Image texture classification using a manifold-distance-based evolutionary clustering method

Maoguo Gong
Licheng Jiao
Xidian University
Institute of Intelligent Information Processing
Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China
Mailbox 224
No. 2 South TaiBai Road
Xi'an 710071, China
E-mail: gong@ieee.org

Liefeng Bo
University of Chicago
Toyota Technological Institute at Chicago
Chicago, Illinois-60637

Ling Wang
Xiangrong Zhang
Xidian University
Institute of Intelligent Information Processing
Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China
Mailbox 224
No. 2 South TaiBai Road
Xi'an 710071, China

Abstract. We perform unsupervised image classification based on texture features by using a novel evolutionary clustering method, named manifold evolutionary clustering (MEC). In MEC, the clustering problem is considered from a combinatorial optimization viewpoint. Each individual is a sequence of real integers representing the cluster representatives. Each datum is assigned to a cluster representative according to a novel manifold-distance-based dissimilarity measure, which measures the geodesic distance along the manifold. After extracting texture features from an image, MEC determines partitioning of the feature vectors using evolutionary search. We apply MEC to solve seven benchmark clustering problems on artificial data sets, three artificial texture image classification problems, and two synthetic aperture radar image classification problems. The experimental results show that in terms of cluster quality and robustness, MEC outperforms the *K*-means algorithm, a modified *K*-means algorithm using the manifold-distance-based dissimilarity measure, and a genetic-algorithm-based clustering technique in partitioning most of the test problems. [DOI: 10.1117/1.2955785]

Subject terms: image classification; texture features; evolutionary algorithms; genetic algorithms; clustering; dissimilarity measure.

Paper 071023R received Dec. 28, 2007; revised manuscript received Apr. 12, 2008; accepted for publication Apr. 28, 2008; published online Jul. 11, 2008.

1 Introduction

Image classification or segmentation based on texture features using unsupervised approaches has been a challenging topic. Texture is an important property of some images. A lot of texture feature extraction methods have been developed over the past three decades. These texture features can be divided into four major categories:[1,2] statistical, geometrical, model-based, and signal-processing. Among them, gray-level cooccurrence features, first proposed by Haralick, Shanmugam, and Dinstein,[3] are among the most common features used in the literature. In some images, the same object region may vary in appearance from image to image as well as within the same image. Thus, the selected training samples in a supervised algorithm may not be sufficient to include all the class variability throughout the image. Under these conditions, unsupervised classification (i.e., clustering) may be more effective. There are a variety of clustering approaches that could be used to assign class labels to the feature vectors. These approaches can be categorized into two groups:[4,5] hierarchical clustering and partitional clustering. Partitional clustering approaches, such as the *K*-means algorithm,[6] partition the data set into a specified number of clusters by minimizing certain criteria. Therefore, they can be treated as an optimization problem. As global optimization techniques, evolutionary algorithms (EAs) are likely to be a good choice for this task.

EAs, including genetic algorithms (GAs), evolutionary strategies (ESs), evolutionary programming (EP), etc., have been commonly used for clustering tasks in the literature.[7–10] A variety of EA representations for clustering solutions have been explored, such as straightforward encoding with each gene coding for the cluster membership of the corresponding datum, and the locus-based adjacency representation.[10] Many researchers[7–9] have chosen to use a more indirect approach that borrows from the *K*-means algorithm: The representation codes for the cluster center only, and each datum is subsequently assigned to a cluster representative according to a chosen dissimilarity measure.

The most popular dissimilarity measure is the Euclidean distance. By using it, these evolutionary clustering methods as well as the *K*-means algorithm yield good performance on data sets with compact supersphere distributions, but tend to fail on data sets organized in more complex and unknown shapes, which indicates that this dissimilarity measure is undesirable when clusters have random distributions. As a result, it is necessary to design a more flexible dissimilarity measure for clustering.

Su and Chou[11] proposed a nonmetric measure based on the concept of point symmetry, according to which a symmetry-based version of the *K*-means algorithm is given. This algorithm assigns data points to a cluster center if they

0091-3286/2008/$25.00

present a symmetrical structure with respect to the cluster center. Therefore, it is suitable for clustering data sets with clear symmetrical structure. Charalampidis[12] recently developed a dissimilarity measure for directional patterns represented by rotation-variant vectors and further introduced a circular *K*-means algorithm to cluster vectors containing directional information.

In order to perform the texture classification task effectively, in this study we design a novel evolutionary clustering method, named manifold evolutionary clustering (MEC). In MEC, we adopt an indirect encoding approach, namely, each individual is a sequence of real integer numbers representing the cluster representatives. Each datum is assigned to a cluster representative according to a novel dissimilarity measure, the geodesic distance along the manifold. After extracting texture features from an image, MEC determines a partitioning of the feature vectors using evolutionary search. The effectiveness of MEC is validated by comparing it with the *K*-means algorithm, a modified *K*-means algorithm using the manifold-distance-based dissimilarity measure,[13] and the genetic-algorithm-based clustering technique proposed by Maulik and Bandyopadhyay,[8] in solving seven benchmark clustering problems on artificial data sets, three artificial texture image classification problems, and two synthetic aperture radar (SAR) image classification problems.

The remainder of this paper is organized as follows: Section 2 describes the novel manifold-distance-based dissimilarity measure. Section 3 describes the evolutionary clustering algorithm based on that dissimilarity measure. In Sec. 4, we summarize and evaluate the experimental results. Finally, concluding remarks are presented.

2 A Novel Manifold-Distance-Based Dissimilarity Measure

A meaningful measure of distance or proximity between pairs of data points plays an important role in partitional clustering approaches. Most of the clusters can be identified by their local or global characteristics. Through a large amount of observation, we have found the following two consistency characteristics of data clustering:

1. *Local* consistency means that data points close in location will have a high affinity.
2. *Global* consistency means that data points located in the same manifold structure will have a high affinity.

For real-world problems, the distribution of data points takes on a complex manifold structure, which implies that the classical Euclidean distance metric can only reflect local consistency, and fails to describe global consistency. We can illustrate this problem by the following example. As shown in Fig. 1, we expect that the affinity between point *a* and point *e* is higher than the affinity between point *a* and point *f*. In other words, we are looking for a measure of dissimilarity according to which point *a* is closer to point *e* than to point *f*. In terms of the Euclidean distance metric, however, point *a* is much closer to point *f* than to *e*. Hence for complicated real-world problems, simply using the Euclidean distance metric as a dissimilarity measure cannot fully reflect the characteristics of data clustering.

Here, we want to design a novel dissimilarity measure

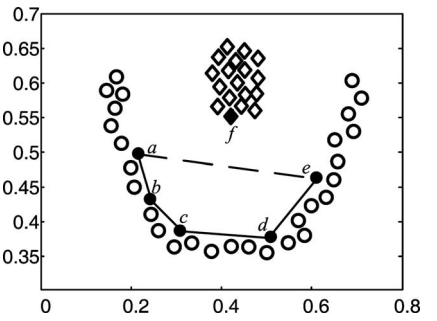

Fig. 1 An illustration of the fact that the Euclidean distance metric can fail to reflect global consistency.

with the ability of reflecting both local and global consistency. As an example, we can observe from the data distribution in Fig. 1 that data points in the same cluster tend to lie in the same manifold.

For our purpose, data points are taken as the nodes V of a weighted undirected graph $G=(V,E)$. Edges $E=\{W_{ij}\}$ reflect the affinity between each pair of data points. We expect to design a dissimilarity measure that assigns high affinity to two points if they can be linked by a path running along a manifold, and low affinity if they cannot. This concept of dissimilarity measure has been shown in experiments to lead to significant improvement in classification accuracy when applied to semisupervised learning.[14,15] The aim of using this kind of measure is to elongate the paths that cross different manifolds, and simultaneously shorten those that do not.

To formalize this intuitive notion of dissimilarity, we need first to define a so-called *manifold length* of line segments. We have found that a distance measure describing the global consistency of clustering does not always satisfy the triangle inequality of the Euclidean metric. As shown in Fig. 1, to describe the global consistency, it is required that the length of a path connected by shorter edges be smaller than that of the direct path, i.e. $\overline{ab}+\overline{bc}+\overline{cd}+\overline{de}<\overline{ae}$. In other words, a direct path between two points is not always the shortest one.

Enlightened by this property, we define a manifold length of line segment as follows.

Definition 1. The manifold length of a line segment (x_i,x_j) is defined as

$$L(x_i,x_j) \triangleq \rho^{\mathrm{dist}(x_i,x_j)} - 1, \tag{1}$$

where $\mathrm{dist}(x_i,x_j)$ is the Euclidean distance between x_i and x_j, and $\rho>1$ is the flexing factor.

Obviously, the manifold length of a line segment possesses the property mentioned, and thus can be utilized to describe global consistency. In addition, the manifold length between two points can be elongated or shortened by adjusting the flexing factor ρ.

According to the manifold length of line segments, we define a new distance metric, called the manifold distance metric, which measures the distance between a pair of points by searching for the shortest path in the graph.

Definition 2. Let data points be the nodes of graph $G=(V,E)$, and $p \in V^l$ be a path of length $l=|p|-1$ connecting the nodes p_1 and $p_{|p|}$, in which $(p_k,p_{k+1}) \in E$, $1 \le k < |p|$. Let $\boldsymbol{P}_{i,j}$ denote the set of all paths connecting data points x_i and x_j. The manifold distance between x_i and x_j is defined as

$$D(x_i,x_j) \triangleq \min_{p \in \boldsymbol{P}_{i,j}} \sum_{k=1}^{|p|-1} L(p_k,p_{k+1}). \tag{2}$$

The manifold distance satisfies the four conditions for a distance metric, i.e., $D(x_i,x_j)=D(x_j,x_i)$; $D(x_i,x_j) \ge 0$; $D(x_i,x_j) \le D(x_i,x_k)+D(x_k,x_j)$ for all x_i,x_j,x_k; and $D(x_i,x_j)=0$ if and only if $x_i=x_j$. As a result, the manifold distance metric can measure the geodesic distance along the manifold, which results in any two points in the same manifold being connected by a lot of shorter edges within the manifold while any two points in different manifolds are connected by a longer edge between manifolds, thus achieving the aim of elongating the distances among data points in different manifolds and simultaneously shortening the distances among data points in the same manifold.

3 Evolutionary Clustering Based on the Manifold Distance

In using EAs to perform clustering tasks, it is necessary to design the individual representation method and the heuristic search operators, and formulate the objective function for optimization.

3.1 *Representation and Operators*

In this study, we consider the clustering problem from a combinatorial optimization viewpoint. Each individual is a sequence of real integer numbers representing the sequence numbers of K cluster representatives. The length of a chromosome is K words, of which the first word (gene) represents the first cluster, the second represents the second cluster, and so on. As an illustration, let us consider the following example.

Example 1. Let the size of the data set be 100, and the number of clusters considered be 5. Then for the individual (6, 19, 91, 38, 64) the 6th, 19th, 91st, 38th, and 64th points are chosen to represent the five clusters, respectively.

This representation method does not mention the data dimension. If the size of the data set is N and the number of clusters is K, then the size of the search space is N^K.

Crossover is a probabilistic process that exchanges information between two parent individuals for generating offspring. In this study, we use the uniform crossover,[16] because it is unbiased with respect to the ordering of genes and can generate any combination of alleles from the two parents.[10,17] An example of the operation of uniform crossover on the encoding employed is the following.

Example 2. Let the two parent individuals be (6, 19, 91, 38, 64) and (3, 29, 17, 61, 6). Randomly generate the mask (1, 0, 0, 1, 0). Then the two offspring after crossover are (6, 29, 17, 38, **64**) and (3, 19, 91, 61, 64). In this case, the first offspring is not (6, 29, 17, 38, **6**) because the 6 in bold would be a repetition; we keep that point unchanged.

Each individual undergoes mutation with probability p_m in the following example.

Example 3. Let the size of the data set be 100, and the number of clusters considered be 5. Then the individual (6, 19, 91, 38, 64) can mutate to (6, $19+\lfloor(100-19)\times \text{random}+1\rfloor$, 91, 38, 64) or (6, $19-\lfloor(19-1)\times \text{random}+1\rfloor$, 91, 38, 64) equiprobably when the second gene is chosen to mutate, where random denotes a uniformly distributed random number in the range $[0, 1)$.

Table 1 Parameter settings for MEC and GAC.

Parameter	MEC	GAC
Maximum number of generations	100	100
Population size	50	50
Crossover probability	0.8	0.8
Mutation probability	0.1	0.1

3.2 *Objective Function*

Each datum is assigned to a cluster representative according to its manifold distance to the cluster representatives. As an illustration, let us consider the following example.

Example 4. Let the 6th, 19th, 91st, 38th, and 64th points represent the five clusters, respectively. For the first point, we compute the manifold distances between it and the 6th, 19th, 91st, 38th, and 64th points. If the manifold distance between the first point and the 38th point is the minimum one, then the first point is assigned to the cluster represented by the 38th point. All the points are assigned in this way.

Subsequently, the objective function is computed as follows:

$$\text{Dev}(C) = \sum_{C_k \in C} \sum_{i \in C_k} D(i,\mu_k), \tag{3}$$

where C is the set of all clusters, μ_k is the representative of cluster C_k, and $D(i,\mu_k)$ is the manifold distance between the ith datum of cluster C_k and μ_k.

3.3 *Manifold Evolutionary Clustering Algorithm*

In MEC, the processes of fitness computation, roulette wheel selection with elitism,[18] crossover, and mutation are executed for a maximum number of generations $G_{\max}$. The best individual in the last generation provides the solution to the clustering problem. The main loop of MEC is as follows:

Algorithm 1: Manifold evolutionary clustering (MEC).

Begin

1. $t=0$

2. randomly initialize population $\boldsymbol{P}(t)$

3. assign all points to clusters as the manifold distance, and compute the objective-function values of $\boldsymbol{P}(t)$

4. if $t<G_{\max}$

5. $t=t+1$

6. select $\boldsymbol{P}(t)$ from $\boldsymbol{P}(t-1)$ using roulette wheel selection with elitism

7. crossover $\boldsymbol{P}(t)$

8. mutate $\boldsymbol{P}(t)$

9. go to step 3

10. end if

11. output the best and stop

End

The initial population in step 2 is initialized to K randomly generated real integer numbers in $[1,N]$ where N is the size of the data set. This process is repeated for each of the P chromosomes in the population, where P is the size of the population.

4 Experimental Study

4.1 *Experimental Setup*

In order to validate the performance of MEC, we first apply it to seven benchmark clustering problems on artificial data sets. The results are compared with those of the K-means algorithm (KM),[6] a modified K-means algorithm using the manifold-distance-based dissimilarity measure (DSKM),[13] and the genetic-algorithm-based clustering technique (GAC) proposed by Maulik and Bandyopadhyay.[8] In all the algorithms, the desired number of clusters is set in advance.

In the second experiment, we solve three artificial texture image classification problems using MEC, GAC, DSKM, and KM.

In the third experiment, we solve the classification problems of one X-band SAR image and one Ku-band SAR image by using MEC, GAC, DSKM, and KM.

In the image classification experiments (the second and third experiments), we use the gray-level cooccurrence matrix (GLCM)[3] method to extract texture features from images. There are many statistics that can be determined from each GLCM, such as angular second moment, contrast, correlation, sum of squares, entropy, and so on. Following Ref. 2, in this study we chose three statistics—dissimilarity, entropy, and correlation—which indicate the degree of smoothness of the texture, the homogeneity, and the correlation between the gray-level pair, respectively. There are four parameters that must be indicated in order to generate cooccurrence data, namely, the interpixel orientation, interpixel distance, number of gray levels, and window size. Typically, the interpixel orientation is set to 0, 45, 90, or 135 deg, since those angles are easiest to implement. Short interpixel distances have typically achieved the best success, so an interpixel distance of 1 is used. This combination of offset and orientation has characterized SAR texture well.[2] The effect of varying the number of gray levels and window size on GLCM statistics has been presented in many references.[2,19] In view of their analysis and fine-tuning experiments, in this study we set the number of gray levels at 16 and the window size at 13×13.

The parameter settings used for MEC and GAC in our experimental study are given in Table 1. For DSKM and KM, the maximum iterative number is set to 500, and the stop threshold is 10^{-10}.

In the first two experiments, the true partitioning is known, we evaluate the performance using two external measures, the adjusted Rand index[10,20,21] and the clustering error.[13]

The adjusted Rand index[20] is a generalization of the Rand index[22] that takes two partitionings as the input and counts the pairwise co-assignments of data between the two partitioning. Given a set of N points $S=\{p_1,p_2,\ldots,p_N\}$, suppose $U=\{u_1,u_2,\ldots,u_K\}$ and $V=\{v_1,v_2,\ldots,v_K\}$ represent two different partitions of the points in S such that $\cup_{i=1}^{K}u_i=\cup_{j=1}^{K}v_j=S$ and $u_i\cap u_{i'}=v_j\cap v_{j'}=\varnothing$ for $1\leq i\neq i'\leq K, 1\leq j\neq j'\leq K$. Suppose that U is the known true partition, and V is a clustering result. Let a be the number of pairs of points in the same class in U and in the same class in V, b be the number of pairs of points in the same class in U but not in the same class in V, c be the number of pairs of points in the same class in V but not in the same class in U, and d be the number of pairs of points in different classes in both partitions. The quantities a and d can be interpreted as agreements, and b and c as disagreements. Then the Rand index is $(a+d)/(a+b+c+d)$. The Rand index lies between 0 and 1; when the two partitions agree perfectly, the Rand index is 1.

A problem with the Rand index is that its expected value for two random partitions does not take a constant value (say zero). The adjusted Rand index proposed by Hubert and Arabie[20] assumes the generalized hypergeometric distribution as the model of randomness, i.e., the partitions U and V are picked at random so that the numbers of points in the classes are fixed. Let n_{ij} be the number of points that are in both class u_i and class v_j. Let $n_{i\cdot}$ and $n_{\cdot j}$ be the numbers of points in class u_i and class v_j respectively. Under the generalized hypergeometric model, it can be shown that

$$E\left[\sum_{i,j}\binom{n_{ij}}{2}\right]=\frac{\Sigma_i\binom{n_{i\cdot}}{2}\cdot\Sigma_j\binom{n_{\cdot j}}{2}}{\binom{n}{2}}. \tag{4}$$

Then the adjusted Rand index is given as

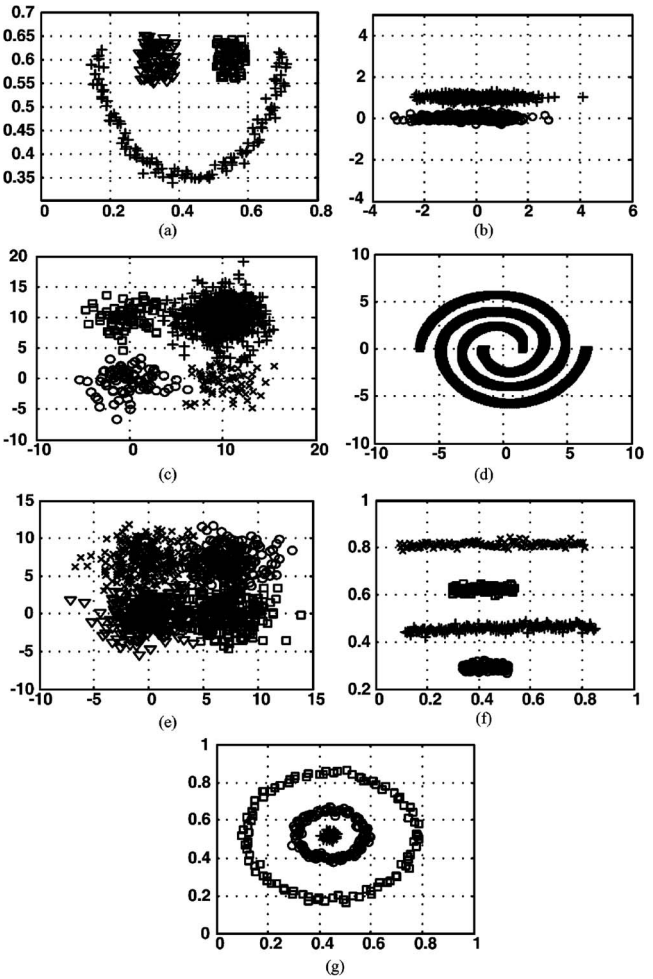

Fig. 2 Typical implementation results on the artificial data sets obtained from MEC: (a) Line-blobs; (b) Long1; (c) Size5; (d) Spiral; (e) Square4; (f) Sticks; (G) Three-circles.

$$R(U,V)=\frac{\Sigma_{i,j}\binom{n_{ij}}{2}-\left[\Sigma_i\binom{n_{i\cdot}}{2}\cdot\Sigma_j\binom{n_{\cdot j}}{2}\right]\Big/\binom{n}{2}}{\frac{1}{2}\left[\Sigma_i\binom{n_{i\cdot}}{2}+\Sigma_j\binom{n_{\cdot j}}{2}\right]-\left[\Sigma_i\binom{n_{i\cdot}}{2}\cdot\Sigma_j\binom{n_{\cdot j}}{2}\right]\Big/\binom{n}{2}}. \tag{5}$$

The adjusted Rand index returns values in the interval [0, 1] and is to be maximized.

Let the known true partition be $U=\{u_1,u_2,\ldots,u_K\}$ and the clustering result be $V=\{v_1,v_2,\ldots,v_K\}$. Then $\forall i,j \in\{1,2,\ldots,K\}$, Confusion (i,j) denotes the number of data points that are both in the true cluster u_i and in the cluster v_j. Then the clustering error is defined as

$$\mathrm{CE}(U,V)=\frac{1}{N}\sum_{i=1}^{K}\sum_{\substack{j=1\\ i\neq j}}^{K}\mathrm{Confusion}(i,j), \tag{6}$$

where N is the size of the data set. Note that there exists a renumbering problem, so the clustering error is computed for all possible renumberings of V, and the minimum is taken. The clustering error also returns values in the interval [0, 1] and is to be minimized.

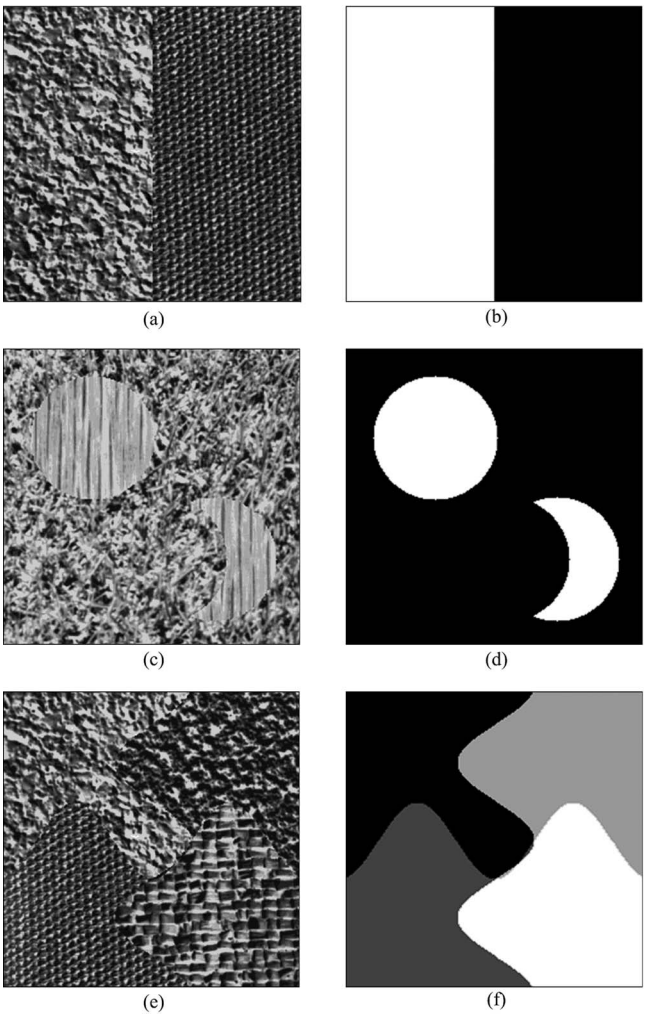

Fig. 3 Artificial texture images and their true partitioning: (a) original Image1; (b) true partitioning of Image1; (c) original Image2; (d) true partitioning of Image2; (e) original Image3; (f) true partitioning of Image3.

4.2 *Implementation Results on Benchmark Clustering Problems*

We first select seven artificial data sets, named Line-blobs, Long1, Size5, Spiral, Square4, Sticks, and Three-circles, to study a range of different interesting data properties. The distribution of data points in these sets can be seen in Fig. 2. We perform 30 independent runs on each problem. The average results of the two metrics, the clustering error and the adjusted Rand index, are shown in Table 2.

From Table 2, we can see clearly that MEC did best on six out of the seven problems, while GAC did best only on the Square4 data set. DSKM also obtained the true clustering on three problems. KM and GAC only obtained the desired clustering for the two spheroid data sets, Size5 and Square4. This is because the structure of the other five data sets does not satisfy convex distribution. On the other hand, MEC and DSKM can successfully recognize these complex

Table 2 Results of MEC, GAC, DSKM, and KM on artificial data sets.

	Clustering error				Adjusted Rand index			
Problem	MEC	GAC	DSKM	KM	MEC	GAC	DSKM	KM
Line-blobs	0	0.263	0.132	0.256	1	0.399	0.866	0.409
Long1	0	0.445	0	0.486	1	0.011	1	0.012
Size5	0.010	0.023	0.015	0.024	0.970	0.924	0.955	0.920
Spiral	0	0.406	0	0.408	1	0.034	1	0.033
Square4	0.065	0.062	0.073	0.073	0.835	0.937	0.816	0.816
Sticks	0	0.277	0	0.279	1	0.440	1	0.504
Three-circles	0	0.569	0.055	0.545	1	0.033	0.921	0.044

clusters, which indicates that the manifold distance metrics are very suitable to measure complicated clustering structures.

In comparisons between MEC and DSKM, MEC obtained the true clustering on Long1, Spiral, Sticks, Line-blobs, and Three-circles in all 30 runs, but DSKM could not do so on Line-blobs and Three-circles. Furthermore, for the Size5 and Square4 problems, MEC did a little better than DSKM in both the clustering error and the adjusted Rand index. The main drawback of DSKM is that it has to recalculate the geometrical center of each cluster with the *K*-means algorithm after cluster assignment, which reduces its ability to reflect global consistency. MEC avoids this drawback by evolutionary searching of the cluster representatives from a combinatorial optimization viewpoint.

In order to show the performance visually, typical simulation results on the eight data sets obtained from MEC are shown in Fig. 2.

4.3 *Implementation Results on Artificial Texture Image Classification*

Image1 is a simple 256×256 bipartite image [Fig. 3(a)]. The original image contains two textures, cork and cotton, selected from the Brodatz texture images.[23] Figure 3(b) represents the true partitioning of Image1. Image2 also contains two textures, as shown in Fig. 3(c), and Fig. 3(d) represents its true partitioning. Image3 is a more complicated synthesized texture image with four classes, and Fig. 3(e) and 3(f) represent the original image and the true partitioning, respectively.

We perform 30 independent runs on each problem. The average results for the two metrics, clustering error and adjusted Rand index, are shown in Table 3. Figures 4–6 are typical implementation results obtained from the four algorithms, MEC, GAC, DSKM, and KM, in clustering the three texture images, respectively.

As shown in Table 3, all the average values of the cluster error obtained from MEC, GAC, DSKM, and KM in clustering Image1 are less than 1%, so all the four algorithms are easily able to segment Image1. The values of the cluster error and adjusted Rand index and Fig. 4 also show that the results obtained from MEC and DSKM are much better than those from of GAC and KM, because both MEC and DSKM assign data according to the manifold distance, while GAC and KM assign data according to the Euclidien distance. However, the computational cost of the manifold distance is much larger than that of the Euclidean distance. MEC and DSKM have similar results in clustering Image1.

In clustering Image2, the average value of the cluster error obtained from MEC is much smaller than the results obtained from GAC, DSKM, and KM, and the average value of the adjusted Rand index of MEC is obviously greater than the results obtained from GAC, DSKM, and KM. So MEC does best on this problem. Figure 5 also

Table 3 Results of MEC, GAC, DSKM, and KM on artificial texture image classification.

	Clustering error				Adjusted Rand index			
Problem	MEC	GAC	DSKM	KM	MEC	GAC	DSKM	KM
Image1	0.0030	0.0069	0.0035	0.0071	0.9462	0.9115	0.9437	0.9113
Image2	0.0037	0.1594	0.0072	0.2017	0.9376	0.9057	0.9109	0.8869
Image3	0.1212	0.2554	0.1858	0.2899	0.8638	0.8012	0.8117	0.8094

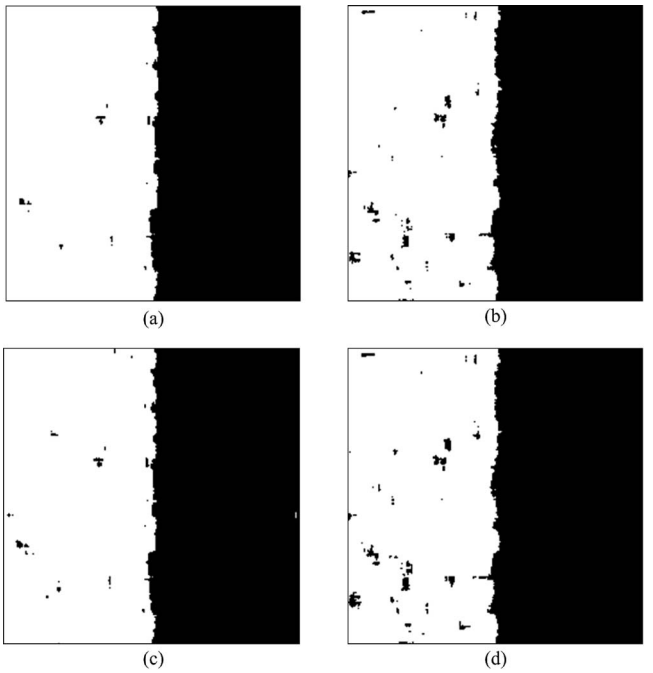

Fig. 4 Typical implementation results obtained from (a) MEC, (b) GAC, (c) DSKM, and (d) KM in clustering Image1.

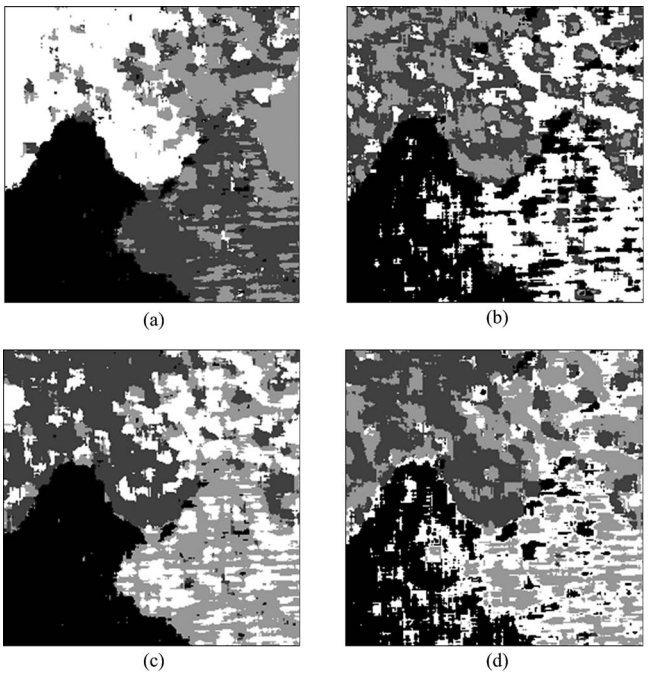

Fig. 6 Typical implementation results obtained from (a) MEC, (b) GAC, (c) DSKM, and (d) KM in clustering Image3.

shows that the MEC result and the DSKM result are obviously better than the GAC result and the KM result, and the MEC result is better than the DSKM result. That MEC segments the two textures better than DSKM may be because MEC searches the two cluster representatives using evolutionary searching but DSKM has to recalculate the geometrical center of each cluster after cluster assignment in each iteration, which reduces its ability to reflect global consistency.

In clustering the more complicated texture image Image3, all the average values of the cluster error are greater than 12%, so none of the four algorithms can segment the image very well based on GLCM features. However, Table 3 and Fig. 6 show that MEC does much better than the other three algorithms.

4.4 *Implementation Results on Remote Sensing Image Classification*

The first image, as shown in Fig. 7(a), is an X-band SAR image of a lakeside in Switzerland. The size of the image is 140×155 pixels. We want to classify the image into three clusters, namely, the lake, the city, and the mountainous region. The second image, as shown in Fig. 7(b), is a

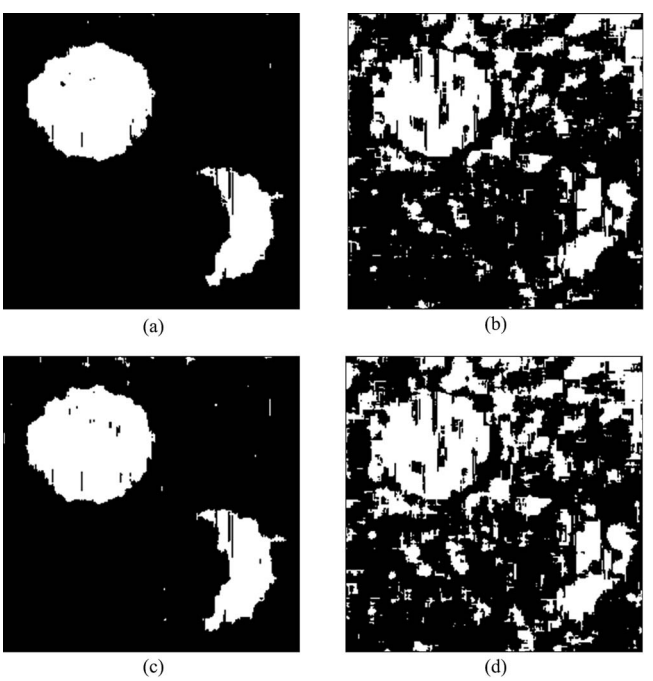

Fig. 5 Typical implementation results obtained from (a) MEC, (b) GAC, (c) DSKM, and (d) KM in clustering Image2.

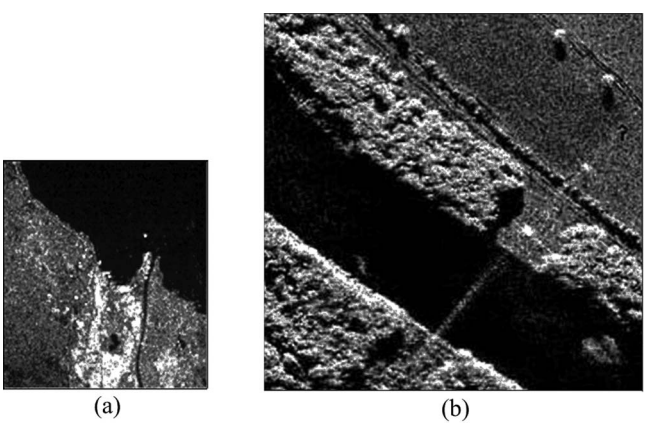

Fig. 7 Original SAR images: (a) X-band, (b) Ku-band.

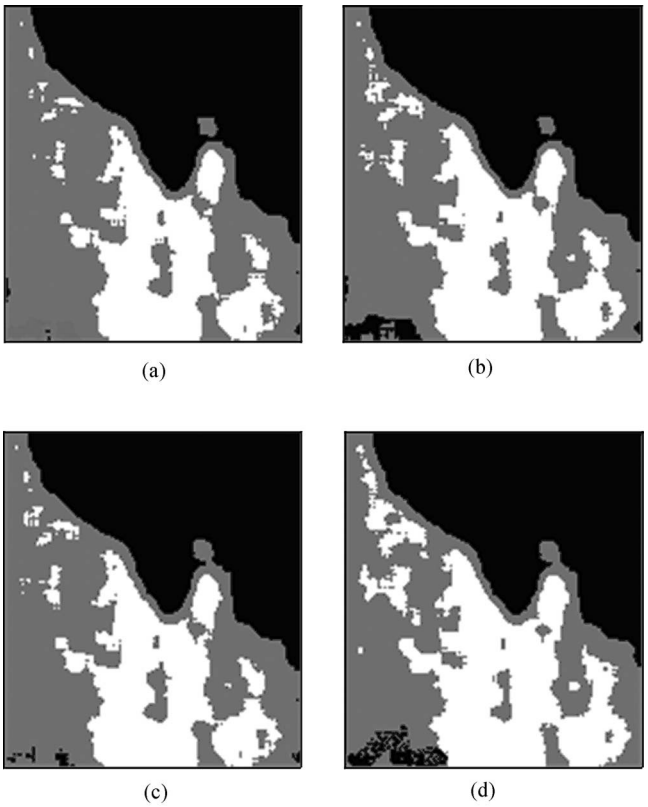

Fig. 8 Implementation results obtained from (a) MEC, (b) GAC, (c) DSKM, and (d) KM in clustering the X-band SAR image.

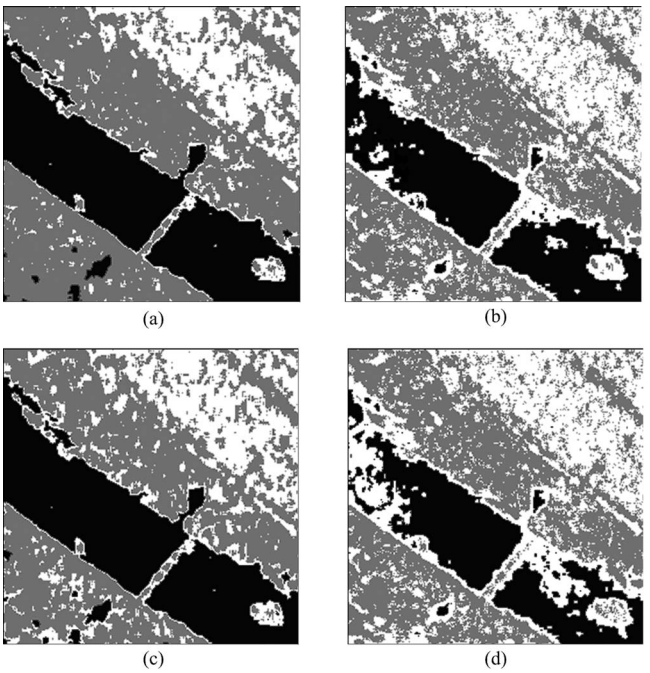

Fig. 9 Implementation results obtained from (a) MEC, (b) GAC, (c) DSKM, and (d) KM in clustering the Ku-band SAR image.

Ku-band SAR image of the Rio Grande River nearby Albuquerque, New Mexico, USA. The size of the image is 256×256 pixels. We want to classify the image into three clusters, namely, the river, the vegetation, and the crop. Figures 8 and 9 show the clustering results obtained from the MEC, DSKM, GAC, and KM in clustering these two SAR images, respectively.

Figure 8 shows that all methods are readily able to perform the classification of the X-band SAR image. Figure 8(b) and 8(d) show that many mountainous regions in the bottom left are identified as lake by KM and GAC. Figure 8(a) and 8(c) show that MEC can recognize these regions and DSKM can reduce the erroneous identifications. Meanwhile, KM badly confuses many mountainous regions in the top left with city regions. MEC largely avoids these errors. Generally speaking, the MEC method outputs better partitioning.

Figure 9 shows that MEC, GAC, DSKM, and KM generate different results, and none of the methods performs as well as on the first SAR image. Generally speaking, the two methods based on the manifold distance generate better partitioning than GAC and KM. The dissimilarity measure based on Euclidean distance tends to confuse the crop with the river. MEC and DSKM generate better partitioning of the river region. In distinguishing the vegetation and crop, the partitioning results of GAC and KM appear more discontinuous than those of MEC and DSKM. GAC and KM tend to confuse the vegetation with the crop along the river, assigning more to the crop than it should. However, MEC and DSKM tend to identify the vegetation in the bottom left as the river, due to the gray level of the lands in that region. DSKM also tends to confuse the vegetation with the crop in the region along the river and in the bottom left of the image. Generally speaking, MEC does better than DSKM, GAC does better than KM, and MEC and DSKM do much better than GAC and KM, in partitioning this Ku-band SAR image.

4.5 *Robustness and Computing Time*

In order to compare the robustness of these methods, we follow the criteria used in Ref. 24. In detail, the relative performance of the algorithm m on a particular data set is represented by the ratio b_m of the mean value of its adjusted rand index (R_m) to the highest mean value of the adjusted Rand index among all the compared methods:

$$b_m = \frac{R_m}{\max_k R_k}. \tag{7}$$

The best method m^* on that data set has $b_{m^*}=1$, and all the other methods have $b_m \leq 1$. The larger the value of b_m, the better the performance of the method m is in relation to the best performance on that data set. Thus the sum of b_m over all data sets provides a good measure of the robustness of the method m. A large value of the sum indicates good robustness.

Figure 10 shows the distribution of b_m of each method over the ten problems. For each method, the ten values of b_m are stacked, and the sum is given on top of the stack. Figure 10 reveals that MEC has the highest sum value. In fact, the b_m values of MEC are equal or very close to 1 on all the test problems, which indicates that MEC performs

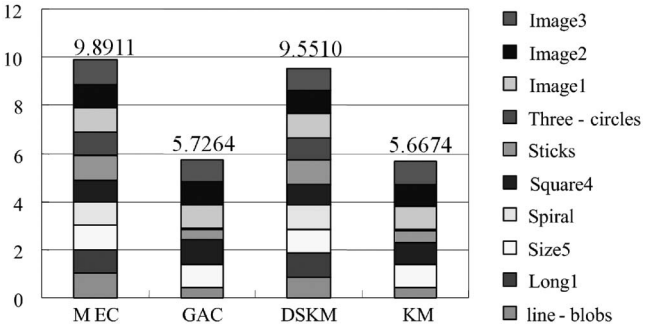

Fig. 10 Robustness of the algorithms compared.

very well in different situations. Thus MEC is the most robust method among the compared methods.

Figure 11 shows the sum of the computing times of the four algorithms in solving the twelve problems on an IBM IntelliStation M Pro 6233. From Fig. 11, it can be seen that the computing time of MEC is obviously longer than the computing time of GAC and KM. The main computational cost of MEC lies in computing the manifold distance between each pair of data points.

5 Concluding Remarks

In this study, we have proposed manifold evolutionary clustering using a novel representation method and a manifold-distance-based dissimilarity measure to perform unsupervised image classification based on texture features. The experimental results on seven artificial data sets with different manifold structure, three artificial texture images, and two SAR images showed that the novel manifold evolutionary clustering algorithm outperformed the KM, GAC, and DSKM in terms of cluster quality and robustness. MEC avoided up the drawbacks of the DSKM by evolutionary searching of cluster representatives from a combinatorial optimization viewpoint instead of recalculating the center of each cluster after cluster assignment.

The manifold evolutionary clustering algorithm is a trade-off of flexibility in clustering data with computational complexity. The main computational cost for the flexibility in detecting clusters lies in searching for the shortest path between each pair of data points, which makes it much slower than GAC and KM. To find a fast or approximate method of computing the manifold distance is part of our future work.

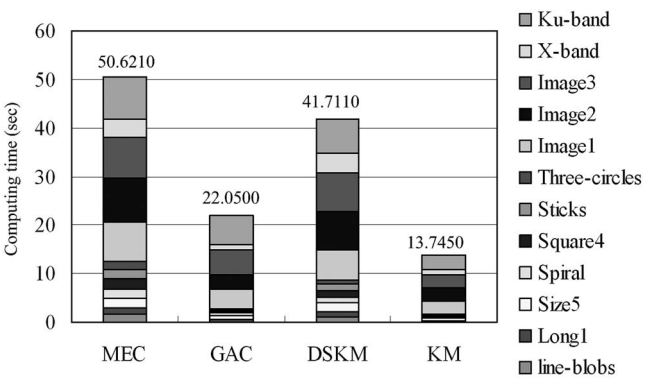

Fig. 11 Computing times of the compared algorithms.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant No. 60703107), the National High Technology Research and Development Program (863 Program) of China (grant No. 2006AA01Z107), and the National Basic Research Program (973 Program) of China (grant No. 2006CB705700). The authors wish to thank the anonymous reviewers for their valuable comments and helpful suggestions, which greatly improved the paper's quality.

References

1. M. Tuceryan and A. K. Jain, "Texture analysis," in *Handbook of Pattern Recognition and Computer Vision*, C. Chen, L. Pau, and P. Wang, Eds., pp. 235–276, World Scientific, Singapore (1993).
2. D. A. Clausi and B. Yue, "Comparing cooccurrence probabilities and Markov random fields for texture analysis of SAR sea ice imagery," *IEEE Trans. Geosci. Remote Sens.* **42**(1), 215–228 (2004).
3. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst. Man Cybern.* **3**(6), 610–621 (1973).
4. H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5), 450–465 (1999).
5. Y. Leung, J. Zhang, and Z. Xu, "Clustering by space-space filtering," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(12), 1396–1410 (2000).
6. J. A. Hartigan and M. A. Wong, "A *K*-means clustering algorithm," *Appl. Stat.* **28**, 100–108 (1979).
7. L. O. Hall, I. B. Ozyurt, and J. C. Bezdek, "Clustering with a genetically optimized approach," *IEEE Trans. Evol. Comput.* **3**(2), 103–112 (1999).
8. U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern Recogn.* **33**(9), 1455–1465 (2000).
9. H. Pan, J. Zhu, and D. Han, "Genetic algorithms applied to multiclass clustering for gene expression data," *Genomics, Peoteomics Bioinf.* **1**(4), 279–287 (2003).
10. J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Trans. Evol. Comput.* **11**(1), 56–76 (2007).
11. M. C. Su and C. H. Chou, "A modified version of the *K*-means algorithm with a distance based on cluster symmetry," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(6), 674–680 (2001).
12. D. Charalampidis, "A modified *K*-means algorithm for circular invariant clustering," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(12), 1856–1865 (2005).
13. L. Wang, L. F. Bo, and L. C. Jiao, "A modified *K*-means clustering with a density-sensitive distance metric," *Lect. Notes Comput. Sci.* **4062**, 544–551 (2006).
14. A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. Eighteenth Int. Conf. on Machine Learning*, pp. 19–26 (2001).
15. O. Bousquet, O. Chapelle, and M. Hein, "Measure based regularization," in *Advances in Neural Information Processing Systems 16 (NIPS)*, MIT Press, Cambridge, MA (2004).
16. G. Syswerda, "Uniform crossover in genetic algorithms," in *Proc. Third Int. Conf. on Genetic Algorithms*, pp. 2–9, Morgan Kaufmann Publishers, San Francisco, CA (1989).
17. D. Whitley, "A genetic algorithm tutorial," *Stat. Comput.* **4**, 65–85 (1994).
18. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA (1989).
19. D. A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Can. J. Remote Sens.* **28**(1), 45–62 (2002).
20. L. Hubert and P. Arabie, "Comparing partitions," *J. Classif.* **2**, 193–218 (1985).
21. K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics* **17**(9), 763–774 (2001).
22. W. Rand, "Objective criteria for the evaluation of clustering methods," *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971).
23. P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover Publications, New York (1966).

24. X. Geng, D. C. Zhan, and Z. H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.* **35**(6), 1098–1107 (2005).

Maoguo Gong is currently an associate professor with the Intelligent Information Processing Innovative Research Team of the Ministry of Education of China at Xidian University, Xi'an, China. He received his BSc degree in electronic engineering from Xidan University, Xi'an, China, in 2003 with the highest honor. He was a master student in the Institute of Intelligent Information Processing, Xidian University, from August 2003 to August 2004. He took the Fund of Excellent Doctor's Dissertation of Xidian University in April 2007. He is a member of the IEEE. His research interests are broadly in the area of computational intelligence. His areas of special interest include artificial immune systems, evolutionary computation, image understanding, data mining, optimization, and some other related areas. He has published round about 30 papers in journals and conferences. (More information at http://see.xidian.edu.cn/iiip/mggong/.)

Licheng Jiao received the BS degree from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the MS and the PhD degree from Xi'an Jiao Tong University, Xi'an, China, in 1984 and in 1990, respectively. His research interests include signal and image processing, natural computation, and intelligent information processing. He is an IEEE senior member, a member of the IEEE Xi'an Section Executive Committee and the chairman of its Awards and Recognition Committee, and an executive committee member of the Chinese Association of Artificial Intelligence. He has charge of about 40 important scientific research projects, and has published more than ten monographs and a hundred papers in international journals and conferences.

Liefeng Bo received his PhD degree from Xidian University, Xi'an, China, in 2007. He currently works as a postdoctoral scholar at Toyota Technological Institute at Chicago (TTI-C), working on human modeling and recognition. (More information at http://ttic.uchicago.edu/~blf0218/index.htm.)

Ling Wang received her bachelor's degree in computational mathematics and her MPhil degree in computer science from Xidian University in 2001 and 2005, respectively. She is currently pursuing a PhD degree at the Institute of Intelligent Information Processing at Xidian University.

Xiangrong Zhang received here BS and MS degrees in computer science and technology from Xidian University, Xi'an, China, in 1999 and 2003, respectively, and the PhD degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2006. She is currently a lecturer at the Institute of Intelligent Information Processing. Her current research interests include SAR image analysis and understanding, pattern recognition, and machine learning.