Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition Letters 24 (2003) 9–19

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Genetic algorithm for affine point pattern matching

Lihua Zhang, Wenli Xu *, Cheng Chang

Department of Automation, Tsinghua University, Beijing 100084, PR China

Received 14 September 2001; received in revised form 25 April 2002

Abstract

Point pattern matching (PPM) is an important topic in the fields of computer vision and pattern recognition. According to if there exists a one to one mapping between the two point sets to be matched, PPM can be divided into the case of complete matching and the case of incomplete matching. According to if utilizing information other than 2-D image coordinates, PPM can be divided into labelled point-matching case and unlabelled point-matching case. Using partial Hausdorff distance, this paper presents a genetic algorithm (GA) based method to solve the incomplete unlabelled matching problem under general affine transformation. Since it successfully reduces the solution space of GA by constructing 'feature ellipses' of point sets, the method can achieve high computing efficiency and good matching results. Theoretical analysis and simulation results show that the new algorithm is very effective.

Keywords: Point pattern matching; Affine transformation; Genetic algorithm; Partial Hausdorff distance; Feature ellipses

1. Introduction

Point pattern matching (PPM) is an important problem in the fields of computer vision and pattern recognition. Its main task is to pair up the points in two images of a same scene. It can be widely used in many applications, such as image registration, object recognition, object tracking, autonomous navigation, pose estimation, etc.

According to if there exists a one to one mapping between the two point sets to be matched, PPM can be divided into the case of complete matching and the case of incomplete matching. Up to now, the case of complete matching has been extensively treated and many effective algorithms (Atkinston, 1987; Griffin and Alexopoulos, 1989; Hong and Tan, 1988; Lavine et al., 1983; Scott and Longuet-Higgins, 1991; Shapiro and Brady, 1992; Simon et al., 1972; Sprinzak and Werman, 1994; Wang, 1983; Zahn, 1974; Zhang Lihua and Xu Wenli, 1999a; Zhang Lihua and Xu Wenli, 1999b) have been put forward. However, in practical applications, it is inevitable that there exist spurious or lost points in the images. So research on the case of incomplete matching is much more important.

According to if utilizing additional information (such as color, intensity, etc.) other than 2-D image coordinates, PPM can be divided into labelled point-matching case and unlabelled point matching. Clearly the unlabelled point matching is more

* Corresponding author.
E-mail address: zhanglihua@ieee.org (L. Zhang).

0167-8655/03/$ - see front matter
PII: S0167-8655(02)00160-5

difficult than the former, but it can be used in more general applications.

This paper will focus on incomplete unlabelled point pattern-matching problem.

Vindo and Ghose (1993) treat the point-matching problem under Euclidean transformation as a 0–1 IPP problem and solve it by an asymmetric neural network. Morgera and Cheong (1995) solve the point-matching problem under rigid transformation by using the theory of Lie group. However, their algorithm can only be used for the case in which one point pattern is a subset of the other. Stockman et al. (1992) use a clustering approach to estimate the similarity transformation parameters and then determine the corresponding point pairs. Goshtasby and Stockman (1985) use the convex hulls of the point sets to reduce the complexity of the matching problems. Unfortunately, if there exist spurious or lost points on the convex hulls, the algorithm fails. The cluster-based approach proposed by Chang et al. (1997) is very effective. But it cannot insure finding out the optimal solutions with as many matching point pairs as possible, since it solves the problem from bottom to top. Recently, Zhang Lihua and Xu Wenli (in press) introduce some definitions of matching clique, support point pair, index set, and index matrix, etc. Based on these definitions, several properties and a theorem are proposed, and a new algorithm to solve the problem of matching two point sets with the different cardinality under rigid transformation is presented. On the contrary to other methods in the literature, the algorithm searches for the optimal solutions from top to bottom. Therefore, it can find as many matching point pairs as possible. However, the algorithm cannot be directly used under similarity transformation and affine transformation.

Compared with the incomplete matching problems under Euclid (rigid) and similarity transformations, the incomplete matching problem under general affine transformation is much more difficult. Therefore, in current literatures, only a small ratio really discusses the incomplete unlabelled point-matching problem.

According to the observation that any triple of points can define a valid plane affine transformation, Lamdan et al. (1988) use a voting scheme—geometric hashing, which considers all possible triples of points. Obviously its computational complexity is very high. In contrast, the method proposed by Huttenlocher (1991) is able to eliminate most matches by considering just pairs of points. The remaining matches are then enumerated. The method is based on a new affine invariant constructed by distance ratios defined by quadruples of points. However, the method will fail if three pre-selected points used for defining the new affine invariant do not appear in both point sets.

The current algorithms found in literature try to solve the difficult problem using very conventional techniques, e.g. method of exhaustion and geometric hashing. Undoubtedly, these conventional techniques are ever very useful for solving Euclid and similarity cases. However, for complex affine transformation case, their efficiency is not very high. Oppositely, using partial Hausdorff distance, this paper presents a method based on an unconventional technique—genetic algorithm (GA) to solve the incomplete point-matching problem. The method can achieve high computing efficiency and good matching results, because it successfully reduces the solution space of GA by constructing 'feature ellipses' of point sets. Theoretical analysis and simulation results show that the new algorithm is very effective.

2. Principle of algorithm

In order to grasp our algorithm as a whole, we firstly introduce its fundamental idea briefly.

Given two point sets P and P' which satisfy a general affine transformation relation, before constructing chromosomes, a triple of points should be chosen on the feature ellipse of P. Here the triple of points may not be the real points of P to be matched and it is named the referenced triplet. The concept of feature ellipse will be described in the section of preliminary knowledge. Each chromosome is composed of a triple of points of P' using binary coding. Similarly chromosomes are not requested to be real points of P', but points locating in the image plane of P'.

Obviously the triple of points represented by each chromosome and the referenced triplet can

determine a valid plane affine transformation T. Under this transformation, point pattern P is transformed to $T(P)$. Clearly, we can judge if T is near the original transformation by comparing point patterns $T(P)$ and P'. Considering the partial bidirectional Hausdorff distance (Huttenlocher et al., 1993) can measure the degree of match between two point sets without explicitly pairing points of $T(P)$ and P', we construct fitness function by the partial bidirectional Hausdorff distance. The output of our GA is the triple of points which has the highest value of fitness function. Using the triplet and the referenced triplet, the transformation parameters can be solved to realize final point matching.

Intuitively the idea of our algorithm is similar to method of exhaustion and geometric hashing, but its computational complexity is far too lower than the two methods. This is determined by two main reasons. One reason is that GA features implicit parallelism. The other one is that the initial population of our genetic is not generalized completely randomly, but generalized in the local area demarcated by the feature ellipse of point set P', which is equivalent to restricting the possible solution space to be searched by genetic algorithm.

3. Feature ellipse of a point set

According to the principle of our algorithm, a referenced triplet should be chosen in the image plane of P and once it is paired with a triple of points represented by each chromosome, parameters of the corresponding transformation can be determined. Thus, the physical explanation of the GA is to find the most matchable triple of points in the image plane of P' to the referenced triplet.

Obviously, if we randomly choose any three points in the image plane of P as the referenced triplet, it is possible that no matchable triplet can be found in the bounded image plane of P'. This is because the numerical range of the triplet, which the referenced triplet is transformed to by the original affine transformation, is beyond the scope that chromosomes can express. Such a referenced triplet is called an unreasonable triplet, or else a reasonable triplet. To find a reasonable triplet, we give the following analysis.

Under the case of no outliers (including spurious points and lost points), suppose the original affine transformation, which transforms P to P', is T. Then the convex polygon region S_1 formed by the convex hull of P is transformed to region $T(S_1)$. Clearly $T(S_1)$ should have the same position and shape with the convex polygon region S_2 formed by the convex hull of point set P'. Therefore, points locating in S_1 will be transformed to points locating in S_1. Based on this observation, the referenced triplet can be chosen from the interior of S_1, because its corresponding matching triplet must lie in S_2, certainly in the bounded image plane of P'. However, if outliers exist, the conclusion may be not true, because the outliers may disturb the convex hulls of the two point sets. The worst situation is the region S_2 does not contain some regions of $T(S_1)$. Obviously if the referenced triplet chosen in S_1 locates in these regions, it must be unreasonable. To solve this, we construct so called feature ellipse of a point set.

Definition 1. (Feature ellipse of a point set) Given a point set $P = \{\boldsymbol{p}_i = (x_i, y_i, 1)^{\mathrm{T}} | i = 1, 2, \ldots, m\}$, its feature ellipse is defined as

$$(\boldsymbol{x} - \boldsymbol{c})^{\mathrm{T}} E^{-1} (\boldsymbol{x} - \boldsymbol{c}) = \frac{1}{S}, \tag{1}$$

where S is a positive integer whose value determines the size of the feature ellipse, and $c = 1/m\sum_{i=1}^{m}\boldsymbol{p}_i$ and $E = 1/m\sum_{i=1}^{m}(\boldsymbol{p}_i - \boldsymbol{c})(\boldsymbol{p}_i - \boldsymbol{c})^{\mathrm{T}}$ are the center point and the second-order center moment of P, respectively.

Clearly the center of the feature ellipse locates in the canter of the point set, and the shape of the feature ellipse approximates the shape of P, i.e. the distribution of points of P. Via adjusting the parameter S, the feature ellipse can be insured to lie in the convex hull of P (certainly in the bounded image plane of P), further in a local region around the center of P.

Draw three rays from the center $\boldsymbol{c}$ of P, of which each pair forms an angle of 120°. The rays intersect the feature ellipse at three points, which is the expected referenced triplet. Clearly the referenced

triplet locates in the local area around the center of P. Here, the request that angles formed by each pair of rays be 120° is to make the referenced triplet nonlinear and interdistance between any two points large enough.

It is easily proved that under the case of no outliers and no image noises, for the same S the feature ellipse of P must correspond to the one of P'. Thus, the matching triplet of the referenced triplet can be found on the feature ellipse of P'. This shows that the chosen referenced triplet is reasonable because the feature ellipse of P' must lie in the bounded image plane of P'. Obviously the usage of the concept of feature ellipse is equivalent to restricting the search space of the GA, that is the algorithm can search for the matching triplet of the referenced triplet just on (notice here not in) the feature ellipse of P'.

Considering that there must exist image noises and outliers in practical applications, the matching triplet corresponding to the referenced triplet may not locate on the feature ellipse of P'. However, it will lie in a local region around the ellipse. Thus, the search space of the GA is reduced to this local area.

4. GA-based algorithm

Professor J. Holland put forward the essential idea of GA in 1975. GA is a highly efficient stochastic global optimization algorithm for problem solving. Beginning from any randomized initial population that consists of a group of chromosomes (i.e. potential search nodes), new populations are arrived at by using various genetic operators with subsequent iterations. A new chromosome replaces a previous one if it is judged to be better than it according to a function called fitness. Two main advantages of GA are it is not easy to slump into a local optimized region and it can be realized by a parallel computational method.

GA has been applied to computer vision in many fields. Some of the works related to medical image registration, image segmentation, model-based matching, and affine invariant recognition can be found in (Bhanu et al., 1991; Grefenstette and Fitzpatrick, 1985; Hill and Taylor, 1992; Tsang, 1997a,b).

To use GA in finding point correspondences, three problems are to be solved, constructing fitness function, coding chromosomes, and choosing suitable genetic operators.

4.1. Coding chromosomes

Given two point sets P and P' with m and n points respectively, each chromosome consists of a triple of points in the image plane of P'. Its detailed construction method is seriating coordinates of the three points and coding them into binary codes. Here using binary coding is because the image resolution is always limited, for example, 256×256 or 512×512, and then the coordinate of points can be represented by integers in the scope of the image resolution. Even if the location precision of points is less than a pixel, we can still obtain enough representation precision by changing the length of binary code.

Given three points p'_{j_0}, p'_{j_1} and p'_{j_2} in the image plane of P' (may be not the points of P'), Table 1 shows the detailed construction method of a chromosome consisting of the three points. Here, each x or y is represented by a binary code. Thus the whole chromosome is a binary sequence. Notice, in order for the genetic operators operating conveniently and obtaining a more precious matching pair of triplets, the chromosome is not divided into genes. The vertical lines are simply used for description. Therefore the operations of genetic operators are all executed with the unit of bit.

4.2. Generating initial population

The initial population is generated randomly. However, the generating range of chromosomes is not arbitrary but limited to the local area around the feature ellipse of P'.

Table 1
Chromosome coding

x'_{j_0}	y'_{j_0}	x'_{j_1}	y'_{j_1}	x'_{j_2}	y'_{j_2}

4.3. Constructing fitness function

Given a chromosome shown in Table 1, the coordinates of the triplet that the chromosome represents are firstly retrieved according to their binary codes. Then the parameters of the affine transformation, denoted by T, can be easily solved from this triplet and the referenced triplet. Obviously, if the degree of match between P and P' under the transformation can be measured, it is equivalent to evaluating the fitness of the chromosome.

Considering the partial bidirectional Hausdorff distance between point sets P' and $T(P)$ to which P is transformed by T, smaller the distance is, the degree of match between P and P' is larger. So the fitness function can be selected as the inverse of the partial bidirectional Hausdorff distance,

$$\text{fitness} = \frac{1}{1 + H_{\text{LK}}(T(P), P')}. \tag{2}$$

Notice here, the denominator is the partial bidirectional Hausdorff distance plus 1 in order to avoid zero appearing in the denominator.

4.4. Defining genetic operators

- *Selection*: Selection operator is used to select good chromosomes that contribute their gene-inherited knowledge for the next generation. Here we use commonly used Roulette-wheel selection process. In Roulette-wheel selection process, the selection probability of each individual is proportional to its fitness value.
- *Crossover*: Although selection operator can keeps fitter individuals (i.e. chromosomes) in evolutionary process, it does not create any new individual. Crossover operator can produce new chromosomes through combining partial structure of two father individuals. Here we adopt commonly used single point crossover operator.
- *Mutation*: In order to prevent the loss of diversity in the evolutionary process, an operator named uniform mutation is designed and carried on in a small probability. It operates on each binary bit of a chromosome in another predefined probability. We realize the mutation operator by reverse the value of the current binary bit, i.e. 0–1, 1–0.

4.5. Full genetic algorithm

Algorithm 1. Genetic algorithm for affine point matching

Given two point patterns to be matched with each other P and P', choose population size N, crossover probability p_c, mutation probability p_m, two fractions f_L and f_K of the partial bidirectional Hausdorff distance and the maximum iterative steps $G_{\max}$.

Step 1: Compute the feature ellipses of P and P', and then solve the referenced triplet in the image plane of P and determine the local search area in the image plane of P' for the GA.

Step 2: Randomly generate N triplets in the local search area and then convert them into chromosomes for initial generation.

Step 3: Compute fitness function values of all chromosomes in current population and then select fitter chromosomes by selection operator.

Step 4: Apply crossover operator at the probability p_c and mutation operator at the probability p_m to the fitter chromosomes and generate the population of next generation.

Step 5: If the maximum iterative steps $G_{\max}$ is not reached, go to Step 3. Otherwise let T_{best} be the affine transformation determined by the best chromosome, and match point patterns $T_{\text{best}}(P)$ and P' according to the simple nearest neighbor rule.

5. Experimental results

To verify the effectivity of our algorithm, a lot of simulated experiments are performed. What follows is an example of the simulations.

Firstly a point pattern P with 15 points is generated randomly in a numerical bound of 500×500, and another point pattern P' is obtained after an affine transformation with

$$A = \begin{bmatrix} 0 & 1.5 \\ 0.9 & 0 \end{bmatrix}$$

and $\boldsymbol{t} = [10\ 10]^{\mathrm{T}}$ is applied to P. After the first point is deleted from P', two randomly generated spurious points are added to P'. Finally Gaussian noises with mean-value zero and variance 1.0 are added to coordinates of each point in P and P'. Here noise range is bounded to less than ± 3. The final point patterns P and P' are shown in Figs. 1 and 2, respectively. And the coordinates of each point are listed in Tables 2 and 3.

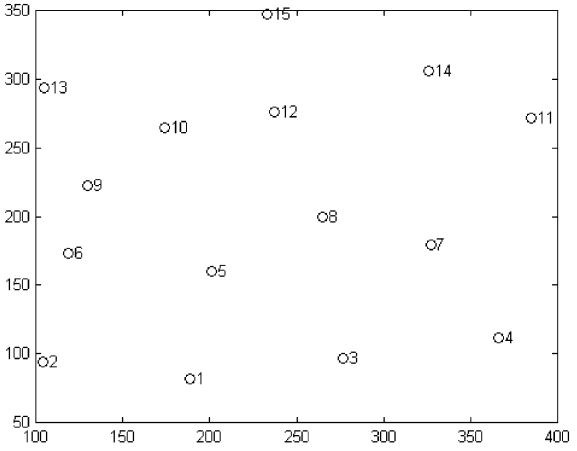

Fig. 1. Point pattern P.

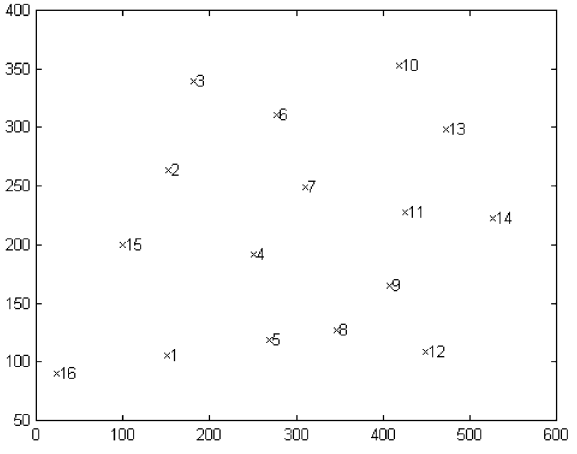

Fig. 2. Point pattern P'.

In the experiment, the parameters of Algorithm 1 are set as, population size $N = 400$, crossover probability $p_c = 0.05$, mutation probability $p_m = 0.02$, two fractions f_L and f_K are all 0.7, and the maximum iterative steps $G_{max} = 100$. The length of binary code for each coordinate is 16 and the parameter S in Eq. (1) is 4.

After randomly running Algorithm 1 for 10 times, the best chromosome is obtained and its corresponding affine transformation parameters are

$$A = \begin{bmatrix} -0.0031 & 1.4874 \\ 0.8663 & 0.0073 \end{bmatrix},$$
$$\boldsymbol{t} = [15.5989 \quad 16.4039]^{\mathrm{T}}.$$

Under this transformation, point pattern P is transformed to $T(P)$. To give an intuitive observation, we plot $T(P)$ together with P' in Fig. 3. It is easily seen that each matchable point approach its pairing point very near. Using the simple nearest neighbor rule, the matching result shown in Table 4 is obtained. Clearly the result is completely correct.

To further demonstrate Algorithm 1's adaptability, here we briefly described another experiment in which a larger point set P with 50 points is randomly generated. Point pattern P' is obtained after a random affine transformation with P.

$$A = \begin{bmatrix} -0.2967 & 0.2683 \\ -0.4695 & -0.5192 \end{bmatrix}, \qquad \boldsymbol{t} = [273 \quad 519]^{\mathrm{T}}$$

is applied to P. Then we randomly deleted 20 points from P'. Meanwhile, 20 randomly generated spurious points are added to P'. Finally, larger Gaussian noises with mean-value zero and variance 5.0 are applied. Clearly, the experiment condition is much more rigorous than the previous experiment.

The final point patterns P and P' are shown in Figs. 4 and 5, respectively, and the coordinates of each point are listed in Tables 5 and 6. During the experiment preparing process, since we could know which points are deleted and which points are added, the expected matching result can be predicted in Table 7.

The experimental parameters of Algorithm 1 are set as, $N = 500$, $p_c = 0.03$, $p_m = 0.03$, f_L and f_K are all 0.6, and $G_{max} = 100$. The length of binary

Table 2
Coordinates of point pattern P

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	189	104	277	366	201	119	327	265	130	174	385	237	105	326	233
y	82	94	97	112	160	173	179	200	222	265	272	276	294	306	347

Table 3
Coordinates of point pattern P'

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x'	151	153	182	251	269	277	310	346	408	419	426	449	473	527	100	24
y'	105	263	339	192	119	311	249	127	165	353	228	108	298	222	200	90

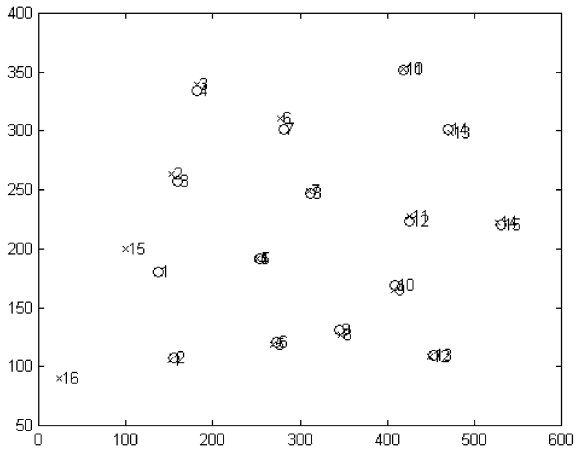

Fig. 3. Point pattern $T_{\text{best}}(P)$ and P'.

code for each coordinate is still 16 and the parameter S is also 4.

After randomly running Algorithm 1 for 10 times, the best chromosome is obtained and its corresponding affine transformation parameters are

$$A = \begin{bmatrix} -0.3264 & 0.2543 \\ -0.4634 & -0.5152 \end{bmatrix},$$

$$\boldsymbol{t} = [\,287.6498 \quad 515.1693\,]^{\mathrm{T}}.$$

Clearly this result is very close to the true affine transformation used to setup the experiment.

Under this transformation, point pattern P is transformed to $T(P)$. Similarly, to give an intuitive observation, $T(P)$ is plotted together with P' in Fig. 6. Using the simple nearest neighbor rule, a matching result is obtained in Table 8. Compared with the expected result listed in Table 7, clearly we correctly matched 25 of 30 matching pairs, the other two matching pairs are incorrect because the spurious points are too close to the original points. So the simple nearest neighbor rule cannot differentiate them from the correct matchings. In fact, the matching result can be further improved by utilizing more advanced optimized matching techniques (Chang et al., 1997; Xu Wenli and Zhang Lihua, 2001) for point pattern matching under rigid transformation instead of the simple nearest neighbor rule. This is because the matching subsets of point pattern $T(P)$ and P' should satisfy a rigid transformation relation instead of an affine transformation.

6. Complexity analysis

Supposing the size of the two point sets to be the same n, determining the complexity of the algorithm proposed in this paper is very straightforward: $n^2 \times N \times G$, where $O(n^2)$ is the complexity of computing the Hausdorff distance between the two point sets, N is the size of the population and G is the number of iterative generations. Here N and G are preset parameters and they are relatively independent of n. In our experiments, N varies from 100 to 1000, and G is adjusted from 60 to 100.

Now we turn to discuss the complexity of an exhausting method. If using the same measurement (Partial Hausdorff distance), for three given points (i.e. a referenced triplet) in point set P',

Table 4
Result of point pattern matching

P	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P'	1	2	3	4	5	6	7	8	9	10	11	12	13	14

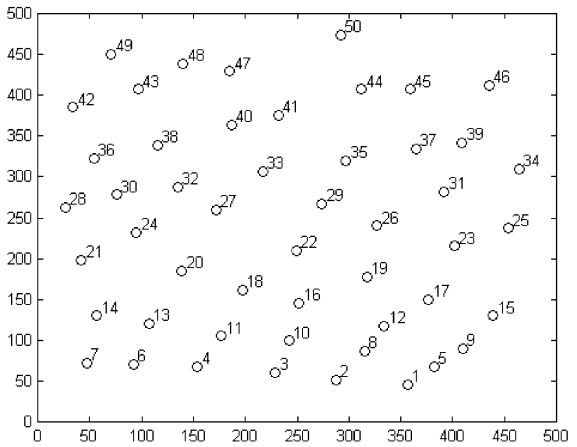

Fig. 4. Point pattern P in experiment 2.

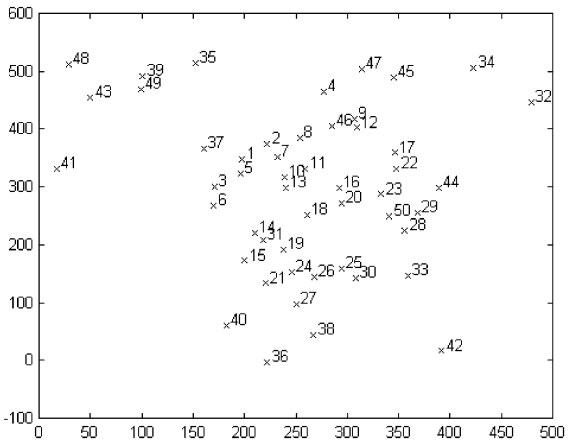

Fig. 5. Point pattern P' in experiment 2.

Table 5
Coordinates of point pattern P

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
x	357	287	229	154	382	92	47	315	410	242	177	334	107	57	439	252	377	197	317	139	42	249	402	95	454
y	45	52	60	67	67	70	72	87	90	99	105	117	120	130	130	145	150	162	177	185	198	210	215	232	237

	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
x	327	172	26	274	76	392	135	217	464	297	54	365	115	409	187	232	34	97	312	359	435	185	140	71	292
y	240	260	263	267	278	282	287	307	310	320	323	334	339	342	364	375	385	407	407	407	412	429	439	450	474

Table 6
Coordinates of point pattern P'

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
x'	198	222	171	277	196	170	232	254	307	239	259	309	240	210	200	292	346	261	238	294	220	348	333	246	294
y'	348	374	300	464	323	267	352	385	418	317	331	402	298	221	172	298	359	251	191	271	133	331	287	153	159

	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
x'	268	250	356	368	308	218	479	359	423	152	222	160	267	101	182	17	392	50	389	345	285	314	29	99	341
y'	145	98	224	255	142	208	446	146	505	514	−4	366	44	492	61	331	16	454	299	489	405	504	512	469	249

a randomized algorithm which exhausts all the possible matching triplets in point set P would have complexity of $n^2 \times C_n^3$. An implicit big assumption here is that the given referenced triplet in P' should

Table 7
Expected result of point pattern matching

P'	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
P	2	3	5	7	8	9	10	11	14	16	18	21	22	23	25	27	28	29	31	33	34	36	38	39	44
P'	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
P	45	46	48	49	50																				

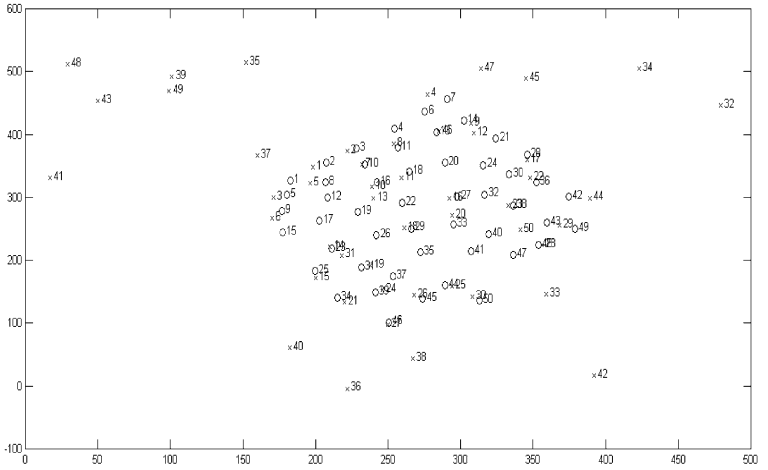

Fig. 6. Point pattern $T_{\text{best}}(P)$ and P' in experiment 2.

Table 8
Testel result of point pattern matching

P'	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
P	2	3	5		8	9	10	11	14	16	18			23	25	27		29	31	33	34	36	38	39	44
P'	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
P	45	46	48		50														42		13				

also appear in the other point set P. Obviously this assumption might not be true in case of incomplete point matching. Suppose the probability of a point appearing in both P' and P is r. Easily verified, to guarantee finding a referenced triplet appearing in both point sets, the least combination number of referenced triplets that should be tried is

$$(C^1_{n(1-r)} \times C^2_{nr} + C^2_{n(1-r)} \times C^1_{nr} + C^3_{n(1-r)}).$$

Therefore, the total computation complexity of an exhausting method would be

$$n^2 \times C^3_n \times (C^1_{n(1-r)} \times C^2_{nr} + C^2_{n(1-r)} \times C^1_{nr} + C^3_{n(1-r)}),$$

i.e. $O(n^8)$.

Comparing $O(n^8)$ and $n^2 \times N \times G$, it is easily concluded that with the increase of n, our GA-based algorithm's computation efficiency is much better than the exhausting method. For instance, in our second experiment, the parameters we used are $n = 50$, $N = 500$, $G = 100$ and $r = 0.60$. Easily know the complexity of the GA is around $1.2e+8$, but the complexity of an exhausted algorithm will

be nearly $1.2e+13$. So Algorithm 1 reduced the computational complexity to $1.0e-5$ of the exhausted method with $n = 50$. This further proves the effectivity of our algorithm.

7. Conclusions

In this paper, using partial Hausdorff distance, a GA-based method for the incomplete point pattern-matching problem under general affine transformation is presented. The method can achieve high computing efficiency and good matching results, because it successfully reduces the solution space of GA by constructing feature ellipses of point sets. Theoretical analysis and simulation results show that the new algorithm is very effective.

Before the paper is finalized, our future research direction will be briefly mentioned. That is, although the proposed algorithm is very effective in normal case, if the referenced triplet of point set P is wrongly chosen, there will be no corresponding triplet in the local search area of P'. To solve this problem, a simple possible solution is to expand the search area in P'. However, this may decrease the algorithm's efficiency to some extent. Therefore, our future research will focus on issuing a better solution.

Acknowledgements

This work was supported by the "985" project of Tsinghua University and the Major Research Project of the tenth-five plan of PR China (no. 2001BA609A).

References

Atkinston, M.D., 1987. An optimal algorithm for geometrical congruence. J. Algorithms 8, 159–172.

Bhanu, B., Lee, S., Ming, J., 1991. Self-optimizing image segmentation system using a genetic algorithm. In: Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego, CA, pp. 362–369.

Chang, S.H., Cheng, F.H., Hsu, W.H., Wu, G.Z., 1997. Fast algorithm for point pattern matching: invariant to translations rotations and scale changes. Pattern Recogn. 30 (2), 311–320.

Goshtasby, A., Stockman, G.C., 1985. Point pattern matching using convex hull edges. IEEE Trans. Syst. Man Cybernet. SMC 15 (5), 631–637.

Grefenstette, J.J., Fitzpatrick, J.M., 1985. Genetic search with approximate function evaluation. In: Proceedings of International Conference on Genetic Algorithms and their Applications, pp. 112–120.

Griffin, P.M., Alexopoulos, C., 1989. Point pattern matching using centroid bounding. IEEE Trans. Syst. Man Cybernet. 19 (5), 1274–1276.

Hill, A., Taylor, C.J., 1992. Model-based image interpretation using genetic algorithm. Image Vision Comput. 10, 295–300.

Hong, J.W., Tan, X.N., 1988. A new approach to point pattern matching. In: Proceedings of Ninth International Conference on Pattern Recognition, pp. 82–84.

Huttenlocher, D.P., 1991. Fast affine point matching: an output-sensitive method. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 263–268.

Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J., 1993. Comparing images using the Hausdorff distance. IEEE Trans. Pattern Anal. Mach. Intell. 15 (9), 850–863.

Lamdan, Y., Schwartz, J.T., Wolfson, H.J., 1988. Object recognition by affine invariant matching. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 335–344.

Lavine, D., Lambird, B.A., Kanal, L.N., 1983. Recognition of spatial point patterns. Pattern Recogn. 16 (3), 289–295.

Morgera, S.D., Cheong, P.L.C., 1995. Rigid body constrained noisy point pattern matching. IEEE Trans. Image Process. 4 (5), 630–641.

Scott, G.L., Longuet-Higgins, H.C., 1991. An algorithm for associating the features of two images. Phil. Trans. Roy. Soc. London B 244, 21–26.

Shapiro, L.S., Brady, J.M., 1992. Feature-based correspondence: an eigenvector approach. Image Vision Comput. 10 (5), 283–288.

Simon, J., Checroun, A., Roche, C., 1972. A method of comparing two patterns independent of possible transformations and small distortions. Pattern Recogn. 4 (1), 73–81.

Sprinzak, J., Werman, M., 1994. Affine point matching. Pattern Recogn. Lett. 15 (4), 337–339.

Stockman, G., Kopstein, S., Benett, S., 1992. Matching images to models for registration and object detection via clustering. IEEE Trans. Pattern Anal. Mach. Intell. PAMI 4 (3), 229–241.

Tsang, P.W.M., 1997a. A genetic algorithm for affine invariant object shape recognition. In: Proceedings of First IEE/IEEE International Symposium on Genetic Algorithm in Engineering Systems, GALESIA, pp. 293–298.

Tsang, P.W.M., 1997b. A genetic algorithm for affine invariant recognition of object shapes from broken boundaries. Pattern Recogn. Lett. 18, 631–639.

Vindo, V.V., Ghose, S., 1993. Point matching using asymmetric neural networks. Pattern Recogn. 26 (8), 1207–1214.

Wang, C., 1983. Some experiments in relaxation image matching using corner features. Pattern Recogn. 16 (2), 167–182.

Xu, Wenli, Zhang, Lihua, 2001. A geometric reasoning based algorithm for point pattern matching. Science in China, Series F 44, 445–452.

Zahn, C.T., 1974. An algorithm for noisy template matching. In: Proceedings of the IFIP Congress, pp. 698–701.

Zhang Lihua, Xu Wenli, in press. New algorithms for point-pattern matching using index matrix. In: The Third Asian Control Conference.

Zhang, Lihua, Xu, Wenli, 1999a. Point-pattern matching using irreducible matrix and relative invariant. Tsinghua Sci. Technol. 4 (4), 1602–1605.

Zhang, Lihua, Xu, Wenli, 1999b. Point pattern matching. China J. Comput. 22 (7), 740–745.