

Segment-Based Stereo Matching*

GERARD MEDIONI AND RAMAKANT NEVATIA

*Intelligent Systems Group, Departments of Electrical Engineering and Computer Science,
University of Southern California, Powell Hall of Engineering PHE 306,
Los Angeles, California 90089-0272*

Received November 5, 1983; accepted January 28, 1985

Images are 2-dimensional projections of 3-dimensional scenes, therefore depth recovery is a crucial problem in image understanding, with applications in passive navigation, cartography, surveillance, and industrial robotics. Stereo analysis provides a more direct quantitative depth evaluation than techniques such as shape from shading, and its being passive makes it more applicable than active range finding imagery by laser or radar. This paper addresses the subproblem of identifying corresponding points in the two images. The primitives we use are groups of collinear connected edge points called *segments*, and we base the correspondence on the "minimum differential disparity" criterion. The result of this processing is a sparse array disparity map of the analyzed scene. © 1985 Academic Press, Inc.

1. INTRODUCTION

Many problems of scene analysis arise due to inherent depth ambiguities in a monocular 2-D image. Humans easily perceive depth in 2-D images, but the process of doing so is not well understood. Stereopsis provides a more direct way of inferring the 3-D range, and is also heavily used by the human visual system. Depth data derived from stereo has been used previously for applications in a variety of domain, such as passive navigation [8, 18], cartography [13, 23], surveillance [12].

Following the terminology of Barnard and Fischler [6], we can view the problem of stereo analysis as consisting of the following steps:

- image acquisition,
- camera modeling,
- feature acquisition,
- image matching,
- depth determination,
- interpolation.

The key step is that of image matching, that is, the process of identifying the corresponding points in two images that are cast by the same physical point in 3-D space. This paper is devoted solely to this problem. The matching is made difficult, in part, by changes in the images of the corresponding points due to different

*This research was supported by the Defense Advanced Research Projects Agency, DARPA Order 3119, and was monitored by the Air Force Wright Aeronautics Laboratory under Contract F-33615-82-K-1786.

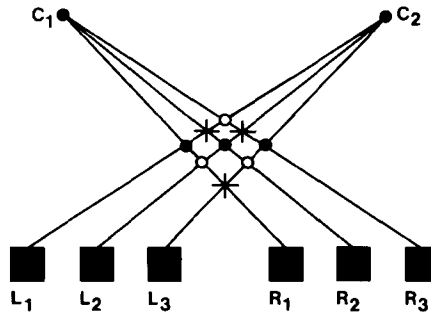


FIG. 1. Example of ambiguity.

perspective view points. The amount of change is dependent on the stereo angle (the angle between the rays joining the object point to the image points). We are interested in analyzing aerial images where the stereo angle may be rather large (up to 60°).

However, the major difficulty of stereo processing arises due to the need for making *global correspondences*. A local point or area in one image may match equally well with a number of points or areas in the other image (this problem is compounded by the fact that the local matches are not perfect due to the reasons described above). These ambiguities in local matches can only be resolved by considering sets of local matches globally and imposing some preference criterion. Note that the human system needs to do so as well. As an example (following [15]) consider Fig. 1, which contains three squares in each of the left and the right images. Each square in one image is similar to any of the three in the other. If we correspond L_1 with R_1 , L_2 with R_2 , and L_3 with R_3 , the three squares will be computed to be at the same height above the background, shown by the filled circles in Fig. 1. However, if L_1 is matched with R_2 , L_2 with R_3 , and L_3 with R_1 , then the computed heights of the squares will be indicated by the cross-marks. Another possible interpretation is shown by the filled circles. Given this image, humans perceive the three squares in one plane, and we may say that this interpretation is preferred as being the *simplest*.

The various stereo matching algorithms differ in the primitives used for matching, the method used for local matching, and the method used for global matching, if any. In this paper, we introduce the use of a line segment as a primitive, and define a new criterion called “minimal differential disparity” for global matching. We believe that this method has several advantages over previous methods and is hence able to handle scenes that are more difficult (such as those containing repetitive structures). These issues are discussed more fully in Section 4, where we present some results and conclusions.

First, we give a brief review of previous stereo methods, then describe our matching method and finally present the results and conclusions.

2. REVIEW OF EXISTING METHODS

Two broad classes of techniques have been used for stereo matching, area-based and feature-based.

2.1. Area-Based Stereo

Ideally, one would like to find a corresponding pixel for each pixel in each image of a stereo pair, but the semantic information conveyed by a single pixel is too low to resolve ambiguous matches, therefore we have to consider an area or neighborhood around each pixel. A local match for an area is found by searching in the other image for a best match defined by a cross-correlation measure, sum of absolute differences of pixel intensity or other similar measures (for a survey see [21]).

The search space can be reduced by observing that the points on one line in one image are constrained to match with points along a certain specific line in the other image; these sets of lines are known as *epipolar lines*. The epipolar lines are intersections of the two image planes with an *epipolar plane*, defined to be the plane passing through an object point and the two camera foci. In a simple geometry, where the two cameras are related by a simple horizontal displacement, the epipolar lines are simply the horizontal lines, i.e., the matching points must have the same row values (see Fig. 2). Other methods of reducing search include use of intermediate views (e.g., [19]) and use of coarse-to-fine match (e.g., see [18]).

Typically, area-based stereo systems do not explicitly attempt a global match, though some global constraints are achieved by preferring continuity in the disparity values of neighboring areas (disparity is the displacement of corresponding points between the two images). All systems based on area-correlation suffer from the following limitations:

- They require the presence of a detectable texture within each correlation window, therefore they tend to fail in featureless or repetitive texture environments.
- They tend to be confused by the presence of a surface discontinuity in a correlation window.
- They are sensitive to absolute intensity, contrast, and illumination.
- They get confused in rapidly changing depth fields (e.g., vegetation).

For these reasons, the existing systems, especially the ones used in “automatic” cartography, require the intervention of human operators to guide them and correct them. Such systems are described in [14, 23, 11, 5, 18].

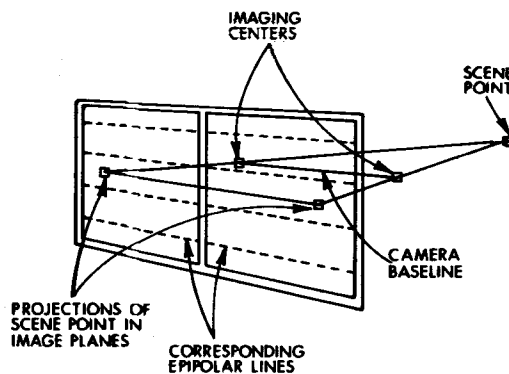


FIG. 2. Collinear epipolar geometry.

2.2. Feature-Based Systems

These systems match features derived from the two images rather than the intensity arrays. The commonly used features in the past have been edges detected in the images. Note that an isolated edge is not very distinguished and explicit global correspondence mechanisms must be employed. The constraints of epipolar lines are still useful, as are the techniques of coarse-to-fine matching and intermediate views. However, the feature matching necessarily leads to a sparse depth map only and the rest of the surface must be reconstructed by interpolation. Some advantages of feature-based systems are:

- They are faster than area-based methods, because there are many fewer points (features) to consider.
- The match is more accurate as edges may be located with sub-pixel precision [7].
- They are less sensitive to photometric variations, since they represent geometric properties of a scene.

Following is a brief survey of previous feature-based stereo systems.

Henderson [12] considered scenes representing cultural sites (man-made structures) and matched edge points on epipolar lines in the two views. The ambiguity is reduced by assuming continuity between consecutive epipolar lines.

Marr and Poggio have suggested use of the following two constraints [16]:

1. *Uniqueness*. Each point in an image may be assigned at most one disparity value. (This assumption is not valid for transparent objects.)
2. *Continuity*. Matter is cohesive, therefore disparity values change smoothly, except at a few depth discontinuities.

They first proposed a cooperative algorithm [15] that works very well on random-dot stereograms, but then rejected it to propose one of a more heuristic nature, implemented by Grimson [9, 10]. In this system, the features are zero-crossings of the convolution with Laplacian-Gaussian edge masks. A match for a zero-crossing in one image is centered in three locations (“pools”) in the other image, one corresponding to the extended disparity, and the others on two sides of it. If two matches are found around the same location (or in the same pool), it is decided that no match exists. If a match is found in more than one pool, then the pool that gives a disparity value similar to disparities of other points in its neighborhood is chosen.

Arnold and Binford also match edges using local context [1]. Baker and Binford match edges on epipolar lines and use the additional constraint that order must be preserved [4]. The ambiguities are resolved by optimizing a matching criterion that is a function of the intervals between the matching edges; the optimization is achieved by a dynamic programming algorithm.

These systems give good performance on the images shown in their publications. However, they have not been tested on complex images with highly repetitive structures and other difficulties.

3. MINIMUM DIFFERENTIAL DISPARITY ALGORITHM

We now describe our algorithm for stereo image matching. Our algorithm uses straight line segments as primitives, which have some advantages discussed below.

We use a different and new “minimal differential disparity” criterion for global matching. This method is described in detail below.

3.1. Choice of Primitives

We believe that the feature-based stereo systems have strong advantages over area-based correlation systems. However, edges as primitives may be too local. In particular, the previous edge-based methods incorporate only intra-scanline constraints in their main search process; continuity across scan-lines (or across epipolar lines) is checked only after this process is completed, if at all. (In recent work, Ohta and Kanade have extended the dynamic programming technique to include the inter-scanline search; however, this results in a 3-D search which is inherently very expensive computationally [22].) In our algorithm, we match line segments, which consist of connected edges, and hence the inter-scanline connectivity is implicit in the matching process. Higher level primitives such as object boundaries, may be even more advantageous; however, detection of such boundaries is a complex scene analysis task and it may require the output of the stereo processing itself.

The line segments used in our algorithm are derived by using the Nevatia–Babu method [20]. This method consists of detecting local edges by using step edge masks in various orientations, then thinning and linking the edges, and finally fitting the curves by piecewise-linear segments. The local edges could alternately come from the zero-crossings of a Laplacian–Gaussian edge mask, as in [17]; our software processes these edges in the same way as those produced by the Nevatia–Babu method and produces line segments in the same form. These segments are described by

- coordinates of the end points
- orientation
- strength (average contrast).

Note that when matching line segments, we need to allow one segment to possibly match with more than one segment in the other image (i.e., to allow for fragmented segments), even if we wish to preserve unique matches for the individual edge points. Also, instead of considering one epipolar line at a time, we have to consider all epipolar lines in which a given segment appears.

3.2. Assumptions and Definitions

We assume that the epipolar lines are along the horizontal axis (there is no loss of generality here, as the images can always be transformed to achieve this property, as long as the camera geometries are known). We also assume a bound on the disparity range allowed for any given segment, let us call it maxd .

Let $A = \{a_i\}$ be the set of segments in the left image and let $B = \{b_j\}$ be the set of segments in the right image.

For each segment a_i in the left image, we define a window $w(a_i)$ in which corresponding segments from the right image must lie and, similarly, for each segment b_j in the right image, we define a window $w(b_j)$ in which corresponding segments from the left image must lie. The shape of this window is a parallelogram, one side is a_i , for left to right match, and the other a horizontal vector of length $2 \cdot \text{maxd}$, as shown in Fig. 3 (one side is b_j for match from right to left). One can see that a_i in $w(b_j)$ implies b_j in $w(a_i)$.

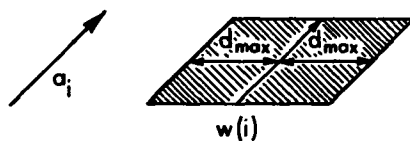


FIG. 3. Example of window construction.

Another notion we need to define is that of “overlapping” of two segments, whether in the same image or not. We say that two segments u and v overlap, and we write $u \leftrightarrow v$, if by sliding one of them in a direction parallel to the epipolar line, they would intersect. The overlap property is important because two segments in two images are considered for matching only if they overlap. Further, we allow a segment in one image to match more than one segment in the other image provided that the candidates do *not* overlap; in that case, the alternatives are not viewed as a conflict but as possible matching fragments of a longer segment. The amount of overlap is also used in weighting a particular match, as discussed later.

We define a boolean function $p(i, j)$ indicating whether two segments are potential matches, as: $p(i, j)$ is true iff

- $b_j \leftrightarrow a_i$;
- a_i and b_j have “similar” contrast
- a_i and b_j have “similar” orientation.

The required similarity in orientation is loose and is a function of the segment length. We have set it to be 25° for long segments and up to 90° for very short segments. For 2 segments of length l_1 and l_2 , let $l = \min(l_1, l_2)$. The tolerance τ is computed as follows:

$$\begin{aligned} \text{if } l < \epsilon \text{ then } \tau &= 90^\circ \\ \text{else } \tau &= \max(25^\circ, \tan^{-1}(\epsilon / \sqrt{l^2 - \epsilon^2})). \end{aligned}$$

In all examples, we used $\epsilon = 4$ pixels.

Two segments are defined to have similar contrast if the absolute value of the difference of the individual contrasts is less than 20% of the larger one.

To each pair (i, j) such that $p(i, j)$ is true, we associate an *average disparity* d_{ij} which is the average of the disparity between the two segments a_i and b_j along the length of their overlap. (Disparity is the displacement along the epipolar line of the corresponding points.)

We define the two sets S_p and $S_{\bar{p}}$ as

$$\begin{aligned} S_p(a_i) &= \{j | b_j \text{ in } w(a_i) \text{ AND } p(i, j) \text{ is true}\} \\ S_{\bar{p}}(a_i) &= \{j | b_j \text{ in } w(a_i) \text{ AND } p(i, j) \text{ is false}\} \\ \text{Card}(a_i) &= \text{number of segments in the set } S_p(a_i) \cup S_{\bar{p}}(a_i) \\ &= \text{number of segments in the window } w(a_i). \end{aligned}$$

Similarly, we define $S_p(b_j)$, $S_{\bar{p}}(b_j)$, and $\text{Card}(b_j)$. It is to be noted that all the functions described above are *static*, meaning that they are *computed only once*.

3.3. Description of the Matching Algorithm

A given line segment may, in general, match with several line segments, based solely on the criterion for computing $p(i, j)$. To assign unambiguous matches, we need to consider sets of matches together. In our method, for each possible match (i, j) we compute an evaluation function $v(i, j)$, described in detail below, which is dependent on how well the disparities of the other line segment matches in the neighborhoods of i and j agree with the average disparity of the matching d_{ij} . We prefer matches with the lowest values of $v(i, j)$; hence the name “minimal differential disparity” algorithm. Intuitively, we are arguing that matches with similar disparities are preferable and correspond to choosing flatter interpretations of a surface, over more jagged interpretations. The function $v(i, j)$ is computed iteratively and changes if the segments in the neighborhood are assigned new *preferred* matches (also defined below).

Computing $v(i, j)$ is rather complex; we first define it formally, then provide an explanation and give an illustrative example.

3.3.1. Formal Description

For each possible pair (i, j) , such that $p(i, j) \neq 0$, we compute $v(i, j)$ as follows:

$$v^{t+1}(i, j) = \sum_{a_h \text{ in } w(b_j)} \min_{b_k \text{ verifies } C_1(a_h)} \lambda_{ijhk} |d_{hk} - d_{ij}| / \text{card}(b_j) \\ + \sum_{b_k \text{ in } w(a_i)} \min_{a_h \text{ verifies } C_2(b_k)} \lambda_{ijhk} |d_{hk} - d_{ij}| / \text{card}(a_i)$$

where $t + 1$ indicates the iteration number, and $t = 0$ for the first iteration; $\lambda_{ijhk} = \min(\text{overlap}(i, j), \text{overlap}(h, k))$, and the overlap function measures the common, overlapped length of the matching segments along the epipolar lines.

Before specifying conditions C_1 and C_2 , it is useful to define a few other terms. A match (i, j) is said to be a *preferred* match at iteration t , if the following holds:

$$\forall k \text{ in } S_p(a_i) \text{ such that } b_k \leftrightarrow b_j, \quad v^t(i, j) < v^t(i, k),$$

and

$$\forall h \text{ in } S_p(b_j) \text{ such that } a_h \leftrightarrow a_i, \quad v^t(i, j) < v^t(h, j).$$

Note that a segment may have more than one preferred match, as long as the alternative candidates do not overlap (this is to allow for the possibility of fragmented segments).

We define a set Q of preferred matches for each segment; thus, if (i, j) is a preferred match, then j is in $Q^t(a_i)$ and i is in $Q^t(b_j)$. At $t = 0$, $Q^0(x) = \emptyset$, for all x .

Now, we can define conditions C_1 and C_2 . We say b_k verifies $C_1(a_h)$ if:

1. If $Q^t(a_h) \neq \emptyset$, b_k is in $Q^t(a_h)$ else b_k is in $S_p(a_h)$,
2. Either $b_k \neq b_j$, or a_h and a_i do not overlap.

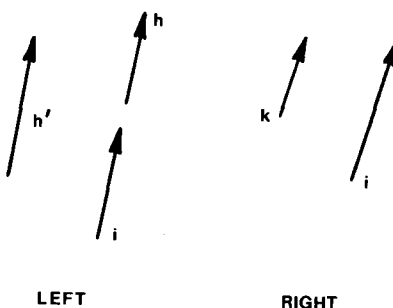


FIG. 4. An acceptable multiple match case (a_h and a_i may match with b_j).

(e.g., in Fig. 4, while evaluating the match (i, j) , h' may match only k , but h may match either k or j , as i and h do not overlap). Similarly, a_h verifies $C_2(b_k)$ if:

1. if $Q'(b_k) \neq \emptyset$, a_h is in $Q'(b_k)$ else a_h is in $S_p(b_k)$,
2. Either $a_h \neq a_i$, or b_k and b_j do not overlap.

3.3.2. Explanation

Simply speaking, the function $v(i, j)$ measures how close the disparity d_{ij} of this match is to the disparities d_{hk} of the segments in the neighborhoods of the segments i and j . For each segment h , in the neighborhood of j , we select the match k that gives the disparity d_{hk} that is closest to that of d_{ij} (and similarly for the reverse match, when a segment k , in the neighborhood of i , is examined). If no match is found for h , then we assign the value $2 \cdot \max d$ to d_{hk} . Note that even though a particular match (h, k) minimizes $|d_{hk} - d_{ij}|$, our selecting the match (i, j) does not imply that (h, k) will also be necessarily selected as a match, though this may usually be expected to be so (i.e., our process is more like “relaxation” than tree or graph search).

In the first iteration, to compute $\min |d_{hk} - d_{ij}|$, we use all segments k that could possibly match h (i.e., k is in $S_p(a_h)$), with the restriction that $b_k \neq b_j$ unless a_h and a_i do not overlap. At any iteration after the first, to compute d_{hk} , if match h has a set of preferred matches $Q(a_h)$, we only match k in this set (the conditions of overlap remain as before). Note that we allow multiple matches for a segment, if the alternatives do not overlap, to allow for matching with fragmented segments.

Finally, the quantities λ_{ijhk} and $\text{card}(a_i)$ are normalization factors used to compensate for the effects of segment length and number of segments in a window.

3.3.3. Termination Criteria

One way to terminate the algorithm would be when $v^{t+1}(i, j) = v^t(i, j)$ for all (i, j) . Unfortunately, our algorithm cannot be guaranteed to have this convergence property and may oscillate between two (or more) alternatives. In practice, we have found three iterations to suffice, even for complex scenes. In the first iteration, we essentially find the preferred matches. The second iteration reevaluates the matches using the preferred matches when available. At the third iteration, we usually find no changes. Our algorithm is stopped after the third iteration in any case. At this time,



FIG. 5. A simple example.

we simply consider as valid the preferred set of matches. Note that the matches are either unambiguous or the alternative candidates are non-overlapping, possibly corresponding to fragmented segments.

3.3.4. An Illustrative Example

Let our 2 views be the ones shown in Fig. 5 above: In absence of any extra information, the correct interpretation is that the 3 points have the same disparity, and the result of the matching is (i, i) for i in $\{1, 2, 3\}$. In this example,

$$S_p(a_i) = S_p(b_i) = \{1, 2, 3\} \quad \text{and} \quad S_{\bar{p}}(a_i) = S_{\bar{p}}(b_j) = \emptyset.$$

The array d_{ij} is

$$\begin{array}{ccc} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{array}$$

Therefore we find

$$v^1(1, 1) = (|d_{22} - d_{11}| + |d_{33} - d_{11}|)/3 + (|d_{22} - d_{11}| + |d_{33} - d_{11}|)/3 = 0$$

compared to

$$v^1(1, 2) = (|d_{23} - d_{12}| + |d_{33} - d_{12}|)/3 + (|d_{21} - d_{12}| + |d_{23} - d_{12}|)/3 = 1$$

and to

$$v^1(1, 3) = (|d_{22} - d_{13}| + |d_{32} - d_{13}|)/3 + (|d_{12} - d_{13}| + |d_{11} - d_{13}|)/3 = 2.67.$$

The calculations are similar for the other pairs, so, at the end of the first iteration, the preferred interpretations are only the correct ones, and further iterations will not alter the results.

3.4. Discussion

The criterion used here, namely the minimal differential disparity, has similarities with the edge interval constraints given in [2] and subsequently used by Baker and Binford [3], but looser in the sense that it does not require ordering of the edges. Since our criterion does not take ordering into account, a dynamic programming implementation is not possible.

To estimate the computational complexity of the algorithm, we consider a simple formulation. Let N be the total number of segments in an image and let n be the number of segments in a window (assumed to be the same for all windows, an average value would give a reasonable answer).

For each segment, we allow an angle tolerance of between 30° and 90° depending on its length. Let us say that the average number of possible matches for each segment is thus $n/3$.

Then, for each a_i , we look at $n/3$ elements in $S_p(a_i)$, then n elements in $w(b_j)$, then $n/3$ elements in $S_p(a_h)$, leading to a total number of matches examined, $C = 2N.n^3/9$. (The factor of 2 is due to our examining left to right, as well as right to left matches.)

Note that the complexity of our computation increases only *linearly* with the size of the image N , if we assume that the number of segments in the disparity window n remains constant. However, if the scene becomes more complex, in the sense that n increases, the computational cost increases as n^3 . For comparison, Baker and Binford's algorithm has a complexity of the order of n^3 for matching one line, where n is the number of edge points on a line.

4. RESULTS AND CONCLUSIONS

We believe that our algorithm performs impressively on a number of complex images shown in this section. However, since our algorithm generates only a sparse disparity map (at the points of the matched segments), it is difficult to display the results in a very convenient form for human viewing. We have a sparse array of 3-D positions—if we display it from the point of view of the camera, we simply get the original line drawings of the matched segments, if we choose a different point of view, we are unable to remove hidden points and the projection is difficult to visualize. Note that this is inherent to the process, and not an artifact of our algorithm. Proper display requires the construction of an interpolating surface, as in [10], however, good interpolation first requires us to solve the object segmentation problem. We believe that these are major problems in their own right and have proposed no solution to them here.

We have chosen to display the matched segments in the left and the right images separately for our output. The information about which segments in one match with which in the other is not provided. We have carefully examined these correspondences one by one, and we provide our summary of what is matched correctly and what is not. We have also chosen to show the right view to the left and the left view to the right, so that those who have mastered the art of “cross-eyed” stereo can perceive them in depth.

We first started our experiments with very simple line drawings, slightly more complex than the one shown in Fig. 5 and the results matched the expectations. In order to remove the effects of the segmentation procedure on the performance of our matching technique, we hand-segmented the images shown in Fig. 6 by tracing the boundaries of the objects on a digitizing table. This image, from Control Data Corporation, is synthetic and has been used by Baker [3]. The resulting segments are shown on Fig. 7, and Fig. 8 displays the results after matching. All the lines that have been matched have the correct correspondence, but some matches are missed. This is due to the fact that when the matcher gets confused by closely competing assignments, it chooses not to assign a label. Also, some edges are not matched because of mistakes in the tracing procedure: we traced the boundaries of some objects in opposite directions in the two views. For all other examples, edge detection was performed automatically using a technique developed by Nevatia and Babu [20] (we used 5×5 masks in 6 directions here).

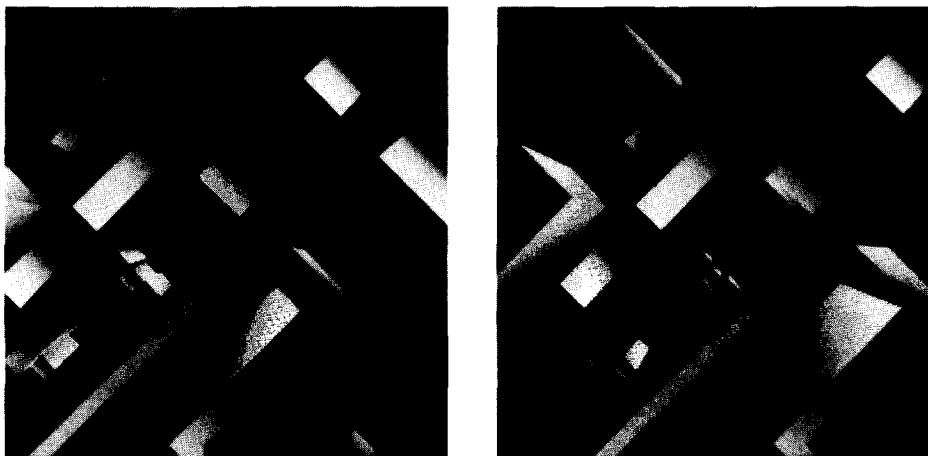


FIG. 6. Synthetic image $[256 \times 256 \times 6]$.

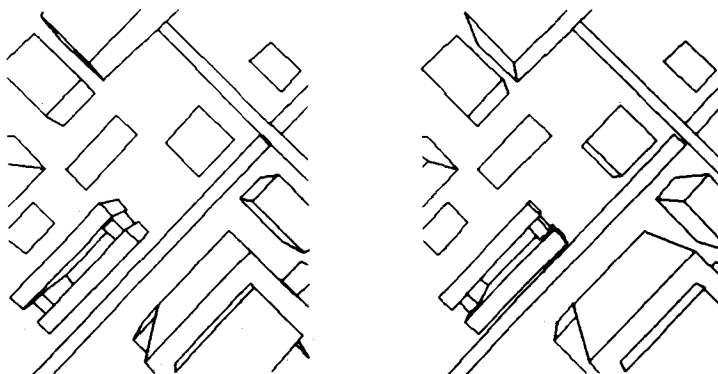


FIG. 7. Hand generated segments.

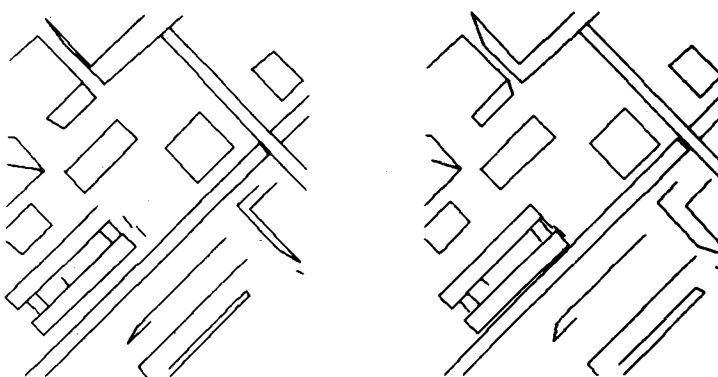


FIG. 8. Results of the matching.

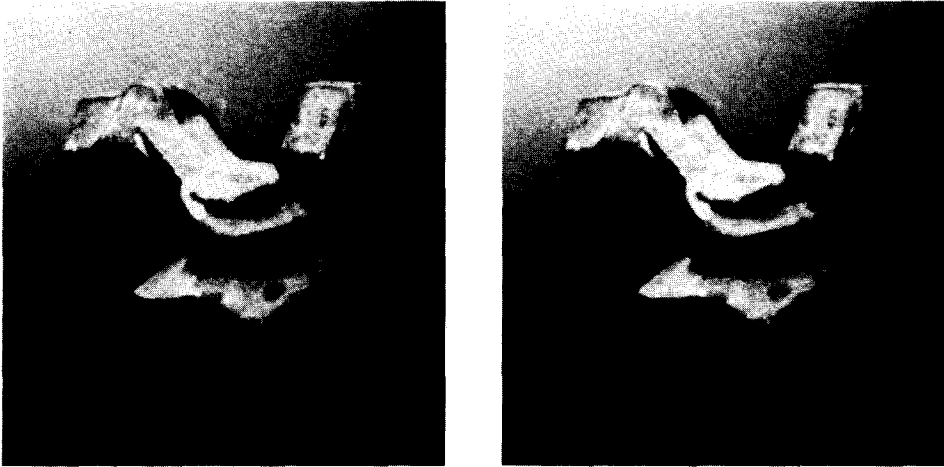


FIG. 9. Industrial part [$256 \times 256 \times 8$].

Consider the industrial part shown in Fig. 9, the original resolution is 256 by 256 and the grey levels are encoded in 8 bits. We applied the matching algorithm to two different resolutions of the image, running it through three iterations. We found that no assignment changed after three iterations in our experiments. Figure 10 shows the original edges, and Fig. 11 displays the results in the above mentioned form. Similarly, Fig. 12 shows the segments at half resolution and Fig. 13 the results. Looking at the segments one by one, we did not notice any incorrect assignment at either resolution, meaning that we captured the shape of the object, even though the density of edges is much larger than in the previous example.

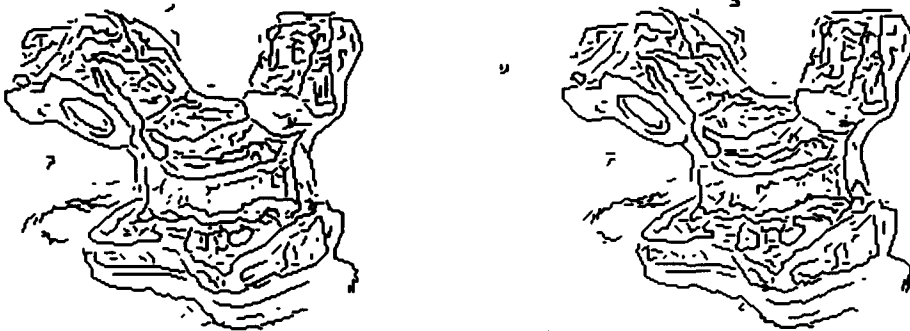


FIG. 10. Segments from the full resolution image.



FIG. 11. Results at full resolution.



FIG. 12. Segments from the half resolution image.

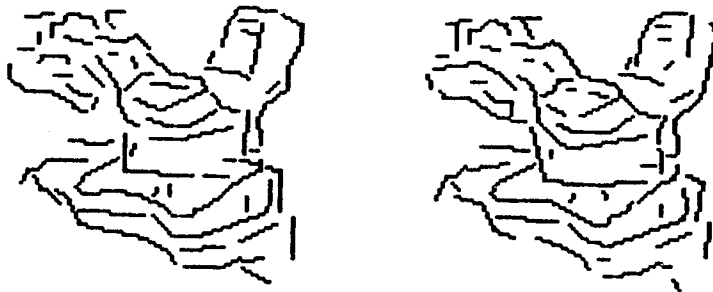


FIG. 13. Results at half resolution.

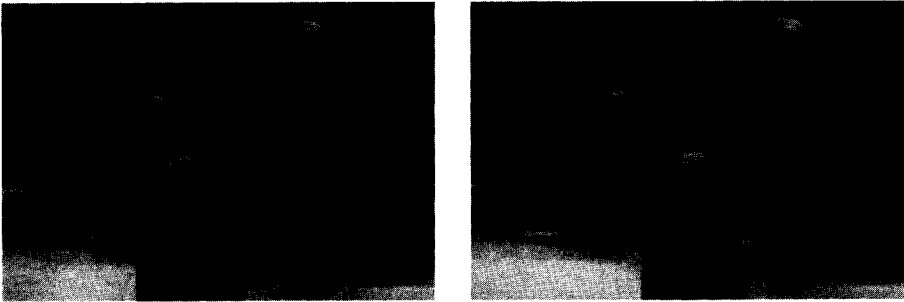
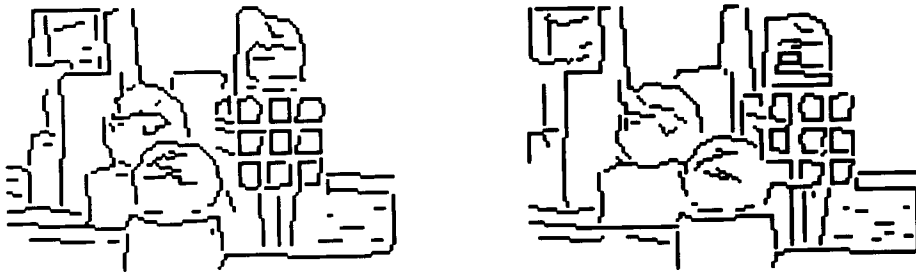
FIG. 14. Image of some blocks $[512 \times 512 \times 7]$.

FIG. 15. Segments at low resolution.

Another, more complex image is shown in Fig. 14. In this image, we have a wide range of disparities, a change of sign in the disparities across the picture, various occlusions, the presence of a repetitive structure (a Rubik's cube) and contrast reversal. We do not expect to get good results with this contrast reversal since one of our preliminary conditions is similarity in contrast, but the other peculiarities are very interesting. We worked at low resolution on the segments shown in Fig. 15 to obtain the results shown in Fig. 16. The interesting points are the following:

—The elongated vertical blocks in the rear of the image are correctly put into correspondence.

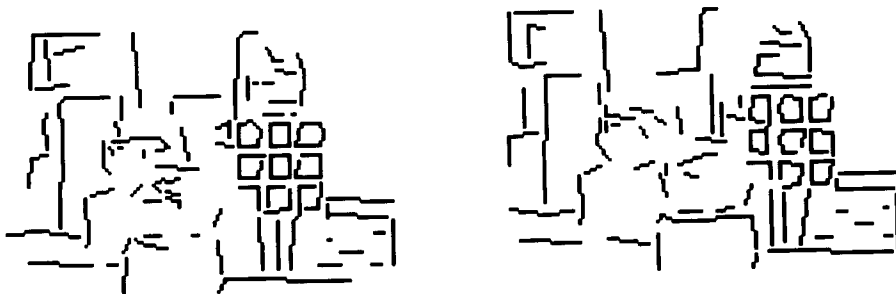


FIG. 16. Results at low resolution.

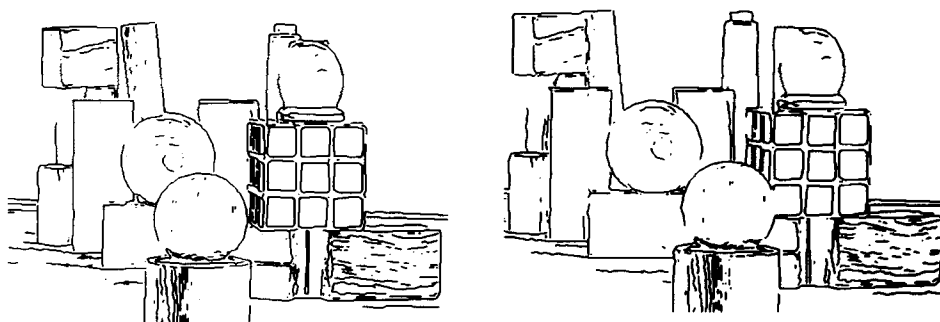


FIG. 17. Segments at high resolution.

—All the squares of the cube that should be identified are correctly matched. The correct labeling appeared at iteration 2 (at iteration 1, most of them are ambiguously matched).

The segments at high resolution are shown in Fig. 17 and the matching results in Fig. 18. We did not use the results at low resolution to guide the matching at high resolution, therefore the elongated block in the rear right is not matched any longer. It is interesting to note that the edges coming from the texture of the wood blocks do not create confusion, but help the matching, on the front cylinder, for example. Once again, most assigned matches are correct.

A program to implement our algorithm was written in SAIL on a PDP-10 running the TOPS-20 operating system. Table 1 presents a summary of the CPU time the program took to process some of the scenes, together with the number of segments in these scenes.

We hope that the above results show the potential of our approach. We believe that our method is successful at matching scenes much more complex than those reported in the literature previously. However, our method does not provide a complete system yet and several improvements to the matching method may be possible. We are actively investigating some of these. Our evaluation function is justified heuristically; a statistical analysis may suggest modifications. Finally, we believe that matching contour alone is inherently limited, and higher level monocular

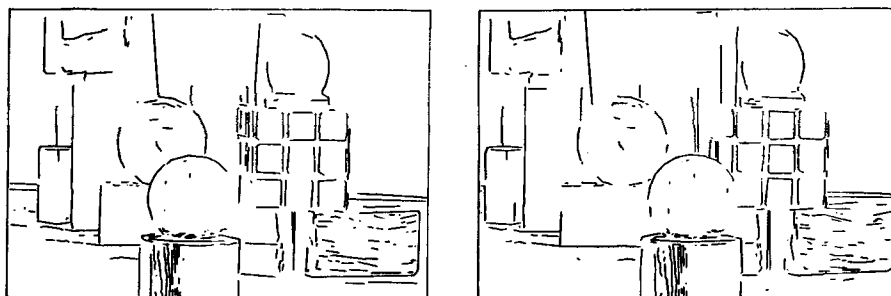


FIG. 18. Results at high resolution.

TABLE 1
Summary of Results

Image name	No. of segments in left	No. of segments in right	Maximum disparity (in pixels)	Time (in s)
Synthetic image	97	97	50	175
Industrial part (high resolution)	638	666	15	475
Blocks (high resolution)	744	760	45	1795
Blocks (low resolution)	172	191	20	99

cues are likely to be needed for complex scenes, but that line matching as suggested in this paper may constitute a major first step of such a system.

REFERENCES

1. D. Arnold, Local context in matching edges for stereo vision, in *Proc. Image Understanding Workshop*, Cambridge, Mass., May 1978, 65–72.
2. D. Arnold and T. Binford, Geometric constraints in stereo vision: Image processing for missile guidance, *Proc. Soc. Photo-Opt. Instrum. Engr.* **238**, 1980, 281–292.
3. H. Baker, *Depth from Edge and Intensity Based Stereo*, Stanford Artificial Intelligence Laboratory AIM 347, Stanford, Calif., Sept. 1982.
4. H. Baker and T. Binford, A system for automated stereo mapping in *Proc. Image Understanding Workshop*, Palo Alto, Calif., 1982, 215–222.
5. S. Barnard and W. Thompson, Disparity analysis of images, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-2**, 1980, 333–340.
6. S. Barnard and M. Fishler, Computational stereo, *ACM Computing Surveys* **14**, No. 4, 1982, 553–572.
7. P. MacVicar-Whelan and T. Binford, Line finding with subpixel precision, in *Proc. Image Understanding Workshop*, Washington, D.C., Apr. 1981, 26–31.
8. D. Gennery, Object detection and measurement using stereo vision, in *Proc. Image Understanding Workshop*, College Park, Md., Apr. 1980, 161–167.
9. W. Grimson and D. Marr, A computer implementation of a theory of human stereo vision, in *Proc. Image Understanding Workshop*, Palo Alto, Calif., Apr. 1979, 41–47.
10. W. Grimson, *From Images to Surfaces*, MIT Press, Cambridge, Mass., 1981.
11. M. Hannah, Bootstrap stereo, in *Proc. Image Understanding Workshop*, College Park, Md., Apr. 1980, 201–208.
12. R. Henderson, R. Miller, and C. Grosch, Automatic stereo reconstruction of man-made targets: Digital processing of aerial images, *Proc. Soc. Photo-Opt. Instrum. Engr.* **186**, No. 6, 1979, 240–248.
13. R. Kelly, P. McConnell, and S. Mildenerger, The Gestalt photomapping system, *Photogrammetric Eng. Remote Sens.* **43**, No. 1407, 1977.
14. B. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, in *Proc. Image Understanding Workshop*, Washington, D.C., Apr. 1981, 121–130.
15. D. Marr and T. Poggio, Cooperative computation of stereo disparity, *Science* (Washington, D.C.) **194**, 1976, 283–287.
16. D. Marr and T. Poggio, *A Theory of Human Stereo Vision*, Artificial Intelligence Laboratory (Memo. 451), MIT, Cambridge, Mass., Nov. 1977.
17. D. Marr and E. Hildreth, Theory of edge detection, *Proc. Roy. Soc. London B* **207**, 1980, 187–217.

18. H. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," Stanford Artificial Intelligence Laboratory AIM 340; Ph.D. thesis, Stanford University, Sept. 1980.
19. R. Nevatia, Depth measurement by motion stereo, *Comput. Graphics Image Process.* **5**, 1976, 203–214.
20. R. Nevatia and K. Babu, Linear feature extraction and description, *Comput. Graphics Image Process.* **13**, 1980, 257–269.
21. R. Nevatia, *Machine Perception*, Prentice-Hall, Englewood Cliffs, N.J., 1982.
22. Y. Ohta and T. Kanade, "Stereo by intra and inter-scanline search using dynamic programming," Technical Report CMU-CS-83-162, October 1983.
23. D. Panton, A flexible approach to digital stereo mapping, *Photogramm. Eng. Remote Sens.* **44**, No. 12, 1978, 1499–1512.