta}T \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} x_w \\ (y_w - t_y)\cos\beta + (z_w - t_z)\sin\beta \\ (y_w - t_y)\dfrac{\sin\beta}{f} - (z_w + t_z)\dfrac{\cos\beta}{f} + 1 \end{bmatrix}. \tag{8.50}$$

Using the third elements in Eq. (8.50) as the scaling factor in the homogeneous coordinate system, the coordinates of the projected image point on the inclined

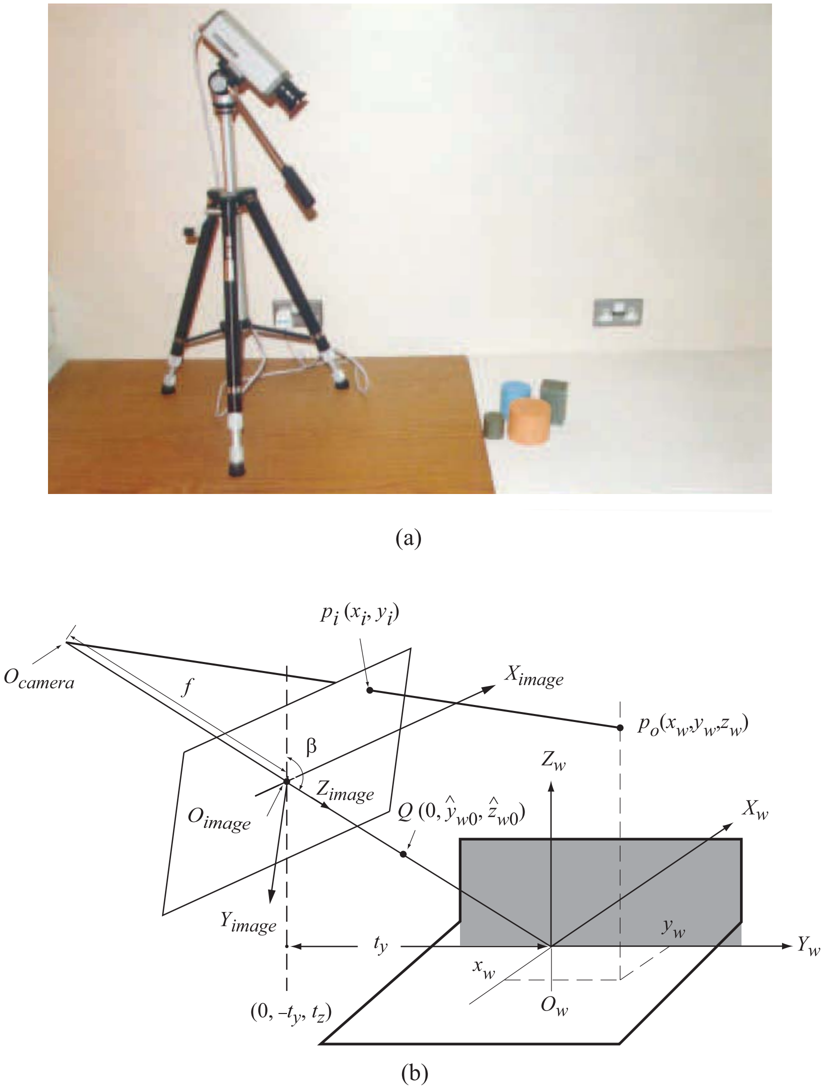


Figure 8.21 (a) Experimental setup of the inclined camera configuration. (b) Relative orientations of the image and the world coordinates.[38]

image plane are derived by[7,39]

$$\left.\begin{aligned} x_i &= \left[\frac{x_w}{(y_w - t_y)\sin\beta - (z_w + t_z)\cos\beta + f}\right] f \\ y_i &= \left[\frac{(y_w - t_y)\cos\beta + (z_w + t_z)\sin\beta}{(y_w - t_y)\sin\beta - (z_w + t_z)\cos\beta + f}\right] f \end{aligned}\right\}. \tag{8.51}$$

Two sets of further modifications are required to include scaling with respect to the pixel resolution of the image sensor, and to relocate the origin of the mathematical geometry from the image plane center to the commonly used image origin at the top left corner of the sensor. With a pixel resolution of (N_x, N_y) and size (L_x, L_y), the scaling factors along the x and y axes are $\hat{k}_x = L_x/N_x$ and $\hat{k}_y = L_y/N_y$. Adding this scaling and biasing to move the origin to the top left location on the image plane yields the following new image coordinates:

$$x_i = \frac{N_x}{2} + \hat{k}_x \left[\frac{x_w}{(y_w - t_y)\sin\beta - (z_w - t_z)\cos\beta + f} \right] f = \frac{N_x}{2} + k_x \left[\frac{x_w}{b y_w - a z_w + d} \right] \tag{8.52a}$$

and

$$y_i = \frac{N_y}{2} + \hat{k}_y \left[\frac{(y_w - t_y)\cos\beta + (z_w + t_z)\sin\beta}{(y_w - t_y)\sin\beta - (z_w + t_z)\cos\beta + f} \right] f = \frac{N_y}{2} + k_y \left[\frac{a y_w + b z_w}{b y_w - a z_w + d} \right], \tag{8.52b}$$

where $k_x = \hat{k}_x f$, $k_y = \hat{k}_x f$, $a = \cos\beta$, and $b = \sin\beta$, with the constraint $t_y \cos\beta = t_z \sin\beta$, giving $d = f - t_z/a$. A characteristic feature of Eqs. (8.52a) and (8.52b) is that for each image point (x_i, y_i), there are many corresponding points in the 3D object space, each with its own z_w coordinate value. The setup requires a precise camera inclination β (taken as an obtuse angle) measurement, but this inclined configuration has one key advantage: the camera model is embedded in only five parameters: k_x, k_y, a, b, and d, with $a^2 + b^2 = 1$. The key stages of a simpler camera calibration procedure are described below.[39,40]

8.8.1 Viewing direction

The viewing direction is assumed to point toward the origin of the world coordinate system, so all points lying on the line segment between O_{image} and O_{world} are projected onto the image center. For one such point $Q(0, \hat{y}_{w0}, \hat{z}_{w0})$ with nonzero $\hat{z}_w$, the camera inclination angle is $\beta = \tan^{-1}(\hat{y}_{w0}/\hat{z}_{w0})$. Then the values of a and b can be derived with manually measured values of $\hat{y}_{w0}$ and $\hat{z}_{w0}$. The accuracy of these two parameters improves as the object moves closer to the camera along the viewing line, giving larger values of $\hat{y}_{w0}$ and $\hat{z}_{w0}$.

8.8.2 Scaling factors

Because of the oblique viewing angle of the camera, any pair of symmetrical points along the X_w axis of the $X_w Y_w$ plane ($z_w = 0$) in the 3D object space will create two asymmetric points on the image plane. In the reverse case, two symmetric points on the image plane will correspond to a pair of asymmetric points in the object space. Using the geometry in Fig. 8.22, for $0 < \hat{y}_{i1} < N_y/2$ and $\hat{z}_w = 0$, the y-coordinate values of two points $(N_x/2, N_y/2 - \bar{y}_{i1})$ and $(N_x/2, N_y/2 + \bar{y}_{i1})$ on the image plane will generate two points $\hat{y}_{w1}$ and $\hat{y}_{w2}$ in the object space and satisfy the following

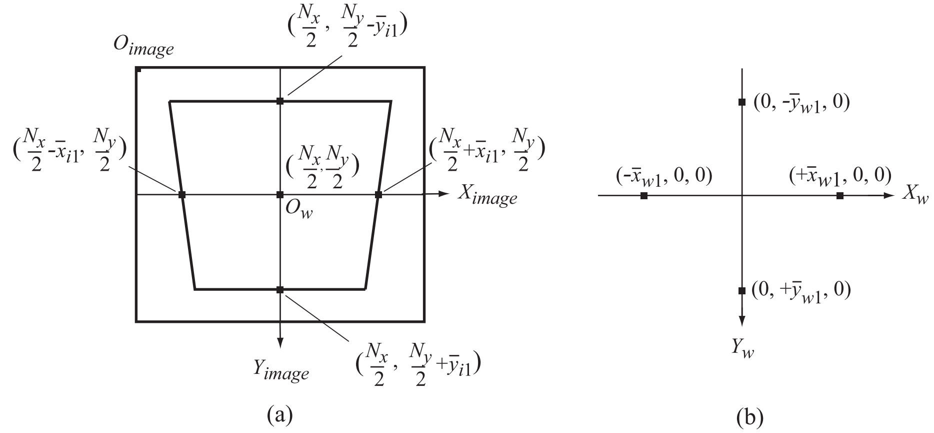

$(\frac{N_x}{2}, \frac{N_y}{2}-\bar{y}_{i1})$
O_{image}
$(\frac{N_x}{2}-\bar{x}_{i1}, \frac{N_y}{2})$
$(\frac{N_x}{2}, \frac{N_y}{2})$
$(\frac{N_x}{2}+\bar{x}_{i1}, \frac{N_y}{2})$
X_{image}
O_w
Y_{image}
$(\frac{N_x}{2}, \frac{N_y}{2}+\bar{y}_{i1})$
(a)
$(0, -\bar{y}_{w1}, 0)$
$(-\bar{x}_{w1}, 0, 0)$
$(+\bar{x}_{w1}, 0, 0)$
X_w
$(0, +\bar{y}_{w1}, 0)$
Y_w
(b)

Figure 8.22 Pair of points on the image plane and their corresponding points on the X_wY_w plane in the object space. The area on the X_wY_w plane of the object surface as viewed by the camera has a trapezium shape on the image plane due to the oblique camera angle.

conditions:

$$\frac{N_y}{2} - \bar{y}_{i1} = \frac{N_y}{2} + k_y \frac{a\bar{y}_{w1}}{b\bar{y}_{w1} + d} \tag{8.53a}$$

and

$$\frac{N_y}{2} + \bar{y}_{i1} = \frac{N_y}{2} + k_y \frac{a\bar{y}_{w2}}{b\bar{y}_{w2} + d}. \tag{8.53b}$$

The division of the two equations and a rearrangement of terms yield $d = -2b$ $(\bar{y}_{w1}\bar{y}_{w2}/\bar{y}_{w1} + \bar{y}_{w2})$. By substituting this value into Eq. (8.52a), the Y_{image} axis-scaling factor is derived as

$$k_y = -(b\bar{y}_{w1} + d)\frac{\bar{y}_{i1}}{a\bar{y}_{w1}} = \frac{d}{a}\left(\frac{(\bar{y}_{w2} - \bar{y}_{w1})}{2\bar{y}_{w1}\bar{y}_{w2}}\right)\bar{y}_{i1}. \tag{8.53c}$$

To derive the scaling factor along the X_{image} axis, two points $(N_x/2 - \bar{x}_{i1}, N_y/2)$ and $(N_x/2 + \bar{x}_{i1}, N_y/2)$ are chosen with the condition $0 < \bar{x}_{i1} < \frac{N_x}{2}$. From Eq. (8.52a), the corresponding points in the object space are given by

$$\frac{N_x}{2} - \bar{x}_{i1} = \frac{N_x}{2} - k_x \frac{\bar{x}_{w1}}{d} \tag{8.54a}$$

and

$$\frac{N_x}{2} + \bar{x}_{i1} = \frac{N_x}{2} + k_x \frac{\bar{x}_{w1}}{d}. \tag{8.54b}$$

The combination of these two equations gives

$$k_x = d\frac{\bar{x}_{i1}}{\bar{x}_{w1}}. \tag{8.54c}$$

The final result of the calibration process is the pair of algebraic equations below that relate the image point coordinates $p_i(x_i, y_i)$ corresponding to an arbitrary object point $p_o(x_w, y_w, z_w)$:

$$x_i = \frac{N_x}{2} + \left(d\frac{\bar{x}_{i1}}{\bar{x}_{w1}}\right)\left(\frac{x_w}{by_w - az_w + d}\right) \tag{8.55a}$$

and

$$y_i = \frac{N_y}{2} + \left(\frac{d(\bar{y}_{w2} - \bar{y}_{w1})\bar{y}_{i1}}{2a\bar{y}_{w1}\bar{y}_{w2}}\right)\left(\frac{ay_w + bz_w}{by_w - az_w + d}\right), \tag{8.55b}$$

where $\{(-\bar{x}_{w1}, 0, 0), (\bar{x}_{w1}, 0, 0)\}$ and $\{(0, -\bar{y}_{w1}, 0), (0, \bar{y}_{w2}, 0)\}$ are four control points on the X_wY_w plane (base plane) of the object space with the corresponding image points marked in Fig. 8.22. Despite being relatively simple and having no provision to account for lens distortions, Eq. (8.55) has been observed to provide errors of 0.25% or better for dimensional measurements of solid objects.[40] An alternate method of deriving 3D object shapes is to use a reference object of known geometry mounted on the target object. In this method, a curve-fitting algorithm is used to derive the relative orientations and locations of a collection of 3D points on the target object with respect to the reference plane shape, making it a calibration routine.[41,42]

The properties of epipolar geometry have been used extensively to formulate analytical conditions for correspondence. These conditions are based on the alignment of two image frames through rotation and translation to relate the two views shown in Fig. 8.23, leading to the following transformation equation from one image to the other:

$$p_{eR} = Rp_{eL} + T. \tag{8.56a}$$

Since p_{eL} and $T \times p_{eR}$ are orthogonal, the vector product yields the null matrix

$$\begin{aligned} 0 &= p_{eL}^T T \times p_{eR} = p_{eL}^T T \times (Rp_{eL} + T) \\ &= p_{eL}^T T \times Rp_{eL} = p_{eL}^T \varepsilon p_{eL}. \end{aligned} \tag{8.56b}$$

The composite transformation matrix $\varepsilon = T \times R = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} R$ is known as the *essential matrix*. If the image points and the corresponding object-to-image lines in the 3D Euclidean space are related by the transformations M_{eL} and M_{eR} for the two

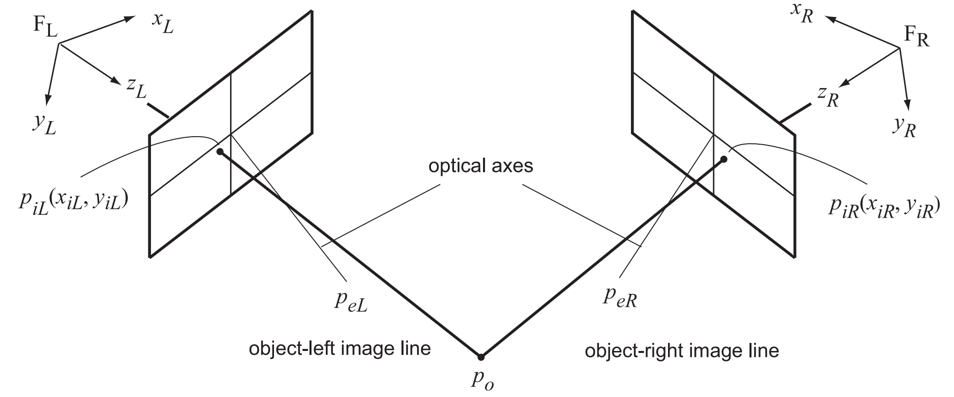


Figure 8.23 Coordinate systems of the two image frames and transformation operations for alignment.

image planes, then the transformation matrix given by $F = M_{eL}^T \varepsilon M_{eR}^T$ is defined as the *fundamental matrix*. Since $M_{e\bullet}$ contains the intrinsic parameters of the left and right cameras (Sec. 8.6), and $p_{e\bullet}$ are in pixel coordinates, the fundamental matrix links the epipolar constraints with the extrinsic parameters of the stereo setup. The geometric theories of correspondence matching are given elsewhere.[24,43,44]

References

1. J. A. Todd, *Projective and Analytical Geometry*, Pitman Publishing, London (1965).
2. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, MA (1987).
3. R. Hartshorne, *Foundations of Projective Geometries*, W.A. Benjamin, New York (1967).
4. L. G. Roberts, "Machine perception of three-dimensional solids," in *Optical and Electro-Optical Information Engineering*, J. D. Tippett, et al. Eds., MIT Press, Cambridge, MA (1965).
5. R. Rosenfeld, "Homogeneous coordinates and perspective planes in computer graphics," in *Computer Graphics and Applications,* Vol. 1, IEEE Press, Piscataway, NJ, pp. 50–55 (1981).
6. R. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley & Sons, New York (1989).
7. K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision and Intelligence*, McGraw-Hill, New York (1986).
8. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA (2000).
9. P. J. McKerrow, *Introduction to Robotics*, Addison-Wesley, Sydney, Australia (1991).

10. R. Y. Wong, "Sensor transformations," *Trans. IEEE Systems, Man and Cybernetics* **SMC7**(12), 836–841 (1977).

11. E. L. Hall, "Measuring curved surfaces for robot vision," *Trans. IEEE on Computers* **C15**(12), 42–54 (1982).

12. C. C. Slama, *Manual of Photogrammetry*, American Society of Photogrammetry and Remote Sensing, Falls Church, VA (1980).

13. W. Faig, "Calibration of close-range photogrammetry systems: mathematical formulation," *Photogrammetric Engineering and Remote Sensing* **41**(12), 1479–1486 (1975).

14. R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV camera and lenses," *IEEE J. of Robotics and Automation* **RA3**(4), 322–344 (1987).

15. Z. Zhang, "A flexible new technique for camera calibration," *Trans. IEEE on Pattern Analysis and Machine Intelligence* **PAMI22**(11), 1330–1334 (2000).

16. K. J. Gåsvik, *Optical Metrology*, John Wiley & Sons, Chichester, UK (1995).

17. R. Wilson, "Modeling and Calibration of Automated Zoom Lenses," Ph.D. thesis, Carnegie Mellon University, Pittsburgh (1993).

18. A. C. Kak, "Depth perception in robotics," in *Handbook of Industrial Robotics*, S. Y. Nof, Ed., John Wiley & Sons, Chichester, UK, pp. 272–319 (1985).

19. F. J. Pipitone and T. G. Marshall, "A wide-field scanning triangulation rangefinder for machine vision," *International J. of Robotics Research* **2**(1), 349–390 (1983).

20. D. Panton, "A flexible approach to digital stereo mapping," *Photogrammetric Engineering and Remote Sensing* **44**(12), 1499–1512 (1978).

21. R. Henderson, R. Miller, and C. Grosch, "Automatic stereo reconstruction of man-made targets: Digital processing of aerial images," *Proc. SPIE* **186**(8), 240–248 (1979).

22. S. Barnard and W. Thompson, "Disparity analysis of images," *Trans IEEE Pattern Analysis and Machine Intelligence* **PAMI4**, 333–340 (1980).

23. Y. C. Kim and J. K. Aggarwal, "Positioning three-dimensional objects using stereo images," *IEEE J. of Robotics and Automation* **RA3**(4), 361–373 (1987).

24. O. Faugeras, *Three Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, MA (1993).

25. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision,* Vol. 1, Addison-Wesley, Reading, MA (1992).

26. J. D. Kroll and W. A. van de Grind, "The double-nail illusion: experiments on binocular vision with nails, needles, and pins," *Perception* **9**(6), 651–669 (1980).

27. J. Schmidt, H. Nieman, and S. Vogt, "Dense disparity maps in real-time with an application to augmented reality," in *Proc. 6th IEEE Workshop on Applications of Computer Vision*, Orlando, FL, pp. 225–230 (2002).

28. D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Royal Society of London* **B207**, 187–217 (1980).

29. V. Torr and T. A. Poggio, "The theory of edge detection," *Trans. IEEE Pattern Analysis and Machine Intelligence* **PAMI8**(2), 147–163 (1986).

30. F. van der Heijden, *Image-based Measurement Systems*, John Wiley & Sons, Chichester, UK (1995).

31. W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, New York (1991).

32. K. R. Castleman, *Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ (1996).

33. A. Papapoulos, *Proability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York (1965).

34. D. Marr and T. A. Poggio, "A theory of human stereo vision," *Proc. Royal Society of London* **B204**, 301–328 (1979).

35. J. N. Kapur and H. C. Saxena, *Mathematical Statistics*, S. Chand, New Delhi (1963).

36. H. Schulteis, "Two-dimensional positioning of objects in space using stereo imaging," Internal Report, Department of Engineering, University of Reading, UK, March 1996.

37. W. E. L. Grimson, "Computational experiments with a feature based stereo algorithm," *Trans. IEEE Pattern Analysis and Machine Intelligence* **PAMI7**(1), 17–34 (1985).

38. Q. H. Hong, "3D Feature Extraction from a Single 2D Image," PhD thesis, Department of Engineering, University of Reading, UK, June 1991.

39. P. K. Sinha and Q. H. Hong, "Recognition of an upright cylinder from a perspective view using Hough transform technique," in *Proc. CG 6th International Conference on Image Processing Analysis*, Como, Italy, pp. 168–172 (1991).

40. P. K. Sinha, and Q. H. Hong, "A Hough transform technique to detect vertical lines in 3D space," in *Proc. IEEE International Conference on Image Processing*, Maastricht, Belgium, pp. 441–444 (1992).

41. B. E. Platin, Z. Gan, and N. Oglac, "3D object configuration sensor utilizing single camera," presented at ASME Winter Annual Meeting, Dallas, TX, November 1990.

42. E. U. Acar, "Experimental Investigation and Implementation of a 3D Configuration Reconstruction Algorithm for an Object Using a Single Camera Image," M.Sc. dissertation, Department of Mechanical Engineering, Middle-East Technical University, Ankara (1995).

43. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK (2004).

44. A. Gruen and T. S. Huang, Eds., *Calibration and Orientations of Cameras in Computer Vision*, Springer, New York (2001).