Chapter 13

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 (Figure 13.1(left)). This chapter is concerned with the design and implementation of algorithms that mimick our ability to perform this task, known as *stereopsis*. Note that a machine (or for that matter the Martian shown in Figure 13.1(right), or an ordinary spider) may be equipped with three eyes or more, and this will lead us to investigate multi-camera approaches to stereopsis at the end of this chapter.

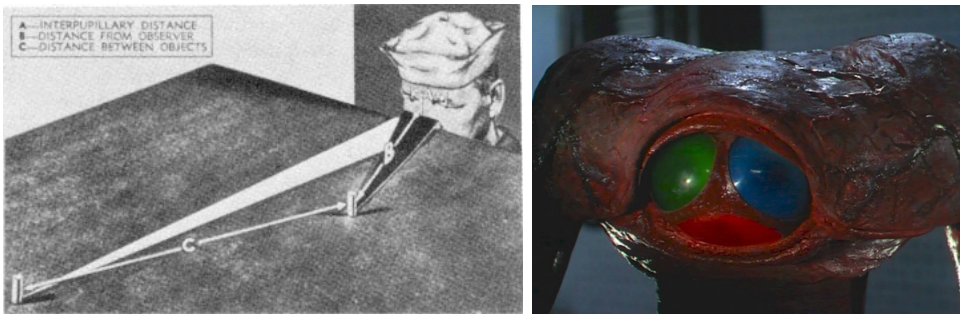


Figure 13.1. The sailor shown in the left picture is, like most people, able to perform stereopsis and gain a sense of depth for the objects within his field of view. Reprinted from [Navy, 1969], Figure 6-8. The right photograph is from the 1953 film "The War of the Worlds", and it shows a close-up of the face of a three-eyed Martian warrior. Why such a configuration may prove benefitial will be explained in Section 13.3.1.

Reliable computer programs for stereoscopic perception are of course invaluable in visual robot navigation (Figure 13.2), cartography, aerial reconnaissance and close-range photogrammetry. They are also of great interest in tasks such as image segmentation for object recognition and, as will be seen in Chapter 26, the construction of three-dimensional scene models in image-based rendering, a new discipline that ties together computer vision and computer graphics.

Stereo vision involves two processes: the *binocular fusion* of features observed by the two eyes, and the *reconstruction* of their three-dimensional preimage. The

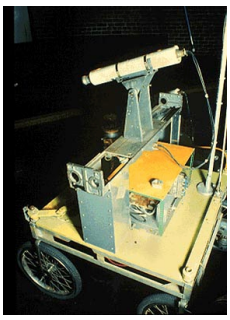
(a)  (b) 

Figure 13.2. Mobile robot navigation is a classical application of stereo vision: (a) the Stanford cart sports a single camera moving in discrete increments along a straight line and providing multiple snapshots of outdoor scenes [Moravec, 1983]; the INRIA mobile robot uses three cameras to map its environment.

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 13.3(left)). Thus, when a single image feature is observed at any given time, stereo vision is easy.[1] 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 13.3(right)).

Although human binocular fusion is effortless and reliabe in most situations, we can be fooled too: the abstract *single-image stereograms* [Thimbley *et al.*, 1994] that were popular in the late nineties demonstrate this quite well: in this case, repetitive patterns or judiciously assembled random dots are used to trick the eyes into focussing on the wrong correspondences, producing a vivid impression of layered planes.[2] This suggests that constructing a reliable stereo vision program is difficult, a fact that will be attested time and again in the rest of this chapter. As should be expected, the geometric machinery introduced in Chapter 12 will prove extremely useful in tackling this problem. We will 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. The case of multiple uncalibrated cameras will be examined in the context of structure from motion in Chapters 14 and 15.

[1] This is actually how some laser range finders work: two cameras observe an object while a laser beam scans its surface one point at a time. After thresholding the two pictures, the bright laser spot is, effectively, the only surface point seen by the cameras. See Chapter 24 for details.

[2] To enjoy this effect without any special equipment or expensive props, you may try to sit down in a place decorated with a repetitive tile pattern such as those often found in bathroom floors. By letting your mind wander and your eyes unfocus, you may be able to see the floor jump up by a foot or so, and even pass your hand through the "virtual" floor. This experiment, best conducted late at night, is quite worth the effort.

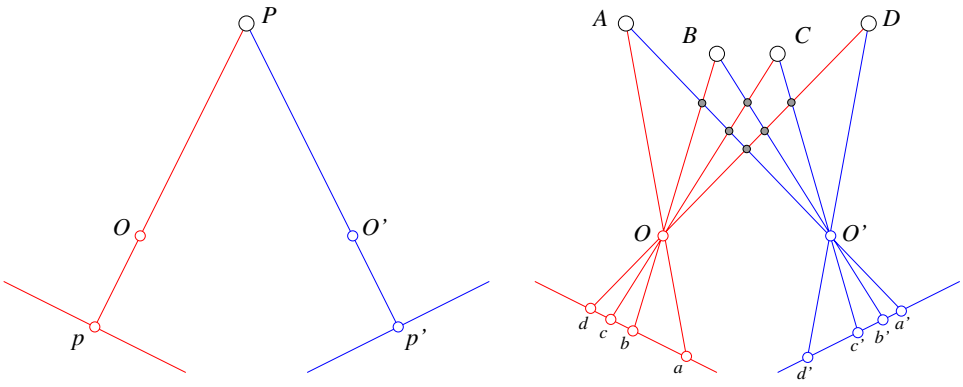


Figure 13.3. The binocular fusion problem: in the simple case of the diagram shown on the left, there is no ambiguity and stereo reconstruction is a simple matter. In the more usual case shown on the right, any of the four points in the left picture may, a priori, match any of the four points in the right one. Only four of these correspondences are correct, the other ones yielding the incorrect reconstructions shown as small grey discs.

13.1 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 13.4). In this context, various reasonable approaches to the reconstruction problem can be adopted. For example, we may choose to construct the line segment perpendicular to R and R' that intersects both rays: the mid-point P of this segment is the closest point to the two rays and can be taken as the pre-image of p and p'. It should be noted that a similar construction was used at the end of Chapter 12 to characterize algebraically the geometry of multiple views in the presence of calibration or measurement errors. The equations (12.4.1) and (12.4.2) derived in that 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

$$\left\{ \begin{array}{l} \boldsymbol{p} \times \mathcal{M}\boldsymbol{P} = 0 \\ \boldsymbol{p}' \times \mathcal{M}'\boldsymbol{P} = 0 \end{array} \right. \iff \left(\begin{array}{c} [\boldsymbol{p}_\times]\mathcal{M} \\ [\boldsymbol{p}'_\times]\mathcal{M}' \end{array} \right) \boldsymbol{P} = 0.$$

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 6. Unlike the previous approach, this reconstruction method does not have an obvious geometric interpretation, but it generalizes

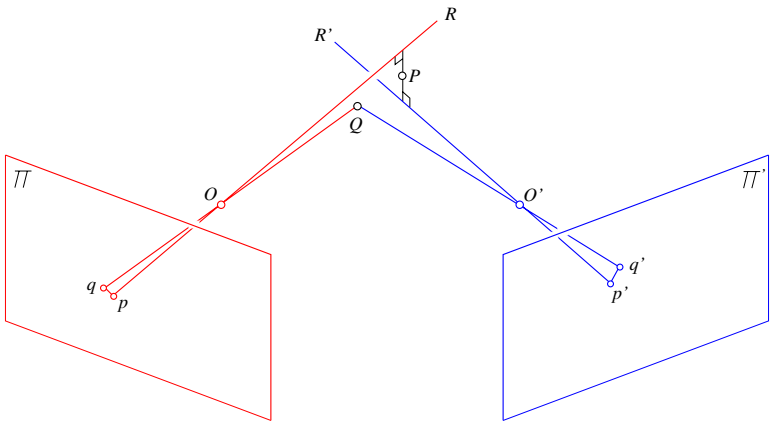


Figure 13.4. Triangulation in the presence of measurement errors. See text for details.

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 13.4). 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 non-linear least-squares techniques such as those introduced in Chapter 6. The reconstruction obtained by either of the other two methods can be used as a reasonable guess to initialize the optimization process. This non-linear approach also readily generalizes to the case of multiple images.

Before moving on to studying the problem of binocular fusion, let us now say a few words about two key components of stereo vision systems: camera calibration and image rectification.

13.1.1 Camera Calibration

As noted in the introduction, we will assume throughout this chapter that all cameras have been carefully calibrated (using, for example, one of the techniques introduced in Chapter 6) so their intrinsic and extrinsic parameters are precisely known relative to some fixed world coordinate system. This is of course a prerequisite for the reconstruction methods presented in the previous section since they require that the projection matrices associated with the two cameras be known, or, equivalently, that a definite ray be associated with every image point. It should also be noted that, once the intrinsic and extrinsic camera parameters are known, it is a simple matter to estimate the multi-view geometry (essential matrix for two views, trifocal tensor for three, etc.) as described in Chapter 12. This will play a fundamental role

in the algorithms for establishing stereo correspondences presented in Sections 13.2 and 13.3.

13.1.2 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 projectively equivalent pictures with a common image plane parallel to the baseline joining the two optical centers (Figure 13.5). The rectification process can be implemented by projecting the original pictures onto the new image plane. With an apropriate choice of coordinate system, the rectified epipolar lines are scanlines of the new images, and they are also parallel to the baseline.

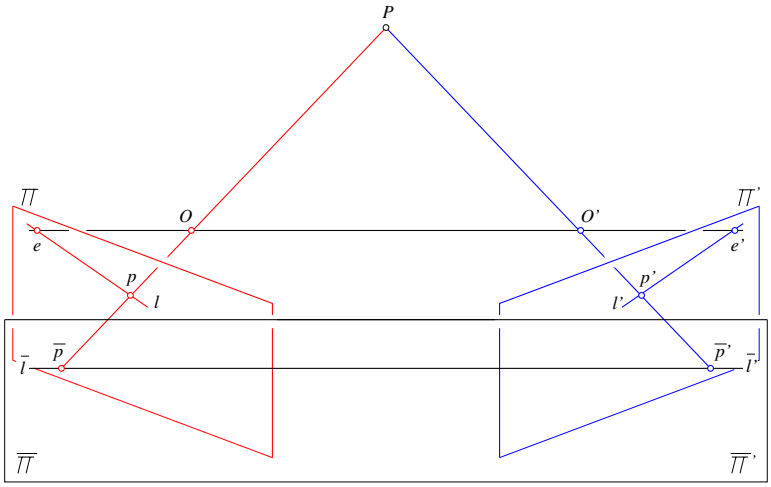


Figure 13.5. A rectified stereo pair: the two image planes Π and Π' are reprojected onto a common plane $\bar{\Pi} = \bar{\Pi}'$ parallel to the baseline. The epipolar lines l and l' associated with the points p and p' in the two pictures map onto a common scanline $\bar{l} = \bar{l}'$ also parallel to the baseline and passing through the reprojected points $\bar{p}$ and $\bar{p}'$. 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 $\bar{\Pi} = \bar{\Pi}'$.

As noted in [Faugeras, 1993], there are two degrees of freedom involved in the choice of the rectified image plane: (1) the distance between this plane and the baseline, which is essentially irrelevant since modifying it will only change the scale of the rectified pictures, an effect easily balanced by an inverse scaling of the image coordinate axes, and (2) 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 informally 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. If B denotes the distance between the optical centers, also called baseline in this context, it is easy to show that the depth of P in the (normalized) coordinate system attached to the first camera is $z = -B/d$ (Figure 13.6).

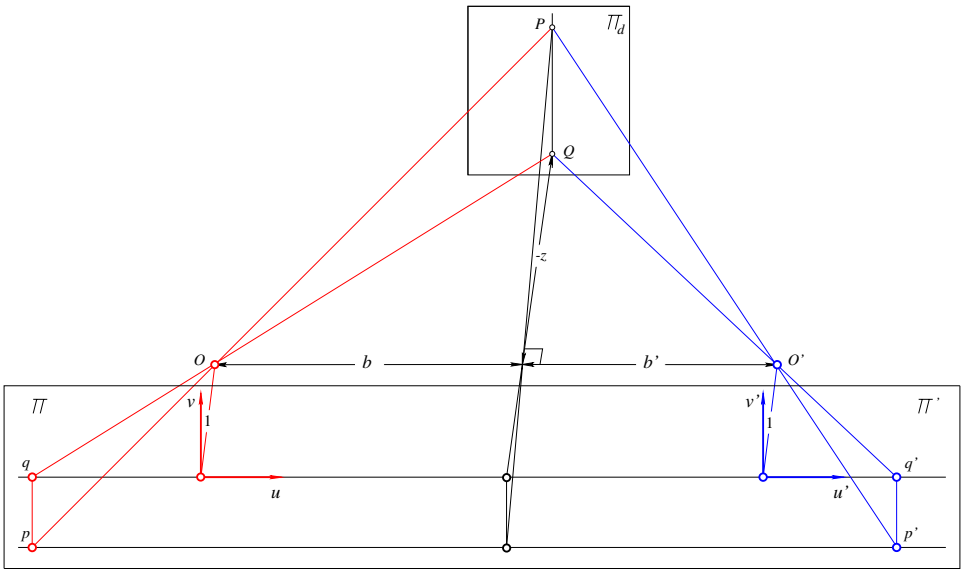


Figure 13.6. Triangulation for rectified images: the rays associated with two points p and p' on the same scanline are by construction guaranteed to intersect in some point P. As shown in the text, the depth of P relative to the coordinate system attached to the left camera is inversely proportional to the disparity $d = u' - u$. In particular, the preimage of all pairs of image points with constant disparity d is a *frontoparallel* plane Π_d (i.e., a plane parallel to the camera retinas).

To show this, let us consider first the points q and q' with coordinates $(u, 0)$ and $(u', 0)$, and the corresponding scene point Q. Let b and b' denote the respective distances between the orthogonal projection of Q onto the baseline and the two optical centers O and O'. The triangles qQq' and OQO' are similar, and it follows immediately that $b = zu$ and $b' = -zu'$. Thus $B = -zd$, which proves the result for q and q'. The general case involving p and p' with $v \neq 0$ follows immediately from the fact that the line PQ is parallel to the two lines pq and $p'q'$ and therefore also parallel to the rectified image plane. In particuliar, 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.

Human Vision: 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 its two images form in the centers of the eyes' foveas. Figure 13.7 illustrates a simplified, two-dimensional situation.

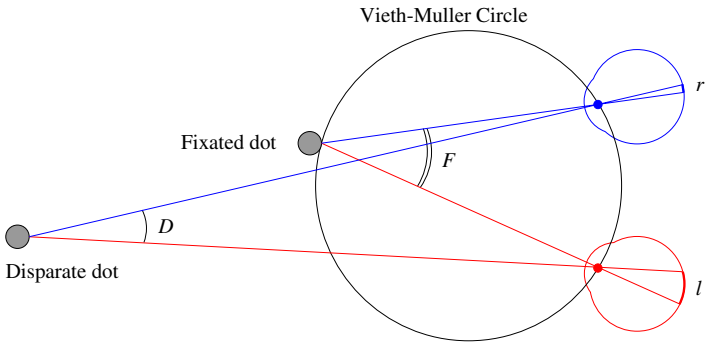


Figure 13.7. This diagram depicts a situation similar to that of the sailor in Figure 13.1. The close-by dot is fixated by the eyes, and it projects onto the center of their foveas, with no disparity. The two images of the far dot deviate from this central position by different amounts, indicating a different depth.

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 13.7). 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 (or *convergent*) disparity, points lying outside it have, as in Figure 13.7, a negative (or *divergent*) disparity,[3] and the locus of all points having a given disparity d forms, as d varies, the pencil 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 in order to reconstruct the absolute position of scene points.

The three-dimensional case is naturally a bit 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 already demonstrated by Wundt and Helmholtz [1909, pp. 313-314] a hundred years ago, there is strong evidence that these angles cannot be measured very accurately by our nervous system. In fact, the human

[3] The terminology comes from the fact that the eyes would have to converge (resp. diverge) to fixate on a point inside (resp. outside) the Vieth-Müller circle. Note that the position of this circle in space depends on the fixation point (even if the fixation angle F is preserved), since the rotation centers of the eyes do not coincide with their anterior nodal points.

visual system can be fooled into believing that threads that actually lie in the same vertical plane lie instead on a convex or concave surface, depending on the distance between the observer and this plane [Helmholtz, 1909, pp. 318-321]. Likewise, the *relief* models used in sculpture to mimick solids with much reduced depths are almost indistinguishable binocularly from the originals (see [Helmholtz, 1909, pp. 324-326] for an analytical justification). On the other hand, *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*) [Helmholtz, 1909, p. 307] (though the stereo disparity threshold increases quickly as one gets away from the horopter, see, for example, [McKee *et al.*, 1990]). It can therefore reasonably be argued that the output of human stereopsis consists mostly of a map of *relative* depth information, conveying a partial depth order between scene points [Julesz, 1971, pp. 176-177].[4] In that context, the main role of eye movements in stereopsis would be to bring the images within *Panum's fusional area*, a disc with a diameter of 6min of arc in the fovea center where fusion can occur [Julesz, 1971, pp. 148] (points can still be vividly perceived in depth for much larger disparities, but they will appear as double images, a phenomenon known as *diplopia*).

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, for example, to quote Julesz [1960, pp. 1133-1134]: "In aerial reconnaissance it is known that objects camouflaged by a complex background are very difficult to detect but jump out if viewed sterescopically." But this is not conclusive: "Though the macropattern (hidden object) is *difficult* to see monocularly, it *can* be seen. Therefore, the evidence is not sufficient to prove that depth can be perceived without monocular macropattern recognition." To gather more conclusive data, Julesz [1960] 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 13.8).

To quote Julesz [1960, p. 1127-1128] 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. ... Of course, depth perception under these conditions takes longer to establish because of the absence of monocular cues. Still, once depth is perceived, it is quite stable. This experiment shows quite clearly that it is possible to perceive depth without monocular macropatterns." By locally perturbing the stereograms in various ways, Julesz proceeds to show that the identification of monocular micropatterns is not necessary for depth perception either. Although monocular perception is certainly also involved in most situations (e.g., making the central region in each image visible by increasing its average brightness has the effect of speeding up depth perception), the conclusion, articulated in [Julesz, 1971], is clear: human binocular fusion cannot be explained by peripheral processes directly associated

[4] Frisby [1980, p. 155] goes even further, suggesting that the depth effect might be a secondary advantage of stereopsis, the primary one being to give the human visual system an effective way of performing grouping and segmentation.

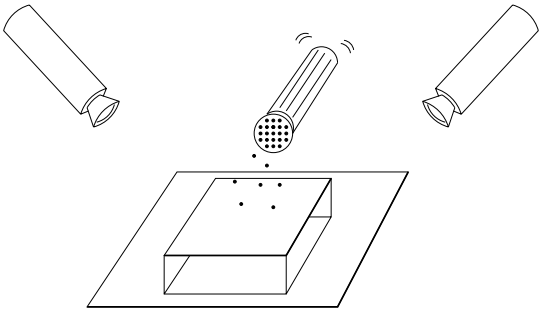


Figure 13.8. Creating random dot stereograms by shaking pepper over a pair of plates observed by two cameras. In the experiments presented in [Julesz, 1960], the two images are of course synthesized by a computer using a random-number generator to decide the dot locations and pixel intensities, that can either be binary values as in the situation described in the text, or more generally random values in the 0..15 range. The two pictures have the same random background and differ in a central region by a constant horizontal offset.

with the physical retinas. Intead, it must involve the central nervous system and an imaginary *cyclopean retina* that combines the left and right image stimuli as a single unit.

Julesz has proposed two models of human stereopsis. The first one represents the binocular field in terms of a finite number of *difference fields* formed by substracting from the first picture the second one shifted by various degrees of disparity [Julesz, 1960]. The matching process amounts in this case to finding various patterns in some of the difference fields. This model has been implemented in the AUTOMAP-1 program that has proven capable of fusing simple randon dot stereograms [Julesz, 1982]. The second model represents each image by a rectangular array of compass needles (or *dipoles*) mounted on spherical joints. A black dot will force the corresponding dipole to point north, and a white dot will force it to point south. After the directions of all dipoles are set, they are coupled to their four neighbors via springs. Finally, the two dipole arrays are superimposed, and left to follow each other's magnetic attraction under various horizontal shifts.

These two models are *cooperative*, with neighboring matches influencing each other to avoid ambiguities and promote a global analysis of the observed scene. The approach proposed by Marr and Poggio [1976] is another instance of such a cooperative process. Their algorithm relies on three constraints: (1) *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), (2) *uniqueness* (a black dot in one image matches at most one black dot in the other picture), and (3) *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 13.9).

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 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).

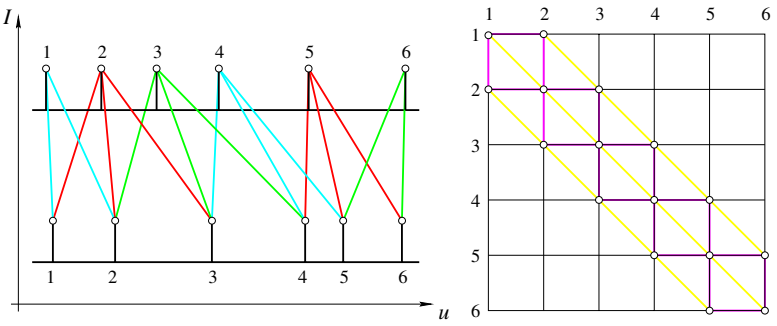


Figure 13.9. A cooperative approach to stereopsis: the Marr-Poggio algorithm [1976]. The left part of the figure shows two intensity profiles along the same scanline of two images. The spikes correspond to black dots. The line segments joining the two profiles indicate possible matches between dots given 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.

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 0. 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 13.10), but not on natural images, perhaps, as suggested by Faugeras [1993], because the constraints it enforces are not sufficient to deal with the complexities of real pictures. Section 13.2 will present a number of algorithms that perform better on most real images, but the original Marr-Poggio algorithm and its implementation retain the interest of offering an early example of a theory of human stereopsis that allows the fusion of random dot stereograms.

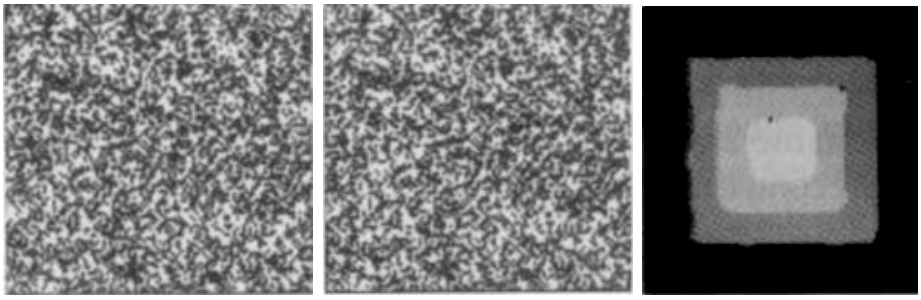


Figure 13.10. From left to right: a random dot stereogram depicting four planes at varying depth (a "wedding cake") and the disparity map obtained after 14 iterations of the Marr-Poggio cooperative algorithm. Reprinted from [Marr, 1982], Figure 3-7.

13.2 Binocular Fusion

13.2.1 Correlation

Correlation methods find pixel-wise image correspondences by comparing intensity profiles in the neighborhood of potential matches, and they are amongst the first techniques ever proposed to solve the binocular fusion problem [Kelly *et al.*, 1977; Gennery, 1980]. More precisely, 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 $\boldsymbol{w}(u, v) \in \mathbb{R}^p$ 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 $\boldsymbol{w}'(u + d, v)$ and define the corresponding (normalized) 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 u, v and d indices 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 13.11).

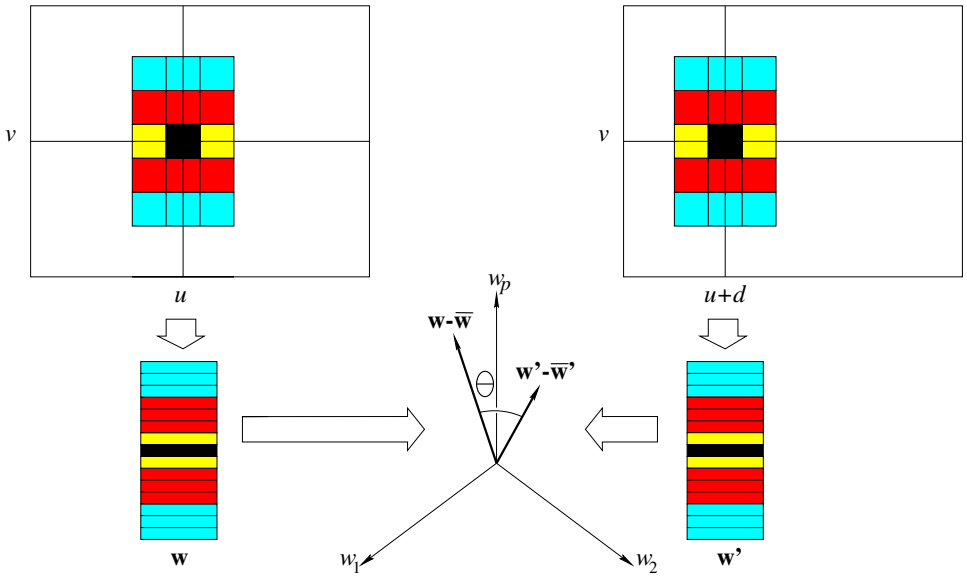


Figure 13.11. Correlation of two 3×5 windows along corresponding epipolar lines. The second window position is separated from the first one by an offset d. The two windows are encoded by vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$ in $\mathbb{R}^{15}$, and the correlation function measures the cosine of the angle θ between the vectors $\boldsymbol{w} - \bar{\boldsymbol{w}}$ and $\boldsymbol{w}' - \bar{\boldsymbol{w}}'$ obtained by substracting from the components of $\boldsymbol{w}$ and $\boldsymbol{w}'$ the average intensity in the corresponding windows.

The normalized correlation function C clearly ranges from -1 to $+1$, and 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 pre-determined range of disparities.[5]

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 correlation function is equivalent to minimizing the norm of the difference between the vectors $(1/|\boldsymbol{w}-\bar{\boldsymbol{w}}|)(\boldsymbol{w}-\bar{\boldsymbol{w}})$ and $(1/|\boldsymbol{w}'-\bar{\boldsymbol{w}}'|)(\boldsymbol{w}'-\bar{\boldsymbol{w}}')$, or equivalently the sum of the squared differences between the pixel values of the normalized windows being compared. 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 establishig stereo correspondences is that they implicitly assume that the observed surface is (locally) parallel to the two image planes (Figure 13.12). 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 [Kass, 1987; Devernay and Faugeras, 1994].

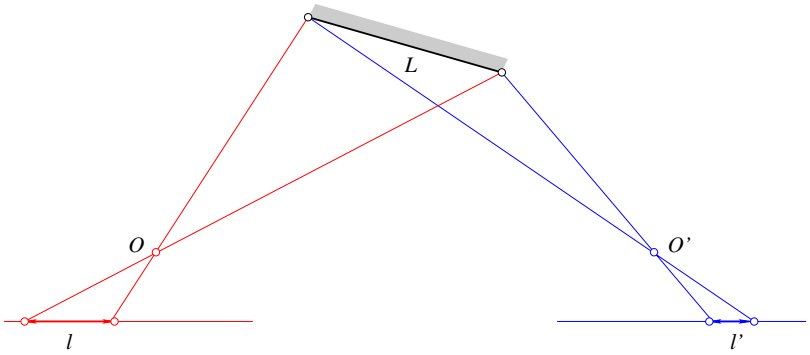


Figure 13.12. The foreshortening of non-frontoparallel surfaces is different for the two cameras: a surface segment with length L projects onto two image segments of different lengths l and l'.

Figure 13.13 shows a reconstruction example obtained by such a method [Devernay and Faugeras, 1994]. In this case, a warped window is associated in the right image with each rectangle in the left image. This window is defined by the

[5] 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 stops.

disparity in the center of the rectangle and its derivatives. An optimization process is used to find the values of the disparity and of its derivatives that maximize the correlation between the left rectangle and the right window, using interpolation to retrieve appropriate values in the right image (see exercises for more details). As shown in Figure 13.13, the reconstruction obtained by this method is clearly better than the reconstruction found by plain correlation.

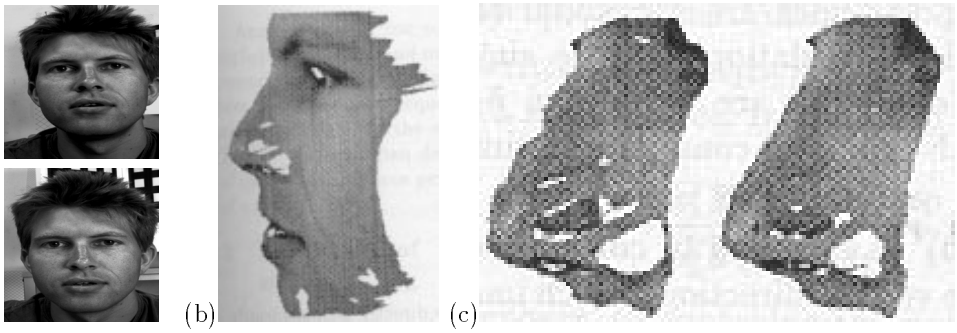


Figure 13.13. Correlation-based stereo matching: (a) a pair of stereo pictures; (b) a texture-mapped view of the reconstructed surface; (c) comparison of the regular (left) and refined (right) correlation methods in the nose region. Reprinted from [Devernay and Faugeras, 1994], Figures 5, 8 and 9.

13.2.2 Multi-Scale Edge Matching

We saw in the last section that slanted surfaces pose problems to correlation-based matchers. Other arguments against correlation can be found in the works of Julesz [1960, p. 1145] ("One might think that the matching of corresponding point domains (instead of corresponding patterns)[6] could be achieved by searching for a best fit according to some similarity criterion (e.g., maximal cross-correlation). ... But such a process cannot work. If the zone [used to search for correspondences] is small, noise can easily destroy any zone-matching; if the zone size is increased, ambiguities arise at the boundaries of objects which are at different distances.") and Marr [1982, p. 105] ("...by and large the primitives that the processes operate on should correspond to physical items that have identifiable physical properties and occupy a definite location on a surface in the world. Thus one should not try to carry out stereo matching between gray-level intensity arrays, precisely because a pixel corresponds only implicitly and not explicitly to a location on a visible surface."). These arguments suggest that correspondences should be found at a variety of scales, and that matches between (hopefully) physically-significant image features such as edges should be prefered to matches between raw pixel intensities.

[6] This remark shows, by the way, that the random dot stereogram experiments of Julesz do not dismiss, at least in his thought, the possibility of a correlation-based process as opposed to a higher-level, pattern recognition one.

Marr and Poggio [1979] propose an algorithm that follows these two principles. Its overall structure is quite simple, as described below.

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 control eye vergence and cause unmatched regions at smaller scales to come into correspondence.

Algorithm 13.1: *The Marr-Poggio-Grimson multi-scale algorithm for establishing stereo correspondences [Marr and Poggio, 1979; Grimson, 1981a].*

Note that 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 [1981a] has shown 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, 1981a] 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 it is illustrated by Figure 13.14. 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 [1981a], and extensively tested on random dot stereograms and natural images. An example appears in Figure 13.15.

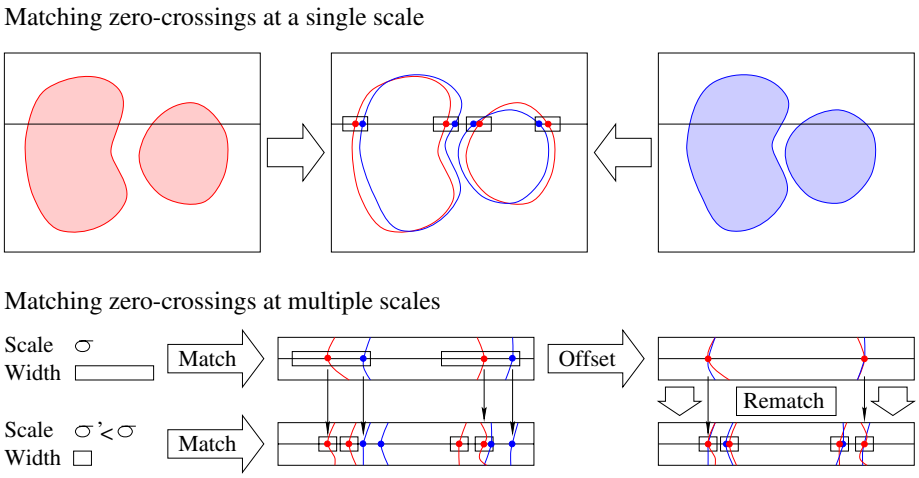


Figure 13.14. Multi-scale matching of zero crossings: the eye movements (or equivalently the image offsets used in matching) are controlled by seeking image regions that have been assigned a disparity value at a scale σ' but not at a scale $\sigma < \sigma'$. These values are used to refine the eye positions and bring the corresponding regions within matchable range. The disparity value associated with a region can be found by various methods, for example by averaging the disparity values found at each matched zero crossing within it.

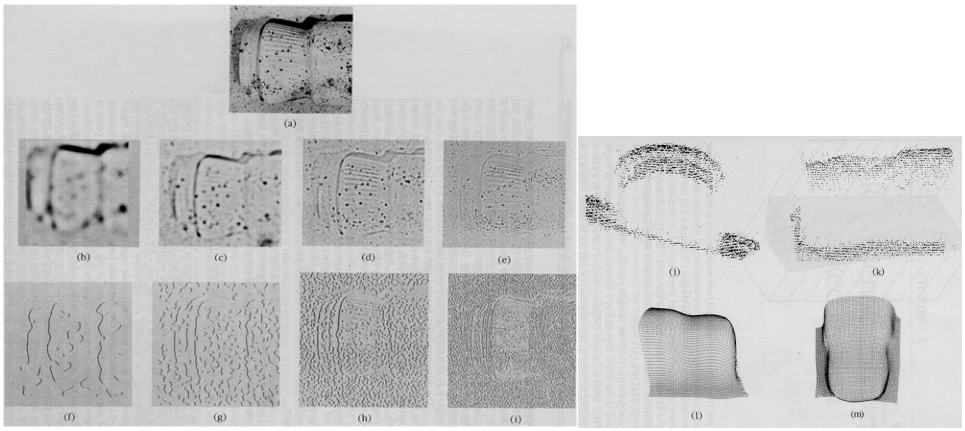


Figure 13.15. Applying the multi-scale matching algorithm of Marr and Poggio [1979] to a pair of images: (a) one of the pictures in the stereo pair; (b)-(e) its convolution with four ∇^2_σ filters of increasing sizes; (f)-(i) the corresponding zero crossings; (j)-(k) two views of the disparity map obtained after matching; (l)-(m) two views of the surface obtained by interpolating the reconstructed dots using the algorithm described in [Grimson, 1981b]. Reprinted from [Marr, 1982], Figure 4-8.

13.2.3 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 13.16(left)). This is the so-called *ordering constraint* that has been used in stereo circles since the early eighties [Baker and Binford, 1981; Ohta and Kanade, 1985]. Interestingly enough, this constraint may not be satisfied by real scenes, in particular when small solids occlude parts of larger ones (Figure 13.16(right)), or more rarely, at least in robot vision, when transparent objects are involved.

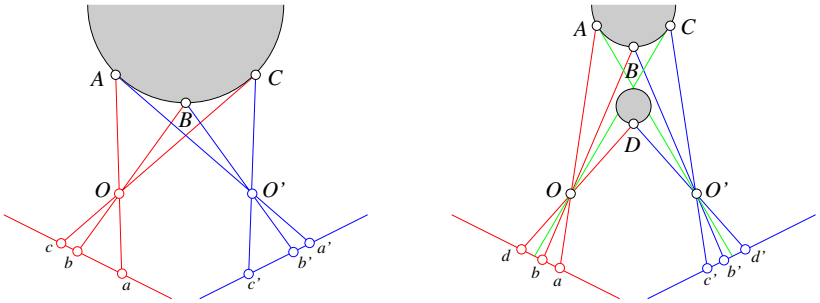


Figure 13.16. Ordering constraints. In the (usual) case shown in the left part of the diagram, the order of feature points along the two (oriented) epipolar lines is the same, and it is the inverse of the order of the scene points along the curve where the observed surface intersects the epipolar plane. 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.

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 13.17). 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 13.17(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.

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, and arcs represent matches between left and right intensity profile intervals bounded by the features of the corresponding nodes (Figure 13.17(right)). This optimization problem can be solved using dynamic programming as shown in Algorithm 13.2 below.

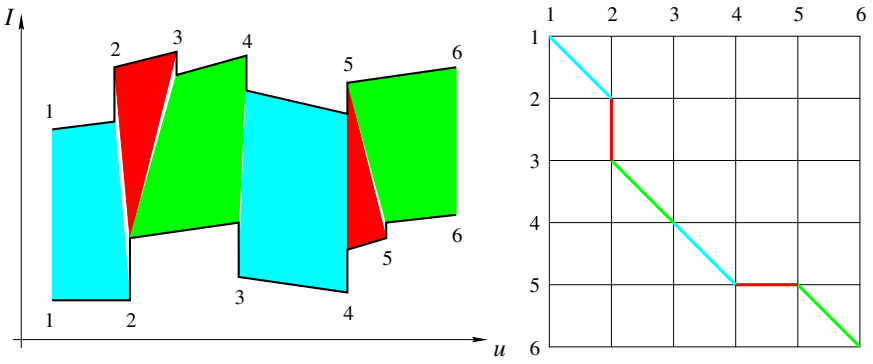


Figure 13.17. Dynamic programming and stereopsis: the left part of the figure shows two intensity profiles along matching epipolar lines. The polygons joining the two profiles indicate matches between successive intervals (some of the matched intervals may have zero length). The right part of the diagram 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.

```
% 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) ← +∞; B(k,l) ← nil;
    % Loop over all inferior neighbors (i,j) of (k,l).
    for (i,j) ∈ Inferior-Neighbors(k,l) do
      % Compute new path cost and update backward pointer if necessary.
      d ← C(i,j) + Arc-Cost(i,j,k,l);
      if d < C(k,l) then C(k,l) ← d; B(k,l) ← (i,j) endif;
      endfor;
    endfor;
  endfor;
% Construct optimal path by following backward pointers from (m,n).
P ← {(m,n)}; (i,j) ← (m,n);
while B(i,j) ≠ nil do (i,j) ← B(i,j); P ← {(i,j)} ∪ P endwhile.
```

Algorithm 13.2: *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.*

As given, Algorithm 13.2 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.[7] 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 13.18 shows a sample result taken from [Ohta and Kanade, 1985].

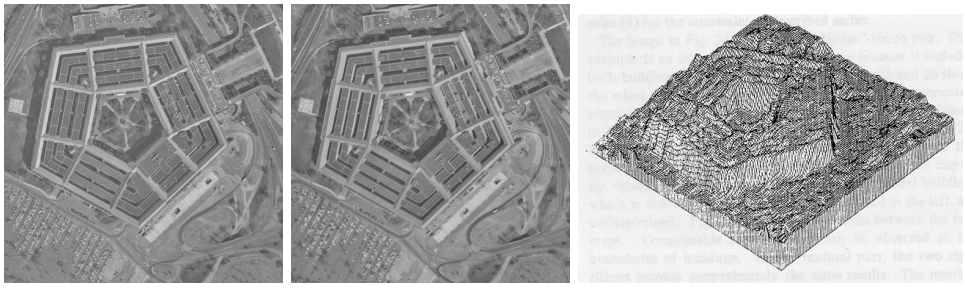


Figure 13.18. Two images of the Pentagon and an isometric plot of the disparity map computed by the dynamic-programming algorithm of Ohta and Kanade [1985]. Reprinted from [Ohta and Kanade, 1985], Figures 18 and 22.

13.3 Using More Cameras

13.3.1 Trinocular Stereo

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 13.19): 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 12, 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.

The trifocal tensor introduced in Chapter 12 can be used to also predict the tangent line to some image curve in one image given the corresponding tangents in the other images (Figure 13.20): given matching tangents l_2 and l_3 in images 2 and 3, we can reconstruct the tangent l_1 in image number 1 using Eq. (12.2.4),

[7] 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].

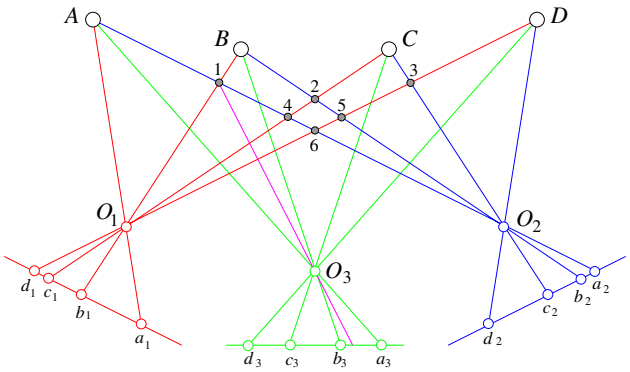


Figure 13.19. The small grey discs indicate the incorrect reconstructions associated with the left and right images of four points. The addition of a central camera removes the matching ambiguity: none of the corresponding rays intersects any of the six discs. Alternatively, matches between points in 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.

rewritten here as:

$$\boldsymbol{l}_1 \approx \begin{pmatrix} \boldsymbol{l}_2^T \mathcal{G}_1^1 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^2 \boldsymbol{l}_3 \\ \boldsymbol{l}_2^T \mathcal{G}_1^3 \boldsymbol{l}_3 \end{pmatrix}, \quad \text{where} \quad \mathcal{G}_1^i = \boldsymbol{t}_2 \boldsymbol{R}_3^{iT} - \boldsymbol{R}_2^i \boldsymbol{t}_3^T \quad \text{for} \quad i = 1, 2, 3,$$

$\boldsymbol{R}_2^i$ and $\boldsymbol{R}_3^i$ $(i = 1, 2, 3)$ denote the columns of the rotation matrices $\mathcal{R}_2$ and $\mathcal{R}_3$ associated with cameras 2 and 3, and $\boldsymbol{t}_2$ and $\boldsymbol{t}_3$ denote the corresponding translation vectors (here "$\approx$" is used to denote equality up to scale).

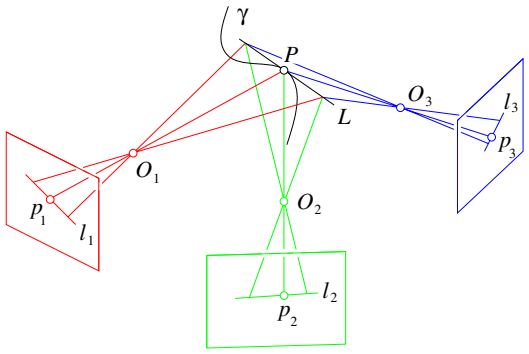


Figure 13.20. Given matches between the points p_1 and p_2 and their tangents l_1 and l_2 in two images, it is possible to predict both the position of the corresponding point p_3 and tangent l_3 in a third image.

Algorithms for trinocular stereo include [Milenkovic and Kanade, 1985; Yachida *et al.*, 1986; Ayache and Lustman, 1987; Robert and Faugeras, 1991]. An example is shown in Figure 13.21.

Figure 13.21. Three images and the correspondences between edges found by the algorithm of Robert and Faugeras [1991; 1995]. Reprinted from [Robert and Faugeras, 1995], Figure 9.

As shown in [Robert and Faugeras, 1991; Robert and Faugeras, 1995], it is in fact also possible to predict the curvature at a point on some image curve given the corresponding curvatures in the other images (see exercises). This fact can be used to effectively reconstruct curves from their images [Faugeras, 1993; Robert and Faugeras, 1995].

13.3.2 Multiple-Baseline Stereo

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 a multi-camera 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 (this is of course, as shown earlier, equivalent to adding the correlation functions associated with the images).

Figure 13.22 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. On the other hand, adding more cameras provides a clear minimum corresponding to the correct match.

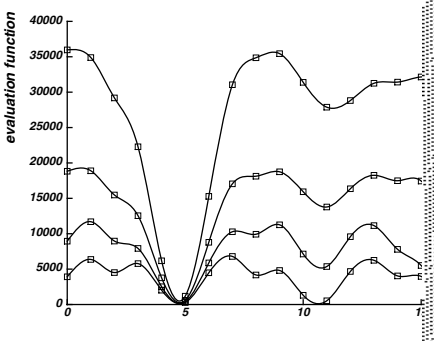


Figure 13.22. Combining multiple-baseline stereo pairs: is plotted here as a function of the inverse depth for various data are taken from a scanline near the top of the images intensity is nearly periodic. The diagram clearly shows tha becomes less and less ambiguous as more images are added. Kanade, 1993], Figure 7.

Figure 13.23 shows a sequence of ten rectified ima reconstructed by the algorithm.

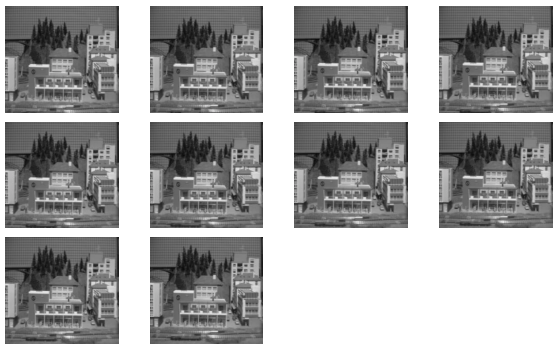


Figure 13.23. A series of ten images and the corresponding reconstruction. The gridboard near the top of the images is the source for the nearly periodic brightness signal giving rise to ambiguities in Figure 13.22. Reprinted from [Okutami and Kanade, 1994], Figure 13(c).

13.4 Notes

The fact that disparity gives rise to stereopsis in human beings was first demonstrated by Wheatstone's invention of the stereoscope [Wheatstone, 1838]. The fact that disparity is sufficient for stereopsis without eye movements was demonstrated

shortly afterwards by Dove [1841], using illumination provided by an electric spark and much too brief for eye vergence to take place [Helmholtz, 1909, p. 455]. Human stereo vision is further discussed in the classical works of Helmholtz [1909] and Julesz [1971] as well as the books by Frisby [1980] and Marr [1982]. Theories of human binocular perception not presented in this chapter for lack of space include [Koenderink and Van Doorn, 1976a; Pollard *et al.*, 1970; Anderson and Nayakama, 1994].

Excellent treatments of machine stereopsis can be found in the books of Grimson [1981b], Marr [1982], Horn [1986] and Faugeras [1993]. Marr focusses on the computational aspects of human stereo vision, while 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].

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 13.24). Indeed, the corresponding image edges are in this case viewpoint dependent, and matching them yields erroneous reconstructions.

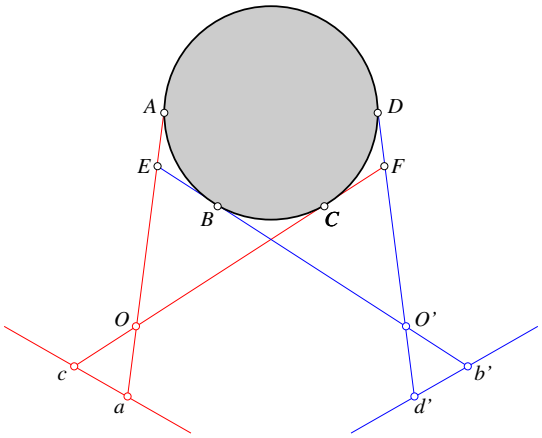


Figure 13.24. Stereo matching fails at smooth object boundaries: for narrow baselines, the pairs (c, d') and (a, b') will be easily matched by most edge-based algorithms, yielding the fictitious points F and E as the corresponding three-dimensional reconstructions.

As shown in [Arbogast and Mohr, 1991; Vaillant and Faugeras, 1992; Cipolla and Blake, 1992; Boyer and Berger, 1996] and the exercises, 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

grey-level matching. The former is accurate near surface markings but only yields a sparse set of measurements, while 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 refered to [Grimson, 1981b; 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 will be discussed in Chapter 26.

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 [Aloimonos *et al.*, 1987; Bajcsy, 1988; Ahuja and Abbott, 1993; Brunnström *et al.*, 1996].

13.5 Assignments

Exercises

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

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

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

4. Give an algorithm for generating single-image random dot stereograms.

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 $\lambda > 0$.

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

7. Recursive computation of the correlation function:

(a) Show that

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

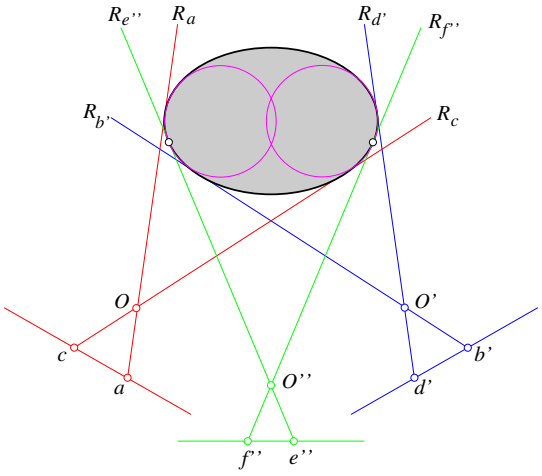
(b) Show that the average intensity $\bar{I}$ can be computed recursively, and estimate the cost of the incremental computation.

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

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.

9. Show how to predict curvature in one image from curvature measurements in two other pictures.

10. Three-camera reconstruction of smooth surfaces' occluding contours: show that, in the planar case, three matching rays provide enough constraints to reconstruct the circle of curvature as shown below.



Programming Assignments

1. Implement the rectification process.

2. Implement the algorithm developed in Exercise 4 for generating single-image random dot stereograms.

3. Implement a correlation-based approach to stereopsis.

4. Implement a multi-scale approach to stereopsis.
5. Implement a dynamic-programming approach to stereopsis.
6. Implement a trinocular approach to stereopsis.