Chapter 12

Calibration

This chapter will present methods for calibrating cameras and depth measurement systems such as binocular stereo and range cameras. The machine vision algorithms developed in previous chapters involved the extraction of measurements within the image plane. Measurements, such as the location of an edge or the centroid of a region, were made in the coordinate system of the image array. We assumed that image plane coordinates could be determined from image locations using simple formulas based on the premise that the location of the central axis of projection in the image plane (the principal point) was at the center of the image. We also assumed that the distance of the image plane from the center of projection (the camera constant) was known. This chapter will cover the calibration problems associated with these assumptions. Measurements based on the simple image geometry assumed in previous discussions can be corrected to true image plane coordinates using the solution to the interior orientation problem, covered in Section 12.9. The position and orientation of a camera in the scene can be determined by solving the exterior orientation problem, covered in Section 12.8. The solution to the interior and exterior orientation problems provides an overall solution to the camera calibration problem that relates the location of pixels in the image array to points in the scene. Camera calibration is covered in Section 12.10. The relative orientation problem for calibrating the relationship between two cameras for binocular stereo is covered in Section 12.4, the method for computing depth measurements from binocular stereo disparities is explained in Section 12.6, and the absolute orientation problem for converting depth measurements in viewer-centered coordinates to an absolute coordinate system for the scene is covered in Section 12.3. Several different coordinate systems encountered in camera calibration are described in Section 12.1, and the mathematics of rigid body transformations is covered in Section 12.2.

The mathematics behind solutions to the problems of calibrating cameras and range sensors has been developed in the field of photogrammetry. Photogrammetry provides a collection of methods for determining the position and orientation of cameras and range sensors in the scene and relating camera positions and range measurements to scene coordinates. There are four calibration problems in photogrammetry:

Absolute orientation determines the transformation between two coordinate systems or the position and orientation of a range sensor in an absolute coordinate system from the coordinates of calibration points.

Relative orientation determines the relative position and orientation between two cameras from projections of calibration points in the scene.

Exterior orientation determines the position and orientation of a camera in an absolute coordinate system from the projections of calibration points in the scene.

Interior orientation determines the internal geometry of a camera, including the camera constant, the location of the principal point, and corrections for lens distortions.

These calibration problems are the classic problems in photogrammetry and originated with the techniques used to create topographic maps from aerial images. In addition to the four basic calibration problems, photogrammetry also deals with the problem of determining the position of points in the scene from binocular stereo disparities and provides methods for resampling stereo images so that the epipolar lines correspond to image rows.

All of the photogrammetric problems for determining the transformation between coordinate systems assume that a set of conjugate pairs is available. The conjugate pairs are obtained by matching feature points between views. These matches must be correct since the classic methods of photogrammetry use least-squares criteria and are very sensitive to outliers due to mismatched features. In fact, calibration algorithms can be ill conditioned, and one should not tempt fate by adding outliers to the normally distributed errors. Section 12.13 includes some discussion on using robust regression for handling mismatched feature points. In some applications, it is necessary to match two sets of points that are not conjugate pairs or to match two curves or surfaces that do not have discrete features. This is the registration problem, discussed in Section 13.9.

12.1 Coordinate Systems

The image processing operations in machine vision are usually done in the coordinate system of the image array, with the origin at the upper left pixel. The rows and columns correspond to integer coordinates for the pixel grid. Subpixel measurements add a fractional part to the image array (pixel) coordinate system, leading to pixel coordinates being represented as floating-point numbers; however, subpixel resolution does not change the geometry of the image array coordinate system. We can convert from pixel coordinates to image plane coordinates using some assumptions about the camera geometry. In Section 12.9, we will show how the camera parameters can be calibrated so that the mapping from image array (pixel) coordinates to image plane coordinates uses the actual geometry of the camera, including accounting for the effects of lens distortions.

The approximate transformation from pixel coordinates to image coordinates assumes that the principal axis intersects the image plane in the center of the image array. If the image array has n rows and m columns, then the center of the image array is

$$\hat{c_x} = \frac{m-1}{2} \tag{12.1}$$

$$\hat{c_y} = \frac{n-1}{2}. \tag{12.2}$$

We use the hat notation to stress that these are estimates for the location of the principal point. The x axis is in the direction of increasing column index, but the direction of increasing row index and the y axis point in opposite directions. The transformation from pixel coordinates $[i, j]$ to image coordinates (x', y') is

$$x' = j - \frac{m-1}{2} \tag{12.3}$$

$$y' = -\left(i - \frac{n-1}{2}\right). \tag{12.4}$$

This transformation assumes that the spacing between rows and columns in the image array is the same and that image plane coordinates should be expressed in this system of units. Let the spacing between columns be s_x and the spacing between rows be s_y. We can add these conversion factors to the transformation from pixel to image coordinates:

$$x' = s_x\left(j - \frac{m-1}{2}\right) \tag{12.5}$$

$$y' = -s_y\left(i - \frac{n-1}{2}\right). \tag{12.6}$$

If the image sensor has square pixels, then the conversion factors are identical and can be omitted. This simplifies the image processing algorithms. The image measurements (such as distances) can be converted to real units later, if necessary. If the image sensor has nonsquare pixels, then $s_x \neq s_y$ and it may be necessary to convert pixel coordinates to image coordinates, using the formulas above, before performing measurements. However, some measurements are not affected by nonsquare pixels. For example, the centroid can be computed using pixel coordinates and converted to image plane coordinates later, but distances and angles are not invariant to unequal scale factors and should be computed from point locations in image coordinates. For example, the centroids of two regions must be converted to image plane coordinates before calculating the distance between the centroids. Because of these problems, it is very common to require cameras with square pixels in machine vision applications. If an application uses a camera with nonsquare pixels, then you must carefully consider how the measurements will be affected by nonsquare pixels. When in doubt, convert to image plane coordinates before making any measurements.

The image plane coordinate system is part of the camera coordinate system, a viewer-centered coordinate system located at the center of projection with x and y axes parallel to the x' and y' axes in the image plane and a z axis for depth. The camera coordinate system is positioned and oriented relative to the coordinate system for the scene, and this relationship is determined through camera calibration, discussed in Section 12.10.

In summary, there are several coordinate systems in camera calibration:

Scene coordinates for points in the scene

Camera coordinates for the viewer-centered representation of scene points

Image coordinates for scene points projected onto the image plane

Pixel coordinates for the grid of image samples in the image array

Image coordinates can be true image coordinates, corrected for camera errors such as lens distortions, or uncorrected image coordinates. Pixel coordinates are also called image array coordinates or grid coordinates.

There may be multiple cameras in the scene, each with its own coordinate system. For example, in binocular stereo there is the left camera coordinate system, the right camera coordinate system, and the stereo coordinate system in which stereo depth measurements are represented. Determining the relationships between these coordinate systems is the purpose behind the various calibration problems discussed in this chapter.

12.2 Rigid Body Transformations

Any change in the position or orientation of an object is a rigid body transformation, since the object moves (changes position or orientation) but does not change size or shape. Suppose that a point $\mathbf{p}$ is visible from two viewpoints. The position of point $\mathbf{p}$ in the coordinate system of the first viewpoint is

$$\mathbf{p}_1 = (x_1, y_1, z_1)^T \tag{12.7}$$

and the position of point $\mathbf{p}$ in the coordinate system of the second viewpoint is

$$\mathbf{p}_2 = (x_2, y_2, z_2)^T. \tag{12.8}$$

The transformation between the two camera positions is rigid body motion, so each point at position $\mathbf{p}_1$ in the first view is transformed to its coordinates in the second view by rotation and translation:

$$\mathbf{p}_2 = R\,\mathbf{p}_1 + \mathbf{p}_0, \tag{12.9}$$

where matrix R is a 3×3 orthonormal matrix for rotation,

$$R = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix}, \tag{12.10}$$

and vector $\mathbf{p}_0$ is the vector for the amount and direction of translation. Point $\mathbf{p}_0$ is the position of the origin of coordinate system one in coordinate system two.

Equation 12.9 can be viewed as a formula for computing the new coordinates of a point that has been rotated and translated or as a formula for computing the coordinates of the same point in space in different coordinate systems. The first interpretation is used in rigid body mechanics: the new coordinates of a point on an object must be computed after the object has been moved to a new position and orientation. The second interpretation is used for calibration problems: the same point has different coordinates when seen from different viewing positions and orientations. The change in coordinates is determined by the rigid body transformation between the two viewpoints, and the calibration problem is to determine the transformation from a set of calibration points (conjugate pairs). For example, consider the same point seen by two identical range cameras at different positions and orientations in space. Since the viewpoints are different, the coordinates are different even though the coordinates represent the same point. Imagine that the first range camera is rotated so that it has the same orientation as the second range camera. Now the coordinate systems of the range cameras have the same orientation but different positions in space. Now imagine that the first range camera is translated to the same position in space as the second range camera. Now the point has the same coordinates in both cameras. This process of aligning the coordinate systems of the two cameras so that identical points have the same coordinates is modeled by the rigid body transformation in Equation 12.9, which says that a point $\mathbf{p}_1$ in the viewer-centered coordinate system of the first camera is first rotated and then translated to change its coordinates to point $\mathbf{p}_2$ in the second camera. This process is illustrated in Figure 12.1. In practice, there may be one or more range cameras, or depth measuring systems such as binocular stereo, that generate point coordinates in their own camera-centered coordinate system, and these

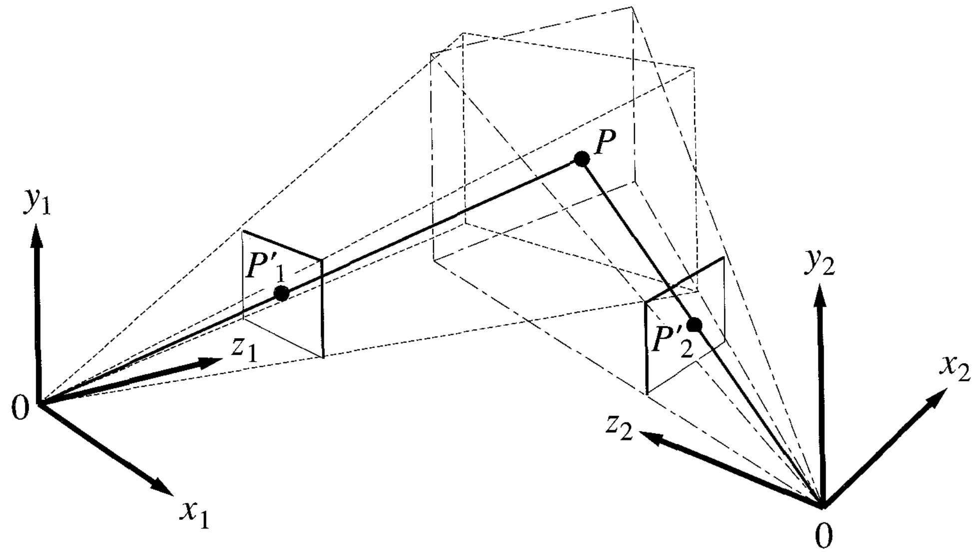

Figure 12.1: Point P is visible from both viewpoints but has different coordinates in each system. The drawing illustrates the rigid body transformation: the coordinate system on the left is first rotated and then translated until it is aligned with the coordinate system on the right.

measurements must be transformed into a common coordinate system that has been predefined for the scene. This common coordinate system is called the world or absolute coordinate system, sometimes called scene coordinates. The term ***absolute coordinates*** is used in this book to make it clear that we are talking about a global coordinate system for points in the scene, rather than the viewer-centered coordinate system for a particular camera or depth measuring system. The terms *viewer-centered* and *camera-centered coordinates* are synonymous. When we speak about camera coordinates, we are referring to the three-dimensional coordinate system for a camera, that is, the coordinate system for a particular viewpoint. As explained in Section 1.4.1, we generally use a left-handed coordinate system with the x and y axes corresponding to coordinates in the image plane in the usual way and the z axis pointing out into the scene. A point in the scene with coordinates

(x, y, z) in a camera-centered coordinate system is projected to a point (x', y') in the image plane through perspective projection:

$$x' = \frac{xf}{z} \tag{12.11}$$

$$y' = \frac{yf}{z}. \tag{12.12}$$

The coordinates (x', y') of the projected point are called image plane coordinates, or just image coordinates, for short. The origin of the image plane coordinate system is located at the intersection of the image plane and the z axis of the camera coordinate system. The z axis is called the principal axis or optical axis for the camera system, and the origin of the image plane coordinate system is called the principal point.

An affine transformation is an arbitrary linear transformation plus a translation. In other words, the vector of point coordinates is multiplied by an arbitrary matrix and a translation vector is added to the result. Affine transformations include changes in position and orientation (rigid body transformations) as a special case, as well as other transformations such as scaling, which changes the size of the object. A transformation that changes the shape of an object in some general way is a nonlinear transformation, also called warping, morphing, or deformation. There are several ways to represent rotation, including Euler angles and quaternions, covered in the following sections.

12.2.1 Rotation Matrices

Angular orientation can be specified by three angles: rotation ω about the x axis, rotation ϕ about the new y axis, and rotation κ about the new z axis. Angle ω is the pitch (vertical angle) of the optical axis, angle ϕ is the yaw (horizontal angle) of the optical axis, and angle κ is the roll or twist about the optical axis. These angles are called the Euler angles. No rotation (zero values for all three angles) means that two coordinate systems are perfectly aligned. Positive ω raises the optical axis above the x–z plane in the direction of positive y, positive ϕ turns the optical axis to the left of the y–z plane in the direction of negative x, and positive κ twists the coordinate system clockwise about the optical axis as seen from the origin. The entries of the rotation matrix R defined in Equation 12.10 in terms of these angles are

$$\begin{aligned}
r_{xx} &= \cos\phi\cos\kappa \\
r_{xy} &= \sin\omega\sin\phi\cos\kappa + \cos\omega\sin\kappa \\
r_{xz} &= -\cos\omega\sin\phi\cos\kappa + \sin\omega\sin\kappa \\
r_{yx} &= -\cos\phi\sin\kappa \\
r_{yy} &= -\sin\omega\sin\phi\sin\kappa + \cos\omega\cos\kappa \\
r_{yz} &= \cos\omega\sin\phi\sin\kappa + \sin\omega\cos\kappa \\
r_{zx} &= \sin\phi \\
r_{zy} &= -\sin\omega\cos\phi \\
r_{zz} &= \cos\omega\cos\phi.
\end{aligned} \tag{12.13}$$

Although this is a common representation for rotation, determining the rotation by solving for the Euler angles leads to algorithms that are not numerically well conditioned since small changes in the Euler angles may correspond to large changes in rotation. Calibration algorithms either solve for the entries of the rotation matrix or use other representations for the rotation angles such as quaternions.

The rotation matrix is an orthonormal matrix,

$$R^T R = I, \tag{12.14}$$

where I is the identity matrix. This means that the matrix inverse is just the transpose of the rotation matrix. A calibration algorithm will produce a rigid body transformation between coordinate systems in one direction; for example, from coordinate system 1 to coordinate system 2,

$$\mathbf{p}_2 = R\mathbf{p}_1 + \mathbf{p}_{2,0}. \tag{12.15}$$

The inverse rigid body transform that converts coordinates in system 2 to coordinates in system 1 is

$$\mathbf{p}_1 = R^T(\mathbf{p}_2 - \mathbf{p}_{2,0}) = R^T\mathbf{p}_2 + \mathbf{p}_{1,0}, \tag{12.16}$$

where the notation $\mathbf{p}_{i,0}$ means the point in coordinate system i that is the origin of the other coordinate system. Note that the inverse translation is not just $-\mathbf{p}_{2,0}$ but must be multiplied by the inverse rotation matrix, because the translation $\mathbf{p}_{2,0}$ is in coordinate system 2 and the inverse translation must be expressed in the same orientation as coordinate system 1.

12.2.2 Axis of Rotation

Rotation can also be specified as a counterclockwise (right-handed) rotation about the axis specified by the unit vector $(\omega_x, \omega_y, \omega_z)$. This is a very intuitive way of viewing rotation, but it has the same problems as Euler angles in numerical algorithms. The angle and axis representation can be converted into a rotation matrix for use in the formula for rigid body transformation (Equation 12.9), but it would be nice to have a scheme for working directly with the angle and axis representation that produced good numerical algorithms. This is part of the motivation for the quaternion representation for rotation, discussed in the next section.

12.2.3 Unit Quaternions

The quaternion is a representation for rotation that has been shown through experience to yield well-conditioned numerical solutions to orientation problems. A quaternion is a four-element vector,

$$\mathbf{q} = (q_0, q_1, q_2, q_3). \tag{12.17}$$

To understand how quaternions encode rotation, consider the unit circle in the x–y plane with the implicit equation

$$x^2 + y^2 = 1. \tag{12.18}$$

Positions on the unit circle correspond to rotation angles. In three dimensions, the unit sphere is defined by the equation

$$x^2 + y^2 + z^2 = 1. \tag{12.19}$$

Positions on the unit sphere in three dimensions encode the rotation angles of ω and ϕ about the x and y axes but cannot represent the twist κ about the z axis. One more degree of freedom is required to represent all three rotation angles. The unit sphere in four dimensions is defined by the implicit equation

$$x^2 + y^2 + z^2 + w^2 = 1. \tag{12.20}$$

All three rotation angles in three-dimensional space can be represented by points on the unit sphere in four dimensions.

Rotation is represented by unit quaternions with

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \tag{12.21}$$

Each unit quaternion and its antipole $-\mathbf{q} = (-q_0, -q_1, -q_2, -q_3)$ represent a rotation in three dimensions.

The rotation matrix for rigid body transformation can be obtained from the elements of the unit quaternion:

$$R(\mathbf{q}) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}. \tag{12.22}$$

After the unit quaternion is computed, Equation 12.22 can be used to compute the rotation matrix so that the rotation can be applied to each point using matrix multiplication.

The unit quaternion is closely related to the angle and axis representation for rotation, described in Section 12.2.2. A rotation can be represented as a scalar θ for the amount of rotation and a vector $(\omega_x, \omega_y, \omega_z)$ for the axis of rotation. A quaternion has a scalar part, which is related to the amount of rotation, and a vector part, which is the axis of rotation.

Let the axis of rotation be represented by the unit vector $(\omega_x, \omega_y, \omega_z)$ and use $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ to represent the coordinate axes so that the unit vector for the rotation axis can be represented as

$$\omega_x\mathbf{i} + \omega_y\mathbf{j} + \omega_z\mathbf{k}. \tag{12.23}$$

The unit quaternion for a counterclockwise rotation by θ about this axis is

$$\begin{aligned} \mathbf{q} &= \cos\frac{\theta}{2} + \sin\frac{\theta}{2}(\omega_x\mathbf{i} + \omega_y\mathbf{j} + \omega_z\mathbf{k}) & (12.24) \\ &= q_0 + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k}. & (12.25) \end{aligned}$$

The first term is called the scalar (real) part of the quaternion, and the other terms are called the vector (imaginary) part. A point $\mathbf{p} = (x, y, z)$ in space has a quaternion representation $\mathbf{r}$ which is the purely imaginary quaternion with vector part equal to $\mathbf{p}$,

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}. \tag{12.26}$$

Let $\mathbf{p}'$ be point $\mathbf{p}$ rotated by matrix $R(\mathbf{q})$,

$$\mathbf{p}' = R(\mathbf{q})\mathbf{p}. \tag{12.27}$$

If $\mathbf{r}$ is the quaternion representation for point $\mathbf{p}$, then the quaternion representation $\mathbf{r}'$ for the rotated point can be computed directly from the elements of quaternion $\mathbf{q}$,

$$\mathbf{r}' = \mathbf{q}\mathbf{r}\mathbf{q}^*, \tag{12.28}$$

where $\mathbf{q}^* = (q_0, -q_x, -q_y, -q_z)$ is the conjugate of quaternion $\mathbf{q}$ and quaternion multiplication is defined as

$$\begin{aligned} \mathbf{r}\mathbf{q} = \quad & (r_0 q_0 - r_x q_x - r_y q_y - r_z q_z, \\ & r_0 q_x + r_x q_0 + r_y q_z - r_z q_y, \\ & r_0 q_y - r_x q_z + r_y q_0 + r_z q_x, \\ & r_0 q_z + r_x q_y - r_y q_x + r_z q_0). \end{aligned} \tag{12.29}$$

Rigid body transformations can be conveniently represented using a seven-element vector, $(q_0, q_1, q_2, q_3, q_4, q_5, q_6)$, in which the first four elements are a unit quaternion and the last three elements are the translation. If we let $R(\mathbf{q})$ denote the rotation matrix corresponding to the unit quaternion in this representation, then the rigid body transformation is

$$\mathbf{p}_2 = R(\mathbf{q})\,\mathbf{p}_1 + (q_4, q_5, q_6)^T. \tag{12.30}$$

We will use quaternions in the next section to present an algorithm for solving the absolute orientation problem.

12.3 Absolute Orientation

The absolute orientation problem is the recovery of the rigid body transformation between two coordinate systems. One application for the absolute orientation problem is to determine the relationship between a depth measuring device, such as a range camera or binocular stereo system, and the absolute coordinate system defined for a scene so that all measurement points may be expressed in a common coordinate system. Let $\mathbf{p}_c = (x_c, y_c, z_c)$ denote the coordinates of a point in camera coordinates and $\mathbf{p}_a = (x_a, y_a, z_a)$ denote the coordinates of a point in absolute coordinates. The input to the absolute orientation problem is a set of conjugate pairs: $\{(\mathbf{p}_{c,1}, \mathbf{p}_{a,1}), (\mathbf{p}_{c,2}, \mathbf{p}_{a,2}), \ldots, (\mathbf{p}_{c,n}, \mathbf{p}_{a,n})\}$.

To develop a solution to the absolute orientation problem, expand the equation for the rigid body transformation from a point $\mathbf{p}_c$ in camera coordinates to a point $\mathbf{p}_a$ in absolute coordinates to expose the components of the rotation matrix:

$$\begin{aligned} x_a &= r_{xx}x_c + r_{xy}y_c + r_{xz}z_c + p_x \\ y_a &= r_{yx}x_c + r_{yy}y_c + r_{yz}z_c + p_y \\ z_a &= r_{zx}x_c + r_{zy}y_c + r_{zz}z_c + p_z. \end{aligned} \tag{12.31}$$

The unknowns are the 12 parameters of the transformation: the 9 elements of the rotation matrix and the 3 components of translation. Each conjugate pair yields three equations. At least four conjugate pairs are needed to get 12 equations for the 12 unknowns; in practice, a much larger number of calibration points is used to improve the accuracy of the result.

If the system of linear equations is solved without constraining the rotation matrix R to be orthonormal, the result may not be a valid rotation matrix. Using a nonorthonormal matrix as a rotation matrix can produce unexpected results: the matrix transpose is not necessarily the inverse of the matrix, and measurement errors in the conjugate pairs may influence the solution in ways that do not yield the best rigid body approximation. Some approaches orthogonalize the rotation matrix after each iteration, but there is no guarantee that the orthogonalized matrix will be the best approximation to the rotation. In Section 12.7, we present a method for solving the absolute orientation problem that guarantees that the solution matrix will be a rotation matrix. Another approach is to solve for the rotation angles rather than the entries in the rotation matrix; however, using the Euler angles leads to nonlinear algorithms with numerical difficulties. In photogrammetry, the nonlinear equations are linearized and solved to get corrections to the nominal values [103], but this approach assumes that good initial estimates are available.

There are many other representations for rotation that may yield good numerical methods, such as unit quaternions. Let $R(\mathbf{q})$ be the rotation matrix corresponding to a unit quaternion $\mathbf{q}$. The rigid body transformation that converts the coordinates of each point in camera coordinates to absolute coordinates is

$$\mathbf{p}_{a,i} = R(\mathbf{q})\,\mathbf{p}_{c,i} + \mathbf{p}_c, \tag{12.32}$$

where $\mathbf{p}_c$ is the location of the origin of the camera in the absolute coordinate system. Now the regression problem has seven parameters: the four elements in the unit quaternion for rotation plus the three elements in the translation vector.

As stated earlier, the input to the absolute orientation problem is a set of conjugate pairs: $\{(\mathbf{p}_{c,1}, \mathbf{p}_{a,1}), (\mathbf{p}_{c,2}, \mathbf{p}_{a,2}), \ldots, (\mathbf{p}_{c,n}, \mathbf{p}_{a,n})\}$. Consider the set of points in camera coordinates and the set of points in absolute coordinates as two sets of points: $P_a = \{\mathbf{p}_{a,1}, \mathbf{p}_{a,2}, \ldots, \mathbf{p}_{a,n}\}$ and $P_c = \{\mathbf{p}_{c,1}, \mathbf{p}_{c,2}, \ldots, \mathbf{p}_{c,n}\}$. The absolute orientation problem is to align these two clouds of points in space. Compute the centroid of each point cloud,

$$\bar{\mathbf{p}}_a = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_{a,i} \tag{12.33}$$

$$\bar{\mathbf{p}}_c = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_{c,i}, \tag{12.34}$$

and subtract the centroid from each point:

$$\mathbf{r}_{a,i} = \mathbf{p}_{a,i} - \bar{\mathbf{p}}_a \tag{12.35}$$

$$\mathbf{r}_{c,i} = \mathbf{p}_{c,i} - \bar{\mathbf{p}}_c. \tag{12.36}$$

After the rotation has been determined, the translation is given by

$$\mathbf{p}_c = \bar{\mathbf{p}}_a - R(\mathbf{q})\,\bar{\mathbf{p}}_c. \tag{12.37}$$

Now we are left with the problem of determining the rotation that will align the two bundles of rays.

In the rest of this derivation of the rotation, the points will be expressed as rays about the centroids and all coordinates will be ray coordinates. Since the bundles of rays were derived from the set of conjugate pairs, we know which ray in the camera bundle corresponds to each ray in the bundle for absolute coordinates. When the two bundles of rays are aligned, each corresponding pair of rays will be coincident, as illustrated in Figure 12.2. In other words, each pair of rays will lie along the same line and point in the same direction. Neglecting the effects of measurement errors, the angle between each pair of rays will be 0 and the cosine of this angle will be 1. Measurement errors will prevent the bundles from being perfectly aligned, but we can achieve

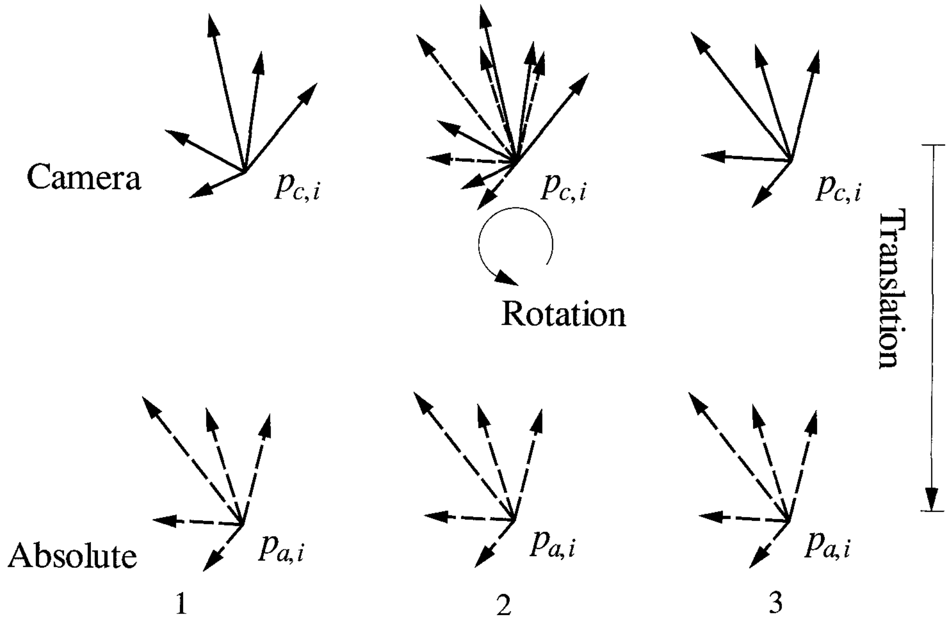

Figure 12.2: After the two bundles of rays (vectors of points about the centroids) are aligned and scaled, the centroid of one bundle can be translated to bring the two coordinate systems into alignment.

the best alignment in a least-squares sense by finding the rotation $R(\mathbf{q})$ that maximizes the scalar product of each ray pair:

$$\chi^2 = \sum_{i=1}^{n} \mathbf{r}_{a,i} \cdot R(\mathbf{q})\mathbf{r}_{c,i}. \tag{12.38}$$

In quaternion notation, this sum is

$$\sum_{i=1}^{n} \mathbf{r}_{a,i} \cdot \mathbf{q}\mathbf{r}_{c,i}\mathbf{q}^* = \sum_{i=1}^{n} (\mathbf{q}\mathbf{r}_{c,i}) \cdot (\mathbf{q}\mathbf{r}_{a,i}). \tag{12.39}$$

The sum can be successively changed into the notation of a quadratic form,

$$\sum_{i=1}^{n} (\mathbf{q}\mathbf{r}_{c,i}) \cdot (\mathbf{r}_{a,i}\mathbf{q}) = \sum_{i=1}^{n} (N_{c,i}\mathbf{q})^T (N_{a,i}\mathbf{q}) \tag{12.40}$$

$$= \sum_{i=1}^{n} \mathbf{q}^T N_{c,i}^T N_{a,i}\mathbf{q} \tag{12.41}$$

$$= \mathbf{q}^T \left(\sum_{i=1}^{n} N_{c,i}^T N_{a,i} \right) \mathbf{q} \tag{12.42}$$

$$= \mathbf{q}^T \left(\sum_{i=1}^{n} N_i \right) \mathbf{q} \tag{12.43}$$

$$= \mathbf{q}^T N \mathbf{q}, \tag{12.44}$$

assuming that $\mathbf{q}$ corresponds to a column vector. The unit quaternion that maximizes this quadratic form is the eigenvector corresponding to the most positive eigenvalue. The eigenvalues can be determined by solving a fourth-order polynomial using the formulas published by Horn [110], or the eigenvalues and eigenvectors can be calculated using standard numerical methods [197].

The matrices $N_{c,i}$ and $N_{a,i}$ are formed from the elements of each ray. Let $\mathbf{r}_{c,i} = (x_{c,i}, y_{c,i}, z_{c,i})$ and $\mathbf{r}_{a,i} = (x_{a,i}, y_{a,i}, z_{a,i})$; then

$$N_{c,i} = \begin{bmatrix} 0 & -x_{c,i} & -y_{c,i} & -z_{c,i} \\ x_{c,i} & 0 & z_{c,i} & -y_{c,i} \\ y_{c,i} & -z_{c,i} & 0 & x_{c,i} \\ z_{c,i} & y_{c,i} & -x_{c,i} & 0 \end{bmatrix} \tag{12.45}$$

$$N_{a,i} = \begin{bmatrix} 0 & -x_{a,i} & -y_{a,i} & -z_{a,i} \\ x_{a,i} & 0 & -z_{a,i} & y_{a,i} \\ y_{a,i} & z_{a,i} & 0 & -x_{a,i} \\ z_{a,i} & -y_{a,i} & x_{a,i} & 0 \end{bmatrix} \tag{12.46}$$

and the matrix N is

$$N = \begin{bmatrix} (S_{xx}+S_{yy}+S_{zz}) & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & (S_{xx}-S_{yy}-S_{zz}) & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & (-S_{xx}+S_{yy}-S_{zz}) & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & (-S_{xx}-S_{yy}+S_{zz}) \end{bmatrix}, \tag{12.47}$$

where the sums are taken over the elements of the ray coordinates in the camera and absolute coordinate systems:

$$S_{xx} = \sum_{i=1}^{n} x_{c,i} x_{a,i} \tag{12.48}$$

$$S_{xy} = \sum_{i=1}^{n} x_{c,i} y_{a,i} \tag{12.49}$$

$$S_{xz} = \sum_{i=1}^{n} x_{c,i} z_{a,i} \tag{12.50}$$

$$\vdots$$

In general, S_{kl} is the sum over all conjugate pairs of the product of coordinate k in the camera point and coordinate l in the absolute point:

$$S_{kl} = \sum_{i=1}^{n} k_{c,i} l_{a,i}. \tag{12.51}$$

The result of these calculations is the unit quaternion that represents the rotation that aligns the ray bundles. A rotation matrix can be obtained from the quaternion using Equation 12.22, and the translation part of the rigid body transformation can be obtained using Equation 12.37. The rigid body transformation can be applied to any point measurements generated by the depth measurement system, whether from a range camera, binocular stereo, or any other scheme, to transform the points into the absolute coordinate system.

12.4 Relative Orientation

The problem of relative orientation is to determine the relationship between two camera coordinate systems from the projections of corresponding points in the two cameras. The relative orientation problem is the first step in calibrating a pair of cameras for use in binocular stereo. We covered binocular stereo algorithms for matching features along epipolar lines in Section 11.2. To simplify the presentation, we assumed that the corresponding epipolar lines in the left and right image planes corresponded to the same rows in the left and right image arrays. This section will cover the solution to the relative orientation problem and show how the location of the epipolar lines in the two image planes can be determined. Section 12.5 will show how the left and right images can be resampled so that the epipolar lines correspond to the image rows, as assumed by the algorithms presented in Section 11.2. The disparities

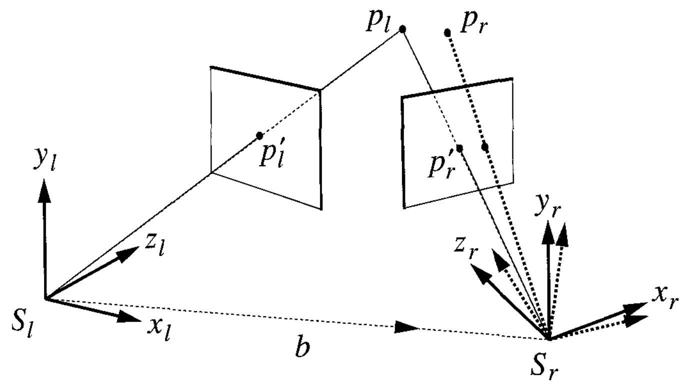

Figure 12.3: Illustration of the relative orientation problem for calibrating stereo cameras using the corresponding projections of scene points.

found by stereo matching are actually conjugate pairs. Section 12.6 will show how these conjugate pairs can be converted to point measurements in the coordinate system of the stereo device. The relative orientation problem is illustrated in Figure 12.3.

Suppose that a point $\mathbf{p}$ in the scene is within the view volume of two cameras, designated as the left and right cameras. Point $\mathbf{p}$ is denoted $\mathbf{p}_l$ in the coordinate system of the left camera and $\mathbf{p}_r$ in the coordinate system of the right camera. The projection of point $\mathbf{p}$ onto the image plane of the left camera is $\mathbf{p}'_l = (x'_l, y'_l)$ and the projection of the point onto the image plane of the right camera is $\mathbf{p}'_r = (x'_r, y'_r)$. From the equations for perspective projection:

$$\frac{x'_l}{f_l} = \frac{x_l}{z_l} \qquad \frac{y'_l}{f_l} = \frac{y_l}{z_l} \tag{12.52}$$

$$\frac{x'_r}{f_r} = \frac{x_r}{z_r} \qquad \frac{y'_r}{f_r} = \frac{y_r}{z_r}. \tag{12.53}$$

The rigid body transformation that transforms coordinates in the left camera system to coordinates in the right camera system is

$$x_r = r_{xx}x_l + r_{xy}y_l + r_{xz}z_l + p_x \tag{12.54}$$

$$y_r = r_{yx}x_l + r_{yy}y_l + r_{yz}z_l + p_y \tag{12.55}$$

$$z_r = r_{zx}x_l + r_{zy}y_l + r_{zz}z_l + p_z. \tag{12.56}$$

Solve the equations for perspective projection for x_l, y_l, x_r, and y_r and plug into the equations for a rigid body transformation to obtain a set of equations for the relationship between the projections of the conjugate pairs:

$$r_{xx}x'_l + r_{xy}y'_l + r_{xz}f_l + p_x\frac{f_l}{z_l} = x'_r\frac{z_r}{z_l}\frac{f_l}{f_r} \tag{12.57}$$

$$r_{yx}x'_l + r_{yy}y'_l + r_{yz}f_l + p_y\frac{f_l}{z_l} = x'_r\frac{z_r}{z_l}\frac{f_l}{f_r} \tag{12.58}$$

$$r_{zx}x'_l + r_{zy}y'_l + r_{zz}f_l + p_z\frac{f_l}{z_l} = x'_r\frac{z_r}{z_l}\frac{f_l}{f_r}. \tag{12.59}$$

The rotation part of the transformation changes the orientation of the left camera so that it coincides with the orientation of the right camera. The translation is the baseline between the two cameras. The variables for translation and depth appear as ratios in the equations, which means that the length of the baseline and depth can be scaled arbitrarily. For example, you can separate the cameras by twice as much and move the points in the scene twice as far away without changing the perspective geometry.

It is not possible to determine the baseline distance from the projections of the calibration points. This is not a serious problem, as the scale factor can be determined later by other means. For now, assume that the translation between cameras is a unit vector. Solving the relative orientation problem provides the three parameters for rotation and the two parameters for a unit vector that represents the direction of baseline. The binocular stereo depth measurements scale with the baseline distance. Assuming a unit baseline distance means that the binocular stereo measurements will be in an arbitrary system of units. The measurements obtained under the assumption of unit baseline distance will be correct except for the unknown scale factor. Relative distances between points will be correct. These arbitrary units can be converted to real units by multiplying by the baseline distance after it is obtained. Section 12.7 shows how the baseline distance can be determined as part of the solution to the absolute orientation problem. The conversion of stereo measurements from arbitrary units to real units and the transformation of point coordinates from viewer-centered coordinates to absolute coordinates can be done simultaneously by applying the transformation obtained from this augmented absolute orientation problem.

The rotation matrix is orthonormal, and this provides six additional constraints in addition to the artificial constraint of unit baseline distance. Given

n calibration points, there are $12 + 2n$ unknowns and $7 + 3n$ constraints. At least five conjugate pairs are needed for a solution, but in practice many more calibration points would be used to provide more accuracy.

The relative orientation problem starts with a set of calibration points and determines the rigid body transformation between the left and right cameras using the projections of these calibration points in the left and right image planes. Each calibration point $\mathbf{p}$ in the scene projects to point $\mathbf{p}'_l$ in the left camera and point $\mathbf{p}'_r$ in the right camera. Each projected point corresponds to a ray from the center of projection of its camera, through the projected point, and into the scene. The rays corresponding to $\mathbf{p}'_l$ and $\mathbf{p}'_r$ should intersect at point $\mathbf{p}$ in the scene, but may not intersect due to errors in measuring the projected locations in the image planes. We want to find the relative position and orientation of the two cameras in space, subject to the constraint of unit baseline distance, so that the errors in the locations of the rays in the image planes are minimized.

Let $\mathbf{r}_l$ be the ray (vector) from the center of projection of the left camera through point $\mathbf{p}'_l$ in the left image plane, let $\mathbf{r}_r$ be the ray from the center of projection of the right camera through point $\mathbf{p}'_r$ in the right image plane, and let $\mathbf{b}$ be the vector from the center of projection of the left camera to the center of projection of the right camera. We need to work with each ray in the same coordinate system, so rotate $\mathbf{r}_l$ so that it is in the same coordinate system as ray $\mathbf{r}_r$ and let $\mathbf{r}'_l$ denote this rotated ray. If the two rays intersect, then they lie in the plane normal to $\mathbf{r}'_l \times \mathbf{r}_r$. The baseline lies in this same plane, so the baseline is normal to $\mathbf{r}'_l \times \mathbf{r}_r$. This relationship is expressed mathematically by saying that the dot product of the baseline with the normal to the plane is zero:

$$\mathbf{b} \cdot (\mathbf{r}'_l \times \mathbf{r}_r) = 0. \tag{12.60}$$

This relationship is called the coplanarity condition.

Due to measurement errors, the rays will not intersect and the coplanarity condition will be violated. We can formulate a least-squares solution to the relative orientation problem by minimizing the sum of the squared errors of deviations from the triple products that represent the coplanarity condition:

$$\chi^2 = \sum_{i=1}^{n} w_i(\mathbf{b} \cdot (\mathbf{r}'_{l,i} \times \mathbf{r}_{r,i}))^2, \tag{12.61}$$

where the weight counteracts the effect that when the triple product is near zero, changes in the baseline and rotation make a large change in the triple product. It is good practice in any regression problem to scale the relationship between the parameters and the data values so that small changes in the parameter space correspond to small changes in the data space. The weight on each triple product is

$$w_i = \frac{\|\mathbf{r}'_{l,i} \times \mathbf{r}_{r,i}\|^2 \sigma_0^2}{\left[(\mathbf{b}\times\mathbf{r}_{r,i})\cdot(\mathbf{r}'_{l,i}\times\mathbf{r}_{r,i})\right]^2 \|\mathbf{r}'_{l,i}\|^2\sigma_l^2 + \left[(\mathbf{b}\times\mathbf{r}'_{l,i})\cdot(\mathbf{r}'_{l,i}\times\mathbf{r}_{r,i})\right]^2 \|\mathbf{r}'_{r,i}\|^2\sigma_r^2}. \tag{12.62}$$

We need to switch to matrix notation, so assume that all vectors are column vectors. The error to be minimized is

$$\chi^2 = \sum_{i=1}^{n} w_i(\mathbf{b}\cdot\mathbf{c}_i)^2 \tag{12.63}$$

$$= \mathbf{b}^T \left(\sum_{i=1}^{n} w_i \mathbf{c}_i \mathbf{c}_i^T\right) \mathbf{b} \tag{12.64}$$

$$= \mathbf{b}^T C \mathbf{b} \tag{12.65}$$

subject to the constraint that $\mathbf{b}^T\mathbf{b} = 1$. The term $\mathbf{c}_i\mathbf{c}_i^T$ is an outer product, a 3×3 matrix formed by multiplying a 3×1 matrix with a 1×3 matrix, and is a real, symmetric matrix. The constrained minimization problem is

$$\chi^2 = \mathbf{b}^T C\mathbf{b} + \lambda(1 - \mathbf{b}^T\mathbf{b}), \tag{12.66}$$

where λ is the Lagrange multiplier. Differentiating with respect to vector $\mathbf{b}$,

$$C\mathbf{b} = \lambda\mathbf{b}. \tag{12.67}$$

The solution for the baseline is the unit eigenvector corresponding to the smallest eigenvalue of the C matrix.

The results above say that given an initial estimate for the rotation between the two cameras, we can determine the unit vector for the baseline. There is no closed-form solution for determining the rotation, but the estimates for the rotation and baseline can be refined using iterative methods. We will develop a system of linear equations for computing incremental improvements for the baseline and rotation.

The improvement $\delta\mathbf{b}$ to the baseline must be perpendicular to the baseline, since the unit vector representing the baseline cannot change length,

$$\mathbf{b} \cdot \delta\mathbf{b} = 0, \tag{12.68}$$

and the improvement to the rotation of the left ray into the right coordinate system is the infinitesimal rotation vector $\delta\boldsymbol{\omega}$. The corrections to the baseline and rotation will change each triple product for each ray from $t_i = \mathbf{b} \cdot (\mathbf{r}'_{l,i} \times \mathbf{r}_{r,i})$ to $t_i + \delta t_i$, with the increment in the triple product given by

$$\mathbf{c}_i \cdot \delta\mathbf{b} + \mathbf{d}_i \cdot \delta\boldsymbol{\omega}, \tag{12.69}$$

where

$$\mathbf{c}_i = \mathbf{r}'_{l,i} \times \mathbf{r}_{r,i} \tag{12.70}$$

$$\mathbf{d}_i = \mathbf{r}'_{l,i} \times (\mathbf{r}_{r,i} \times \mathbf{b}). \tag{12.71}$$

The corrections are obtained by minimizing

$$\chi^2 = \sum_{i=1}^{n} w_i (t_i + \mathbf{c}_i \cdot \delta\mathbf{b} + \mathbf{d}_i \cdot \delta\boldsymbol{\omega})^2, \tag{12.72}$$

subject to the constraint that $\delta\mathbf{b} \cdot \mathbf{b} = 0$. The constraint can be added onto the minimization problem using the Lagrange multiplier λ to get a system of linear equations for the baseline and rotation increments and the Lagrange multiplier:

$$\begin{bmatrix} C & F & \mathbf{b} \\ F^T & D & 0 \\ \mathbf{b}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\mathbf{b} \\ \delta\boldsymbol{\omega} \\ \lambda \end{bmatrix} = - \begin{bmatrix} \bar{\mathbf{c}} \\ \bar{\mathbf{d}} \\ 0 \end{bmatrix}, \tag{12.73}$$

where

$$C = \sum_{i=1}^{n} w_i \mathbf{c}_i \mathbf{c}_i^T \tag{12.74}$$

$$F = \sum_{i=1}^{n} w_i \mathbf{c}_i \mathbf{d}_i^T \tag{12.75}$$

$$D = \sum_{i=1}^{n} w_i \mathbf{d}_i \mathbf{d}_i^T \tag{12.76}$$

$$\bar{c} = \sum_{i=1}^{n} w_i t_i \mathbf{c}_i^T \tag{12.77}$$

$$\bar{d} = \sum_{i=1}^{n} w_i t_i \mathbf{d}_i^T. \tag{12.78}$$

Once we have the corrections to the baseline and rotation, we have to apply the corrections in a way that preserves the constraints that the baseline is a unit vector and the rotation is represented correctly. For example, if rotation is represented as an orthonormal matrix, the corrected matrix must be orthonormal. It is difficult to update a rotation matrix without violating orthonormality, so the rotation will be represented as a unit quaternion.

The baseline is updated using the formula

$$\mathbf{b}^{n+1} = \mathbf{b}^n + \delta\mathbf{b}^n. \tag{12.79}$$

The updated baseline should be explicitly normalized to guarantee that numerical errors do not lead to violation of the unit vector constraint.

The infinitesimal rotation can be represented by the unit quaternion

$$\delta\mathbf{q} = \sqrt{1 - \frac{1}{4}\|\delta\boldsymbol{\omega}\|^2} + \frac{1}{2}\delta\boldsymbol{\omega}. \tag{12.80}$$

This formula guarantees that the quaternion will be a unit quaternion even if the rotation is large. If $\mathbf{r}$ is the quaternion representing the rotation, then the updated rotation is the quaternion $\mathbf{r}'$ given by

$$\mathbf{r}' = \mathbf{q}\mathbf{r}\mathbf{q}^*, \tag{12.81}$$

where multiplication is performed according to the rules for multiplying quaternions (Section 12.2.3), and $\mathbf{q}^*$ is the conjugate of quaternion $\mathbf{q}$.

12.5 Rectification

Rectification is the process of resampling stereo images so that the epipolar lines correspond to image rows. The basic idea is simple: if the left and right image planes are coplanar and the horizontal axes are colinear (no rotation about the optical axes), then the image rows are epipolar lines and stereo correspondences can be found by searching for matches along corresponding rows.

In practice, this condition can be difficult to achieve and some vergence (inward rotation about the vertical camera axes) may be desirable, but if the pixels in the left and right images are projected onto a common plane, then the ideal epipolar geometry is achieved. Each pixel in the left (right) camera corresponds to a ray in the left (right) camera coordinate system. Let T_l and T_r be the rigid body transformations that bring rays from the left and right cameras, respectively, into the coordinate system of the common plane. Determine the locations in the common plane of the corners of each image, create new left and right image grids, and transform each grid point back into its original image. Bilinear interpolation, discussed in Section 13.6.2, can be used to interpolate pixel values to determine the pixel values for the new left and right images in the common plane.

12.6 Depth from Binocular Stereo

Binocular stereo matches feature points in the left and right images to create a set of conjugate pairs, $\{(\mathbf{p}_{l,i}, \mathbf{p}_{r,i})\}, i = 1, \ldots, n$. Each conjugate pair defines two rays that (ideally) intersect in space at a scene point. The space intersection problem is to find the three-dimensional coordinates of the point of intersection. Due to errors in measuring the image plane coordinates and errors in the cameras, the rays will not intersect, so the problem of computing depth from stereo pairs is to find the coordinates of the scene point that is closest to both rays.

We will assume that stereo measurements will be made in a coordinate system that is different from the coordinate systems of either camera. For example, the stereo coordinate system might be attached to the frame that holds the two cameras. There will be two rigid body transformations: one aligns the left camera with stereo coordinates and the other aligns the right camera with stereo coordinates. The left transformation has rotation matrix R_l and translation $\mathbf{p}_l = (x_l, y_l, z_l)$, and the right transformation has rotation matrix R_r and translation $\mathbf{p}_r = (x_r, y_r, z_r)$. To represent point measurements in the coordinate system of the right (left) camera, use the rigid body transformation obtained from solving the relative orientation problem (or its inverse) for the left camera transformation and use the identity transformation for the right (left) camera.

The coordinates of the conjugate pair in three dimensions are $(x'_{l,i}, y'_{l,i}, f_l)$ and $(x'_{r,i}, y'_{r,i}, f_r)$. Rotate and translate the left camera coordinates into stereo coordinates,

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_l \\ y_l \\ z_l \end{pmatrix} + t_l R_l \begin{pmatrix} x'_{l,i} \\ y'_{l,i} \\ f_l \end{pmatrix}, \tag{12.82}$$

and rotate and translate the right camera coordinates into stereo coordinates,

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} + t_r R_r \begin{pmatrix} x'_{r,i} \\ y'_{r,i} \\ f_r \end{pmatrix}. \tag{12.83}$$

In order to find the point that is close to both rays, find the values for t_l and t_r that correspond to the minimum distance between the rays by minimizing the norm

$$\chi^2 = \left[\begin{pmatrix} x_l \\ y_l \\ z_l \end{pmatrix} + t_l R_l \begin{pmatrix} x'_{l,i} \\ y'_{l,i} \\ f_l \end{pmatrix} - \begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} - t_r R_r \begin{pmatrix} x'_{r,i} \\ y'_{r,i} \\ f_r \end{pmatrix} \right]^2 \tag{12.84}$$

$$= [\mathbf{b} + t_l \mathbf{r}_l - t_r \mathbf{r}_r]^2, \tag{12.85}$$

where $\mathbf{b}$ is the baseline in stereo coordinates, and $\mathbf{r}_l$ and $\mathbf{r}_r$ are the left and right rays rotated into stereo coordinates. To solve the ray intersection problem, differentiate with respect to t_l and t_r and set the result equal to zero. Solve the equations for t_l and t_r and plug the solution values for the parameters into the ray equations (Equation 12.82 and 12.83) to obtain the point on each ray that is closest to the other ray. Average the two point locations to obtain the depth estimate.

The stereo point measurements are in the coordinate system of the stereo system, either the left or right camera, or a neutral coordinate system. If the algorithm for relative orientation presented in Section 12.4 was used to determine the baseline, then the measurements are in a unitless system of measurement. If the rigid body transformations between the left and right cameras and another coordinate system for the stereo hardware were obtained by solving the exterior orientation problem (Section 12.8) or by other means, then the stereo point measurements are in the units that were used for the calibration points. Regardless of how the stereo system was calibrated, we have to transform the point measurements into an absolute coordinate system for the scene. We can convert the measurements into a system of units appropriate to the scene at the same time by solving the absolute orientation problem with a scale factor.

12.7 Absolute Orientation with Scale

The formulation of the absolute orientation problem in Section 12.3 does not allow transformations that include a scale change; the transformation between coordinate systems is a rigid body transformation which includes only rotation and translation. Scale changes occur, for example, in binocular stereo when the baseline between the stereo cameras is unknown or incorrect or between range cameras with different measurement units.

The absolute orientation problem, presented in Section 12.3, can be extended to include scale changes. The solution to this extended problem will be a transformation that includes rotation and translation to align the viewpoint to an absolute coordinate system and includes a scale factor to convert camera-specific measurement units to the common system of units. Consider a point $\mathbf{p}$ with coordinates $\mathbf{p}_1 = (x_1, y_1, z_1)$ in one coordinate system and coordinates $\mathbf{p}_2 = (x_2, y_2, z_2)$ in another coordinate system. The transformation between coordinates is

$$\mathbf{p}_2 = sR\mathbf{p}_1 + \mathbf{p}_0, \tag{12.86}$$

where s is the scale change. This increases the number of parameters in the absolute orientation problem to seven: three parameters for rotation, three parameters for translation, and the scale factor. The scaling transformation is uniform scaling: the coordinates of each axis are scaled by the same amount.

The input to the absolute orientation problem is a set of n conjugate pairs for the first and second views: $\{(\mathbf{p}_{1,i}, \mathbf{p}_{2,i})\}$. The regression problem is to find the rotation R, translation $\mathbf{p}_0$, and scale s that minimize

$$\sum_{i=1}^{n} \left(\mathbf{p}_{2,i} - sR\mathbf{p}_{1,i} - \mathbf{p}_0\right)^2. \tag{12.87}$$

The solution requires at least three points to get nine equations for the seven unknowns. In practice, more points are used to provide better accuracy.

Ignore for the moment that the correspondence between points is known and imagine that the two sets of points (the set of points in the first coordinate system and the set in the second system) are two clouds of points in absolute coordinate space. Compute the centroid of each cloud of points:

$$\bar{\mathbf{p}}_1 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_{1,i} \tag{12.88}$$

$$\bar{\mathbf{p}_2} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_{2,i} \tag{12.89}$$

and transform each cloud of points to a bundle of vectors about the centroid:

$$\mathbf{r}_{1,i} = \mathbf{p}_{1,i} - \bar{\mathbf{p}_1}, \qquad \mathbf{r}_{2,i} = \mathbf{p}_{2,i} - \bar{\mathbf{p}_2}. \tag{12.90}$$

The scale parameter can be determined by computing the mean length of the vectors in each bundle:

$$s^2 = \frac{\sum_{i=1}^{n} \|\mathbf{r}_{2,i}\|^2}{\sum_{i=1}^{n} \|\mathbf{r}_{1,i}\|^2}. \tag{12.91}$$

The scale factor can be computed without knowing either the rotation or the translation. This is a very useful formula for calibrating the baseline distance in binocular stereo and is more accurate than using only a few points.

After the rotation and scale factor are determined, the translation can be easily computed from the centroids:

$$\mathbf{p}_0 = \bar{\mathbf{p}_2} - sR\bar{\mathbf{p}_1}. \tag{12.92}$$

Computing the rotation is essentially the problem of determining how to align the bundles of rays about the centroids. Form the matrix M from the sum of scalar products of the coodinates of the rays in the first and second views:

$$M = \sum_{i=1}^{n} \mathbf{r}_{2,i}(\mathbf{r}_{1,i})^T. \tag{12.93}$$

Let matrix $Q = M^TM$. The rotation matrix is

$$R = MS^{-1}, \tag{12.94}$$

where matrix S is

$$S = Q^{1/2}. \tag{12.95}$$

The eigenvalue-eigenvector decomposition of matrix Q is

$$Q = \lambda_1\mathbf{v}_1\mathbf{v}_1^T + \lambda_2\mathbf{v}_2\mathbf{v}_2^T + \lambda_3\mathbf{v}_3\mathbf{v}_3^T \tag{12.96}$$

The eigenvalues of M^TM are obtained by solving a cubic equation. The roots of the cubic equation can be computed from direct formulas [197]. Use the eigenvalues to solve the linear equations

$$\left(M^TM - \lambda_i I\right)\mathbf{v}_i = 0, \tag{12.97}$$

for the orthogonal eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$. Matrix S is the square root of matrix Q. Fortunately, matrix square roots and their inverses are easy to compute in the eigensystem representation (Equation 12.96). The inverse of matrix S is

$$S^{-1} = \left(M^T M\right)^{-1/2} = \frac{1}{\sqrt{\lambda_1}}\mathbf{v}_1\mathbf{v}_1^T + \frac{1}{\sqrt{\lambda_2}}\mathbf{v}_2\mathbf{v}_2^T + \frac{1}{\sqrt{\lambda_3}}\mathbf{v}_3\mathbf{v}_3^T. \tag{12.98}$$

Compute the outer products of the eigenvectors, divided by the square root of the eigenvectors, and multiply this matrix by M to get the rotation matrix R. This method of construction guarantees that matrix R will be an orthonormal matrix.

This algorithm provides a closed form (noniterative) solution for the rotation matrix. The scale can be determined without determining either the translation or rotation using the formula in Equation 12.91 and, finally, the translation can be determined using the formula in Equation 12.92. The transformation in Equation 12.86 can be applied to the point measurements from any depth measurement system, including binocular stereo or a range camera, to align the point measurements to an absolute coordinate system and convert the measurements into the units of the absolute coordinate system. The measurement units will be whatever system of units was used for the coordinates of the calibration points used for the absolute orientation problem. For example, if the calibration points in the absolute coordinate system are in millimeters and the depth measurements are from binocular stereo with a unit baseline, then the rigid body transformation obtained by solving the absolute orientation problem with these calibration points will transform stereo measurements to millimeters. The system of measurement must be the same along each coordinate axis since the same scale factor s is applied to each coordinate.

12.8 Exterior Orientation

The problem of exterior orientation is to determine the relationship between image plane coordinates (x', y') and the coordinates (x, y, z) of scene points in an absolute coordinate system. The exterior orientation problem is called the hand–eye problem in robotics and machine vision. Recall from Section 1.4.1 on perspective projection that a point (x, y, z) in the viewer-centered coordinate system of the camera is projected to a point (x', y') in the image plane. Until this section, we have been content to represent the coordinates of points in the scene in the coordinate system of the camera; in many applications, though, it is necessary to relate the coordinates of measurements computed in the image plane coordinate system to the absolute coordinate system defined for the scene. Each point (x', y') in the image plane defines a ray from the center of projection, passing through (x', y') in the image plane, and continuing on into the scene. The position of the camera in the scene is the location of the center of projection, and the orientation of the camera determines the orientation of the bundle of rays from the center of projection, passing through image plane points. An image plane point does not correspond to a unique point in the scene, but we may be able to use the equation for the ray passing through the image point, along with other information about the scene geometry, to determine a unique point in absolute coordinates. For example, if we know that an image plane point corresponds to a point on a wall, and if we know the equation for the plane that models the wall, then the exact location of the point on the wall can be obtained by solving the system of ray and plane equations for the intersection.

The exterior orientation problem is to determine the position and orientation of the bundle of rays corresponding to image plane points so that the coordinates of each image plane point may be transformed to its ray in the absolute coordinate system of the scene. The problem is illustrated in Figure 12.4. The position and orientation of the bundle of rays is represented as the rigid body transformation from camera coordinates to absolute coordinates. An image plane point (x', y') has coordinates (x', y', f) in the three-dimensional coordinate system of the camera, with the image plane at a distance f in front of the center of projection. The center of projection corresponds to the origin of the camera coordinate system. The position of the camera in the scene is the location of the center of projection in absolute coordinates. In camera coordinates, the parametric equation of the ray passing through point (x', y') in the image plane is

$$(x, y, z) = t(x', y', f), \tag{12.99}$$

with parameter t going from zero (at the center of projection) to infinity. At $t = 1$, the point (x, y, z) in the camera coordinate system is the image plane point (x', y', f). Given a measured location (x', y') in the image and

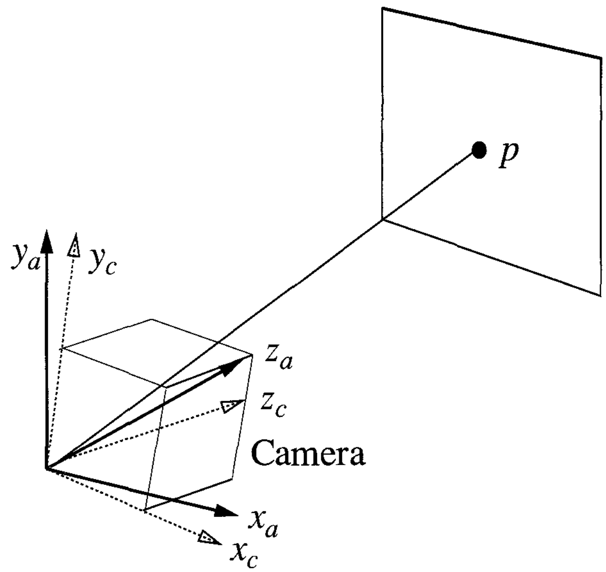

Figure 12.4: Illustration of how the position and orientation of the camera are determined by aligning the bundle of rays with corresponding scene points.

an estimate for the camera constant f, we have the equation for the ray in camera-centered coordinates.

Let $\mathbf{p} = (x, y, z)^T$ and $\mathbf{p}' = (x', y', f)^T$. The equation for the ray in the absolute coordinate system for the scene is obtained by applying the rigid body transformation from camera coordinates to absolute coordinates to the parametric equation for the ray (Equation 12.99),

$$\mathbf{p} = tRp' + \mathbf{p}_0. \tag{12.100}$$

The rigid body transformation can be determined by measuring the position (x_i', y_i') of the projection of calibration points in the scene with known position (x_i, y_i, z_i), relating the image plane points to the scene points using the equations for perspective projection, and solving for the rigid body transformation. The parameters for the exterior orientation of the camera (rotation angles and translation vector to the camera origin) are called the extrinsic parameters, as opposed to the intrinsic parameters of the internal geometry of the camera such as the camera constant.

The exterior orientation problem can be succinctly stated: what is the rigid body transformation from absolute coordinates to camera coordinates that positions and orients the bundle of rays in space so that each ray passes through its corresponding calibration point? To make the coordinate systems clear, we will use subscripts to distinguish absolute coordinates from camera coordinates. The position of a point in absolute coordinates is

$$\mathbf{p}_a = (x_a, y_a, z_a)^T \tag{12.101}$$

and the position of a point in camera coordinates is

$$\mathbf{p}_c = (x_c, y_c, z_c)^T. \tag{12.102}$$

We will develop the exterior orientation problem as a transformation from absolute (scene) coordinates to camera coordinates. The inverse of this transformation, needed for practical applications, is given by Equation 12.16. The rigid body transformation from absolute coordinates to camera coordinates is

$$x_c = r_{xx}x_a + r_{xy}y_a + r_{xz}z_a + p_x \tag{12.103}$$

$$y_c = r_{yx}x_a + r_{yy}y_a + r_{yz}z_a + p_y \tag{12.104}$$

$$z_c = r_{zx}x_a + r_{zy}y_a + r_{zz}z_a + p_z. \tag{12.105}$$

The positions of the points in the coordinate system of the camera are unknown, but the projection of the points onto the image plane is determined by the equations for perspective projection:

$$\frac{x'}{f} = \frac{x_c}{z_c} \tag{12.106}$$

$$\frac{y'}{f} = \frac{y_c}{z_c}. \tag{12.107}$$

Solve the perspective equations for x_c and y_c, and combine the results for the first two equations for the transformation from absolute to camera coordinates (Equations 12.103 and 12.104) with Equation 12.105 as the denominator to obtain two equations relating image plane coordinates (x', y') to the absolute coordinates for a calibration point:

$$\frac{x'}{f} = \frac{r_{xx}x_a + r_{xy}y_a + r_{xz}z_a + p_x}{r_{zx}x_a + r_{zy}y_a + r_{zz}z_a + p_z} \tag{12.108}$$

$$\frac{y'}{f} = \frac{r_{yx}x_a + r_{yy}y_a + r_{yz}z_a + p_y}{r_{zx}x_a + r_{zy}y_a + r_{zz}z_a + p_z}. \tag{12.109}$$

Each calibration point yields two equations that constrain the transformation:

$$\begin{aligned} x'(r_{zx}x_a+r_{zy}y_a+r_{zz}z_a+p_z) - f(r_{xx}x_a+r_{xy}y_a+r_{xz}z_a+p_x) &= 0 \quad (12.110) \\ y'(r_{zx}x_a+r_{zy}y_a+r_{zz}z_a+p_z) - f(r_{yx}x_a+r_{yy}y_a+r_{yz}z_a+p_y) &= 0. \quad (12.111) \end{aligned}$$

Six calibration points would yield 12 equations for the 12 transformation parameters, but using the orthonormality constraints for the rotation matrix reduces the minimum number of calibration points to four. Many more points would be used in practice to provide an accurate solution. Replace the elements of the rotation matrix with the formulas using Euler angles and solve the nonlinear regression problem.

12.8.1 Calibration Example

A robot is equipped with a suction pickup tool at the end of the arm. The tool is good for picking up small, flat objects if the suction tool is positioned near the center of the object. There are flat objects on a table within reach of the arm. The absolute coordinate system is at one corner of the table. A camera is positioned above the table, with the table within the field of view. The position of a point in the image plane is (x', y'). If the object has good contrast against the background of the table top, then the image plane position can be estimated using first moments. The position of a point in the absolute coordinate system of the table is (x, y, z). The position and orientation of the camera relative to the absolute coordinate system can be determined by solving the exterior orientation problem.

Given the position (x', y') of the center of the object in the image plane, the location (x, y, z) in the absolute coordinate system of the center of the part on the table is given by intersecting the ray from the camera origin through (x', y') with the plane of the table top. Both the equation for the table top

$$ax + by + cz + d = 0 \quad (12.112)$$

and the equation for the ray from the camera origin

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = t \begin{pmatrix} x' \\ y' \\ f \end{pmatrix} \quad (12.113)$$

must be in the absolute coordinate system. The ray from the camera can be transformed to the absolute coordinate system using the transformation obtained by solving the exterior orientation problem. If the origin of the absolute coordinate system is in the plane of the table with the z axis normal to the table, then Equation 12.112 reduces to $z = 0$ and the intersection is easy to compute.

12.9 Interior Orientation

The problem of interior orientation is to determine the internal geometry of the camera. The geometry is represented by a set of camera parameters:

Camera constant for the distance of the image plane from the center of projection

Principal point for the location of the origin of the image plane coordinate system

Lens distortion coefficients for the changes in image plane coordinates caused by optical imperfections in the camera

Scale factors for the distances between the rows and columns

The interior orientation problem is the problem of compensating for errors in the construction of the camera so that the bundle of rays inside the camera obeys the assumptions of perspective projection. The camera parameters are called the intrinsic parameters, as opposed to the extrinsic parameters for the exterior orientation of the camera. The interior orientation problem is the regression problem for determining the intrinsic parameters of a camera.

The camera constant is not the same as the focal length of the lens. When the lens is focused at infinity, then the camera constant is equal to the focal length; otherwise, the camera constant is less than the focal length. The principal point is where the optical axis intersects the image plane. It establishes the origin of the image plane coordinate system, which up to this point has been assumed to be the center of the image array (see Equations 12.1 and 12.2). Although the camera constant is close to the focal length and the principal point is close to the center of the image, these approximations may not be good enough for many applications. The spacing between the rows and columns of pixels in the image sensor can be determined from the camera specifications, but frame grabbers may introduce errors that must be calibrated.

Some calibration algorithms solve the interior orientation problems and exterior orientation problems at the same time. The motivation for this is that the true location of the calibration points on the image plane cannot be known until the exterior orientation of the camera has been determined. However, the interior orientation problem can be solved by itself, and there are several methods for determining the camera constant, location of the principal point, and lens distortions without knowing the exterior orientation of the camera in absolute coordinates. Methods for determining both the intrinsic and extrinsic parameters are covered in Section 12.10.

The fundamental idea in determining the intrinsic camera parameters independent of the extrinsic parameters is to use a calibration image with some regular pattern, such as a grid of lines. Distortions in the pattern are used to estimate the lens distortions and calculate corrections to the nominal values for the other intrinsic parameters.

Lens distortions include two components: radial distortion that bends the rays of light by more or less than the correct amount, and decentering caused by a displacement of the center of the lens from the optical axis. The radial distortion and decentering effects are modeled as polynomials; the interior orientation algorithm estimates the coefficients of these polynomials. Figure 12.5 illustrates the radially symmetric nature of most lens distortions, in the absence of lens decentering errors. Light rays are bent toward the optical axis by more or less than the correct amount, but this error is the same at all positions on the lens (or in the image plane) that are the same distance from the principal point.

The radial distortion can be modeled as a polynomial in even powers of the radius, since error in the amount of bending of the rays of light is rotationally symmetric. Let (x', y') denote the true image coordinates and $(\tilde{x}, \tilde{y})$ denote the uncorrected image coordinates obtained from the pixel coordinates i and j using an estimate for the location of the principal point:

$$\tilde{x} = j - \hat{c_x} \tag{12.114}$$

$$\tilde{y} = -(i - \hat{c_y}). \tag{12.115}$$

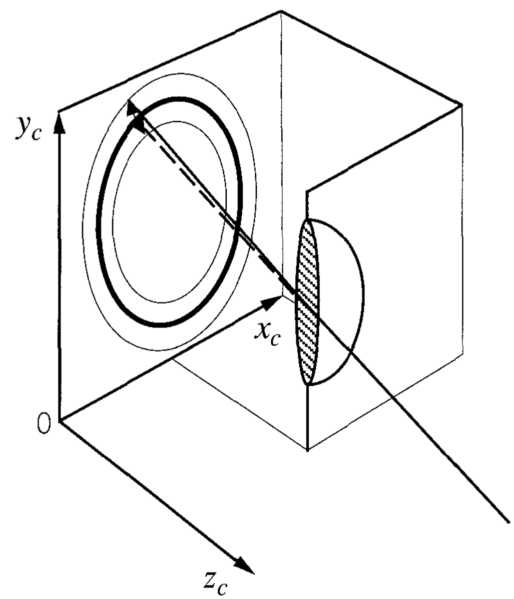

Figure 12.5: Most lens distortions are radially symmetric. The rays are bent toward the center of the image by more or less than the correct amount. The amount of radial distortion is the same at all points in the image plane that are the same distance from the true location of the principal point.

The corrections $(\delta x, \delta y)$ will be added to the uncorrected coordinates to get the true image plane coordinates:

$$x' = \tilde{x} + \delta x \tag{12.116}$$
$$y' = \tilde{y} + \delta y. \tag{12.117}$$

The corrections for radial lens distortions are modeled by a polynomial in even powers of the radial distance from the center of the image plane:

$$\delta x = (\tilde{x} - x_p)(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6) \tag{12.118}$$
$$\delta y = (\tilde{y} - y_p)(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6), \tag{12.119}$$

where (x_p, y_p) is the refinement to the location of the principal point and

$$r^2 = (\tilde{x} - x_p)^2 + (\tilde{y} - y_p)^2 \tag{12.120}$$

is the square of the radial distance from the center of the image. Note that x_p and y_p are not the same as $\hat{c_x}$ and $\hat{c_y}$ in Equations 12.114 and 12.115, which are used to compute the uncorrected image coordinates; x_p and y_p are corrections to $\hat{c_x}$ and $\hat{c_y}$. After calibration, the corrections can be applied to the initial estimates:

$$c_x = \hat{c_x} + x_p \tag{12.121}$$

$$c_y = \hat{c_y} - y_p. \tag{12.122}$$

The calibration problem for correcting radial distortion is to find the coefficients, κ_1, κ_2, and κ_3, of the polynomial. Lens distortion models beyond the sixth degree are rarely used; in fact, it may not be necessary to go beyond a second-degree polynomial. The location of the principal point is included in the calibration problem since an accurate estimate for the location of the principal point is required to model the lens distortions. More powerful models for lens distortions can include tangential distortions due to effects such as lens decentering:

$$\begin{aligned}\delta x = {} & (\tilde{x}-x_p)(\kappa_1 r_2+\kappa_2 r_4+\kappa_3 r_6) \\ & + \left[p_1\left(r^2+2(\tilde{x}-x_p)^2\right) + 2p_2(\tilde{x}-x_p)(\tilde{y}-y_p)\right](1+p_3 r^2)\end{aligned} \tag{12.123}$$

$$\begin{aligned}\delta y = {} & (\tilde{y}-y_p)(\kappa_1 r_2+\kappa_2 r_4 + \kappa_3 r_6) \\ & + \left[2p_1(\tilde{x}-x_p)(\tilde{y}-y_p) + 2p_2\left(r^2+2(\tilde{y}-y_p)^2\right)\right](1+p_3 r^2).\end{aligned} \tag{12.124}$$

Use a calibration target consisting of several straight lines at different positions and orientations in the field of view. The only requirement is that the lines be straight; the lines do not have to be perfectly horizontal or vertical. This method does not involve solving the exterior orientation problem simultaneously. It is easy to make a grid of horizontal and vertical lines using a laser printer. Diagonal lines are rendered less accurately, but since the exterior orientation does not matter, you can shift and rotate the grid to different positions in the field of view, acquire several images, and gather a large set of digitized lines for the calibration set. Mount the grid on a flat, rigid surface that is normal to the optical axis. Since the lines do not have to be parallel, any tilt in the target will not affect the calibration procedure. Determine the location of the edge points to subpixel resolution by computing the first moment over small windows throughout the image. The window size should be somewhat larger than the width of the lines, but smaller than the spacing between lines. The Hough transform can be used to group edges into lines and determine initial estimates for the line parameters.

The equation for each line l in true (corrected) image coordinates is

$$x' \cos\theta_l + y' \sin\theta_l - \rho_l = 0. \tag{12.125}$$

Since the precise position and orientation of each line is unknown, the estimates for the line parameters must be refined as part of the interior orientation problem. Let $(\tilde{x_{kl}}, \tilde{y_{kl}})$ denote the coordinates of edge point k along line l. Replace the true image coordinates (x', y') in Equation 12.125 with the uncorrected coordinates $\tilde{x_{kl}} + \delta x$ and $\tilde{y_{kl}} + \delta y$ using the model for the corrections given above. This yields an equation of the form

$$f(\tilde{x_{kl}}, \tilde{y_{kl}}; x_p, y_p, \kappa_1, \kappa_2, \kappa_3, p_1, p_2, p_3, \rho_l, \theta_l) = 0 \tag{12.126}$$

for each observation (edge point) and the intrinsic parameters. The set of equations for n edge points is a system of n nonlinear equations that must be solved using nonlinear regression. The initial values for corrections to the location of the principal point are zero, and the coefficients for radial lens distortion and decentering can also be initialized to zero. The overall minimization criterion in

$$\chi^2 = \sum_{k=1}^{n} (f(x'_{kl}, y'_{kl}; x_p, y_p, \kappa_1, \kappa_2, \kappa_3, p_1, p_2, p_3, \rho_l, \theta_l))^2. \tag{12.127}$$

This nonlinear regression problem can be solved for the location of the principal point (x_p, y_p); the parameters for the radial lens distortion κ_1, κ_2, and κ_3; and the parameters for the lens decentering p_1, p_2, and p_3. The parameters of each line are estimated as a byproduct of determining the intrinsic parameters and can be discarded.

Example 12.1 *Suppose that a calibration table has been prepared to compensate for lens and camera distortions. The table provides a correction $(\delta x, \delta y)$ for every row and column in the image array. How would this information be used in the overall system presented as an example in Section 12.8.1? After computing the position $(\tilde{x}, \tilde{y})$ of the centroid of the object in the image plane, interpolate the corrections between pixels and add the corrections to the centroid to get correct coordinates for the ray from the camera origin to the object.*

12.10 Camera Calibration

The camera calibration problem is to relate the locations of pixels in the image array to points in the scene. Since each pixel is imaged through perspective projection, it corresponds to a ray of points in the scene. The camera calibration problem is to determine the equation for this ray in the absolute coordinate system of the scene. The camera calibration problem includes both the exterior and interior orientation problems, since the position and orientation of the camera and the camera constant must be determined to relate image plane coordinates to absolute coordinates, and the location of the principal point, the aspect ratio, and lens distortions must be determined to relate image array locations (pixels coordinates) to positions in the image plane. The camera calibration problem involves determining two sets of parameters: the extrinsic parameters for rigid body transformation (exterior orientation) and the intrinsic parameters for the camera itself (interior orientation).

We can use an initial approximation for the intrinsic parameters to get a mapping from image array (pixel) coordinates to image plane coordinates. Suppose that there are n rows and m columns in the image array and assume that the principal point is located at the center of the image array:

$$c_x = \frac{m-1}{2} \tag{12.128}$$

$$c_y = \frac{n-1}{2}. \tag{12.129}$$

The image plane coordinates for the pixel at grid location $[i, j]$ are

$$\tilde{x} = \tau_x d_x (j - c_x) \tag{12.130}$$

$$\tilde{y} = -d_y (i - c_y), \tag{12.131}$$

where d_x and d_y are the center-to-center distances between pixels in the x and y directions, respectively, and τ_x is a scale factor that accounts for distortions in the aspect ratio caused by timing problems in the digitizer electronics. The row and column distances, d_x and d_y, are available from the specifications for the CCD camera and are very accurate, but the scale factor τ_x must be added to the list of intrinsic parameters for the camera and determined through calibration. Note that these are uncorrected image coordinates, marked with a tilde to emphasize that the effects of lens distortions have not been removed. The coordinates are also affected by errors in the estimates for the location of the principal point (c_x, c_y) and the scale factor τ_x.

We must solve the exterior orientation problem before attempting to solve the interior orientation problem, since we must know how the camera is positioned and oriented in order to know where the calibration points project into the image plane. Once we know where the projected points should be, we can use the projected locations p'_i and the measured locations $\tilde{p}_i$ to determine the lens distortions and correct the location of the principal point and the image aspect ratio. The solution to the exterior orientation problem must be based on constraints that are invariant to the lens distortions and camera constant, which will not be known at the time that the problem is solved.

12.10.1 Simple Method for Camera Calibration

This section explains the widely used camera calibration method published by Tsai [234]. Let $\mathbf{p}'_0$ be the location of the origin in the image plane, $\mathbf{r}'_i$ be the vector from $\mathbf{p}'_0$ to the image point $\mathbf{p}'_i = (x'_i, y'_i)$, $\mathbf{p}_i = (x_i, y_i, z_i)$ be a calibration point, and $\mathbf{r}_i$ be the vector from the point $(0, 0, z_i)$ on the optical axis to $\mathbf{p}_i$. If the difference between the uncorrected image coordinates $(\tilde{x}_i, \tilde{y}_i)$ and the true image coordinates (x'_i, y'_i) is due only to radial lens distortion, then $\mathbf{r}'_i$ is parallel to $\mathbf{r}_i$. The camera constant and translation in z do not affect the direction of $\mathbf{r}'_i$, since both image coordinates will be scaled by the same amount. These constraints are sufficient to solve the exterior orientation problem [234].

Assume that the calibration points lie in a plane with $z = 0$ and assume that the camera is placed relative to this plane to satisfy the following two crucial conditions:

1. The origin in absolute coordinates is not in the field of view.

2. The origin in absolute coordinates does not project to a point in the image that is close to the y axis of the image plane coordinate system.

Condition 1 decouples the effects of radial lens distortion from the camera constant and distance to the calibration plane. Condition 2 guarantees that the y component of the rigid body translation, which occurs in the denominator of many equations below, will not be close to zero. These conditions are easy to satisfy in many imaging situations. For example, suppose that the camera is placed above a table, looking down at the middle of the table. The absolute coordinate system can be defined with $z = 0$ corresponding to the plane of the table, with the x and y axes running along the edges of the table, and with the corner of the table that is the origin in absolute coordinates outside of the field of view.

Suppose that there are n calibration points. For each calibration point, we have the absolute coordinates of the point (x_i, y_i, z_i) and the uncorrected image coordinates $(\tilde{x}_i, \tilde{y}_i)$. Use these observations to form a matrix A with rows a_i,

$$a_i = (\tilde{y}_i x_i, \tilde{y}_i y_i, -\tilde{x}_i x_i, -\tilde{x}_i y_i, \tilde{y}_i). \tag{12.132}$$

Let $u = (u_1, u_2, u_3, u_4, u_5)$ be a vector of unknown parameters that are related to the parameters of the rigid body transformation:

$$u_1 = \frac{r_{xx}}{p_y} \tag{12.133}$$

$$u_2 = \frac{r_{xy}}{p_y} \tag{12.134}$$

$$u_3 = \frac{r_{yx}}{p_y} \tag{12.135}$$

$$u_4 = \frac{r_{yy}}{p_y} \tag{12.136}$$

$$u_5 = \frac{p_x}{p_y}. \tag{12.137}$$

Form a vector $\mathbf{b} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ from the n observations of the calibration points. With more than five calibration points, we have an overdetermined system of linear equations,

$$A\mathbf{u} = \mathbf{b}, \tag{12.138}$$

for the parameter vector $\mathbf{u}$. Solve this linear system using singular value decomposition, and use the solution parameters, u_1, u_2, u_3, u_4, and u_5, to compute the rigid body transformation, except for p_z, which scales with the camera constant f and will be determined later.

First, compute the magnitude of the y component of translation. If u_1 and u_2 are not both zero and u_3 and u_4 are not both zero, then

$$p_y^2 = \frac{U - \left[U^2 - 4(u_1u_4 - u_2u_3)^2\right]^{1/2}}{2(u_1u_4 - u_2u_3)^2}, \tag{12.139}$$

where $U = u_1^2 + u_2^2 + u_3^2 + u_4^2$; otherwise, if u_1 and u_2 are both zero, then

$$p_y^2 = \frac{1}{u_3^2 + u_4^2}; \tag{12.140}$$

otherwise, using u_1 and u_2,

$$p_y^2 = \frac{1}{u_1^2 + u_2^2}. \tag{12.141}$$

Second, determine the sign of p_y. Pick the calibration point $\mathbf{p} = (x, y, z)$ that projects to an image point that is farthest from the center of the image (the scene point and corresponding image point that are farthest in the periphery of the field of view). Compute r_{xx}, r_{xy}, r_{yx}, r_{yy}, and p_x from the solution vector obtained above:

$$r_{xx} = u_1p_y \tag{12.142}$$

$$r_{xy} = u_2p_y \tag{12.143}$$

$$r_{yx} = u_3p_y \tag{12.144}$$

$$r_{yy} = u_4p_y \tag{12.145}$$

$$p_x = u_5p_y. \tag{12.146}$$

Let $\xi_x = r_{xx}x + r_{xy}y + p_x$ and $\xi_y = r_{yx}x + r_{yy}y + p_y$. If ξ_x and $\tilde{x}$ have the same sign and ξ_y and $\tilde{y}$ have the same sign, then p_y has the correct sign (positive); otherwise, negate p_y. Note that the parameters of the rigid body transformation computed above are correct, regardless of the sign of p_y, and do not need to be changed.

Third, compute the remaining parameters of the rigid body transformation:

$$r_{xz} = \sqrt{1 - r_{xx}^2 - r_{xy}^2} \tag{12.147}$$

$$r_{yz} = \sqrt{1 - r_{yx}^2 - r_{yy}^2}. \tag{12.148}$$

Since the rotation matrix must be orthonormal, it must be true that $R^T R = I$. Use this fact to compute the elements in the last row of the rotation matrix:

$$r_{zx} = \frac{1 - r_{xx}^2 - r_{xy}r_{yx}}{r_{xz}} \tag{12.149}$$

$$r_{zy} = \frac{1 - r_{yx}r_{xy} - r_{yy}^2}{r_{yz}} \tag{12.150}$$

$$r_{zz} = \sqrt{1 - r_{zx}r_{xz} - r_{zy}r_{yz}}. \tag{12.151}$$

If the sign of $r_{xx}r_{yx} + r_{xy}r_{yy}$ is positive, negate r_{yz}. The signs of r_{zx} and r_{zy} may need to be adjusted after computing the camera constant in the following step.

Fourth, compute the camera constant f and p_z, the z component of translation. Use all of the calibration points to form a system of linear equations,

$$A\mathbf{v} = \mathbf{b}, \tag{12.152}$$

for estimating f and p_z. Use each calibration point to compute the corresponding row of the matrix,

$$a_i = (r_{yx}x_i + r_{yy}y_i + p_y, -d_y\tilde{y}_i), \tag{12.153}$$

and the corresponding element of the vector on the right side of Equation 12.152,

$$b_i = (r_{zx}x_i + r_{zy}y_i)d_y\tilde{y}_i. \tag{12.154}$$

The vector $\mathbf{v}$ contains the parameters to be estimated:

$$\mathbf{v} = (f, p_z)^T. \tag{12.155}$$

Use singular value decomposition to solve this system of equations. If the camera constant $f < 0$, then negate r_{zx} and r_{zy} in the rotation matrix for the rigid body transformation.

Fifth, use the estimates for f and p_z obtained in the previous step as the initial conditions for nonlinear regression to compute the first-order lens distortion κ_1 and better estimates for f and p_z. The true (corrected) image plane coordinates (x', y') are related to the calibration points in camera coordinates (x_c, y_c, z_c) through perspective projection:

$$x' = f\frac{x_c}{z_c} \tag{12.156}$$

$$y' = f\frac{y_c}{z_c}. \tag{12.157}$$

Assume that the true (corrected) image plane coordinates are related to the measured (uncorrected) image plane coordinates using the first term in the model for radial lens distortion:

$$x' = \tilde{x}(1 + \kappa_1 r^2) \tag{12.158}$$

$$y' = \tilde{y}(1 + \kappa_1 r^2), \tag{12.159}$$

where the radius r is given by

$$r = \sqrt{\tilde{x}^2 + \tilde{y}^2}. \tag{12.160}$$

Note that the uncorrected (measured) image plane coordinates $(\tilde{x}, \tilde{y})$ are not the same as the pixel coordinates $[i, j]$ since the location of the image center (c_x, c_y), the row and column spacing d_x and d_y, and the estimated scale factor τ_x have already been applied.

Use the y components of the equations for perspective projection, lens distortion, and the rigid body transformation from absolute coordinates to camera coordinates to get a constraint on the camera constant f, z translation, and lens distortion:

$$\tilde{y}_i(1 + \kappa_1 r^2) = f\,\frac{r_{yx}x_{a,i} + r_{yy}y_{a,i} + r_{yz}z_{a,i} + p_y}{r_{zx}x_{a,i} + r_{zy}y_{a,i} + r_{zz}z_{a,i} + p_z}. \tag{12.161}$$

This leads to a nonlinear regression problem for the parameters p_z, f, and κ_1. We use the measurements for y, rather than x, because the x measurements are affected by the scale parameter τ_x. The spacing between image rows d_y is very accurate and readily available from the camera specifications and is not affected by problems in the digitizing electronics.

Since the calibration points were in a plane, the scale factor τ_x cannot be determined. Also, the location of the image center, c_x and c_y, has not been calibrated. The list of further readings provided at the end of this chapter provides references to these calibration problems.

12.10.2 Affine Method for Camera Calibration

The interior orientation problem can be combined with the exterior orientation problem to obtain an overall transformation that relates (uncalibrated) image coordinates to the position and orientation of rays in the absolute coordinate system. Assume that the transformation from uncorrected image coordinates to true image coordinates can be modeled by an affine transformation within the image plane. This transformation accounts for several sources of camera error:

Scale error due to an inaccurate value for the camera constant

Translation error due to an inaccurate estimate for the image origin (principal point)

Rotation of the image sensor about the optical axis

Skew error due to nonorthogonal camera axes

Differential scaling caused by unequal spacing between rows and columns in the image sensor (nonsquare pixels)

However, an affine transformation cannot model the errors due to lens distortions.

In the development of the exterior orientation problem (Section 12.8), we formulated equations for the transformation from absolute coordinates to image coordinates. Now we will add an affine transformation from true image coordinates to measured (uncorrected) image coordinates to get the overall transformation from absolute coordinates to measured image coordinates.

The affine transformation in the image plane that models the distortions due to errors and unknowns in the intrinsic parameters is

$$\tilde{x} = a_{xx}x' + a_{xy}y' + b_x \tag{12.162}$$

$$\tilde{y} = a_{yx}x' + a_{yy}y' + b_y, \tag{12.163}$$

where we are mapping from true image plane coordinates (x', y') to uncorrected (measured) image coordinates $(\tilde{x}, \tilde{y})$. Use the equations for perspective projection,

$$\frac{x'}{f} = \frac{x_c}{z_c} \tag{12.164}$$

$$\frac{y'}{f} = \frac{y_c}{z_c}, \tag{12.165}$$

to replace x' and y' with ratios of the camera coordinates:

$$\frac{\tilde{x}}{f} = a_{xx}\left(\frac{x_c}{z_c}\right) + a_{xy}\left(\frac{y_c}{z_c}\right) + \frac{b_x}{f} \tag{12.166}$$

$$\frac{\tilde{y}}{f} = a_{yx}\left(\frac{x_c}{z_c}\right) + a_{yy}\left(\frac{y_c}{z_c}\right) + \frac{b_y}{f}. \tag{12.167}$$

Camera coodinates are related to absolute coordinates by a rigid body transformation:

$$x_c = r_{xx}x_a + r_{xy}y_a + r_{xz}z_a + p_x \tag{12.168}$$

$$y_c = r_{yx}x_a + r_{yy}y_a + r_{yz}z_a + p_y \tag{12.169}$$

$$z_c = r_{zx}x_a + r_{zy}y_a + r_{zz}z_a + p_z. \tag{12.170}$$

We can use these equations to replace the ratios of camera coordinates in the affine transformation with expressions for the absolute coordinates,

$$\frac{\tilde{x} - b_x}{f} = \frac{s_{xx}x_a + s_{xy}y_a + s_{xz}z_a + t_x}{s_{zx}x_a + s_{zy}y_a + s_{zz}z_a + t_z} \tag{12.171}$$

$$\frac{\tilde{y} - b_y}{f} = \frac{s_{yx}x_a + s_{yy}y_a + s_{yz}z_a + t_y}{s_{zx}x_a + s_{zy}y_a + s_{zz}z_a + t_z}, \tag{12.172}$$

where the coefficients are sums of products of the coefficients in the affine transformation and the rigid body transformation. What we have is a pair of equations, similar to the equations for exterior orientation (Equations 12.108 and 12.109), that relate absolute coordinates to uncorrected image coordinates. The affine model for camera errors has been absorbed into the transformation from absolute to camera coordinates. Equations 12.171 and 12.172 can be written as

$$\frac{\tilde{x} - b_x}{f} = \frac{\tilde{x}_c}{\tilde{z}_c} = \frac{s_{xx}x_a + s_{xy}y_a + s_{xz}z_a + t_x}{s_{zx}x_a + s_{zy}y_a + s_{zz}z_a + t_z} \tag{12.173}$$

$$\frac{\tilde{y} - b_y}{f} = \frac{\tilde{y}_c}{\tilde{z}_c} = \frac{s_{yx}x_a + s_{yy}y_a + s_{yz}z_a + t_y}{s_{zx}x_a + s_{zy}y_a + s_{zz}z_a + t_z} \tag{12.174}$$

to show that the (uncorrected) image coordinates are related to the camera coordinates by perspective projection, but the space of camera coordinates has been warped to account for the camera errors.

Returning to Equations 12.171 and 12.172, we can absorb the corrections to the location of the principal point, b_x and b_y, into the affine transformation to get

$$\tilde{x}_i(s_{zx}x_{a,i}+s_{zy}y_{a,i}+s_{zz}z_{a,i}+t_z) - f(s_{xx}x_{a,i}+s_{xy}y_{a,i}+s_{xz}z_{a,i}+t_x) = 0 \quad (12.175)$$
$$\tilde{y}_i(s_{zx}x_{a,i}+s_{zy}y_{a,i}+s_{zz}z_{a,i}+t_z) - f(s_{yx}x_{a,i}+s_{yy}y_{a,i}+s_{yz}z_{a,i}+t_y) = 0, \quad (12.176)$$

which shows that each calibration point and its corresponding measured location in the image plane provides two linear equations for the parameters of the transformation. The nominal value f for the camera constant is not absorbed into the affine transformation since it is needed for constructing the ray in camera coordinates.

The set of calibration points yields a set of homogeneous linear equations that can be solved for the coefficients of the transformation. At least six points are needed to get 12 equations for the 12 unknowns, but more calibration points should be used to increase accuracy. To avoid the trivial solution with all coefficients equal to zero, fix the value of one of the parameters, such as t_x or t_y, and move it to the right side of the equation. Form a system of linear equations,

$$A\mathbf{u} = \mathbf{b}, \quad (12.177)$$

where $\mathbf{u}$ is the vector of transformation coefficients; row i of the A matrix is filled with absolute coordinates for calibration point i and products of the absolute coordinates and $\tilde{x}_i$, $\tilde{y}_i$, or f; and element i of the $\mathbf{b}$ vector is the constant chosen for t_x or t_y. Since the affine transformation within the image plane is combined with the rotation matrix for exterior orientation, the transformation matrix is no longer orthonormal. The system of linear equations can be solved, without the orthonormality constraints, using common numerical methods such as singular value decomposition.

The transformation maps absolute coordinates to measured image coordinates. Applications require the inverse transformation, given by

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = S^{-1}\left[\begin{pmatrix} \tilde{x}_i \\ \tilde{y}_i \\ f \end{pmatrix} - \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}\right], \quad (12.178)$$

which can be used to determine the equation of a ray in absolute coordinates from the measured coordinates in the image. Note that the camera constant f has been carried unchanged through the formulation of the calibration algorithm. Since corrections to the camera constant are included in the affine transformation (Equations 12.162 and 12.163), the focal length of the lens can be used for f. Finally, the transformation from pixel coordinates $[i, j]$ to image coordinates,

$$\tilde{x} = s_x(j - c_x) \tag{12.179}$$

$$\tilde{y} = -s_y(i - c_y), \tag{12.180}$$

is an affine transformation that can be combined with the model for camera errors (Equations 12.162 and 12.163) to develop a transformation between absolute coordinates and pixel coordinates.

12.10.3 Nonlinear Method for Camera Calibration

Given a set of calibration points, determine the projections of the calibration points in the image plane, calculate the errors in the projected positions, and use these errors to solve for the camera calibration parameters. Since it is necessary to know where the calibration points should project to in the image plane, the exterior orientation problem is solved simultaneously. The method presented in this section is different from the procedure explained in Section 12.10.2, where the interior and exterior orientation problems were combined into a single affine transformation, in that the actual camera calibration parameters are obtained and can be used regardless of where the camera is later located in scene.

The principle behind the solution to the camera calibration problem is to measure the locations (x'_i, y'_i) of the projections of the calibration points onto the image plane, calculate the deviations $(\delta x_i, \delta y_i)$ of the points from the correct positions, and plug these measurements into the equations that model the camera parameters. Each calibration point yields two equations. The solution requires at least enough equations to cover the unknowns, but for increased accuracy more equations than unknowns are used and the overdetermined set of equations is solved using nonlinear regression.

Assume that the approximate position and orientation of the camera in absolute coordinates is known. Since we have initial estimates for the rotation angles, we can formulate the exterior orientation problem in terms of the Euler angles in the rotation matrix. The parameters of the regression problem are the rotation angles ω, ϕ, and κ; the position of the camera in absolute coordinates p_x, p_y, and p_z; the camera constant f; the corrections to the location of the principal point (x_p, y_p); and the polynomial coefficients for radial lens distortion κ_1, κ_2, and κ_3. The equations for the exterior orientation problem are

$$\frac{x'}{f} = \frac{r_{xx}x_a + r_{xy}y_a + r_{xz}z_a + p_x}{r_{zx}x_a + r_{zy}y_a + r_{zz}z_a + p_z} \tag{12.181}$$

$$\frac{y'}{f} = \frac{r_{yx}x_a + r_{yy}y_a + r_{yz}z_a + p_y}{r_{zx}x_a + r_{zy}y_a + r_{zz}z_a + p_z}. \tag{12.182}$$

Replace x' and y' with the corrected positions from the camera model,

$$\frac{(\tilde{x}-x_p)(1+\kappa_1 r^2+\kappa_2 r^4+\kappa_3 r^6)}{f} = \frac{r_{xx}x_a + r_{xy}y_a + r_{xz}z_a + p_x}{r_{zx}x_a + r_{zy}y_a + r_{zz}z_a + p_z} \tag{12.183}$$

$$\frac{(\tilde{y}-y_p)(1+\kappa_1 r^2+\kappa_2 r^4+\kappa_3 r^6)}{f} = \frac{r_{yx}x_a + r_{yy}y_a + r_{yz}z_a + p_y}{r_{zx}x_a + r_{zy}y_a + r_{zz}z_a + p_z}, \tag{12.184}$$

and replace the elements of the rotation matrix with the formulas for the rotation matrix entries in terms of the Euler angles, provided in Equation 12.13. Solve for the camera parameters and exterior orientation using nonlinear regression. The regression algorithm will require good initial conditions. If the target is a plane, the camera axis is normal to the plane, and the image is roughly centered on the target, then the initial conditions are easy to obtain. Assume that the absolute coordinate system is set up so that the x and y axes are parallel to the camera axes. The initial conditions are:

$$\begin{aligned}
\omega &= \phi = \kappa = 0 \\
x &= \text{translation in } x \text{ from the origin} \\
y &= \text{translation in } y \text{ from the origin} \\
z &= \text{distance of the camera from the calibration plane} \\
f &= \text{focal length of the lens} \\
x_p &= y_p = 0 \\
\kappa_1 &= \kappa_2 = \kappa_3 = 0.
\end{aligned}$$

It is easy to build a target of dots using a laser printer. The uncorrected positions of the dots in the image can be found by computing the first moments of the connected components.

The disadvantage to nonlinear regression is that good initial values for the parameters are needed, but the advantage is that there is a body of literature on nonlinear regression with advice on solving nonlinear problems and methods for estimating errors in the parameter estimates.

12.11 Binocular Stereo Calibration

In this section we will discuss how the techniques presented in this chapter can be combined in a practical system for calibrating stereo cameras and using the stereo measurements. This provides a forum for reviewing the relationships between the various calibration problems.

There are several tasks in developing a practical system for binocular stereo:

1. Calibrate the intrinsic parameters for each camera.

2. Solve the relative orientation problem.

3. Resample the images so that the epipolar lines correspond to image rows.

4. Compute conjugate pairs by feature matching or correlation.

5. Solve the stereo intersection problem for each conjugate pair.

6. Determine baseline distance.

7. Solve the absolute orientation problem to transform point measurements from the coordinate system of the stereo cameras to an absolute coordinate system for the scene.

There are several ways to calibrate a binocular stereo system, corresponding to various paths through the diagram in Figure 12.6. To start, each camera must be calibrated to determine the camera constant, location of the

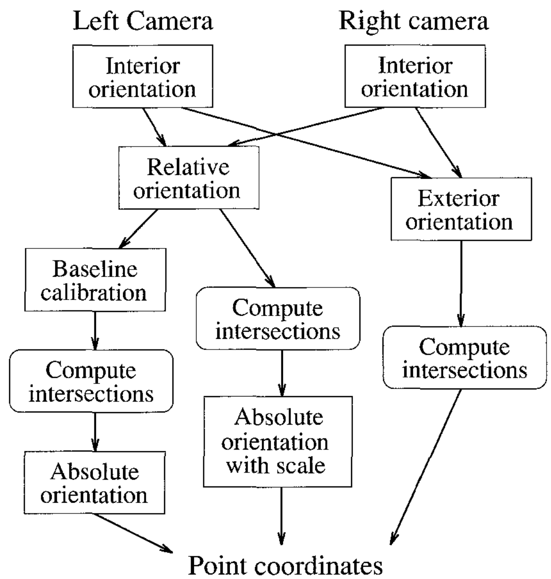

Figure 12.6: A diagram of the steps in various procedures for calibrating a binocular stereo system.

principal point, correction table for lens distortions, and other intrinsic parameters. Once the left and right stereo cameras have been calibrated, there are basically three approaches to using the cameras in a stereo system.

The first approach is to solve the relative orientation problem and determine the baseline by other means, such as using the stereo cameras to measure points that are a known distance apart. This fully calibrates the rigid body transformation between the two cameras. Point measurements can be gathered in the local coordinate system of the stereo cameras. Since the baseline has been calibrated, the point measurements will be in real units and the stereo system can be used to measure the relationships between points on objects in the scene. It is not necessary to solve the absolute orientation problem, unless the point measurements must be transformed into another coordinate system.

The second approach is to solve the relative orientation problem and obtain point measurements in the arbitrary system of measurement that results from assuming unit baseline distance. The point measurements will be correct, except for the unknown scale factor. Distance ratios and angles will be correct, even though the distances are in unknown units. If the baseline distance is obtained later, then the point coordinates can be multiplied by the baseline distance to get point measurements in known units. If it is necessary to transform the point measurements into another coordinate system, then solve the absolute orientation problem with scale (Section 12.7), since this will accomplish the calibration of the baseline distance and the conversion of point coordinates into known units without additional computation.

The third approach is to solve the exterior orientation problem for each stereo camera. This provides the transformation from the coordinate systems of the left and right camera into absolute coordinates. The point measurements obtained by intersecting rays using the methods of Section 12.6 will automatically be in absolute coordinates with known units, and no further transformations are necessary.

12.12 Active Triangulation

This section will cover methods for determining the coordinates of a point using an active sensor that projects a plane of light onto opaque surfaces in the scene. A method for calibrating an active triangulation system will be presented.

We will start with a simple geometry in camera-centered coordinates and proceed to the general case in absolute coordinates. Suppose that the plane of light rotates about an axis that is parallel to the y axis and displaced along the x axis by b_x. Let θ be the orientation of the plane relative to the z axis. With $\theta = 0$, the plane of light is parallel to the y–z plane and positive values for θ correspond to counterclockwise rotation about the y axis. In terms of vector geometry, the normal to the plane is

$$\mathbf{n} = (n_x, n_y, n_z) = (\cos(\theta), 0, \sin(\theta)), \tag{12.185}$$

the baseline is

$$\mathbf{b} = (b_x, b_y, b_z) = (b_x, 0, 0), \tag{12.186}$$

and point $\mathbf{p} = (x, y, z)$ lies in the plane if

$$(\mathbf{p} - \mathbf{b}) \cdot \mathbf{n} = 0. \tag{12.187}$$

The plane of light illuminates the scene and intersects an opaque surface to produce a curve in space that is imaged by the camera. A line detection operator is used to estimate the locations of points along the projected curve in the image plane. Suppose that an estimated line point has coordinates (x', y'). This corresponds to a ray in space represented by the equation

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = t \begin{pmatrix} x' \\ y' \\ f \end{pmatrix}, \tag{12.188}$$

where f is the distance of the image plane from the center of projection. Replace $\mathbf{p}$ in Equation 12.187 with the equation for the ray and solve for t,

$$t = \frac{b_x \cos\theta}{x' \cos\theta - f \sin\theta}, \tag{12.189}$$

and plug into the equation for the ray to get the coordinates of the point in camera-centered coordinates:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{x' b_x \cos\theta}{x' \cos\theta - f \sin\theta} \\ \frac{y' b_x \cos\theta}{x' \cos\theta - f \sin\theta} \\ \frac{f\, b_x \cos\theta}{x' \cos\theta - f \sin\theta} \end{pmatrix}. \tag{12.190}$$

If the exterior orientation of the camera has been calibrated, then the ray can be represented in absolute coordinates; if the position and orientation of the plane are also represented in absolute coordinates, then the point measurements will be in absolute coordinates.

It is easy to generalize these equations to allow an arbitrary position and orientation for the plane in space. Suppose that the plane is rotated by θ counterclockwise about an axis ω and the normal $\mathbf{n}$ corresponds to the orientation with $\theta = 0$. Let $R(\theta)$ be the rotation matrix. A point $\mathbf{p}$ lies in the plane if

$$(\mathbf{p} - \mathbf{b}) \cdot R(\theta)\mathbf{n} = 0. \tag{12.191}$$

We can also change the position of the plane in space by varying $\mathbf{b}$,

$$(\mathbf{p} - \mathbf{b}(d)) \cdot R(\theta)\mathbf{n} = 0, \tag{12.192}$$

where d is the control parameter of a linear actuator.

12.13 Robust Methods

All of the calibration methods presented in this chapter have used least-squares regression, which is very sensitive to outliers due to mismatches in forming the conjugate pairs. There are two approaches to making calibration robust: use a resampling plan or change the regression procedure to use a robust norm.

To implement a resampling plan, it is necessary to consider all combinations of m conjugate pairs from the set of n conjugate pairs obtained from the calibration data. The size m of the subset must be large enough to get a good solution to the calibration problem. Choose the best set of parameters by comparing the fit of the transformation to all conjugate pairs according to the least median of squares criterion [207]. Use the best parameter estimates to remove outliers from the set of conjugate pairs, and repeat the calibration procedure using all of the remaining conjugate pairs.

The other approach involves replacing the square norm with a robust norm [197, pp. 558–565]. If the calibration procedure uses linear regression, then use a weighted least-squares solution method. Calculate the weights so that the weighted least-squares regression problem is equivalent to the unweighted robust regression problem. If the robust norm for residual r_i is $\rho(r_i)$, then solve

$$w_i r_i^2 = \rho(r_i) \tag{12.193}$$

for the weight w_i and use this weight for the conjugate pair. This leads to iterative reweighted least-squares: a sequence of regression problems is solved with the weights adjusted between iterations. For nonlinear regression, the solution method may allow the robust norm to be used directly.

12.14 Conclusions

Several methods for calibration have been presented in this chapter, including the basic problems in photogrammetry: absolute, relative, exterior, and interior orientation. The interior orientation problem should be solved for any camera to ensure that the camera obeys the assumptions of image formation assumed by most machine vision algorithms. The remaining calibration problems can be divided into two groups: methods used in image analysis and methods used in depth measurement. The exterior orientation problem must be solved for an image analysis application when it is necessary to relate image measurements to the geometry of the scene. The relative orientation problem is used to calibrate a pair of cameras for obtaining depth measurements with binocular stereo. The absolute orientation problem is used to calibrate the position and orientation of any system for depth measurement, including binocular stereo or active sensing, so that the depth measurements in camera coordinates can be translated into the coordinate system used in the application.

Further Reading

An excellent introduction to photogrammetry is provided by Horn [109], who has also published a closed-form solution to the absolute orientation problem using orthonormal matrices [113] and unit quaternions [110]. The American Society of Photogrammetry publishes a comprehensive collection of articles on photogrammetry [224], and there are several books on photogrammetry [17, 95, 170, 254]. Photogrammetry was originally developed for preparing topographic maps from aerial photographs—hence the name, which means making measurements in photographs. Terrestrial photogrammetry is covered in the *Handbook of Non-Topographic Photogrammetry* and includes methods of use in machine vision [137]. The camera calibration technique developed by Tsai [234] is widely used. Further work on camera calibration has been done by Lenz and Tsai [154]. Horn [111] published an excellent description of the solution to the relative orientation problem. Brown [51] published the algorithm for correcting radial lens distortions and decentering.

Photogrammetry usually assumes that objects and cameras move by rigid body transformations. In many cases, an object will deform as it moves, in addition to undergoing translation, rotation, and change in scale. Bookstein has published an extensive body of literature on modeling nonrigid deformations in diverse fields using thin-plate splines [39, 40, 41, 42].

Exercises

12.1 How many types of orientations are possible in calibration? Consider some examples and illustrate the difference among them.

12.2 Why is photogrammetry concerned with the camera calibration problem? Illustrate using an example.

12.3 What is the effect of rectangular pixels on image measurements? How can you compensate for the rectangularity of pixels in measuring areas and centroids of regions? How will it affect measurements of relative locations and orientations of regions? What care would you take to get correct measurements in the real world?

12.4 Define affine transformation. Give three examples of objects that will undergo an affine transformation in an image at different time instants and three examples of objects that will not.

12.5 Define Euler angles. Where are they used? What are their strengths and weaknesses?

12.6 What are quaternions? Why are they considered a good representation for rotation in calibration? Demonstrate this considering the calibration of absolute orientation.

12.7 Define the coplanarity constraint. Where and how is it used in camera calibration?

12.8 Suppose that you are designing a hand–eye system. In this system a camera is mounted at a fixed position in the work space. The hand, a robot arm, is used to pick and place objects in the work space. What kind of calibration scheme will you require?

12.9 In the above problem, if the camera is mounted on the robot arm itself, what kind of calibration scheme will you require to find the location of objects? How many fixed known points will you require to solve the problem?

12.10 Now let's provide a stereo system on our robot arm so that we can compute the depth of points easily and perform more dexterous manipulations. What kind of calibration scheme is required in this case? What are the criteria for selecting scene points for calibration? What arrangements of points are bad for calibration?

12.11 What camera parameters should be determined for calibrating a camera? How can you determine these parameters without directly measuring them? What parameters are available from the data sheets provided by the camera manufacturer?

Computer Projects

12.1 You want to develop a camera calibration algorithm for preparing a three-dimensional model of a football field using lane markers. Assume that a sufficient number of lane markers, their intersections, and other similar features are visible in the field of view. Develop an approach to solve the camera calibration problem and use it to determine the location of each player, their height, and the location of the ball.

12.2 Develop an algorithm for the calibration of stereo cameras. Use this algorithm to calibrate a set of cameras mounted on a mobile robot. Develop an approach that will use known calibration points in an environment to determine the camera calibration and then use this to get the exact distance to all points.