7

Stereopsis

> Two are better than one;
> because they have a good reward for their labor.
>
> *Ecclesiastes* 4:9

This chapter is an introduction to *stereopsis*, or simply *stereo*, in computer vision.

Chapter Overview

Section 7.1 is an informal introduction; it subdivides stereo in two subproblems, *correspondence* and *reconstruction*, and analyzes a simple stereo system.

Section 7.2 deals with the problem of establishing correspondences between image elements of a stereo pair.

Section 7.3 is dedicated to the geometry of stereo, the *epipolar geometry*, and how it can be recovered and used.

Section 7.4 discusses three methods for the reconstruction of the 3-D structure of a scene, each assuming a different amount of knowledge on the intrinsic and extrinsic parameters of the cameras.

What You Need to Know to Understand this Chapter

- Working knowledge of Chapters 2, 4, 5, 6.
- Singular value decomposition and constrained optimization (Appendix, section A.6).
- Basic notions of projective geometry (Appendix, section A.4).

7.1 Introduction

Stereo vision refers to the ability to *infer information on the 3-D structure and distance of a scene from two or more images taken from different viewpoints*. You can learn a great deal of the basic principles (and problems) of a stereo system through a simple experiment. Hold one thumb at arm's length and close the right and left eye alternatively. What do you expect to see? With presumably little surprise you find that the relative position of thumb and background appears to change, depending on which eye is open (or closed).[1] It is precisely this difference in retinal location that is used by the brain to reconstruct a 3-D representation of what we see.

7.1.1 The Two Problems of Stereo

From a computational standpoint, a stereo system must solve two problems. The first, known as *correspondence*, consists in determining *which item in the left eye corresponds to which item in the right eye* (Figure 7.1). A rather subtle difficulty here is that some parts of the scene are visible by one eye only. In the thumb experiment, for example, which part of the background is occluded by your thumb depends on which eye is open. Therefore, a stereo system must also be able to determine the image parts that should *not* be matched.

The second problem that a stereo system must solve is *reconstruction*. Our vivid 3-D perception of the world is due to the interpretation that the brain gives of the computed difference in retinal position, named *disparity*, between corresponding items.[2] The disparities of all the image points form the so-called *disparity map*, which can be displayed as an image. If the geometry of the stereo system is known, the disparity map can be converted to a 3-D map of the viewed scene (the reconstruction). Figure 7.2 shows an example of stereo reconstruction with a human face. Figure 7.3 illustrates an application of computational stereopsis in space research: reconstructing the relief of the surface of Venus from two satellite (SAR) images. The images were recorded by the Magellan spacecraft, and cover an area of approximately 120 × 40 km; each pixel corresponds to 75 m.

Definitions: Stereo Correspondence and Reconstruction

The correspondence problem: Which parts of the left and right images are projections of the same scene element?

The reconstruction problem: Given a number of corresponding parts of the left and right image, and possibly information on the geometry of the stereo system, what can we say about the 3-D location and structure of the observed objects?

[1] If you think this is obvious, try to explain why, with both eyes open, you almost invariably see only *one* thumb, well separated in depth from the background.

[2] This is best demonstrated by the now popular *autostereograms*, in which the perception of depth is induced by no cue other than disparity. See the Further Readings for more on autostereograms.

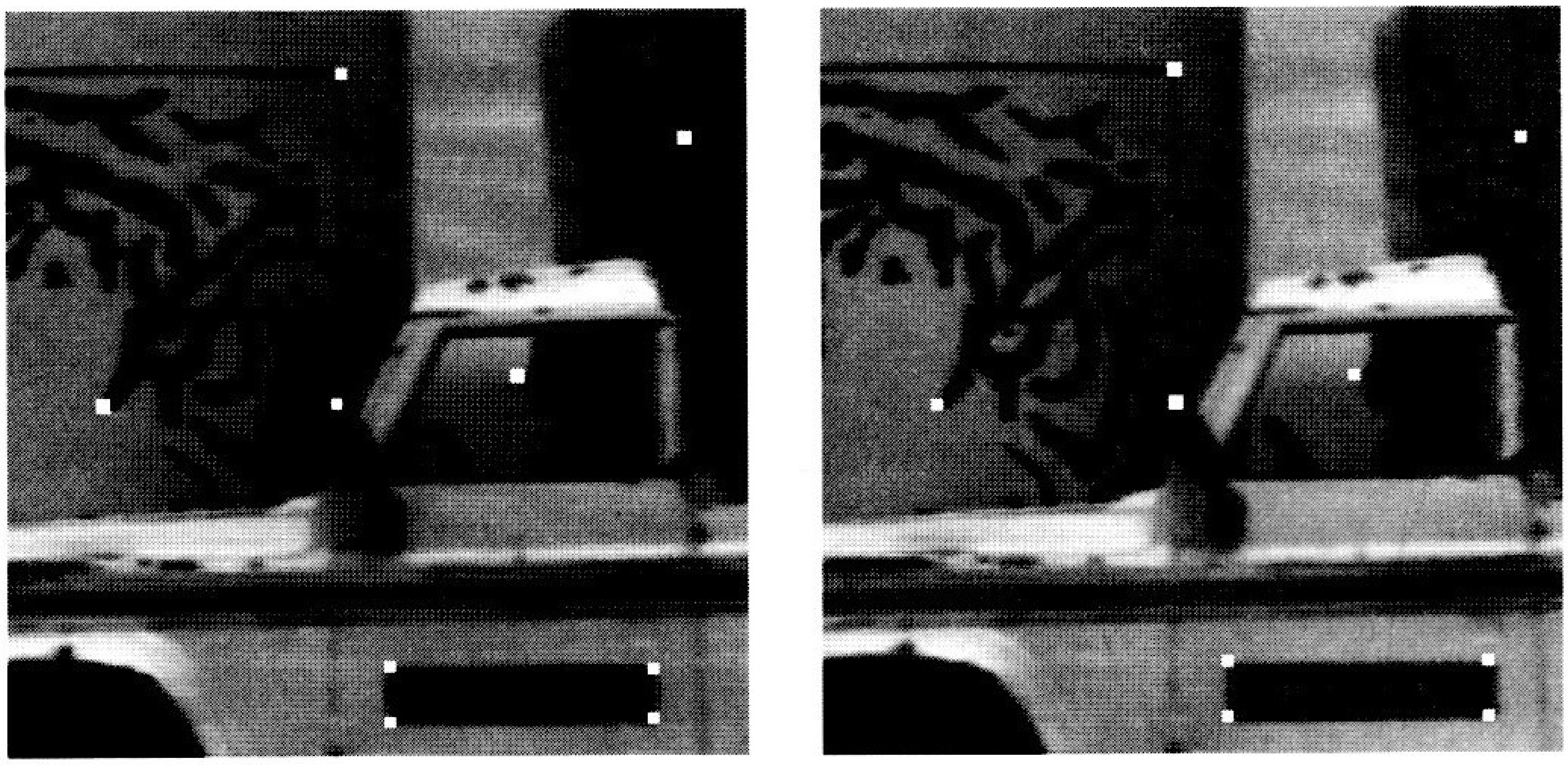


Figure 7.1 An illustration of the correspondence problem. A matching between corresponding points of an image pair is established (only some correspondences are shown).

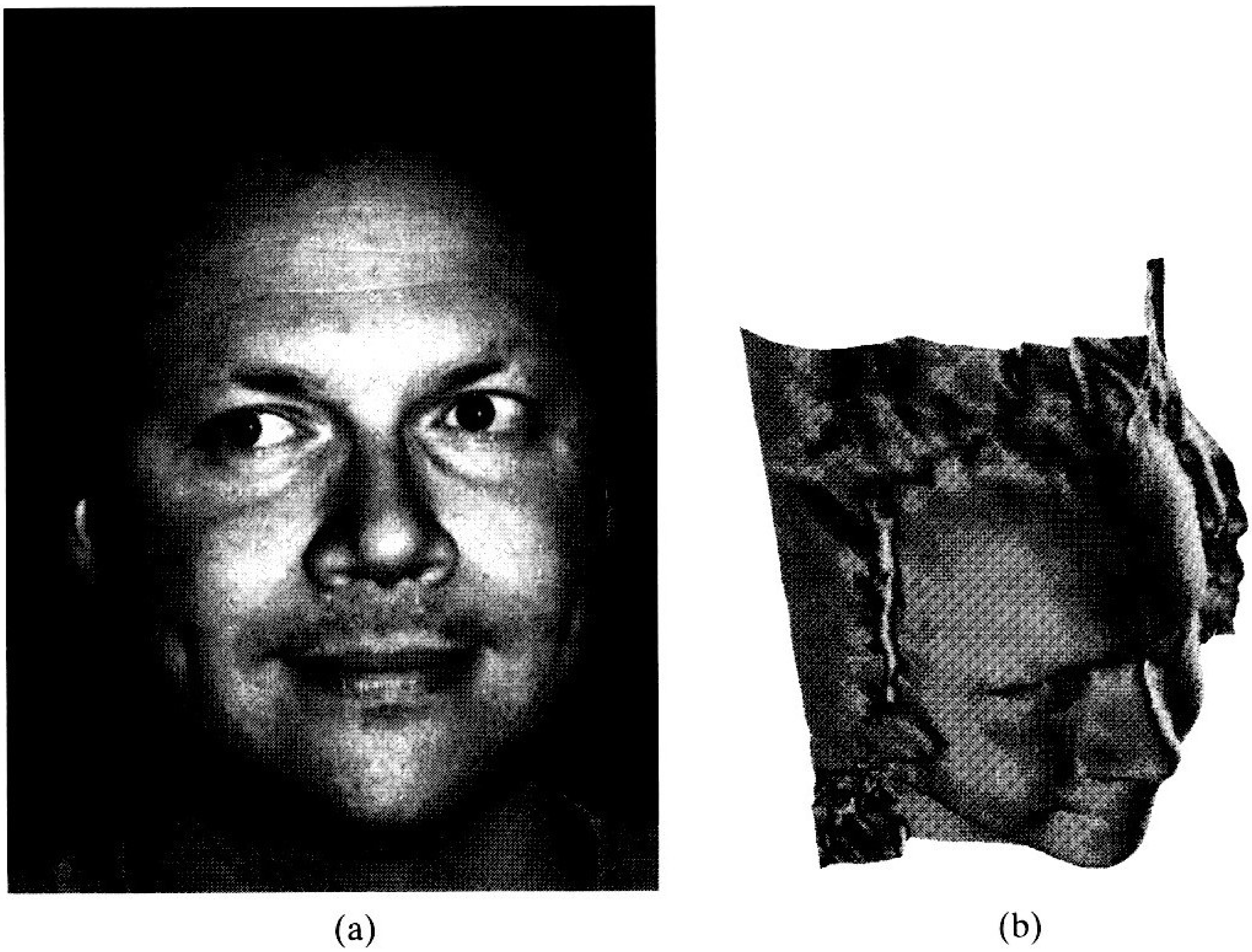


Figure 7.2 (a) One image from a stereo pair of Emanuele Trucco's face. (b) 3-D rendering of stereo reconstruction. Courtesy of the Turing Institute, Glasgow (UK).

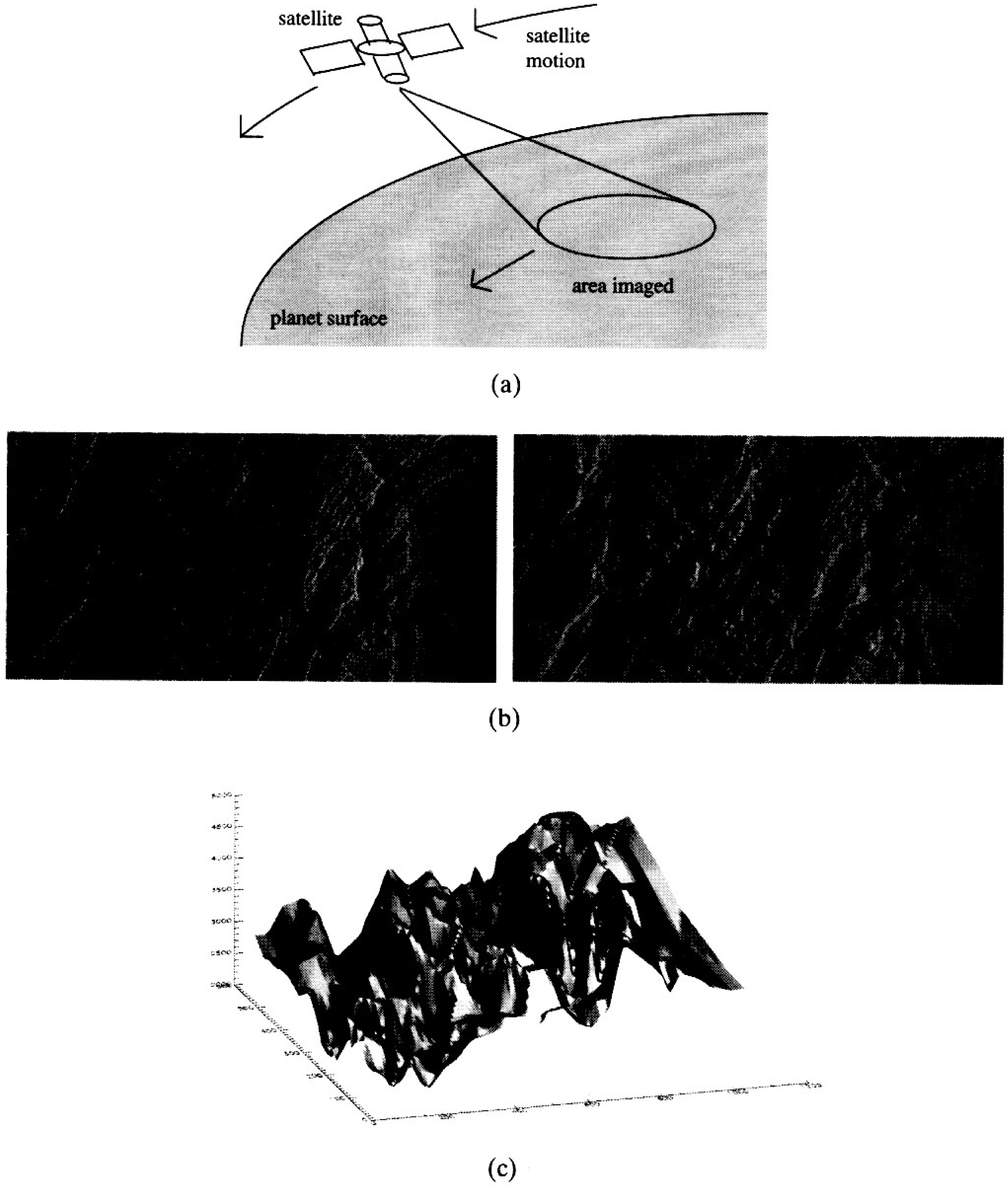
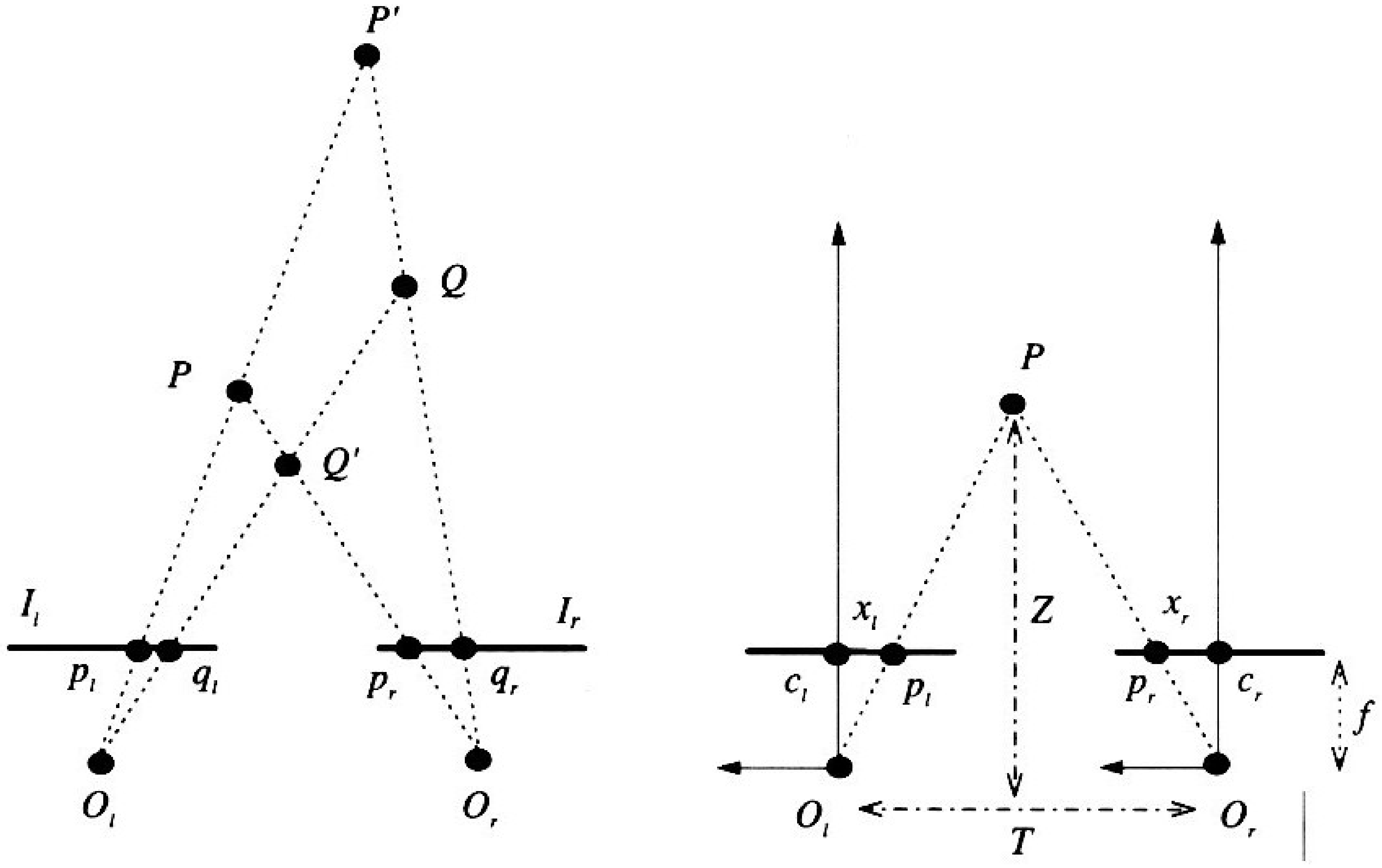


Figure 7.3 Stereo reconstruction of the surface of Venus from a pair of SAR images. (a) Illustration of the application, showing the satellite orbiting around the planet. (b) The stereo pair of the surface of Venus acquired by the Magellan satellite. (c) 3-D rendering of the reconstructed surface. Courtesy of Alois Goller, Institute for Computer Graphics and Vision, Technical University of Graz.



(a) (b)

Figure 7.4 A simple stereo system. 3-D reconstruction depends on the solution of the correspondence problem (a); depth is estimated from the disparity of corresponding points (b).

7.1.2 A Simple Stereo System

Before starting our investigation of the correspondence and reconstruction problems with the necessary mathematical machinery, it is useful to learn as much as we can from the very simple model illustrated in Figure 7.4(a). The diagram shows the top view of a stereo system composed of two pinhole cameras. The left and right image planes are coplanar, and represented by the segments I_l and I_r respectively. O_l and O_r are the centers of projection. The optical axes are parallel; for this reason, the *fixation point*, defined as the point of intersection of the optical axes, lies infinitely far from the cameras.

The way in which stereo determines the position in space of P and Q (Figure 7.4(a)) is *triangulation*, that is, by intersecting the rays defined by the centers of projection and the images of P and Q, p_l, p_r, q_l, q_r. Triangulation depends crucially on the solution of the correspondence problem: If (p_l, p_r) and (q_l, q_r) are chosen as pairs of corresponding image points, intersecting the rays $O_l p_l - O_r p_r$ and $O_l q_l - O_r q_r$ leads to interpreting the image points as projections of P and Q; but, if (p_l, q_r) and (q_l, p_r) are the selected pairs of corresponding points, triangulation returns P' and Q'. Note that both interpretations, although dramatically different, stand on an equal footing once we accept the respective correspondences. We will have more to say about the correspondence problem and its solutions in Section 7.2.

Let us now assume that the correspondence problem has been solved, and turn to reconstruction. It is instructive to write the equation underlying the triangulation of Figure 7.4. We concentrate on the recovery of the position of a single point, P, from its projections, p_l and p_r (Figure 7.4(b)). The distance, T, between the centers of projection O_l and O_r, is called the *baseline* of the stereo system. Let x_l and x_r be the coordinates of p_l and p_r with respect to the principal points c_l and c_r, f the common focal length, and Z the distance between P and the baseline. From the similar triangles (p_l, P, p_r) and (O_l, P, O_r) we have

$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}. \tag{7.1}$$

Solving (7.1) for Z we obtain

$$Z = f\frac{T}{d}, \tag{7.2}$$

where $d = x_r - x_l$, the *disparity*, measures the difference in retinal position between the corresponding points in the two images. From (7.2) we see that *depth is inversely proportional to disparity*. You can verify this by looking at moving objects outside: distant objects seem to move more slowly than close ones.

7.1.3 The Parameters of a Stereo System

As shown by (7.2), in our simple example depth depends on the focal length, f, and the stereo baseline, T; the coordinates x_l and x_r are referred to the principal points, c_l and c_r. The quantities f, T, c_l, c_r are the *parameters of the stereo system*, and finding their values is the *stereo calibration* problem. There are two kinds of parameters to be calibrated in a general stereo system.

The Parameters of a Stereo System

The *intrinsic parameters* characterize the transformation mapping an image point from camera to pixel coordinates, in each camera.

The *extrinsic parameters* describe the relative position and orientation of the two cameras.

The intrinsic parameters are the ones introduced in Chapter 2; a minimal set for each camera includes the coordinates of the principal point and the focal lengths in pixel. The extrinsic parameters, instead, are slightly different: they describe the rigid transformation (rotation and translation) that brings the reference frames of the two cameras onto each other.[3]

Since in many cases the intrinsic parameters, or the extrinsic parameters, or both, are unknown, *reconstruction is often a calibration problem*. Rather surprisingly, you will learn in Section 7.4 that a stereo system can compute a great deal of 3-D information without *any* prior knowledge of the stereo parameters (*uncalibrated stereo*). In order

[3] In Chapter 2, the same extrinsic parameters were used to define the 3-D rigid motion that brings the camera and the world reference frame onto each other; here, the reference frame of one camera is taken as the world reference frame.

to deal properly with reconstruction, we need to spend some time on the geometry of stereo, the so-called *epipolar geometry* (Section 7.3). As a byproduct, the epipolar geometry will prove useful to get a better understanding of the computational problems of stereo, and to devise more efficient (and effective) correspondence algorithms.

☞ Before starting our investigation of stereo, a word of warning about the validity of the conclusions that can be drawn from the simple stereo model (7.2) referring to Figure 7.4. They illustrate well the main issues of stereo, but are too simple to tell the entire story. In particular, (7.2) may lead you to conclude that the disparity can only decrease as the distance of the object from the cameras increases; instead, in a typical stereo system with converging cameras,[4] disparity actually *increases* with the distance of the objects *from the fixation point*. Clearly, the reason why you cannot infer this property from our example is that the fixation point is at infinity.

7.2 The Correspondence Problem

Let us first discuss the correspondence problem ignoring quantitative knowledge of the cameras' parameters.

7.2.1 Basics

We will start off with the common assumptions underlying most methods for finding correspondences in image pairs.

Assumptions

1. Most scene points are visible from both viewpoints.
2. Corresponding image regions are similar.

These assumptions hold for stereo systems in which the distance of the fixation point from the cameras is much larger than the baseline. In general, however, both assumptions may be false and the correspondence problem becomes considerably more difficult. For the time being, we take the validity of these assumptions for granted and view the correspondence problem as a *search* problem: *given an element in the left image, we search for the corresponding element in the right image*. This involves two decisions:

- which image element to match, and
- which similarity measure to adopt.

☞ We postpone the discussion of the problem arising from the fact that not all the elements of one image have necessarily a corresponding element in the other image.

[4] That is, the optical axes intersect in a fixation point at a finite distance from the cameras.

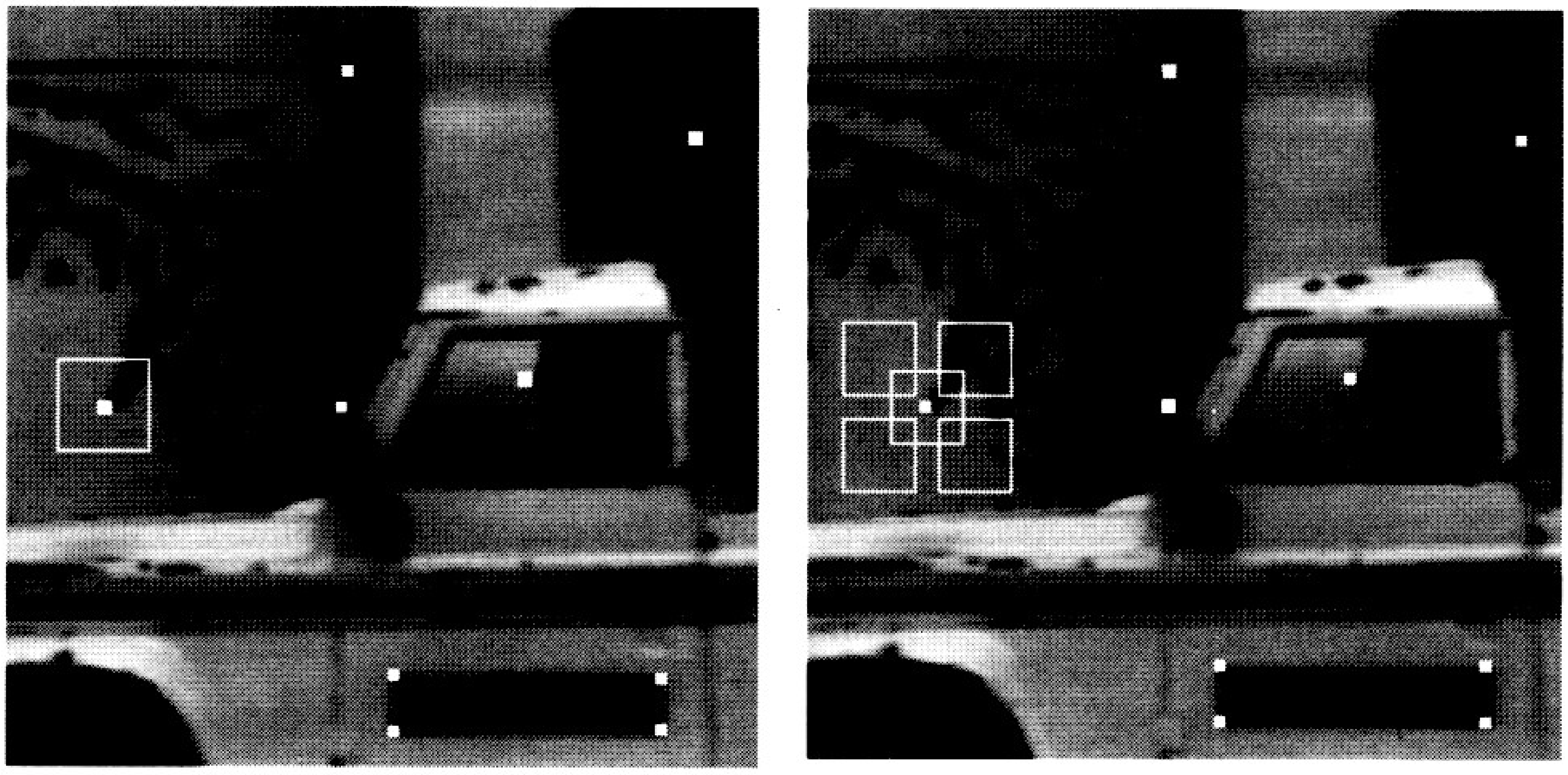


Figure 7.5 An illustration of correlation-based correspondence. We look for the right-image point corresponding to the central pixel of the left-image window. This window is correlated to several windows of the same size in the right image (only a few are shown here). The center of the right-image window producing the highest correlation is the corresponding point sought.

For the sake of convenience, we classify correspondence algorithms in two classes, *correlation-based* and *feature-based* methods, and discuss them separately. Although almost indistinguishable from a conceptual point of view, the two classes lead to quite different implementations: for instance, correlation-based methods apply to the totality of image points; feature-based methods attempt to establish a correspondence between sparse sets of image features.

7.2.2 Correlation-Based Methods

In correlation-based methods, the elements to match are *image windows* of fixed size, and the similarity criterion is a measure of the correlation between windows in the two images. The corresponding element is given by the window that maximizes the similarity criterion within a search region (Figure 7.5). As usual, we give a summary of the algorithm.

Algorithm CORR_MATCHING

The input is a stereo pair of images, I_l (left) and I_r (right).

Let $\mathbf{p}_l$ and $\mathbf{p}_r$ be pixels in the left and right image, $2W+1$ the width (in pixels) of the correlation window, $R(\mathbf{p}_l)$ the search region in the right image associated with $\mathbf{p}_l$, and $\psi(u, v)$ a function of two pixel values, u, v.

For each pixel $\mathbf{p}_l = [i, j]^\top$ of the left image:

1. for each displacement $\mathbf{d} = [d_1, d_2]^\top \in R(\mathbf{p}_l)$ compute

$$c(\mathbf{d}) = \sum_{k=-W}^{W} \sum_{l=-W}^{W} \psi(I_l(i+k, j+l), I_r(i+k-d_1, j+l-d_2)); \tag{7.3}$$

2. the disparity of $\mathbf{p}_l$ is the vector $\bar{\mathbf{d}} = \left[\bar{d}_1, \bar{d}_2\right]^\top$ that maximizes $c(\mathbf{d})$ over $R(\mathbf{p}_l)$:

$$\bar{\mathbf{d}} = \arg\max_{d \in R} \{c(\mathbf{d})\}\,.$$

The output is an array of disparities (the *disparity map*), one per each pixel of I_l.

Two widely adopted choices for the function $\psi = \psi(u, v)$ in (7.3) are

$$\psi(u, v) = uv, \tag{7.4}$$

which yields the *cross-correlation* between the window in the left image and the search region in the right image, and

$$\psi(u, v) = -(u - v)^2, \tag{7.5}$$

which performs the so called *SSD (sum of squared differences)* or *block matching*.

☞ The relation between cross-correlation and block matching becomes apparent expanding (7.5) (Exercise 7.3).

Notice that the two choices of ψ in (7.4) and (7.5) lead to the same set of correspondences *if* the *energy* of the right image inside each window, defined as the sum of the squares of the intensity values in the window, is constant across the search region. In many practical situations, this is unlikely to happen, and (7.5) *is usually preferable to (7.4)*. The reason is that SSD, unlike cross-correlation, is not biased by the presence of regions with very small or very large intensity values (see Exercise 7.3 for a quantitative estimation of this effect and the definition of *normalized cross-correlation*).

☞ In an implementation of CORR_MATCHING it is certainly worthwhile to precompute and store the values of the function ψ in a lookup table. For most choices of ψ, this is likely to speed up the algorithm substantially.

We must still discuss how to choose W and R in the implementation of the algorithm. The window width, $2W + 1$, is actually a free parameter, and its choice is left to your ability to grasp the most important spatial scale of the problem from the images you are dealing with.[5]

Fortunately, something more can be said on the initial location and size of the search region, $R(\mathbf{p}_l)$. If the cameras are fixating a common point at a distance much larger than the baseline, the initial location can be chosen to be $\mathbf{p}_r = [i, j]^\top$; that is, choosing the pixel in the right image at exactly the same location of the pixel $\mathbf{p}_l = [i, j]^\top$ in the left image. The size of $R(\mathbf{p}_l)$ can be estimated from the maximum range of

[5] See the Further Readings for stereo systems determining the size of the window automatically.

distances that you expect to find in the scene (remember that disparity increases with the inverse of the distance from the fixation point). If nothing can be assumed on the geometry of the two cameras, the initialization of the search region in the right image is more difficult. At first sight, it might even appear that for each point in the left image you have to search over the entire right image. Fortunately, as shown in Section 7.3, *the search region can always be reduced to a 1-D segment*, independent of the relative position of the two cameras.

7.2.3 Feature-based Methods

Feature-based methods restrict the search for correspondences to a sparse set of features. Instead of image windows, they use numerical and symbolic properties of features, available from feature descriptors (Chapters 4 and 5); instead of correlationlike measures, they use a measure of the distance between feature descriptors. Corresponding elements are given by the most similar feature pair, the one associated to the minimum distance.

Most methods narrow down the number of possible matches for each feature by enforcing suitable *constraints* on feasible matches. These constraints can be

- *geometric*, like the *epipolar constraint* discussed in the next section, or
- *analytical*, for instance the *uniqueness constraint* (each feature can at most have one match) or the *continuity constraint* (disparity varies continuously almost everywhere across the image).

Here, we restrict our attention to *unconstrained methods*; pointers to feature-based methods relying on geometric and analytical constraints are given in the Further Readings, and Section 7.3 gives you the basic elements necessary to add the epipolar constraint to our correspondence algorithms.

Typical examples of image features used in stereo are edge points, lines, and corners (either points where edge lines meet, or corners formed by intensity patterns as described in Chapter 4). For example, a feature descriptor for a line could contain values for

- the length, l
- the orientation, o
- the coordinates of the midpoint, $[x, y]^\top$
- the average contrast along the edge line, c

For virtually every possible feature, there is a long list of correspondence algorithms. As you might expect, no single feature type will work at best in all situations. The choice of the feature type (and correspondence method) depends on numerous factors, like the kind of objects you are looking at, the overall conditions of illumination, and the average image contrast. As an example, we suggest a simplified scheme for matching edge lines, assuming the line descriptor above.[6]

An example of similarity criterion between feature descriptors is the inverse of the weighted average, S, of the distances between each of the properties in the descriptors:

$$S = \frac{1}{w_0(l_l - l_r)^2 + w_1(\theta_l - \theta_r)^2 + w_2(m_l - m_r)^2 + w_3(c_l - c_r)^2}, \tag{7.6}$$

where $w_0, \ldots, w_3$ are weights, and the subscripts l and r refer to the left and right image, respectively. This leaves us with the nontrivial task of determining the weights that yield the best matches. A possible strategy is to determine a working point for the various weights from a subset of easy matches that you believe to be correct. In general, there is not much to say about this problem, except perhaps that if you build a complex and heterogeneous feature descriptor, determining the weights becomes a difficult problem of parameter estimation.

A very simple, feature-based correspondence algorithm is sketched below.

Algorithm FEATURE_MATCHING

The input is a stereo pair of images, I_l and I_r, and two corresponding sets of feature descriptors.

Let $R(f_l)$ be the search region in the right image associated with a feature descriptor f_l, and $\mathbf{d}(f_l, f_r)$ the disparity between two corresponding features, f_l and f_r.

For each f_l in the left image set:

1. Compute the similarity measure between f_l and each image feature in $R(f_l)$.
2. Select the right-image feature, f_r that maximizes the similarity measure.
3. Save the correspondence and the disparity of f_l (the displacement between the points defining the feature's position).

The output is formed by a list of feature correspondences and a disparity map.

☞ The initial location and size of the search region, R, can be determined as in the case of correlation-based methods.

7.2.4 Concluding Remarks

Unfortunately, there is no cut-and-dried correspondence method giving optimal results under all possible circumstances. You have to live with the fact that choosing a method depends on factors like the application, the available hardware, or the software requirements. Having said that, it is useful to keep in mind a few general considerations.

Correlation-based methods are certainly easier to implement (and debug) and provide dense disparity maps. As you may guess, the latter property can be very helpful

[6] In reality, feature-based methods can be a great deal more complicated than our solution, especially when enforcing constraints on the search.

for the purpose of reconstructing surfaces. They need textured images to work well. However, due to foreshortening effects and change in illumination direction, they are inadequate for matching image pairs taken from very different viewpoints. Also, the interpolation necessary to refine correspondences from pixel to subpixel precision can make correlation-based matching quite expensive.[7]

Feature-based methods are suitable when *a priori* information is available about the scene, so that optimal features can be used. A typical example is the case of indoor scenes, which usually contain many straight lines but rather untextured surfaces. Feature-based algorithms can also prove faster than correlation-based ones, but any comparison of specific algorithms must take into account the cost of producing the feature descriptors. The sparse disparity maps generated by these methods may look inferior to the dense maps of correlation-based matching, but in some applications (e.g., visual navigation) they may well be all you need in order to perform the required tasks successfully. Another advantage of feature-based techniques is that they are relatively insensitive to illumination changes and highlights.

The performance of any correspondence methods is jeopardised by *occlusions* (points with no counterpart in the other image) and *spurious matches* (false corresponding pairs created by noise). Appropriate constraints reduce the effects of both phenomena: two important ones are the *left-right consistency constraint* (only corresponding pairs found matching left-to-right *and* right-to-left are accepted), and the *epipolar constraint*, explained in the next section.

7.3 Epipolar Geometry

We now move on to study the geometry of stereo in its full generality. This will enable us to clarify what information is needed in order to perform the search for corresponding elements only along image lines. First of all, we need to establish some basic notations.

7.3.1 Notation

The geometry of stereo, known as *epipolar geometry*, is shown in Figure 7.6. The figure shows two pinhole cameras, their projection centers, O_l and O_r, and image planes, π_l and π_r. The focal lengths are denoted by f_l and f_r. As usual, each camera identifies a 3-D reference frame, the origin of which coincides with the projection center, and the Z-axis with the optical axis. The vectors $\mathbf{P}_l = [X_l, Y_l, Z_l]^\top$ and $\mathbf{P}_r = [X_r, Y_r, Z_r]^\top$ refer to the same 3-D point, P, thought of as a vector in the left and right camera reference frames respectively (Figure 7.6). The vectors $\mathbf{p}_l = [x_l, y_l, z_l]^\top$ and $\mathbf{p}_r = [x_r, y_r, z_r]^\top$ refer to the projections of P onto the left and right image plane respectively, and are expressed in the corresponding reference frame (Figure 7.6). Clearly, for all the image points we have $z_l = f_l$ or $z_r = f_r$, according to the image. Since each image plane can be thought of as a subset of the projective space P^2, image points can be equivalently thought of as points of the projective space P^2 (see Appendix, section A.4).

[7] One of the projects suggested at the end of this chapter deals with a parsimonious implementation of correlation-based matching.

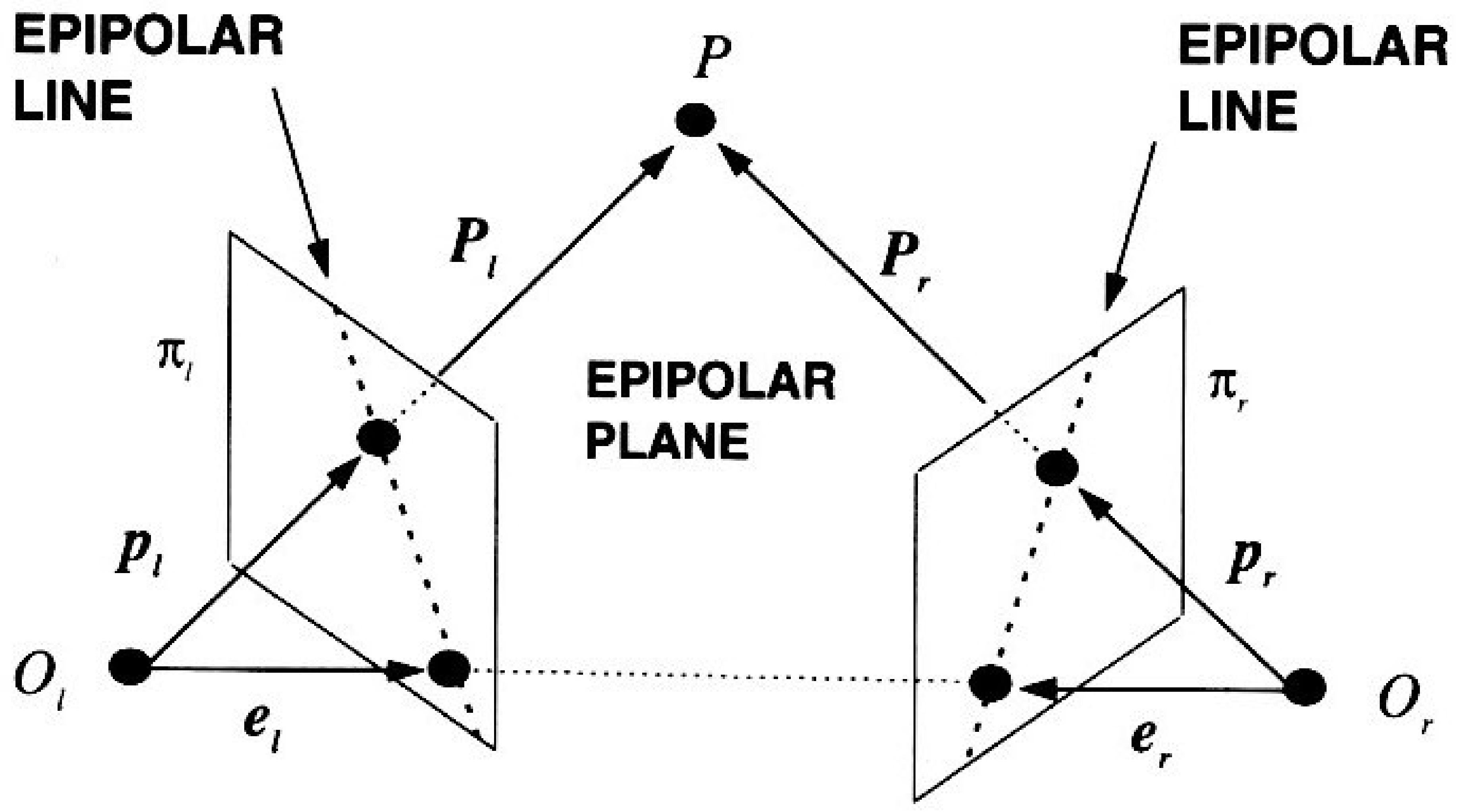


Figure 7.6 The epipolar geometry.

☞ Note that point vectors denoted by the *same* bold capital letter but by *different* subscripts, like $\mathbf{P}_l$ and $\mathbf{P}_r$ identify the *same point* in space. The subscript l or r tells you *the reference frame in which the vectors are expressed* (left or right). Instead, point vectors denoted by the *same* bold small letter but by a *different* subscript, like $\mathbf{p}_l$ and $\mathbf{p}_r$, identify *different points* in space (i.e., belonging to different image planes). In this case, the subscript tells you *also* the image plane on which the vectors lie. This is a slightly unfair but very effective abuse of notation.

7.3.2 Basics

The reference frames of the left and right cameras are related via the extrinsic parameters. These define a rigid transformation in 3-D space, defined by a translation vector, $\mathbf{T} = (O_r - O_l)$, and a rotation matrix, R. Given a point P in space, the relation between $\mathbf{P}_l$ and $\mathbf{P}_r$ is therefore

$$\mathbf{P}_r = R(\mathbf{P}_l - \mathbf{T}). \tag{7.7}$$

The name *epipolar geometry* is used because the points at which the line through the centers of projection intersects the image planes (Figure 7.6) are called *epipoles*. We denote the left and right epipole by $\mathbf{e}_l$ and $\mathbf{e}_r$ respectively. By construction, *the left epipole is the image of the projection center of the right camera and vice versa.*

☞ Notice that, if the line through the centers of projection is parallel to one of the image planes, the corresponding epipole is the point at infinity of that line.

The relation between a point in 3-D space and its projections is described by the usual equations of perspective projection, in vector form:

$$\mathbf{p}_l = \frac{f_l}{Z_l}\mathbf{P}_l \tag{7.8}$$

and

$$\mathbf{p}_r = \frac{f_r}{Z_r}\mathbf{P}_r. \tag{7.9}$$

The practical importance of epipolar geometry stems from the fact that the plane identified by P, O_l, and O_r, called *epipolar plane*, intersects each image in a line, called *epipolar line* (see Figure 7.6). Consider the triplet P, $\mathbf{p}_l$, and $\mathbf{p}_r$. Given $\mathbf{p}_l$, P can lie anywhere on the ray from O_l through $\mathbf{p}_l$. But, since the image of this ray in the right image is the epipolar line through the corresponding point, $\mathbf{p}_r$, *the correct match must lie on the epipolar line*. This important fact is known as the *epipolar constraint*. It establishes a mapping between points in the left image and lines in the right image and *vice versa*.

☞ Incidentally, since all rays include the projection center by construction, this also proves that all the epipolar lines go through the epipole.

So, if we determine the mapping between points on, say, the left image and corresponding epipolar lines on the right image, we can restrict the search for the match of $\mathbf{p}_l$ along the corresponding epipolar line. *The search for correspondences is thus reduced to a 1-D problem.* Alternatively, the same knowledge can be used to verify whether or not a candidate match lies on the corresponding epipolar line. This is usually a most effective procedure to *reject false matches* due to occlusions. Let us now summarize the main ideas encountered in this section:

Definition: Epipolar Geometry

Given a stereo pair of cameras, any point in 3-D space, P, defines a plane, π_P, going through P and the centers of projection of the two cameras. The plane π_P is called *epipolar plane*, and the lines where π_P intersects the image planes *conjugated epipolar lines*. The image in one camera of the projection center of the other is called *epipole*.

Properties of the Epipoles

With the exception of the epipole, only one epipolar line goes through any image point.

All the epipolar lines of one camera go through the camera's epipole.

Definition: Epipolar Constraint

Corresponding points must lie on conjugated epipolar lines.

The obvious question at this point is, can we estimate the epipolar geometry? Or equivalently, how do we determine the mapping between points in one image and epipolar lines in the other? This is the next problem we consider. Its solution also makes clear the relevance of epipolar geometry for reconstruction.

7.3.3 The Essential Matrix, E

The equation of the epipolar plane through P can be written as the coplanarity condition of the vectors $\mathbf{P}_l$, $\mathbf{T}$, and $\mathbf{P}_l - \mathbf{T}$ (Figure 7.6), or

$$(\mathbf{P}_l - \mathbf{T})^\top \mathbf{T} \times \mathbf{P}_l = 0. \qquad \equiv (P_L - T) \cdot (T \times P_L) = 0$$

Using (7.7), we obtain

$$(R^\top \mathbf{P}_r)^\top \mathbf{T} \times \mathbf{P}_l = 0. \tag{7.10}$$

Recalling that a vector product can be written as a multiplication by a rank-deficient matrix, we can write

$$\mathbf{T} \times \mathbf{P}_l = S\mathbf{P}_l$$

where

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}. \tag{7.11}$$

Using this fact, (7.10) becomes

$$\mathbf{P}_r^\top E \mathbf{P}_l = 0, \tag{7.12}$$

with

$$E = RS. \tag{7.13}$$

Note that, by construction, S has always rank 2. The matrix E is called the *essential matrix* and *establishes a natural link between the epipolar constraint and the extrinsic parameters of the stereo system.* You will learn how to recover the extrinsic parameters from the essential matrix in the next section. In the meantime, observe that, using (7.8) and (7.9), and dividing by $Z_r Z_l$, (7.12) can be rewritten as

$$\mathbf{p}_r^\top E \mathbf{p}_l = 0. \tag{7.14}$$

As already mentioned, the image points $\mathbf{p}_l$ and $\mathbf{p}_r$, which lie on the left and right image planes respectively, can be regarded as points in the projective plane P^2 defined by the left and right image planes respectively (Appendix, section A.4). Consequently, you are entitled to think of $E\mathbf{p}_l$ in (7.14) as the projective line in the right plane, $\mathbf{u}_r$, that goes through $\mathbf{p}_r$ and the epipole $\mathbf{e}_r$:

$$\mathbf{u}_r = E\mathbf{p}_l. \tag{7.15}$$

As shown by (7.14) and (7.15), *the essential matrix is the mapping between points and epipolar lines we were looking for.*

☞ Notice that the whole discussion used coordinates in the *camera* reference frame, but what we actually measure from images are pixel coordinates. Therefore, in order to be able to make profitable use of the essential matrix, we need to know the transformation from *camera coordinates* to *pixel coordinates*, that is, the intrinsic parameters. This limitation is removed in the next section, but at a price.

7.3.4 The Fundamental Matrix, F

We now show that the mapping between points and epipolar lines can be obtained from corresponding points only, *with no prior information on the stereo system.*

Let M_l and M_r be the matrices of the intrinsic parameters (Chapter 2) of the left and right camera respectively. If $\bar{\mathbf{p}}_l$ and $\bar{\mathbf{p}}_r$ are the points in *pixel* coordinates corresponding to $\mathbf{p}_l$ and $\mathbf{p}_r$ in camera coordinates, we have

$$\mathbf{p}_l = M_l^{-1}\bar{\mathbf{p}}_l \tag{7.16}$$

and

$$\mathbf{p}_r = M_r^{-1}\bar{\mathbf{p}}_r. \tag{7.17}$$

By substituting (7.16) and (7.17) into (7.14), we have

$$\bar{\mathbf{p}}_r^{\top} F \bar{\mathbf{p}}_l = 0, \tag{7.18}$$

where

$$F = M_r^{-\top} E M_l^{-1}. \tag{7.19}$$

F is named *fundamental matrix*. The essential and fundamental matrix, as well as (7.14) and (7.18), are formally very similar. As with $E\mathbf{p}_l$ in (7.14), $F\bar{\mathbf{p}}_l$ in (7.18) can be thought of as the equation of the projective epipolar line, $\bar{\mathbf{u}}_r$, that correspond to the point $\bar{\mathbf{p}}_l$, or

$$\bar{\mathbf{u}}_r = F\bar{\mathbf{p}}_l. \tag{7.20}$$

The most important difference between (7.15) and (7.20), and between the essential and fundamental matrices, is that *the fundamental matrix is defined in terms of pixel coordinates, the essential matrix in terms of camera coordinates.* Consequently, if you can estimate the fundamental matrix from a number of point matches in pixel coordinates, *you can reconstruct the epipolar geometry with no information at all on the intrinsic or extrinsic parameters.*

☞ This indicates that the epipolar constraint, as the mapping between points and corresponding epipolar lines, can be established with *no* prior knowledge of the stereo parameters.

The definitions and basic mathematical properties of these two important matrices are worth a summary.

Definition: Essential and Fundamental Matrices

For each pair of corresponding points $\mathbf{p}_l$ and $\mathbf{p}_r$ in *camera* coordinates, the *essential matrix* satisfies the equation

$$\mathbf{p}_r^{\top} E \mathbf{p}_l = 0.$$

For each pair of corresponding points $\bar{\mathbf{p}}_l$ and $\bar{\mathbf{p}}_r$ in *pixel* coordinates, the *fundamental matrix* satisfies the equation

$$\bar{\mathbf{p}}_r^{\top} F \bar{\mathbf{p}}_l = 0.$$

Properties

Both matrices enable full reconstruction of the epipolar geometry.

If M_l and M_r are the matrices of the intrinsic parameters, the relation between the essential and fundamental matrices is given by

$$F = M_r^{-T} E M_l^{-1}.$$

The essential matrix:

1. encodes information on the extrinsic parameters only (see (7.13))
2. has rank 2, since S in (7.11) has rank 2 and R full rank
3. its two nonzero singular values are equal

The fundamental matrix:

1. encodes information on both the intrinsic and extrinsic parameters
2. has rank 2, since T_l and T_r have full rank and E has rank 2

7.3.5 Computing E and F: The Eight-point Algorithm

How do we compute the essential and fundamental matrices? Of the various methods possible, the eight-point algorithm is by far the simplest and definitely the one you cannot ignore (if you are curious about other techniques, look into the Further Readings). We consider here the fundamental matrix only, and leave it to you to work out the straightforward modification needed to recover the essential matrix.

The idea behind the eight-point algorithm is very simple. Assume that you have been able to establish n point correspondences between the images. Each correspondence gives you a homogeneous linear equation like (7.18) for the nine entries of F; these equations form a homogeneous linear system. If you have at least eight correspondences (i.e., $n \geq 8$) and the n points do not form degenerate configurations,[8] the nine entries of F can be determined as the nontrivial solution of the system. Since the system is homogeneous, the solution is unique up to a signed scaling factor. If one uses more than eight points, so that the system is overdetermined, the solution can once again be obtained by means of SVD related techniques. If A is the system's matrix and $A = UDV^\top$, the solution is the column of V corresponding to the only null singular value of A (see Appendix, section A.6),

☞ Because of noise, numerical errors and inaccurate correspondences, A is more likely to be full rank, and the solution is the column of V associated with the *least* singular value of A.

☞ The estimated fundamental matrix is almost certainly nonsingular. We can enforce the singularity constraint by adjusting the entries of the estimated matrix F as done in Chapter 6

[8] For a thorough discussion of the degenerate configurations of eight or more points, as well as of the instabilities in the estimation of the essential and fundamental matrices, see the Further Readings.

for rotation matrices: we compute the singular value decomposition of the estimated matrix, $\hat{F} = UDV^\top$, and set the smallest singular value on the diagonal of the matrix D equal to 0. If D' is the corrected D matrix, the corrected estimate, F' is given by $F' = UD'V^\top$ (see Appendix, section A.6).

The following is the basic structure of the eight-point algorithm:

Algorithm EIGHT_POINT

The input is formed by n point correspondences, with $n \geq 8$.

1. Construct system (7.18) from n correspondences. Let A be the $n \times 9$ matrix of the coefficients of the system and $A = UDV^\top$ the SVD of A.
2. The entries of F (up to an unknown, signed scale factor) are the components of the column of V corresponding to the least singular value of A.
3. To enforce the singularity constraint, compute the singular value decomposition of F:

$$F = UDV^\top.$$

4. Set the smallest singular value in the diagonal of D equal to 0; let D' be the corrected matrix.
5. The corrected estimate of F, F', is finally given by

$$F' = UD'V^\top.$$

The output is the estimate of the fundamental matrix, F'.

☞ In order to avoid numerical instabilities, the eight-point algorithm should be implemented with care. The most important action to take is *to normalize the coordinates of the corresponding points so that the entries of A are of comparable size*. Typically, the first two coordinates (in pixels) of an image point are referred to the top left corner of the image, and can vary between a few pixels to a few hundreds; the differences can make A seriously ill-conditioned (Appendix, section A.6). To make things worse, the third (homogeneous) coordinate of image points is usually set to one. A simple procedure to avoid numerical instability is to translate the first two coordinates of each point to the centroid of each data set, and scale the norm of each point so that the average norm over the data set is 1. This can be accomplished by multiplying each left (right) point by two suitable 3×3 matrices, H_l and H_r (see Exercise 7.6 for details on how to compute both H_l and H_r). The algorithm EIGHT_POINT is then used to estimate the matrix $\bar{F} = H_r F H_l$, and F obtained as $H_r^{-1} \bar{F} H_l^{-1}$.

7.3.6 Locating the Epipoles from E and F

We can now establish the relation between the epipoles and the two matrices E *and* F. Consider for example the fundamental matrix, F. Since $\bar{\mathbf{e}}_l$ lies on all the epipolar lines of the left image, we can rewrite (7.18) as

$$\bar{\mathbf{p}}_r^\top F \bar{\mathbf{e}}_l = 0$$

for every $\bar{\mathbf{p}}_r$. But since F is not identically zero, this is possible if and only if

$$F\bar{\mathbf{e}}_l = 0. \tag{7.21}$$

From (7.21) and the fact that F has rank 2, it follows that *the epipole, $\bar{\mathbf{e}}_l$, is the null space of F*; similarly, $\bar{\mathbf{e}}_r$ is the null space of $F^\top$.

We are now in a position to present an algorithm for finding the epipoles. Accurate epipole localization is helpful for refining the location of corresponding epipolar lines, checking the geometric consistency of the entire construction, simplifying the stereo geometry, and recovering 3-D structure in the case of uncalibrated stereo.

Again we present the algorithm in the case of the fundamental matrix. The adaptation to the case of the essential matrix is even simpler than before. The algorithm follows easily from (7.21): To determine the location of the epipoles, it is sufficient to find the null spaces of F and $F^\top$.

☞ These can be determined, for instance, from the singular value decomposition $F = UDV^\top$ and $F^\top = VDU^\top$ as column of V and U respectively corresponding to the null singular value in the diagonal matrix D.

Algorithm EPIPOLES_LOCATION

The input is the fundamental matrix F.

1. Find the SVD of F, that is, $F = UDV^\top$.
2. The epipole $\mathbf{e}_l$ is the column of V corresponding to the null singular value.
3. The epipole $\mathbf{e}_r$ is the column of U corresponding to the null singular value.

The output are the epipoles, $\mathbf{e}_l$ and $\mathbf{e}_r$.

☞ Notice that we can safely assume that there is exactly one singular value equal to 0 because algorithm EIGHT_POINT enforces the singularity constraint explicitly.

It has to be noticed that there are alternative methods to locate the epipoles, not based on the fundamental matrix and requiring as few as 6 point correspondences. More about them in the Further Readings.

7.3.7 Rectification

Before moving on to the problem of 3-D reconstruction, we want to address the issue of *rectification*. Given a pair of stereo images, rectification determines a transformation (or *warping*) of each image such that *pairs of conjugate epipolar lines become collinear and parallel to one of the image axes*, usually the horizontal one. Figure 7.7 shows an example. The importance of rectification is that the correspondence problem, which involves 2-D search in general, *is reduced to a 1-D search on a scanline identified trivially*. In other words, to find the point corresponding to (i_l, j_l) of the left image, we just look along the scanline $j = j_l$ in the right image.

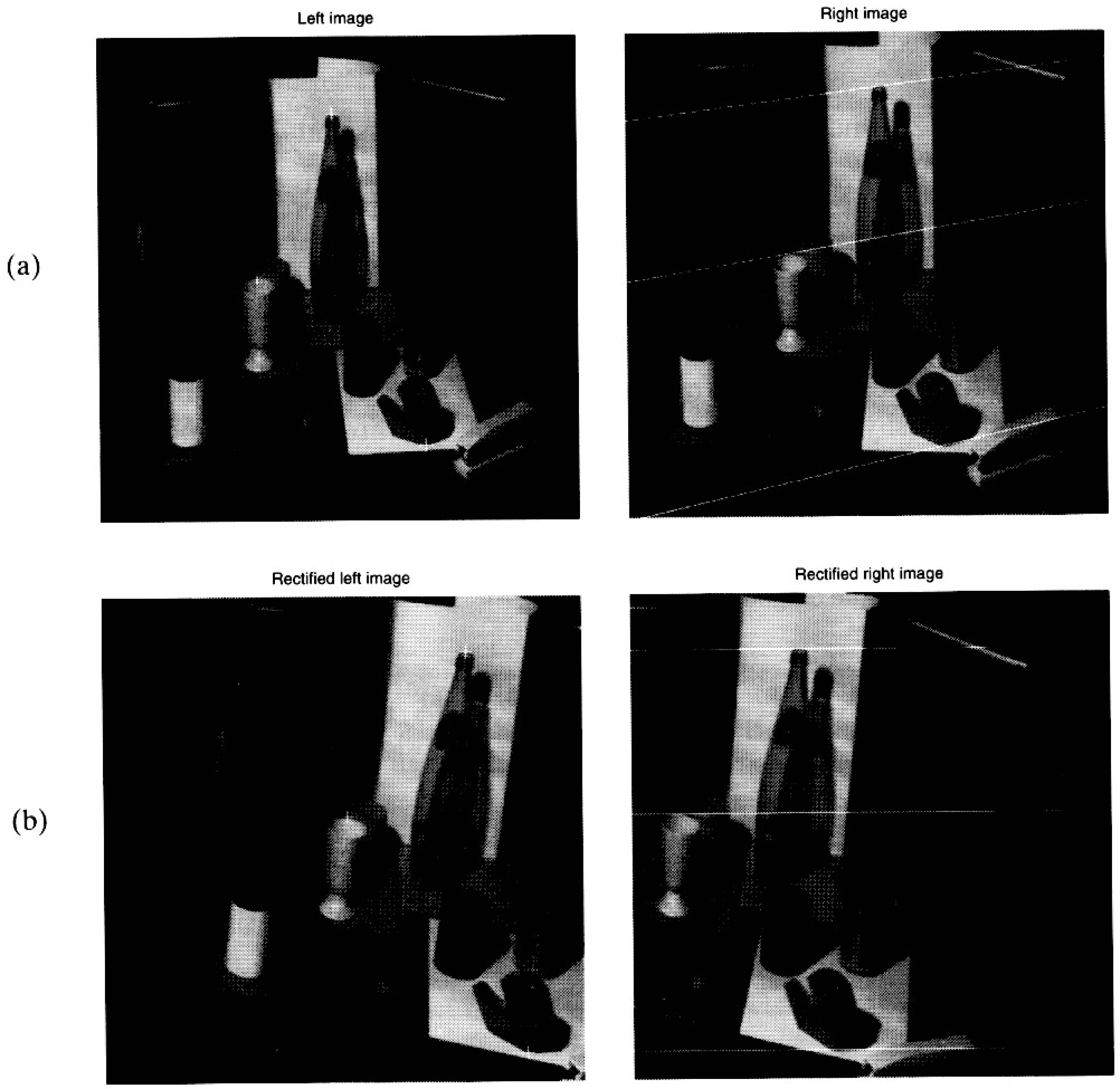


Figure 7.7 (a) A stereo pair. (b) The pair rectified. The left images plot the epipolar lines corresponding to the points marked in the right pictures. Stereo pair courtesy of INRIA (France).

Let us begin by stating the problem and our assumptions.

Assumptions and Problem Statement

Given a stereo pair of images, the intrinsic parameters of each camera, and the extrinsic parameters of the system, R and $\mathbf{T}$, compute the image transformation that makes conjugated epipolar lines collinear and parallel to the horizontal image axis.

The assumption of knowing the intrinsic and extrinsic parameters is not strictly necessary (see Further Readings) but leads to a very simple technique. How do we

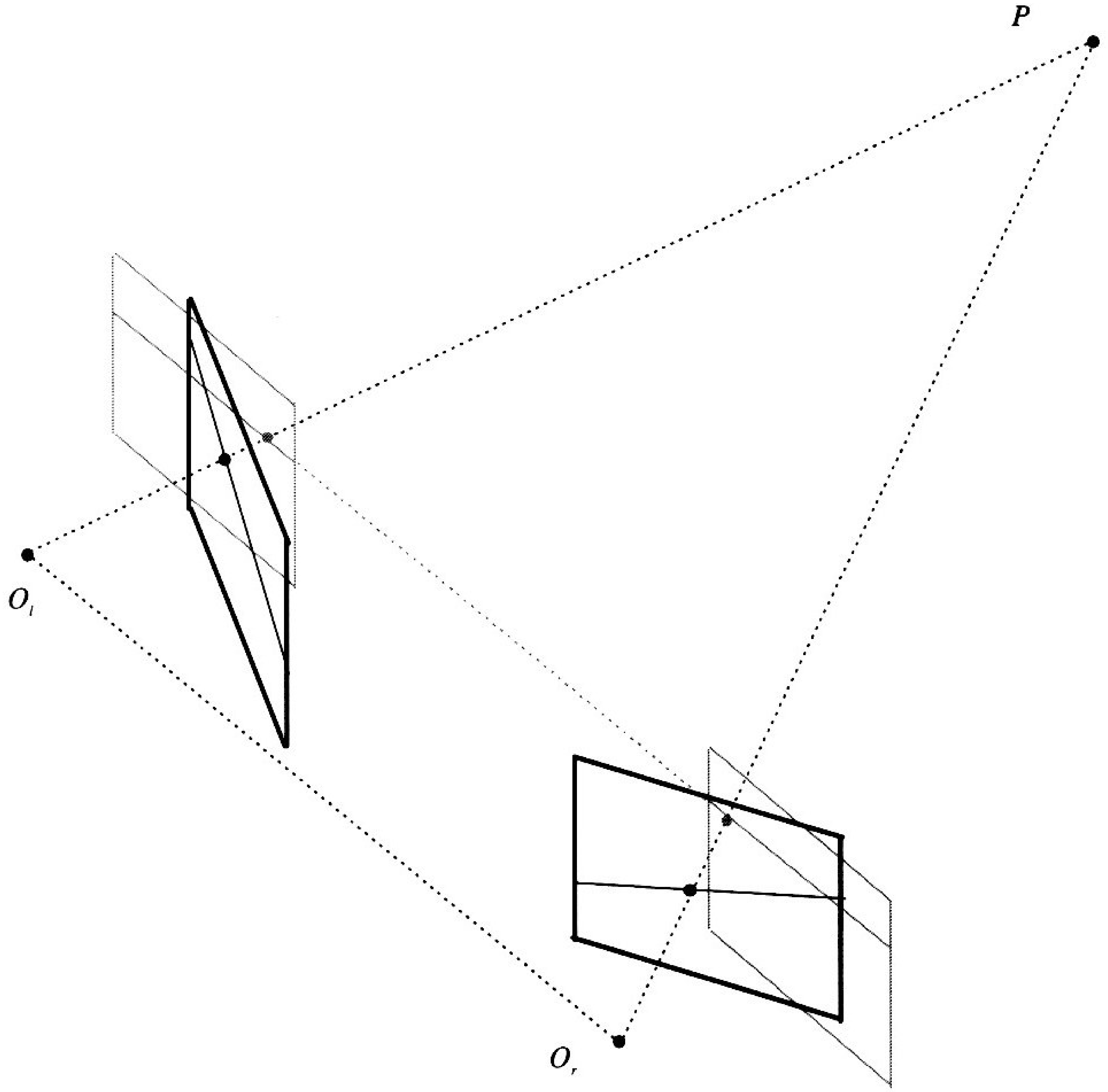


Figure 7.8 Rectification of a stereo pair. The epipolar lines associated to a 3-D point P in the original cameras (black lines) become collinear in the rectified cameras (light grey). Notice that the original cameras can be in any position, and the optical axes may not intersect.

go about computing the rectifying image transformation? The rectified images can be thought of as acquired by a new stereo rig, obtained by rotating the original cameras around their optical centers. This is illustrated in Figure 7.8, which shows also how the points of the rectified images are determined from the points of the original images and their corresponding projection rays.

We proceed to describe a rectification algorithm assuming, without losing generality, that in both cameras

1. the origin of the image reference frame is the principal point;
2. the focal length is equal to f.

The algorithm consists of four steps:

- Rotate the left camera so that the epipole goes to infinity along the horizontal axis.
- Apply the same rotation to the right camera to recover the original geometry.
- Rotate the right camera by R.
- Adjust the scale in both camera reference frames.

To carry out this method, we construct a triple of mutually orthogonal unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$. Since the problem is underconstrained, we are going to make an arbitrary choice. The first vector, $\mathbf{e}_1$, is given by the epipole; since the image center is in the origin, $\mathbf{e}_1$ coincides with the direction of translation, or

$$\mathbf{e}_1 = \frac{\mathbf{T}}{\|\mathbf{T}\|}.$$

The only constraint we have on the second vector, $\mathbf{e}_2$, is that it must be orthogonal to $\mathbf{e}_1$. To this purpose, we compute and normalize the cross product of $\mathbf{e}_1$ with the direction vector of the optical axis, to obtain

$$\mathbf{e}_2 = \frac{1}{\sqrt{T_x^2 + T_y^2}} \left[-T_y, T_x, 0\right]^\top.$$

The third unit vector is unambiguously determined as

$$\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2.$$

It is easy to check that the orthogonal matrix defined as

$$R_{rect} = \begin{pmatrix} \mathbf{e}_1^\top \\ \mathbf{e}_2^\top \\ \mathbf{e}_3^\top \end{pmatrix} \tag{7.22}$$

rotates the left camera about the projection center in such a way that the epipolar lines become parallel to the horizontal axis. This implements the first step of the algorithm. Since the remaining steps are straightforward, we proceed to give the customary algorithm:

Algorithm RECTIFICATION

The input is formed by the intrinsic and extrinsic parameters of a stereo system and a set of points in each camera to be rectified (which could be the whole images). In addition, Assumptions 1 and 2 above hold.

1. Build the matrix R_{rect} as in (7.22);
2. Set $R_l = R_{rect}$ and $R_r = RR_{rect}$;

3. For each left-camera point, $\mathbf{p}_l = [x, y, f]^\top$ compute

$$R_l \mathbf{p}_l = [x', y', z']$$

and the coordinates of the corresponding rectified point, $\mathbf{p}'_l$, as

$$\mathbf{p}'_l = \frac{f}{z'}[x', y', z'].$$

4. Repeat the previous step for the right camera using R_r and $\mathbf{p}_r$.

The output is the pair of transformations to be applied to the two cameras in order to rectify the two input point sets, as well as the rectified sets of points.

Notice that the rectified coordinates are in general not integer. Therefore, if you want to obtain integer coordinates (for instance if you are rectifying the whole images), you should implement RECTIFICATION backwards, that is, starting from the *new* image plane and applying the *inverse* transformations, so that the pixel values in the *new* image plane can be computed as a bilinear interpolation of the pixel values in the *old* image plane.

☞ A rectified image is not in general contained in the same region of the image plane as the original image. You may have to alter the focal lengths of the rectified cameras to keep all the points within images of the same size as the original.

We are now fully equipped to deal with the reconstruction problem of stereo.

7.4 3-D Reconstruction

We have learned methods for solving the correspondence problem and determining the epipolar geometry from at least eight point correspondences. At this point, the 3-D reconstruction that can be obtained depends on the amount of *a priori* knowledge available on the parameters of the stereo system; we can identify three cases.[9] First, if both intrinsic and extrinsic parameters are known, you can solve the reconstruction problem unambiguously by triangulation, as detailed in section 7.1. Second, if only the intrinsic parameters are known, you can still solve the problem and, at the same time, estimate the extrinsic parameters of the system, but only *up to an unknown scaling factor*. Third, if the pixel correspondences are the only information available, and neither the intrinsic nor the extrinsic parameters are known, you can still obtain a reconstruction of the environment, but only *up to an unknown, global projective transformation*. Here is a visual summary.

[9] In reality there are several intermediate cases, but we concentrate on these three for simplicity.

A Priori Knowledge	3-D Reconstruction from Two Views
Intrinsic and extrinsic parameters	Unambiguous (absolute coordinates)
Intrinsic parameters only	Up to an unknown scaling factor
No information on parameters	Up to an unknown projective transformation of the environment

We now consider these three cases in turn.

7.4.1 Reconstruction by Triangulation

This is the simplest case. If you know both the intrinsic and the extrinsic parameters of your stereo system, reconstruction is straightforward.

Assumptions and Problem Statement

Under the assumption that the intrinsic and extrinsic parameters are known, compute the 3-D location of the points from their projections, $\mathbf{p}_l$ and $\mathbf{p}_r$.

As shown in Figure 7.6, the point P, projected into the pair of corresponding points $\mathbf{p}_l$ and $\mathbf{p}_r$, lies at the intersection of the two rays from O_l through $\mathbf{p}_l$ and from O_r through $\mathbf{p}_r$ respectively. In our assumptions, the rays are known and the intersection can be computed. The problem is, since parameters and image locations are known only approximately, *the two rays will not actually intersect in space*; their intersection can only be estimated as the point of minimum distance from both rays. This is what we set off to do.

Let $a\mathbf{p}_l$ ($a \in \mathbb{R}$) be the ray, l, through O_l and $\mathbf{p}_l$. Let $\mathbf{T} + bR^\top \mathbf{p}_r$ ($b \in \mathbb{R}$) be the ray, r, through O_r and $\mathbf{p}_r$ expressed in the left reference frame. Let $\mathbf{w}$ be a vector orthogonal to both l and r. Our problem reduces to determining the midpoint, P', of the segment parallel to $\mathbf{w}$ that joins l and r (Figure 7.9).

This is very simple because the endpoints of the segment, say $a_0\mathbf{p}_l$ and $\mathbf{T} + b_0R^\top \mathbf{p}_r$, can be computed solving the linear system of equations

$$a\mathbf{p}_l - bR^\top \mathbf{p}_r + c(\mathbf{p}_l \times R^\top \mathbf{p}_r) = \mathbf{T} \tag{7.23}$$

for a_0, b_0, and c_0. We summarize this simple method below:

Algorithm TRIANG

All vectors and coordinates are referred to the left camera reference frame. The input is formed by a set of corresponding points; let $\mathbf{p}_l$ and $\mathbf{p}_r$ be a generic pair.

Let $a\mathbf{p}_l$, $a \in \mathbb{R}$, be the ray l through O_l $(a=0)$ and $\mathbf{p}_l$ $(a=1)$. Let $\mathbf{T} + bR^\top\mathbf{p}_r$, $b \in \mathbb{R}$, the ray r through O_r $(b=0)$ and $\mathbf{p}_r$ $(b=1)$. Let $\mathbf{w} = \mathbf{p}_l \times R^\top\mathbf{p}_r$ the vector orthogonal to both l and r, and $a\mathbf{p}_l + c\mathbf{w}$, $c \in \mathbb{R}$, the line w through $a\mathbf{p}_l$ (for some fixed a) and parallel to $\mathbf{w}$.

1. Determine the endpoints of the segment, s, belonging to the line parallel to $\mathbf{w}$ that joins l and r, $a_0\mathbf{p}_l$ and $\mathbf{T} + b_0R^\top\mathbf{p}_r$, by solving (7.23).
2. The triangulated point, P', is the midpoint of the segment s.

The output is the set of reconstructed 3-D points.

☞ The determinant of the coefficients of system (7.23) is the triple product of $\mathbf{p}_l$, $R^\top\mathbf{p}_r$, and $\mathbf{p}_l \times R^\top\mathbf{p}_r$. Therefore, as expected from geometric considerations, the system has a unique solution if and only if the two rays l and r are not parallel.

☞ Reconstruction can be performed from rectified images directly; that is, without going back to the coordinate frames of the original pair (Exercise 7.7).

How often can we assume to know the intrinsic and extrinsic parameters of a stereo system? If the geometry of the system does not change with time, the intrinsic and extrinsic parameters of each camera can be estimated through the procedures of Chapter 6. If $\mathbf{T}_l$, R_l, and $\mathbf{T}_r$, R_r are the extrinsic parameters of the two cameras in the world reference frame, it is not difficult to show that the extrinsic parameters of the stereo system, $\mathbf{T}$ and R, are

$$\begin{aligned} R &= R_r R_l^\top \\ \mathbf{T} &= \mathbf{T}_l - R^\top\mathbf{T}_r. \end{aligned} \tag{7.24}$$

Try to derive (7.24) yourself. If you need help, see Exercise 7.10.

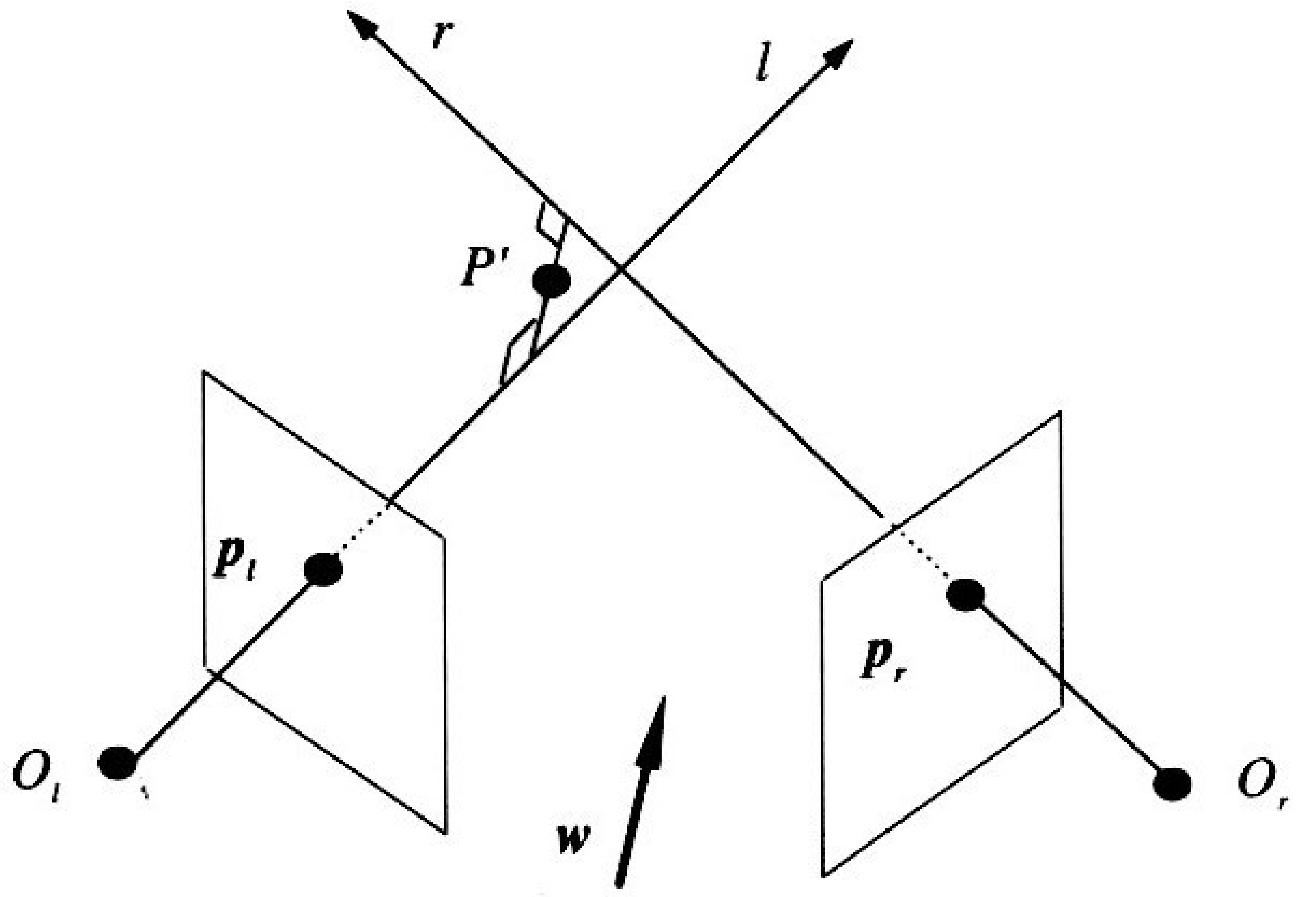


Figure 7.9 Triangulation with nonintersecting rays.

7.4.2 Reconstruction up to a Scale Factor

We now consider the case in which *only the intrinsic parameters of both the cameras are known* and derive a method to estimate the extrinsic parameters of the stereo system as well as the 3-D structure of the scene. Since the method makes use of the essential matrix, we must assume that at least eight point correspondences have been established.

Assumptions and Problem Statement

Assuming that only the intrinsic parameters and n point correspondences are given, with $n \geq 8$, compute the location of the 3-D points from their projections, $\mathbf{p}_l$ and $\mathbf{p}_r$.

Unlike triangulation, in which the geometry of the stereo system was fully known, the solution cannot rely on sufficient information to locate the 3-D points unambiguously. Intuitively, *since we do not know the baseline of the system, we cannot recover the true scale of the viewed scene*. Consequently, the reconstruction is unique only up to an unknown scaling factor. This factor can be determined if we know the distance between two points in the observed scene.

The origin of this ambiguity is quite clear in the method that we now present. The first step requires estimation of the essential matrix, E, which can only be known up to an arbitrary scale factor; therefore, we look for a convenient normalization of E. From the definition of the essential matrix, (7.13), we have

$$E^\top E = S^\top R^\top R S = S^\top S,$$

or

$$E^\top E = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_y T_x & T_z^2 + T_x^2 & -T_y T_z \\ -T_z T_x & -T_z T_y & T_x^2 + T_y^2 \end{bmatrix}. \tag{7.25}$$

From (7.25) we have that the trace of $EE^\top$ is

$$Tr(E^\top E) = 2\|\mathbf{T}\|^2,$$

so that dividing the entries of the essential matrix by

$$N = \sqrt{Tr(E^\top E)/2}$$

is equivalent to normalizing the length of the translation vector to unit.

☞ Notice that, by effect of this normalization, the difference between the true essential matrix and the one estimated through the eight-point algorithm is, at most, a global sign change.

Using this normalization, (7.25) can be rewritten as

$$\hat{E}^\top \hat{E} = \begin{bmatrix} 1-\hat{T}_x^2 & -\hat{T}_x\hat{T}_y & -\hat{T}_x\hat{T}_z \\ -\hat{T}_y\hat{T}_x & 1-\hat{T}_y^2 & -\hat{T}_y\hat{T}_z \\ -\hat{T}_z\hat{T}_x & -\hat{T}_z\hat{T}_y & 1-\hat{T}_z^2 \end{bmatrix}, \tag{7.26}$$

where $\hat{E}$ is the normalized essential matrix and $\hat{\mathbf{T}} = \mathbf{T}/\|\mathbf{T}\|$ the normalized translation vector. Recovering the components of $\hat{\mathbf{T}}$ from any row or column of the matrix $\hat{E}^\top\hat{E}$ is now a simple matter. However, since each entry of the matrix $\hat{E}^\top\hat{E}$ in (7.26) is quadratic in the components of $\hat{\mathbf{T}}$, the estimated components might differ from the true components by a global sign change. Let us assume, for the time being, that $\hat{\mathbf{T}}$ has been recovered with the proper global sign; then the rotation matrix can be obtained by simple algebraic computations. We define

$$\mathbf{w}_i = \hat{\mathbf{E}}_i \times \hat{\mathbf{T}}, \tag{7.27}$$

with $i = 1, 2, 3$ and $\hat{\mathbf{E}}_i$ the three rows of the normalized essential matrix $\hat{E}$, thought of as 3-D vectors. If $\mathbf{R}_i$ are the rows of the rotation matrix R, again thought of as 3-D vectors, easy but rather lengthy algebraic calculations yield

$$\mathbf{R}_i = \mathbf{w}_i + \mathbf{w}_j \times \mathbf{w}_k \tag{7.28}$$

with the triplet (i, j, k) spanning all cyclic permutations of $(1, 2, 3)$.

In summary, given an estimated, normalized essential matrix, we end up with four different estimates for the pair $(\hat{\mathbf{T}}, R)$. These four estimates are generated by the twofold ambiguity in the sign of $\hat{E}$ and $\hat{\mathbf{T}}$. The 3-D reconstruction of the viewed points resolves the ambiguity and finds the only correct estimate. For each of the four pairs $(\hat{\mathbf{T}}, R)$, we compute the third component of each point in the left camera reference frame. From (7.7) and (7.9), and since $Z_r = \mathbf{R}_3^\top(\mathbf{P}_l - \hat{\mathbf{T}})$, we obtain

$$\mathbf{p}_r = \frac{f_r R(\mathbf{P}_l - \hat{\mathbf{T}})}{\mathbf{R}_3^\top(\mathbf{P}_l - \hat{\mathbf{T}})}.$$

Thus, for the first component of $\mathbf{p}_r$ we have

$$x_r = \frac{f_r \mathbf{R}_1^\top(\mathbf{P}_l - \hat{\mathbf{T}})}{\mathbf{R}_3^\top(\mathbf{P}_l - \hat{\mathbf{T}})}. \tag{7.29}$$

Finally, plugging (7.8) into (7.29) with $\mathbf{T} = \hat{\mathbf{T}}$, and solving for Z_l,

$$Z_l = f_l \frac{(f_r \mathbf{R}_1 - x_r \mathbf{R}_3)^\top \hat{\mathbf{T}}}{(f_r \mathbf{R}_1 - x_r \mathbf{R}_3)^\top \mathbf{p}_l}. \tag{7.30}$$

We can recover the other coordinates of $\mathbf{P}_l$ from (7.8), and the coordinates of $\mathbf{P}_r$ from the relation

$$\mathbf{P}_r = R(\mathbf{P}_l - \hat{\mathbf{T}}). \tag{7.31}$$

It turns out that only one of the four estimates of $(\hat{\mathbf{T}}, R)$ yields geometrically consistent (i.e., positive) Z_l and Z_r coordinates for *all* the points. The actions to take in order to determine the correct solution are detailed in the box below, which summarizes the entire algorithm.

Metric?

Algorithm EUCLID_REC

The input is formed by a set of corresponding image points in camera coordinates, with $\mathbf{p}_l$ and $\mathbf{p}_r$ a generic pair, and an estimate of the normalized essential matrix, $\hat{E}$.

1. Recover $\hat{\mathbf{T}}$ from (7.26).
2. Construct the vectors $\mathbf{w}$ from (7.27), and compute the rows of the matrix R through (7.28).
3. Reconstruct the Z_l and Z_r coordinates of each point using (7.30), (7.8) and (7.31).
4. If the signs of Z_l and Z_r of the reconstructed points are
   (a) both negative for some point, change the sign of $\hat{\mathbf{T}}$ and go to step 3;
   (b) one negative, one positive for some point, change the sign of each entry of $\hat{E}$ and go to step 2;
   (c) both positive for all points, exit.

The output is the set of reconstructed 3-D points (up to a scale factor).

☞ When implementing EUCLID_REC, make sure that the algorithm does not go through more than 4 iterations of steps 2-4 (since there are only 4 possible combinations for the unknown signs of $\hat{\mathbf{T}}$ and $\hat{E}$). Keep in mind that, in the case of very small displacements, the errors in the disparity estimates may be sufficient to make the 3-D reconstruction inconsistent; when this happens, the algorithm keeps going through steps 2-4.

7.4.3 Reconstruction up to a Projective Transformation

The aim of this section is to show that you can compute a 3-D reconstruction even in the absence of *any* information on the intrinsic and extrinsic parameters. The price to pay is that *the reconstruction is unique only up to an unknown projective transformation of the world*. The Further Readings point you to methods for determining this transformation.

Assumptions and Problem Statement

Assuming that only n point correspondences are given, with $n \geq 8$ (and therefore the location of the epipoles, $\mathbf{e}$ and $\mathbf{e}'$), compute the location of the 3-D points from their projections, $\mathbf{p}_l$ and $\mathbf{p}_r$.

☞ It is worth noticing that, if no estimates of the intrinsic and extrinsic parameters are available and nonlinear deformations can be neglected, the accuracy of the reconstruction is only affected by that of the algorithms computing the disparities, not by calibration.

The plan for this section is as follows. We show that, mapping five arbitrary scene points into the standard projective basis of P^3, and using the epipoles, the projection matrix of each camera can be explicitly recovered up to an unknown projective transformation (the one associating the standard basis to the five points selected, which is unknown as we do not know the location of the five 3-D points in camera coordinates).[10] Once the projection matrices are determined, the 3-D location of an arbitrary point in space is obtained by triangulation in projective space. You can find the essential notions of projective geometry needed to cope with all this in the Appendix, section A.4.

Determining the Projection Matrices. In order to carry out our plan, we introduce a slight change of notation. In what follows, we drop the l and r subscripts and adopt the unprimed and primed letters to indicate points in the left and right images respectively. In addition, capital letters now denote points in the projective space P^3 (four coordinates), while small letters points in P^2 (three coordinates). The 3-D space is regarded as a subset of P^3, and each image plane as a subset of P^2. This means that we regard the 3-D point $[X, Y, Z]^\top$ of $\mathbb{R}^3$ as the point $[X, Y, Z, 1]^\top$ of P^3, and a point $[x, y]^\top$ of $\mathbb{R}^2$ as the point $[x, y, 1]^\top$ of P^2. Let $\mathbf{O}$ and $\mathbf{O}'$ denote the projection centers.

We let $\mathbf{P}_1, \ldots, \mathbf{P}_n$ be the points in P^3 to be recovered from their left and right images, $\mathbf{p}_1, \ldots, \mathbf{p}_n$ and $\mathbf{p}'_i, \ldots, \mathbf{p}'_n$, and assume that, of the first five $\mathbf{P}_i$ $(\mathbf{P}_1, \mathbf{P}_2, \ldots \mathbf{P}_5)$, no three are collinear and no four are coplanar.

We first show that, if we choose $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_5$ as the standard projective basis of P^3 (see Appendix, section A.4), *each projection matrix can be determined up to a projective factor that depends on the location of the epipoles*. Since a *spatial projective transformation* is fixed if the destiny of five points is known, we can, without losing generality, set up a projective transformation that sends $\mathbf{P}_1, \mathbf{P}_2, \ldots \mathbf{P}_5$ into the standard projective basis of P^3, $\mathbf{P}_1 = [1, 0, 0, 0]^\top$, $\mathbf{P}_2 = [0, 1, 0, 0]^\top$, $\mathbf{P}_3 = [0, 0, 1, 0]^\top$, $\mathbf{P}_4 = [0, 0, 0, 1]^\top$, and $\mathbf{P}_5 = [1, 1, 1, 1]^\top$.

For the corresponding image points $\mathbf{p}_i$ in the left camera, we can write

$$M\mathbf{P}_i = \rho_i \mathbf{p}_i, \tag{7.32}$$

where M is the projection matrix and $\rho_i \neq 0$. Similarly, since a *planar projective transformation* is fixed if the destiny of four points is known, we can also set up a projective transformation that sends the first four $\mathbf{p}_i$ into the standard projective basis of P^2, that is, $\mathbf{p}_1 = (1, 0, 0)^\top$, $\mathbf{p}_2 = (0, 1, 0)^\top$, $\mathbf{p}_3 = (0, 0, 1)^\top$, and $\mathbf{p}_4 = (1, 1, 1)^\top$. In what follows, it is assumed that the coordinates of the fifth point, $\mathbf{p}_5$, of the epipole $\mathbf{e}$, and of any other image point, $\mathbf{p}_i$, are obtained applying this transformation to their *old* coordinates.

The purpose of all this is to simplify the expression of the projection matrix: substituting $\mathbf{P}_1, \ldots, \mathbf{P}_4$ and $\mathbf{p}_1, \ldots, \mathbf{p}_4$ into (7.32), we see that the matrix M can be rewritten as

$$M = \begin{bmatrix} \rho_1 & 0 & 0 & \rho_4 \\ 0 & \rho_2 & 0 & \rho_4 \\ 0 & 0 & \rho_3 & \rho_4 \end{bmatrix}. \tag{7.33}$$

[10] You should convince yourself that knowing the locations of the five points *in the camera reference frame* amounts to camera calibration, which rather defeats the point of uncalibrated stereo.

Let $[\alpha, \beta, \gamma]^\top$ be the coordinates of $\mathbf{p}_5$ in the standard basis; (7.32) with $i = 5$ makes it possible to eliminate ρ_1, ρ_2 and ρ_3 from (7.33), obtaining

$$M = \begin{bmatrix} \alpha\rho_5 - \rho_4 & 0 & 0 & \rho_4 \\ 0 & \beta\rho_5 - \rho_4 & 0 & \rho_4 \\ 0 & 0 & \gamma\rho_5 - \rho_4 & \rho_4 \end{bmatrix}. \tag{7.34}$$

Finally, since a projection matrix is defined only up to a scale factor, we can divide each entry of matrix (7.34) by ρ_4, obtaining

$$M = \begin{bmatrix} \alpha x - 1 & 0 & 0 & 1 \\ 0 & \beta x - 1 & 0 & 1 \\ 0 & 0 & \gamma x - 1 & 1 \end{bmatrix}. \tag{7.35}$$

where $x = \rho_5/\rho_4$. *The projection matrix of the left camera has been determined up to the unknown projective parameter x.*

In order to determine x, it is useful to relate the entries of M to the coordinates of the projection center, $\mathbf{O}$. This can be done by observing that M models a perspective projection with $\mathbf{O}$ as projection center. Therefore, M *projects* every point of P^3, with the exception of $\mathbf{O}$, into a point of P^2. Since M has rank 3, the null space of M is nontrivial and consists necessarily of $\mathbf{O}$:

$$M\mathbf{O} = 0. \tag{7.36}$$

Equation (7.36) can be solved for O_x, O_y and O_z:

$$\mathbf{O} = \left[\frac{1}{1-\alpha x}, \frac{1}{1-\beta x}, \frac{1}{1-\gamma x}, 1\right]^\top. \tag{7.37}$$

Corresponding relations and results can be obtained for the right camera (in the primed reference frame). In particular, we can write

$$M' = \begin{bmatrix} \alpha' x' - 1 & 0 & 0 & 1 \\ 0 & \beta' x' - 1 & 0 & 1 \\ 0 & 0 & \gamma' x' - 1 & 1 \end{bmatrix}.$$

and

$$\mathbf{O}' = \left[\frac{1}{1-\alpha' x'}, \frac{1}{1-\beta' x'}, \frac{1}{1-\gamma' x'}, 1\right]^\top. \tag{7.38}$$

Since the location of the epipoles is known, x and x' (and hence the full projection matrices and the centers of projection) can be determined from

$$M\mathbf{O}' = \sigma\mathbf{e} \tag{7.39}$$

and

$$M'\mathbf{O} = \sigma'\mathbf{e}' \tag{7.40}$$

with $\sigma \neq 0$ and $\sigma' \neq 0$.[11]

Let's see first what we can recover from (7.39). Substituting (7.35) and (7.38) into (7.39), we obtain the following system of equations

$$\begin{bmatrix} \alpha & -\alpha' & \alpha' e_x \\ \beta & -\beta' & \beta' e_y \\ \gamma & -\gamma' & \gamma' e_z \end{bmatrix} \begin{pmatrix} x \\ x' \\ \sigma x' \end{pmatrix} = \begin{pmatrix} \sigma e_x \\ \sigma e_y \\ \sigma e_z \end{pmatrix} \tag{7.41}$$

Since σ is unknown, system (7.41) is homogeneous and nonlinear in the three unknown x, x', and σ. However, we can regard it as a linear system in the unknown x, x' and $\sigma x'$, so that solving for $\sigma x'$ we have

$$\sigma x' = \sigma \frac{\mathbf{e}^\top(\mathbf{p}_5 \times \mathbf{p}_5')}{\mathbf{v}^\top(\mathbf{p}_5 \times \mathbf{p}_5')} \tag{7.42}$$

with $\mathbf{v} = (\alpha' e_x, \beta' e_y, \gamma' e_z)$. Since $\mathbf{e}$, $\mathbf{p}_5$, $\mathbf{p}_5'$, and $\mathbf{v}$ are known and the unknown factor σ cancels out, (7.42) actually determines x'.

A similar derivation applied to (7.40) yields

$$x = \frac{\mathbf{e}'^\top(\mathbf{p}_5 \times \mathbf{p}_5')}{\mathbf{v}'^\top(\mathbf{p}_5 \times \mathbf{p}_5')} \tag{7.43}$$

with $\mathbf{v}' = (\alpha e_x', \beta e_y', \gamma e_z')$. Having determined both x and x' we can regard both the projection matrices and the centers of projections as completely determined.

Computing the Projective Reconstruction. We are now in a position to reconstruct *any* point in P^3 given its corresponding image points, $\mathbf{p} = [p_x, p_y, p_z]^\top$ and $\mathbf{p}' = [p_x', p_y', p_z']^\top$. The reconstruction is unique up to the unknown projective transformation fixed by the choice of $\mathbf{P}_1, \ldots, \mathbf{P}_5$ as the standard basis for P^3. Observe that the projective line l defined by

$$\lambda\mathbf{O} + \mu\left[O_x p_x, O_y p_y, O_z p_z, 0\right]^\top, \tag{7.44}$$

with $\lambda, \mu \in \mathbb{R}$ and not both 0, goes through $\mathbf{O}$ (for $\lambda = 1$ and $\mu = 0$) and also through $\mathbf{p}$, since

$$M \begin{pmatrix} O_x p_x \\ O_y p_y \\ O_z p_z \\ 0 \end{pmatrix} = \mathbf{p}.$$

[11] Since the epipoles and the centers of projection lie on a straight line, (7.39) and (7.40) are not independent. For the purpose of this brief introduction, however, this can be safely ignored.

Similarly, the projective line l'

$$\lambda' \mathbf{O}' + \mu' \left[O'_x p'_x, O'_y p'_y, O'_z p'_z, 0 \right]^\top ,$$

with $\lambda', \mu' \in \mathbb{R}$ and not both 0, goes through $\mathbf{O}'$ and $\mathbf{p}'$. *The projective point* $\mathbf{P}$ *can thus be obtained by intersecting the two projective lines* l *and* l'. This amounts to looking for the non-trivial solution of the homogeneous system of linear equations

$$\begin{bmatrix} O_x & O_x p_x & -O'_x & -O'_x p'_x \\ O_y & O_y p_y & -O'_y & -O'_y p'_y \\ O_z & O_z p_z & -O'_z & -O'_z p'_z \\ 1 & 0 & -1 & 0 \end{bmatrix} \begin{pmatrix} \lambda \\ \mu \\ \lambda' \\ \mu' \end{pmatrix} = 0. \tag{7.45}$$

☞ Once again, singular value decomposition $UDV^\top$ of the system matrix of (7.45) provides a numerically stable procedure for solving this linear system: The solution is given by the column of V associated with the smallest singular value along the diagonal of D.

Algorithm UNCAL_STEREO

The input is formed by n pairs of corresponding points, $\mathbf{p}_i$ and $\mathbf{p}'_i$, with $i =, \dots, n$ and $n \geq 5$, images of n points, $\mathbf{P}_1, \dots, \mathbf{P}_n$. We assume that, of the first five $\mathbf{P}_i$ ($\mathbf{P}_1, \mathbf{P}_2, \dots \mathbf{P}_5$), no three are collinear and no four are coplanar.

We assume to have estimated the location of the epipoles, $\mathbf{e}$ and $\mathbf{e}'$, using EPIPOLES_LOCATION. Let $\mathbf{P}_1, \dots, \mathbf{P}_5$ be the standard projective basis of P^3. We assume the same notation used throughout the section.

1. Determine the planar projective transformations T and T' that map the $\mathbf{p}_i$ and $\mathbf{p}'_i$ ($i = 1, \dots, 4$) into the standard projective basis of P^2 on each image plane. Apply T to the $\mathbf{p}_i$ and the epipole $\mathbf{e}$, and T' to the $\mathbf{p}'_i$ and the epipole $\mathbf{e}'$. Let (α, β, γ) and $(\alpha', \beta', \gamma')$ be the new coordinates of $\mathbf{p}_5$ and $\mathbf{p}'_5$.
2. Determine x and x' from (7.42) and (7.43).
3. Determine $\mathbf{O}$ and $\mathbf{O}'$ from (7.37) and (7.38).
4. Given a pair of corresponding points $\mathbf{p}$ and $\mathbf{p}'$, reconstruct the location of the point $\mathbf{P}$ in the standard projective basis of P^3 using (7.44) with λ and μ nontrivial solution of (7.45).

The output is formed by the coordinates of $\mathbf{P}_1, \dots, \mathbf{P}_n$ in the standard projective basis.

Having found a projective reconstruction of our points, *how do we go back to Euclidean coordinates*? If we know the location of $\mathbf{P}_1, \dots, \mathbf{P}_5$ in the world frame, we can determine the projective transformation introduced at the beginning of this section that mapped these five points, thought of as points of P^3, into the standard projective basis (see the Appendix, section A.4 for details on how to do it). The Further Readings point to (nontrivial) algorithms for Euclidean reconstruction which relax this assumption, but need more than two images.