and then the line invariant (6) was computed for each pair of line sets and compared using the metric (7).

The sets of lines chosen are given in the following table (refer to Fig. 1).

$$S_1 = \{B, C, J, K\}, \quad S_2 = \{B, G, J, N\}$$
$$S_3 = \{A, B, H, I\}, \quad S_4 = \{B, D, E, G\}$$
$$S_5 = \{A, C, O, J\}, \quad S_6 = \{B, I, L, N\}$$

The table in (9) shows the results. The only bad entry in this matrix is in the position (4, 4). This is because of the fact that the four lines chosen contained three coplanar lines (lines B, D and E). This causes the values of the invariant to be indeterminate (that is (0, 0, 0)), and shows that such instances must be detected and avoided.

$$\begin{array}{cccccc} \mathbf{0.0129} & 0.6741 & 0.3027 & 0.6885 & 0.6425 & 0.4494 \\ 0.6469 & \mathbf{0.0338} & 0.7414 & 0.8382 & 0.7069 & 0.2216 \\ 0.0620 & 0.6912 & \mathbf{0.2292} & 0.7075 & 0.7082 & 0.4613 \\ 0.2866 & 0.6076 & 0.1823 & \mathbf{0.8903} & 0.8558 & 0.3839 \\ 0.6566 & 0.7212 & 0.8996 & 0.7189 & \mathbf{0.0035} & 0.6943 \\ 0.4731 & 0.2390 & 0.5552 & 0.9479 & 0.7192 & \mathbf{0.0332} \end{array} \tag{9}$$

Once again, the four-line invariant is shown to be a useful discriminator between sets of four lines.

ACKNOWLEDGMENT

I am indebted to J. Mundy for introducing me to the subject of projective invariants, and for many enlightening conversations during the preparation of this correspondence.

REFERENCES

[1] K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. New York: John Wiley, 1989.

[2] C. Coelho, A. Heller, J. Mundy, D. Forsyth, and A. Zisserman, "An experimental evaluation of projective invariants," in J. L. Mundy and A. Zisserman, Eds., *Geometric Invariance in Computer Vision*Boston, MA:MIT Press, 1992, pp. 87–104.

[3] O. D. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig," in *Computer Vision—ECCV '92, LNCS-Series*, vol. 588. New York: Springer-Verlag, 1992, pp. 563–578.

[4] R. Hartley, R. Gupta, and T. Chang, "Stereo from uncalibrated cameras," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, 1992, pp. 761–764.

[5] R. I. Hartley, "Estimation of relative camera positions for uncalibrated cameras," in *Computer Vision—ECCV '92, LNCS-Series*, vol. 588. New York: Springer-Verlag, 1992, pp. 579–587.

[6] ——, "Invariants of Lines in Space," in *Proc. DARPA Image Understanding Workshop*, pp. 737–744, 1993.

[7] J. B. Burns, R. S. Weiss, and E. M. Riseman, "The non-existence of general-case view-invariants," in *Geometric Invariance in Computer Vision*, J. L. Mundy and A. Zisserman, Eds. Boston, MA: MIT Press, 1992, pp. 120–134.

[8] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.

[9] S. J. Maybank, "The projective geometry of ambiguous surfaces," *Phil. Trans. Roy. Soc. Lon. A*, vol. 332, pp. 1–47, 1990.

[10] J. L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision*. Boston, MA: MIT Press, 1992.

[11] J. Ponce, T. Cass, and D. Marimont, "Relative stereo and motion reconstruction," Univ. of Illinois Tech. Rep., number UIUC-BI-AI-RCV-93-07, Nov. 1993.

[12] W. H. Press, B. P. Flannery, S. A. Teukolsky, and William T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, MA: Cambridge Univ. Press, 1988.

[13] L. Quan, "Invariants of 6 points from 3 uncalibrated images," in *Computer Vision—ECCV'94, LNCS-Series*, vol. II. New York: Springer–Verlag, 1994, pp. 459–470.

[14] C. C. Slama, Ed., *Manual of Photogrammetry*, fourth ed. Falls Church, VA: Amer. Soc. of Photogrammetry, 1980.

[15] R. Sturm, "Das Problem der Projectivität und seine Anwendung auf die Flächen zweiten Grades," *Math. Ann.*, vol. 1, pp. 533–574, 1869.

[16] A. Zisserman, D. A. Forsyth, J. L. Mundy, and C. A. Rothwell, "Recognizing general curved objects efficiently," in *Geometric Invariance in Computer Vision*, J. L. Mundy and A. Zisserman, Eds. Boston, MA: MIT Press, 1992, pp. 228–251.

Matching Point Features with Ordered Geometric, Rigidity, and Disparity Constraints

Xiaoping Hu and Narendra Ahuja

***Abstract*— This correspondence presents a matching algorithm for obtaining feature point correspondences across images containing rigid objects undergoing different motions. First point features are detected using newly developed feature detectors. Then a variety of constraints are applied starting with simplest and following with more informed ones. First, an intensity-based matching algorithm is applied to the feature points to obtain unique point correspondences. This is followed by the application of a sequence of newly developed heuristic tests involving geometry, rigidity, and disparity. The geometric tests match two-dimensional geometrical relationships among the feature points, the rigidity test enforces the three dimensional rigidity of the object, and the disparity test ensures that no matched feature point in an image could be rematched with another feature, if reassigned another disparity value associated with another matched pair or an assumed match on the epipolar line. The computational complexity is proportional to the numbers of detected feature points in the two images. Experimental results with indoor and outdoor images are presented, which show that the algorithm yields only correct matches for scenes containing rigid objects.**

***Index Terms*— Disparity, feature detection, geometric consistency, matching, multiresolution, point features, rigidity.**

I. INTRODUCTION

This correspondence is about finding pairs of point features in two perspective views of a scene such that each pair corresponds to the same scene point. The relationship between the viewpoints used to obtain the two images is *unknown*, unlike the case of stereo matching. Further, the objects in the scene may be stationary or moving. The focus of the work reported in this correspondence is to apply a range of constraints and heuristics to determine unique matches. The constraints are applied starting with the simplest and following with more informed ones. The final matches are those that meet all constraints.

The work reported in this correspondence is related to other work on feature matching ([1]–[8], [10], [40]]). Feature matching differs from intensity-based matching ([18], [19]), optical flow based matching ([17], [38], [39]), and (affine) transformation method ([11], [12]) in that correspondences are obtained for feature points only.

To match point features across images, point features need to be detected first in both images. Although the matching algorithm in this correspondence does not rely on any particular feature detector, a continuous and robust detection of distinct point features across images will definitely ensure better matching results. A number of feature detectors are sufficiently good for our matching method,

Manuscript received November 6, 1991; revised May 27, 1994. This work was supported by the National Science Foundation and Defense Advanced Research Projects Agency under Grant IRI-89-02728, and the U.S. Army Advanced Construction Technology Center under Grant DAAL 03-87-K-0006. Recommended for acceptance by Associate Editor Y. Aloimonos.

X. Hu is with Sun Microsystems Computer Corp., 2550 Garcia Avenue, Mountain View, CA 94043 USA; e-mail: xiaoping.hu@Eng.Sun.Com.

N. Ahuja is with the Beckman Institute and the Department of Electrical and Computer Engineering, 405 N. Mathews Avenue, Urbana, IL 61801 USA; e-mail: ahuja@vision.csl.uiuc.edu.

IEEE Log Number 9405242.

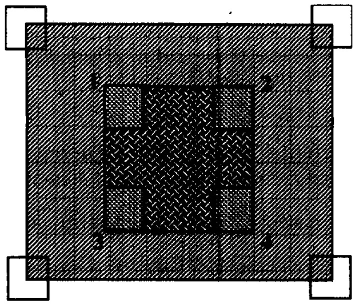


Fig. 1. To match points at depth discontinuity, the window should best contain pixels that belong to the same surface. Therefore, to match point features of type 1, 2, 3, or 4 shown in the figure, the points should be on the upper-left, upper-right, lower-left, and lower-right corner of the window respectively.

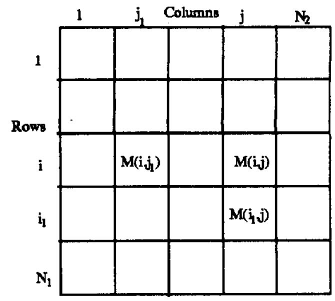


Fig. 2. A pair (i, j) is a candidate match, if and only if: 1). $M(i, j) < \delta_1$; 2). $M(i, j)$ is the minimum inthe ith row and jth column in matrix $\mathbf{M}$; 3). $M(i, j)$ is smaller than the second best match $M(i, j_1)$ in the ith row by an amount δ_2; 4) $M(i, j)$ is smaller than thesecond best match $M(i_1, j)$ in the jth column by an amount δ_2.

among which we recommend the algorithms of Deriche and Giraudon ([8]), Moravec ([3], [4]), and Hu ([35]). In particular, the features in the experiments of this correspondence are detected using the improved Moravec interest operator ([35], [36]).

The matching problem is approached by finding two subsets of points each from a set of detected feature points in a different image such that the two subsets of points are locally intensity-matched and preserve the interpoint geometrical structure. As is true for all matching algorithms, the underlying principle for matching is geometrical similarity and/or intensity (or color) similarity. It appears that geometrical similarity is more fundamental and stable than intensity similarity since intensities are more liable to change. Nevertheless, intensities add more information about each point and hence should be used whenever applicable. On the other hand, it is computationally intractable to use geometric constraint alone to match point features since the number of combinations that a set of points yields grows exponentially with the number of points. Therefore, a combined use of intensity information and geometric constraints is more promising than using intensity or geometric constraints alone. This correspondence presents an algorithm which makes combined use of both.

II. The Matching Algorithm

The algorithm consists of five stages I–V, and is applied repeatedly to images of increasingly fine resolution, starting with the coarsest resolution. These stages are:

I) *Point Feature Detection:* detect distinct point features.
II) *Intensity Based Matching:* obtain candidate matches based on similarity of local intensities.
III) *Geometry Based Elimination:* identify and remove points that do not preserve two dimensional geometric consistency in the image plane.
IV) *Rigidity Based Elimination:* eliminate such points that do not construct a rigid triangle with their two closest neighbors.
V) *Disparity Based Elimination:* eliminate all point pairs for which one of the points could be rematched and yield a disparity value possessed by some other matched pair.

The following sections describe the motivation and details of these stages.

A. Point Feature Detection

The matching algorithm does not depend on any particular feature detectors. However, a robust detection of distinct point features across images is necessary for robust matching results. Among the feature point detectors have been reported in the literature ([1]–[5], [7]–[9]), we recommend the algorithms of Deriche and Giraudon ([18]), Moravec ([3], [4]), and Hu ([35]), which are sufficiently good for the matching method of this correspondence. In the experiments of this correspondence, the features are detected using the improved Moravec interest operator (see [7] for detail).

B. Intensity Based Matching

After a number of feature points are detected in each image, the next goal is to find a subset of the points from each image such that they correspond to the same scene points. This is done in several steps. The first step is to obtain correspondence pairs which have similar intensity environments. The method below applies to images having small local distortions.

Given the point features detected in each of the two images, a *difference measure* is computed for all possible feature pairs over a window W. Let $I_i(x, y)$ be the image intensity at the ith grid point and $\underline{I}_i$ be the average intensity over a window W centered at i. Let $P_i(x, y)$ denote the intensity for $(x, y) \in W$ after the window mean is subtracted, i.e.,

$$P_i(x, y) = I_i(x, y) - \underline{I}_i, (x, y) \in W. \quad (1)$$

Then, the difference measure $M(i, j)$ for two pixels is defined as the sum of absolute differences of corresponding values in $P_i(x, y)$ in the first image and $P_j'(x, y)$ in the second image:

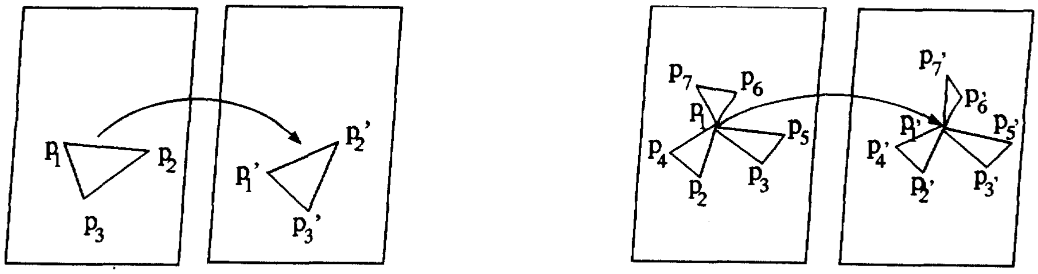
$$M(i, j) = \frac{1}{n} \sum_{(x,y)\in W} |P_i(x, y) - P_j'(x, y)|, \quad (2)$$

where n is the number of points in the window. To allow rotations around the optical axis across images, we find the orientation of the second window such that the difference measure for the two windows is minimized. That is, we rotate $P_j{}'(x, y)$ such that $M(i, j)$ is minimized.

To allow matching of points close to occlusion boundaries, matching of other windows not centered at those points, is also attempted. The value of M is computed for four additional types of windows, with the point located at the four corners, respectively. Each of these four windows responds the best to corners of a specific orientation, shown in Fig. 1. Thus, a total of five windows are used at each point to adapt to different environments. The smallest value of M obtained from the five windows represents the quality of match for a given pair. In general, more windows give more robust performance, at the cost of more computation.

Therefore the matching process is divided into two steps: given a point p in the first image, first search for the orientation of the window around a candidate point q in the second image to get the best match between p and q; then search all candidate points in the second image to get the best overall match for p.

Two thresholds δ_1 and δ_2 are used to select the best and unique matches. Assume there are N_1 points in the first view and N_2 points



(a) (b)

Fig. 3. (a) A feature point p_1 is considered as having a valid match p_1' if the triangle constructed by p_1 and its two closest neighbor features p_2 and p_3 is similar to the triangle constructed by their correspondences p_i', $i = 1, 2, 3$. p_1 provides supports for p_2 and p_3. Or, (b) if p_2 and p_3 all have valid matches p_2' and p_3' and p_1 is among the two closest neighbors of both p_2 and p_3, then p_1 is considered as having a valid match, even if p_1 has a bad neighbor p_7 which has a wrong match.

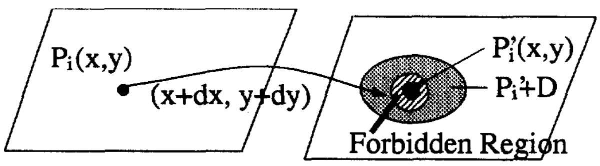


Fig. 4. Disparity Test: if $p_i(x, y)$ and $p_i'(x', y')$ is a candidate matched pair, then for each disparity vector $\vec{d}_j = (dx, dy)$ in the disparity vector set D such that $p_i + \vec{d}_j$ falls out of the forbidden region of $p_i'(x', y')$, we determine if p_i has a good intensity-based match with $p_i + \vec{d}_j$. If so, p_i has more than one match and the pair (p_i, p_i') is removedas a spurious match; if not, pair (p_i, p_i') is retained.

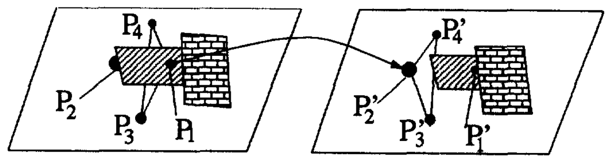


Fig. 5. Only when all of the following four conditions are satisfied, can a mismatched pair p_1 and p_2' survive the rigidity tests and disparity test: (1) the correct match p_1' of p_1 is undetected in the second image; (2) the correct match p_2 of p_2' is undetected in the first image; (3) the local intensities of p_1 matches the local intensities of p_2'; and (4) thetriangle constructed by p_1 and its two closest neighbors p_3 and p_4 pass the geometric and rigidity tests with respect tothe triangle constructed by p_2', p_3', and p_4'.

in the second view. The matchedness measures $M(i, j)$ between the ith point in the first view and the jth point in the second view, $i = 1, \cdots, N_1$, $j = 1, \cdots, N_2$, construct a matrix $\mathbf{M}_{N_1 \times N_2}$. The method for choosing a candidate match is described in the caption of Fig. 2.

C. Geometric Tests

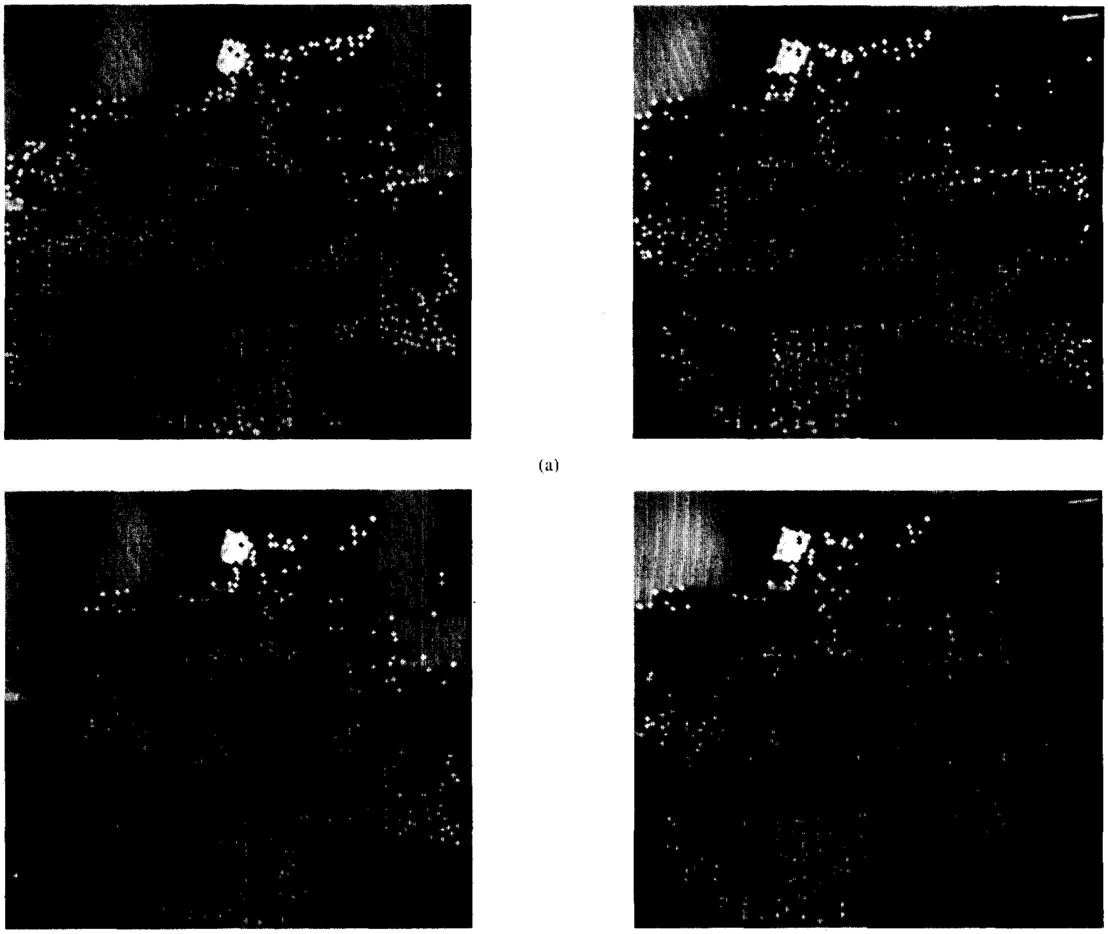
The intensity based matching of Stage II is bound to contain errors in practice. For example, if there are two similar points in the scene such that one is detected in the first image only and the other in the second image only, then mismatching may occur. Even if a scene point is visible in both images, intensities, being variant, are not alone sufficient to find only correct matches. The purpose of this stage is to enforce geometrical consistency across views to further eliminate bad matches.

Ideally matching should be done in the 3-D space. It is best to examine rigidity of all subsets of points so that a pair of matched points correspond to the same scene point. However, in the beginning, the depths cannot be computed because the motion is unknown and the computational complexity forbids us to consider all combinations of points simultaneously since N points yield 2^N combinations. Therefore we start with finding points that preserve local geometric consistency in the 2-D space. One simple way to test for local geometric consistency is to consider only triples of points simultaneously and update the set of matched pairs (the rigid set) recursively. The goal of these tests is to identify those points that have correct matches by considering groups of points together. In the beginning, all points belong to a set called *working set*. As matched dots pass the tests, they are transferred away to another set called *rigid set*.

Test A (Geometric Test): In this test, triples of matched pairs are considered simultaneously (Fig. 3). Each point and its two closest neighbors in the first image construct a triangle. The correspondences of the 3 points in the second image also construct a triangle. When the motion is rigid and small, the distortion of the triangle caused by motion is also small. Therefore, if all of three pairs of correspondences are correct, the two triangles in the image plane must be similar (the similarity of two triangles is examined by computing the proportions of the lengths of the corresponding sides of the triangles). If one or more pairs are wrongly matched, then the similarity can be only accidently preserved. The probability with which three points in the first image wrongly match three points in the second image such that the two triangles are similar is almost zero (see the discussion later on *Disparity Test*). The exact method of Test A is described in the caption of Fig. 3. It is well possible that some correct matches will be eliminated if their nearest neighbors are wrongly matched. This effect is called *bad neighborhood effect*. To achieve higher reliability, more than 3 points can be considered together, although more correct matches could also be eliminated and more computation is needed.

Test B (Geometric Test): If a correctly matched point has a neighbor which is either wrongly matched or lies across an object boundary in another object, then the correctly matched point may also be classified as a bad match by Test A. If we have already had some valid matches, we can use these correspondences to judge if other correspondences are valid or not. In this test, a point is grouped with its nearest two points that have passed Test A to form a triangle. Therefore, the bad neighborhood effect in Test A can be eliminated. Two sets of input data are required for Test B. Let these be denoted by S_W and S_R. S_W is the set of points that are to be checked (working set), while S_R is the set of points that have passed Test A, or the rigid set. For each point p in S_W, we find its two closest neighbors in S_R. We now determine if the triangles constructed by the three points and their correspondences are similar; if yes, p is put into S_R; otherwise p is discarded.

Test C (Geometric Test): In the above tests, the nearest neighbors are defined in terms of image plane distances. Perspective distortion could be a serious problem when the neighboring points in the image plane have large depth differences. This is called the *closest neighborhood effect*. To reduce this effect, the closest neighboring points could be required to have nearly the same disparities. That is,



(a)

(b)

Fig. 6. Example I. (a) Features detected. (b) The result of intensity-based matching.

given a point p of disparity $\vec{d}$, the nearest neighbours of p are chosen as two points closest to p whose disparities are within $(\vec{d}-\vec{\epsilon}, \vec{d}+\vec{\epsilon})$, where $\vec{d}$ and $\vec{\epsilon}$ are vectors in the image plane. The choice of the value of $\vec{\epsilon}$ is discussed in Section III. Test C differs from Test B only in the choice of closest neighbors. This test is to adapt the matching algorithm to images of large depth difference or perspective distortion.

D. Rigidity Test

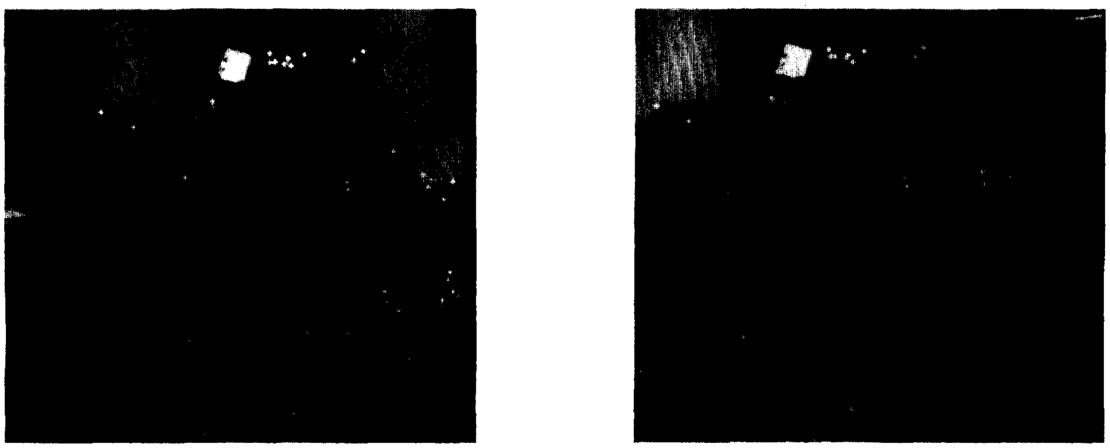
The above geometric tests are based on 2-D similarity and may hence result in error. A more profound constraint is the 3-D rigidity of shape and size. The following test enforces exactly this constraint, but unlike the geometric tests, it applies only when there is a single relative motion between the camera and the scene. However, to compute depths motion parameters are needed, which cannot be estimated without an initial set (≥ 6) of matched points. The geometric tests serve to provide the correspondences needed for motion estimation. Points that preserve 3-D rigidity may not preserve 2-D geometric similarity and vice versa. But only in rare situations does it happen that points preserving 2-D geometric similarity do not preserve 3-D rigidity.

Test D (Rigidity Test): Given two views, we can only construct one 3-D interpretation of the scene, thus it seems that we need three or more views to apply the 3-D rigidity test. However, since all points undergo the same 3-D motion, we compute the depths for the points that have passed tests A, B, and C in the first and second views by using the motion parameters which are estimated from a small number of matches of high reliability. (The algorithms used for motion estimation and depth computation are referred to [9], [35].) Then we test if each 3-D triangle in the first view has the same size as the corresponding 3-D triangle in the second view. If so, the rigidity is preserved; otherwise, not. So this test is essentially the same as Test B except that the depth data are used and the equality of a pair of 3-D triangles instead of the similarity of a pair of planar triangles is examined each time.

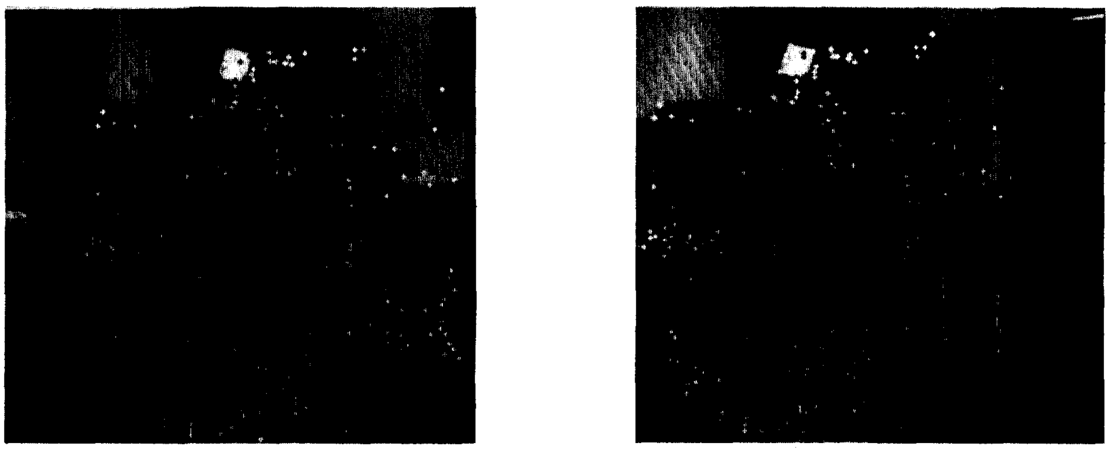
When only two views are used, Test D is equivalent to enforcing the motion epipolar line constraint and the positive depth constraint. Therefore, this test fails for mismatches that satisfy the motion epipolar line equation and yield positive depths. We call this effect as *epipolar effect*, which also exists in stereo matching. If a group of points is mismatched to another group of points that have similar

TABLE I
THE PARAMETERS USED

W	$\vec{\epsilon}$	δ_1	δ_2	γ	ζ
$15\times15 \sim 21\times21$	(5,5)	20	1	0.33	5



(c)



(d)

Fig. 6. Example I. (c) The result after Test A. (d) The result after Test B.

interpoint geometry and satisfy the epipolar line equation, then all above tests may not eliminate the epipolar effect. This occurs for images of periodic patterns or of identical objects (see [34], [35]). One way to eliminate the epipolar effect is to consider more than two views at each time. Another way is to apply the disparity test that follows. However, when a sequence of images is available, Test D is stronger than the union of the epipolar constraint and the positive depth constraint since Test D requires further that the reconstructed depths of the points be consistent from frame to frame.

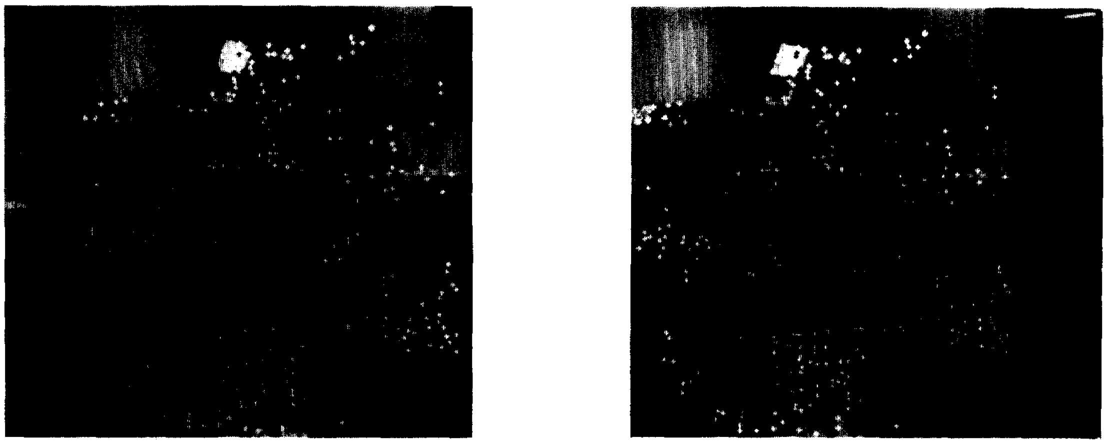
E. Disparity Test

As has been pointed out above, for two views the epipolar effect cannot be overcome by the rigidity test. Therefore, at this stage we still cannot guarantee that all valid matches are correct, although only a small number of mismatches may survive the above tests. The *disparity test* discussed below is intended to remove the remaining mismatches.

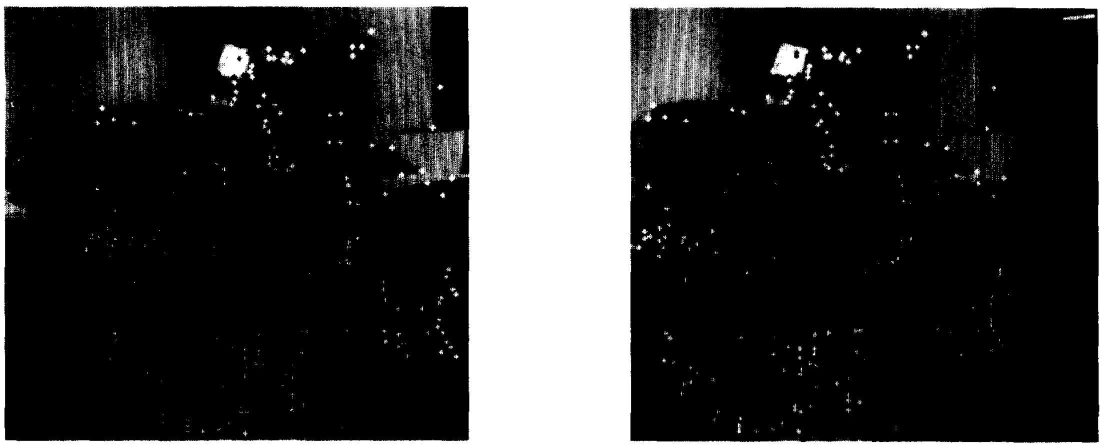
The basic idea is to check if any point in a matched pair can be rematched by another point on the epipolar line which is not detected as a feature point. Ideally, all possible matches on the epipolar line that yield positive depths should be examined, which would result in a considerable amount of computation and possible elimination of correct matches. However, when motion is unknown or when multiple motions are involved, the search space would be too large if one tests for all points in the image plane. Therefore, an efficient method that does not use motion parameters is needed. The disparity test below is designed to limit the search space to the disparities that are possessed by the obtained matches. This method works under the assumption that disparities possessed by the already matched pairs include those possessed by other correct, yet unknown matches.

The disparity test below can also be applied after the motion is known. In this case, the set of *disparity vectors* in the test below should be replaced by all disparities on the epipolar line that yield positive depths. Then all possible matches on the epipolar line that yield positive depths are examined. This rules out all ambiguous matches. However, if wrong matches are used for motion estimation, the estimated motion parameters may be wrong and hence the disparity test may not be effective. Therefore, a robust motion algorithm which gives good results for noisy data is very important. In any case, disparity test will not worsen the results.

Test E (Disparity Test): Given a set of pairs of matches $p_i = (x_i, y_i)$ and $p'_i = (x'_i, y'_i)$, Test E examines them for uniqueness and eliminates any nonunique ones as spurious matches. A set of disparity vectors D is formed by collecting the disparity vectors $\vec{d}_i$ for currently matched pairs p_i and p'_i for all i (if motion is available, D is formed by the disparity vectors of all possible matches on the epipolar line that yield positive depths), where $\vec{d}_i = (x'_i - x_i, y'_i - y_i)$. When the number of matches is small (say, when less than 100 matches are



(e)



(f)

Fig. 6. Example I. (e) The result after Test C. (f) The result of using Test A again.

available in a 512×521 image), D is enlarged by adding neighboring vectors (say in a 3×3 window) of each disparity vector corresponding to the matches. A matched pair p_i and p_i' is accepted if for each $\vec{d}_j \in D$ such that $\|p_i + \vec{d}_j - p_i'\| > \zeta$, (when motion is known, $p_i + \vec{d}_i$ must also lie on the epipolar line defined by p_i), it is true that

$$M\left(p, p + \vec{d}_j\right) - M(p, p') > \delta_2, \tag{3}$$

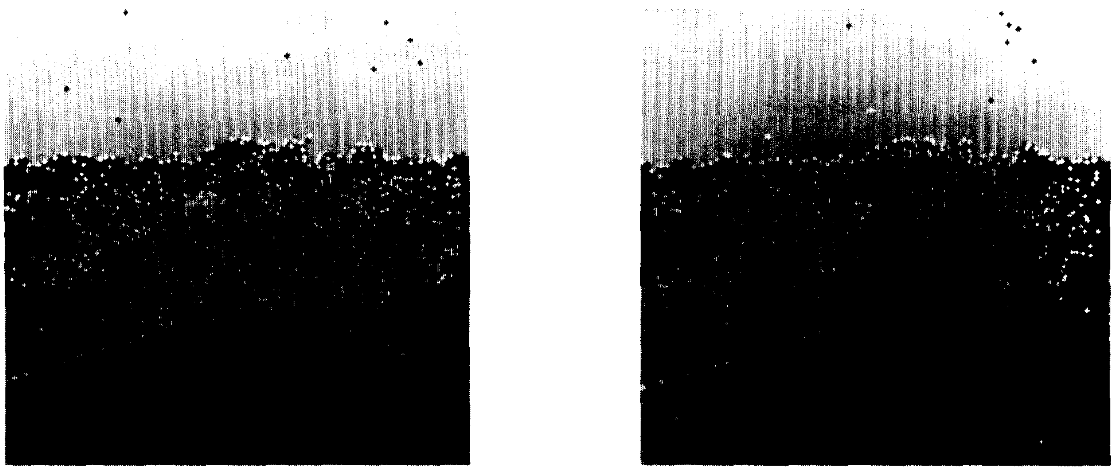
where ζ is a constant denoting the radius of a *forbidden region* surrounding p_i'. The reason for excluding disparity vectors falling within the forbidden region in Test E is that when p_i matches p_i', the difference measure for p_i and any point near p_i' will also be low. This method is shown in Fig. 4. Again, this test is done in both directions: from the first image to the second image and vice versa. For images related by single, known motion, the disparity space reduces to a line segment along the motion epipolar line.

After the above tests have been applied, only mismatches in rare situations may survive, which are described in Fig. 5. If a point p_1 with true correspondence p_1' is mismatched to p_2' whose true correspondence is p_2, then according to the feature detection algorithm, p_1' and p_2' must be separated by some distance and therefore do not fall within each other's forbidden region. Unless both p_1' and p_2 are undetected as point features (e.g., when p_1' and p_2 are occluded in their own images), the pair p_1 and p_2' are unlikely to survive the intensity-based matching. Thus, as long as $(p_1' - p_1)$ is in the disparity vector set, the pair p_1 and p_2' would not survive the disparity test. Even when the pair does survive, to escape the geometric Test A, (p_1, p_2') must preserve the triangle similarity with their closest neighbor points. The probability for this to occur is essentially zero, irrespective of whether the images contain regular patterns. If more than 3 points are tested at a time for geometric consistency, mismatches may be completely eliminated (although more good matches may also be eliminated due to perspective distortion).

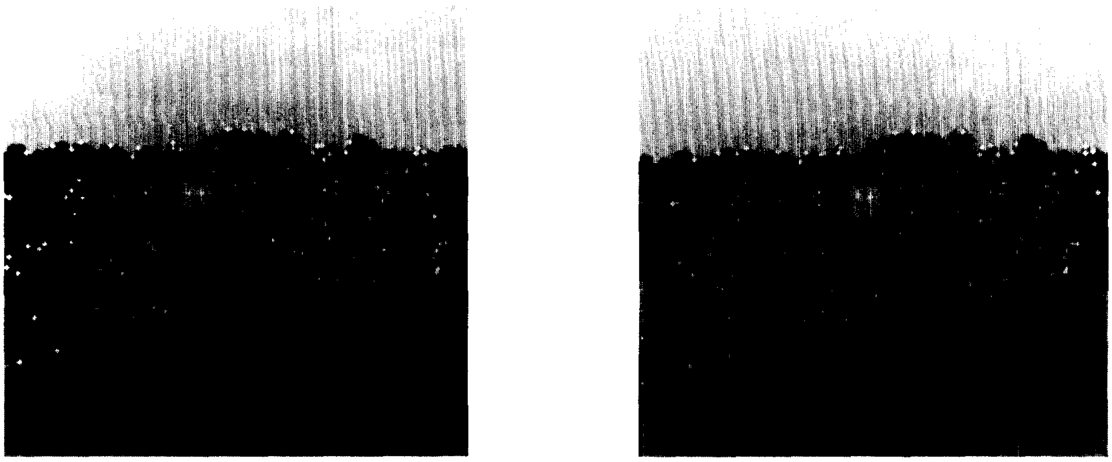
Test E can be used twice, once before the motion is solved for and once after the motion is solved for. At each time, the disparity space D has different contents as described above.

F. The Stepwise Algorithm

The essence of the algorithm consists of the tests which must be arranged in logical order to function properly. Test A is used to start rigidity based testing since it does not require valid matches as references. Tests B and C need reference points. Test D needs knowledge of motion parameters. To achieve better performance, quadruples or larger sets of points can be used by each test. After Tests B and C are applied, generally speaking, more point correspondences will be found. However, it is possible that some newly obtained correspondences will not satisfy the local geometric constraints and are hence wrong matches. This situation occurs very often near the



(a)



(b)

Fig. 7. Example II. (a) Point features detected. (b) The result of intensity-based matching.

occlusion boundary, especially for views involving multiple objects of different motions. Therefore, it is necessary to apply Test A or D again to the matches obtained by Test B and/or Test C.

In the stepwise algorithm below, W_1 and W_2 are the two working sets which are the input for each step, and R_1 and R_2 are two rigid sets containing points that are the outcome of each step. Both working sets and rigid sets are updated constantly.

Step 1: Cross-match the detected feature points in View 1 with those in View 2 based on intensities. Assume W_1 and W_2 are the sets of points of unique matches in View 1 and View 2, respectively.

Step 2: Now apply Test A to matches given by W_1 and W_2 and those pairs which pass the test either in View 1 or View 2 are moved into R_1 and R_2, respectively. Test A is repeatedly applied to points in W_1 and W_2 until no more good pairs can be obtained.

Step 3: Test the points in W_1 with reference to the points in R_1 and points in W_2 with reference to the points in R_2 using Test B. Again, matched pairs that pass the test in either view are moved into R_1 and R_2 respectively.

Step 4: This step is similar to Step 3 except that Test C is used.

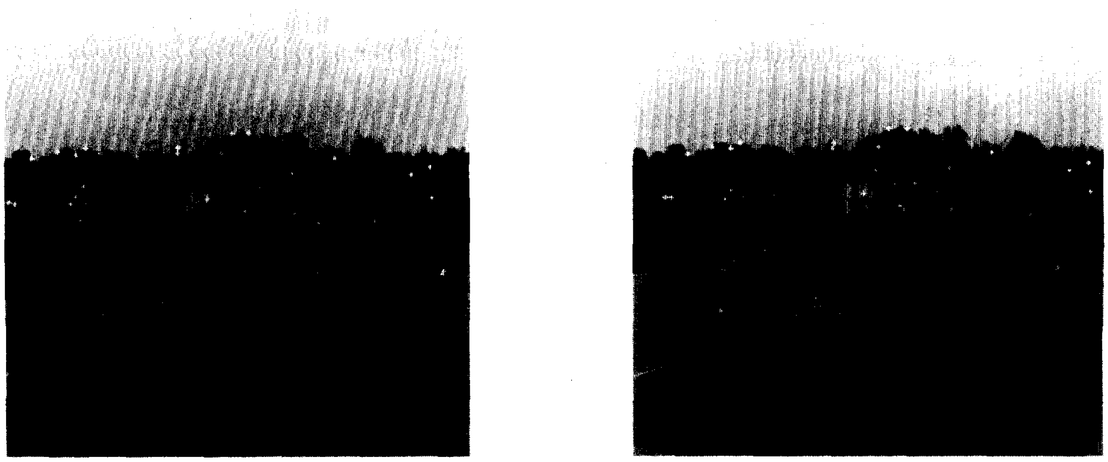
Step 5:(Now apply Test D (If single relative motion) or A (if multiple relative motions) to R_1 and R_2. A pair (p, q) is removed from (R_1, R_2) if either p fails the test in R_1 or q fails in R_2.

Step 6: Apply the (Disparity) Test E to R_1 and R_2. If motion is unknown, the set of disparity vectors D is obtained from the correspondence data in R_1 and R_2. If motion is known, then for each point in concern, D consists of all possible disparities defined by the pixels on the epipolar line that yield positive depths. Then a pair (p, q) is removed from (R_1, R_2) if for any disparity vector $\vec{d}$ in D such that $\|p + \vec{d} - q\| > \zeta$, either p matches point $p + \vec{d}$ or q matches point $q - \vec{d}$. The remaining pairs in (R_1, R_2) comprise the results of the algorithm.

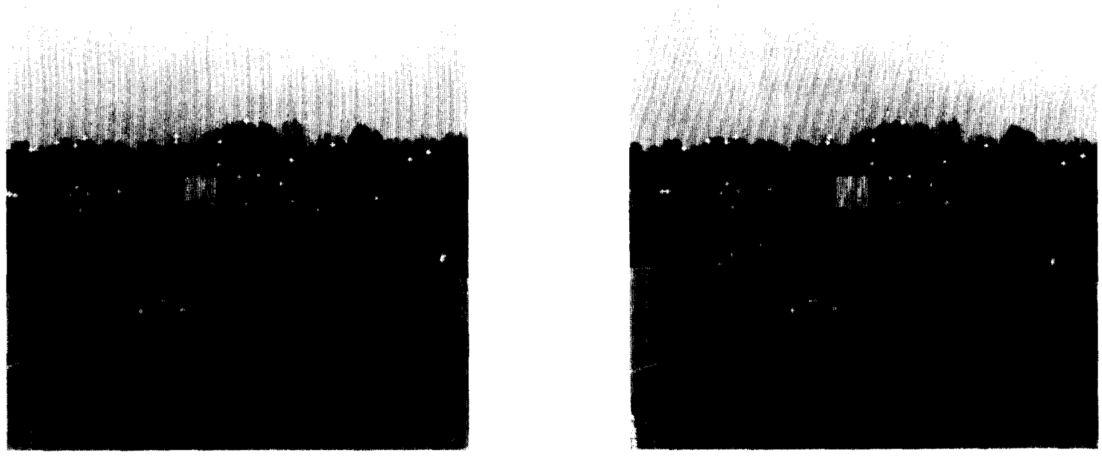
III. Implementation and Experimental Results

In this section, we describe some experiments with real image data while also presenting implementation details through the examples. The parameters used in the algorithm are given in Table I. As long as the parameters are in a reasonable range, the algorithm is not sensitive to the choice of them. Typically, on a SUN Sparc station, the intensity-based matching algorithm takes a few minutes to 20 minutes to match two 512×512 images containing 1,000 points. The geometric and rigidity tests take negligible time and the disparity test takes a few minutes.

Example I: The first example illustrates the effect of the rigidity tests. Fig. 6(a) shows about 500 points detected by the point feature



(c)



(d)

Fig. 7. Example II. (c) The result after geometric and rigidity tests. (d) The final result after disparity test.

detector in two indoor images resulting from a single, rigid motion. In these images, only corners form good point features. Fig. 6(b) shows the matched pairs obtained by the intensity-based matching. We see that there are some mismatches near the vertical boundaries of the images. However, after applying Test A, only a small number of correspondences survive, as shown in Fig. 6(c). More correct correspondences are obtained after using Test B, as shown in Fig. 6(d). Fig. 6(e) shows that even more matches are obtained by using Test C. Fig. 6(f) shows the final matches. Some good matches are also removed along with the bad matches. The disparity test is useless here since no mismatches survive over the geometric and rigidity tests.

Example II: The second example (the original images were provided by Dr. M. Leung and Prof. T. S. Huang) shows the performance of the matching algorithm on images of two objects undergoing different motions (a truck moves in the background and the camera moves accordingly to track the truck). In this example, the disparity test plays a significant role in removing mismatches occurring in similar objects. Fig. 7(a) shows the about 500 points detected by the point feature detector. Fig. 7(b) shows the intensity-matched pairs. There are many mismatches in the background and the truck. The geometric tests remove all but three mismatches which are on the tire as shown in Fig. 7(c). It is interesting to note that the three points are located on different tires in the two images but preserve the local geometrical relations. It is hard to determine that they are mismatches by examining the local region. However, the disparity test successfully, and easily, identifies the matches as ambiguous and removes them from the set of matched pairs, as shown in Fig. 7(d).

IV. Summary

The matching algorithm in this correspondence uses newly developed geometric tests, rigidity test and disparity test to obtain unique point correspondences in two different views of a scene containing rigid objects. We have applied the algorithm to a variety of images and obtained good results (cf. [35]–[37] for more data). The matches found are reliable, although possibly small in number. Therefore, they allow reliable estimation of motion parameters.

The algorithm yields only correct matches under the following assumptions: 1) a fair number ($\geq$ 3) of *distinct* point features corresponding to the same scene points can be detected in both images; 2) local intensity change and geometric distortion from view to view is small enough to allow initial matching using intensity-based or token based methods and to allow the use of geometric tests; 3) for the rigidity test, at least 6 pairs of initial matches from a single, rigid object must be available; 4) for the disparity test, either the true disparities of every mismatched pair is possessed by other correct matches or the motion parameters are accurate enough to allow the use of the motion epipolar lines.

REFERENCES

[1] O. A. Zuniga and R. M. Haralick, "Corner detection using the facet model," in *Proc. Comput. Vision Pattern Recognition7*, 1983, pp. 30–3.

[2] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognit. Lett.*, vol. 1, pp. 95–102, 1982.

[3] H. P. Moravec, "Towards automatic visual obstacle," in *Proc. Int. Joint Conf. on Artificial Intell.*, 1977, p. 584.

[4] H. P. Moravec, "Visual mapping by a robot rover," in *Proc. Int. Joint Conf. Artificial Intell.*, 1979, pp. 598–600.

[5] A. Singh and M. Shneier, "Grey level corner detection: a generalization and a robust real time implementation," *Comput. Vision, Graphics, Image Processing*, vol. 51, pp. 54–69, 1990.

[6] I. K. Sethi and R. Jain, "Finding trajectories of feature points in a monocular image sequence," *Pattern Anal. Machine Intell.*, vol. 9, no. 1, pp. 56–73, Jan. 1987.

[7] P. R. Beaudet, "Rotationally invariant image operators," in *Proc. Int. Joint Conf. Pattern Recognit.*, 1978, pp. 573–583.

[8] R. Deriche and G. Giraudon, "Accurate corner detection: An analytical study," in *Proc. Int. Joint Conf. Comput. Vision*, 1990, pp. 66–70.

[9] A. Rattarangsi and R. T. Chin, "Scale-based detection of corners of planar curves," in *Proc. Int. Joint Conf. Pattern Recognit.*, 1990, pp. 923–930.

[10] C.-L. Cheng and J. K. Aggarwal, "A two-stage hybrid approach to the correspondence problem via forward-searching and backward-correcting," in *Proc. Int. Joint Conf. Pattern Recognit.*, 1990, pp. 173–179.

[11] Y. Lamdan and J. T. Schwartz, and H. J. Wolfson, "Object recognition by affine invairiant matching," in *Proc. Int. Joint Conf. Pattern Recognit.*, 1988, pp. 335–344.

[12] M. S. Costa and R. M. Haralick, and L. Shapiro, "Optimal affine-invariant point matching," in *Proc. Int. Joint Conf. Pattern Recognit.*, 1990, pp. 233–236.

[13] J. R. Beveridge and R. Weiss, and E. M. Riseman, "Combinatorial optimization applied to variable scale 2-D model matching," in *Proc. Int. Joint Conf. Pattern Recognit.*, 1990, pp. 18–22.

[14] X.-J. Wang, J. Fu, and L. Wu, "A matching algorithm based on hierarchical primitive structure," in *Proc. Int. Joint Conf. Pattern Recognit.*, 1990, pp. 285–287.

[15] W. K. Gu, J. Y. Yang, and T. S. Huang, "Matching perspective views of a polyhedron using circuits," *Pattern Anal. Machine Intell.*, vol. 9, no. 3, pp. 390–400, Mar. 1987.

[16] J. Canny, "A computational approach to edge detection," *Pattern Anal. Machine Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[17] B. K. P. Horn and B. G. Schunk, "Determining optical flow," *Artifial Intell.*, vol. 17, pp. 185–203, 1981.

[18] J. Weng, "A theory of image matching," in *Proc. Int. Joint Conf. Comput. Vision*, 1990, pp. 200–209.

[19] J. Weng, N. Ahuja, and T. S. Huang, "Two-view matching," in *Proc. Int. Joint Conf. Comput. Vision*, 1988, pp. 64–73.

[20] W. E. L. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robotics Res.*, vol. 3, no. 3, pp. 3–35, 1984.

[21] ——, "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 4, pp. 469–482, 1987.

[22] W. E. L. Grimson, "Recognition of object families using parameterized models," in *First Int. Conf. Comput. Vision*, 1987, pp. 93–101.

[23] ——, "The combinatorics of object recognition in cluttered enviroments using constrained search," in *Second Int. Conf. Comput. Vision*, 1988, pp. 218–227.

[24] ——, "The effect of indexing on the complexity of object recognition," in *Third Int. Conf. Comput. Vision*, 1990, pp. 644–651.

[25] D. Huttenlocher and S. Ullman, "Object recognition using alignment," in *First Int. Conf. Comput. Vision*, 1987, pp. 102–111.

[26] P. Flynn and A. Jain, "Bonsai: 3D object recognition using constrained search," in *Third Int. Conf. Comput. Vision*, 1990, pp. 263–267.

[27] R. G. Gonzalez and P. Wintz, *Digital Image Processing*. New York: Addison-Wesley , 1977.

[28] D. Marr, *Vision*. New York: W. H. Freeman, 1982.

[29] D. Marr amd T. Poggio, "A computational theory of human stereo vision," in *Proc. Roy. Soc. Lon.*, vol. B, no. 204, pp. 301–328, 1979.

[30] W. Hoff and N. Ahuja, "Surfaces from stereo: integrating feature matching, disparity estimation and contour detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 121–136, Feb. 1989.

[31] W. Hoff and N. Ahuja, "Surfaces from stereo images: an integrated approach," in *First Int. Conf. Comput. Vision*, 1987, pp. 284–294.

[32] L. Cohen, L. Vinet, P. Sander, and A. Gagalowicz, "Hierarchical region based stereo matching," in *Proc. Comput. Vision Pattern Recognit.*, 1989, pp. 416–421.

[33] Y. C. Hsieh, F. Perlant, and D. M. Mckeown, "Recovering 3D information from complex aerial imagery," in *Proc. Int. Joint Conf. Pattern Recognit.*, 1990, pp. 136–146.

[34] X. Hu and N. Ahuja, "Estimation of motion of constant acceleration from image sequences," in *Proc. Int. Joint Conf. Pattern Recognit.*, Hague, The Netherlands, vol. 1, 1992, pp. 655–659.

[35] X. Hu, "Perception of shape and motion," Ph.D. dissertation, Dep. Elec. and Comput. Eng., Univ. of Illinois at Urbana-Champaign, 1993.

[36] X. Hu and N. Ahuja, "Feature extraction and matching as signal detection," in *Proc. SPIE Conf. Applicat. Artificial Intell. XI: Machine Vision and Robotics*, Orlando, FL, 1993.

[37] ——, "Feature extraction and matching as signal detection," to appear in *Int. J. Pattern Recognit. Artificial Intell.*, vol. 8, no. 6, 1994.

[38] K. Prazdny, "Egomotion and relative depth map from optical flow," *Biolog. Cybern.*, vol. 36, pp. 87–102, 1980.

[39] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 4, pp. 384–401, 1985.

[40] V. Salari and I. K. Sethi, "Feature point correspondence in the presence of occlusion," *Pattern Anal. Machine Intell.*, vol. 12, no. 1, pp. 87–91, 1990.

[41] J. J. Rodriguez and J. K. Aggarwal, "Matching aerial images to 3-D terrain maps," *Pattern Anal. Machine Intell.*, vol. 12, no. 12, pp. 1138–1149, 1990.

[42] W. Y. Kim and A. C. Kak, "3-D object recognition using bipartite matching embedded in discrete relaxation," *Pattern Anal. Machine Intell.*, vol. 13, no. 3, pp. 224–251, 1991.

Control Structure for Interpreting Handwritten Addresses

Edward Cohen, Jonathan J. Hull, and Sargur N. Srihari

***Abstract*—This correspondence describes the control structure for an intelligent handwritten address interpretation system. The system takes a grey-level address image, segments the address into lines and words, parses the address into meaningful syntactic categories, recognizes words using dynamically generated lexicons, and determines the destination code with the aid of postal directories.**

***Index Terms*—Text processing, handwriting recognition, image processing**

I. INTRODUCTION

The Handwritten Address Interpretation System (HWAIS) takes an off-line handwritten postal address image and determines a unique mail delivery point (e.g., a mailbox). In the United States, each

Manuscript received May 20, 1992; revised February 21, 1994. Recommended for acceptance by Associate Editor T. Dean.

E. Cohen is with Accu-Sort Systems Inc., 511 School House Road, Telford, PA 18969 USA; e-mail: cohen@cis.udel.edu.

J. J. Hull, and S. N. Srihari are with the Center for Document Analysis and Recognition (CEDAR), State University of New York at Buffalo, 226 Bell Hall, Buffalo, NY 14260 USA; e-mail: hull@cs.buffalo.edu and srihari@cs.buffalo.edu.

IEEE Log Number 9405251.