Issues In Parallel Stereo Matching

by

Walter Eisner Gillett

A.B. Mathematics/Physics, Harvard College
(1981)

Submitted to the Department of Brain and Cognitive Sciences
in Partial Fulfillment of the Requirements for the Degree of

Master of Science In Brain and Cognitive Sciences

at the

Massachusetts Institute of Technology
May 1988

Signature of Author______________________________
Department of Brain and Cognitive Sciences
May, 1988

Certified by______________________________
Dr. Tomaso A. Poggio
Thesis Supervisor

Accepted by______________________________
Dr. William G. Quinn, Jr.
Chairman, Department Graduate Committee

Issues In Parallel Stereo Matching

by

Walter Eisner Gillett

Submitted to the Department of Brain and Cognitive Sciences
in Partial Fulfillment of the Requirements for the Degree of
Master of Science In Brain and Cognitive Sciences
at the Massachusetts Institute of Technology
May 1988

Abstract

Discontinuities of surface properties are the most important locations in a scene. In particular, depth discontinuities are crucial for image segmentation because they often coincide with object boundaries. Standard approaches to discontinuity detection postprocess disparity data from a stereo algorithm, treating the stereo algorithm as a "black box". We have developed two techniques for locating depth discontinuities using information internal to the stereo algorithm of [Drumheller, Poggio 1986], rather than by postprocessing the stereo data. One technique analyzes patchwise matching scores internal to the algorithm, and the other makes use of the effects of occlusion.

We have also improved and extended the original stereo algorithm. Edge-based matching now uses the Canny edge detector [Canny, 1983; Little et.al. 1987] rather than Marr-Hildreth zero-crossings [Marr, Hildreth 1980] for better localization; takes advantage of a gradient matching constraint; and normalizes the matching scores to take into account variations in edge density. Finally, we have addressed the problem of identifying areas that are outside of the matcher's fusional range.

This research is part of a project to build a "Vision Machine" [Poggio and staff, 1988] at MIT that attempts to integrate the output from different early vision modules in order to perform such tasks as recognition and navigation in close to real-time. Massively parallel hardware is employed to achieve high performance. Our study of stereo matching therefore also represents a study in efficient parallel algorithms.

Thesis Supervisor: Dr. Tomaso Poggio
Professor of Vision Sciences and Biophysics

Acknowledgements

Many members of the Artificial Intelligence Laboratory helped to bring these efforts to fruition. In particular, I'd like to thank Jim Little and Mike Drumheller for many thoughtful discussions; Mike Drumheller for the stereo software that was the starting point for this research; Harry Voorhees, Paul Oppenheimer, and Dave Clemens for their Vision Utilities; and Jerry Roylance, Chris Lindblad, and David Siegel for the health and well-being of the Laboratory's computing facilities. I'd also like to thank my wife Sharon for her enthusiasm and affection, and Jan Ellertsen for efficient administration, snatching order from the jaws of chaos. Most of all, I am grateful to Tomaso Poggio for excellent ideas, advice, and energetic support.

This thesis describes research performed at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the U.S. Department of Defense under Office of Naval Research contract N00014-85-K-0124 and Army contract DACA76-85-C-0010. Primary support for this research came from NIGMS Training Grant #T32-GM07484, administered by the Department of Brain and Cognitive Sciences. The financial support of Hughes Aircraft Corporation is also gratefully acknowledged.

Contents

1 Introduction

Vision is the process of computing a symbolic description of a scene from one or more images, two-dimensional arrays of brightness values captured by a camera or an eye. We can consider vision in terms of two hierarchies. The first is Marr's information-processing hierarchy [Marr, 1982]. An information-processing task can be analyzed at three levels: the computational theory, the representation and algorithm, and the hardware implementation. This thesis is primarily concerned with the first two levels.

The second hierarchy describes the flow of information through successive stages of processing. From this point of view, stereopsis is part of *early vision*, the first level of visual processing. In Marr's model, depth data contributed by stereo is used in the construction of the*$2^{1}/_{2}$-D sketch*, a viewer-centered description of the visible surfaces in a scene. In [Barrow and Tennenbaum, 1981], depth data forms an *intrinsic image* , one of a set of parametric images describing spatial properties of the scene.

1.1 Issues in stereo matching

The vision group at MIT is currently building a "Vision Machine" [Poggio and staff, 1988] that attempts to integrate the output from different early vision modules in order to perform tasks such as recognition and navigation in close to real-time. The integration stage computes a map of the visible discontinuities in the scene, identifying their physical origin. This thesis contributes to the Vision Machine effort in several directions. Our major achievement is the development of techniques for locating depth discontinuities within the stereo module, rather than by postprocessing the stereo data. We have devised two techniques for discontinuity location, one based on an analysis of patchwise matching scores internal to the algorithm, and the other based on the effects of occlusion. In a related effort, we have explored improving the performance of stereo near depth discontinuities.

We have also improved and extended the original stereo module of [Drumheller and Poggio, 1986]. Edge-based matching now uses the Canny edge detector [Canny, 1983; Little et.al. 1987] rather than Marr-Hildreth zero-crossings [Marr, Hildreth 1980] for better localization; takes advantage of a gradient matching constraint; and normalizes the matching scores to take into account variations in edge density.

Finally, we have addressed the problem of identifying areas that are outside of the matcher's fusional range. Due to time and space limitations, the matcher can only search a limited set of disparities, the fusional range. It is important to be able to identify parts of the scene that are outside of the fusional range, rather than just blindly assigning the disparity value in range with the best matching score.

1.2 Exploiting parallelism

Early vision is computationally intensive. The computation is mostly local and isotropic, meaning that the same processing takes place at separate locations in the images. This suggests that a SIMD parallel architecture is a good choice to meet the performance requirements of the Vision Machine. Specifically, our computational engine is the Connection Machine [Hillis, 1985], described briefly below.[1]

[1] The Connection Machine description was adapted from [Poggio 1988].

1.2.1 The Connection Machine

The CM-1 version of the Connection Machine [Hillis, 1985] is a parallel computing machine with between 16K and 64K processors, operating under a single instruction stream broadcast to all processors. It is a Single Instruction Multiple Data (SIMD) machine; all processors execute the same control stream. Each processor is a simple 1-bit processor with 4K bits of memory. To allow the machine to manipulate data structures with more than 64K elements, the Connection Machine supports virtual processors. A single physical processor can operate as a set of multiple virtual processors by serializing operations in time, and partitioning the memory of each processor. This is otherwise invisible to the user.

The processors can be envisioned as being the vertices of a 16-dimensional hypercube (the machine is physically wired as a 12-dimensional hypercube, where each vertex is a chip containing 16 processors). There are two modes of communication among the processors. First, the NEWS network (North-East-West-South) allows rapid communication between neighboring processors in a two-dimensional mesh. Second, the router allows messages to be sent from any processor to any other processor in the machine. Each processor in the Connection Machine is identified by its hypercube address, imposing a linear order on the processors. This address denotes the destination of messages handled by the router. Messages pass along the edge of the hypercube from source processors to destination processors. The Connection Machine also has facilities for returning to the host machine the result of various operations on a field in all processors; it can return the global maximum, minimum, sum, logical AND, and logical OR of the field.

The Connection Machine operates under the control of a conventional front end such as a Lisp Machine or a VAX. Software developed by the vision group is written in *LISP, a parallel extension of Common Lisp. Parallel extensions of the languages C and FORTRAN are also available. Programs that use the Connection Machine are developed, tested, and debugged on the front end in the same fashion as serial programs.

1.3 The Drumheller-Poggio parallel stereo algorithm

The Drumheller-Poggio algorithm [Drumheller and Poggio, 1986] served as an experimental testbed for the research described in this thesis. An extended version of the algorithm runs as part of the Vision Machine; the resulting stereo data is one of the inputs to the MRF-based integration stage. This section briefly reviews the original stereo algorithm.[1]

Stereo matching is an ill-posed problem [Bertero et.al., 1987] that cannot be solved without taking advantage of natural constraints. The *continuity constraint* [see, for instance, Marr, Poggio 1976] asserts that the world consists primarily of piecewise smooth surfaces. If the scene contains no transparent objects, then the *uniqueness constraint* applies: there can be only one match along the left or right lines of sight. If there are no narrow occluding objects, the *ordering constraint* [Poggio and Yuille, 1984] holds: any two points must be imaged in the same relative order in the left and right eyes.

The specific a priori assumption on which the algorithm is based is that the disparity - that is, the depth of the surface - is locally constant in a small region surrounding a pixel. It is a restrictive assumption which, however, may be a satisfactory *local* approximation in many cases (it can be extended to more general surface assumptions in a straightforward way but at high computational cost). Let $E_L(x,y)$ and $E_R(x,y)$ represent the left and right image of a stereo pair or some transformation of it, such as filtered images or a map of the zero-crossings in the two images (more generally, they can be maps containing a feature vector at each location (x,y) in the image).

We look for a discrete disparity d(x,y) at each location (x,y) in the image that minimizes

$$|| E_L(x,y) - E_R(x+d(x,y),y) ||_{\text{patch}_i} \qquad (1.1)$$

where the norm is a summation over a local neighborhood centered at each location (x,y); d(x) is assumed constant in the neighborhood. Equation (1.1) implies that we should look at each (x,y) for d(x,y) such that

$$\int_{\text{patch}_i} \left(E_L(x,y)E_R(x+d(x,y),y)\right)^2 dxdy \qquad (1.2)$$

is maximized.

[1] The discussion of the Drumheller-Poggio stereo algorithm was adapted from [Drumheller, Poggio 1986].

The algorithm actually implemented on the Connection Machine is somewhat more complicated, since it involves geometric constraints that affect the way the maximum operation is performed (see [Drumheller and Poggio, 1986]). The original implementation uses zero-crossings of the Marr-Hildreth edge detector [Marr and Hildreth, 1980] obtained from each image for E_L and E_R. (Extensions to other types of features are described later in the thesis.) The Marr-Hildreth edge detector will be referred to in the rest of the thesis by the alternative name DOG edge detector, where DOG refers to the Difference-Of-Gaussians approximation to the Laplacian-of-a-Gaussian operator. In more detail, the algorithm is composed of the following steps:

- Compute features for matching.
- Compute potential matches between features.
- Determine the degree of continuity around each potential match.
- Choose correct matches based on the constraints of continuity, uniqueness, and ordering.

Potential matches between features are computed in the following way. Assuming that the images are registered so that the epipolar lines are horizontal, the stereomatching problem becomes one-dimensional: an edge in the left image can match any of the edges in the corresponding horizontal scan line in the right image. Sliding the right image over the left image horizontally, we compute a set of *potential match planes*, one for each horizontal disparity. Let p(x,y,d) denote the value of the (x,y) entry of the potential match plane at disparity d. We set p(x,y,d) = 1 if there is an edge at location (x,y) in the left image and a compatible edge at location (x-d,y) in the right image; otherwise set p(x,y,d) = 0. In the case of the DOG edge detector, two edges are compatible if the signs of the convolution for each edge (the *edge polarities*) agree.

To determine the degree of continuity around each potential match (x,y,d), we compute a local support score

$$s(x,y,d) = \sum_{R} p(x,y,d) \qquad (1.3)$$

where R is a small neighborhood of (x,y,d) within the *d-th* potential match plane. In effect, nearby points in R can "vote" for the disparity d. If the continuity constraint is satisfied near (x,y,d) then R will contain many votes and the score s(x,y,d) will be high. The above sum corresponds to the integral over the patch in Equation (1.2).

Finally, we attempt to select the correct matches by applying the uniqueness and ordering constraints (see above). To apply the uniqueness constraint, each match suppresses all other matches along the left and right lines of sight with weaker scores. To enforce the ordering constraint, if two matches are not imaged in the same relative order in left and right views, we discard the match with the smaller support score. In effect, each match suppresses matches with lower scores in its forbidden zone [Poggio and Yuille, 1984]. This step corresponds to choosing the disparity value that maximizes the integral of Equation (1.2).

Figure 1.1 shows a stereo scene and depth data derived by the algorithm. Displaying non-binary data on the printed page entails a loss of spatial resolution. We will usually display isodisparity contours of the interpolated depth map rather than the depth data itself. Unless otherwise specified, stereo results in this thesis have been obtained with the following parameters:

- edge-based matching using Canny edge detector
- gradient matching constraint: active
- support region width = 23
- dense depth data (matches decided not just at edges)
- non-normalized matching scores
- 256x256 images, downsampled from 512x512 using simple averaging (we use downsampled images because of CM-1 memory limitations)

The meaning of these terms will become clear later in the thesis.

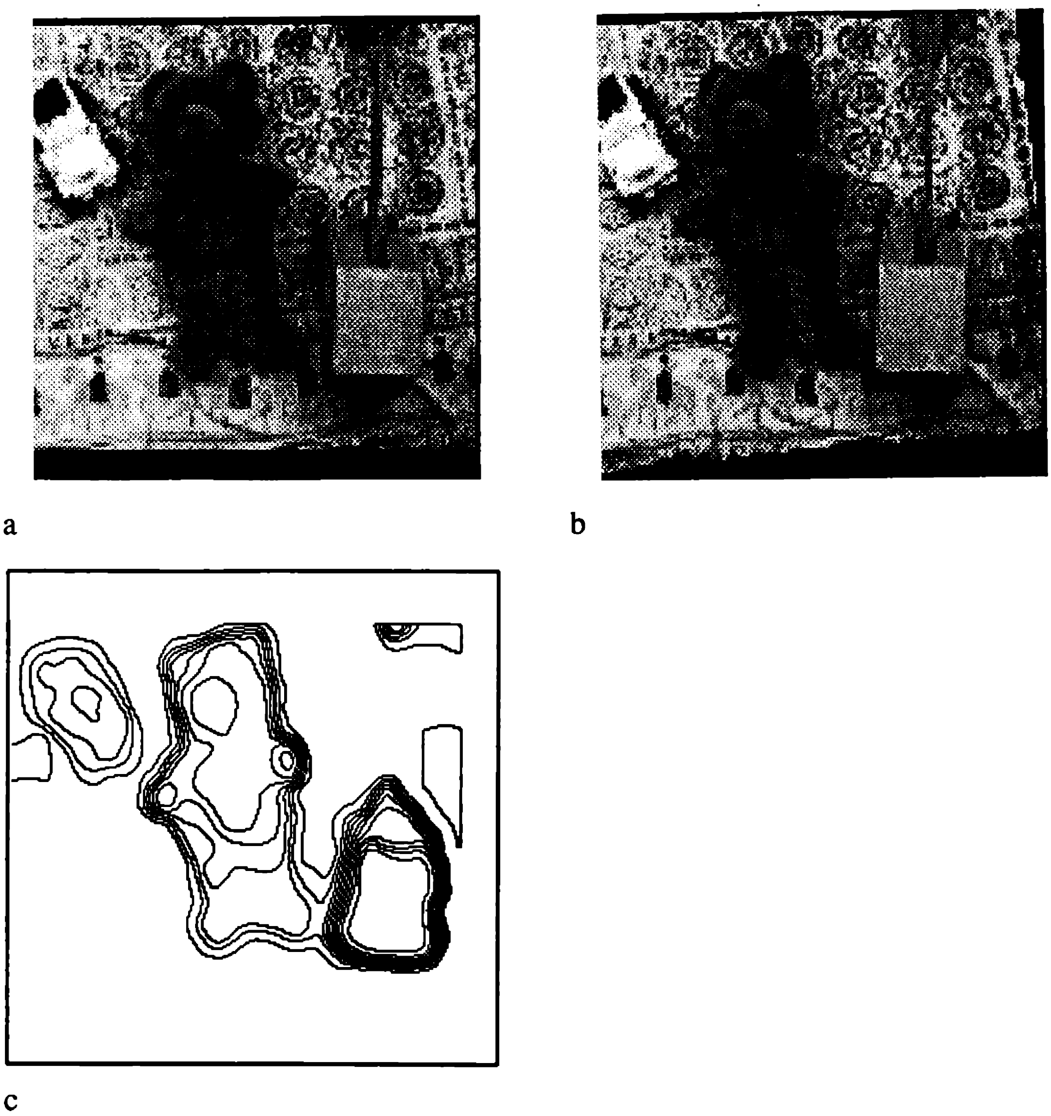


Figure 1.1 (a) Left view of truck, teddy bear, and crane. (b) Right view. (c) Isodisparity contours.

1.3.1 Comparison with the Marr-Poggio-Grimson algorithm

The Drumheller-Poggio algorithm is similar in spirit to the first stereo algorithm proposed by Marr and Poggio [Marr, Poggio 1976], a cooperative algorithm in which potential matches reinforce other matches that lie on the same surface and inhibit other matches that violate the uniqueness constraint. It is interesting to contrast the Drumheller-Poggio algorithm with the later Marr-Poggio-Grimson algorithm, hereinafter referred to as DP and MPG, respectively. We will consider both issues of stereo theory and computational efficiency.

DP solves the correspondence problem through a kind of cross-correlation, comparing patches from each image and assigning scores to each potential match. Matches are then pruned according to the uniqueness and ordering constraints. DP has several strengths. First, given a reasonable patch size, DP can search a large disparity range without being confused by false targets, because the patch contains a lot of information. Second, the ordering constraint eliminates many bad matches; there is no analogy to the ordering constraint in MPG. A weakness of DP is the use of flat support neighborhoods (a potential match at (x,y,d) gathers support only from other potential matches at the same disparity). In order for this to work, the image patches to be compared must have the same shape - there can be no foreshortening due to differences in viewing angle. If the distance from the viewer to objects in the scene is large compared to the optical baseline, then this condition holds. An equivalent assumption is that perspective projection can be approximated by the combination of orthographic projection and linear scale. As [Drumheller, Poggio 1986] point out, this assumption can be eliminated by using multiple support neighborhoods, albeit at a significant computational cost.

MPG solves the correspondence problem by matching pairs of edges (DOG zero-crossings) using a multiscale analysis. The false target problem is reduced by restricting the disparity range searched at a given scale according to the width of the smoothing filter used in edge detection. A weakness of this approach is the transition between scales. Disparity values determined at a coarse scale are used to focus the search at the next, finer scale. However, it is possible that in some areas of the scene there may be no information at the coarse scale, making it impossible to determine an appropriate disparity range in those areas for the next level of search. An advantage of MPG is that it has a very efficient serial implementation (it may also have an efficient parallel implementation - see below). MPG operates on edges, which are sparse, rather than examining every pixel. The use of multiple scales makes MPG roughly logarithmic in computation time with respect to the total disparity range, whereas the computation time of DP increases linearly with the disparity range.

The matching scores of DP are valuable information. They provide a confidence level associated with each match that can be used to arbitrate between competing matches, as in forbidden zone suppression (using the ordering constraint). The description of the stereo algorithm in section 1.3 implies that scores are computed only for points p and q that are potential matches (there are compatible edges at p and q). In fact, although matches are only permitted at potential match sites, matching scores are computed everywhere with no additional computation. These extra scores can be used to derive dense stereo results: a strong score at (x,y,d) indicates that the point (x,y) in the left image probably matches the point (x+d,y) in the right image, whether or not the two points coincide with edges. Computing depth between edges by using the scores is a more informed approach than using an interpolation technique that must make *a priori* assumptions about the surfaces present in the scene. The extra scores also help to suppress bad matches within occluded areas of the scene, a point that we will return to in our discussion of occlusion (section 2.1.2). All stereo data used in the thesis is dense unless otherwise specified.

The remainder of this section is devoted to speculation. First we will discuss the possible use of multiscale analysis in DP. An important DP parameter is the size of the image patches that are compared. Patches must be large to overcome the false target problem, but small for good localization. Multiple scales could allow us to avoid this tradeoff by using the results from coarse scales to focus the search at finer scales, as in MPG. There are several ways to vary the scale: change the patch size; change the scale of the edge detector (like MPG); or use a pyramid scheme with subsampled versions of the original images. The last approach may be the most appealing because smaller images require fewer virtual processors and can therefore be processed more quickly. Note that a multiscale DP would probably still be linear in computation time with respect to the total disparity range. If one wants to check, say, an 80-pixel disparity range on the Connection Machine by sliding the right image over the left, it will take a full 80 steps, even if each pixel in the left image is only interested in a specific 10-pixel part of the disparity range.

Second, let us consider a parallel implementation of MPG. For the sake of efficiency, it is advisable to move from a spatial representation - one processor per pixel - to an edge-based representation - one processor per edge. Specifically:

1) Extract edges at a variety of scales.
2) For each scale (proceeding from coarse to fine):
   a. Move the edges for the left image into processors with sequential hypercube addresses, ordering the edges lexicographically by y, x (in other words, from top to bottom across scan lines and from left to right within scan lines). Do the same for the edges of the right image.
   b. Slide the right edge vector over the left edge vector. Each left edge looks for matching right edges within the disparity range determined by disparity values from the previous scale. Edges carry with them their original (x,y) position as well as other information required for matching.

This is only the roughest sketch of an algorithm, but should convince the reader that MPG has a natural parallel implementation. Parallel MPG would benefit from fewer virtual processors than DP, but multiscale on the Connection Machine requires running through the full disparity range for each scale, as discussed above. The stereo results should be the same, since parallel MPG could mirror the computation of serial MPG.

2 Depth discontinuities

Discontinuities of surface properties are the most important locations in a scene [Poggio and staff, 1988]. In particular, depth discontinuities often coincide with object boundaries. The major result of this thesis is that data internal to the stereo algorithm can be used to locate depth discontinuities. We have also explored the use of this information to improve the performance of the stereo algorithm near boundaries.

2.1 Detecting discontinuities within stereo

We describe two discontinuity detection techniques in this section, one based on an analysis of matching scores for different disparities and the other arising from the effects of occlusion.

2.1.1 Close winners

The *close winners* technique is based on an analysis of stereo matching scores. Before proceeding with the analysis, we need to introduce some terminology. For each point p = (x,y) in the left image and q = (x+d,y) in the right image, the matcher computes a score s(x,y,d) indicating the likelihood that p matches q, i.e., that p and q are images of the same physical point in the scene. The matcher examines only disparities in the fixed interval [id,fd], called the *fusional range*, where the user-controlled parameters id and fd are the initial and final disparities, respectively. Define the *score vector* v(p) = {s(x,y,id), s(x,y,id+1), ..., s(x,y,fd)}, the sequence of matching scores for point p.

We begin with a simple example. Figure 2.1 shows a random-dot stereogram (RDS) and a schematic representation of the scene (left view), which fuses to yield the impression of a square floating in front of the background. The square is 192x192, centered in the 256x256 left view. The dark strip on the left-hand side is an occluded part of the background that can be seen in the left view but not the right view.

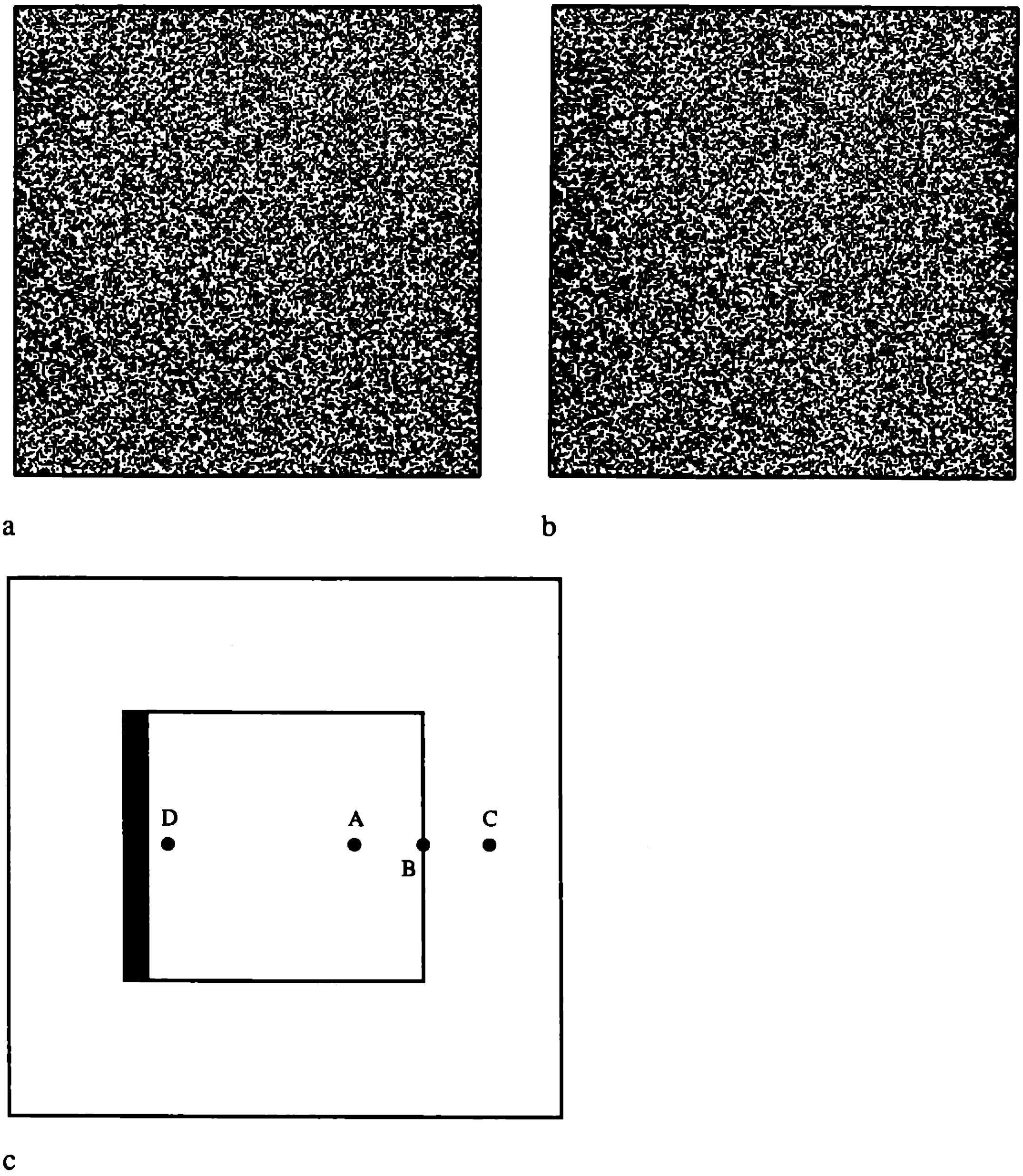


Figure 2.1 (a) Left view of RDS. (b) Right view. (c) Line drawing of scene: floating square.

Point B is located on the boundary of the floating square. The local support neighborhood of point B is divided between points on the square and points on the background. Almost half of the edges in the neighborhood (perhaps more than half) will vote for the wrong disparity, namely the background disparity. If we plot the score vector v(B) as a graph of matching score vs. disparity, the graph should be bimodal, with one peak at the foreground disparity and another peak at the background disparity.[1] By contrast, v(A) and v(B)

[1] [Spoerri, Ullman, 1987] used a similar scheme to detect motion boundaries; [Voorhees, 1987] employed a related technique to locate texture boundaries.

should be unimodal, since their support regions are all at the same disparity. Figure 2.2 shows real score vectors computed for the random-dot stereogram (RDS).

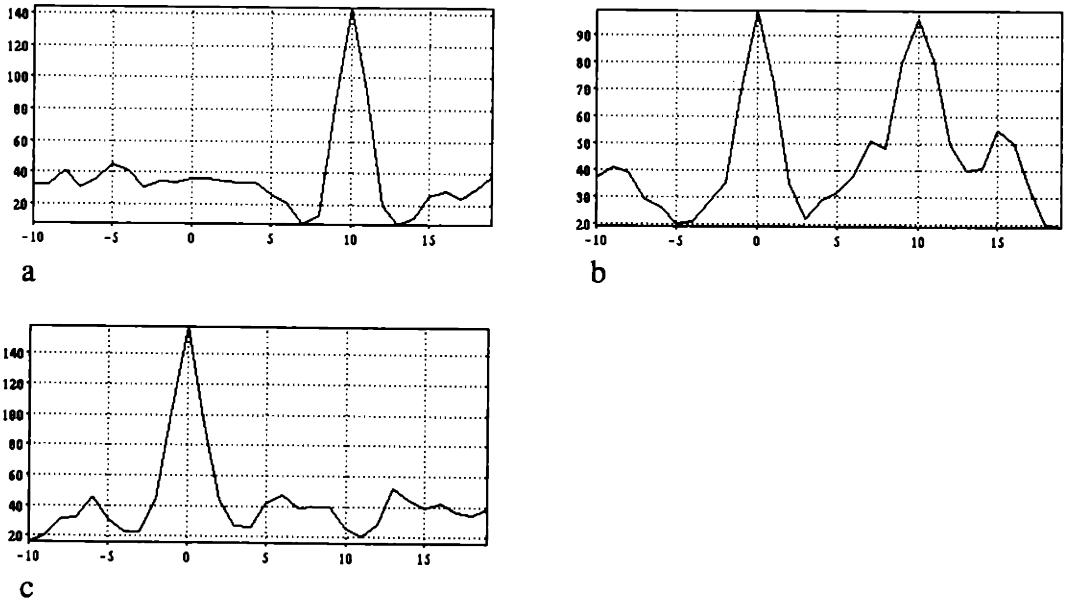


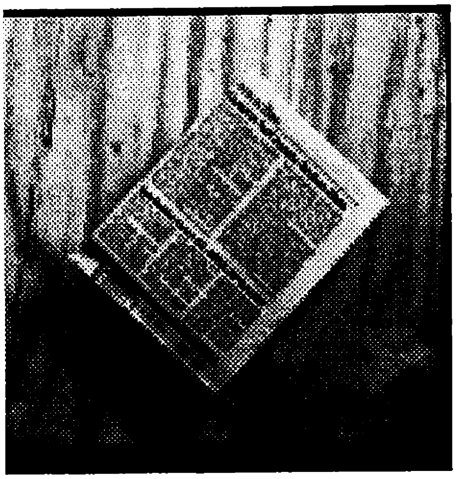
Figure 2.2 Score vectors for different locations in an RDS. (a) (x,y) = (162, 128). (b) (x,y) = (192,128). (c) (x,y) = (222,128). Because the support neighborhood is 23x23, the support for point A is contained entirely within the floating square and the support for point C is contained entirely within the background; the support for point B is split between both regions.

We call point B a *close winner* because the "winning" disparity has a close competitor. The key observation is that such points are likely to be located at depth discontinuities. For all points p in the left image, use the following procedure to determine whether p is a close winner:

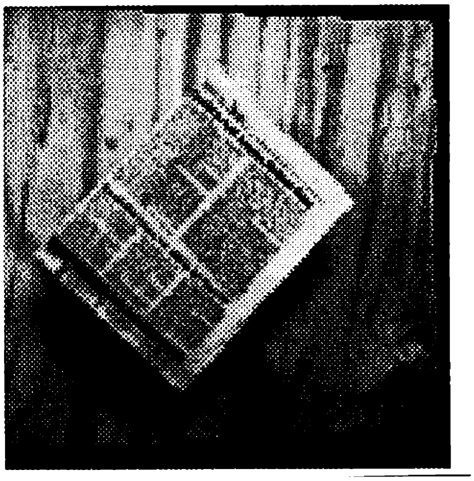
1) Suppose $v'(p) = \{s_{id}, s_{id+1}, \ldots, s_{fd}\}$. Check every s_i, $id < i < fd$, to see whether $s_{i-1} < s_i < s_{i+1}$. If so, then s_i is called a *peak.*[1].
2) If $v'(p)$ has two or more peaks then pick the two largest, say α and β, $\alpha \geq \beta$. Define the margin $m = (\alpha-\beta)/\alpha$. If $m \leq M$ (a parameter that has been set to 0.2 for the results in this thesis) then declare p to be a close winner.

Figure 2.3 shows close winners for several stereo scenes.

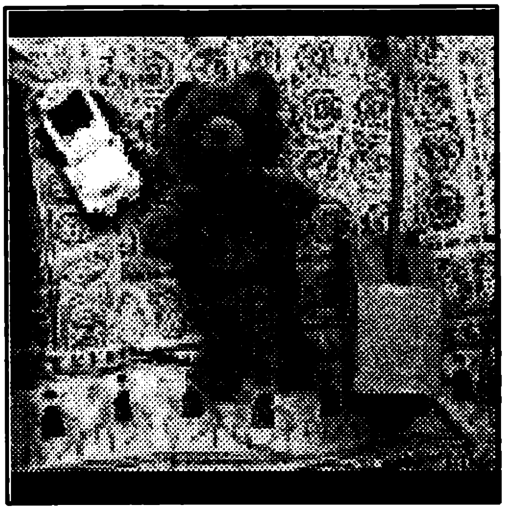
[1] This simple rule won't find a peak in a sequence like 98,103,103,92, although it should. Such boundary cases are improbable and can be omitted without any real loss.



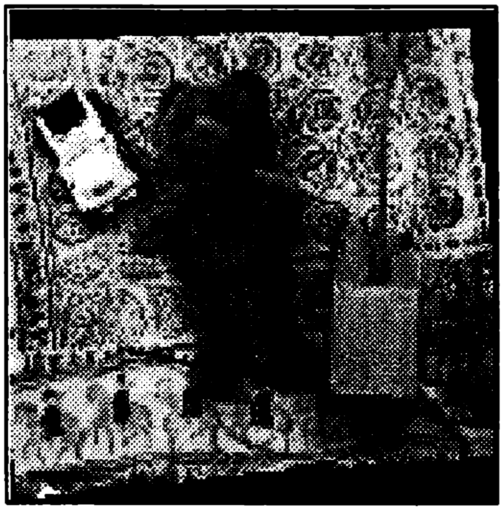
a



b



c



d

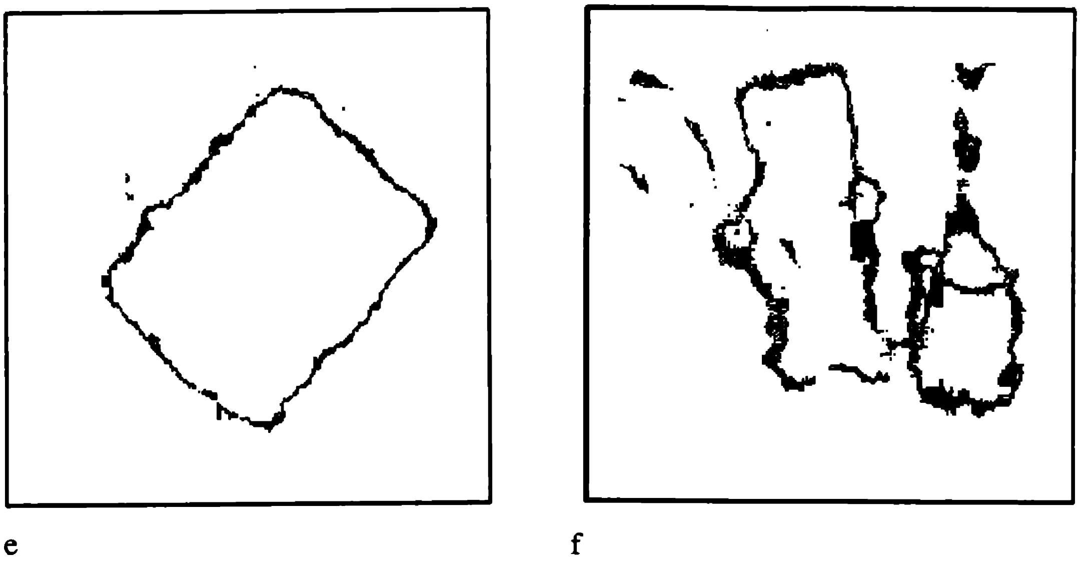
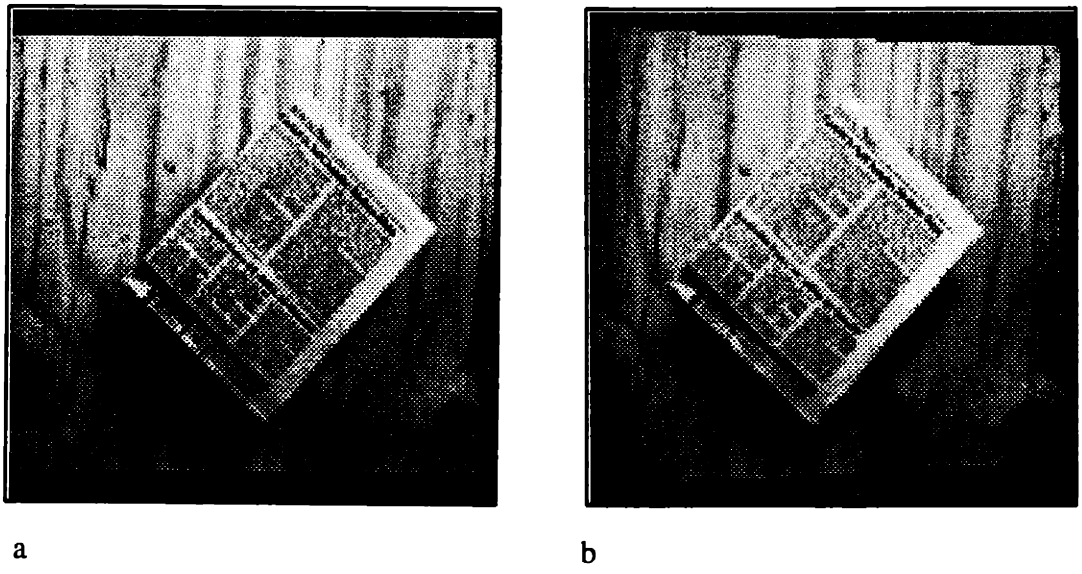


Figure 2.3 Close winners for several stereo scenes. (a) Newspaper on wood: left view. (b) Right view. (e) Close winners. (c) Truck, teddy bear, crane: left view. (d) Right view. (f) Close winners.

A major problem with the close winners technique is localization. Although a bimodal score vector usually indicates a nearby depth discontinuity, it is unclear how to locate the discontinuity precisely. In the vicinity of a discontinuity contour, the set of points with bimodal score vectors typically forms a thin strip in the vicinity of the contour. How do we select the points that lie on the contour? One idea is to pick the points with the smallest margin, which means that the two peaks in the score vector are as equal as possible. Unfortunately, this approach yields the best answer only in the case of a linear contour, which splits the support neighborhood evenly for a point on the contour. If the object boundary is concave, the point with smallest margin will be located outside the object; if the object boundary is convex, the point with smallest margin will be located inside the object. Narrow objects such as a thin bar may be missed entirely, depending on the margin threshold and neighborhood size. In general, a smaller neighborhood size provides better localization but a lower signal-to-noise ratio, a tradeoff similar to that for the smoothing parameter in edge detection. Even given poor localization, close winners can still be helpful in indicating regions that are likely to contain discontinuities; the integration stage of the Vision Machine can make use of this information.

Occlusion also causes problems. Consider the support neighborhood for point D in figure 2.1. The occluded part of the neighborhood has no match in the right view, since it is visible only to the left eye. Therefore edges in the occluded area will vote randomly, adding noise to the score vector. (If the occluded area is wider than the support neighborhood, the boundary will be missed entirely.) Note that for a linear occluding contour, the close winner with the smallest margin is located in the middle of the occluded area. This effect is visible in figure 2.4c, which shows close winners for the newspaper scene superimposed on a silhouette of the newspaper. Close winners for the left boundary are displaced into the occluded region.

With a little further analysis, we can compensate for the effects of occlusion. The stereo algorithm uses the left image as the reference image for the depth map, matching from left to right (an arbitrary implementation decision). When matching from left to right, occlusion degrades the localization of left boundaries (boundaries on the left sides of objects) but not the localization of right boundaries. When matching from right to left, the reverse is true. So, use close winners computed in left-to-right matching to determine right boundaries and close winners computed in right-to-left matching to determine left boundaries. (Left and right boundaries can be distinguished by the gradient direction of the interpolated depth map.) Right-to-left close winners are computed in right-image coordinates, so they must be transformed into left-image coordinates using the depth map. Because the depth map is ambiguous at boundaries, the transformation splits the winners into two separate contours. We resolve the ambiguity by choosing the depth value that corresponds to the occluding contour, using again the gradient direction of the interpolated depth map. Figure 2.4 displays the intermediate and final results.



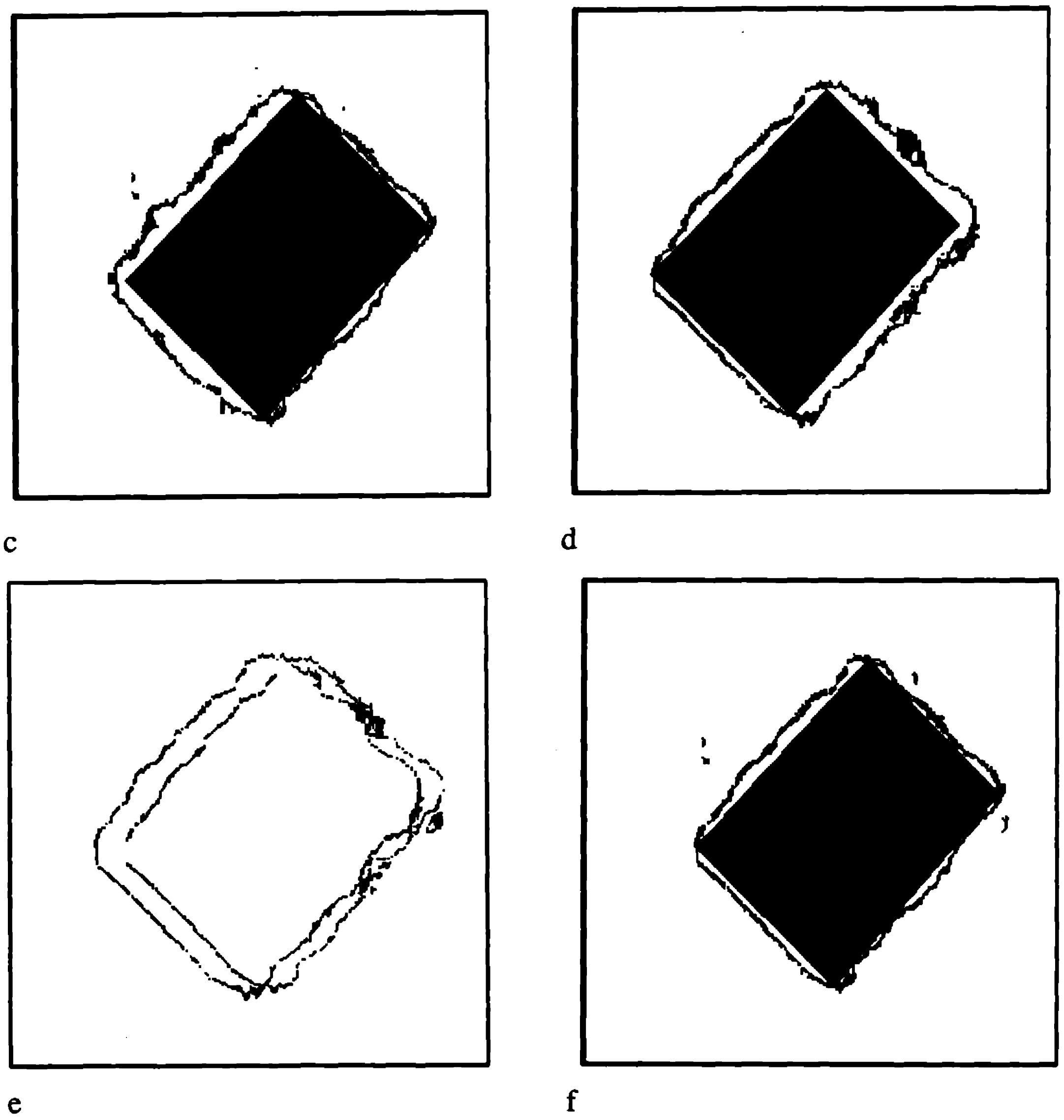


Figure 2.4 Combining left-to-right and right-to-left close winners. (a) Newspaper on wood: left view. (b) Right view. (c) Left-to-right close winners superimposed on newspaper silhouette (left view). (d) Right-to-left close winners (right view). (e) Right-to-left close winners transformed into left image coordinates via the depth map, with ambiguous results. (f) Combined left-to-right-close winners and correctly transformed right-to-left close winners.

2.1.2 Occlusion

When one surface lies in front of another, the foreground surface typically occludes a portion of the background surface (see the discussion of the ordering constraint in section 1.3). The location of the occluded region depends on the viewpoint. Since the boundary on the uphill side of an occluded region is the discontinuity contour, identifying an occluded region leads us directly to the associated depth discontinuity. This technique can be used to locate any depth discontinuity with the exception of extended horizontal boundaries, which are not associated with occlusion.

Our goal is to identify occluded areas. Let us begin by considering only right-occluded areas, that is, areas that are visible from the left but not the right view. By definition such an area does not have a match in the right image. Thus we could look for weak matching scores as an indicator of occlusion. However, weak matching scores can arise from a number of causes, including disparities outside of the fusional range of the algorithm (a problem that we address later in the thesis), so this will not work. A better cue is provided as a side effect of the ordering constraint. Recall that every potential match is surrounded by an hourglass-shaped region extending through the *d* and *x* dimensions, the *forbidden zone* (see [Yuille, Poggio 1984]), as pictured in figure 2.5a.

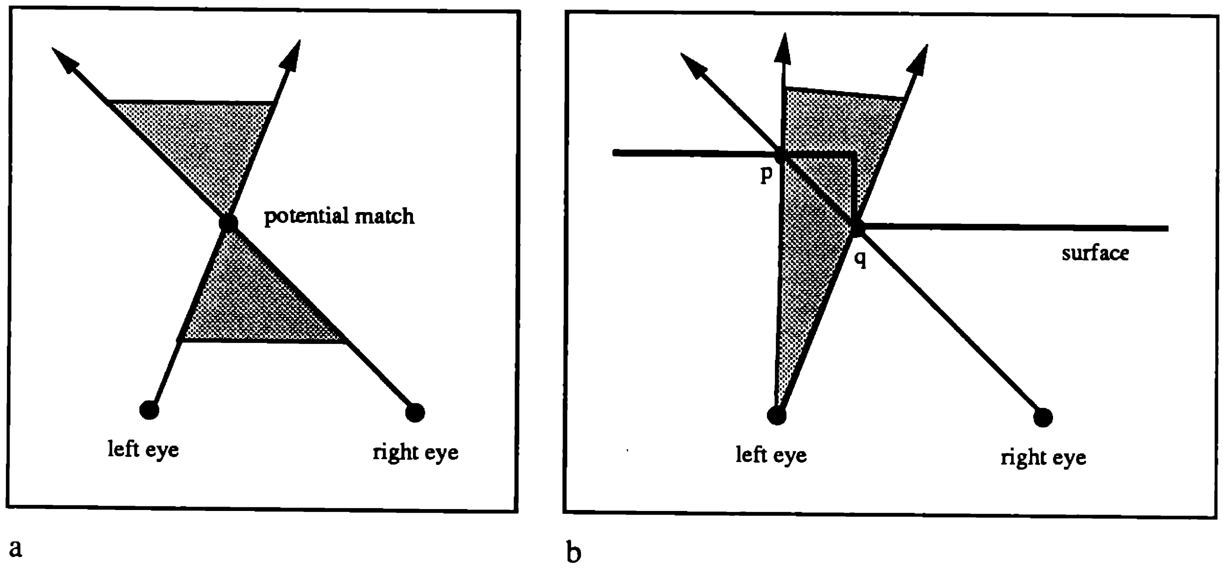
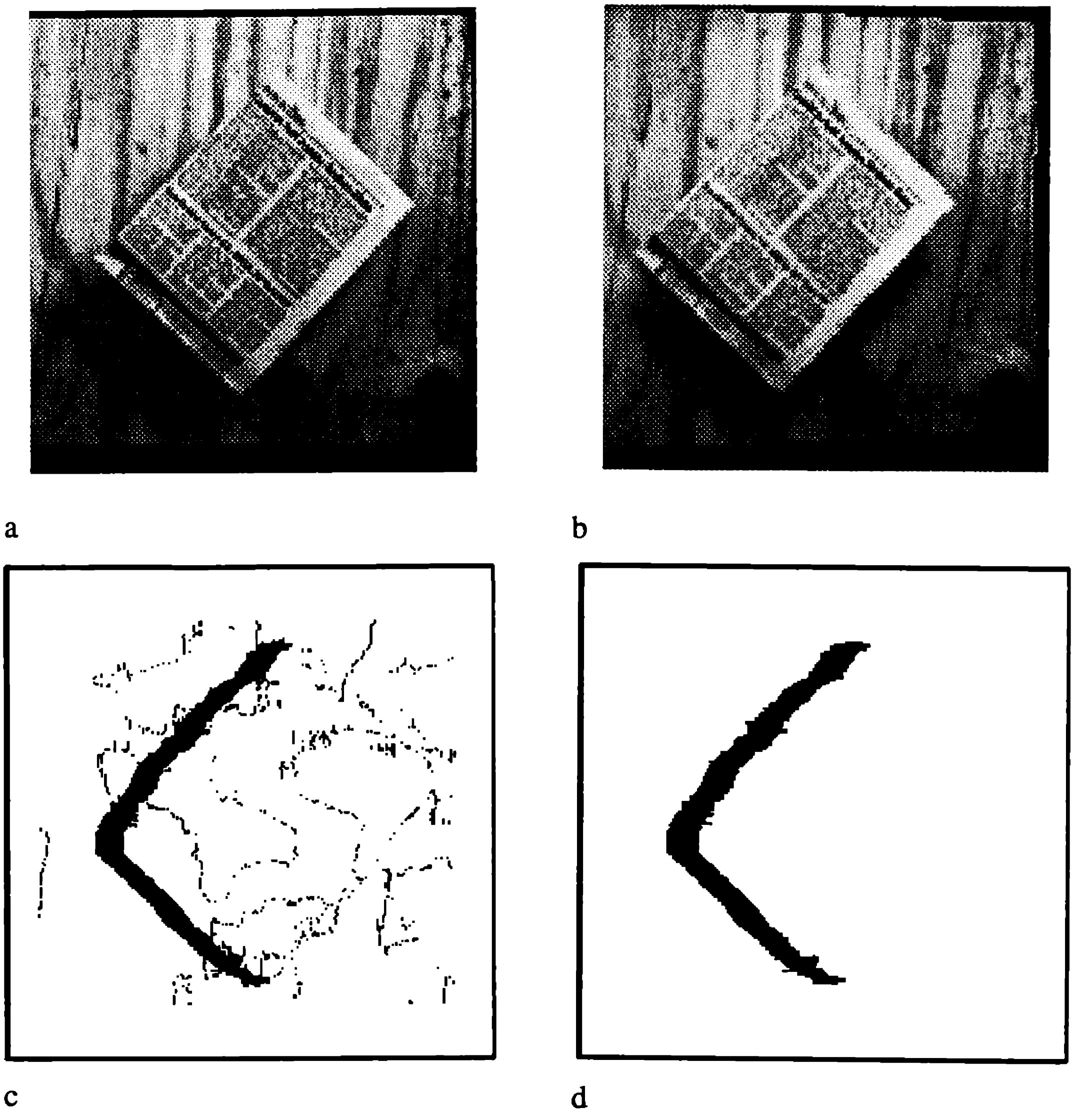


Figure 2.5 (a) The forbidden zone (shaded) for a particular potential match. (b) The shaded region is contained within the union of the forbidden zones for points p and q, showing that no match will be permitted there.



a

b

c

d

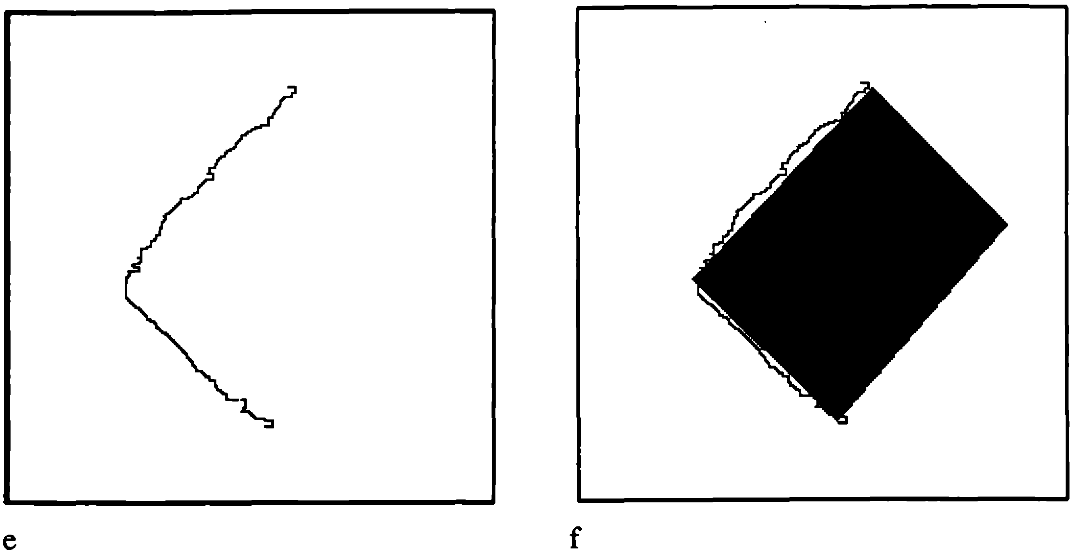
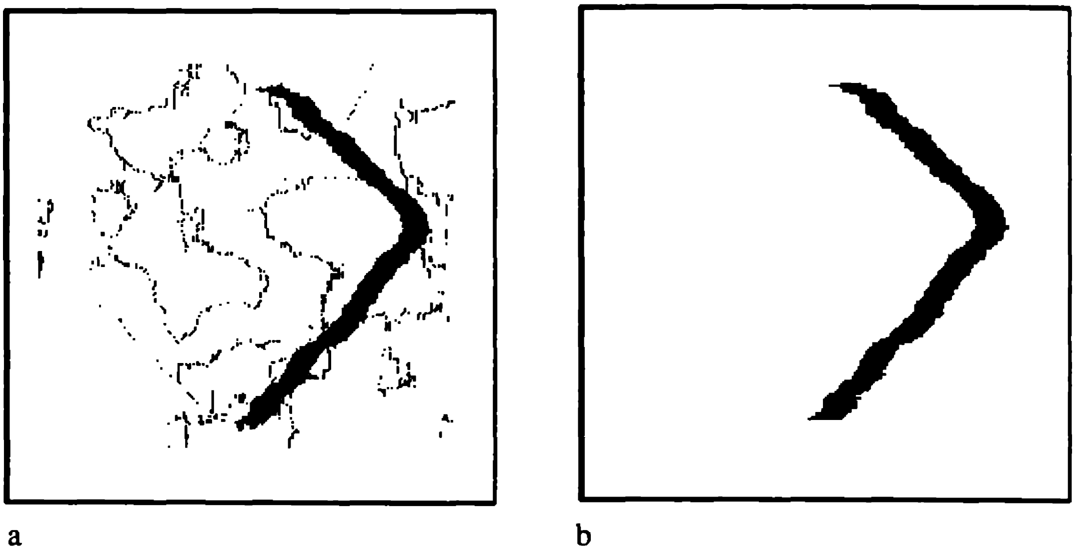


Figure 2.6 Identifying right-occluded regions. (a) Newspaper on wood: left view. (b) Right view. (c) Suppressed points for right-occlusion. (d) Filtered suppressed points. (e) Associated depth discontinuities. (f) Discontinuities superimposed on newspaper silhouette (left view).



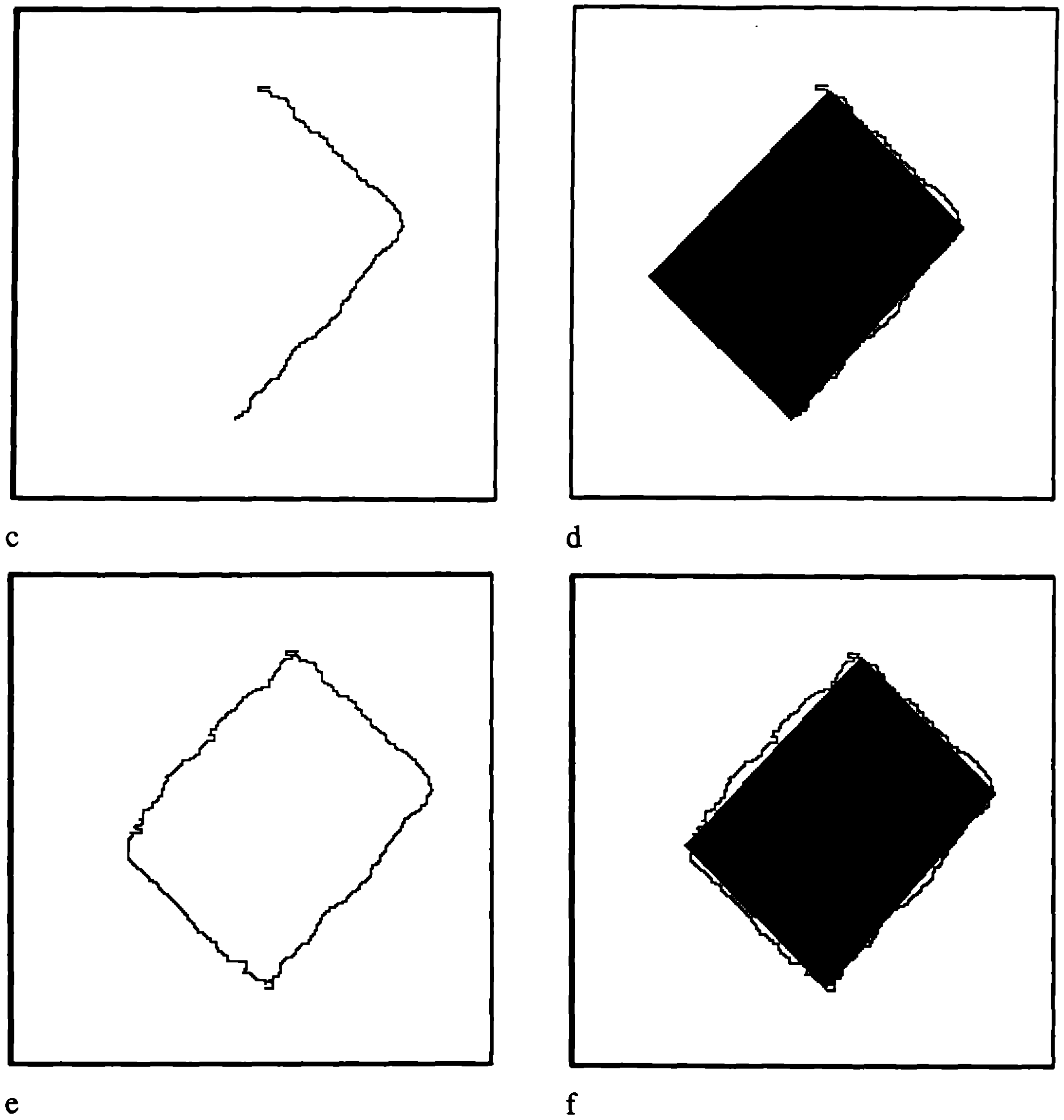
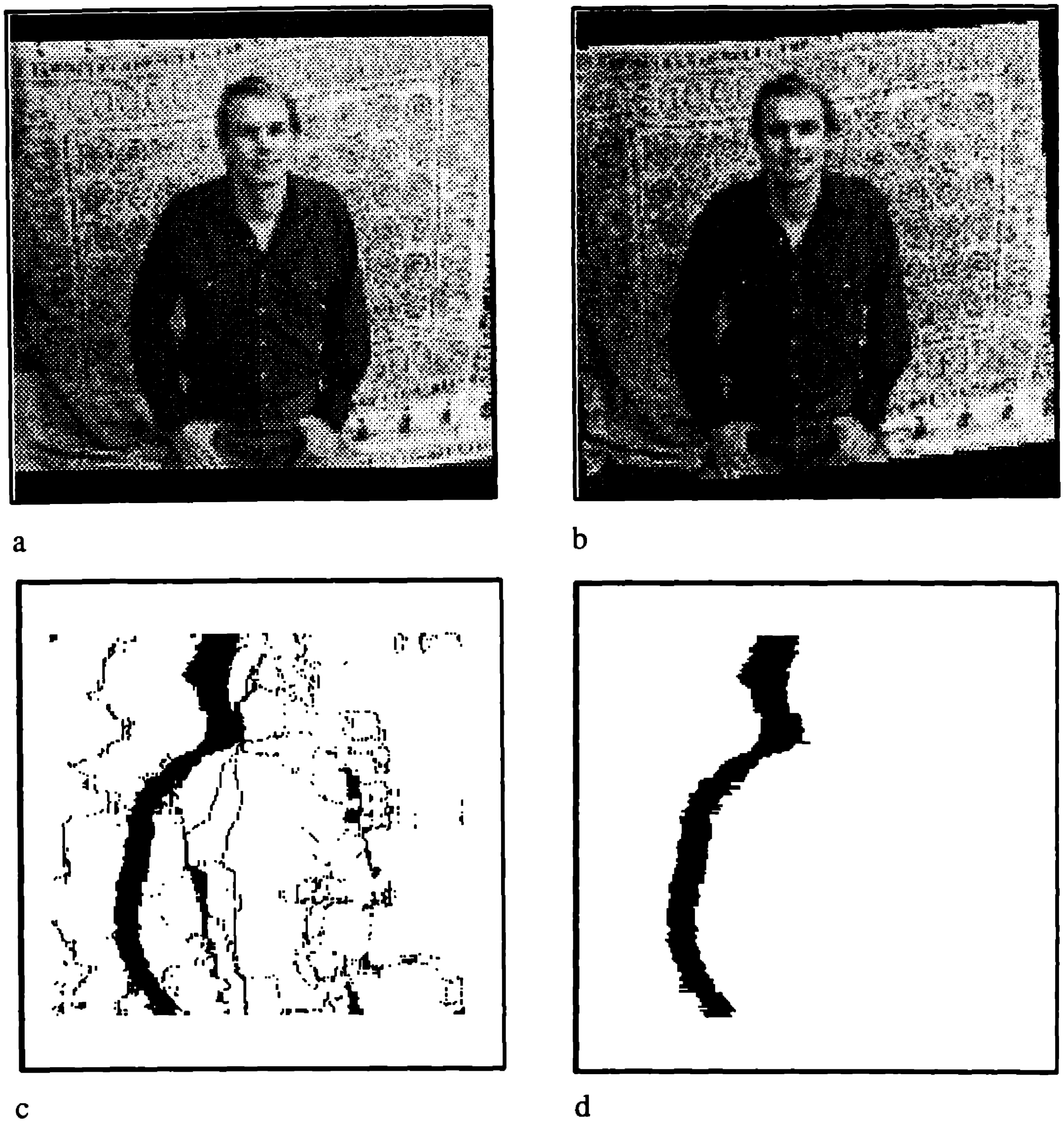


Figure 2.7 Identifying left-occluded regions. (a) Suppressed points for left-occlusion. (b) Filtered suppressed points. (c) Associated depth discontinuities. (d) Discontinuities superimposed on newspaper silhouette (right view). (e) Merged discontinuities from left-occlusion and right occlusion. (f) Merged discontinuities superimposed on newspaper silhouette (left view).


a
b
c
d

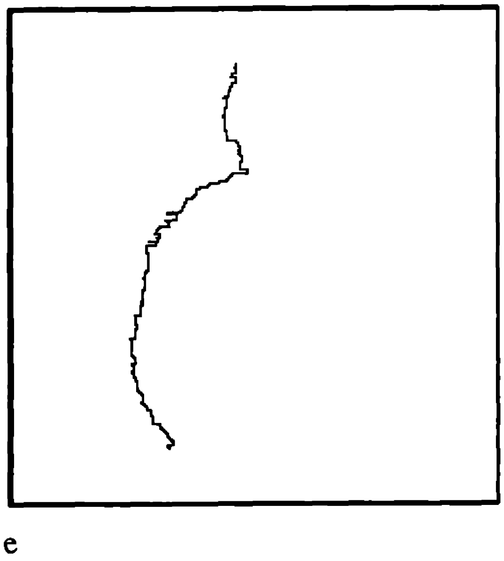
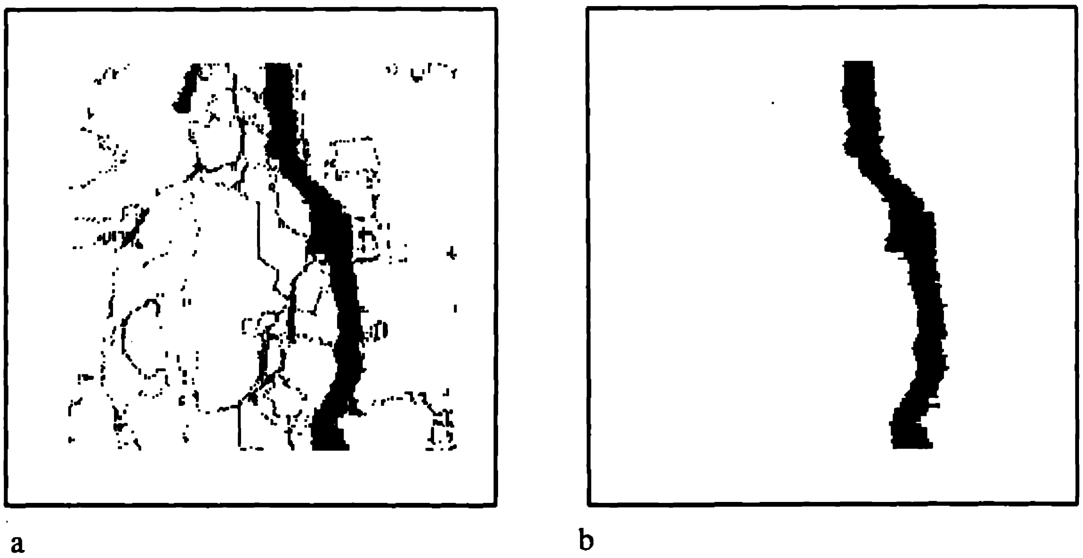


Figure 2.8 Identifying right-occluded regions. (a) Graduate student: left view. (b) Right view. (c) Suppressed points for right-occlusion. (d) Filtered suppressed points. (e) Associated depth discontinuities.



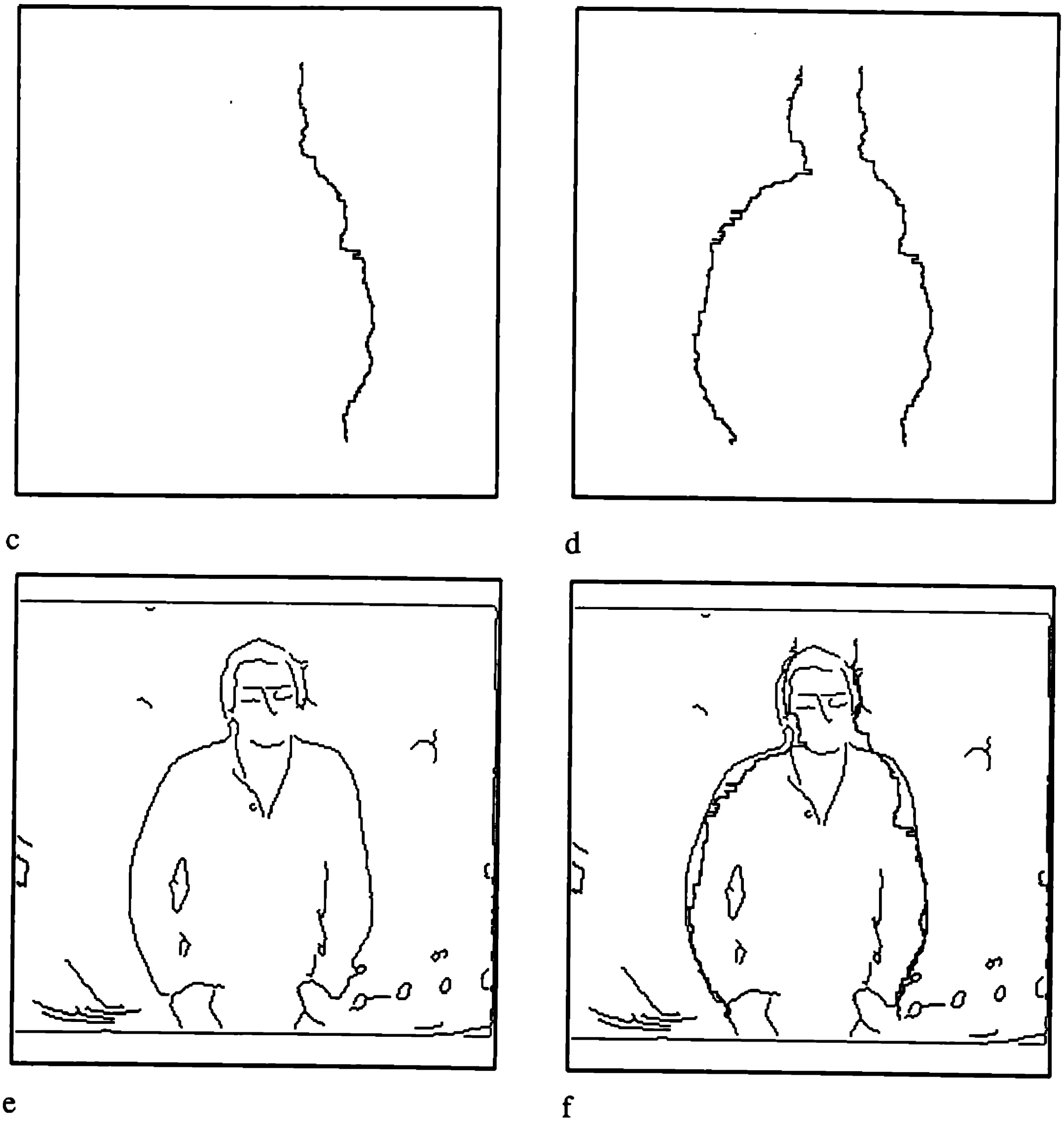
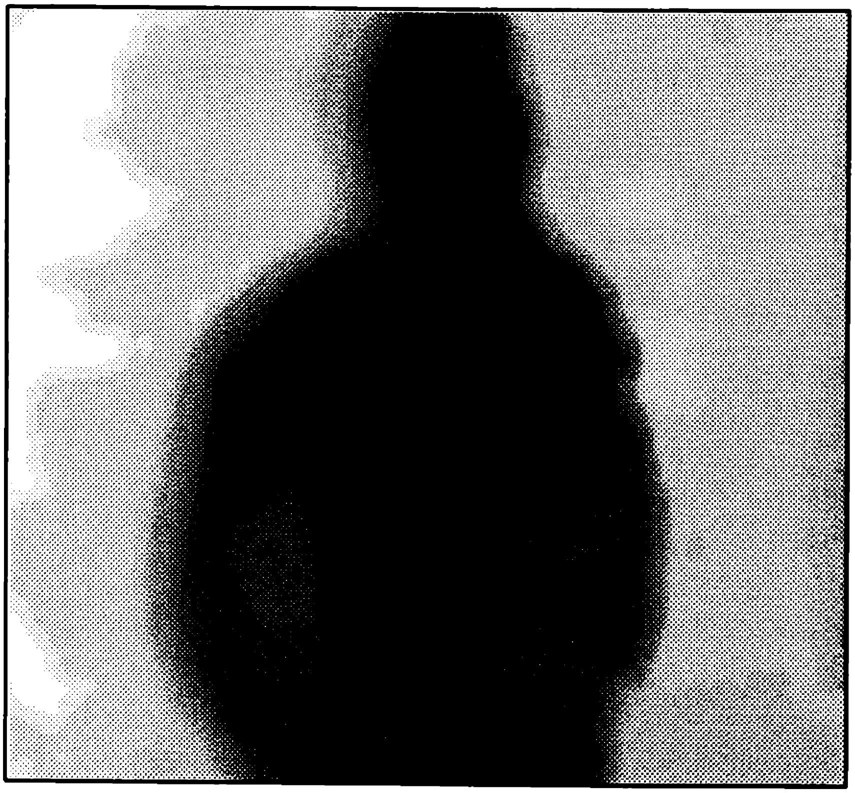


Figure 2.8 (continued) Identifying left-occluded regions. (a) Suppressed points for left-occlusion. (b) Filtered suppressed points. (c) Associated depth discontinuities. (d) Merged discontinuities from left-occlusion and right occlusion. (e) Intensity edges of left view. (f) Merged discontinuities combined with intensity edges to show boundary localization (left view).

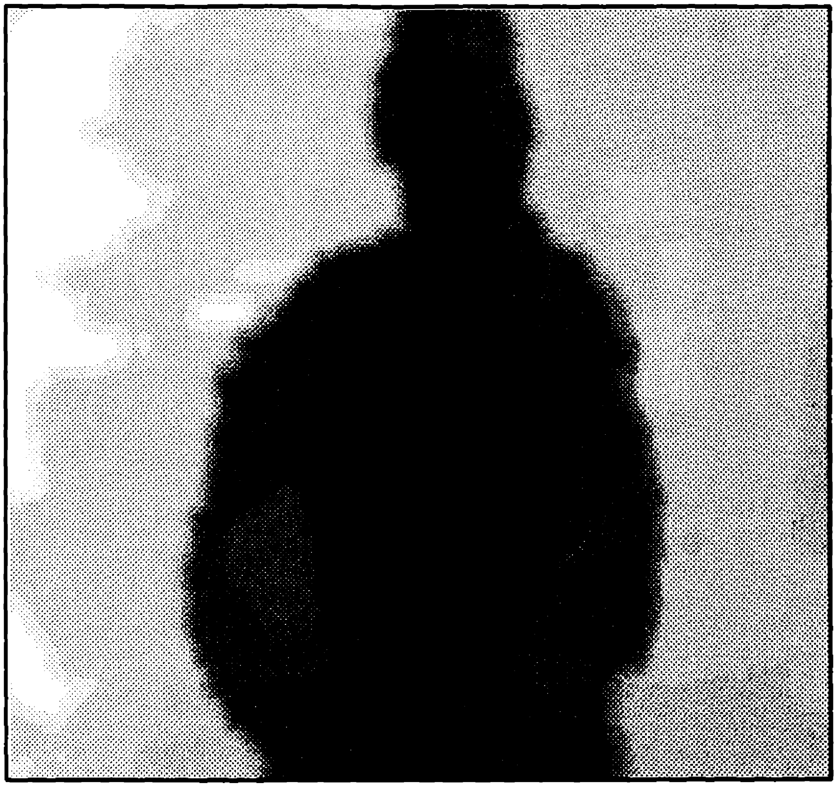
There still remains the problem of detecting left-occluded areas (visible from the right view but not from the left view). We can find left-occluded areas by running the same analysis, but matching the right image to the left image instead of the reverse. For left-occluded areas, the associated depth discontinuities lie on the left-hand (again, uphill) side of the occlusion. These discontinuities have been located in the right image and must be mapped back into the left image. The disparity at a discontinuity is ambiguous (discontinuities are close winners), but the correct disparity is always the uphill value. Given the disparity, we can map right image discontinuities into the left image. Our analysis is depicted in figure

2.7. Figure 2.8 repeats the sequence of figures 2.6 and 2.7 for a different scene. The discontinuity contours that emerge are somewhat convoluted; this could be remedied by, for example, replacing the contours by spline approximations.

Finally, we should point out an additional benefit of identifying occluded areas. For many applications, it is useful to have a complete disparity map with disparity values assigned to every location in the reference image. A standard method for obtaining such a map is to interpolate the disparity values from stereo, filling in gaps. Given knowledge of occlusion, we can do better than naive interpolation. Interpolation blurs discontinuities by filling in occluded areas with stereo data from both sides. A more intelligent approach is to assume that an occluded area is at the same disparity as the background, filling in right-occluded regions with disparity values from left to right. Figure 2.9 compares the results of traditional interpolation with this "knowledge-based" interpolation; notice the sharpness of discontinuities in the second case.



a



b

Figure 2.9 Interpolation of disparity data. (a) Standard "rubber sheet" interpolation. (b) Improved interpolation: the left-hand depth boundary is considerably sharper.

2.2 Improving stereo performance near discontinuities

Most stereo algorithms depend on the continuity constraint, the assumption that depth varies smoothly almost everywhere. It is not surprising that stereo does poorly at object boundaries, where depth is discontinuous. In this section we explore the possibility that knowledge of depth discontinuities can be used by stereo to improve performance near boundaries. In the particular case of the Drumheller-Poggio algorithm, the problem is that support neighborhoods can cross depth boundaries and pick up misleading information from the other side, as shown in figure 2.10.

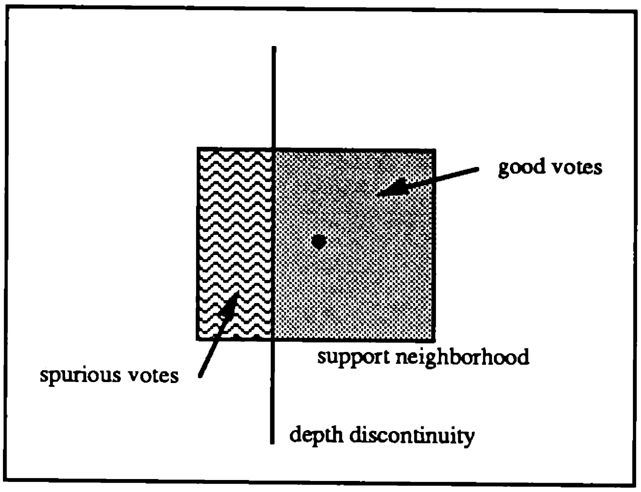


Figure 2.10 The support neighborhood for the black dot includes some spurious votes.

Given knowledge of discontinuity locations, a natural idea on how to improve stereo performance is to reshape support neighborhoods so that they do not cross the discontinuities. In figure 2.10, we would keep the dotted region and throw away the rest. The discontinuities used to bound neighborhoods can be provided by a first pass of the stereo algorithm, using either the close winners or occlusion techniques described earlier, or can be fed back from the integration stage of the Vision Machine. Another possibility is to use intensity edges. However, intensity edges alone are not very informative, since only a few of them coincide with depth boundaries; the role of the integration stage is precisely to select such edges. The problem with reshaping the neighborhoods in this way is that errors in localization of the discontinuities can be catastrophic; the entire remaining support neighborhood may be on the wrong side of the discontinuity.

A possible alternative approach would be to run stereo at a series of scales, coarse to fine, as discussed in section 1.3.1. Specifically, different scales are implemented by using different support neighborhood sizes, a large neighborhood for a coarse scale. Use disparity values from one scale to guide search at the next scale, except near depth boundaries. For a point near a depth boundary, a large neighborhood will cross the boundary, as discussed above. So, at each scale, look for close winners. For those points that are not close winners, use the disparity value obtained to guide search at the next scale, as usual. For those points that are close winners, the disparity value determined at that scale may be completely wrong, so let search at the next scale be unconstrained. The advantage of this approach is that it does not require a priori knowledge of discontinuity locations.

Figure 2.11 below shows an exaggerated example of such an approach. We have run stereo on the RDS of figure 2.1 at two different scales, width = 23 and width = 7. The true disparity of the central floating square is 10. Each picture shows the boundaries of the connected component in the disparity data with value 10. For (a), the large scale, the boundary is clean but is rounded off at the corners. For (b), the small scale, the boundary is noisy but less rounded off at the corners. (c) combines the two, using the small-scale depth map at locations identified as close winners at the large scale. Notice that (c) has the clean plateau at disparity 10 of the large scale while retaining the sharp corners of the small scale. Unfortunately, the signal-to-noise ratio is low at the small scale, so the boundary is rather erratic.

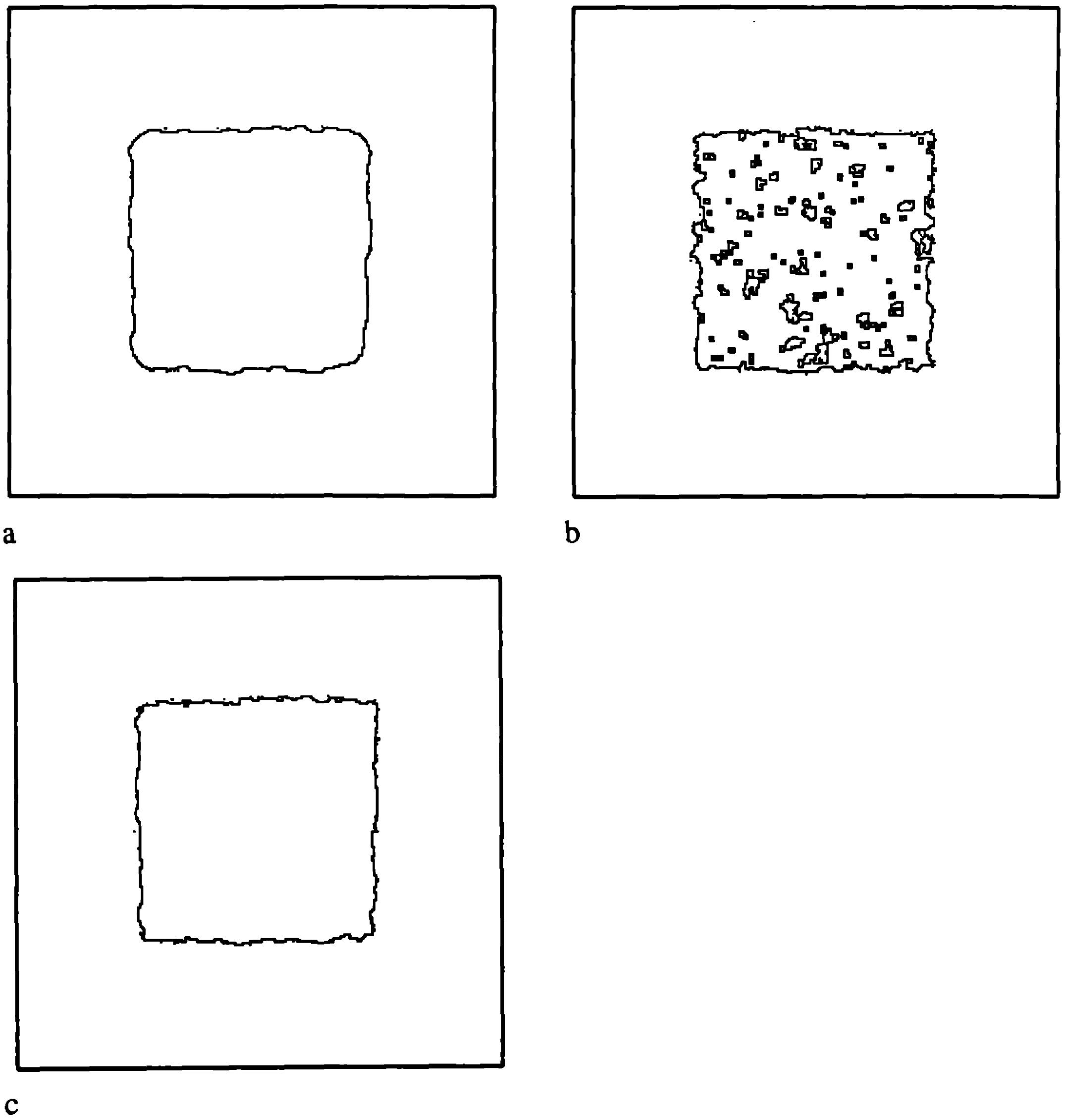


Figure 2.11 Multiple scales: boundary of region with disparity = 10. (a) Large scale - support width = 23. (b) Small scale - support width = 7. (c) Combination of scales.

3 Improvements to the stereo matcher

The original Drumheller-Poggio algorithm matched DOG zero-crossings, where the local support score counted the number of zero-crossings in the left image patch matching edges in the right image patch, at a given disparity. We have improved edge-based matching and developed a new intensity-based matching scheme.

3.1 Edge-based matching

Improvements to edge-based matching described in this section are the use of Canny edges as features, comparing gradient directions, and normalizing the matching scores.

3.1.1 Canny edges as features

The matcher now uses edges derived by Todd Cass's parallel implementation of the Canny edge detector [Canny, 1983; Little et.al., 1987] rather than DOG zero-crossings, for better localization. Generalizing DP in this way is relatively straightforward, requiring only a definition of what it means for two Canny edges to be compatible, that is, to represent a potential match.[1] Recall that two zero-crossings are compatible if the signs of the convolution for each edge (the *edge polarities*) agree. There is no direct analogy to the sign of the convolution for the Canny edge detector, which is not a linear filter. However, the idea is to distinguish whether an edge represents a transition from light to dark or from dark to light. We can define the polarity for a Canny edge to be the sign of the x component of the image intensity gradient.[2] Figure 3.1 shows stereo data obtained using this definition. The new definition has the advantage of numerical stability, since it depends on a first derivative of intensity rather than a third derivative of intensity, and therefore requires less prior smoothing of the image.

[1] Note that generalizing MPG to use Canny edges would be tricky. MPG depends on an estimate of the density of DOG zero-crossings as a function of scale in order to mitigate the false target problem. Such an estimate may be impossible to obtain for the Canny edge detector, which is highly non-linear.

[2] Unless otherwise specified, the phrase "intensity gradient" refers to the gradient of the smoothed intensity image generated by the Canny edge detector.

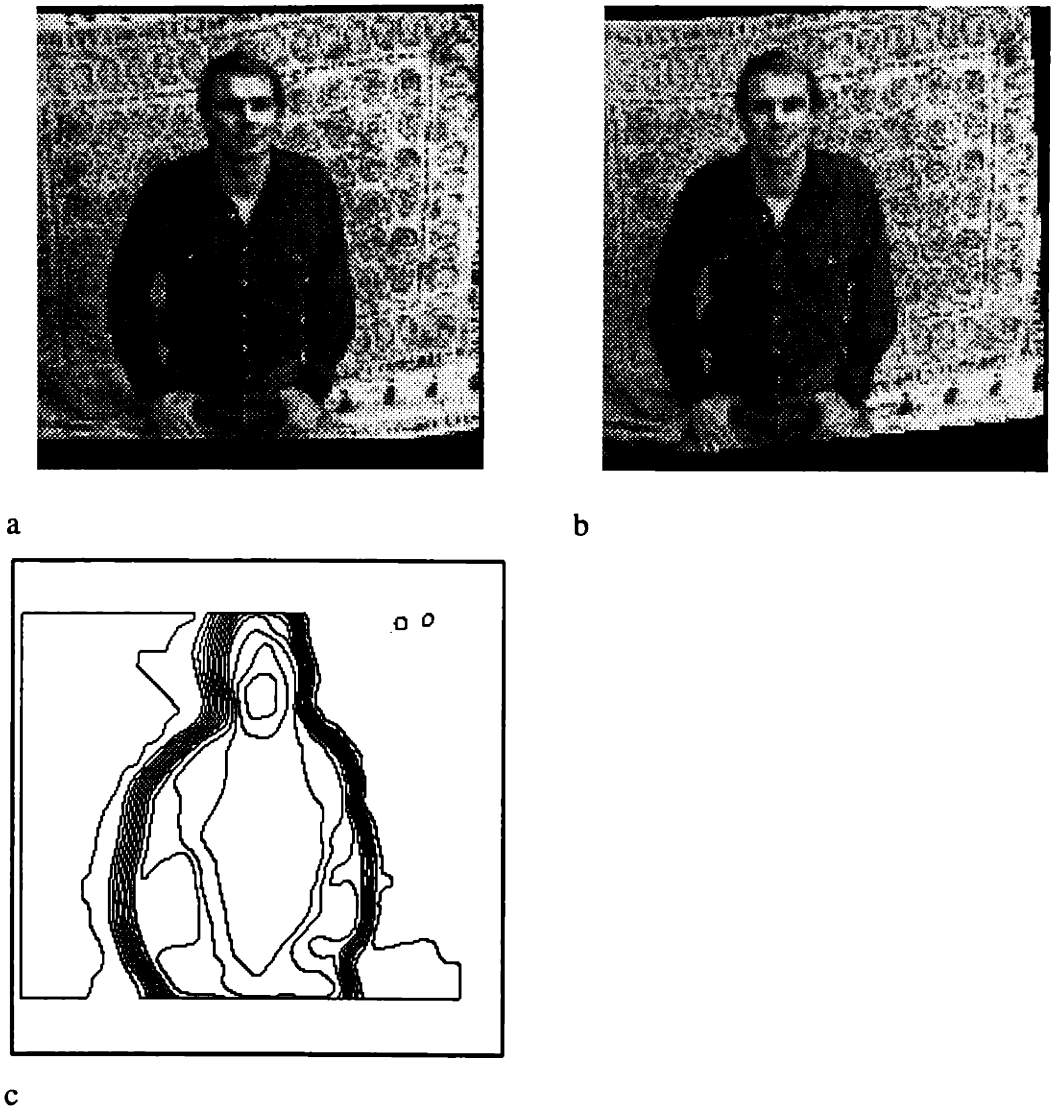
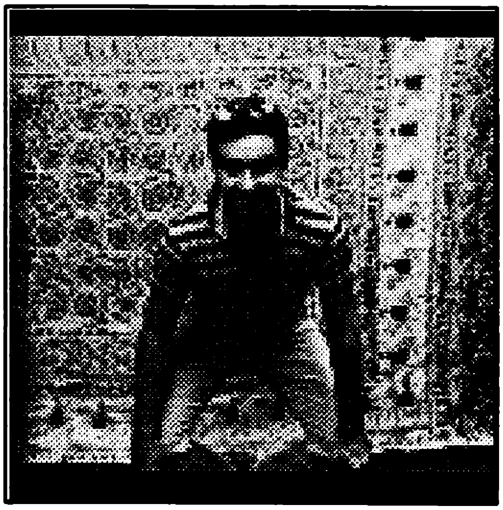


Figure 3.1 Stereo matching using edge polarity. (a) Graduate student, left view. (b) Right view. (c) Isodisparity contours.

3.1.2 Comparing gradient directions

Edge polarity has several defects as a criterion for edge compatibility. First, it is too loose a constraint. There are only two flavors of polarity, plus and minus, so that the probability that two randomly chosen edges will match is $^{1}/_{2}$. Second, polarity is unstable for a horizontal edge segment, which may be imaged with different polarities in the left and right views. [Grimson 1985] argues (p.50), "Horizontally oriented segments of the zero-crossing contours may be ignored, since they do not have a well defined disparity." While it is true that the disparity of a horizontally oriented edge is ambiguous if one examines only the matches for that particular edge, in the DP case information from other edges in the support neighborhood will often resolve the ambiguity.

We have adopted a more powerful constraint on gradient direction, allowing two Canny edges to match only if the associated intensity gradient directions are aligned within a parameterized tolerance; the default is 30°. (This is directly analogous to the restriction in MPG that two zero-crossings can match only if the gradient directions of the DOG convolution output are approximately equal.) With a tolerance of 30°, the chance that two randomly selected edges match is a more acceptable 1/12. Furthermore, with the gradient direction constraint there is no instability for edges at any orientation. Figure 3.2 shows an example in which matching based on edge polarity produces results inferior to matching based on the gradient constraint, because of horizontal edges at the top of the scene.



a

b

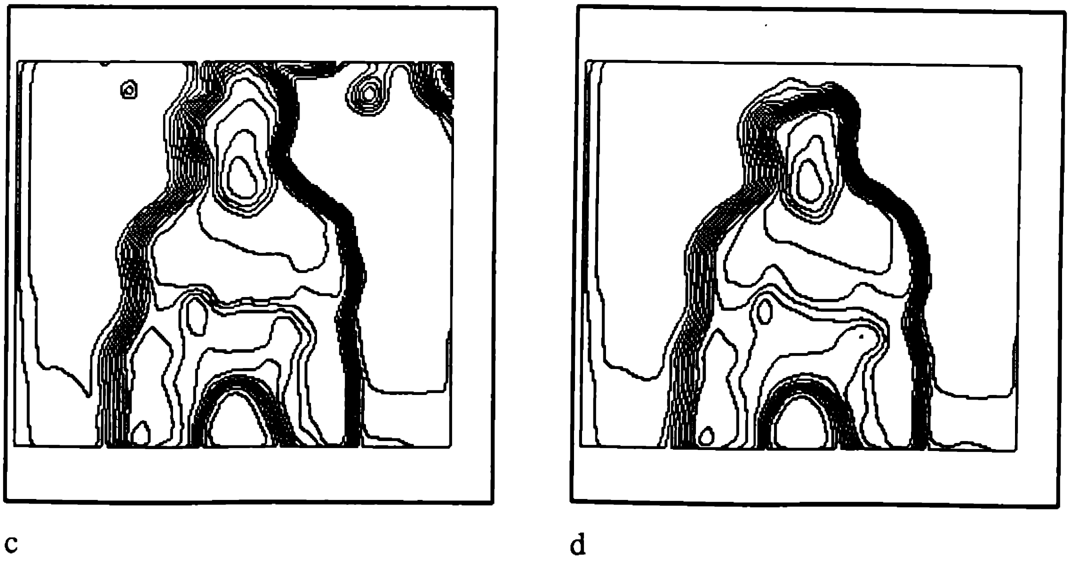


Figure 3.2 Comparison of edge polarity and the gradient constraint. (a) Vision researcher, left view. (b) Right view. (c) Edge polarity-based disparity contours. (d) Gradient constraint-based disparity contours.

The gradient direction constraint yields another benefit. One thorny element of Canny edge detection is noise estimation, estimating a threshold below which gradient values are discarded. Existing noise estimators are not entirely satisfactory. Since the probability that two randomly selected edges will match is low under the gradient direction constraint, we have chosen to eliminate the noise estimation step, setting a Canny gradient threshold of zero. We can more easily afford to add some noisy edges than to toss out some useful edges: "any news is good news". Skipping the noise estimation and hysteresis steps of the Canny detector also reduces computation time.

3.1.3 Normalizing scores

The DP matching score for two image patches is computed by aligning the two patches and counting potential matches. This formula does not take into account the number of edges in each image patch available for matching, so patches with high edge densities can in principle generate artificially high scores. We have experimented with normalized matching scores, using the following equation:

$$s(x,y,d) = \frac{\sum_{A} p(x,y,d)}{\left(\sum_{A} e(x,y)\right)\left(\sum_{B} e'(x+d,y)\right)} \tag{3.1}$$

This is just the original matching score (sum of potential matches) divided by the product of the number of edges in the left image and the number of edges in the right image. When either of the factors in the denominator is zero, define s(x,y,d) = 0.

Terms used in (3.1) have the following definitions:

- A is the support neighborhood around the point (x,y) in the left image.
- B is the corresponding neighborhood around (x+d,y) in the right image.
- p(x,y,d) = 1 if there is a potential match at (x,y,d) and 0 otherwise.
- e(x,y) = 1 if there is an edge at (x,y) in the left image and 0 otherwise.
- e'(x,y) = 1 if there is an edge at (x,y) in the right image and 0 otherwise.

Figure 3.3 compares stereo results obtained with normalized and non-normalized matching scores. For the examples that we have tried, normalized scores do not appear to make much of a difference. With no noise thresholding, edge density is often quite uniform throughout the images. Furthermore, there is considerable neighborhood overlap; we typically use 23x23 neighborhoods and a disparity range of 20. However, we have experimented with both non-uniform edge densities and larger (40 pixel) disparity ranges, without noting any significant differences.

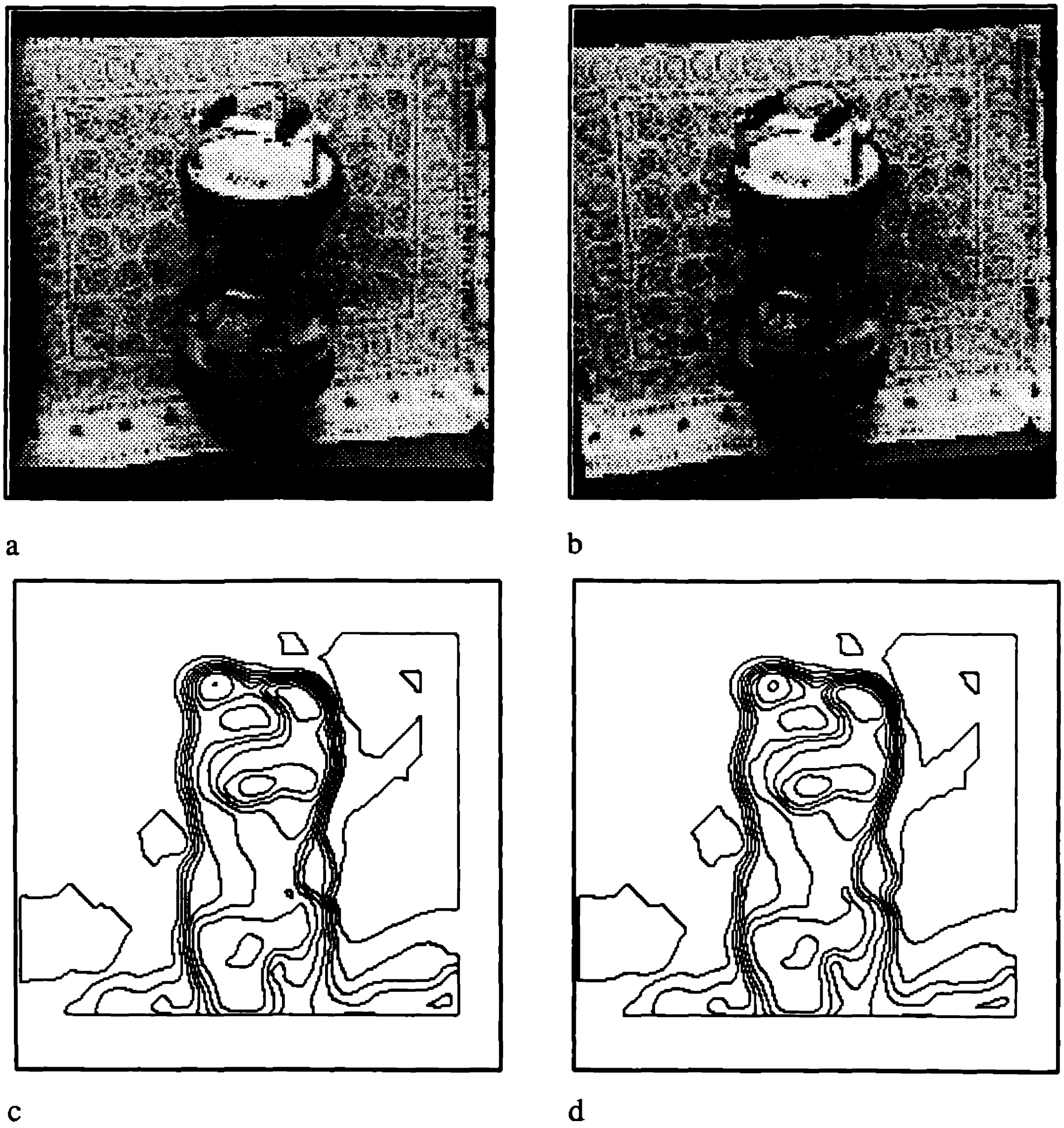


Figure 3.3 Comparison of normalized and non-normalized matching scores. (a) Robot Allen, left view. (b) Right view. (c) Disparity contours for non-normalized case. (d) Disparity contours for normalized case: almost identical.

3.2 Intensity-based matching

3.2.1 Physical invariance

There has been a constant debate within the vision community for some time over the use of edge-based methods versus direct grey-level methods. David Marr asserted ([Marr 1982], p.105), "...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 grey-level intensity arrays, precisely because a pixel corresponds only implicitly and not explicitly to a location on a visible surface." However, edges themselves are not physical invariants (p.106): "The edges of a uniform cylindrical lamppost give rise to perfectly good edges in the images seen by the left and right eyes, but these edges correspond to different lines on the physical surface." Marr concluded that for a biological system, the extra speed gained by using early representations such as grey-levels and edges might in some cases be worth the possible error.

Requiring visual processes to operate on physically-localized primitives creates a bootstrapping problem, because features that are physically localized are not "primitive"; considerable processing is required to derive them. The Vision Machine solution to this conundrum is to assume that early vision modules operate on features that are not guaranteed to be localized. The outputs of these modules are noisy and error-prone, but by combining them the integration stage can obtain physically-localized results superior to the output of any individual module. For example, highlights (specular reflections) give rise to intensity peaks and edges that are not associated with fixed surface locations, causing problems for stereo. By combining stereo with information on reflectance and light source direction, we could either blank out disparity data at highlights or perhaps correct the data. There is also the possibility for feedback from the integration stage to earlier modules, as discussed in section 3.2.

3.2.2 Grey-level correlation

To match grey levels rather than edges, we compute a support score that is the sum of the absolute values of pixel-wise intensity differences between the left and right images.[1] Specifically,

$$s(x,y,d) = \sum_{R} \left| g(x,y) - g'(x,y,d) \right| \qquad (3.2)$$

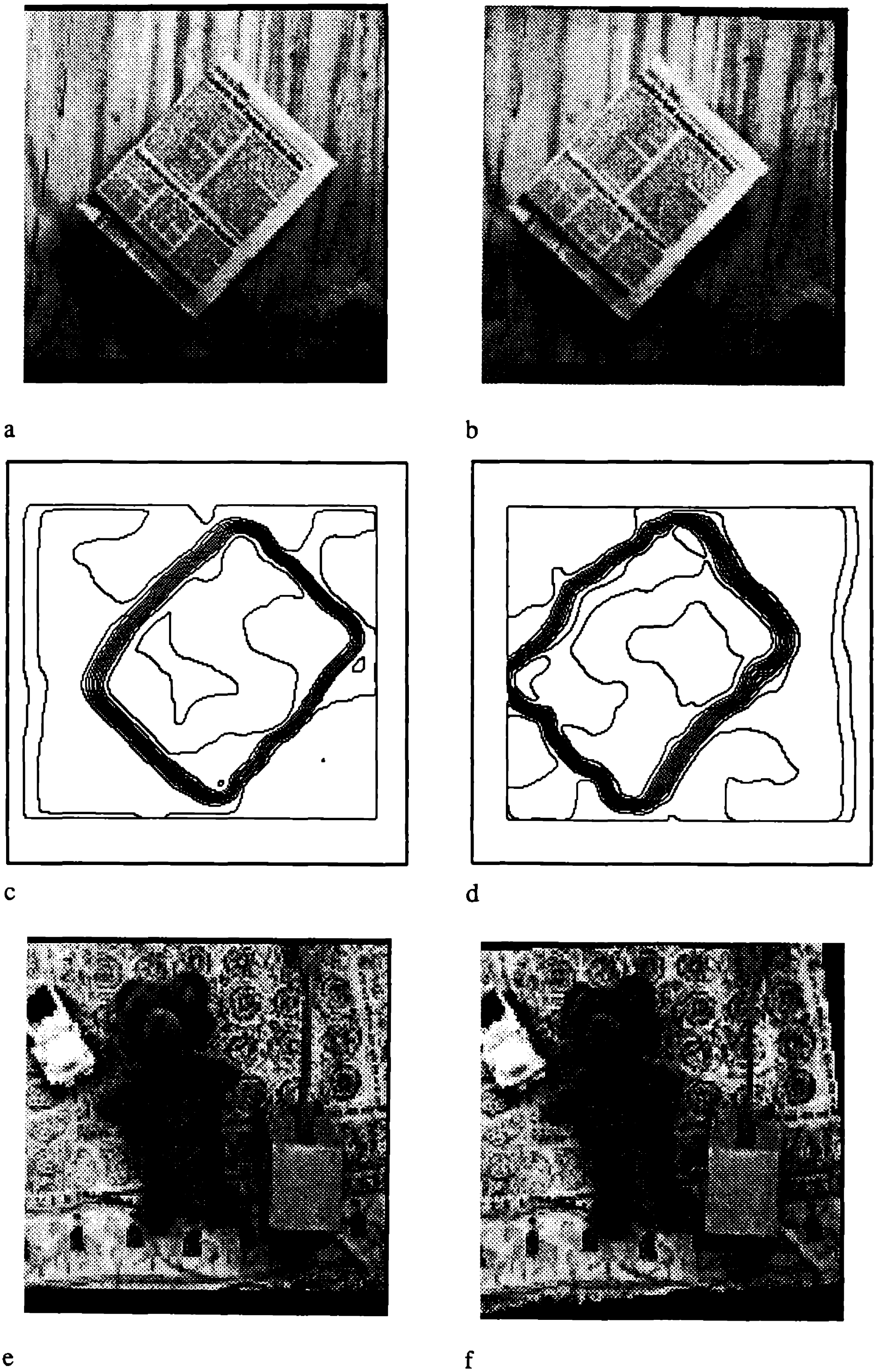
where

- R is the support neighborhood in the left image around (x,y).
- g(x,y) is the left image intensity.
- g'(x,y) is the right image intensity.

[Little, Bulthoff, Poggio 1987] employ a related scheme to compute optical flow. [Gennert 1987] also developed intensity-based stereo, accounting explicitly for foreshortening, among other factors. The above equation does not handle foreshortening (see the discussion in section 1.3.1) since we assume a constrained viewing situation.

Having obtained support scores, subsequent processing is the same as edge-based matching. The sense of intensity-based scores is different from edge-based scores: the larger the scores, the worse the match. In order to be able to process both types of scores in the same way, we invert the intensity-based scores by subtracting all scores from the maximum possible score. Figure 3.4 compares edge-based and intensity-based stereo for several examples.

[1] We use the sum of absolute differences rather than the sum of squared differences because the former has a lower dynamic range, requiring less memory (memory in the CM-1 model of the Connection Machine is severely limited).


a
b
c
d
e
f

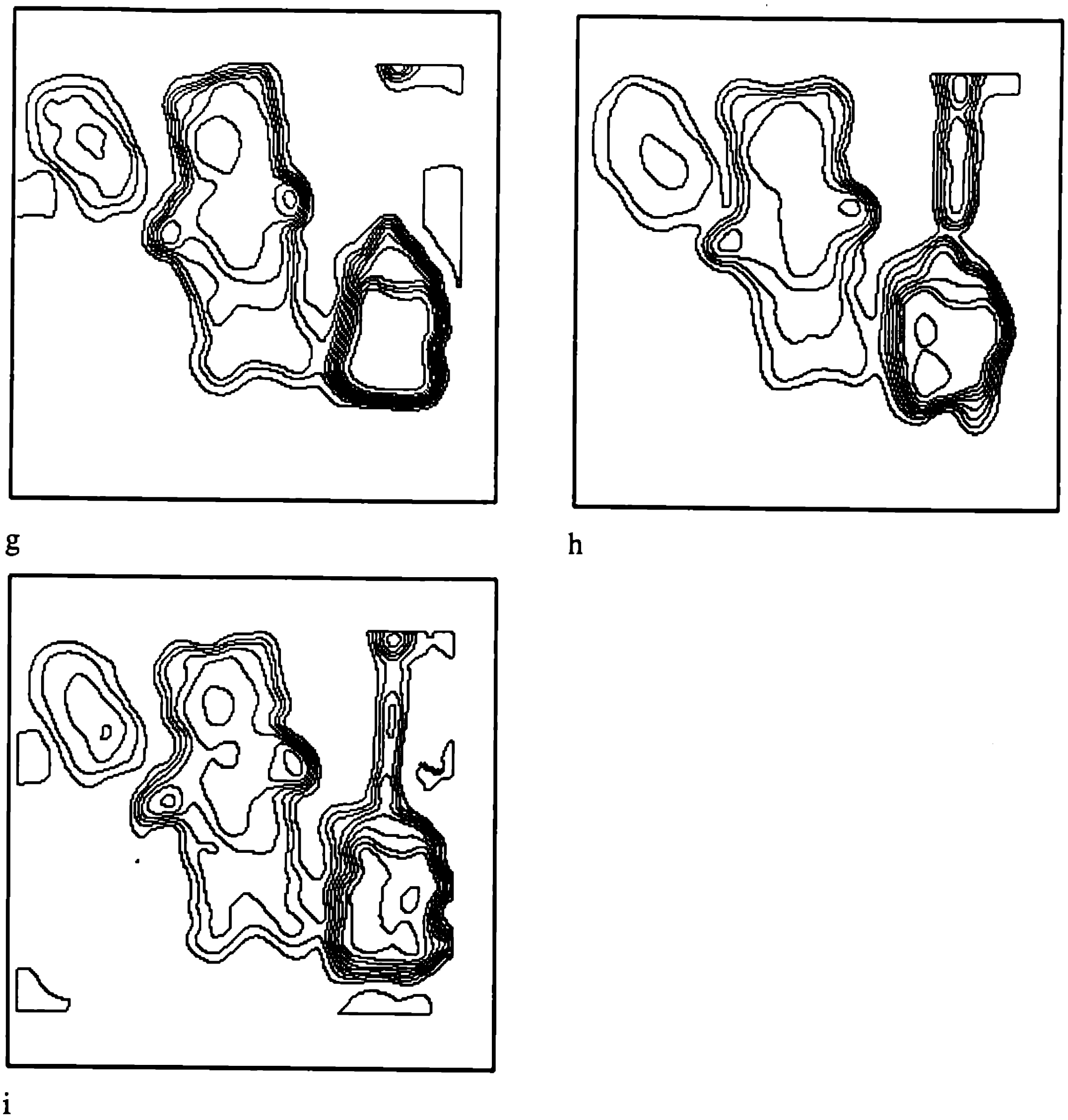


Figure 3.4 Comparing edge-based and intensity-based stereo. (a) Newspaper on wood - left view. (b) Right view. (c) Edge-based disparity data for newspaper scene. (d) Intensity-based data for newspaper scene. (e) Truck, teddy bear, crane - left view. (f) Right view. (g) Edge-based disparity data for teddy bear scene, support width = 23 (standard value). (h) Intensity-based stereo data for teddy bear scene. (i) Edge-based disparity data for teddy bear scene, support width = 17.

Given the simplicity of this algorithm, it works surprisingly well. However, the localization of depth boundaries is clearly inferior to the edge-based scheme. Notice one pleasant surprise: in (h) intensity-based stereo sees the arm of the crane, which edge-based stereo misses almost completely. By reducing the support width from 23 to 17 (i), edge-based stereo is able to capture the arm of the crane as well at the cost of some noise in the stereo data.

3.2.3 Relation to edge-based matching

Intensity-based matching and edge-based matching are closely related by the following theorem (see [Poggio and staff, 1988]:

If f(x,y) and g(x,y) are zero mean jointly normal processes, their cross-correlation is determined fully by the correlation of the sign of f and the sign of g (and determine it).

Hence cross-correlation of the sign bit is equivalent to cross-correlation of the signal itself, under the assumptions of the theorem. The sign bit has the same information as the zero-crossings of the signal. This theorem may have limited practical importance because there is no guarantee that the hypotheses of the theorem will hold in for a given stereo pair of natural images.

It is worth noting that given a spatial representation of the image, intensity-based matching is not much slower on the Connection Machine than edge-based matching, since in each case one processor is assigned to each pixel.

4 Identifying areas outside of the fusional range

The stereo algorithm searches a limited disparity range, selected manually. Every potential match in the scene is assigned the in-range disparity with the highest score, even though the correct disparity may be out of range. How can we tell when an area of the scene is out of range?

The most effective approach that we have attempted to date is to look for region with low matching scores. Two patches that are incorrectly matched will, in general, produce a low matching score. Note that it is essential to normalize the matching scores to eliminate the effect of edge density on the scores. It is also necessary for the edge detector to do a good job of noise estimation, because noisy edges correlate badly, resulting in low scores. In principle, when applying the gradient constraint with a tolerance of 30 degrees, if two randomly generated images are matched the non-normalized score will be spq/12, where s is the size of the support region in pixels and p and q are the edge densities of the left and right image patches. Since the normalized scores remove the effects of edge density, we are left with the uniform threshold of s/12.

In practice, the scores are noisy enough that a simple thresholding is not enough to do the job. Figure 3.5 shows the results of thresholding the newspaper scene, where the disparity range has intentionally been set so that the newspaper is out of range. Notice that small parts of the newspaper appear mistakenly to be in range.

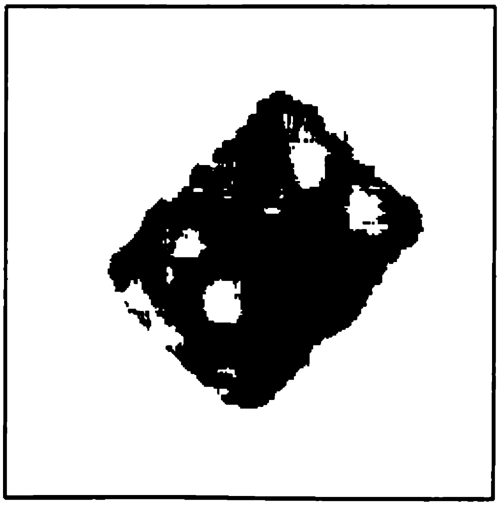


Figure 3.5 Thresholded scores from newspaper scene.

5 Conclusion

This thesis has addressed a number of topics in stereo, including the detection of discontinuities and improved matching techniques, within the context of efficient, parallel implementation. Despite considerable research effort expended over the past twenty years or so in the area of stereo vision, many important questions remain open. Examples are detecting and compensating for specularities; dealing with periodicity in the scene; and rectifying the images automatically to eliminate perspective distortion.

6 References

Bertero, M., T. Poggio and V. Torre. "Ill-Posed Problems in Early Vision," *Artificial Intelligence Laboratory Memo 924*, Massachusetts Institute of Technology, Cambridge, MA, 1986.

Barrow, H.G. and J. M. Tennenbaum. "Computational Vision," *Proc IEEE*, **69**, 5,May 1981, 572-595

Canny, J.F. "Finding Edges and Lines," *Artificial Intelligence Laboratory Technical Report 720*, Massachusetts Institute of Technology, Cambridge, MA, 1983.

Drumheller, M. and T. Poggio. "On Parallel Stereo," *Proc. Intl. Conf. on Robotics and Automation*, IEEE, 1986.

Gamble, E.B. and T. Poggio. "Visual Integration and Detection of Discontinuities: the Key Role of Intensity Edges," *Artificial Intelligence Laboratory Memo 970*, Massachusetts Institute of Technology, Cambridge, MA, 1987.

Grimson, W.E.L. "A computer implementation of a theory of human stereo vision," *Artificial Intelligence Laboratory Memo 565*, Massachusetts Institute of Technology, Cambridge, MA, 1980.

Grimson, W.E.L. **From Images to Surfaces**, The MIT Press, Cambridge, MA, 1981.

Hillis, D. "The Connection Machine," Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1985.

Horn, B.K.P. **Robot Vision**, The MIT Press, Cambridge, MA, 1986.

Little, J., G.E. Blelloch, and T. Cass. "Parallel Algorithms for Computer Vision on the Connection Machine," *Proc. Intl. Conf. on Computer Vision*, 587-591, Los Angeles, 1987.

Little, J., Bulthoff, H., and T. Poggio. "Parallel Optical Flow Computation," *Proc Image Understanding Workshop*, 915-920, L. Bauman (ed.), Los Angeles, CA 1987

Marr, D. **Vision**, Freeman, San Francisco, 1982.

Marr, D. and E. Hildreth. "Theory of Edge Detection," *Proc. Roy. Soc. Lond. B*, **207**, 187-217, 1980.

Marr, D. and T. Poggio. "Cooperative Computation of Stereo Disparity," Science, **194**, 283-287, 1976.

Marr, D. and T. Poggio. "A Computational Theory of Human Stereo Vision," *Proc. Roy. Soc. Lond. B*, **204**, 301-328, 1979.

Poggio and staff. "The MIT Vision Machine," *Proc Image Understanding Workshop*, L. Bauman (ed.), SAI Corp., McLean, VA 1988

Spoerri, A. and S. Ullman. "The Early Detection of Motion Boundaries," *Proc. First Intl. Conf. on Computer Vision*, 209-218, London, 1987

Voorhees, H.L. and T. Poggio. "Detecting Textons and Texture Boundaries in Natural Images," *Proc. First Intl. Conf. on Computer Vision*, 250-258, London, 1987

Yuille, A.L. and T. Poggio. "A generalized ordering constraint for stereo correspondence," *Artificial Intelligence Laboratory Memo 777*, Massachusetts Institute of Technology, Cambridge, MA, 1984