C H A P T E R 7

Stereopsis

Fusing the pictures recorded by our two eyes and exploiting the difference (or *disparity*) between them allows us to gain a strong sense of depth. This chapter is concerned with the design and implementation of algorithms that mimic our ability to perform this task, known as *stereopsis*. Reliable computer programs for stereoscopic perception are of course invaluable in visual robot navigation (Figure 7.1), cartography, aerial reconnaissance, and close-range photogrammetry. They are also of great interest in tasks such as image segmentation for object recognition or the construction of three-dimensional scene models for computer graphics applications.

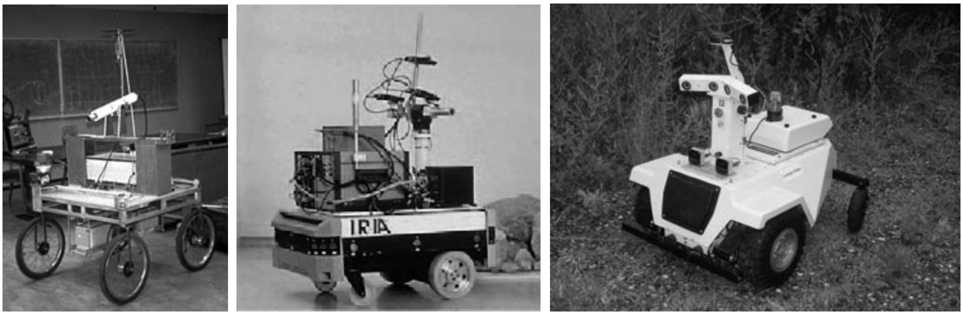

FIGURE 7.1: **Left:** The Stanford cart sports a single camera moving in discrete increments along a straight line and providing multiple snapshots of outdoor scenes. **Center:** The INRIA mobile robot uses three cameras to map its environment. **Right:** The NYU mobile robot uses two stereo cameras, each capable of delivering an image pair. As shown by these examples, although two eyes are sufficient for stereo fusion, mobile robots are sometimes equipped with three (or more) cameras. The bulk of this chapter is concerned with binocular perception but stereo algorithms using multiple cameras are discussed in Section 7.6. *Photos courtesy of Hans Moravec, Olivier Faugeras, and Yann LeCun.*

Stereo vision involves two processes: The *fusion* of features observed by two (or more) eyes and the *reconstruction* of their three-dimensional preimage. The latter is relatively simple: The preimage of matching points can (in principle) be found at the intersection of the rays passing through these points and the associated pupil centers (or pinholes; see Figure 7.2, left). Thus, when a single image feature is observed at any given time, stereo vision is easy. However, each picture typically consists of millions of pixels, with tens of thousands of image features such as edge elements, and some method must be devised to establish the correct correspondences and avoid erroneous depth measurements (Figure 7.2, right).

We start this chapter by examining in Section 7.1 the geometric *epipolar constraint* associated with a pair of cameras, which is a key to controlling the cost of the binocular fusion process. Next, we stay on the geometric side of things in Section 7.2 as we present a number of methods for binocular reconstruction. After

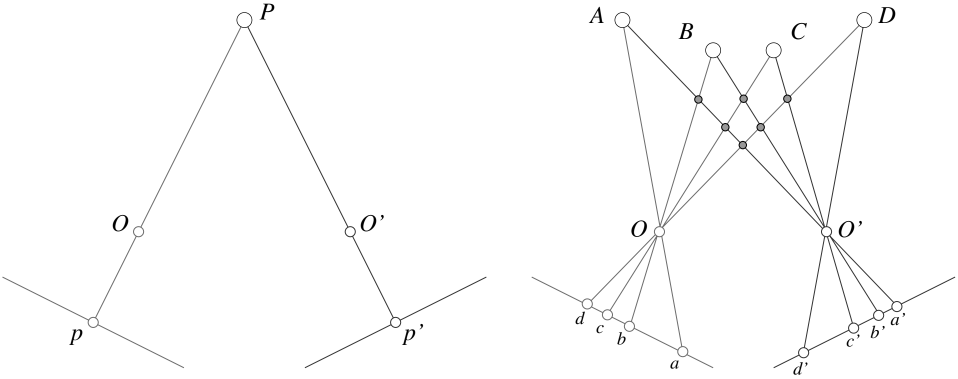

FIGURE 7.2: The binocular fusion problem: In the simple case of the diagram shown on the **left**, there is no ambiguity, and stereo reconstruction is a simple matter. In the more usual case shown on the **right**, any of the four points in the left picture may, *a priori*, match any of the four points in the right one. Only four of these correspondences are correct; the other ones yield the incorrect reconstructions shown as small gray discs.

a brief incursion into human stereopsis (Section 7.3), we switch with Section 7.4 to the presentation of several algorithms for binocular fusion that rely on the comparison of *local* brightness or edge patterns to establish correspondences. Section 7.5 shows that ordering and smoothness constraints among nearby pixels can be incorporated in the matching process. In this setting, stereo fusion is naturally cast as a combinatorial optimization problem, which can be solved by several efficient algorithms (Chapter 22). We conclude in Section 7.6 with a discussion of multi-camera stereo fusion (see also Chapter 19 for applications of multi-view stereopsis to image-based modeling and rendering).

Note: We assume throughout that all cameras have been carefully calibrated so their intrinsic and extrinsic parameters are precisely known relative to some fixed world coordinate system. The case of uncalibrated cameras is examined in the context of structure from motion in Chapter 8.

7.1 BINOCULAR CAMERA GEOMETRY AND THE EPIPOLAR CONSTRAINT

As noted in the introduction, it appears *a priori* that, given a stereo image pair, any pixel in the first (or *left*) image may match any pixel in the second (or *right*) one. As shown in this section, matching pairs of pixels are in fact restricted to lie on corresponding *epipolar lines* in the two pictures. This constraint plays a fundamental role in the stereo fusion process because it reduces the quest for image correspondences to a set of one-dimensional searches.

7.1.1 Epipolar Geometry

Consider the images p and p' of a point P observed by two cameras with optical centers O and O'. These five points all belong to the *epipolar plane* defined by the two intersecting rays OP and $O'P$ (Figure 7.3). In particular, the point p' lies on the line l' where this plane and the retina Π' of the second camera intersect.

The line l' is the *epipolar line* associated with the point p, and it passes through the point e' where the *baseline* joining the optical centers O and O' intersects Π'. Likewise, the point p lies on the epipolar line l associated with the point p', and this line passes through the intersection e of the baseline with the plane Π.

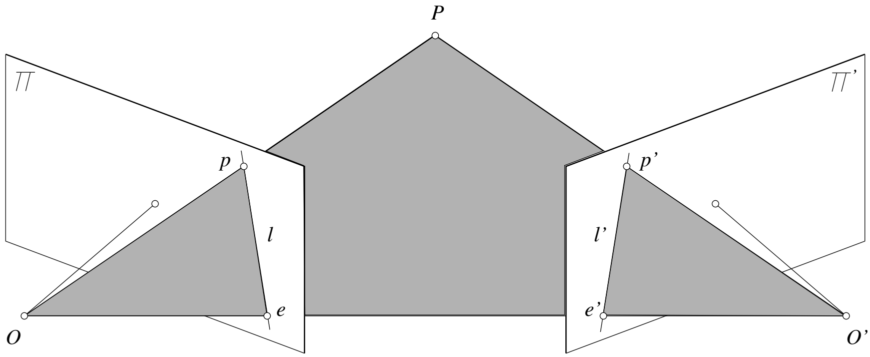

FIGURE 7.3: Epipolar geometry: The point P, the optical centers O and O' of the two cameras, and the two images p and p' of P all lie in the same plane. Here, as in the other figures of this chapter, cameras are represented by their pinholes and a *virtual* image plane located *in front* of the pinhole. This is to simplify the drawings; the geometric and algebraic arguments presented in the rest of this chapter hold just as well for *physical* image planes located *behind* the corresponding pinholes.

The points e and e' are called the *epipoles* of the two cameras. The *epipole* e' is the projection of the optical center O of the first camera in the image observed by the second camera, and vice versa. As noted before, if p and p' are images of the same point, then p' must lie on the epipolar line associated with p. This *epipolar constraint* plays a fundamental role in stereo vision and motion analysis.

In the setting studied in the rest of this chapter, where the cameras are internally and externally calibrated, the most difficult part of constructing an artifical stereo vision system is to find effective methods for establishing correspondences between the two images—that is, deciding which points in the second picture match the points in the first one. The epipolar constraint greatly limits the search for these correspondences. Indeed, since we assume that the rig is calibrated, the coordinates of the point p completely determine the ray joining O and p, and thus the associated epipolar plane $OO'p$ and epipolar line l'. The search for matches can be restricted to this line instead of the whole image (Figure 7.4). In the motion analysis setting studied in Chapter 8, each camera may be internally calibrated, but the rigid transformation separating the two camera coordinate systems is unknown. In this case, the epipolar geometry constrains the set of possible motions.

As shown next, it proves convenient to characterize the epipolar constraint in terms of the bilinear forms associated with two 3×3 *essential* and *fundamental* matrices.

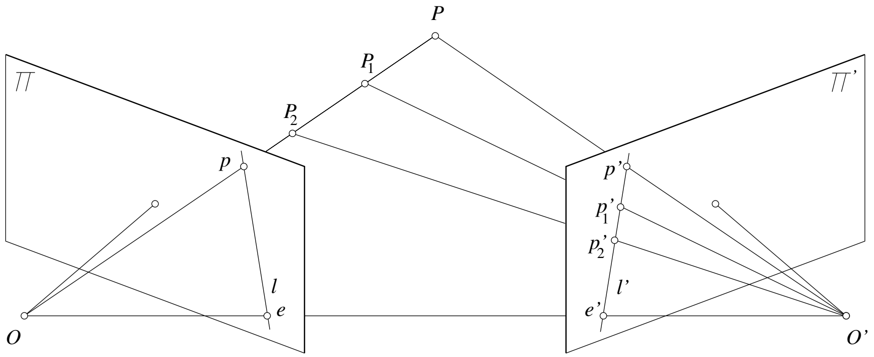

FIGURE 7.4: Epipolar constraint: Given a calibrated stereo rig, the set of possible matches for the point p is constrained to lie on the associated epipolar line l'.

7.1.2 The Essential Matrix

We assume in this section that the intrinsic parameters of each camera are known, and work in *normalized* image coordinates—that is, take $\boldsymbol{p} = \hat{\boldsymbol{p}}$. According to the epipolar constraint, the three vectors $\overrightarrow{Op}$, $\overrightarrow{O'p'}$, and $\overrightarrow{OO'}$ must be coplanar. Equivalently, one of them must lie in the plane spanned by the other two, or

$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0.$$

We can rewrite this coordinate-independent equation in the coordinate frame associated to the first camera as

$$\boldsymbol{p} \cdot [\boldsymbol{t} \times (\mathcal{R}\boldsymbol{p}')] = 0, \tag{7.1}$$

where $\boldsymbol{p}$ and $\boldsymbol{p}'$ denote the *homogeneous* normalized image coordinate vectors of p and p', $\boldsymbol{t}$ is the coordinate vector of the translation $\overrightarrow{OO'}$ separating the two coordinate systems, and $\mathcal{R}$ is the rotation matrix such that a free vector with coordinates $\boldsymbol{w}'$ in the second coordinate system has coordinates $\mathcal{R}\boldsymbol{w}'$ in the first one. In this case, the two projection matrices are given in the coordinate system attached to the first camera by $[\text{Id} \quad \mathbf{0}]$ and $[\mathcal{R}^T \quad -\mathcal{R}^T\boldsymbol{t}]$.

Equation (7.1) can finally be rewritten as

$$\boldsymbol{p}^T \mathcal{E} \boldsymbol{p}' = 0, \tag{7.2}$$

where $\mathcal{E} = [\boldsymbol{t}_\times]\mathcal{R}$, and $[\boldsymbol{a}_\times]$ denotes the skew-symmetric matrix such that $[\boldsymbol{a}_\times]\boldsymbol{x} = \boldsymbol{a} \times \boldsymbol{x}$ is the cross-product of the vectors $\boldsymbol{a}$ and $\boldsymbol{x}$. The matrix $\mathcal{E}$ is called the *essential matrix*, and it was first introduced by Longuet–Higgins (1981). Its nine coefficients are only defined up to scale, and they can be parameterized by the three degrees of freedom of the rotation matrix $\mathcal{R}$ and the two degrees of freedom defining the direction of the translation vector $\boldsymbol{t}$.

Note that $\boldsymbol{l} = \mathcal{E}\boldsymbol{p}'$ can be interpreted as the coordinate vector of the epipolar line l associated with the point p' in the first image. Indeed, Equation (7.2) can be written as $\boldsymbol{p} \cdot \boldsymbol{l} = 0$, expressing the fact that the point p lies on l. By symmetry, it is also clear that $\boldsymbol{l}' = \mathcal{E}^T \boldsymbol{p}$ is the coordinate vector representing the epipolar line l' associated with p in the second image. Essential matrices are singular because $\boldsymbol{t}$ is parallel to the coordinate vector $\boldsymbol{e}$ of the first epipole, so that $\mathcal{E}^T \boldsymbol{e} = -\mathcal{R}^T [\boldsymbol{t}_\times] \boldsymbol{e} = 0$. Likewise, it is easy to show that $\boldsymbol{e}'$ is in the nullspace of $\mathcal{E}$. As shown by Huang and Faugeras (1989), essential matrices are in fact characterized by the fact that they are singular with two equal nonzero singular values (see the problems).

7.1.3 The Fundamental Matrix

The Longuet–Higgins relation holds in normalized image coordinates. In native image coordinates, we can write $\boldsymbol{p} = \mathcal{K}\hat{\boldsymbol{p}}$ and $\boldsymbol{p}' = \mathcal{K}'\hat{\boldsymbol{p}}'$, where $\mathcal{K}$ and $\mathcal{K}'$ are the 3×3 calibration matrices associated with the two cameras. The Longuet–Higgins relation holds for these vectors, and we obtain

$$\boldsymbol{p}^T \mathcal{F} \boldsymbol{p}' = 0, \tag{7.3}$$

where the matrix $\mathcal{F} = \mathcal{K}^{-T} \mathcal{E} \mathcal{K}'^{-1}$, called the *fundamental matrix*, is not, in general, an essential matrix. It has again rank two, and the eigenvector of $\mathcal{F}$ (resp. $\mathcal{F}^T$) corresponding to its zero eigenvalue is as before the position $\boldsymbol{e}'$ (resp. $\boldsymbol{e}$) of the epipole. Likewise, $\boldsymbol{l}' = \mathcal{F}\boldsymbol{p}'$ (resp. $\boldsymbol{l} = \mathcal{F}^T \boldsymbol{p}$) represents the epipolar line corresponding to the point $\boldsymbol{p}'$ (resp. $\boldsymbol{p}$) in the first (resp. second) image.

The matrices $\mathcal{E}$ and $\mathcal{F}$ can readily be computed from the intrinsic and extrinsic parameters. Let us close this section by noting that Equations (7.2) and (7.3) also provide constraints on the entries of these matrices, *irrespective* of the 3D position of the observed points. In particular, this suggests that $\mathcal{E}$ and $\mathcal{F}$ can be computed from a sufficient number of image correspondences *without* the use of a calibration chart. We will come back to this issue in Chapter 8. For the time being, we will assume that the cameras are calibrated and that the epipolar geometry is known.

7.2 BINOCULAR RECONSTRUCTION

Given a calibrated stereo rig and two matching image points p and p', it is in principle straightforward to reconstruct the corresponding scene point by intersecting the two rays $R = Op$ and $R' = O'p'$ (Figure 7.2). However, the rays R and R' never actually intersect in practice, due to calibration and feature localization errors. In this context, various reasonable approaches to the reconstruction problem can be adopted. For example, consider the line segment perpendicular to R and R' that intersects both rays (Figure 7.5): its mid-point P is the closest point to the two rays and can be taken as the preimage of p and p'.

Alternatively, one can reconstruct a scene point using a purely algebraic approach: given the projection matrices $\mathcal{M}$ and $\mathcal{M}'$ and the matching points p and p', we can rewrite the constraints $Z\boldsymbol{p} = \mathcal{M}\boldsymbol{P}$ and $Z'\boldsymbol{p}' = \mathcal{M}\boldsymbol{P}$ as

$$\begin{cases} \boldsymbol{p} \times \mathcal{M}\boldsymbol{P} = 0 \\ \boldsymbol{p}' \times \mathcal{M}'\boldsymbol{P} = 0 \end{cases} \iff \begin{pmatrix} [\boldsymbol{p}_\times]\mathcal{M} \\ [\boldsymbol{p}'_\times]\mathcal{M}' \end{pmatrix} \boldsymbol{P} = 0.$$

This is an overconstrained system of four independent linear equations in the homogeneous coordinates of P that is easily solved using the linear least-squares techniques introduced in Chapter 22. Unlike the previous approach, this reconstruction

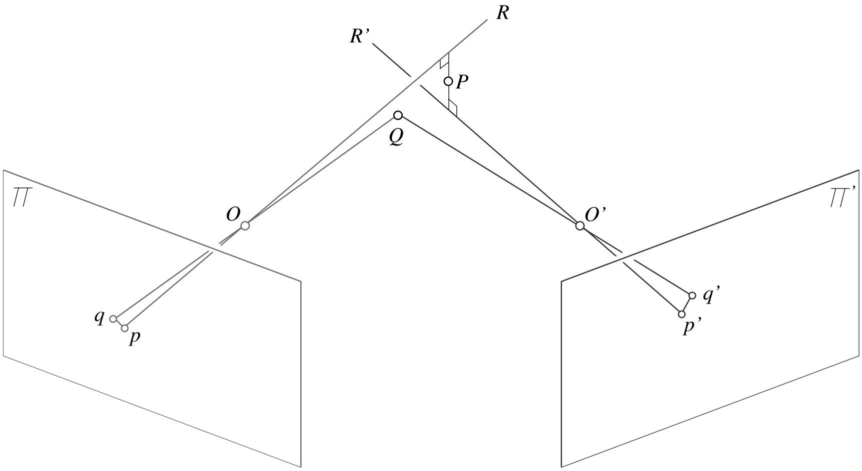

FIGURE 7.5: Triangulation in the presence of measurement errors. See text for details.

method does not have an obvious geometric interpretation, but generalizes readily to the case of three or more cameras, with each new picture simply adding two additional constraints.

Finally, one can reconstruct the scene point associated with p and p' as the point Q with images q and q' that minimizes $d^2(p,q) + d^2(p',q')$ (Figure 7.5). Unlike the two other methods presented in this section, this approach does not allow the closed-form computation of the reconstructed point, which must be estimated via nonlinear least-squares techniques such as those introduced in Chapter 22. The reconstruction obtained by either of the other two methods can be used as a reasonable guess to initialize the optimization process. This nonlinear approach also readily generalizes to the case of multiple images.

7.2.1 Image Rectification

The calculations associated with stereo algorithms are often considerably simplified when the images of interest have been *rectified*—that is, replaced by two equivalent pictures with a common image plane parallel to the baseline joining the two optical centers (Figure 7.6). The rectification process can be implemented by projecting the original pictures onto the new image plane. With an appropriate choice of coordinate system, the rectified epipolar lines are scanlines of the new images, and they are also parallel to the baseline. There are two degrees of freedom involved in the choice of the rectified image plane: (a) the distance between this plane and the baseline, which is essentially irrelevant because modifying it only changes the scale of the rectified pictures—an effect easily balanced by an inverse scaling of the image coordinate axes; and (b) the direction of the rectified plane normal in the plane perpendicular to the baseline. Natural choices include picking a plane parallel to the line where the two original retinas intersect and minimizing the distortion associated with the reprojection process.

In the case of rectified images, the informal notion of disparity introduced at

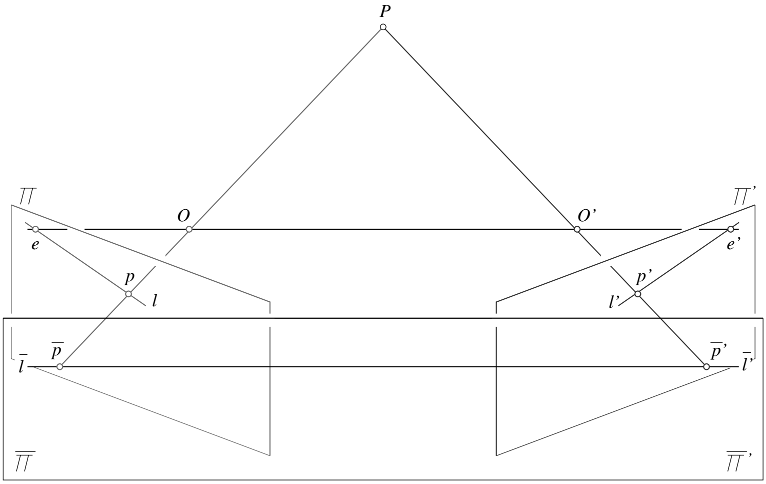

FIGURE 7.6: A rectified stereo pair: The two image planes Π and Π' are reprojected onto a common plane $\bar{\Pi} = \bar{\Pi}'$ parallel to the baseline. The epipolar lines l and l' associated with the points p and p' in the two pictures map onto a common scanline $\bar{l} = \bar{l}'$ also parallel to the baseline and passing through the reprojected points $\bar{p}$ and $\bar{p}'$. With modern computer graphics hardware and software, the rectified images are easily constructed by considering each input image as a polyhedral mesh and using texture mapping to render the projection of this mesh onto the plane $\bar{\Pi} = \bar{\Pi}'$.

the beginning of this chapter takes a concrete meaning: given two points p and p' located on the same scanline of the left and right images, with coordinates (x, y) and (x', y), the disparity is defined as the difference $d = x' - x$. We assume in the rest of this section that image coordinates are normalized—that is, as before, $\boldsymbol{p} = \hat{\boldsymbol{p}}$. As shown in the problems, if B denotes the distance between the optical centers, also called the baseline in this context, the depth of P in the (normalized) coordinate system attached to the first camera is $Z = -B/d$. In particular, the coordinate vector of the point P in the frame attached to the first camera is $\boldsymbol{P} = -(B/d)\boldsymbol{p}$, where $\boldsymbol{p} = (x, y, 1)^T$ is the vector of normalized image coordinates of p. This provides yet another reconstruction method for rectified stereo pairs.

7.3 HUMAN STEREOPSIS

Before moving on to algorithms for establishing binocular correspondences, let us pause for a moment to discuss the mechanisms underlying human stereopsis. First, it should be noted that, unlike the cameras rigidly attached to a passive stereo rig, the two eyes of a person can rotate in their sockets. At each instant, they *fixate* on a particular point in space (i.e., they rotate so that the corresponding images form in the centers of their foveas). Figure 7.7 illustrates a simplified, two-dimensional situation: if l and r denote the (counterclockwise) angles between the vertical planes of symmetry of two eyes and two rays passing through the same scene point, we define the corresponding disparity as $d = r - l$. It is an elementary exercise in trigonometry to show that $d = D - F$, where D denotes the angle between these

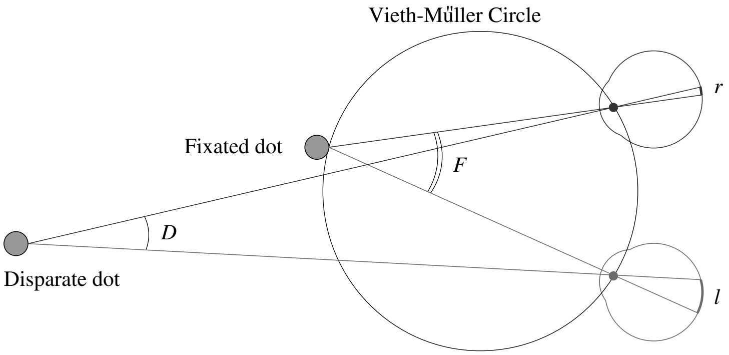

FIGURE 7.7: In this diagram, the close-by dot is fixated by the eyes, and it projects onto the center of their foveas with no disparity. The two images of the far dot deviate from this central position by different amounts, indicating a different depth.

rays, and F is the angle between the two rays passing through the fixated point. Points with zero disparity lie on the *Vieth–Müller circle* that passes through the fixated point and the optical centers of the eyes. Points lying inside this circle have a positive disparity, points lying outside it have, as in Figure 7.7, a negative disparity, and the locus of all points having a given disparity d forms, as d varies, the family of all circles passing through the two eyes' optical centers. This property is clearly sufficient to rank order dots that are near the fixation point according to their depth. However, it is also clear that the *vergence angles* between the vertical *median plane* of symmetry of the head and the two fixation rays must be known to reconstruct the absolute position of scene points.

The three-dimensional case is naturally more complicated, with the locus of zero-disparity points becoming a surface, the *horopter*, but the general conclusion is the same, and absolute positioning requires the vergence angles. There is some evidence that these angles cannot be measured accurately by our nervous system (Helmholtz 1909). However, *relative* depth, or rank ordering of points along the line of sight, can be judged quite accurately. For example, it is possible to decide which one of two targets near the horopter is closer to an observer for disparities of a few seconds of arc (*stereoacuity threshold*), which matches the minimum separation that can be measured with one eye (*monocular hyperacuity threshold*).

Concerning the construction of correspondences between the left and right images, Julesz (1960) asked the following question: Is the basic mechanism for binocular fusion a monocular process (where local brightness patterns [micropatterns] or higher organizations of points into objects [macropatterns] are identified *before* being fused), a binocular one (where the two images are combined into a single field where all further processing takes place), or a combination of both? To settle this matter, he introduced a new device, the *random dot stereogram*: a pair of synthetic images obtained by randomly spraying black dots on white objects, typically one (or several) small square plate(s) floating over a larger one (Figure 7.8). The results were striking. To quote Julesz: "When viewed monocularly, the images appear completely random. But when viewed stereoscopically, the image pair

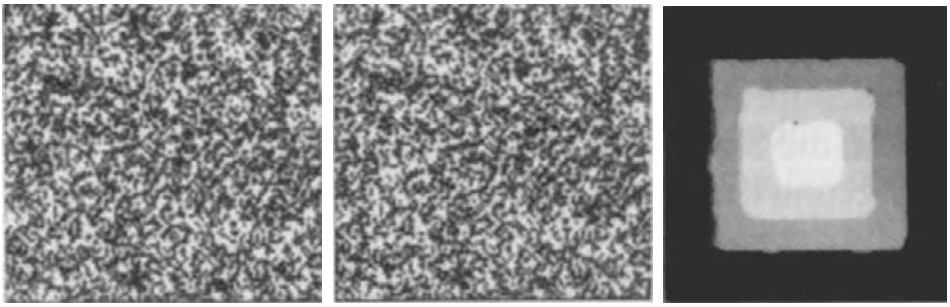

FIGURE 7.8: From **left** to **right**: the two pictures forming a random dot stereogram that depicts four planes at varying depth (a “wedding cake”), and the disparity map obtained by the Marr-Poggio (1976) algorithm. The layered structure of the scene is correctly recovered. *Reprinted from Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, by David Marr, © 1982 by David Marr. Reprinted by permission of Henry Holt and Company, LLC.*

gives the impression of a square markedly in front of (or behind) the surround.” The conclusion is clear: Human binocular fusion cannot be explained by peripheral processes directly associated with the physical retinas. Instead, it must involve the central nervous system and an imaginary *cyclopean retina* that combines the left and right image stimuli as a single unit.

Several *cooperative* models of human stereopsis—where near-by matches influence each other to avoid ambiguities and promote a global scene analysis—have been proposed, including Julesz’s own *dipole* model (1960) and that of Marr and Poggio (1976). Although the latter has been implemented, allowing the reliable fusion of random dot stereograms (Figure 7.8), it fails on most natural images. In contrast, the algorithms proposed in the following sections do not attempt to model the human visual system, but they usually give good results on natural imagery.

7.4 LOCAL METHODS FOR BINOCULAR FUSION

We start here by introducing simple methods for stereo fusion that exploit purely local information, such as the similarity of brightness patterns near candidate matches, to establish correspondences.

7.4.1 Correlation

Correlation methods find pixel-wise image correspondences by comparing intensity profiles in the neighborhood of potential matches, and they are among the first techniques ever proposed to solve the binocular fusion problem (Kelly, McConnell & Mildenberger 1977; Gennery 1980). Concretely, let us consider a *rectified* stereo pair and a point (x, y) in the first image (Figure 7.9). We associate with the window of size $p = (2m+1) \times (2n+1)$ centered in (x, y) the vector $\boldsymbol{w}(x, y) \in \mathbb{R}^p$ obtained by scanning the window values one row at a time (the order is in fact irrelevant as long as it is fixed). Now, given a potential match $(x+d, y)$ in the second image, we can construct a second vector $\boldsymbol{w}'(x+d, y)$ and define the corresponding *normalized*

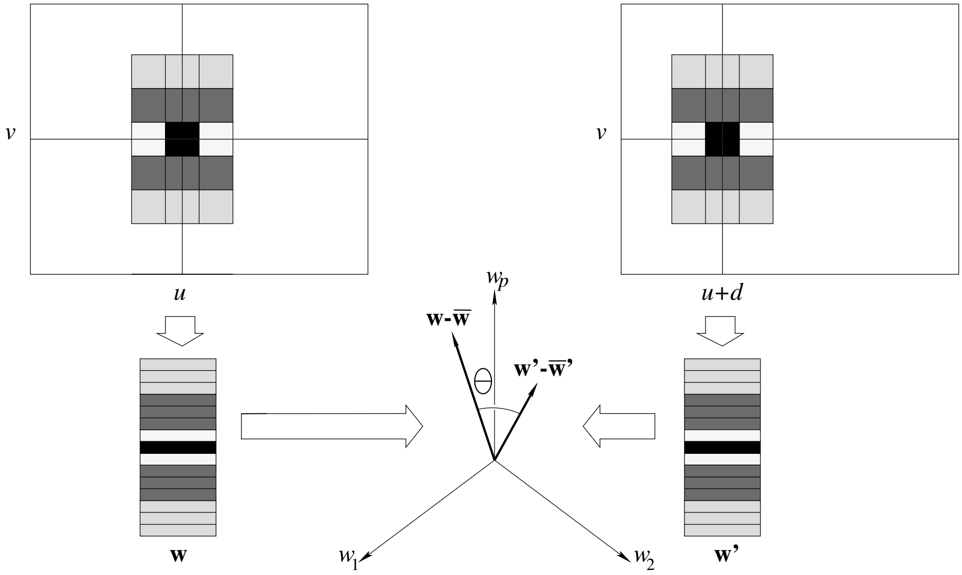

FIGURE 7.9: Correlation of two 3 × 5 windows along corresponding epipolar lines. The second window position is separated from the first one by an offset d. The two windows are encoded by vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$ in $\mathbb{R}^{15}$, and the correlation function measures the cosine of the angle θ between the vectors $\boldsymbol{w} - \bar{\boldsymbol{w}}$ and $\boldsymbol{w}' - \bar{\boldsymbol{w}}'$ obtained by subtracting from the components of $\boldsymbol{w}$ and $\boldsymbol{w}'$ the average intensity in the corresponding windows.

correlation function as

$$C(d) = \frac{1}{||\boldsymbol{w} - \bar{\boldsymbol{w}}||} \frac{1}{||\boldsymbol{w}' - \bar{\boldsymbol{w}}'||} [(\boldsymbol{w} - \bar{\boldsymbol{w}}) \cdot (\boldsymbol{w}' - \bar{\boldsymbol{w}}')],$$

where the x, y, and d indexes have been omitted for the sake of conciseness and $\bar{\boldsymbol{a}}$ denotes the vector whose coordinates are all equal to the mean of the coordinates of $\boldsymbol{a}$.

The normalized correlation function C clearly ranges from -1 to $+1$. It reaches its maximum value when the image brightnesses of the two windows are related by an affine transformation $I' = \lambda I + \mu$ for some constants λ and μ with $\lambda > 0$ (see the problems). In other words, maxima of this function correspond to image patches separated by a constant offset and a positive scale factor, and stereo matches can be found by seeking the maximum of the C function over some predetermined range of disparities.[1]

At this point, let us make a few remarks about matching methods based on correlation. First, it is easily shown (see the problems) that maximizing the

[1] The invariance of C to affine transformations of the brightness function affords correlation-based matching techniques some degree of robustness in situations where the observed surface is not quite Lambertian or the two cameras have different gains or lenses with different f-numbers.

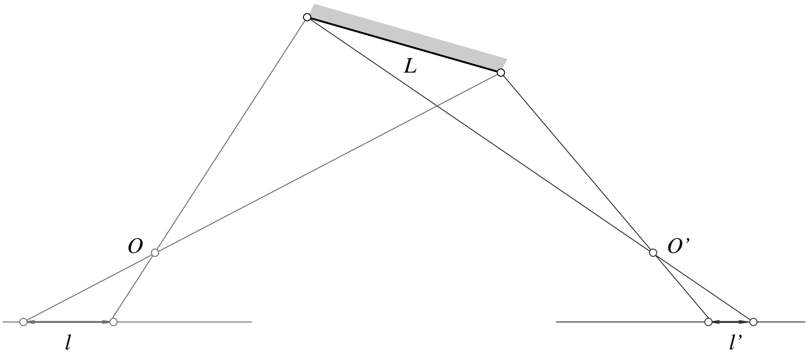

FIGURE 7.10: The foreshortening of an oblique plane is not the same for the left and right cameras: $l/L \neq l'/L$.

correlation function is equivalent to minimizing

$$|\frac{1}{||\boldsymbol{w}-\bar{\boldsymbol{w}}||}(\boldsymbol{w}-\bar{\boldsymbol{w}})-\frac{1}{||\boldsymbol{w}'-\bar{\boldsymbol{w}}'||}(\boldsymbol{w}'-\bar{\boldsymbol{w}}')|^2,$$

or equivalently the sum of the squared differences between the pixel values of the two windows after they have been submitted to the corresponding normalization process. Second, although the calculation of the normalized correlation function at every pixel of an image for some range of disparities is computationally expensive, it can be implemented efficiently using recursive techniques (see problems). Third, other functions, such as the *sum of absolute difference* $\sum_{i=1}^{p}|w_i - w_i'|$, can be used to measure the discrepancy between two brightness patterns, and they may give better results in certain situations (Scharstein and Szeliski 2002). Finally, a major problem with correlation-based techniques for establishing stereo correspondences is that they implicitly assume that the observed surface is (locally) parallel to the two image planes, since the foreshortening of (oblique) surfaces depends on the position of the cameras observing them (Figure 7.10).

This suggests a two-pass algorithm where initial estimates of the disparity are used to warp the correlation windows to compensate for unequal amounts of foreshortening in the two pictures. For example, Devernay and Faugeras (1994) propose to define a warped window in the right image for each rectangle in the left one, using the disparity in the center of the rectangle and its derivatives. An optimization process is used to find the values of the disparity and its derivatives that maximize the correlation between the left rectangle and the right window, using interpolation to retrieve appropriate values in the right image. Figure 7.11 illustrates this approach with an example.

7.4.2 Multi-Scale Edge Matching

Slanted surfaces pose problems to correlation-based matchers. Other arguments against correlation can be found in Julesz (1960) and Marr (1982), suggesting that correspondences should be found at a variety of scales, with matches between (hopefully) physically significant image features such as edges preferred to matches be-

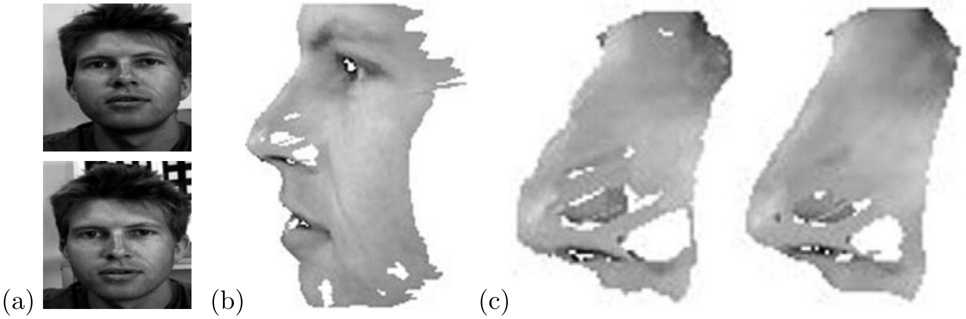

FIGURE 7.11: Correlation-based stereo matching: (a) a pair of stereo pictures; (b) a texture-mapped view of the reconstructed surface; (c) comparison of the regular (**left**) and refined (**right**) correlation methods in the nose region. The latter clearly gives better results. *Reprinted from "Computing Differential Properties of 3D Shapes from Stereopsis Without 3D Models," by F. Devernay and O.D. Faugeras, Proc. IEEE Conference on Computer Vision and Pattern Recognition, (1994). © 1994 IEEE.*

tween raw pixel intensities. These principles are implemented in Algorithm 7.1, which is due to Marr and Poggio (1979).

1. Convolve the two (rectified) images with $\nabla^2 G_\sigma$ filters of increasing standard deviations $\sigma_1 < \sigma_2 < \sigma_3 < \sigma_4$.
2. Find zero crossings of the Laplacian along horizontal scanlines of the filtered images.
3. For each filter scale σ, match zero crossings with the same parity and roughly equal orientations in a $[-w_\sigma, +w_\sigma]$ disparity range, with $w_\sigma = 2\sqrt{2}\sigma$.
4. Use the disparities found at larger scales to offset the images in the neighborhood of matches and cause unmatched regions at smaller scales to come into correspondence.

Algorithm 7.1: The Marr–Poggio (1979) Multi-Scale Binocular Fusion Algorithm.

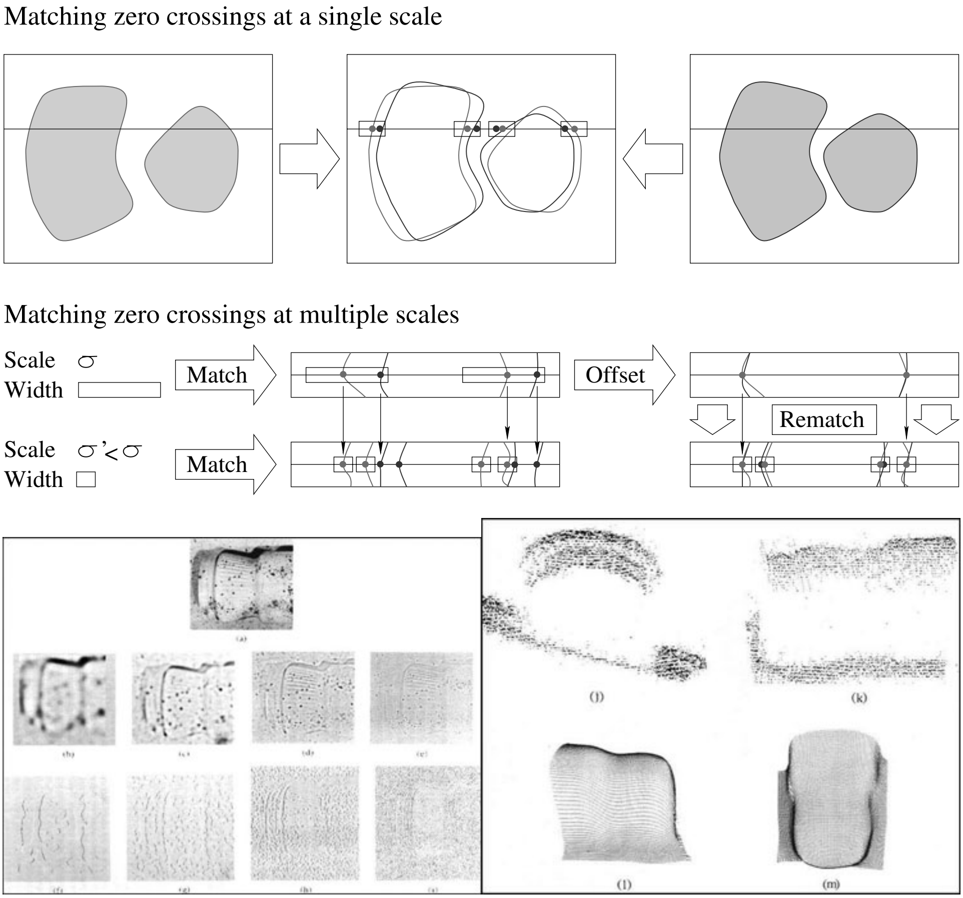

FIGURE 7.12: **Top:** Single-Scale matching. **Middle:** Multi-Scale matching. **Bottom:** Results. **Bottom left:** The input data (including one of the input pictures, the output of four $\nabla^2 G_\sigma$ filters, and the corresponding zero crossings). **Bottom right:** Two views of the disparity map constructed by the matching process and two views of the surface obtained by interpolating the reconstructed points. *Reprinted from Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, by David Marr, © 1982 by David Marr. Reprinted by permission of Henry Holt and Company, LLC.*

Matches are sought at each scale in the $[-w_\sigma, w_\sigma]$ disparity range, where $w_\sigma = 2\sqrt{2}\sigma$ is the width of the central negative portion of the $\nabla^2 G_\sigma$ filter. This choice is motivated by psychophysical and statistical considerations. In particular, assuming that the convolved images are white Gaussian processes, Grimson (1981*a*) showed that the probability of a false match occurring in the $[-w_\sigma, +w_\sigma]$ disparity range of a given zero crossing is only 0.2 when the orientations of the matched features are within 30° of each other. A simple mechanism can be used to disambiguate the multiple potential matches that might still occur within the matching range. See Grimson (1981*a*) for details. Of course, limiting the search for matches to the $[-w_\sigma, +w_\sigma]$ range prevents the algorithm from matching *correct* pairs of zero crossings whose disparity falls outside this interval. Since w_σ is proportional to the scale σ at which matches are sought, eye movements (or equivalently image offsets) controlled by the disparities found at large scales must be used to bring large-disparity pairs of zero crossings within matchable range at a fine scale. This process occurs in Step 4 of Algorithm 7.1 and is illustrated by Figure 7.12 (top). Once matches have been found, the corresponding disparities can be stored in a buffer called the $2\frac{1}{2}$-*dimensional sketch* by Marr and Nishihara (1978). This algorithm has been implemented by Grimson (1981*a*), and extensively tested on random dot stereograms and natural images. An example appears in Figure 7.12 (bottom).

7.5 GLOBAL METHODS FOR BINOCULAR FUSION

The stereo fusion techniques presented in the previous section are purely local, in the sense that they match brightness or edge patterns around individual pixels, but ignore the constraints that may link nearby points. In contrast, we present in this section two *global* approaches to stereo fusion, that formulate this problem as the minimization of a single energy function incorporating *ordering* or *smoothness* constraints among adjacent pixels.

7.5.1 Ordering Constraints and Dynamic Programming

It is reasonable to assume that the order of matching image features along a pair of epipolar lines is the inverse of the order of the corresponding surface attributes along the curve where the epipolar plane intersects the observed object's boundary (Figure 7.13, left). This is the so-called *ordering constraint* introduced in the early 1980s (Baker & Binford 1981; Ohta & Kanade 1985). Interestingly enough, it might not be satisfied by real scenes, in particular when small solids occlude parts of larger ones (Figure 7.13, right) or more rarely, at least in robot vision, when transparent objects are involved. Despite these reservations, the ordering constraint remains a reasonable one, and it can be used to devise efficient algorithms relying on *dynamic programming* (Forney 1973; Aho, Hopcroft, & Ullman 1974) to establish stereo correspondences (see Figure 7.14 and Algorithm 7.2).

Specifically, let us assume that a number of feature points (say, edgels) have been found on corresponding epipolar lines. Our objective here is to match the intervals separating those points along the two intensity profiles (Figure 7.14, left). According to the ordering constraint, the order of the feature points must be the same, although the occasional interval in either image may be reduced to a single

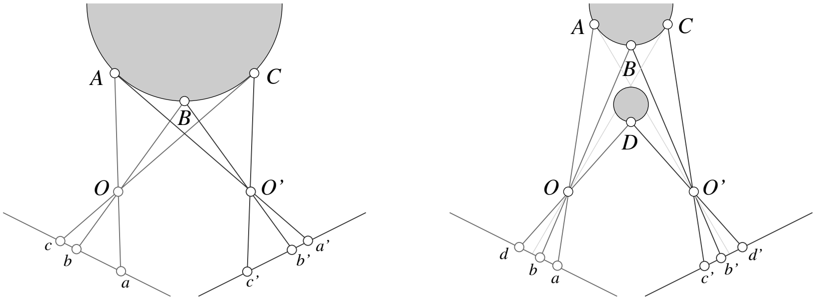

FIGURE 7.13: Ordering constraints. In the (usual) case shown in the **left** part of the diagram, the order of feature points along the two (oriented) epipolar lines is the same. In the case shown in the **right** part of the figure, a small object lies in front of a larger one. Some of the surface points are not visible in one of the images (e.g., A is not visible in the right image), and the order of the image points is not the same in the two pictures: b is on the right of d in the left image, but b' is on the left of d' in the right image.

point corresponding to missing correspondences associated with occlusion and/or noise.

This setting allows us to recast the matching problem as the optimization of a path's cost over a graph whose nodes correspond to pairs of left and right image features; and arcs represent matches between left and right intensity profile intervals bounded by the features of the corresponding nodes (Figure 7.14, right). The cost of an arc measures the discrepancy between the corresponding intervals (e.g., the squared difference of the mean intensity values). This optimization problem can be solved, exactly and efficiently, using dynamic programming (Algorithm 7.2). As given, this algorithm has a computational complexity of $O(mn)$, where m and n denote the number of edge points on the matched left and right scanlines, respectively.[2] Variants of this approach have been implemented by Baker and Binford (1981), who combine a coarse-to-fine intra-scanline search procedure with a cooperative process for enforcing inter-scanline consistency, and Ohta and Kanade (1985), who use dynamic programming for both intra- and inter-scanline optimization, the latter procedure being conducted in a three-dimensional search space. Figure 7.15 shows a sample result taken from Ohta and Kanade (1985).

7.5.2 Smoothness Constraints and Combinatorial Optimization over Graphs

Dynamic programming is a *combinatorial optimization* algorithm aimed at minimizing an error function (a path cost) over some discrete variables (correspondences between pairs of features). It was used in the previous section to incorporate ordering constraints in the matching process. We now present a different approach to stereo fusion that relies instead on smoothness constraints, and a different combinatorial optimization technique aimed at minimizing certain energy functions defined over graphs.

[2] Our version of the algorithm assumes that all edges are matched. To account for noise and edge-detection errors, it is reasonable to allow the matching algorithm to skip a bounded number of edges, but this does not change its asymptotic complexity (Ohta and Kanade 1985).

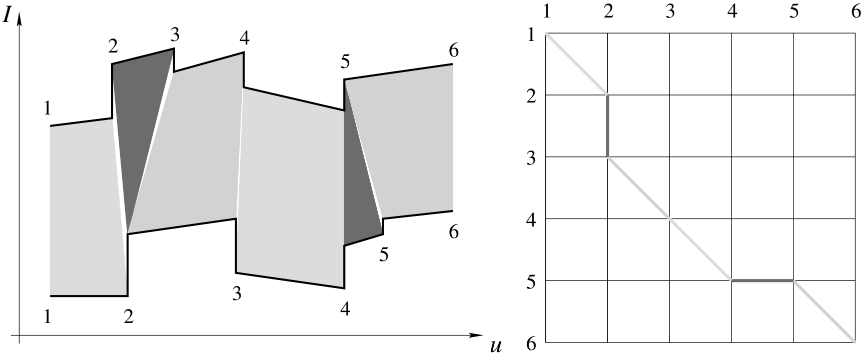

FIGURE 7.14: Dynamic programming and stereopsis: The **left** part of the figure shows two intensity profiles along matching epipolar lines. The polygons joining the two profiles indicate matches between successive intervals (some of the matched intervals may have zero length). The **right** part of the diagram represents the same information in graphical form: an arc (thick line segment) joins two nodes (i, i') and (j, j') when the intervals (i, j) and (i', j') of the intensity profiles match each other.

We assume the scanlines have m and n edge points, respectively (the endpoints of the scanlines are included for convenience). Two auxiliary functions are used: Inferior-Neighbors(k, l) returns the list of neighbors (i, j) of the node (k, l) such that $i \le k$ and $j \le l$, and Arc-Cost(i, j, k, l) evaluates and returns the cost of matching the intervals (i, k) and (j, l). For correctness, $C(1, 1)$ should be initialized with a value of zero.

```
% Loop over all nodes $(k, l)$ in ascending order.
for $k = 1$ to $m$ do
   for $l = 1$ to $n$ do
      % Initialize optimal cost $C(k, l)$ and backward pointer $B(k, l)$.
      $C(k, l) \leftarrow +\infty; B(k, l) \leftarrow$ nil;
      % Loop over all inferior neighbors $(i, j)$ of $(k, l)$.
      for $(i, j) \in$ Inferior-Neighbors$(k, l)$ do
         % Compute new path cost and update backward pointer if necessary.
         $d \leftarrow C(i, j) +$ Arc-Cost$(i, j, k, l)$;
         if $d < C(k, l)$ then $C(k, l) \leftarrow d; B(k, l) \leftarrow (i, j)$ endif;
         endfor;
      endfor;
   endfor;
% Construct optimal path by following backward pointers from $(m, n)$.
$P \leftarrow \{(m, n)\}; (i, j) \leftarrow (m, n)$;
while $B(i, j) \neq$ nil do $(i, j) \leftarrow B(i, j); P \leftarrow \{(i, j)\} \cup P$ endwhile.
```

Algorithm 7.2: A Dynamic-Programming Algorithm for Establishing Stereo Correspondences Between Two Corresponding Scanlines.

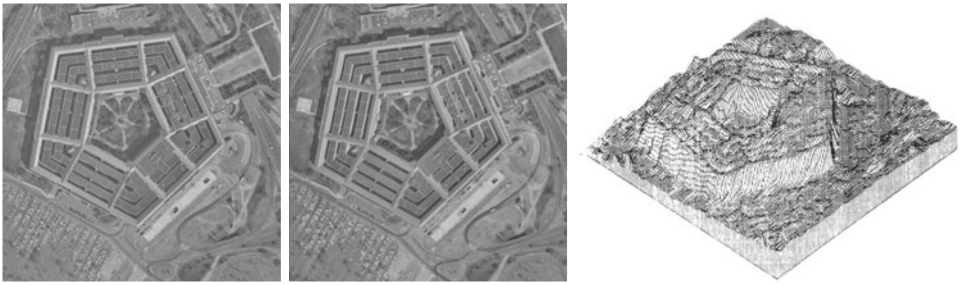

FIGURE 7.15: Two images of the Pentagon and an isometric plot of the disparity map computed by the dynamic-programming algorithm of Ohta and Kanade (1985). *Reprinted from "Stereo by Intra- and Inter-Scanline Search," by Y. Ohta and T. Kanade, IEEE Transactions on Pattern Analysis and Machine Intelligence, 7(2):139–154, (1985). © 1985 IEEE.*

Let us assume as usual that the two input images have been rectified, and define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose n nodes are the pixels of the first image and whose edges link pairs of adjacent pixels on the image grid (not necessarily on the same scanline). Given some allowed disparity range $\mathcal{D} = \{-K, \ldots, K\} \subset \mathbb{Z}$, we can define an energy function $E : \mathcal{D}^n \to \mathbb{R}$ by

$$E(\boldsymbol{d}) = \sum_{p \in \mathcal{V}} U_p(d_p) + \sum_{(p,q) \in \mathcal{E}} B_{pq}(d_p, d_q), \tag{7.4}$$

where $\boldsymbol{d}$ is a vector of n integer disparities d_p associated with pixels p, $U_p(d_p)$ (*unary term*) measures the discrepancy between pixel p in the left image and pixel $p + d_p$ in the second one, and $B_{pq}(d_p, d_q)$ (*binary term*) measures the discrepancy between the pair of assignments $p \to p + d_p$ and $q \to q + d_q$.[3] The first of these terms records the similarity between p and $p + dp$. It may be, for example, the sum of squared differences $U_p(d_p) = \sum_{q \in \mathcal{N}(p)} [I(q) - I'(q + dp)]^2$, where $\mathcal{N}(p)$ is some neighborhood of p. The second one is used to *regularize* the optimization process, making sure that the disparity function is smooth enough. For example, a sensible choice may be $B_{pq}(d_p, d_q) = \gamma_{pq}|d_p - d_q|$ for some $\gamma_{pq} > 0$.

Under this model, binocular fusion can be formulated as the minimization of $E(\boldsymbol{d})$ with respect to $\boldsymbol{d}$ in $\mathcal{D}^n$. As discussed in Chapter 22 (Section 22.4), this is a particular instance of a general combinatorial optimization problem, related to maximum a posteriori (MAP) inference in first-order Markov random fields (Geman and Geman 1984), which is in general NP-hard but admits effective approximate and even exact algorithmic solutions under certain so-called *submodularity* assumptions. In particular, it can be shown (Ishikawa 2003; Schlesinger & Flach 2006; Darbon 2009) that when $B_{pq}(d_p, d_q) = \gamma_{pq}|d_p - d_q|$ for some $\gamma_{pq} > 0$ (*total-variation prior*) or, more generally, when $B_{pq} = g(d_p - d_q)$ for some convex real function $g : \mathbb{Z} \to \mathbb{R}$, minimizing $E(\boldsymbol{d})$ reduces to a submodular *quadratic pseudo-Boolean problem* that involves only binary variables and can be solved *exactly* in polynomial

[3] Here we abuse the notation and, if the images coordinates of pixel p are (u_p, v_p), denote by $p + d_p$ the pixel with coordinates $(u_p + d_p, v_p)$.

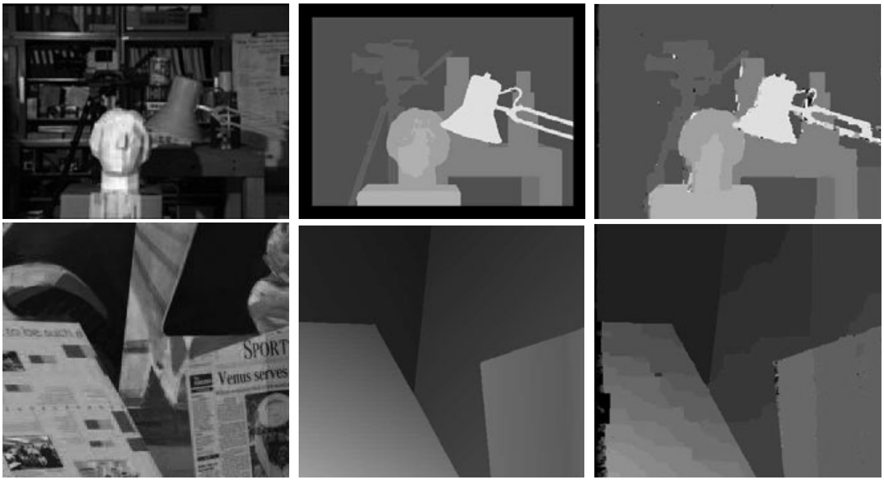

FIGURE 7.16: An application of alpha expansion to stereo fusion. The data used in this experiment is part of the benchmark described in Scharstein and Szeliski (2002), for which ground truth disparities are available. From **left** to **right**: Input image, ground truth disparities, and disparities recovered using alpha expansion. *Reprinted from "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," by D. Scharstein and R. Szeliski, International Journal of Computer Vision, 47(1/2/3):7–42, (2002). © 2002 Springer.*

time by an efficient *min-cut/max-flow* algorithm (Ford & Fulkerson 1956; Goldberg & Tarjan 1988; Boykov & Kolmogorov 2004).

In practice, however, it may prove important to use binary terms that do not lead to submodular problems and thus cannot be solved in an exact manner. The Potts model, where $B_{pq}(d_p, d_q) = \gamma_{pq}\chi(d_p \neq d_q)$, the characteristic function χ is one if its argument is true and zero otherwise, and $\gamma_{pq} > 0$, is a typical example. Using it instead of, say, a total-variation prior to encourage the disparity function to be smooth, does not overpenalize the disparity discontinuities naturally associated with occlusion boundaries. In this setting, an *approximate* solution to the minimization of $E(\boldsymbol{d})$ over $\mathcal{D}^n$ can be found using *alpha expansion* (Boykov *et al.* 2001), an iterative procedure that also solves a min-cut/max-flow problem at each step, but makes weaker assumptions on the energy function it minimizes. Figure 7.16 shows the result of an experiment using this approach, taken from Scharstein and Szeliski (2002).

7.6 USING MORE CAMERAS

Adding a third camera eliminates (in large part) the ambiguity inherent in two-view point matching. In essence, the third image can be used to check hypothetical matches between the first two pictures (Figure 7.17): The three-dimensional point associated with such a match is first reconstructed and then reprojected into the third image. If no compatible point lies nearby, then the match must be wrong.

In most trinocular stereo algorithms, potential correspondences are hypothe-

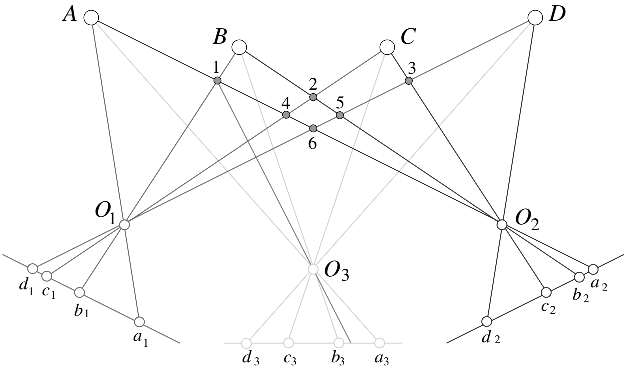

FIGURE 7.17: The small gray discs indicate the incorrect reconstructions associated with the left and right images of four points. The addition of a central camera removes the matching ambiguity: none of the corresponding rays intersects any of the six discs. Alternatively, matches between points in the first two images can be checked by reprojecting the corresponding three-dimensional point in the third image. For example, the match between b_1 and a_2 is obviously wrong because there is no feature point in the third image near the reprojection of the hypothetical reconstruction numbered 1 in the diagram.

sized using two of the images, then confirmed or rejected using the third one. In contrast, Okutami and Kanade (1993) have proposed to find matches simultaneously in three or more pictures. The basic idea is simple, but elegant: assuming that all the images have been rectified, the search for the correct disparities is replaced by a search for the correct depth, or rather its inverse. Of course, the inverse depth is proportional to the disparity for each camera, but the disparity varies from camera to camera, and the inverse depth can be used as a common search index. Picking the first image as a reference, Okutami and Kanade add the sums of squared differences associated with all other cameras into a global evaluation function E (as shown earlier, this is of course equivalent to adding the correlation functions associated with the images).

Figure 7.18 plots the value of E as a function of inverse depth for various subsets of 10 cameras observing a scene that contains a repetitive pattern (Figure 7.19). In that case, using only two or three cameras does not yield a single, well-defined minimum. However, adding more cameras provides a clear minimum corresponding to the correct match. Figure 7.19 shows a sequence of 10 rectified images and a plot of the surface reconstructed by the algorithm.

7.7 APPLICATION: ROBOT NAVIGATION

Applications of *wide-baseline* multi-view stereopsis to the construction of three-dimensional object and scene models are discussed in Chapter 19. Let us briefly discuss here an application of binocular stereo vision to navigation for the robot shown in Figure 7.1 (right). The system described in Hadsell *et al.* (2009) and Sermanet *et al.* (2009) uses two Point Grey Bumblebee stereo cameras, each capable of delivering a pair of 1024 × 768 color images at 15 frames per second, and runs a

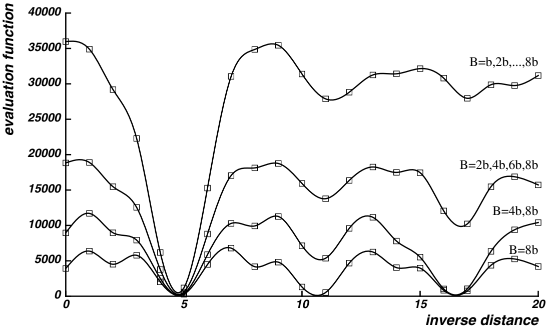

FIGURE 7.18: Combining multiple views: The sum of squared differences is plotted here as a function of the inverse depth for various numbers of input pictures. The data are taken from a scanline near the top of the images shown in Figure 7.19, whose intensity is nearly periodic. The diagram clearly shows that the minimum of the function becomes less and less ambiguous as more images are added. *Reprinted from "A Multiple-Baseline Stereo System," by M. Okutami and T. Kanade, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(4):353–363, (1993). © 1993 IEEE.*

separate binocular stereo process for each pair (Figure 7.20). The fusion algorithm itself is local, and uses the sum of absolute differences as a matching criterion, with additional heuristics to filter out outliers. The ground plane is then found by a voting procedure before obstacles are detected, based on the recovered point cloud distribution. The overall process runs at 5–10 160×120 frames per second, but its useful range is limited to 5 meters. A slower program (one 512×384 frame per second), combining stereo vision with *convolutional nets* used for classification, yields useful depth measurements for distances up to 12 meters, and detects obstacles up to 50 meters away. The overall system has been successfully used to drive the robot in field experiments with many outdoor settings, including parks and backyards, open fields, urban and suburban environments, military bases, sandy areas near beaches, forests with and without paths, etc. See (Hadsell *et al.* 2009; Sermanet *et al.* 2009) for details.

7.8 NOTES

The essential matrix as an algebraic form of the epipolar constraint was introduced in the computer vision community by Longuet-Higgins (1981), and its properties have been elucidated by Huang and Faugeras (1989). The fundamental matrix was introduced by Luong and Faugeras (1992, 1996). Just as a bilinear constraint holds for the image coordinates of two point matches, trilinear constraints hold among matching triples of points (Hartley 1997) and lines (Spetsakis & Aloimonos 1990; Weng, Huang & Ahuja 1992; Shashua 1995), and quadrilinear constraints also hold among matching quadruples of points (Faugeras and Mourrain 1995; Triggs 1995; Faugeras & Papadopoulo 1997). See the problems for some examples. Similar constraints have also been studied for decades in the photogrammetry domain (Slama *et al.* 1980).

The fact that disparity gives rise to stereopsis in human beings was first

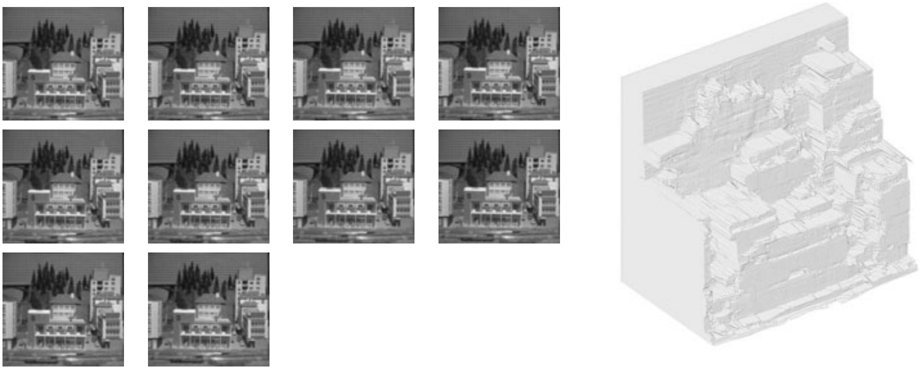

FIGURE 7.19: A series of 10 images and the corresponding reconstruction. The gridboard near the top of the images is the source for the nearly periodic brightness signal giving rise to ambiguities in Figure 7.18. *Reprinted from "A Multiple-Baseline Stereo System," by M. Okutami and T. Kanade, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(4):353–363, (1993). © 1993 IEEE.*

demonstrated by Wheatstone's (1838) invention of the stereoscope. That disparity is sufficient for stereopsis without eye movements was demonstrated shortly afterward by Dove (1841) with illumination provided by an electric spark too brief for eye vergence to take place. Human stereopsis is further discussed in the classical book of Helmholtz (1909), an amazing read for anyone interested in the history of the field, as well as the books by Julesz (1960, 1971), Frisby (1980), and Marr (1982). Theories of human binocular perception not presented in this chapter for lack of space include Koenderink and Van Doorn (1976*a*), Pollard, Mayhew, and Frisby (1970), McKee, Levi, and Brown (1990), and Anderson and Nayakama (1994).

Excellent treatments of machine stereopsis can be found in the books of Grimson (1981*b*), Marr (1982), Horn (1986), and Faugeras (1993). Marr focuses on the computational aspects of human stereo vision, whereas Horn's account emphasizes the role of photogrammetry in artificial stereo systems. Grimson and Faugeras emphasize the geometric and algorithmic aspects of stereopsis. The constraints associated with stereo matching are discussed by Binford (1984). Early techniques for line matching in binocular stereo include Medioni and Nevatia (1984) and Ayache and Faugeras (1987). Algorithms for trinocular fusion include Milenkovic and Kanade (1985), Yachida, Kitamura, and Kimachi (1986), Ayache and Lustman (1987), and Robert and Faugeras (1991). Global approaches to dense stereo fusion based on combinatorial optimization and the underlying min-cut/max-flow algorithms include Ishikawa and Geiger (1998), Roy and Cox (1998), Boykov, Veksler, and Zabih (2001), and Kolgomorov and Zabih (2001). Variational approaches have also been used in this context, see Faugeras and Keriven (1998) for example.

All of the algorithms presented in this chapter (implicitly) assume that the images being fused are quite similar. This is equivalent to considering a *narrow baseline*. The *wide-baseline* case is treated in Chapter 19 in the context of image-based modeling and rendering. We have also limited our attention here to stereo rigs with

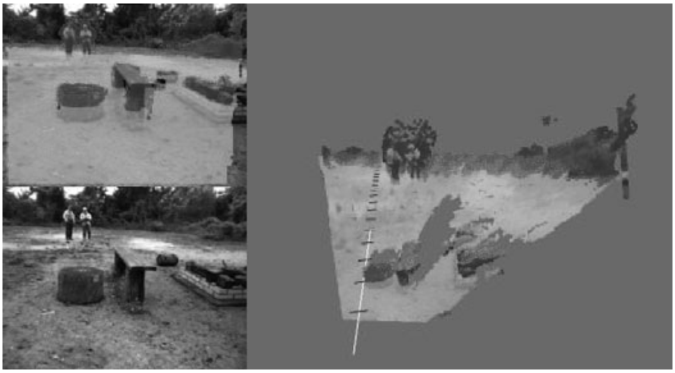

FIGURE 7.20: Robot navigation using the approach proposed in Hadsell *et al.* (2009) and Sermanet *et al.* (2009). The detected ground plane (lighter shade) and obstacles (darker one) are overlaid on one of the input images as well as a top view of the stereo reconstruction. Image courtesy of Yann LeCun.

fixed intrinsic and extrinsic parameters. *Active vision* is concerned with the construction of vision systems capable of dynamically modifying these parameters, e.g., changing camera zoom and vergence angles, and taking advantage of these capabilities in perceptual and robotic tasks (Aloimonos, Weiss & Bandyopadhyay 1987; Bajcsy 1988; Ahuja & Abbott 1993; Brunnström, Ekhlund, & Uhlin 1996).

Finally, let us mention the very useful resource assembled by D. Scharstein and R. Szeliski at `http://vision.middlebury.edu/stereo/`. One can find there benchmark data, an evaluation of various algorithms on this data, and code for many classical approaches to stereo fusion. See the web-site and Scharstein and Szeliski (2002) for details.

PROBLEMS

7.1. Show that one of the singular values of an essential matrix is 0 and the other two are equal. (Huang and Faugeras [1989] have shown that the converse is also true; that is, any 3×3 matrix with one singular value equal to 0 and the other two equal to each other is an essential matrix.)
Hint: The singular values of $\mathcal{E}$ are the eigenvalues of $\mathcal{E}\mathcal{E}^T$ (Chapter 22).

7.2. *Infinitesimal epipolar geometry.* Here we consider the case of *infinitesimal* camera displacements, and derive the instantaneous form of the Longuet–Higgins relation, Equation (7.2), which captures the epipolar geometry in the discrete case.

(a) We consider a moving camera with translational velocity $\boldsymbol{v}$ and rotational velocity $\boldsymbol{\omega}$. The matrix associated with the rotation whose axis is the unit vector $\boldsymbol{a}$ and whose angle is θ can be shown to be equal to

$$\mathcal{R} = e^{\theta[\boldsymbol{a}_\times]} \stackrel{\text{def}}{=} \sum_{i=0}^{+\infty} \frac{1}{i!}(\theta[\boldsymbol{a}_\times])^i.$$

Consider two frames separated by a small time interval δt, and denote by $\dot{\boldsymbol{p}} = (\dot{u}, \dot{v}, 0)^T$ the velocity of the point p, or *motion field.* Use this *expo-*

nential representation of rotation matrices to show that (to first order):

$$\left\{\begin{array}{l} \boldsymbol{t} = \delta t\, \boldsymbol{v}, \\ \mathcal{R} = \mathrm{Id} + \delta t\, [\boldsymbol{\omega}_\times], \\ \boldsymbol{p}' = \boldsymbol{p} + \delta t\, \dot{\boldsymbol{p}}. \end{array}\right. \tag{7.5}$$

(b) Use this result to show that Equation (7.2) reduces to

$$\boldsymbol{p}^T([\boldsymbol{v}_\times][\boldsymbol{\omega}_\times])\boldsymbol{p} - (\boldsymbol{p} \times \dot{\boldsymbol{p}}) \cdot \boldsymbol{v} = 0. \tag{7.6}$$

for infinitesimal motions.

7.3. *The focus of expansion.* Consider an infinitesimal translational motion ($\boldsymbol{\omega} = \mathbf{0}$). We define the *focus of expansion* (or *infinitesimal epipole*) as the point where the line passing through the optical center and parallel to the velocity vector $\boldsymbol{v}$ pierces the image plane. Use Equation (7.6) to show that the motion field points toward the focus expansion in this pure translational case.

7.4. Show that, in the case of a rectified pair of images, the depth of a point P in the normalized coordinate system attached to the first camera is $Z = -B/d$, where B is the baseline and d is the disparity.

7.5. Use the definition of disparity to characterize the accuracy of stereo reconstruction as a function of baseline and depth.

7.6. Give reconstruction formulas for verging eyes in the plane.

7.7. Give an algorithm for generating an ambiguous random dot stereogram that can depict two different planes hovering over a third one.

7.8. Show that the correlation function reaches its maximum value of 1 when the image brightnesses of the two windows are related by the affine transform $I' = \lambda I + \mu$ for some constants λ and μ with $\lambda > 0$.

7.9. Prove the equivalence of correlation and sum of squared differences for images with zero mean and unit Frobenius norm.

7.10. Recursive computation of the correlation function.

(a) Show that $(\boldsymbol{w} - \bar{\boldsymbol{w}}) \cdot (\boldsymbol{w}' - \bar{\boldsymbol{w}}') = \boldsymbol{w} \cdot \boldsymbol{w}' - (2m+1)(2n+1)\bar{I}\bar{I}'$.

(b) Show that the average intensity $\bar{I}$ can be computed recursively, and estimate the cost of the incremental computation.

(c) Generalize the prior calculations to all elements involved in the construction of the correlation function, and estimate the overall cost of correlation over a pair of images.

7.11. Show how a first-order expansion of the disparity function for rectified images can be used to warp the window of the right image corresponding to a rectangular region of the left one. Show how to compute correlation in this case using interpolation to estimate right-image values at the locations corresponding to the centers of the left window's pixels.

7.12. *Trifocal and quadrifocal matching constraints.* We show in this exercise the existence of trilinear and quadrilinear constraints that must be satisfied by matching points in three or four images, and generalize the epipolar constraint to that case.

(a) Suppose that we have four views of a point, with known intrinsic parameters and projection matrices $\mathcal{M}_i$ ($i = 1, 2, 3, 4$). Write an 8×4 homogeneous system of linear equations in the coordinate vector $\boldsymbol{P}$ in $\mathbb{R}^4$ of this point that must be satisfied by its projections into the four images.

Hint: Rewrite each projection equation as two linear equations in $\boldsymbol{P}$, parameterized by the corresponding projection matrix and image coordinates.

- **(b)** Use the fact that this homogeneous system of linear equations has $\boldsymbol{P}$ as a nontrivial solution to characterize matching constraints using two, three or four images.

  Hint: Use determinants.
- **(c)** Show that the conditions involving two images (say, the first and second one) reduces to the epipolar constraints of Equation (7.2) when we take $\mathcal{M}_1 = (\text{Id} \quad \mathbf{0})$ and $\mathcal{M}_2 = (\mathcal{R}^T \quad -\mathcal{R}^T\boldsymbol{t})$.
- **(d)** Show that the conditions involving three images are trilinear in the image coordinates and derive an explicit form for these conditions when $\mathcal{M}_1 = (\text{Id} \quad \mathbf{0})$, $\mathcal{M}_2 = (\mathcal{R}_2^T \quad -\mathcal{R}_2^T\boldsymbol{t}_2)$, and $\mathcal{M}_3 = (\mathcal{R}_3^T \quad -\mathcal{R}_3^T\boldsymbol{t}_3)$.
- **(e)** Show that the conditions involving four images are quadrilinear in the image coordinates.
- **(f)** Can you imagine a method for deriving matching constraints involving more than four images?

7.13. Generalize the constructions of the previous problem to the uncalibrated case.

PROGRAMMING EXERCISES

7.14. Implement the rectification process.

7.15. Implement a correlation-based approach to stereopsis.

7.16. Implement a multi-scale approach to stereopsis.

7.17. Implement a dynamic-programming approach to stereopsis.

7.18. Implement a trinocular approach to stereopsis.