Correspondence

Point Pattern Matching Using Convex Hull Edges

ARDESHIR GOSHTASBY, MEMBER, IEEE, AND
GEORGE C. STOCKMAN, SENIOR MEMBER, IEEE

***Abstract*—Algorithms for matching two sets of points in a plane are given. These algorithms search in the parameter space and find the transformation parameters that can match the most points in the two sets. Since an exhaustive search for the best parameters is not affordable as the number of points in the sets becomes large, a subset selection method is given in order to reduce the search domain. Subsets are chosen as points on the boundary of the convex hulls of the sets. The algorithms are tested on generated and real data and their performance is compared.**

I. INTRODUCTION

Two images of the same scene can be registered by selecting a set of point features (control points) from each image and matching the two sets of points. Point features could be zero-crossings [1], [2], line intersections [3], high curvature points [3], centers of gravity of closed-boundary regions [4], or centers of windows located at high variance [5] and unique areas [6] in an image. In the following, we will assume two given sets of point features of one type in a plane and discuss the problem of determining the transformation parameters (translation, rotation, and scaling) that can best match the two sets (best in the sense that the sum of squared errors in matching is minimum). It is assumed that the two sets of points may have translational, rotational, and scaling differences, that there might be some points existing in only one set, and that the sets might be noisy.

Point pattern matching by an iterative approach is given by Ranade and Rosenfeld [7] for two sets having translational differences and by Wong *et al*, [8] for two sets having translational and rotational differences. Zahn [9] has given a procedure for matching of point sets that have translational, rotational, and scaling differences by matching minimum spanning trees (MSTs) of the two sets. MST's are matched using degree, minimum angle, and length ratio of edges that make the minimum angle in each node.

A technique to match point patterns by a clustering approach is given by Stockman *et al*, [10]. In the clustering approach, matching is carried out between all possible pairs of points in the two sets. While matching point pairs, the translational, rotational, and scaling difference between them are determined and a point is entered into the parameter space showing the parameter values. Correct matches tend to make a cluster while mismatches randomly fill the parameter space. The parameter values corresponding to the densest cluster center are taken as the transformation parameters that can match the two sets of points.

These techniques for point pattern matching become very slow as the number of points in each set increases. Fischler and Bolles have proposed the idea of using subsets of points for matching in order to reduce the computation time [11]. Subsets are chosen randomly from each set and transformation parameters are determined from the subsets and then verified on other points.

This correspondence describes a procedure for matching two sets of points ($S1$ and $S2$) based on a patterned search of subsets determining the transformation parameters that can match the largest number of points in the two sets. Since the search becomes very slow as the number of points in the sets increase, a subset from each set is chosen to speed up the search process. Rather than choosing subsets randomly, they are chosen selectively in order to maximize the number of common points in the two subsets. In this approach, points falling on the boundary of the convex hull in each set are chosen as subset for matching. The procedure for point pattern matching is as follows.

1) Determine the convex hull of each set of points and choose the points on the boundary of the convex hull as the representative subset. Let the subsets be $C1$ and $C2$ for sets 1 and 2, respectively.
2) Determine the transformation parameters ($R =$ rotation, $S =$ scaling, and $T =$ translation) needed for matching a pair of points in $C1$ to a pair of points in $C2$.
3) For each such (R, S, T), determine the number of other points in $S1$ that match with points in $S2$ within a given distance threshold value D.
4) Let (R_m, S_m, T_m) be the transformation parameters that match the most points in $S1$ and $S2$, then take (R_m, S_m, T_m) as an estimate for the transformation parameters.
5) Using (R_m, S_m, T_m), map one set into another and determine the corresponding points in the two sets that match by the threshold value D.
6) Determine the optimal transformation parameters by using the matched points in the two sets and minimizing the sum of squared errors.

In the following each step is described in more detail.

II. SUBSET SELECTION

Some of the properties of a point set that are invariant under translation, rotation, and scaling are the convex hull and the minimum spanning tree of the set. Depending on the set, we may get at least three points (two in the degenerate case) and at most all points in the set on the boundary of the convex hull (BCH). The MST of a set, however, involves all points in the set. In the following, we will be taking points in the set falling on the BCH as a subset for matching. If one set is merely a translated, rotated, and scaled version of another set, we will have exactly the same points on the BCH's of the two sets. If one set is a noisy version of another and/or some points are missing from either sets, we may not get exactly the same points on the BCH's, but we still expect some common points between them. A random process that deletes or adds points to a set will usually do so in the interior of the set and not on the boundary. Thus the BCH should be reasonably robust under deletions or additions. Moreover, BCH points are expected to be relatively far apart, a property which will increase the accuracy of registration parameters derived from them.

Determination of the convex hull of a set of points will not be discussed here. Efficient algorithms already exist that can determine the convex hull of a set of points in a plane. An excellent

---

Manuscript received October 15, 1984; revised May 13, 1985. Data for this work was provided by the Center for Remote Sensing, Michigan State University.

A. Goshtasby is with the Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA.

G. C. Stockman is with the Department of Computer Science, Michigan State University, East Lansing, MI 48823, USA.

0018-9472/85/0900-0631$01.00 ©1985 IEEE

survey of convex hull algorithms is given by Allison and Noga [12].

III. Estimation of Transformation Parameters

Once each set is reduced to a subset, matching is carried out between the two subsets for estimation of transformation parameters (R, S, T). The following algorithm estimates the transformation parameters needed for matching $S1$ and $S2$.

Algorithm 1: Determination of transformation parameters between two sets of points $S1$ and $S2$ in a plane by matching all edges in complete graphs from the convex hulls $C1$ and $C2$ of the two sets.

1) Determine the complete graphs of $C1$ and $C2$. If there are $m1$ points in $C1$ and $m2$ points in $C2$, we obtain $t1 = m1(m1-1)/2$ edges in $C1$ and $m2(m2-1)/2$ edges in $C2$. Since there are two ways to match a pair of edges directionally, for every edge in $C2$ we have to reverse the edge and match with edges in $C1$ again: we obtain $t2 = m2(m2-1)$ directional edges in $C2$.
2) For $i = 1, t1$
3) For $j = 1, t2$
4) Match edge i in $C1$ to edge j in $C2$ and determine (R_{ij}, S_{ij}, T_{ij}).
5) Knowing parameters (R_{ij}, S_{ij}, T_{ij}), determine the number N_{ij} of other points in $S1$ and $S2$ that match within a small threshold distance D by successively transforming points of $S1$ and checking a corresponding point in $S2$.
6) Let $N_{IJ} = \max_{i,j}\{N_{ij}\}$: then (R_{IJ}, S_{IJ}, T_{IJ}) are the estimated transformation parameters.

Consider two small point sets of Fig. 1(a). Algorithm 1 first determines the convex hull complete graphs of the two sets as shown in Fig. 1(b). Then it matches all edge combinations from the two sets. The edge combination that can make a configuration to match more points in the two sets than any other combination, determines the transformation parameters between the two sets.

The transformation which can map two sets of points in a plane with translational, rotational, and scaling differences is

$$x' = S(x\cos\theta - y\sin\theta) + h \tag{1}$$
$$y' = S(x\sin\theta + y\cos\theta) + k$$

where (x, y) and (x', y') are coordinates of corresponding points in $S1$ and $S2$, respectively, and $R = \theta$, S, and $T = (h, k)$ are the transformation parameters. Step 4 of algorithm 1 implies determination of the transformation parameters by substituting the coordinates of two pairs of points into (1) and solving for S, θ, h, and k.

Step 5 of algorithm 1 says, knowing the transformation parameters, map $S1$ into $S2$ and for each point in $S1$ check for a point of $S2$ within threshold distance D. If a point is found add one to the count N_{ij}. We note here that step 5 could be exited whenever a partial sum N_{ij} reached a certain value, giving a faster suboptimal algorithm.

If quad-tree search is employed the search of step 5 would take $\log_{2(n2-2)}$ operations [13]. If there are $n1$ points in $S1$, there is a need to repeat the search $n1-2$ times. Therefore, step 5 involves $(n1-2)\log_{2(n2-2)}$ distance computations, and since step 5 is executed $m1(m1-1)m2(m2-1)/2$ times, algorithm 1 involves $0.5m1(m1-1)m2(m2-1)(n1-2)\log_{2(n2-2)}$ distance computations. Since computation times of steps 1, 4, and 6 are negligible compared to that of step 5, the computational complexity of algorithm 1 is $0.5m1(m1-1)m2(m2-1)(n1-2)\log_{2(n2-2)}$. It is evident by this analysis that we should assign the largest set to $S2$ for faster speed.

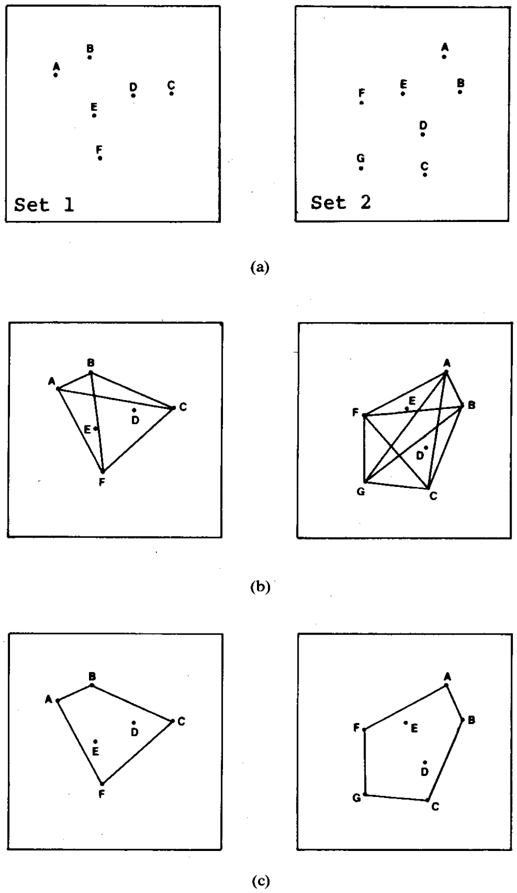


Fig. 1. (a) Two sets of point patterns. Set 2 is rotated by 90 degrees compared to Set 1 with added point G. (b) Convex hull complete graph edges of two sets, used by algorithm 1. (c) BCH edges of the two sets, used by algorithm 2.

Rather than matching every edge in the complete graphs $C1$ and $C2$, we may match only those edges that fall on the BCH's with $m1$ and $m2$ points, respectively, $(t1 = (m1-1)$, $t2 = 2(m2-1))$, the number of times steps 4 and 5 are executed in algorithm 2 can be reduced from $m1(m1-1)m2(m2-1)/2$ to $2(m1-1)(m2-1)$. This idea yields the following algorithm.

Algorithm 2: Determination of transformation parameters between two sets $S1$ and $S2$ of points in a plane by matching edges on the BCH's $C1$ and $C2$ of the two sets.

1) Determine the edges falling on the BCH's $C1$ and $C2$. If there are $m1$ points in $C1$ and $m2$ points in $C2$, we obtain $t1 = m1 - 1$ nondirectional edges in $C1$ and $t2 = 2(m2-1)$-directional edges in $C2$.

Steps 2, 3, 4, 5, and 6 are similar to the corresponding ones in algorithm 1.

Compared to algorithm 1 that matched all convex hull complete graph edge combinations, algorithm 2 matches only the convex hull edge combinations. For point sets of Fig. 1(a), this difference can be seen from Fig. 1(b) and (c). Algorithms exist [14] that directly output the convex hull represented as its boundary edges as shown in Fig. 1(c).

Since computation time for determination of convex hulls is negligible compared to the search time, computational complexity of algorithm 2 is $2(m1-1)(m2-1)(n1-2)\log_{2(n2-2)}$, which is faster than algorithm 1 by a factor of $m1m2/4$. However, there are cases, where algorithm 2 may fail to estimate the correct transformation parameters, while algorithm 1 will succeed (such as when two convex hulls have some common points but have no common edges). However, as the number of points in the two sets increases, the occurrence of such cases will be very rare, and algorithm 2 is preferable over algorithm 1. Should algorithm 2 fail to match a satisfactory number of points, algorithm 1 can be automatically invoked.

IV. DETERMINATION OF CORRESPONDING POINTS

Knowing the transformation parameters between the two sets, we can now map one set into another and find out points from set 1 that fall within a threshold distance of points in set 2. If more than one point from set 1 falls within the threshold value of a point in set 2, the closest point is taken as the corresponding one.

V. COMPUTATION OF THE OPTIMAL TRANSFORMATION PARAMETERS

In Section III, the transformation parameters needed to match two sets of points in a plane were estimated, and in Section IV these parameters were used to determine the corresponding points in the two sets. Knowing the coordinates of corresponding points in the two sets, we can now determine the optimal transformation parameters between the sets (optimal in the sense that the sum of squared errors in matching the two sets of points is minimum).

If N pairs of corresponding points $[(x_i,y_i),(x_i',y_i')]$, $i=1,N$ from the two sets are available ($N \geqslant 2$) we can determine $R=\theta$, S, and $T=(h,k)$ by minimizing the sum of squared errors

$$E=\sum_{i=1}^{N}\left\{\left[x_i'-S(x_i\cos\theta-y_i\sin\theta)-h\right]^2 +\left[y_i'-S(x_i\sin\theta+y_i\cos\theta)-k\right]^2\right\}$$

where (x_i,y_i) and (x_i',y_i') are coordinates of corresponding points in $S1$ and $S2$, respectively. Note that there will be no points lying out of the fitting since a distance test has already been passed to match the points. If we replace $S\cos\theta$ by a and $S\sin\theta$ by b, we obtain

$$E=\sum_{i=1}^{N}\left\{x_i'-(ax_i-by_i)-h\right]^2+\left[y_i'-(bx_i+ay_i)-k\right]^2\right\}.$$

If we find the partial derivatives of E with respect to a, b, h, and k, and set them equal to zero, we obtain the following classical linear system of equations:

$$\begin{bmatrix} \sum_i(x_i^2+y_i^2) & 0 & \sum_i x_i & \sum_i y_i \\ 0 & \sum_i(x_i^2+y_i^2) & -\sum_i y_i & \sum_i x_i \\ \sum_i x_i & -\sum_i y_i & n & 0 \\ \sum y_i & \sum_i x_i & 0 & n \end{bmatrix} \begin{bmatrix} a \\ b \\ h \\ k \end{bmatrix} = \begin{bmatrix} \sum_i(x_i'x_i+y_i'y_i) \\ \sum_i(y_i'x_i-x_i'y_i) \\ \sum_i x_i' \\ \sum_i y_i' \end{bmatrix}$$

by which we can determine the optimal transformation parameters.

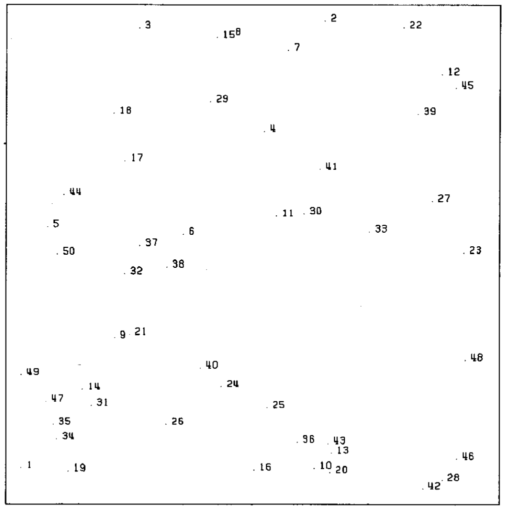


Fig. 2. Point patterns in Set 1, $S1$.

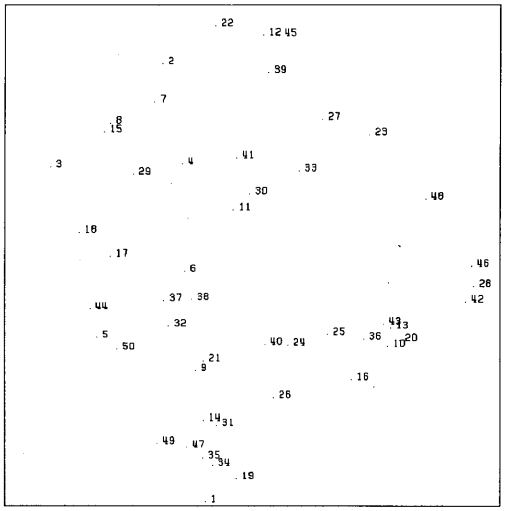


Fig. 3. Point patterns in Set 2, $S2$.

VI. TESTING AND RESULTS

To measure the performance of the proposed point pattern matching technique, several experiments were carried out. The first of those reported were done with controlled synthetic data. Then the procedure was used to match feature point sets derived automatically from different satellite images of the same area.

For the synthetic data, in each case 50 points were generated from a uniform distribution over (0.0, 100.0) in two dimensions to get $S1$ (see Fig. 2 and its convex hull in Fig. 4). $S1$ was scaled by a factor of 0.81, rotated by 0.7 radians, and translated by (10.0, 10.0) yielding $S2$ (see Fig. 3 and its convex hull in Fig. 5).

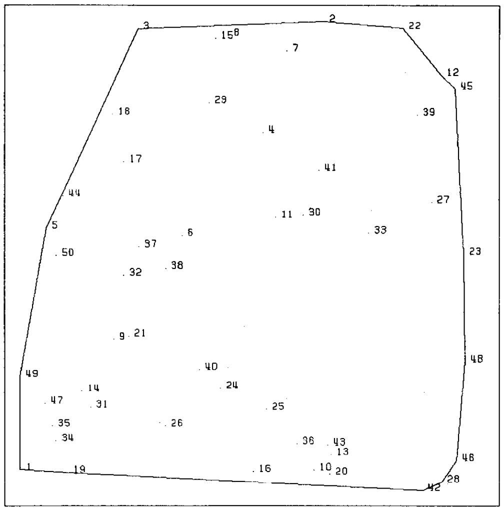


Fig. 4. The convex hull of Set 1, $C1$.

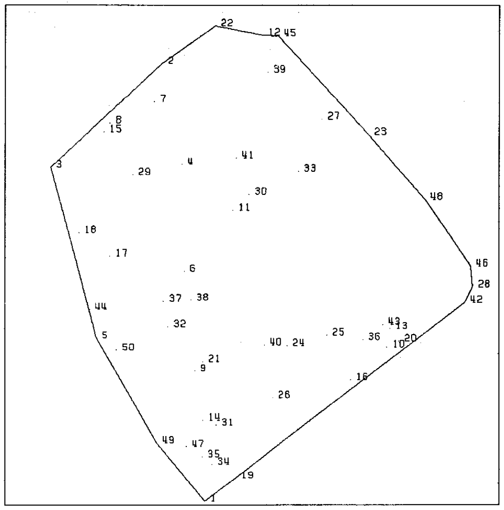


Fig. 5. The convex hull of Set 2, $C2$.

TABLE I
SPEED AND ACCURACY OF POINT PATTERN MATCHING[1]

	A		*B*		*C*	
	Exhaustive search		Selective search algorithm 1		Selective search algorithm 2	
	Number of trials	Number failed	Number of trials	Number failed	Number of trials	Number failed
1) The two sets have R, S, and T differences.	1	0	100	0	100	0
2) The two sets have R, S, and T differences and 10-percent different points.	1	0	100	0	100	1[2]
3) The two sets have R, S, and T differences and noise add-to one set.	1	0	100	0	100	0
4) The two sets have R, S, and T differences, noise added to one set, and 10-percent different points.	1	0	100	0	100	1[2]
Execution time[3]	61 hours each trial		28 seconds each trial		2 seconds each trial	

[1] On two synthetically generated sets of 50 points each. R, S, and T show translation, rotation, and scaling, respectively.

[2] Maximum number of points matched was very poor for the best match case. This fact can be used to detect a possible mismatch. If a possible mismatch is detected, algorithm 1 should be employed.

[3] In measurement of the times, direct search was employed. If quadtree search is used, smaller execution times should be obtained.

Four variations on this data were used to test three versions of the matching algorithm. The variations of the test data were as follows.

1) Points of $S1$ and $S2$ generated exactly as described.

2) Points of $S1$ and $S2$ as generated earlier. Then five points randomly deleted from $S2$ and five new randomly generated points added to $S2$.

3) Points of $S1$ and $S2$ generated as above and then noise generated uniformly over $(0.0, 1.0)$ added to all points of $S2$.

4) Points generated according to 2) and 3), that is, with additions, deletions, and random noise.

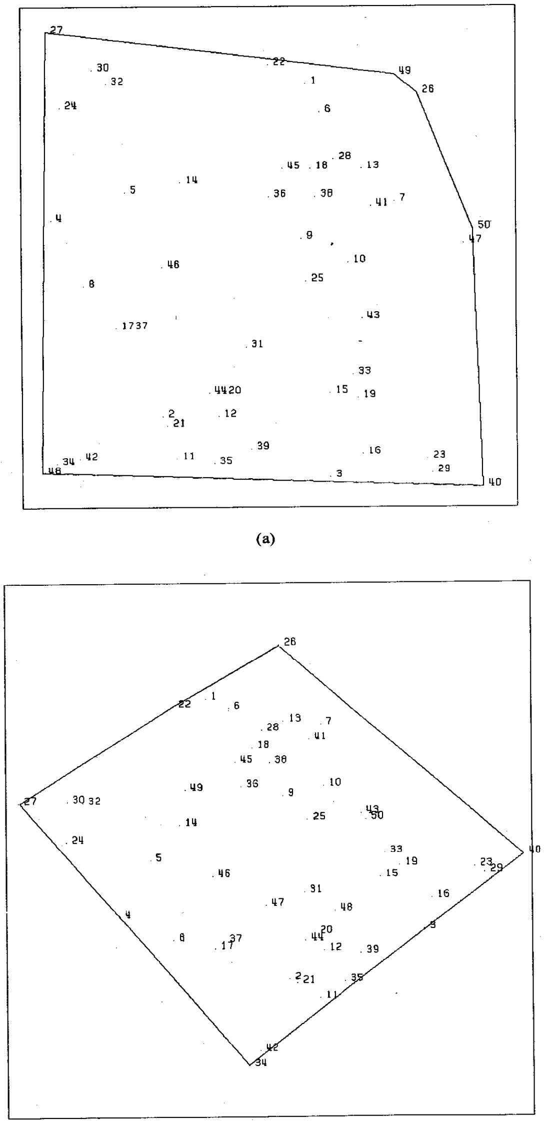
The three versions of the matching algorithm tested were as follows.

1) Matching of pairs of edges from the complete graphs of the full sets $S1$ and $S2$.

2) *Algorithm* 1: Matching of pairs of edges from the complete graphs of the BCH points $C1$ and $C2$ (See Figure 1.b).

3) *Algorithm* 2: Matching of pairs of edges wholly contained in the BCH (See Fig. 1(c)).

The results are summarized in Table I. As the first row of the table shows, all three algorithms produced correct results on the first test data in all trials but with greatly varying computation



(a)

(b)

Fig. 6. Two convex hulls. Although they have three common points, they do not have any common edges.

time. Note that the algorithms were sure to succeed on this data since $N_{IJ} = 50$ would be achieved during one of the iterations of Step 5 of the matching algorithms. Row three of Table I also shows correct matching results for all trials even with noise of amplitude 1.0.

Rows 2 and 4 of Table I show one mismatch when using algorithm 2 on data sets 2) and 4). The situation from which the mismatch was obtained is depicted in Fig. 6. The two convex hull boundaries have some common points but no common edges, causing the correct match to be missed: since the correct hypothesis (R_m, S_m, T_m) was never generated it could not be verified.

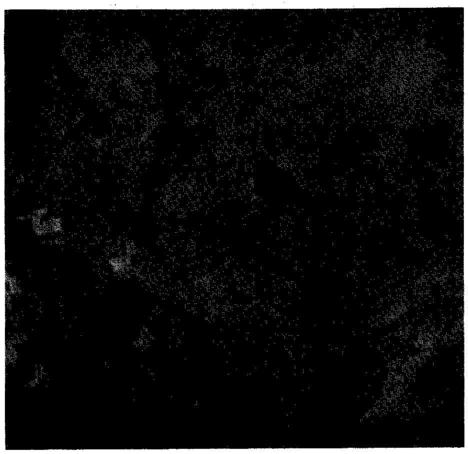


Fig. 7. Landsat TM image band-4 acquired from an area over Kalkaska County, Michigan, on Oct. 18, 1982.

Fig. 8. Landsat MSS image band-7 acquired from the same area as Fig. 7 on June 17, 1980.

To test on real data, two satellite images were used for registration. Fig. 7 shows a band-4 Landsat Thematic Mapper (TM) image acquired from an area over Kalkaska County, Michigan, on 18 Oct. 1982. Fig. 8 shows a band-7 Landsat multispectral scanner (MSS) image acquired from about the same area on June, 17, 1980. The TM image has been reduced to its half resolution so that it could cover about the same area as the MSS image. Both images are of size 240 × 240.

The two images were segmented in order to extract water bodies in the area. The segmentation process itself is reported elsewhere [4]. The segmentation results are shown in Figs. 9 and 10. The closed-boundary regions that were larger than six pixels in the perimeter were extracted and arbitrarily labeled as shown in Figs. 11 and 12. The centers of gravity of the regions were used as point features (see Table II). Distance threshold value $D = 1.0$ pixel was assumed, and applying algorithm 2, transformation parameters $R = 6.23$, $S = 0.98$, and $T = (-20.13, 0.46)$ were obtained. Using these transformation parameters with $D = 1.0$ pixel, corresponding points in the two images were determined as

Fig. 9. Segmentation of the TM image.

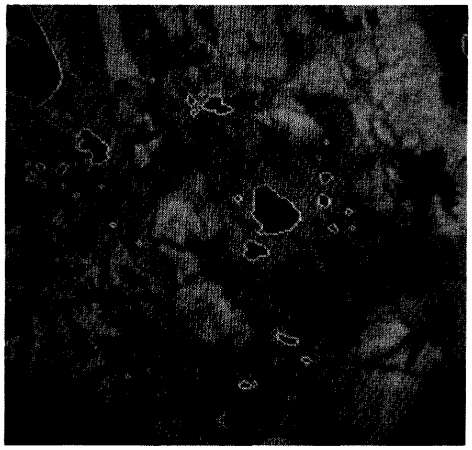


Fig. 10. Segmentation of the MSS image.

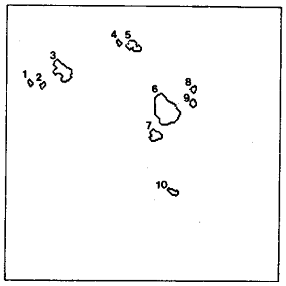


Fig. 11. Regions with perimeter greater than six pixels in the TM image.

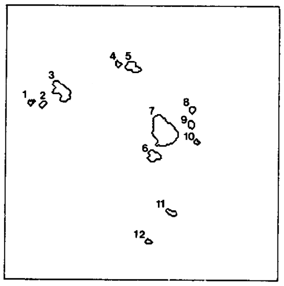


Fig. 12. Regions with perimeter greater than six pixels in the MSS image.

(1,1), (2,2), (3,3), (4,4), (5,5), (6,7), (7,6), (8,8), (9,9), and (10,11), where the first number shows the label of the region in the TM image and the second number shows the label of the region in the MSS image. Using the corresponding points in the two images and using the least-squares criterion, the optimal transformation parameters were computed to be $R = 6.27$, $S = 0.99$, and $T = (-20.59, 0.41)$.

Using the optimal transformation parameters, the MSS image was resampled (by the nearest neighbor rule) to register with the TM image (see Fig. 13). The resampled image is overlaid with the TM image in Fig. 14 to visualize the registration.

TABLE II

COORDINATES OF CENTERS OF GRAVITY OF REGIONS IN TM AND MSS IMAGES

	Coordinates of Centers of Gravity	
Region labels	TM Image	MSS Image
1	65.2, 21.0	84.4, 21.1
2	67.0, 31.8	85.7, 31.8
3	54.8, 49.2	74.0, 48.4
4	31.2, 99.0	50.1, 98.0
5	33.7, 111.4	52.6, 110.5
6	89.7, 139.6	130.2, 129.7
7	111.4, 130.3	108.6, 138.6
8	71.4, 163.9	89.6, 163.4
9	83.6, 163.4	102.2, 162.6
10	161.6, 145.7	117.6, 167.0
11		180.3, 145.8
12		205.4, 125.7

VII. DISCUSSION

Image registration can be accomplished by 1) selecting point features from the images and 2) determining which point features in the two images correspond to estimate the registration parameters. This work concentrated on stage 2 of the process. In some sense our stage 2 is similar to the random sample consensus (RANSAC) procedure [11]: candidate transformations are hypothesized by making arbitrary matches of certain pairs (edges) and then evaluating their validity on all the points. We have shown that a very reliable matching procedure can be computationally affordable by methodically generating an appropriate set of candidate transformations.

If the number of points is large, consideration of all possible point pairs becomes very time consuming. Use of a subset of points (algorithm 1) or edges (algorithm 2) lying only on the boundary of the convex hull of the original point set was shown to be reliable and efficient. Some advantages of using the convex hull are 1) invariance to translation, rotation, and scaling, 2) high likelihood of having a sparse subset, and 3) tendency to produce longer edges and hence more accurate estimates of the registration parameters. One disadvantage is that convex hull boundary points may differ in a pair of images with significantly less than

Fig. 13. The resampled MSS image.

Fig. 14. The overlaid MSS and TM images.

100-percent overlap. In these cases the best match configuration could be missed. Also, when only a small set of feature points are available, any subsetting procedure is suceptible to error.

Although our programs were not written with total attention on speed, the timing results show that stage 2 (control-point correspondence) of the registration process can be performed in less time than stage 1 (control-point selection). Since, automatic control point selection is a more difficult stage, the deliberate search for an optimal match with imperfect data seems justifiable for stage 2.

With feature points spread somewhat uniformly in the plane, the computational savings due to use of the convex hull will become larger as the number of points increases. The chance of missing the correct match becomes smaller provided that by increasing the points in a set, the number of points on the boundary of its convex hull increases also. If the two sets of points have only translational, rotational, and scaling differences, there is no chance of missing the correct match by any version of the matching algorithm because at some step in the iteration, two correctly corresponding edges from the convex hulls must be paired and subsequently all N point correspondences verified. As the amount of noise in either or both sets increases or the number of points that exist in only one set increases, the likelihood of missing the correct match will also increase.

If point features of different types from the images can be used, then the number of pairings of edges can be reduced as done by Stockman [10]. Each point might have a label such as "center of gravity" or "T intersection", which would greatly reduce the number of points with which it could correspond. The algorithm can also make good use of *a priori* bounds on the transformation parameters. Execution of the expensive step 5 of the algorithms could be bypassed whenever out-of-bounds parameter estimates are reached.

REFERENCES

[1] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Royal Soc. London*, vol. B-207, 1980, pp 187–217.

[2] S. Yam and L. S. Davis, "Image registration using generalized Hough transform," in *Proc. Pattern Recognition and Image Processing*, 1981, pp. 525–533.

[3] L. N. Kanal, B. A. Lambird, D. Levine, and G. C. Stockman, "Digital registration of images from similar and dissimilar sensors," in *Proc. Int. Conf. Cybern. Society*, 1981, pp. 347–351.

[4] A. Goshtasby, "A symbolically-assisted approach to digital image registration with application in computer vision," Tech. Rep. 83-013, Dept. of Computer Science, Michigan State University, East Lansing, MI, 1983.

[5] W. A. Davis and S. K. Kenue, "Automatic selection of control points for the registration of digital images," in *Proc. 4th Int. Joint Conf. Pattern Recognition*, 1978, pp. 171–193.

[6] E. L. Hall, D. L. Davis, and M. E. Casey, "The selection of critical subsets for signal, image, and scene matching," *IEEE Trans. Patt. Anal. Mach. Intell.*, pp. 313–322, 1980.

[7] S. Ranade and A. Rosenfeld, "Point pattern matching by relaxation," *Pattern Recognition*, vol. 12, 1980, pp. 269–275.

[8] C. Wong, H. Sun, S. Yamada, and A. Rosenfeld, "Some experiments in relaxation image matching using corner features," *Pattern Recognition*, vol. 16, 1983, pp. 167–182.

[9] C. T. Zahn, "An algorithm for noisy template matching," in *IFIP Cong.* 1974, pp. 698–701.

[10] G. C. Stockman, S. Kopstein, and S. Benett, "Matching images to models for registration and object detection via clustering," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 4, no. 3, 1982, pp. 229–241.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, June 1981, pp 381–395.

[12] D. C. S. Allison and M. T. Noga, "Some performance tests of convex hull algorithms," *BIT*, vol. 24, 1984, pp. 2–13.

[13] H. Samet and A. Rosenfeld, "Quadtree structures for image processing," *Proc. 5th Int. Conf. Pattern Recognition*, 1980, pp. 815–818.

[14] R. A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Inform. Processing Lett.*, vol. 2, 1973 pp 18–21.

An Application of the *c*-Varieties Clustering Algorithms to Polygonal Curve Fitting

JAMES C. BEZDEK, MEMBER, IEEE, AND IAN M. ANDERSON

***Abstract*—An algorithm is described that fits boundary data of planar shapes in either rectangular coordinate or chain-coded format with a set of straight line segments. The algorithm combines a new vertex detection method, which locates initial vertices and segments in the data, with the *c*-elliptotype clustering algorithm, which iteratively adjusts the location of these initial segments, thereby obtaining a best polygonal fit for the data in the mean-squared error sense. Several numerical examples are given to exemplify the implementation and utility of this new approach.**

I. INTRODUCTION

Let $\mathrm{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_N\}$ be N points in the plane ordered along the boundary of some planar shape. For such a data set, we

---

Manuscript received October 6, 1984; revised May 13, 1985. This work was supported in part by the National Science Foundation under grants IST-84-07860 and MCS-80-02328.

J.C. Bezdek is with the Department of Computer Science, University of South Carolina, Columbia, SC 29208, USA. I. M. Anderson is with the Department of Mathematics, Utah State University, Logan, UT 84322, USA.

0018-9472/85/0900-0637$01.00 ©1985 IEEE