Direct Evidence for Occlusion in Stereo and Motion

by

Jim J. Little and Walter E. Gillett

Technical Report 90-05
November 1990

Direct evidence for occlusion in stereo and motion

James J Little and Walter E Gillett*

Discontinuities of surface properties are the most important locations in a scene; they are crucial for segmentation because they often coincide with object boundaries[1]. Standard approaches to discontinuity detection decouple detection of disparity discontinuities from disparity computation. We have developed techniques for locating disparity discontinuities using information internal to the stereo algorithm of Drumheller and Poggio[2], rather than by post-processing the stereo data. The algorithm determines displacements by maximizing the sum, at overlapping small regions, of local comparisons. The detection methods are motivated by analysis of the geometry of matching and occlusion, and the fact that detection is not just a pointwise decision. Our methods can be used in combination to produce robust performance. This research is part of a project to build a Vision Machine[3] *at MIT that integrates outputs from early vision modules. Our techniques have been extensively tested on real images.*

Keywords: machine vision, occlusion, stereo, motion

This investigation describes a component of the MIT Vision Machine[3], that integrates outputs of early vision modules for tasks such as recognition and navigation. The integration stage computes maps of scene properties augmented by an explicit representation of discontinuities in the scene, identifying their physical origin. Our major achievement is the development of techniques for locating disparity discontinuities using information internal to the stereo and motion modules, rather than by post-processing the output. Later processing to detect discontinuties[4] can then operate with substantially more information about their location. We have devised techniques for discontinuity location based on an analysis of patchwise matching scores internal to the algorithm, and based on the effects of occlusion. These methods suggest improvements to the performance of stereo near disparity discontinuities.

Stereo and motion both compute similar quantities – image displacements of image elements. In the algorithms discussed here, we restrict ourselves to displacements (disparities) that are integer multiples of the pixel spacing. Thus, we can search for the best displacement for each point. We use, both in stereo and motion, a dense set of overlapping matching operators to compute displacements between the two images in stereo and motion. Both stereo and motion apply uniqueness and continuity constraints. Scene geometries differ, however, and so do interpretations of ordering constraints.

Exploiting parallelism

Early vision is computationally intensive. The computation is mostly local and isotropic: local – the result at a location in the image depends only on nearby locations, and isotropic – the same processing occurs at separate locations in the image. This suggests that a SIMD parallel architecture is a good choice to meet the performance requirements of early vision. Specifically, our computational engine is the Connection Machine[5], a fine-grain SIMD parallel computer. A further discussion of early vision and parallel computers can be found in Reference 6.

Drumheller-Poggio parallel stereo algorithm

The Drumheller-Poggio algorithm[2] served as an experimental testbed for the research described here. An extended version of the algorithm forms part of the Vision Machine: the resulting stereo data is one of the inputs to the MRF-based integration stage[4]. This section briefly reviews the original stereo algorithm, based on the description in Reference 2.

Stereo matching is an ill-posed problem[7] that cannot be solved without taking advantage of natural constraints. The continuity constraint (e.g. Reference 8) 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[9] holds: any two points must be imaged in the same relative order in the left and right eyes.

Laboratory for Computational Vision, Department of Computer Science, University of British Columbia, Vancouver, British Columbia, Canada V6T 1W5

Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA

The specific *a priori* assumption used is that the disparity 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 the images, 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[10]). We look for a discrete disparity $d(x, y)$ at each location (x, y) in the image that minimizes:

$$\|E_L(x, y) - E_R(x + d(x, y))\|_{N(x, y)} \qquad (1)$$

where the norm is a summation over a *local support neighbourhood* $N(x, y)$ centred at each location (x, y); $d(x, y)$ is assumed constant in the neighbourhood. The correlation of E_L and E_R is often used as a measure to maximize equation (1) for each (x, y):

$$\int_{N(x,y)} E_L(x, y)\, E_R(x + d(x, y), y)\, dx\, dy \qquad (2)$$

Without normalization, however, the correlation is incorrect; the proper measure is:

$$\frac{\displaystyle\int_{N(x,y)} E_L(x, y)\; E_R(x + d(x, y),\; y)\; dx\, dy}{\displaystyle\int_{N(x,y)} E_L(x, y)\; E_L(x, y),\; dx\, dy} \qquad (3)$$

The algorithm actually implemented is somewhat more complicated, since it involves geometric constraints (ordering and uniqueness) that affect the way the maximum operation is performed[1]. The Drumheller-Poggio algorithm is similar in spirit to the first stereo algorithm proposed by Marr and Poggio[8], 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 also belongs in the family of correlation-based stereo algorithms[11, 12]; the geometric constraints separate it from early algorithms. The algorithm is composed of the following steps:

1 Compute features for matching (edge detection or band-pass filtering).
2 Compute matches scores between features.
3 Determine the degree of continuity around each potential match.
4 Identify disparities based on the constraints of continuity, uniqueness and ordering.

Potential matches between features are computed as follows. The images are registered so that the epipolar lines are horizontal[13], so the stereomatching problem becomes one-dimensional: a token in the left image can match any token in the corresponding horizontal line in the right image. Sliding the right image over the left image horizontally, we compute a set of match score planes, one for each horizontal disparity. Let $p(x, y, d)$ denote the value of the (x, y) entry of the match score plane at disparity d. For edge-based tokens, the results of comparison are binary. We set $p(x, y, d) = 1$ if there is a token at location (x, y) in the left image and a compatible token at location $(x - d, y)$ in the right image; otherwise, set $p(x, y, d) = 0$. In the case of the Marr-Hildreth edge detector[14], two tokens (edges) are compatible if the signs of the convolution for each edge (the edge polarities) agree. For brightness-based matching, the matching score continuously varies (E_L and E_R vary over some finite range and the norm of their difference can take on a range of values, not just 0 and 1 – see equation (1)). The Canny edge detector[15] was used in most of the stereo examples. No thresholds were used to select edges. Our implementation required that matched the gradients of image brightness at matched edge features be within some small angle (usually 30 degrees).

The value computed by equation (1) measures the degree of continuity around each potential match at (x, y, d). For edge-based matching, pointwise feature comparison is binary and summation counts the 'votes' for the disparity d in the d^{th} match plane. If the continuity constraint is satisfied near (x, y, d) then $N(x, y)$ contains many votes and the score $s(x, y, d)$ is high (see equation (2)). When the matching is comparison of filtered brightnesses, the quantity in equation (1) attains a minimum at the correct displacement. We mostly discuss the edge-based methods used in the stereo investigation and therefore will try to maximize the normalized correlation and will speak of peaks in the measured values. Finally, we select the correct matches by applying the uniqueness and ordering constraints. To apply the uniqueness constraint, each match suppresses all other matches along the left and right lines of sight with lower 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[9, 16] (see below).

The matching scores of the stereo algorithm are valuable information. They provide a confidence level for each match that can discriminate between competing matches, as in forbidden zone suppression (using the ordering constraint). The description of the stereo algorithm implies that scores are computed only for points p and q that are potential matches (there are compatible tokens at p and q). In fact, although matches are only permitted at potential match sites, matching scores are computed everywhere with no additional computation (because of the homogeneous nature of computation in SIMD machines). Similarly, brightness-based matching produces dense information. These 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 tokens. Computing disparity between tokens 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 scores also help to suppress bad matches within occluded areas of the scene (see below). All stereo data used here is dense unless otherwise specified. Figure 1 shows a stereo scene and disparity data derived by the algorithm; isodisparity

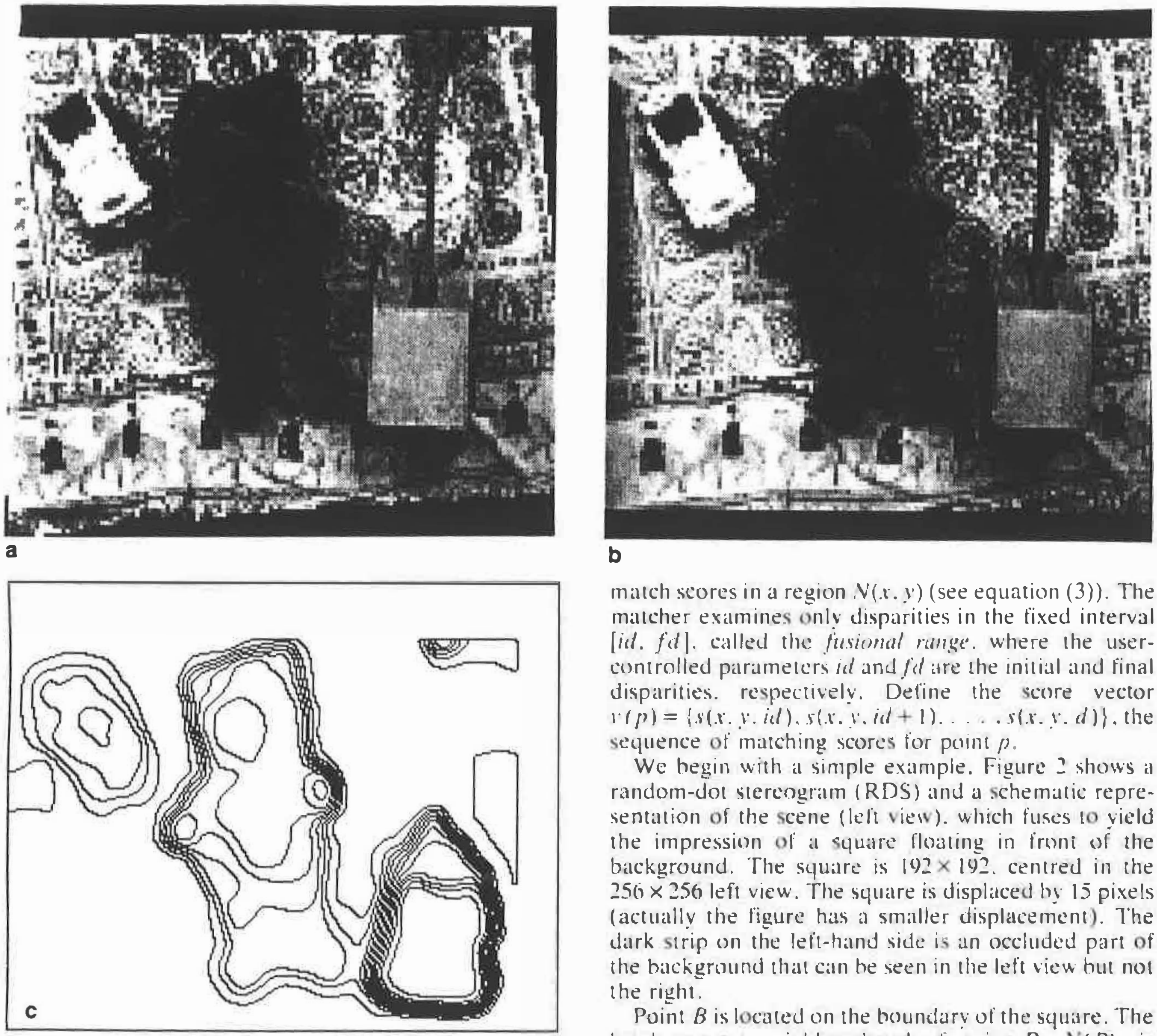


Figure 1. (a) Left view of truck, teddy bear, and crane, (b) right view, (c) isodisparity contours

contours of the interpolated disparity map depict the disparities.

DISPARITY DISCONTINUITIES

We describe two discontinuity detection techniques, arising from analysis of the behaviour of matching methods near occluding boundaries. One method is based on an analysis of matching scores for different disparities, and the other uses the effects of geometric constraints near occlusions.

Close winners

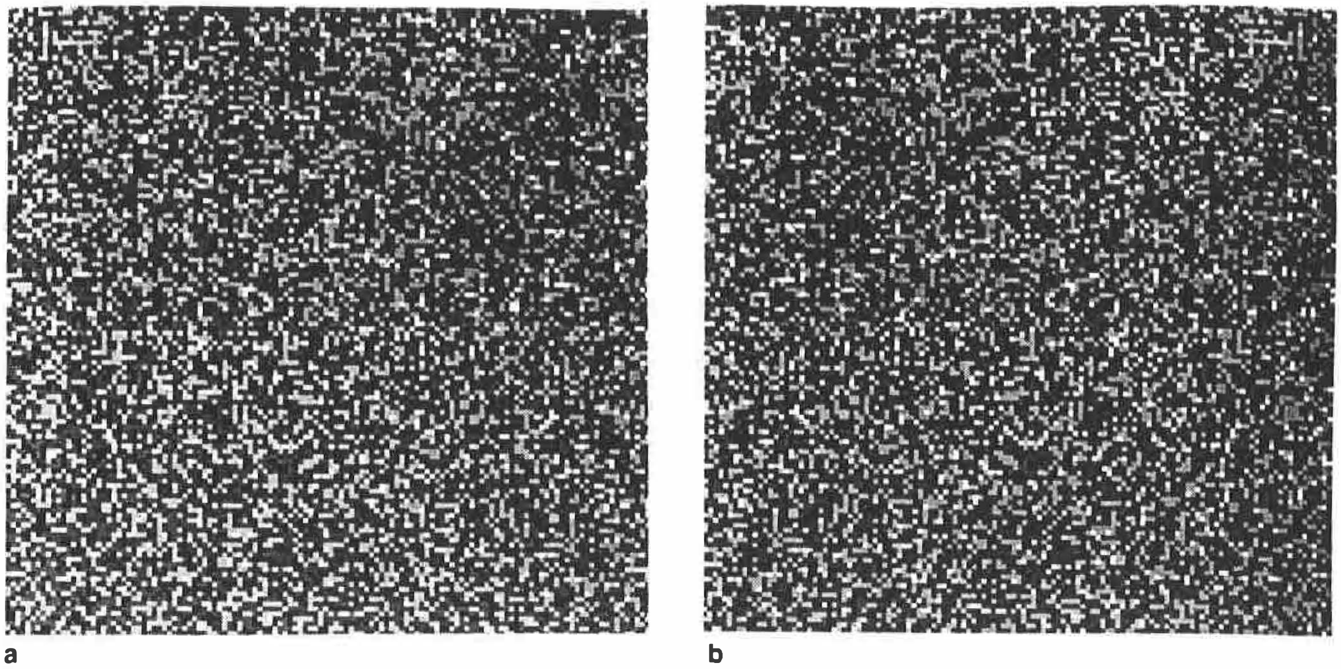
The *close winners* technique analyses stereo and motion matching scores. For each point $p = (x, y)$ in the left image and $q = (x + d, y)$ in the right image, the matcher computes a score $s(x, y, d)$ indicating the likelihood that p matches q, i.e. that p and q are images of the same physical point in the scene. The score at a point $s(x, y, d)$ is the result of integrating the pointwise match scores in a region $N(x, y)$ (see equation (3)). 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 $v(p) = \{s(x, y, id), s(x, y, id + 1), \ldots, s(x, y, d)\}$, the sequence of matching scores for point p.

We begin with a simple example. Figure 2 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 192 × 192, centred in the 256 × 256 left view. The square is displaced by 15 pixels (actually the figure has a smaller displacement). 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.

Point B is located on the boundary of the square. The local support neighbourhood of point B, $N(B)$, is divided between the square and the background. Approximately half of the edges in $N(B)$ will vote for the wrong disparity, namely the background disparity. The score vector $v(B)$, plotted as a graph of matching score *versus* disparity, is bimodal, with one peak at the foreground disparity and another peak at the background disparity. In contrast, $v(A)$ and $v(C)$ are unimodal, since their support regions cover constant disparity regions.

Figure 3 shows score vectors computed for the random-dot stereogram (RDS) – high scores represent best matches. Note that it is critical that the diameter of the support neighbourhood be larger than the largest disparity gap in the image – otherwise, the two peaks will not be detected using close winners. Also, the maximum value for the match score at B will at most be *half* that of the score at points such as A and C; this leads to a method for discontinuity identification using local spatial extrema of the match score (see Reference 16 and below).

We call point B a close winner because the 'winning' disparity has a close competitor; such points are likely to be located at disparity discontinuities. For all points p in the left image, use the following procedure to



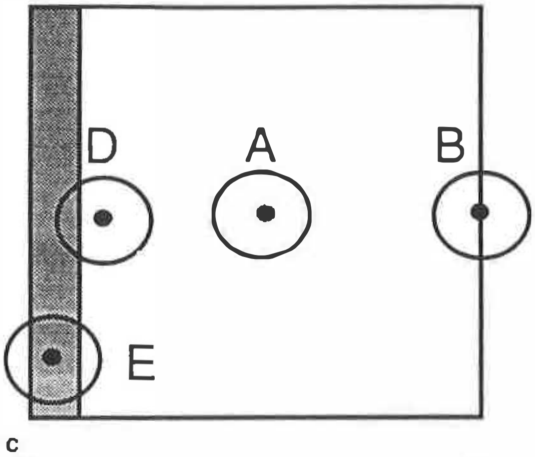


Figure 2. (a) Left view of Random Dot Stereogram, (b) right view, (c) line drawing of scene: floating square

determine whether p is a close winner:

1 Identify peaks in $v(p) = \{s_{id}, s_{id-1}, \ldots, s_{fd}\}$.

2 If $v(p)$ has two or more peaks, pick the two largest, α and β, $\alpha \geqslant \beta$. Let the margin $m = (\alpha - \beta)/\alpha$. If $m \geqslant$ (0.2 here), then p is a close winner.

When m is set to 0.2, the magnitude of the smaller peak must be greater than 80% of the larger peak. This simple peak-detection rule will not detect a peak which is actually a plateau, but this flaw is trivial to remove. Figure 4 shows close winners for several stereo scenes. The 'peak-ratio' method of Spoerri and Ullman[17] determines motion boundaries by a similar analysis of histograms of the displacement of tokens.

Figure 3. Score vectors for different locations in an RDS. (a) A: (128, 128), (b) B: (192, 128), (c) D: (85, 128). Because the support neighbourhood is 17 × 17, the support for A is contained entirely within the floating square; the support for B is split between both regions; the support for D contains the object and a portion of the left image not seen in the right

The preceding analysis assumes that the surface are frontoparallel. This assumption can be relaxed without weakening the detection of occlusions. The 'teddy' example contains surfaces that not only are not frontoparallel, but are also not planar, and the detection of discontinuities is qualitatively correct. Figure 5 shows a synethetic image of two tilted planes with randomly textured colouring, viewed under perspective projection. The normals of the background and foreground planes at (0.48, 0.55, 0.69) and (−0.47, 0.39, 0.79), respectively. The disparities on the background plane range from 5.6 to 7.6 and on the foreground from 19.3 to 20.8. Occlusion detection is poor on the right-hand side of the square since the occluded region lies there in the left image. The left-hand side is accurately located.

Spoerri and Ullman[17] used a similar method, among others, to derive a scheme for motion segmentation. Their method combines measurements of normal velocity with token matching. Local measurements determine a histogram which assumes a bimodal distribution. We base our histogram on a similar histogram of matched tokens, but without the normal velocity, which is meaningless in the context of binocular stereo. When using the 'close winners' technique in motion, the normal velocity can be used to sharpen the compatibility function for tokens, but seems to be unnecessary, given the satisfactory results demonstrated here. It is a useful check on the consistency of measurements from a displacement mechanism and an instananeous velocity mechanism producing the normal velocities. Voorhees and Poggio[18] employed a related technique to locate texture boundaries, from a statistical analysis of the local distribution of oriented tokens.

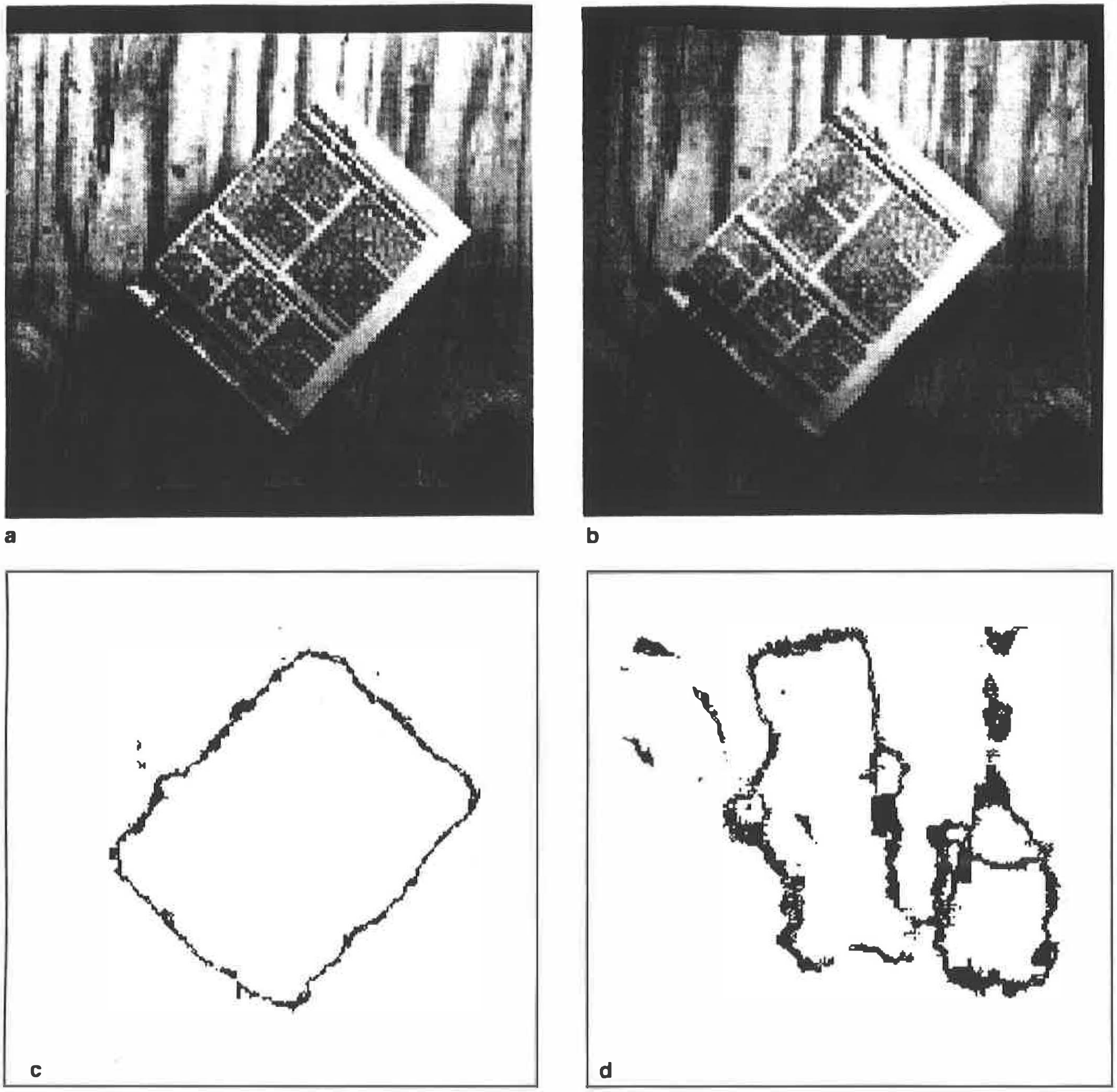


Figure 4. Close winners for several stereo scenes. (a) Newspaper on wood: left view, (b) right view, (c) close winners (newspaper), (d) close winners for teddy (see Figure 1)

Localization and close winners

Localization with the close winners technique requires additional computation. Although a bimodal score vector usually indicates a nearby disparity discontinuity, it is unclear how to locate the discontinuity precisely. In the vicinity of a discontinuity contour, the points with bimodal score vectors typically form a thin strip in the vicinity of the contour. How do we select the points that lie on the contour? Points with the smallest margin, where the two peaks in the score vector are as equal as possible, are possible candidates. Unfortunately, this approach yields the best answer only in the case of a linear contour, which splits the support neighbourhood evenly for a point on the contour. If the object boundary is convex, the point with smallest margin will be located outside the object; if the object boundary is concave, the point with smallest margin will be displaced outside of the object. Of course, these two facts are dual, merely the reference frame (the object) differs, without changing the boundary. Narrow objects such as a thin bar may be missed entirely, depending on the margin threshold and neighbourhood size. In general, a smaller neighbourhood size provides better localization but a lower signal-to-noise ratio, a trade off similar to that for the smoothing parameter in edge detection. Despite poor localization, close winners still indicate regions that are likely to contain discontinuities; a later integration stage can use this information.

The foregoing picture is an oversimplification. Consider the support neighbourhood for point D in

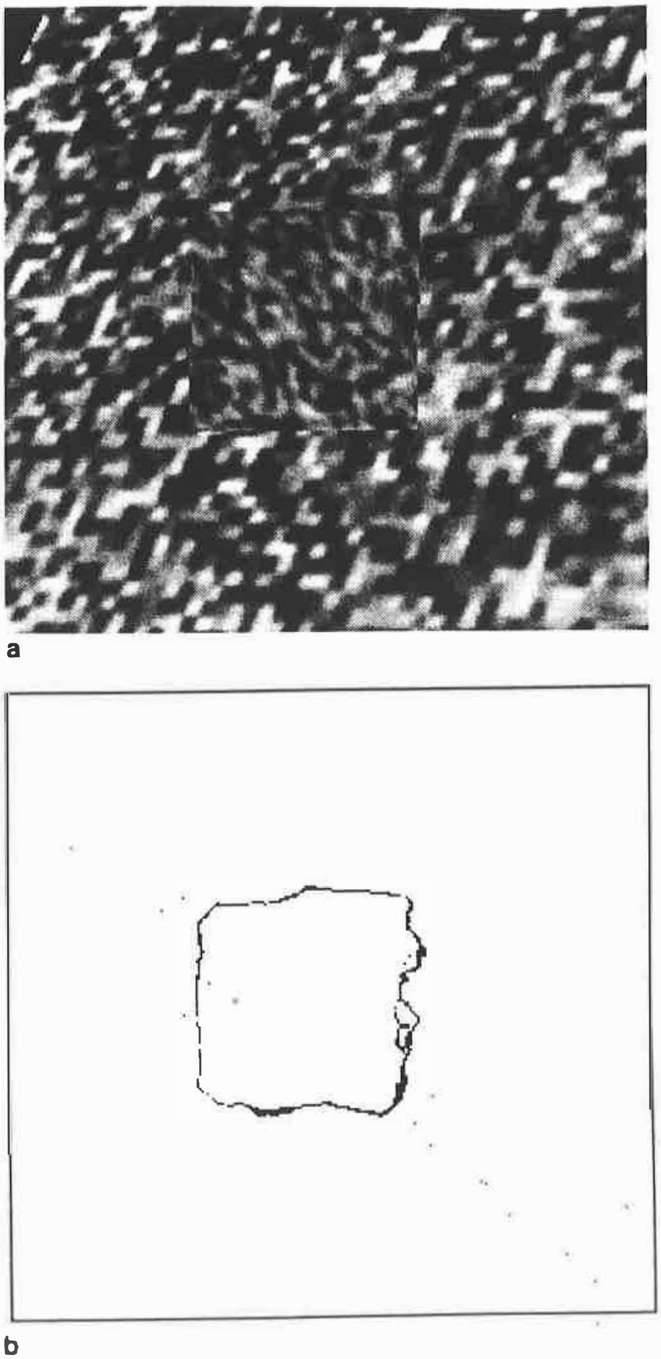


Figure 5. Close winners for tilted planes. (a) Synthetic image of tilted surfaces, (b) close winners (m = 0.15)

Figure 2. The occluded part of the neighbourhood 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 neighbourhood, 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 6d, 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.

Further analysis allows us to compensate for the effects of occlusion. The stereo algorithm uses the left image as the reference image for the disparity 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, we 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 slope of disparity change along epipolar lines, in the disparity map.) Right-to-left close winners are computed in right-image coordinates, so they must be transformed into left-image coordinates using the disparity map. Because the disparity map is ambiguous at boundaries, the transformation splits the winners into two separate contours. We resolve the ambiguity by choosing the disparity value that corresponds to the occluding contour, using the disparity change in the interpolated disparity map, marking the discontinuity at the higher disparities. Figure 6 displays the intermediate and final results.

Suppression using ordering constraint

When one surface lies in front of another, the foreground surface occludes a portion of the background surface. The location of the occluded region depends on the viewpoint. Since the boundary on the near side of an occluded region is the discontinuity contour, identifying an occluded region leads us directly to the associated disparity discontinuity. This technique can be used to locate any disparity 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, i.e. areas that are visible from the left but not the right view (see Figure 7). By definition such an area does not have a match in the right image. Thus, we could look for low matching scores as an indicator of occlusion. However, low matching scores can arise from a number of causes, including disparities outside of the fusional range of the algorithm. Later we show how the spatial variation of the matching score can be used for determining occlusions (see below). 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[9], as pictured in Figure 7a.

Consider a simple step discontinuity (Figure 7b) where the portion of the surface between points p and q is right-occluded. The shaded region contains all points that are imaged between p and q in the left view. Observe that the shaded region is contained entirely within the union of the forbidden zones for p and q: the area above the line joining p and q is in the forbidden zone for q, and the area below the line is in the forbidden zone for p. Therefore all possible matches in the left view between the images of p and q – the occluded area – have been suppressed. Match suppression is the key to locating occluded areas. In a token-based scheme, such as that of Mutch and Thompson[19], it is possible to identify occlusions at regions where most tokens are unmatched.

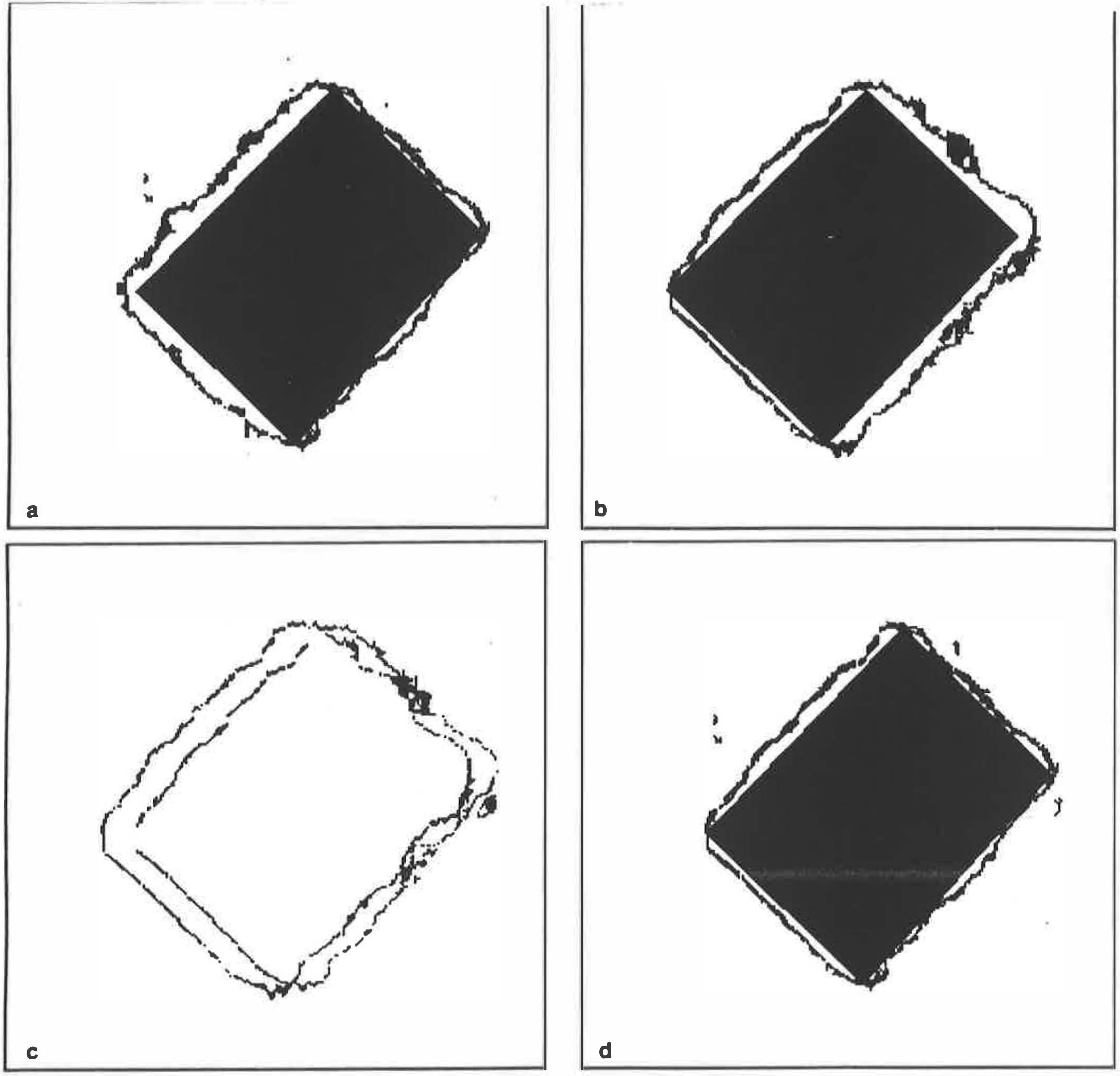


Figure 6. Combining left-to-right and right-to-left close winners. (a) Left-to-right close winners superimposed on newspaper silhouette (left view), (b) right-to-left close winners (right view), (c) right-to-left close winners transformed into left image coordinates via the disparity map, with ambiguous results, (d) combined left-to-right close winners and correctly transformed right-to-left close winners

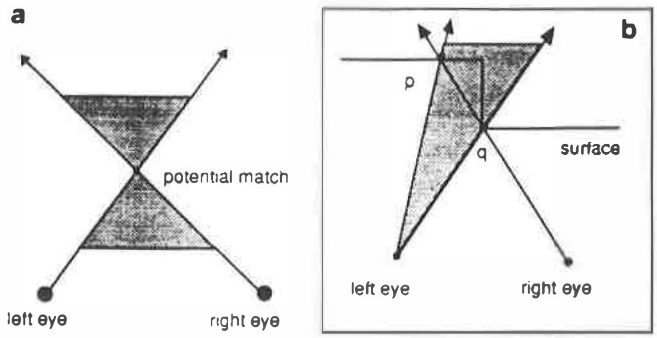


Figure 7. (a) The forbidden zone (shaded) for a particular potential match, (b) the shaded region is contained within the union of the forbidden zones for points p and q, showing that no match will be permitted there

Mechanics of match suppression

While computing matches and applying the ordering constraint, we can keep track of suppressed matches. Since matching scores are computed at all points, stereo produces dense suppression of competing matches at occlusions.

A point (x, y) in the left image is suppressed if, for all disparities d in the fusional range, the potential match at (x, y, d) has been suppressed. Suppressed points collectively determine regions of suppression that correspond to right-occluded areas. Disparity discontinuities are points on the right-hand side of suppressed regions because that is the near side in the case of right-occlusion. Others have noted[20] the connection between matching and identification of occlusions, but do not tie it in to the full ordering constraint. Figure 8 shows the

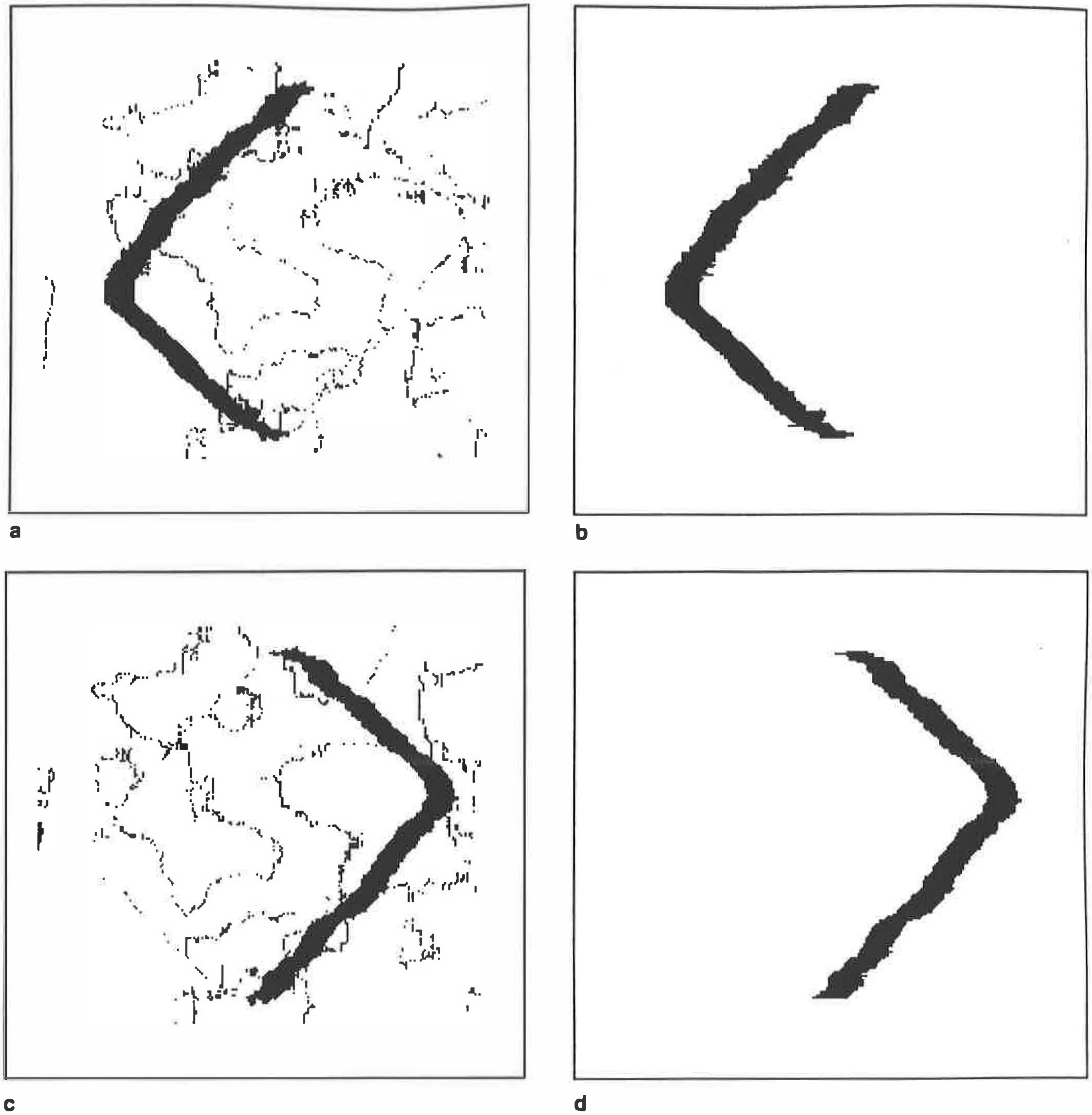


Figure 8. Identifying right-occluded and left-occluded regions for newspaper scene. (a) Suppressed points for right-occlusion, (b) filtered suppressed points, (c) suppressed points for left-occlusion, (d) filtered suppressed points

suppressed regions for the newspaper scene. Some suppressed points are part of significant occluded regions and others result from incorrect matches or disparity quantization effects. (A region at disparity k occludes a pixel-wide strip of a region at disparity $k-1$ adjoining it on the left – this is actually an occlusion of 1 disparity level!) As a simple measure to select significant regions, we threshold the width of contiguous strips of suppressed points. A more robust procedure would be to find connected components of suppressed points, then threshold the size of the connected component. Spurious matches that occasionally occur inside a completely occluded region can then be located and discarded. Figure 8 shows the suppressed regions and filtered suppressed regions for the left-occluded and right-occluded regions of the newspaper-on-wood scene.

There still remains the problem of detecting left-occluded areas (visible from the right view but not from the left view). Left-occluded areas are found by running the same analysis, but matching the right image to the left image instead of the reverse. For left-occluded areas, the associated disparity discontinuities lie on the left-hand (again, near) 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, but the correct disparity is always the near (larger) value.

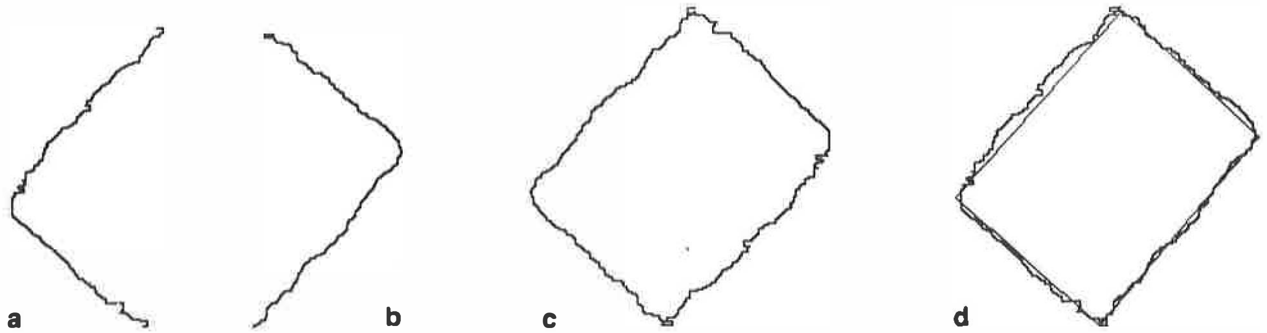


Figure 9. Identifying left-occluded regions. (a) Associated disparity discontinuities (left), (b) associated disparity discontinuities (right), (c) merged discontinuities from left-occlusion and right occlusion, (d) merged discontinuities superimposed on newspaper silhouette (left view)

Given the disparity, right image discontinuities can be mapped into the left image. Our analysis is depicted in Figure 9.

Finally, there is an additional benefit of identifying occluded areas. Knowledge of occlusion can improve naive interpolation. Interpolation blurs discontinuities, filling in occluded areas with depth data from both sides. A better approach assumes that an occluded area has the same disparity as the background, e.g. filling in right-occluded regions with disparity values from left to right[21].

Improving stereo performance near discontinuities

Most stereo algorithms depend on the continuity constraint, the assumption that disparity varies smoothly almost everywhere. It is not surprising that stereo performs poorly at object boundaries, where disparity is discontinuous. In this section we explore the possibility that knowledge of disparity 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 neighbourhoods can cross disparity boundaries and pick upon misleading information from the other side, as shown in Figure 10.

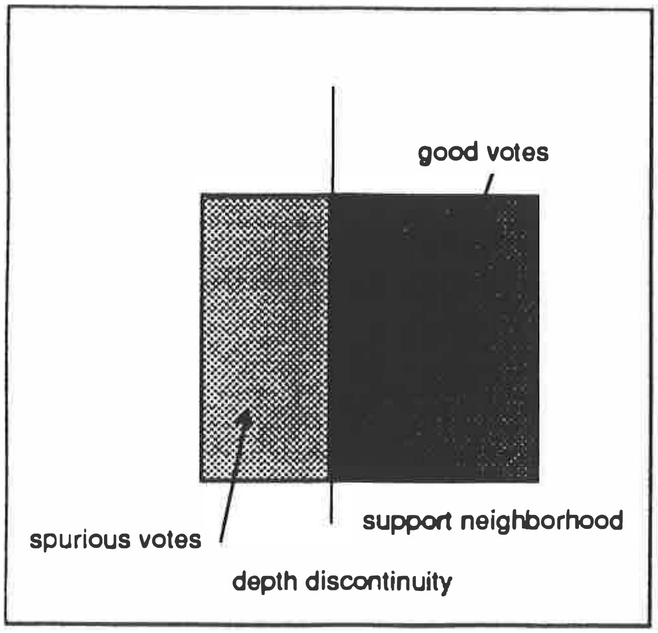


Figure 10. The suport neighbourhood for the black dot includes some spurious votes

Given knowledge of discontinuity locations, one can consider improving stereo performance by reshaping support neighbourhoods so that they do not cross discontinuities. In Figure 10, we would keep the dotted region and throw away the rest. The discontinuities used to bound neighbourhoods 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 brightness edges[22]. However, brightness edges alone are not very informative, since only a few of them coincide with disparity boundaries; the role of the integration stage is precisely to select such edges. The problem with reshaping the neighbourhoods in this way is that errors in localization of the discontinuities can be catastrophic; the entire remaining support neighbourhood 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. Specifically, different scales are implemented by using different support neighbourhood sizes, a large neighbourhood for a coarse scale. Use disparity values from one scale to guide search at the next scale, except near disparity boundaries. For a point near a disparity boundary, a large neighbourhood 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 11 shows an exaggerated example of such an

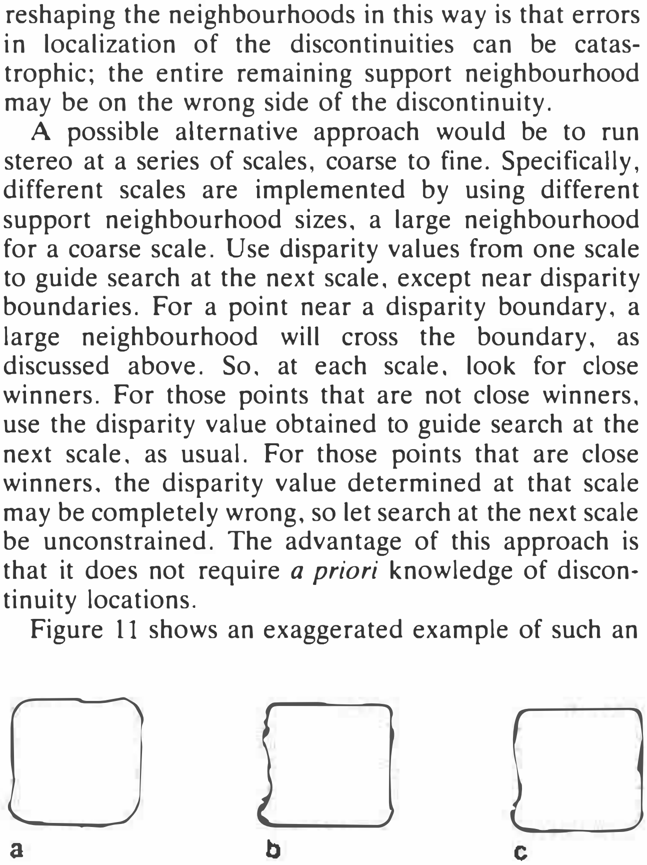


Figure 11. Multiple scales: boundary of region with disparity = 10. (a) Large scale – support width = 23, (b) small scale – support width = 7, (c) combination of scales

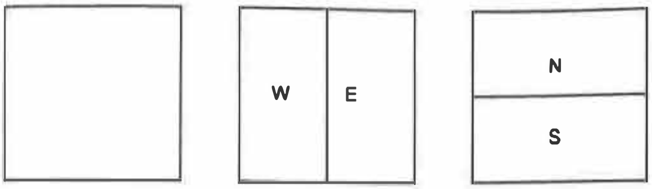


Figure 12. Half neighbourhoods: a support region and subregions

approach. We have run stereo on the RDS of Figure 2 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 disparity 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.

Varying support regions

The output of the stereo and motion modules depends on the size of the support neighbourhood. When a depth or motion discontinuity bisects the support region, the results are less reliable, but can be analysed to identify occlusions. By using a set of smaller support regions that divide a support region (see Figure 12), we can get the response of the matching module over the range of supports. At a discontinuity, the set of detectors should give different outputs, and otherwise should agree. So, we mark the points where displacement is detected but the set of detectors do not agree as discontinuities (Figure 13). This work is preliminary, but promising. From this we should be able to expand the number of orientations cutting the neighbourhood and identify the orientation of discontinuities.

SEGMENTATION IN MOTION

As mentioned above, motion is analogous to binocular stereo – time between images in motion replaces displacement of viewpoints in stereo. There are no problems with camera parameters, since there is only one camera. But the set of displacements is larger and two-dimensional. In stereo, horizontal lines give no disparity cue; in motion, no particular orientation is ambiguous, but when the image brightness function is locally well approximated by a plane, ambiguity occurs.

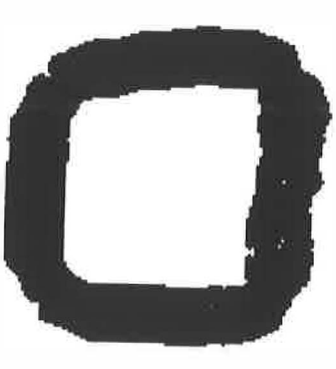


Figure 13. Slanted planes: occlusion identification points where multiple support subregions differ

The motion algorithm of Little *et al.*[16] and Bülthoff *et al.*[23] is analogous to the Drumheller-Poggio[2] stereo algorithm. The algorithm searches for a discrete displacement at each (x, y) to minimize the summed differences of local patches of the two images. A simple uniqueness constraint is employed; each point in the second image finds its best match in the first. This is improved by adding a second constraint, namely, that matching be symmetric, that each point in the first image find its best match, and that the match be also the best match for the point in the second image. This exactly corresponds to suppression along both lines of sight as in stereo.

The full forbidden zone (see above) of stereo has an analogous structure for motion. For stereo, the structure of the disparity planes leads to a 3-dimensional structure, in which the zone of suppressed points is 2-dimensional, lying in an epipolar plane. Since the displacements for motion are 2-dimensional, the displacement space is 4-dimensional – the matching score at each point must be compared with the score at all points whose $x-y$ coordinate can be reached by a displacement within the given range. One dimension is disparity, while the other two are those spanned by the $x-y$ displacements in the image.

Occlusion in motion

Several simple methods have proven effective for detecting occluding boundaries, based on the local spatial variation of the matching scores. Again, these techniques recognize that, at the image of boundaries, a detector of finite spatial extent will overlap regions having differing displacements.

Local differences in the flow field

The detection of occlusion can occur before or after determination of local motion measurements, i.e. the flow field computation. One technique, inspired by models derived from studies of the fly[24], computes, for each component of the displacement, the average magnitude of the velocity over two spatial scales, λ_1 and $\lambda_2 << \lambda_2$. It marks as boundary points those locations where the magnitude at scale λ_1 is significantly larger (or smaller) than the magnitude at scale λ_2. The difference in magnitudes in the two scales is approximately zero within regions of slowly varying displacement and large at points adjacent to rapid changes in magnitude of either component of the displacement. Note that this in fact consists of marking the high outputs of a filter that is essentially a difference of Gaussians. This can be applied either to the entire vector $v(x, y)$ or to each component, and then combined. Nakayama and Loomis[25] suggested a similar method, but did not implement it. The lobula method, of course, examines the output of the flow, and is not thus a *direct* technique.

Hildreth[26] detects motion boundaries by finding locations where there is locally a change in the sign of the normal flow component. This has the advantage

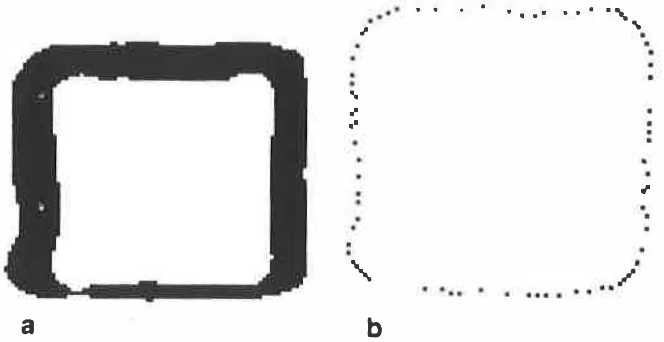
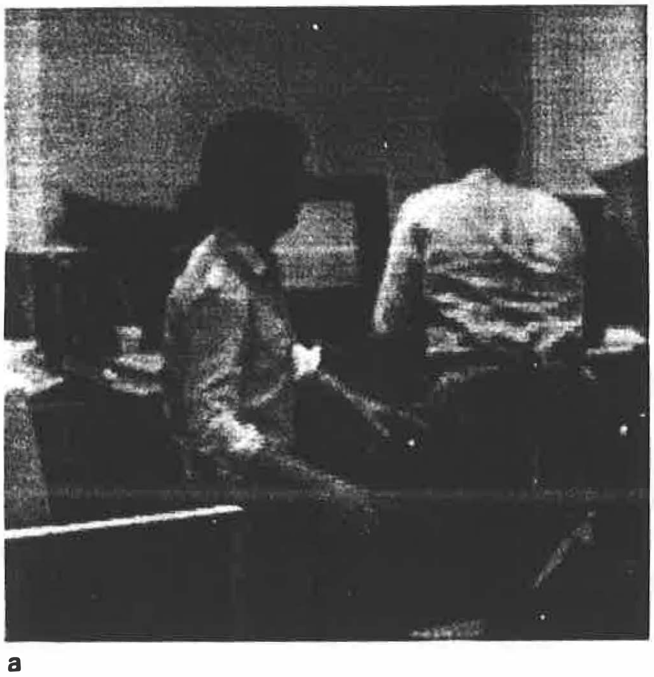
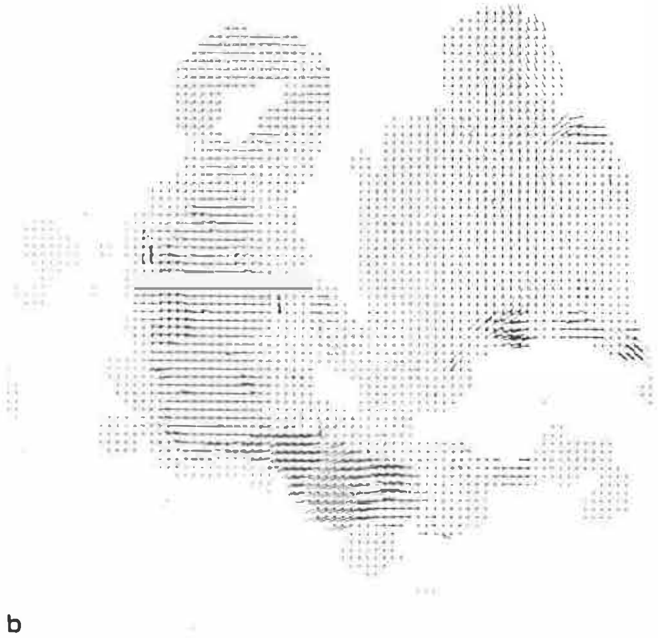


Figure 14. Random Dot Stereogram: occlusion identification. (a) Locally low scores, (b) low scores and local minima





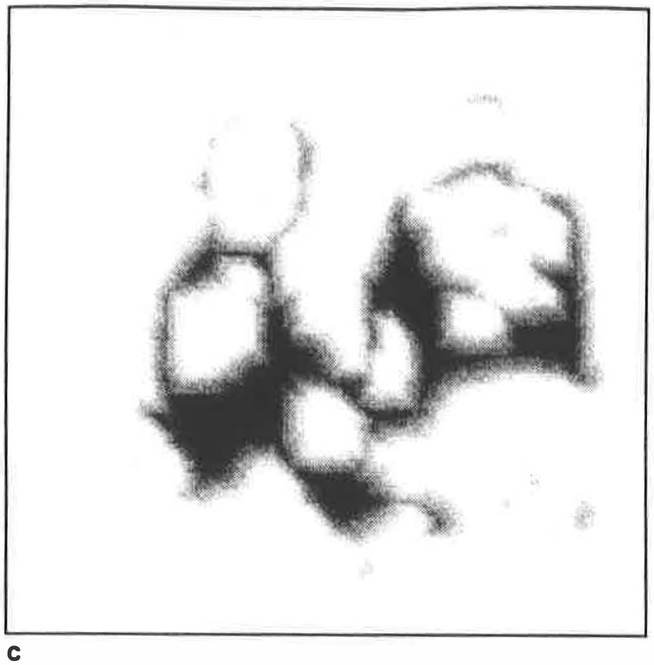


Figure 15. (a) Rotating and translating persons, (b) motion field, (c) continuous skeleton of locally low scores

of deciding on boundary location before any local smoothing blurs their location. Even normal components, however, need non-local information.

Locally low scores

Because of the behaviour of matching near occlusions, the score vector (in motion a 2-dimensional histogram) should be bimodal. Neither of these peaks are as high as the peaks recorded by points whose patch does not overlap occlusions. Thus, the magnitude of the best matching score should vary spatially near boundaries. When the best score is *locally low*, small with respect to the local average, it is likely that the point lies on a boundary. Again, detecting this event involves marking points where the difference of Gaussians, each appropriately scaled, produces negative values. Unlike the previous method, this technique is applied to matching scores and not the velocity field and its components, and thus is a direct technique. The locally low scores method is attractive since it does require analysing the distribution of scores (as in the close winners method), which is prohibitively expensive for motion. Two facts increase the cost – the number of displacements may be large and the analysis must be done in two dimensions, unlike stereo. Figure 14 shows the results of applying the locally low score method to the Random Dot Stereogram of Figures 2c and d.

Figure 15 shows the motion input, vector field and segmentation by locally low scores applied to a motion sequence of two researchers: the left figure rotates toward the left and the right is moving upward.

Both of these methods are less reliable in areas where there is little information – either represented by few edges or, equivalently, small spatial variation in brightness. It is important to suppress points where the local information available to the matcher is small, when detecting boundaries by either of the above

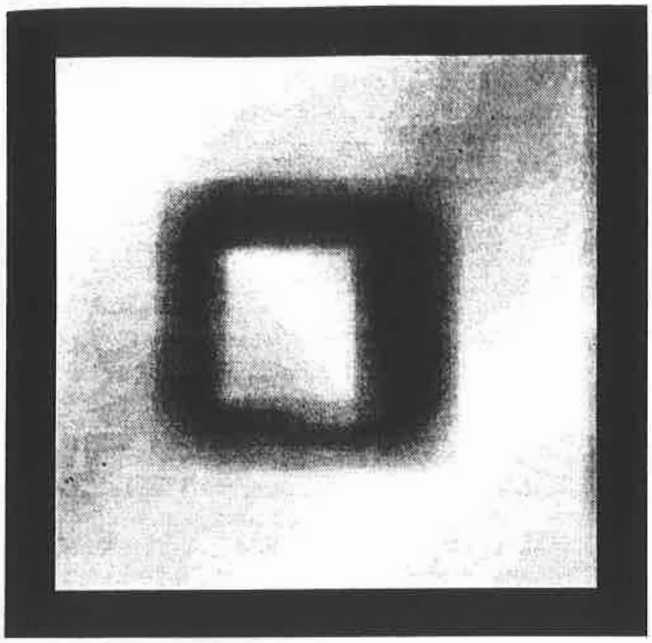


Figure 16. Scores for the tilted plane stereo stimulus

methods. We have implemented techniques in the motion algorithm that measure the average image curvature in an area, and suppress output of discontinuities based on low scores in those areas. Anandan[27] has devised a multi-scale correlational motion technique that provides oriented local measures of matching. His method uses small (5×5) local summation regions and correlation. He has observed that the scores are low at occluding boundaries, and suggested that occlusion could be determined by appropriate analysis.

Note that the locally low scores method also works for stereo examples such as the tilted lanes shown in Figure 5. The scores for those matches are shown in Figure 16.

CONCLUSION

We have addressed the detection of discontinuities in stereo and motion, within the context of efficient, parallel implementation. The techniques we have examined all use information internal to the correspondence process to identify discontinuities. Any later processing to determine the figure/ground relation and to improve surface description (such as interpolation) begins with an almost complete description of the location of discontinuities.

While we have restricted the implementation and discussion to integer displacements, there is nothing inherent in the method that precludes computing displacement at subpixel precision. These techniques all can easily be implemented on a SIMD parallel computer, suggesting that their implementation in simple circuits is feasible.

ACKNOWLEDGEMENTS

This report describes research done within the Artificial Intelligence Laboratory at MIT as well as at the University of British Columbia. Support for the AI Laboratory's artificial intelligence research is provided by the Advanced Research Projects Agency of the Department of Defense under Army contract DACA76-85-C-0010, and in part under Office of Naval Research (ONR) contract N00014-85-K-0124. Primary support for Mr Gillett came from NIGMS Training Grant T32-GM07484, administered by the MIT Department of Brain and Cognitive Sciences. Support for this research is also provided by a Hughes Aircraft Corporation gift to the Artificial Intelligence Center for T Poggio, by National Science Foundation Contract No. MIP-8814612, and by a grant from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

1 **Thompson, W B, Mutch, K M and Berzins, V A** 'Dynamic occlusion analysis in optical flow fields' *IEEE Trans. PAMI* Vol 7 No 4 (1985)

2 **Drumheller, M and Poggio, T** 'On parallel stereo' *Proc. IEEE Conf. Robot. & Automat.* Washington, DC, USA (1986) pp 1439–1448

3 **Poggio, T, Little, J J, Gamble, E, Gillett, W, Geiger, D, Weinshall, D, Villalba, M, Larson, N, Cass, T, Bülthoff, H, Drumheller, M, Oppenheimer, P, Yang, W and Hurlbert, A** 'The MIT Vision Machine' in **Winston, P (eds)** *Artificial Intelligence at MIT: Expanding Frontiers* MIT Press, USA (1990)

4 **Poggio, T, Gamble, E B and Little, J J** 'Parallel integration of vision modules' *Science* Vol 242 No 4877 (October 21 1988) pp 436–440

5 **Hillis, L D** *The Connection Machine* MIT Press, MA, USA (1985)

6 **Little, J J, Blelloch, G E and Cass, T** 'Algorithmic techniques for vision on a fine-grained parallel machine' *IEEE Trans. PAMI* Vol 11 No 3 (March 1989) pp 244–257

7 **Bertero, M, Poggio, T and Torre, V** 'Ill-posed problems in early vision' *Proc. IEEE* Vol 76 No 8 (August 1988) pp 869–889

8 **Marr, D and Poggio, T** 'Cooperative computation of stereo disparity' *Science* Vol 194 No 4262 (15 October 1976) pp 283–287

9 **Yuille, A L and Poggio, T** 'A generalized ordering constraint for stereo correspondence' AI Memo No 777, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, MA, USA (1984)

10 **Kass, M** 'Computing visual correspondence' *From Pixels to Predicates* Ablex Publishing Corporation, NH, USA (1986)

11 **Levine, M D, O'Handley, D A and Yagi, G M** 'Computer determination of depth maps' *Comput. Vision Graph. & Image Process.* Vol 4 No 4 (October 1973) pp 131–150

12 **Nishihara, H K** Practical real-time imaging stereo matcher' *Optical Eng.* Vol 23 No 5 (1984) pp 536–545

13 **Horn, B K P** *Robot Vision* MIT Press, MA, USA (1986)

14 **Marr, D and Hildreth, E** 'Theory of edge detection' *Proc. Royal Soc. Lond. B* Vol 207 (1980) pp 187–217

15 **Canny, J F** 'A computational approach to edge detection' *IEEE Trans. PAMI* Vol 8 No 6 (1986)

pp 679–698

16 **Little, J L, Bülthoff, H H and Poggio, T** 'Parallel optical flow using local voting' *Proc. Int. Conf. Comput. Vision* Tarpon Springs, FL, USA (December 1988)

17 **Spoerri, A and Ullman, U** 'The early detection of motion boundaries' *Proc. Int. Conf. Comput. Vision* London, UK (June 1987)

18 **Voorhees, M and Poggio, T** 'Computing texture boundaries from images' *Nature* Vol 333 No 6171 (1988) pp 364–367

19 **Mutch, K M and Thompson, W B** 'Analysis of accretion and deletion at boundaries in dynamic scenes' *IEEE Trans. PAMI.* Vol 7 (1985) pp 133–138

20 **Weng, J, Ahuja, P P and Huang, T J** 'Two-view matching' *Proc. Int. Conf. Comput. Vision* Tarpon Springs, FL, USA (December 1988)

21 **Gillett, W** *Issues in parallel stereo matching* Master's thesis, Massachusetts Institute of Technology, MIT, USA (1988)

22 **Gamble, E B and Poggio, T** 'Visual integration and detection of discontinuities: the key role of intensity edges' AI Memo No. 970, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, MIT, USA (October 1987)

23 **Bülthoff, H H, Little, J L and Poggio, T** 'A parallel algorithm for real-time computation of optical flow' *Nature* No 337 (1989) pp 549–553

24 **Reichardt, W and Poggio, T** 'Figure-ground discrimination by relative movement in the visual system of the fly. Part I: Experimental results' *Biol. Cybern.* Vol 35 (1979) pp 81–100

25 **Nakayama, K and Loomis, J M** 'Optical velocity patterns, velocity-sensitive neurons, and space perception: a hypothesis' *Perception* Vol 3 (1974) pp 63–80

26 **Hildreth, E C** *The Measurement of Visual Motion* MIT Press, MA, USA (1984)

27 **Anandan, P** 'A computational framework and an algorithm for the measurement of visual motion' *Int. J. Comput. Vision* Vol 2 (1989) pp 283–310