Computer Vision
A Modern Approach

David A. Forsyth
University of California at Berkeley

Jean Ponce
University of Illinois at Urbana-Champaign

An Alan R. Apt Book

Prentice Hall
Upper Saddle River, New Jersey 07458

11

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 11.1), cartography, aerial reconnaissance, and close-range photogrammetry.

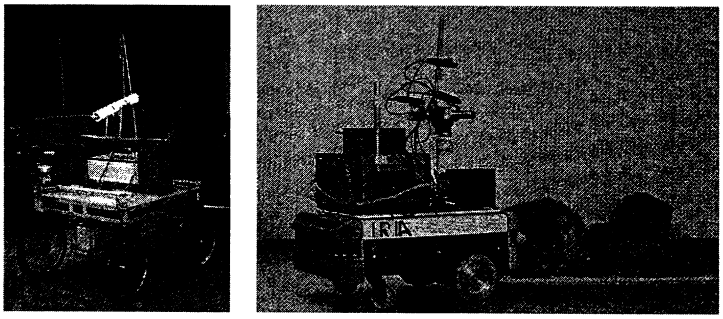

Figure 11.1 Left: The Stanford cart sports a single camera moving in discrete increments along a straight line and providing multiple snapshots of outdoor scenes. Right: The INRIA mobile robot uses three cameras to map its environment. As shown by these examples, although two eyes are sufficient for 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 11.4. *Photos courtesy of Hans Moravec and Olivier Faugeras.*

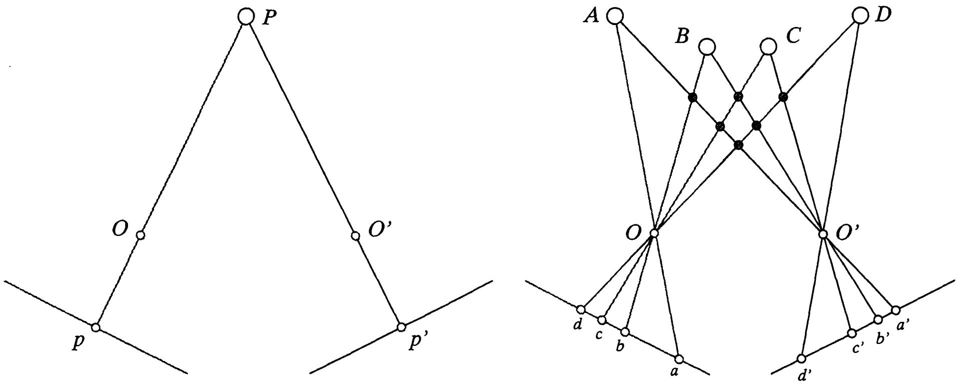

Figure 11.2 The binocular fusion problem: In the simple case of the diagram shown on the left, there is no ambiguity, and stereo reconstruction is a simple matter. In the more usual case shown on the right, any of the four points in the left picture may, a priori, match any of the four points in the right one. Only four of these correspondences are correct; the other ones yield the incorrect reconstructions shown as small gray discs.

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.

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 11.2, left). Thus, when a single image feature is observed at any given time, stereo vision is easy. However, each picture consists of hundreds of thousands 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 11.2, right). The epipolar constraint plays a fundamental role in this process since it restricts the search for image correspondences to matching epipolar lines.

We assume in the rest of this chapter that all cameras have been carefully calibrated so their intrinsic and extrinsic parameters are precisely known relative to some fixed world coordinate system (this implies of course that the essential matrices and/or trifocal tensors associated with pairs or triples of cameras are known as well). The case of uncalibrated cameras is examined in the context of structure from motion in chapters 12 and 13.

11.1 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 $R' = O'p'$. However, the rays R and R' will never, in practice, actually intersect due to calibration and feature localization errors (Figure 11.3). In this context, various reasonable approaches to the reconstruction problem can be adopted. For example, we can construct the line segment perpendicular to R and R' that intersects both rays: Its mid-point P is the closest point to the two rays and can be taken as the preimage of p and p'. It should be noted that a similar construction was used at the end of chapter 10 to characterize algebraically the geometry of multiple views in the presence of calibration or measurement errors. Equations (10.22) and (10.23) derived in that

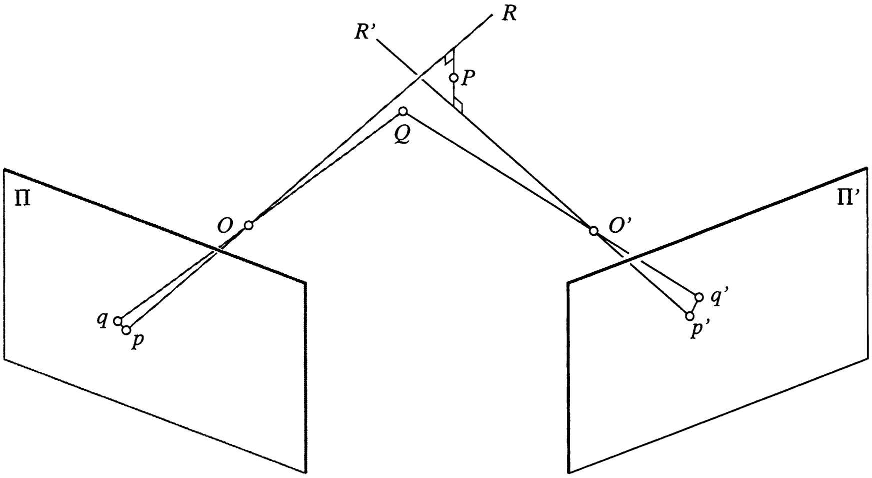

Figure 11.3 Triangulation in the presence of measurement errors. See text for details.

chapter are readily adapted to the calculation of the coordinates of P in the frame attached to the first camera.

Alternatively, we 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 coordinates of P that is easily solved using the linear least-squares techniques introduced in chapter 3. Unlike the previous approach, this reconstruction method does not have an obvious geometric interpretation, but generalizes readily to the case of three or more cameras, each new picture simply adding two additional constraints.

Finally, we 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 11.3). 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 3. 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.

11.1.1 Image Rectification

The calculations associated with stereo algorithms are often considerably simplified when the images of interest have been *rectified* (i.e., replaced by two equivalent pictures with a common image plane parallel to the baseline joining the two optical centers; see Figure 11.4). 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 since modifying it only changes the scale of the recti-

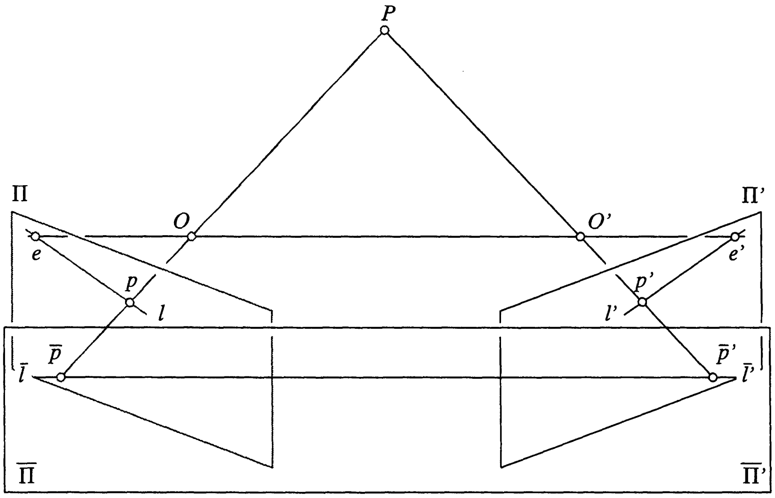

Figure 11.4 A rectified stereo pair: The two image planes Π and Π' are reprojected onto a common plane $\bar{\Pi} = \bar{\Pi}'$ parallel to the baseline. The epipolar lines l and l' associated with the points p and p' in the two pictures map onto a common scanline $\bar{l} = \bar{l}'$ also parallel to 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 into the plane $\bar{\Pi} = \bar{\Pi}'$.

fied 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 notion of disparity introduced earlier takes a precise meaning: Given two points p and p' located on the same scanline of the left and right images, with coordinates (u, v) and (u', v), the disparity is defined as the difference $d = u' - u$. Let us assume from now on normalized image coordinates. As shown in the exercises, if B denotes the distance between the optical centers, also called 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 $\boldsymbol{p} = (u, v, 1)^T$ is the vector of normalized image coordinates of p. This provides yet another reconstruction method for rectified stereo pairs.

11.2 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 11.5 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 $d = r - l$ (Figure 11.5).

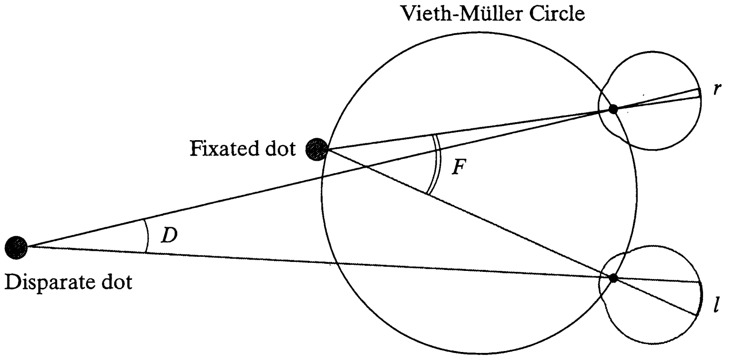

Figure 11.5 In this diagram, the close-by dot is fixated by the eyes, and it projects onto the center of their foveas with no disparity. The two images of the far dot deviate from this central position by different amounts, indicating a different depth.

It is an elementary exercise in trigonometry to show that $d = D - F$, where D denotes the angle between these 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 anterior nodal points of the eyes. Points lying inside this circle have a positive disparity, points lying outside it have, as in Figure 11.5, 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' nodal points. This property is clearly sufficient to rank order in depth dots that are near the fixation point. 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, 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. As demonstrated by Wundt and Helmholtz (1909) nearly 100 years ago, there is strong evidence that these angles cannot be measured accurately by our nervous system. 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) asks 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? Some anecdotal evidence hints at a binocular mechanism. To quote Julesz: "In aerial reconnaissance it is known that objects camouflaged by a complex background are very difficult to detect but jump out if viewed sterescopically." To gather more conclusive data and settle the matter, Julesz introduces a new device, the *random dot stereogram*, a pair of synthetic images obtained by randomly spraying black dots on white objects, typically a small square plate floating over a larger one (Figure 11.6). To quote him again: "When viewed monocularly, the images appear completely random. But when viewed stereoscopically, the image pair gives the impression of a square markedly in front of (or behind) the surround." The conclusion is clear:

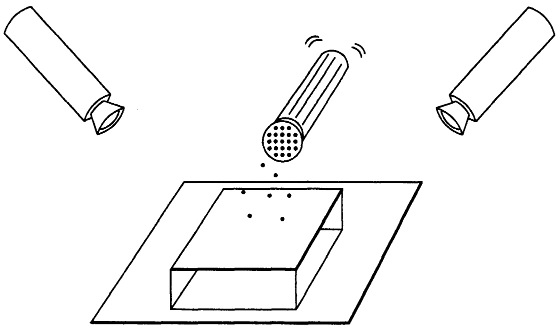

Figure 11.6 Random dot stereograms: Shaking (virtual) pepper over two plates.

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.

The *dipole* model of stereopsis proposed by Julesz is *cooperative*, with neighboring matches influencing each other to avoid ambiguities and promote a global analysis of the scene. The approach proposed by Marr and Poggio (1976) is another instance of a cooperative process that performs quite well on random dot stereograms. Their algorithm relies on three constraints: (a) *compatibility* (black dots can only match black dots, or, more generally, two image features can only match if they have possibly arisen from the same physical marking), (b) *uniqueness* (a black dot in one image matches at most one black dot in the other picture), and (c) *continuity* (the disparity of matches varies smoothly almost everywhere in the image). Given a number of black dots on a pair of corresponding epipolar lines, Marr and Poggio build a graph that reflects possible correspondences (Figure 11.7).

The nodes of the graph are pairs of black dots within some disparity range, reflecting the compatibility constraint; vertical and horizontal arcs represent inhibitory connections associated

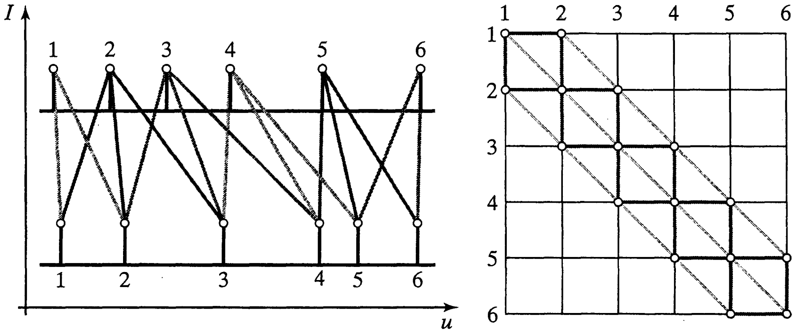

Figure 11.7 A cooperative approach to stereopsis: The Marr–Poggio (1976) algorithm. The left part of the figure shows two intensity profiles along the same scanline of two images. The spikes correspond to black dots. The line segments joining the two profiles indicate possible matches between dots given some maximum disparity range. These matches are also shown in the right part of the figure, where they form the nodes of a graph. The vertical and horizontal arcs of this graph join nodes associated with the same dot in the left or right image. The diagonal arcs join nodes with similar disparities.

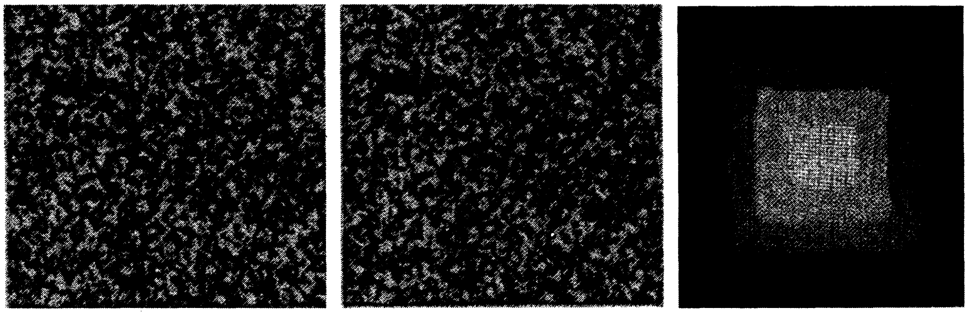

Figure 11.8 From left to right: A random dot stereogram depicting four planes at varying depth (a "wedding cake") and the disparity map obtained after 14 iterations of the Marr–Poggio cooperative algorithm. *Reprinted from VISION: A COMPUTATIONAL INVESTIGATION INTO THE HUMAN REPRESENTATION AND PROCESSING OF VISUAL INFORMATION by David Marr, © 1982 by David Marr. Reprinted by permission of Henry Holt and Company, LLC.*

with the uniqueness constraint (any match between two dots should discourage any other match for both the left dot—horizontal inhibition—and the right one—vertical inhibition—in the pair); and diagonal arcs represent excitory connections associated with the continuity constraint (any match should favor nearby matches with similar disparities).

In this approach, a quality measure is associated with each node. It is initialized to 1 for every pair of potential matches within some disparity range. The matching process is iterative and parallel, each node being assigned at each iteration a weighted combination of its neighbors' values. Excitory connections are assigned weights equal to 1, and inhibitory ones weights equal to $-w$ (where w is a suitable weighting factor). A node is assigned a value of 1 when the corresponding weighted sum exceeds some threshold and a value of 0 otherwise. This approach works quite reliably on random dot stereograms (Figure 11.8), but not on natural images. As we return to computer vision in the next section, we present a number of techniques that perform better on most real pictures, but the original Marr–Poggio algorithm and its implementation retain the interest of offering an early example of a theory of stereopsis that allows the fusion of random dot stereograms.

11.3 BINOCULAR FUSION

11.3.1 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 *et al.*, 1977, Gennery, 1980). Concretely, let us consider a rectified stereo pair and a point (u, v) in the first image. We associate with the window of size $p = (2m + 1) \times (2n + 1)$ centered in (u, v) the vector $\mathbf{w}(u, v) \in \mathbb{R}^p$ 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 $(u + d, v)$ in the second image, we can construct a second vector $\mathbf{w}'(u + d, v)$ and define the corresponding *normalized correlation function* as

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

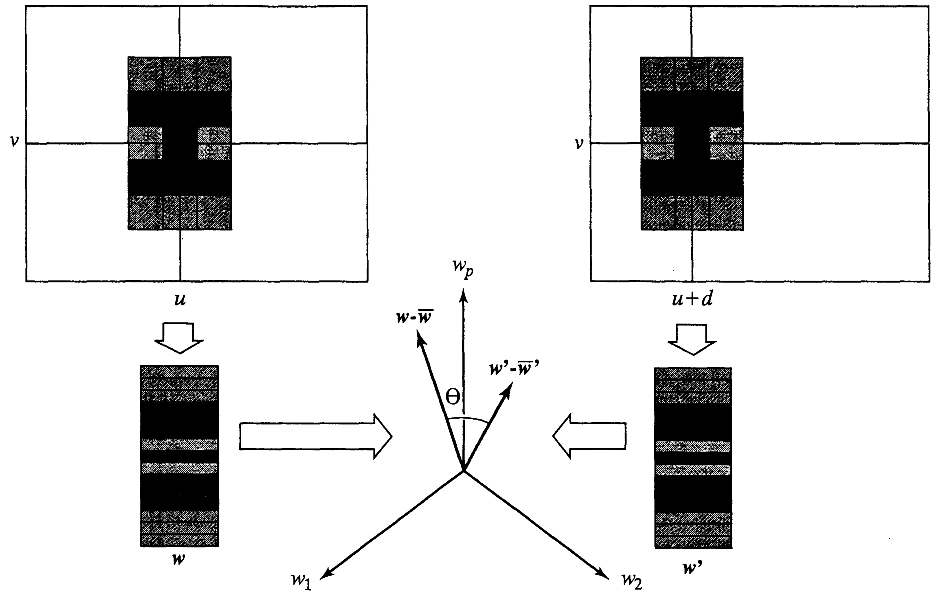

Figure 11.9 Correlation of two 3 × 5 windows along corresponding epipolar lines. The second window position is separated from the first one by an offset d. The two windows are encoded by vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$ in $\mathbb{R}^{15}$, and the correlation function measures the cosine 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.

where the u, v 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 $\boldsymbol{a}$ (Figure 11.9).

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 Exercises). 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 Exercises) that maximizing the normalized correlation function is equivalent to minimizing

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

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 Exercises). Finally, a major problem with correlation-based techniques for estab-

[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.

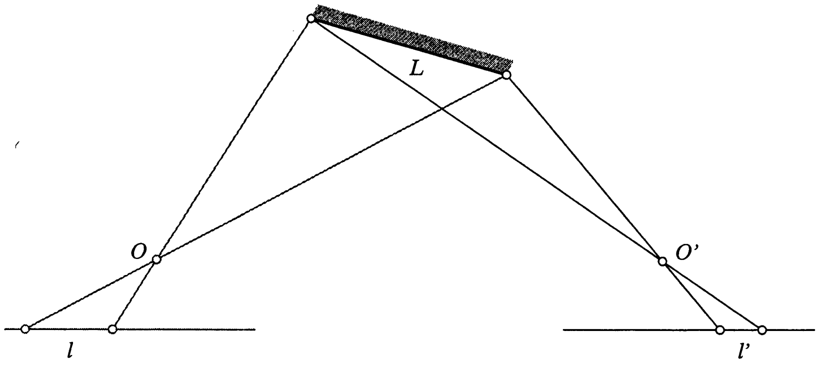

Figure 11.10 The foreshortening of (oblique) surfaces depends on the position of the cameras observing them: $l/L \neq l'/L$.

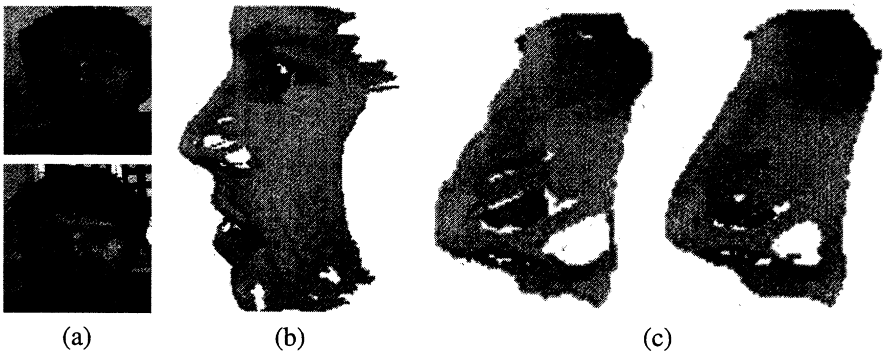

Figure 11.11 Correlation-based stereo matching: (a) a pair of stereo pictures; (b) a texture-mapped view of the reconstructed surface; (c) comparison of the regular (left) and refined (right) correlation methods in the nose region. The latter clearly gives better results. *Reprinted from "Computing Differential 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.*

lishing stereo correspondences is that they implicitly assume that the observed surface is (locally) parallel to the two image planes (Figure 11.10).

This suggests a two-pass algorithm where initial estimates of the disparity are used to warp the correlation windows to compensate for inequal amounts of foreshortening in the two pictures. Figure 11.11 shows an example, where a warped window is defined in the right image for each rectangle in the left image using the disparity in the center of the rectangle and its derivatives (Devernay & Faugeras, 1994). 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.

11.3.2 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

Algorithm 11.1: The Marr–Poggio (1979) Multiscale Binocular Fusion Algorithm.

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.

features such as edges preferred to matches between raw pixel intensities. These principles are implemented in Algorithm 11.11.1, which is due to Marr and Poggio (1979).

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 may 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 the algorithm and is illustrated by Figure 11.12. 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 11.12 (bottom).

11.3.3 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 11.13, left). This is the so-called *ordering constraint* introduced in the early 1980s (Baker and Binford, 1981; Ohta and Kanade, 1985). Interestingly enough, it may not be satisfied by real scenes, in particular when small solids occlude parts of larger ones (Figure 11.13, 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 *et al.*, 1974) to establish stereo correspondences (Figure 11.14). 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 11.14, 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 point corresponding to missing correspondences associated with occlusion and/or noise.

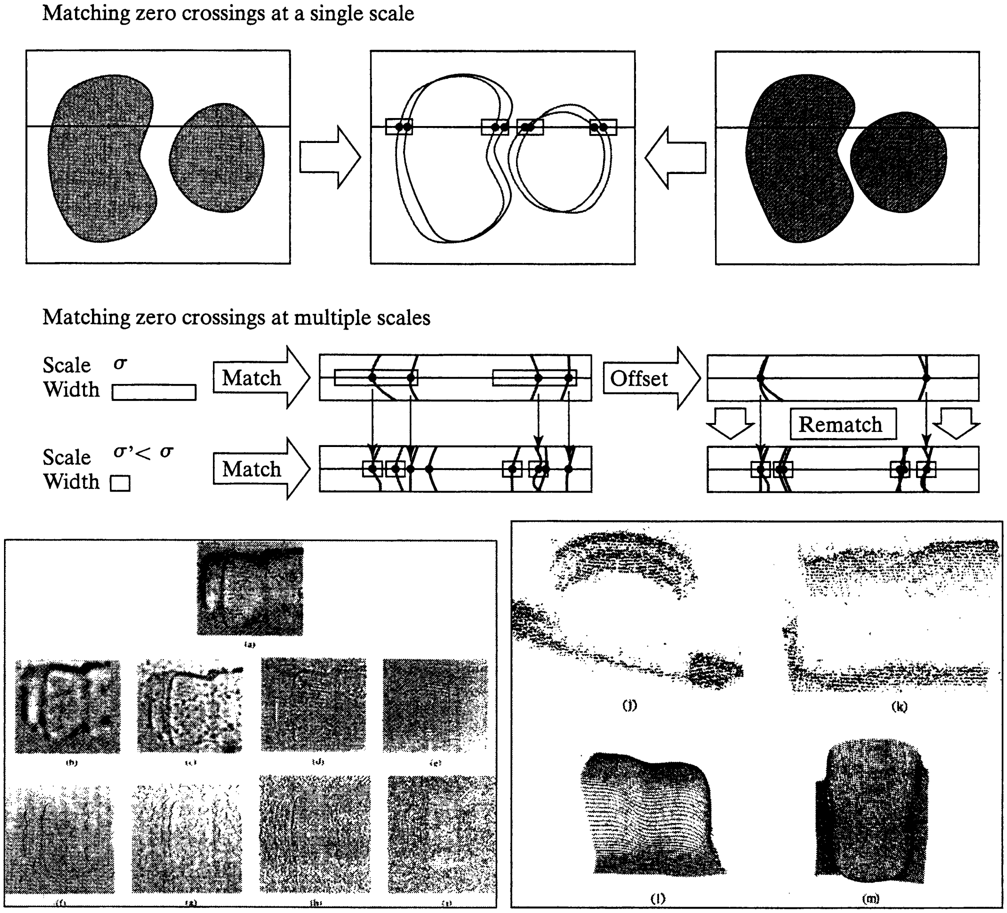

Figure 11.12 Top: Singlescale matching. Middle: Multiscale matching. Bottom: Results. Bottom left: The input data (including one of the input pictures, the output of four $\nabla^2 G_\sigma$ filters, and the corresponding zero crossings). Bottom right: Two views of the depth map constructed by the matching 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.*

This setting allows us to restate 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; arcs represent matches between left and right intensity profile intervals bounded by the features of the corresponding nodes (Figure 11.14, 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 using dynamic programming as shown in Algorithm 11.11.2.

As given, Algorithm 11.11.2 has a computational complexity of $O(mn)$, where m and n, respectively, denote the number of edge points on the matched left and right scanlines.[2] Variants

[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).

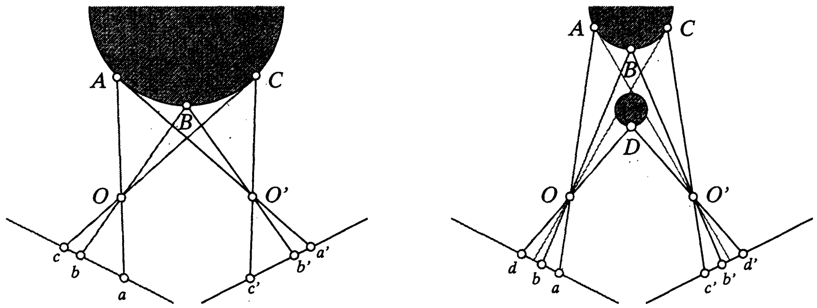

Figure 11.13 Ordering constraints. In the (usual) case shown in the left part of the diagram, the order of feature points along the two (oriented) epipolar lines is the same. In the case shown in the right part of the figure, a small object lies in front of a larger one. Some of the surface points are 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.

Algorithm 11.2: A dynamic-programming algorithm for establishing stereo correspondences between two corresponding scanlines with 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 $i \leq k$ and $j \leq l$, 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.
```

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 interscanline 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 11.15 shows a sample result taken from (Ohta and Kanade, 1985).

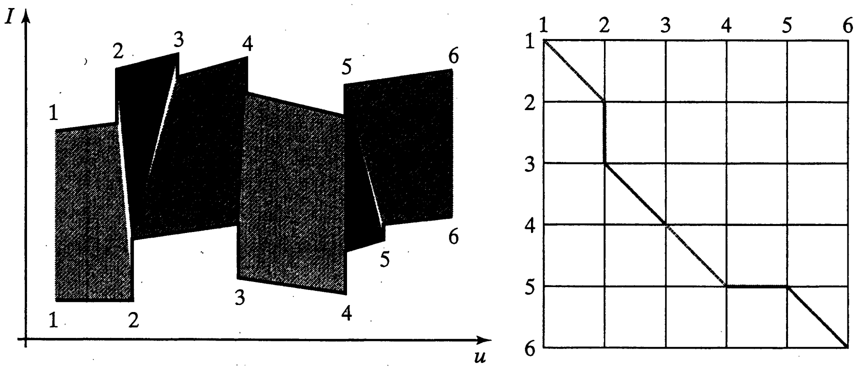

Figure 11.14 Dynamic programming and stereopsis: The left part of the figure shows two intensity profiles along matching epipolar lines. The polygons joining the two profiles indicate matches between successive intervals (some of the matched intervals may have zero length). The right part of the diagram 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.

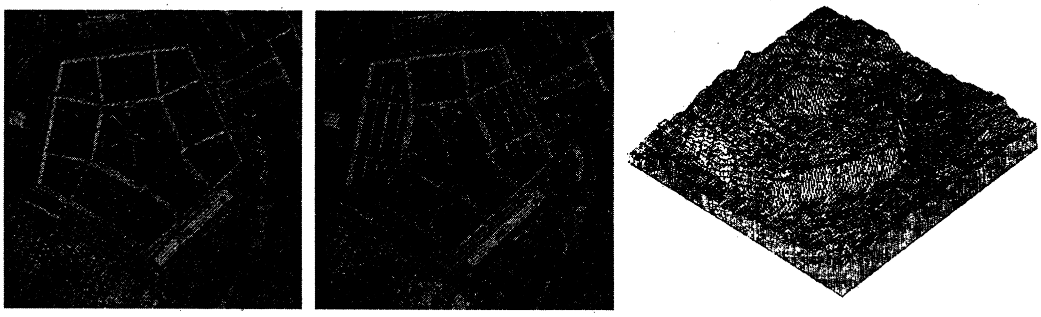

Figure 11.15 Two images of the Pentagon and an isometric plot of the disparity map computed by the dynamic-programming algorithm of Ohta and Kanade (1985). *Reprinted from "Stereo by Intra- and Inter-Scanline Search," by Y. Ohta and T. Kanade, IEEE Transactions on Pattern Analysis and Machine Intelligence, 7(2):139–154, (1985). © 1985 IEEE.*

11.4 USING MORE CAMERAS

11.4.1 Three 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 11.16): The three-dimensional point associated with such a match is first reconstructed then reprojected into the third image. If no compatible point lies nearby, then the match must be wrong. In fact, the reconstruction/reprojection process can be avoided by noting, as in chapter 10, that given three weakly (and a fortiori strongly) calibrated cameras and two images of a point one can always predict its position in a third image by intersecting the corresponding epipolar lines.

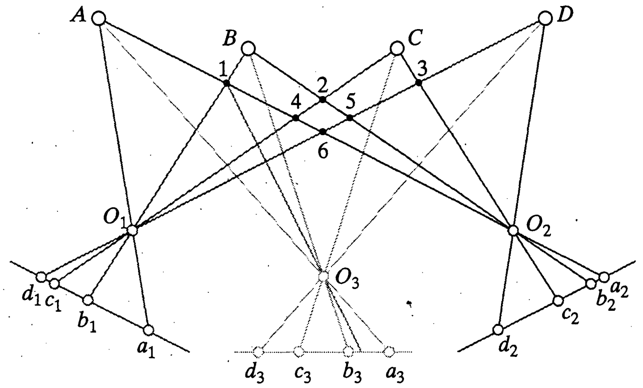

Figure 11.16 The small gray discs indicate the incorrect reconstructions associated with the left and right images of four points. The addition of a central camera removes the matching ambiguity: None of the corresponding rays intersects any of the six discs. Alternatively, matches between points in 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 since there is no feature point in the third image near the reprojection of the hypothetical reconstruction numbered 1 in the diagram.

11.4.2 Multiple Cameras

In most trinocular stereo algorithms, potential correspondences are hypothesized using two of the images, then confirmed or rejected using the third one. In contrast, Okutami and Kanade (1993) have proposed a multicamera method where matches are found using all pictures at the same time. 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 11.17 plots the value of E as a function of inverse depth for various sets of cameras. It should be noted that the corresponding images contain a repetitive pattern and that 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 11.18 shows a sequence of 10 rectified images and a plot of the surface reconstructed by the algorithm.

11.5 NOTES

The fact that disparity gives rise to stereopsis in human beings was first 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 (Helmholtz, 1909, p. 455). 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)

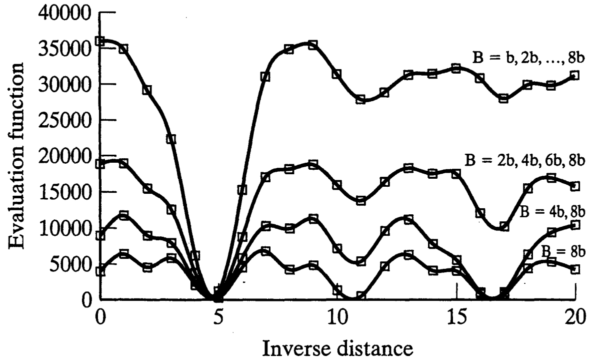

Figure 11.17 Combining multiple views: The sum of squared differences is plotted here as a function of the inverse depth for various numbers of input pictures. The data are taken from a scanline near the top of the images shown in Figure 11.18, whose intensity is nearly periodic. The diagram clearly 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.*

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 *et al.* (1970), McKee *et al.* (1990), and Anderson and Nakayama (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 in (Binford, 1984). Early techniques for line matching in binocular stereo include Medioni and Nevatia (1984) and

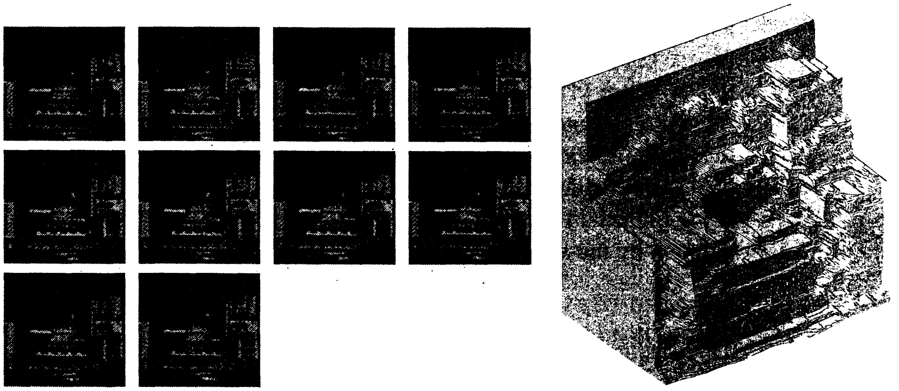

Figure 11.18 A series of 10 images and the corresponding reconstruction. The gridboard near the top of the images is the source for the nearly periodic brightness signal giving rise to ambiguities in Figure 11.17. *Reprinted from "A Multiple-Baseline Stereo System," by M. Okutami and T. Kanade, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(4):353–363, (1993). © 1993 IEEE.*

Ayache and Faugeras (1987). Algorithms for trinocular fusion include Milenkovic and Kanade (1985), Yachida *et al.* (1986), Ayache and Lustman (1987), and Robert and Faugeras (1991). As shown in Robert and Faugeras (1991) and the exercises, the trifocal tensor introduced in chapter 10 can be used to also predict the tangent and curvature along some image curve in one image given the corresponding quantities measured in in the other images. This fact can be used to effectively match and reconstruct curves from three images.

As noted earlier, image edges are often used as the basis for establishing binocular correspondences, at least in part because they can (in principle) be identified with physical properties of the imaging process, corresponding for example to albedo, color, or occlusion boundaries. A point rarely taken into account by stereo-matching algorithms is that binocular fusion *always* fails along the contours of solids bounded by smooth surfaces (Figure 11.19). Indeed, in this case, the corresponding image edges are viewpoint dependent, and matching them yields erroneous reconstructions. As shown in Arbogast and Mohr (1991), Vaillant and Faugeras (1992), Cipolla and Blake (1992), and Boyer and Berger (1996), three cameras are sufficient in this case to reconstruct a local second-degree surface model.

It is not quite clear at this point whether feature-based matching is preferable to gray-level matching. The former is accurate near surface markings, but only yields a sparse set of measurements, whereas the latter may give poor results in uniform regions but provides dense correspondences in textured areas. In this context, the topic of dense surface interpolation from sparse samples is important, although it has hardly been mentioned in this chapter. The interested reader is referred to Grimson (1981*b*) and Terzopoulos (1984) for more details.

A different approach to stereo vision that we have also failed to discuss for lack of space involves higher level interpretation processes—for example, prediction/verification methods operating on graphical image descriptions (Ayache and Faverjon, 1997) or hierarchical techniques matching curves, surfaces, and volumes found in two images (Lim and Binford, 1988).

All of the algorithms presented in this chapter (implicitly) assume that the images being fused are quite similar. This is equivalent to considering a short baseline. An effective algorithm for dealing with wide baselines can be found in Pritchett and Zisserman (1998). Another, model-based approach is discussed in chapter 26.

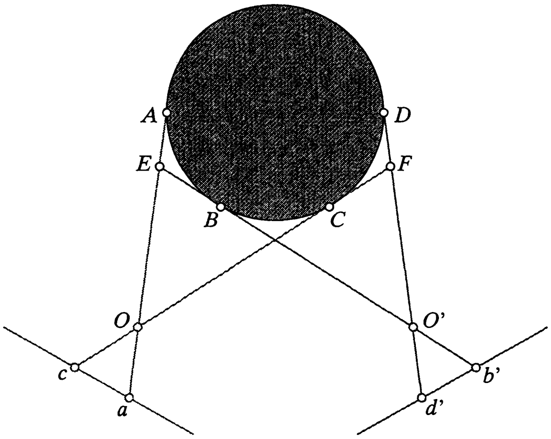

Figure 11.19 Stereo matching fails at smooth object boundaries: For narrow baselines, the pairs (c, d') and (a, b') are easily matched by most edge-based algorithms, yielding the fictitious points F and E as the corresponding three-dimensional reconstructions.

Finally, we have limited our attention to stereo rigs with 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; see Aloimonos *et al.*, 1987, Bajcsy, 1988, Ahuja and Abott 1993, Brunnström *et al.*, 1996).

PROBLEMS

11.1. 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.

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

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

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

11.5. 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 $\lambda > 0$.

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

11.7. Recursive computation of the correlation function.

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

(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.

11.8. 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.

11.9. Show how to use the trifocal tensor to predict the tangent line along an image curve from tangent line measurements in two other pictures.

Programming Assignments

11.10. Implement the rectification process.

11.11. Implement a correlation-based approach to stereopsis.

11.12. Implement a multiscale approach to stereopsis.

11.13. Implement a dynamic-programming approach to stereopsis.

11.14. Implement a trinocular approach to stereopsis.