10

Evolutionary-based Image Segmentation Methods

Licheng Jiao
Xidian University, Xi'an, China

1. Introduction

Nearly 72 years ago, Wertheimer [1] pointed out the importance of perceptual grouping and organization in vision and listed several key factors, such as similarity, proximity, and good continuation, which lead to visual grouping. However, even to this day, many of the computational issues of perceptual grouping have remained unresolved. Since there are many possible partitions of an image into subsets, how do you know which one is right? There are two aspects to be considered here. The first is that there may not be a single correct answer. The second aspect is that the partitioning is inherently hierarchical. Prior literature on the related problems of clustering, grouping and image segmentation is huge. Unfortunately, there is not a general method existing to solve the problem.[2]

Image segmentation is one of the central problems in computer vision and pattern recognition. It refers to the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. The result of image segmentation is a set of segments (sets of pixels) that collectively cover the entire image. Pixels in the same region are similar with respect to some characteristics or computed properties, such as color, intensity, and texture. Adjacent regions are significantly different with respect to the same characteristics. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.[3]

There are many general-purpose approaches available for image segmentation such as threshold methods[4], edge-based methods[5], region-based methods[6], and graph-based methods[7]. Threshold techniques make decisions based on local pixel information. Edge-based methods are based on connecting together broken contour lines. It is prone to failure in the presence of blurring. A region-based method usually partitions an image into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some characteristics. Graph-based techniques generally represent the problem in terms of a graph where each node corresponds to a pixel in the image, and an edge connects each pair of vertices. A weight is associated with each edge based on some property of the pixels that it connects, such as their image intensities. Hybrid techniques using a mix of the methods above are also popular.

What listed above also exposed two basic questions:

- What is the precise criterion for a good segmentation?
- How can such a segmentation be computed efficiently?

In contrast to the heuristic nature of the methods above, one would formalize an objective criterion for evaluating a given segmentation. This would allow us to formulate the segmentation problem as an optimization problem. The objective function that one would seek to optimize is the interclass variance that is used in cluster analysis. An optimizer can lead to efficient solutions for optimal segmentation. But the objective function is usually not a monotone chain, therefore the problem is general NP-hard. Following this way, some clustering methods have been applied to solve image segmentation problems. Among them, K-means algorithm [8] is the most popular and simplest one. It can partition an image into K clusters by using an iterative technique. Although it can be proven that the procedure will always terminate, the K-means algorithm does not necessarily find the most optimal configuration, corresponding to the minimum global objective function. The algorithm is also significantly sensitive to the initial randomly-selected cluster centers. As global optimization techniques, evolutionary algorithms (EAs) are likely to be good tools for image segmentation task. In the past two decades, EAs have been applied to image segmentation with promising results [9-16]. These algorithms exploited the metaphor of natural evolution in the context of image segmentation.

In this chapter, in order to solve the image segmentation problem more efficiently, we propose two evolutionary-bsed image segmentation algorithms with different objective functions.The first is a novel approach based on memetic algorithm (MISA). Watershed segmentation is applied to segment original images into non-overlap small regions before performing the portioning process by MISA. MISA adopts a straightforward representation method to find the optimal combination of watershed regions under the criteria of interclass variance in feature space. After implementing cluster-based crossover and mutation, an individual learning procedure moves exocentric regions in current cluster to the one they should belong to according to the distance between these regions and cluster centers in feature space. In order to evaluate the new algorithm, six texture images, three remote sensing images and three natural images are employed in experiments. The experimental results show that MISA outperforms its genetic version, the Fuzzy c-means algorithm, and K-means algorithm in partitioning most of the test problems, and is an effective approach when compared with two state-of-the-art image segmentation algorithms including an efficient graph-based algorithm and a spectral clustering ensemble-based algorithm. The second is manifold evolutionary clustering (MEC). In MEC, the clustering problem is considered from a combinatorial optimization viewpoint. Each individual is a sequence of real integer numbers representing the cluster representatives. Each data item is assigned to a cluster representative according to a novel manifold distance-based dissimilarity measure which can measure 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 of 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.

In the following sections, we will give the descriptions for the two algorithms in detail. Finally, concluding remarks are presented.

2. Natural and remote sensing image segmentation using memetic computing

2.1 Introduction

This work focuses on image segmentation based on memetic computation. Memetic algorithms (MAs) are one of the recent growing areas in evolutionary computation [17]. They are now widely considered as population-based metaheuristic search approaches which may be regarded as a marriage between an evolutionary or any population-based global search and local improvement procedures. Recently, MAs have been well used across a wide range of problem domains. A lot of studies have demonstrated that MAs converge to high-quality solutions more efficiently than their conventional counterparts in many real-world applications [17-32]. MAs have also been applied in the image processing field [33-38]. For example, Fernandez, Garana and Cabello [33] proposed a MA-based method for the correction of illumination inhomogeneities in images. Batenburg [34] designed an EA with hillclimb operator for finding a binary image that satisfies prescribed horizontal and vertical projections. Tirronen, Neri et al. [35] studied the defect detection in paper production by means of image-processing techniques based on memetic differential evolution frameworks. Gesù, Bosco et al. [36,37] introduced a new memetic approach for the reconstruction of binary images. Zhang, Wang and Zhang [38] proposed a novel image watermarking scheme using a MA and wavelet transform.

In this method, in order to solve the image segmentation problem more efficiently, we propose a MA-based approach, Memetic Image Segmentation Algorithm (MISA), and compare the new method with its genetic version (MISA without learning), the K-means algorithm [8], fuzzy c-means algorithm [39], and two state-of-the-art image segmentation algorithms including an efficient graph-based algorithm [7] and a spectral clustering ensemble-based algorithm [40] in segmenting artificial texture images, remote sensing images and natural images.

The rest of this section is organized as follows: Section 1.2 describes some related background including the technologies used in preprocessing the original image and the brief background of MAs. Section 1.3 describes the proposed MA-based image segmentation algorithm. In Section 1.4, we summarize and evaluate the experimental results.

2.2 Related background

Before performing the portioning process by the proposed MISA, some preprocessing should be done on original images, including feature extraction based on the gray-level co-occurrence matrix (GLCM) [41] and wavelet decomposition [42], and over segmenting object images into small regions by watershed segmentation algorithm [6]. In this section, we will introduce these techniques briefly. The MAs will also be introduced in this section.

2.2.1 GLCM based statistic features

GLCM was frequently used in texture analysis and extraction for images [41,43,44]. Texture features are demonstrated by the statistics over the GLCM. Four usually used statistics are angular second moment (also called energy), correlation, entropy, and inverse difference moment (also called inertia). In this method, the 14 statistics, including the previous four suggested by Haralick, Shanmugam and Dinstein [41] are exploited. There are four parameters that must be indicated to generate a GLCM, i.e., the interpixel orientation, distance, grey level quantization, and window size. Here we set the interpixel orientation to $0°$ for convenient

calculation. Short interpixel distances typically generate the preferred texture features in image analysis, so we set interpixel distance to 1. The role of different values for gray levels and windows size with respect to statistics from GLCM has been investigated in many literatures [43, 44]. According to their analysis and fine-tuned experiments, in this study, we set the image quantization to 16 and the window size to 9×9 .

2.2.2 Wavelet energy features

Wavelet transform has the ability to examine a signal at different scales [42]. In this section, the undecimated wavelet-based feature vector composed by the energies of the subband coefficients is used. Here we implement three-level wavelet decomposition on each square local area with size of 16×16 . The features of each pixel can be represented as a 10-dimension vector $(e_{LL-1}, e_{LH-1}, e_{HL-1}, e_{HH-1}, e_{LH-2}, e_{HL-2}, e_{HH-2}, e_{LH-3}, e_{HL-3}, e_{HH-3})$, in which, for example e_{LL-1} denotes the energy of the LL subimage in the first level. Wavelet transform generates localized spatial and spectral information simultaneously. The energy of the low-pass subimage e_{LL-1} describes the spectral information. The other features except e_{LL-1} characterize the textural properties. Incorporating all the features will enhance the performance.

2.2.3 Watershed segmentation

We use watershed segmentation [6] to over segment object image into small regions. Each region is almost homogeneous in feature space, while neighboring-regions possess different characters. We operate these regions as a basic unit during most of MISA unless it is necessary to come back to pixels for precise evaluation. The basic concept of watershed segmentation is described as follows.

In an image, ideal step edges do not often exist since every edge is blurred to some contents. A blurred edge can be modeled by a ramp. For a ramp edge, a usual gradient operator will generate a slope of the edge. Thus, the ramp edge cannot be separated from noise if the slope of the edge is small. Wang proposed a multi-scale gradient operator to solve the above problem [45]:

$$MG(f) = \frac{1}{n}\sum_{i=1}^{n}\left[\left((f \oplus B_i) - (f \Theta B_i)\right)\Theta B_{i-1}\right] \tag{1}$$

where $\oplus$ and Θ denote dilation and erosion respectively, and B_i is called structural element of size $(2i-1)\times(2i-1)$, and f is the original image.

If watershed regions are too large, big ones may contain more than one focused subject in image, so that the texture feature in that region may not be homogeneous. If the watershed regions are too small, the computational complexity will increase. This is because there will be more basic units to operate during most of MISA. In order to control the number of watershed regions, we use the watershed segmentation algorithm with markers [6]. In this method, a threshold can be adjusted to get an expected number of regions. In this method, the number of regions is about 1500 in a 256×256 image.

2.2.4 Memetic algorithms

Evolutionary algorithms perform well for global searching because they are capable of quickly finding and exploiting promising regions of search space, but they take a relatively

long time to converge to a local optimum. Local improvement procedures quickly find the local optimum of a small region of the search space, but are typically poor global searchers. Thus, several researchers proposed hybrid algorithms combining excellent global exploration characteristics of EAs and efficient refinement capabilities of local search algorithms [46,47]. These hybrid algorithms are known as memetic algorithms (MAs).
MAs assume that combining the features of different methods in a complementary fashion may result in more robust and effective optimization tools. They are population-based meta-heuristic search approaches that have been receiving increasing attention in recent years. They are inspired by Darwinian's principles of natural evolution and Dawkins' notion of a meme defined as the basic unit of cultural transmission. In MAs, a meme is generally considered as an individual learning procedure capable of performing local refinements. Over the past 15 years, MAs have been a hot topic in the fields of both computer science and operational research [17-38]. The typical issues pertinent to MA design [17] include (i) how often individual learning should be applied, (ii) on which solutions individual learning should be used, (iii) how long individual learning should be run, (iv) what maximum computational budget to allocate for individual learning, and (v) what individual learning method or meme should be used for a particular problem, sub-problem or individual.

2.3 The proposed memetic algorithm for image segmentation

2.3.1 Preprocessing and representation

Preprocessing is the first step of our method. In this procedure, we use gray-level co-occurrence matrix and wavelet decomposition for feature extraction of object images. This is done for every pixel in the object image. Some algorithms (such as [15]) operate pixels directly as basic units, which will lead to high computational cost, especially for large images. To reduce the computational complexity, we employ watershed segmentation to segment images into non-overlap small regions. MISA operates these regions instead of every pixel as the basic unit during most of its evolution unless it is necessary to return back to pixels for more precise computation in fitness calculation, learning and crossover operations. Fig.1 illustrates the watershed segmentation process.

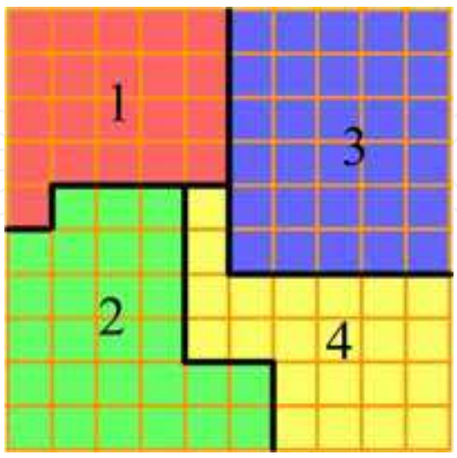


Fig. 1. Illustration of watershed segmentation

In Fig.1, there is an example of an image of 10×10 pixels with each small square divided by the orange lines representing a pixel. We can see that this image contains 4 regions marked

by different colors as red, blue, yellow, and green. The watershed segmentation will segment the image into four regions divided by the black boldface lines. Then, these four regions are the basic unit that we operate during most of MISA.

After preprocessing, we get the following data from the object image: (i) feature matrix, which stores the features of each pixel and is used in initialization with minimum spanning tree, learning and fitness calculation; (ii) coordinate of pixels, which are used in fitness calculation; (iii) pixel number of each region, which is used in learning; (iv) dissimilarity matrix, which is used in initialization with minimum spanning tree.

Many EC-based clustering algorithms [48-51] have used an indirect representation approach that borrows from the K-Means algorithm - the representation codes are for cluster center only, and each data item is subsequently assigned to a cluster representative according to an appointed dissimilarity measure. The most popular dissimilarity measure is the Euclidean distance. By using Euclidean distance as a measure of dissimilarity, these evolutionary clustering methods as well as the K-Means algorithm have good performance on the data set with compact super-sphere distribution, but tend to fail on the data set organized in more complex and unknown shapes. This indirect representation approach can not overcome the main drawback of the K-means algorithm unless a more flexible dissimilarity measure is adopted [15]. Furthermore, this indirect representation approach is not suitable for individual evolutionary and learning operation. In this study, a straightforward encoding method is used here for image segmentation problem. In this coding, every locus represents the corresponding region in the image, and the gene on each locus is the cluster label of the corresponding region.

In our method, coding length (CL) in MISA is the number of regions obtained from watershed segmentation of the object image. The serial labels of locus present the corresponding regions in image, that means the i-th locus represents the i-th regions. The segmentation of an image is expressed by encoding individual with cluster labels of regions. Cluster label of each region is put on its corresponding locus as the gene. Fig. 2 illustrates this code structure.

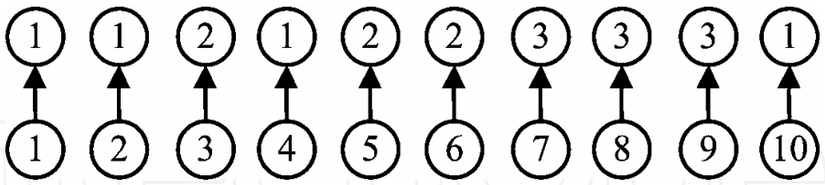


Fig. 2. Illustration of coding structure

Here we exhibit the code of an image which owns 10 regions. The lower numbers denote the region labels. They are arranged in order from 1 to 10. The upper numbers are cluster labels corresponding to each region. In this individual, regions 1, 2, 4 and 10 belong to cluster 1, regions 3, 5 and 6 belong to cluster 2, and regions 7, 8 and 9 belong to cluster 3. This straightforward representation method is very convenient for evolutionary and learning operation as shown in Section 3.2.

In order to guide the evolution to a meaningful direction related to optimal segmentation, Euclid distances in feature space between every pixel and the centroids of the cluster it belongs to are calculated. Then, we sum up all these distances and use the reciprocal of this summation as the fitness of MISA. This segmentation fitness is defined in Equation (2).

$$fitness = \frac{1}{\sum_{i=1}^{CN}\sum_{j=1}^{RN_i}\sum_{k=1}^{PN_j} dis(p_{ijk}, m_i)} \tag{2}$$

where RN_i is the number of regions in the *i*-th cluster, PN_j is the number of pixels in the *j*-th region, p_{ijk} denotes the *k*-th pixel in the *j*-th region of the *i*-th cluster, and m_i is the centroid of all pixels of the *i*-th cluster in feature space. Although we operate watershed regions of image during most of the evolution, it is necessary to return to every pixel, so that we can get a more precise segmentation evaluation. In Equation (2),

$$dis(p_{ijk}, m_i) = \sqrt{\sum_{g=1}^{FN}\left(p_{ijk_g} - m_{i_g}\right)^2} \tag{3}$$

FN is the number of features and p_{ijk_g} and m_{i_g} denote the *g*-th feature of the current pixel and cluster centroid respectively. Therefore, the aim of MISA is to find an individual with maximum fitness.

2.3.2 Memetic image segmentation algorithm

The main loop of MISA is as follows.

Algorithm 1.1. Memetic Image Segmentation Algorithm (MISA)

Step 1. **Preprocessing**. Feature extraction based on GLCM and wavelet decomposition, and over segmenting object image into non-overlapping small regions by watershed segmentation.

Step 2. **Initialization**. Giving the termination criterion, setting the initial parameters, generating a diverse population.

Step 3. **Crossover and Mutation**. A cluster-based crossover operator and random mutation operator are implemented in this step.

Step 4. **Learning**. A supervised local searching strategy is used to improve some individuals with probability.

Step 5. **Evaluation and Selection**. Tournament selection with elitism strategy is used after fitness calculation.

Step 6. **Termination Test:** If termination criterion is satisfied, export the individual with the highest fitness and return the corresponding image segmentation result, stop the algorithm; otherwise, go to **Step 3**.

2.3.2.1 Initialization

The following parameters have to be set in this step: population size (*PS*), the proportion of initial individuals generated using minimum spanning tree (*MSTP*), crossover probability (*CP*), mutation probability (*MP*), learning intensity (*LI*), and cluster number (*CN*). The termination criterion is to run until the number of generations reaches the maximum value (*Gmax*).

The initial population is generated by two different methods to get a diverse population. Some initial individuals are generated based on minimum spanning tree (MST), and the others are generated randomly. The details are as follows:

$\lceil MSTP \times PS \rceil$ individuals are generated based on minimum spanning tree (MST). The idea of MST is introduced to find a meaningful distribution of all regions. Dissimilarity matrix

from preprocessing is used to build a complete MST by Prim's algorithm. In dissimilarity matrix, the number of the *i*-th row and the *j*-th column represents the dissimilarity degree of the *i*-th region to the *j*-th region. The smaller of the number is, the closer between the *i*-th region and the *j*-th region. Here the dissimilarity degree between the *i*-th region r_i and the *j*-th region r_j is calculated in Equation (4).

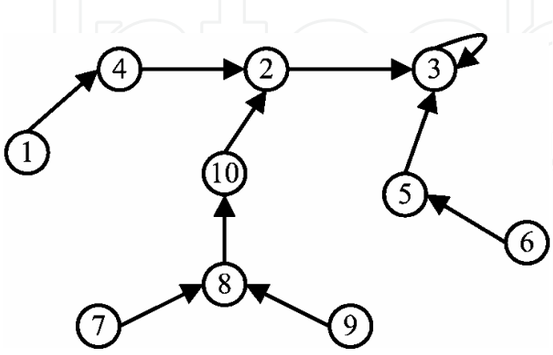
$$Dissimilarity\left(r_i, r_j\right) = \sqrt{\sum_{k=1}^{FN}\left(r_{i_k} - r_{i_k}\right)^2} \tag{4}$$

Here FN is the number of features and r_{i_k} and r_{i_k} are the value of the *k*-th feature of r_i and r_j.

Based on MST, we break up links selected randomly to produce individuals. The whole MST represents one cluster which includes all the regions in the image; and removing n links leads to n more clusters. Breaking up different links will produce different distribution among clusters. Fig. 3 and 4 show this procedure with dissimilarity matrix given in Table 1.

Region No.	1	2	3	4	5	6	7	8	9	10
1	Inf	0.7	4.67	0.69	4.73	4.72	4.63	2.56	2.44	2.46
2	0.7	Inf	4.42	0.58	4.53	4.52	4.33	2.28	2.25	2.21
3	4.67	4.42	Inf	4.63	0.53	0.67	6.27	4.79	5.29	4.99
4	0.69	0.58	4.63	Inf	4.75	4.77	4.58	2.5	2.45	2.31
5	4.73	4.53	0.53	4.75	Inf	0.29	6.17	4.76	5.25	4.98
6	4.72	4.52	0.67	4.77	0.29	Inf	6.19	4.78	5.26	5.03
7	4.63	4.33	6.27	4.58	6.17	6.19	Inf	2.16	2.22	2.41
8	2.56	2.28	4.79	2.5	4.76	4.78	2.16	Inf	0.64	0.52
9	2.44	2.25	5.29	2.45	5.25	5.26	2.22	0.64	Inf	0.73
10	2.46	2.21	4.99	2.31	4.98	5.03	2.41	0.52	0.73	Inf

Table 1. An example of dissimilarity matrix



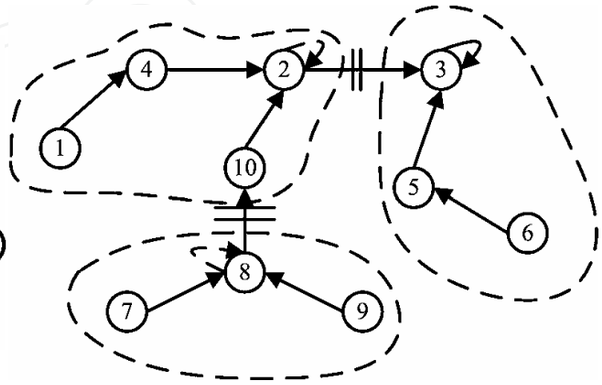


Fig. 3. MST generated from dissimilarity matrix Fig. 4 Clusters generated from breaking up links

Based on dissimilarity matrix shown in Table 1, we can get a Prim MST starting from region 3, which is illustrated in Fig. 3. If we break up the links between 2 and 3, 8 and 10, we can get a cluster distribution as $\{1,2,4,10\}$, $\{3,5,6\}$ and $\{7,8,9\}$, as shown in Fig. 4.

The rest $PS-\lceil MSTP\times PS\rceil$ individuals are generated randomly, namely, for each locus and we put a random cluster label ranges from 1 to CN as its gene.

2.3.2.2 Crossover and Mutation

Based on the meaning of crossover that children should get part of the information from parents, we create a novel crossover operator, called cluster-based crossover (CC). In this algorithm, the role of crossover is to change the cluster distribution of one parent individual by the cluster distribution information of the other parent individual. To make the crossover meaningful for segmentation, we try to cross corresponding clusters of two parents, individuals A and B. For example, we divide an image into three clusters, where A is composed of clusters α_1, α_2, α_3 and B is composed of clusters β_1, β_2, β_3. Here $\alpha_1=\{a_x \mid a_x \in \text{cluster 1}\}$, $\alpha_2=\{a_y \mid a_y \in \text{cluster 2}\}$, $\alpha_3=\{a_z \mid a_z \in \text{cluster 3}\}$, $\beta_1=\{b_x \mid b_x \in \text{cluster 1}\}$, $\beta_2=\{b_y \mid b_y \in \text{cluster 2}\}$, $\beta_3=\{b_z \mid b_z \in \text{cluster 3}\}$, $(x\in(1,CL),\ y\in(1,CL),\ z\in(1,CL), x\neq y\neq z)$. First, mean values of every feature of each pixel in each cluster are calculated. Then, the modulus values of these mean values for each cluster are sorted from small to large. This calculation is similar to Equation (4) in section 3.2.1. For example, if the orders of clusters in A and B are $\alpha_2\alpha_1\alpha_3$ and $\beta_3\beta_2\beta_1$ respectively, then, α_2 will be crossed with β_3, α_1 with β_2, and α_3 with β_1. This idea is similar to multi-point crossover. The number of 'points' equals to the number of clusters. For convenient operation, we re-label $\alpha_2\alpha_1\alpha_3$ and $\beta_3\beta_2\beta_1$ as $\alpha_1\alpha_2\alpha_3$ and $\beta_1\beta_2\beta_3$ respectively, without removing regions of any cluster.

Two individuals are chosen from the population randomly, and then the corresponding clusters between them are fixed and then crossed as mentioned above. The detailed steps are described as follows, where nc is the number of regions to be "crossed" in current cluster and α_{i_j} denotes the j-th region in cluster α_i and CN denotes the cluster number.

Algorithm 1.2. Cluster-based Crossover (CC)

***Crossover*: A cross B**

Step 1. Set $i=1$

Step 2. If $i\le CN$, go to **Step 3**, otherwise, **stop**

Step 3. Set $nc=$ random number from 1 to the size of the i-th cluster in A.

Step 4. Select nc regions randomly from the i-th cluster of A, set $j=1$.

Step 5. If α_{i_j} does not exist in β_i, then put i on the α_{i_j}-th gene of B

Step 6. $j=j+1$, if $j\le nc$ go to **Step 5**, otherwise, $i=i+1$, go to **Step 2**.

In CC, A crosses B and B crosses A, producing two children. Both of them inherit parts of the information from the two parents. We illustrate CC in Fig. 5.

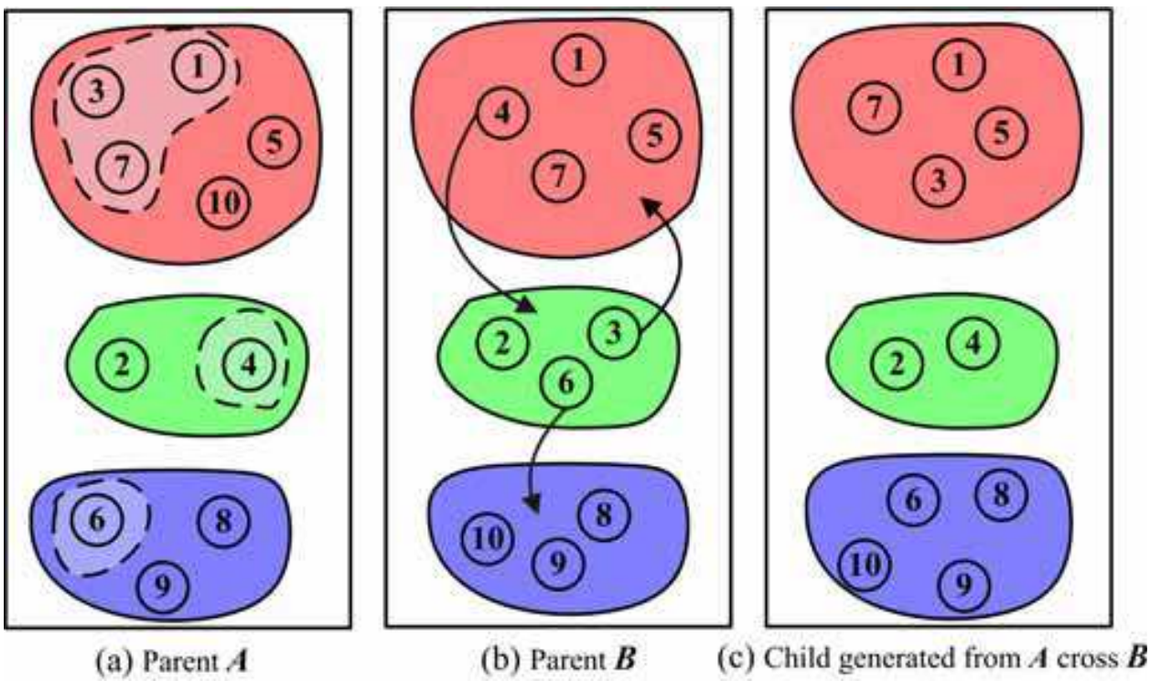


Fig. 5. An illustration of cluster-based crossover

Fig. 5 shows that in individual A, the cluster 1 to 3 are $\{1,3,7,5,10\}$, $\{2,4\}$, and $\{6,8,9\}$, respectively. In individual B, the cluster distribution is $\{1,4,5,7\}$, $\{2,3,6\}$ and $\{8,9,10\}$. For example, we begin by selecting 3 regions in cluster 1 of A, which is $\{1,3,7\}$ with light red shadow. In individual B, region 1 and 7 are in cluster 1, but region 3 exists in cluster 2, so we move region 3 from cluster 2 to cluster 1. In the same way, after all the same procedures are applied on the rest clusters, we get the child individual illustrated in Fig.5(c).

In the mutation phase, we change the cluster distribution randomly as follows, where A denotes an individual, A_j is the j-th gene of A. MP denotes mutation probability, CN denotes the cluster number, and CL denotes coding length.

Algorithm 1.3. Mutation

Step 1. Set $j = 1$

Step 2. If $j \le CL$, set r = a random number range from 0 to 1, $i = A_j$, go to **Step 3**; Otherwise, **stop**

Step 3. If $r < MP$, set l = a random number range from 1 to $CN \wedge l \neq i$.

Step 4. $A_j = l$, $j = j + 1$, go to **Step 2**.

In mutation, if a gene should be mutated due to probability, it will change into another cluster number generated randomly. This means the region corresponding to this locus will be moved into another cluster.

2.3.2.3 Individual Learning Method

Within the computational intelligence community, research on MA has since grown significantly and the term has come to be associated with the pairing of meta-heuristics or population-based methodologies with a separate lifetime learning process that materializes

in various forms of individual learning methods or memes, which are generally regarded as lifetime learning procedures capable of generating refinement on given individual(s). [32] In MISA, we introduce an individual learning method based on cluster distribution as is described in Algorithm 1.4. The basic idea is firstly, identify the most distinct region in each cluster. Then, for each of these regions, we find the cluster with the most similar average feature values, and assign the region to this cluster. In Algorithm 1.4, AG_i is the average feature values of the *i*-th cluster and AG_{im_i} is the average feature values of the m_i -th region in the *i*-th cluster where *PS* denotes population size, *CN* denotes the cluster number, *LI* denotes learning intensity, and *CL* denotes coding length.

Algorithm 1.4. Individual Learning

Step 1. Set $a = 1$

Step 2. If $a \le PS$, go to **Step3**, otherwise **stop.**

Step 3. For the a-th individual in population, r is a random number from 0 to 1, if $r > 0.5$, go to **Step4**, otherwise, $a = a + 1$, go to **Step3**.

Step 4. Set $c = 1$.

Step 5. For $1 \le i \le CN$, find $m_i = \arg\max \left\| AG_i - AG_{im_i} \right\|_2$.

Step 6. For $1 \le i \le CN$, move the m_i -th region to cluster $j = \arg\min \left\| AG_j - AG_{im_i} \right\|_2$. If $i = j$, that means the m_i -th region doesn't need to be moved.

Step 7. $c = c + 1$. If $c \le \lfloor LI \times CL \rfloor$, go to **Step 5**, otherwise, $a = a + 1$, go to **Step 2**.

For more precise evaluation, we return back to pixels to calculate AG_i instead of using the average feature values of each region in cluster i directly, which is shown in Equation (5). As described above, AG_i is a vector that comprises *FN* features.

$$AG_i = \frac{\sum_{j=1}^{RN_i} \sum_{k=1}^{PN_j} p_{ijk}}{\sum_{j=1}^{RN_i} PN_j} \tag{5}$$

where RN_i is the number of regions in the *i*-th cluster, PN_j is the number of pixels in the *j*-th region, p_{ijk} denotes the feature vector of the *k*-th pixel in the *j*-th region of the *i*-th cluster, defined as Equation (6).

$$p_{ijk} = \left(p_{ijk_1}, p_{ijk_2}, \ldots, p_{ijk_{FN}} \right) \tag{6}$$

Fig. 6 illustrates an example of the individual learning procedure.

In Fig. 6, $\{1,2,3,4,5\}$, $\{6,7,8,9,10,11\}$ and $\{12,13,14,15,16\}$ demonstrate the three clusters before learning, we label them as cluster1, cluster2 and cluster3, respectively. After the calculation shown in Step 5 and Step 6 of Algorithm 1.4, region 4, 8 and 12 are shown to be the most distinct regions of each cluster. And cluster3, cluster1, and cluster2 are the ones with most similar features for region 4, 8 and 12 respectively. Therefore, the learning processing will remove region 4, 8 and 12 to cluster3, cluster1 and cluster2 respectively.

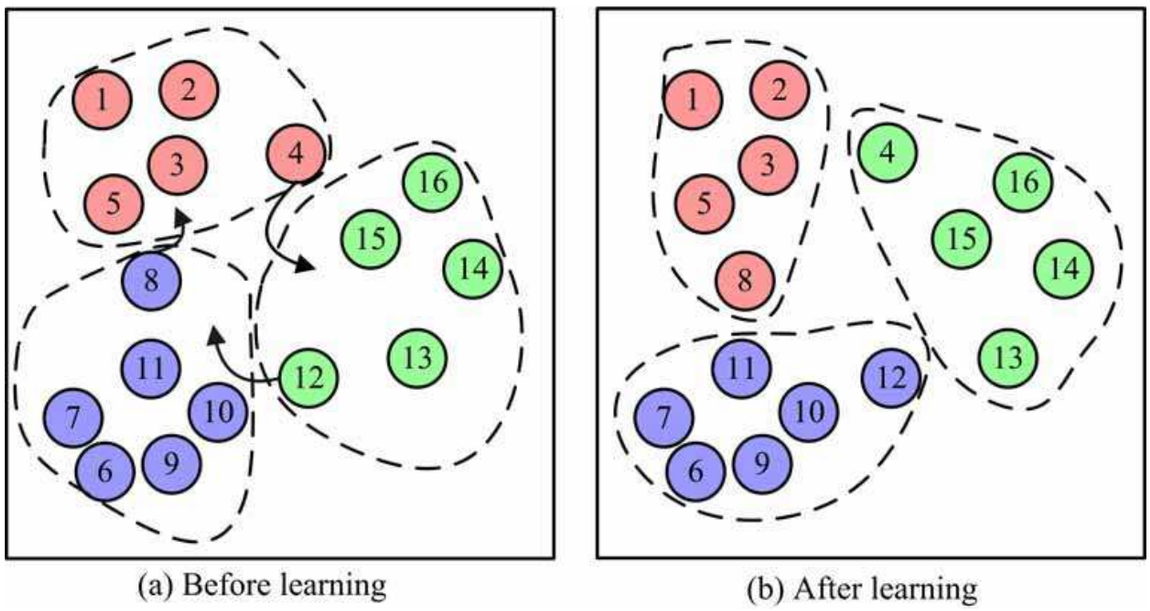


Fig. 6. Illustration of an example of the individual learning procedure

From the description above, we can see that half of the whole population is selected randomly each time to learn. For each individual in learning, $\lfloor LI \times CL \rfloor$ regions are detected. LI should be adjusted properly as if it is too small, the learning's effect is not enough. However, it is also unnecessary to set the LI too big. This is because learning is a local searching strategy so when it comes to the local optimum, the following learning will not change any cluster, which makes it meaningless and a waste of the computation.

2.4 Experimental study

2.4.1 Experimental setup

In order to validate the performance of MISA, we apply it to twelve image segmentation problems including six artificial texture images, three remote sensing images and three natural images. The results will be compared with the K-means algorithm (KM) [8], Fuzzy c-means algorithm (FCM) [39], the genetic image segmentation algorithm (GISA, MISA without learning), and two state-of-the-art image segmentation algorithms including an efficient graph-based image segmentation algorithm (EGSA) [7] and the spectral clustering ensemble algorithm (SCEA) [40].

Based on the parameter sensitivity analysis (as described in Section 1.4.5), for MISA, the parameters are set as follows: The maximum number of generations is 50, population size is 30, MST initialization proportion is 0.6, crossover probability is 0.8, mutation probability is 0.005, and learning intensity is 0.5. For GISA, the parameters are set as follows: The maximum number of generations is 200, population size is 30, MST initialization proportion is 0.6, crossover probability is 0.8, and mutation probability is 0.005. Under these parameter settings, the computational costs of MISA and GISA are about equivalent. For KM and FCM, the maximum iterative number is set to 500, and the stop threshold is 10^{-10}. The software of EGSA is downloaded directly from the author's homepage (http://people.cs.uchicago.edu/~pff/segment/). SCEA was proposed by us in 2008. We

will perform it under the tuned parameters. Here 30 component spectral clustering with Nyström method are combined, and 100 regions are sampled in each component spectral clustering algorithm. The scaling parameter for each component is randomly selected from the interval [3, 12].

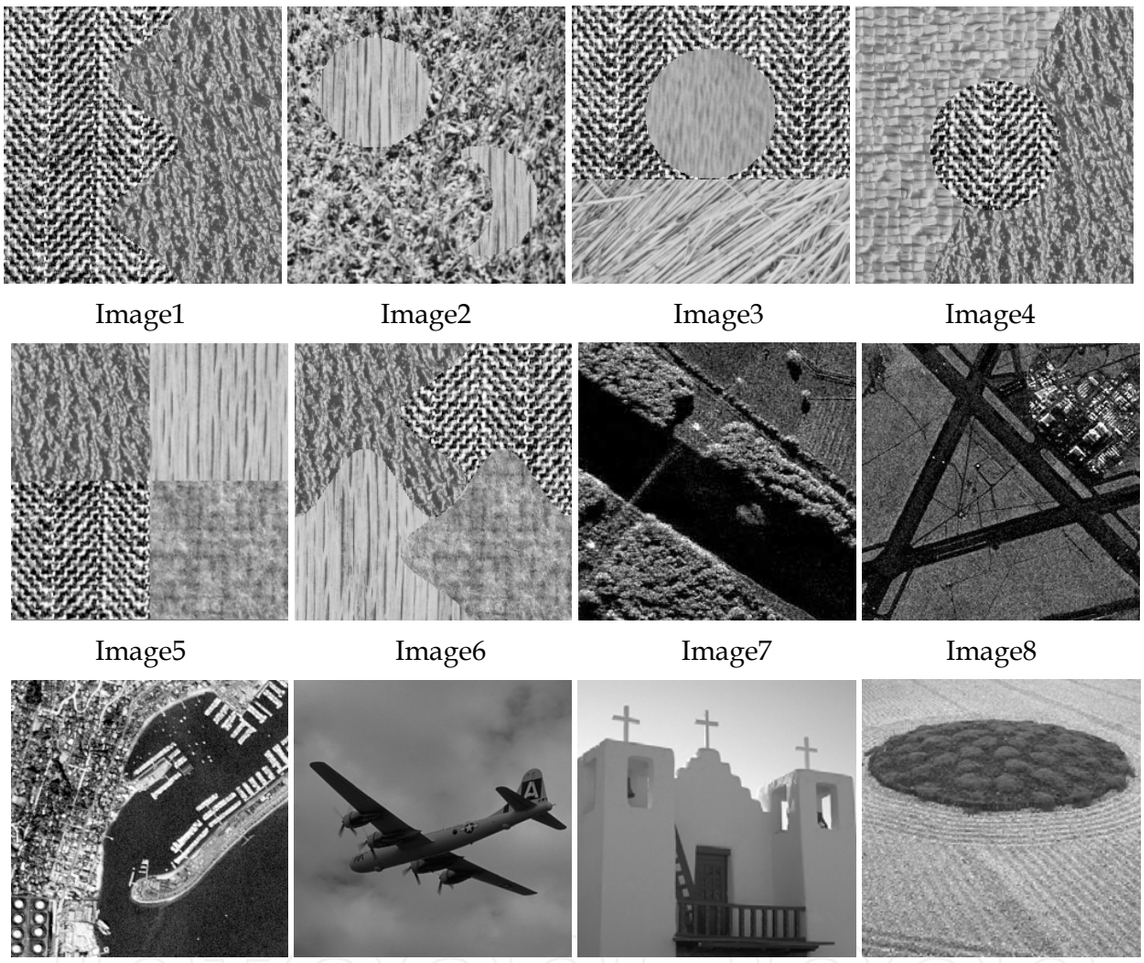


Fig. 7. The original images used in this study

The original images are shown in Fig. 7. Image1 to Image6 are artificial texture images with the size of 256×256 from the Brodatz texture images [52]. Their standard classification images are shown in Fig. 8. Image7 to Image9 are remote sensing images. Image7 is a Ku-band SAR image of the Rio Grande River nearby Albuquerue, New Mexico, USA. Image8 is a Ku-band image of the China Lake Airport, California, USA. Image9 is an optical remote sensing image of Shelter Island, San Diego. The sizes of these images are all 256×256, too. Image10 to Image12 are natural images. The sizes of them are 256×256, 320×320, and 330×320, respectively.

The watershed segmentation results of the twelve test images are shown in Fig. 9.

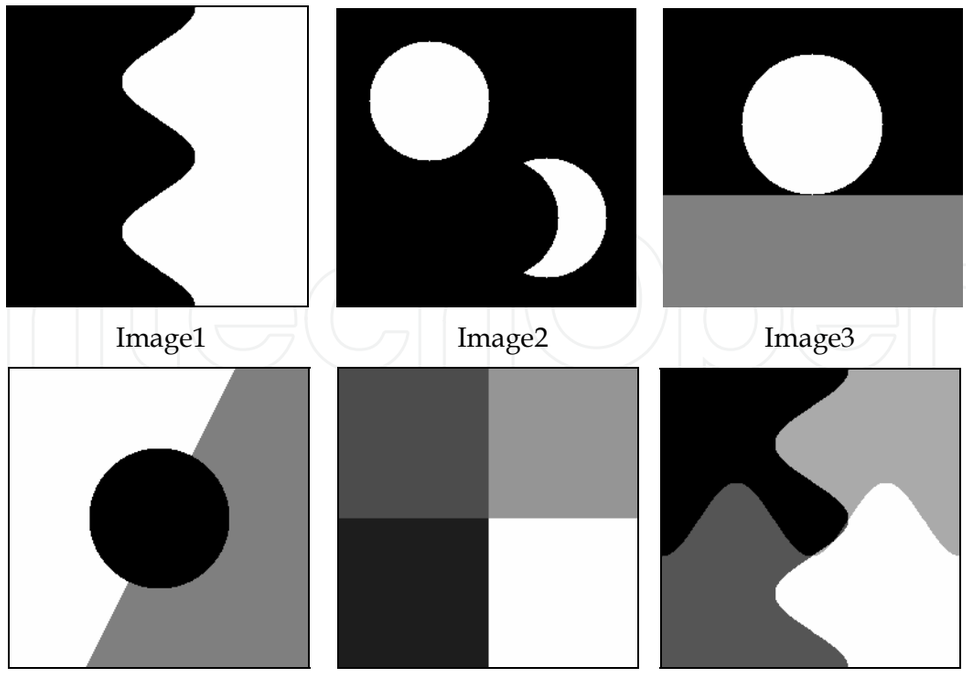


Fig. 8. The true partitioning of the artificial texture images

2.4.2 Results on artificial texture images

Image1 and Image2 contain two textures, Image3 and Image4 contain three textures and Image5 and Image6 contain four textures. Gray level, GLCM and wavelet features are used. In these experiments, the true partitioning is known. Thus we can calculate the clustering correct ratio directly by contrasting a segmentation result to the corresponding true

Problem	Correct Ratio (Standard Deviation)					
	MISA	GISA	EGSA	SCE	FCM	KM
Image1	**0.9844** (0.0069)	0.9837 (0.0021)	0.9705 (0)	0.9842 (0.0005)	0.9841 (0)	0.9835 (0)
Image2	**0.9770** (0.0012)	0.9746 (0.0015)	0.9341 (0)	0.8630 (0.0158)	0.9745 (0)	0.9719 (0)
Image3	0.9475 (0.0018)	0.9442 (0.0026)	**0.9517** (0)	0.8983 (0.0950)	0.7863 (0.2938)	0.7807 (0.2883)
Image4	0.9539 (0.0010)	0.9505 (0.0418)	0.9421 (0)	**0.9614** (0.0014)	0.9502 (0)	0.9505 (0.0001)
Image5	**0.9622** (0.0008)	0.9588 (0.0014)	0.8326 (0)	0.9114 (0.0783)	0.7400 (0.2718)	0.7711 (0.2608)
Image6	**0.9493** (0.0004)	0.9485 (0.0006)	0.8871 (0)	0.9319 (0.0408)	0.8243 (0.2737)	0.8243 (0.2642)

Table 2. Statistic results obtained from MISA, GISA, EGSA, SCEA, FCM and KM on the artificial texture images

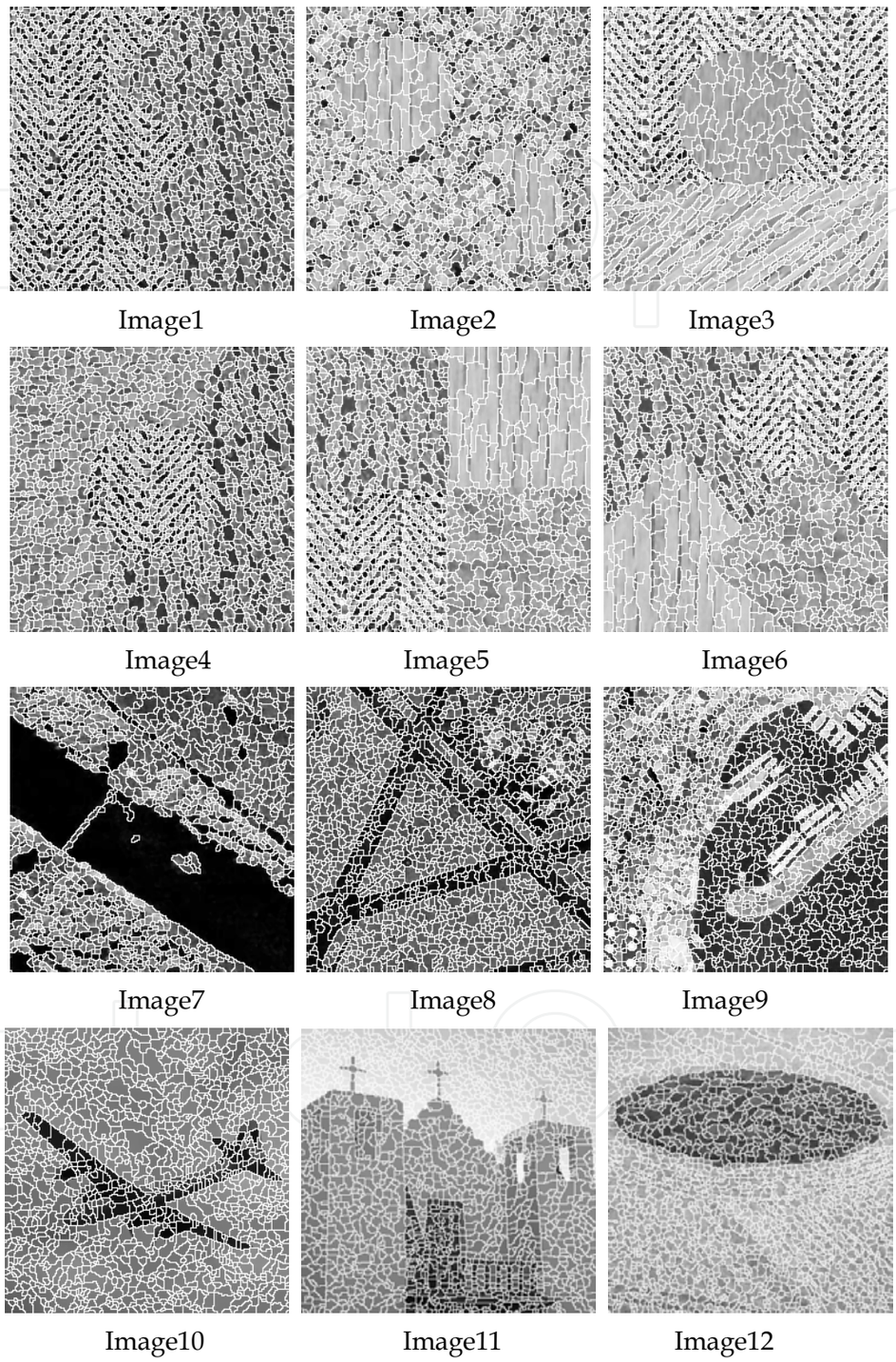
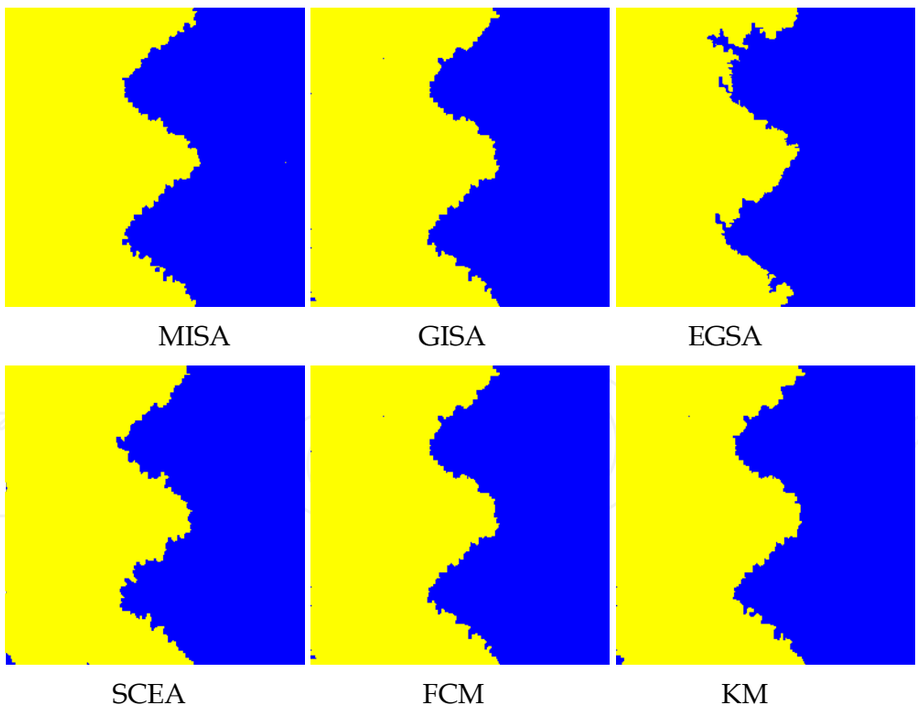
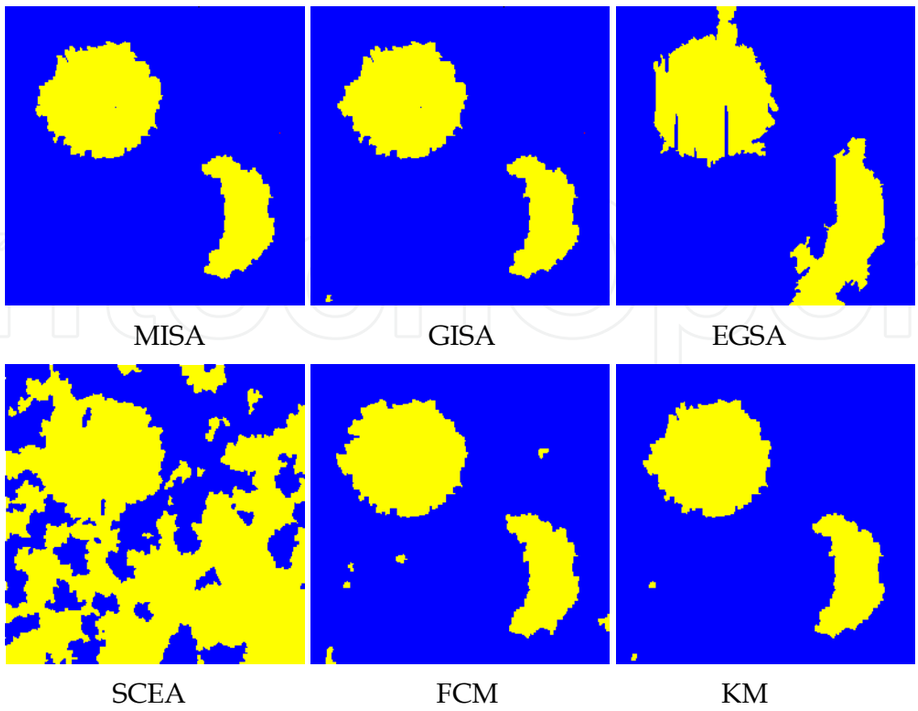


Fig. 9. The watershed segmentation results of Image 1 to Image 12.

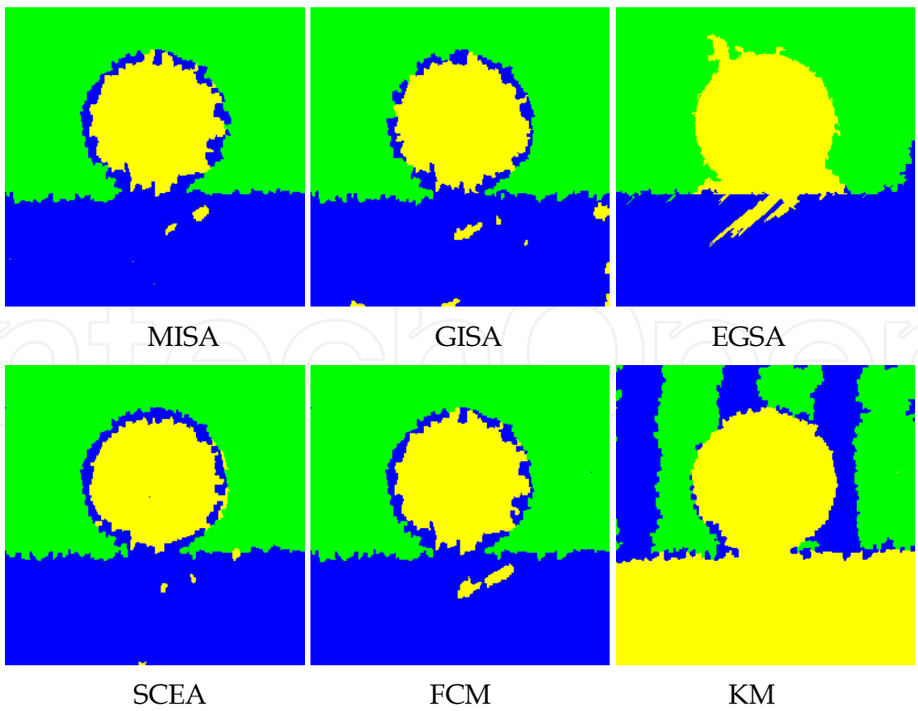
partitioning. The average results of clustering correct ratios and standard deviations based on 30 independent runs are shown in Table 2. Fig. 10 shows the typical results of them obtained from the six algorithms. For each test image, the six segmentation images arranged from upper left to lower right are typical results of MISA, GISA, EGSA, SCEA, FCM and KM, respectively. Table 2 shows that MISA obtains higher correct ratio than GISA, FCM and KM do for all these six images, and gets the best correct ratio among all the six algorithms on Image1, Image2, Image5 and Image6. EGSA and SCEA get better correct ratios than MISA does on Image3 and Image4, respectively. EGSA is not a random algorithm, so all the standard deviations of its results are 0. MISA exceeds EGSA greatly on Image5 and Image6 and surpasses SCEA apparently on Image2 and Image3 in correct ratios. MISA exceeds KM and FCM greatly on Image3, Image5 and Image6 in both the correct ratios and standard deviation. Fig. 10 illustrates the above numerical comparisons visually.



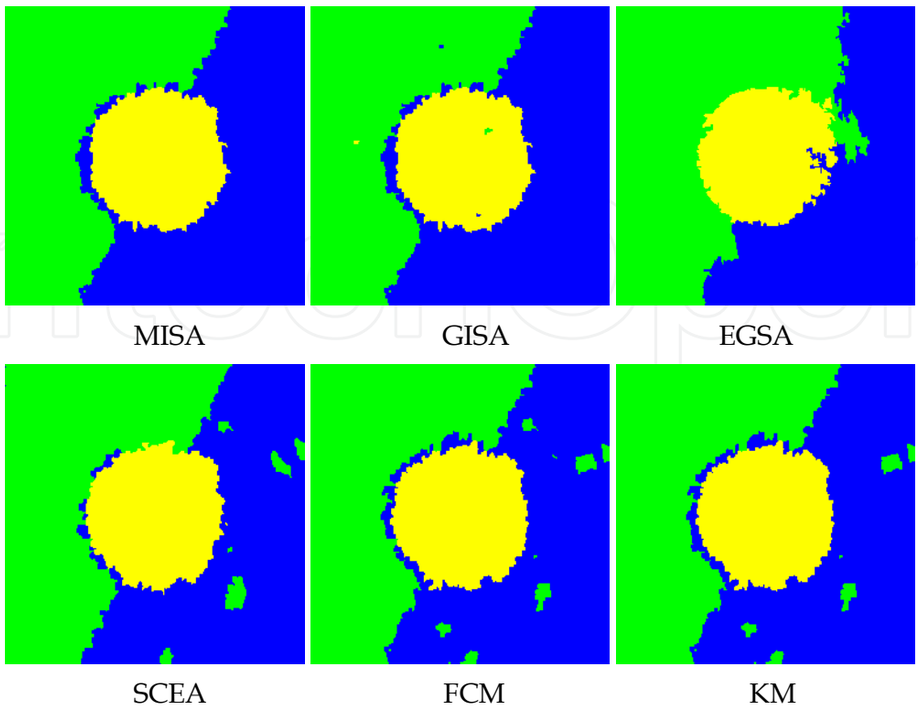
(a) Segmentation results of Image1



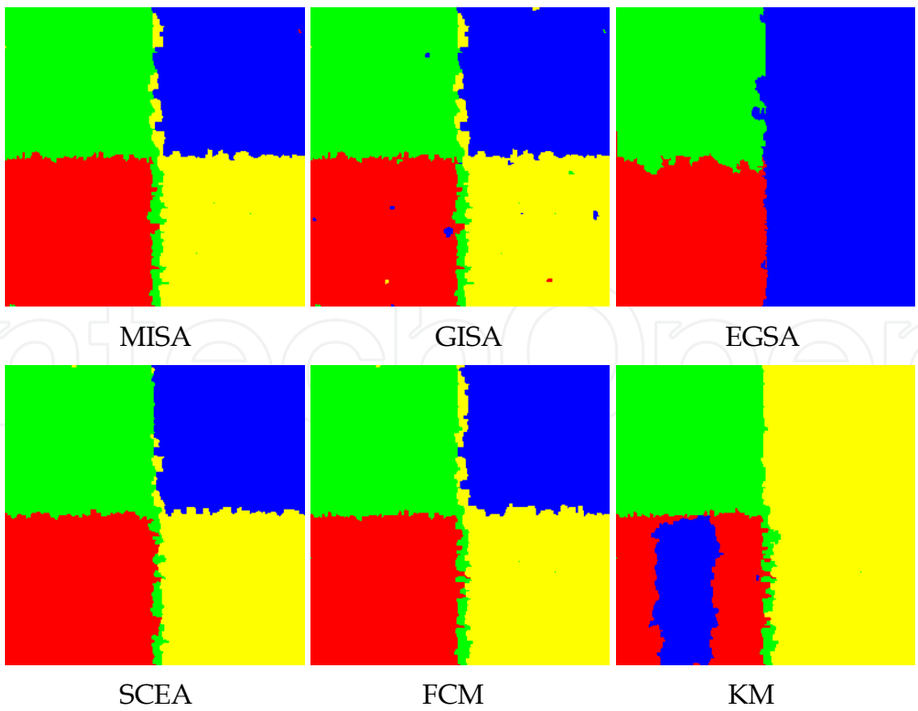
(b) Segmentation results of Image2



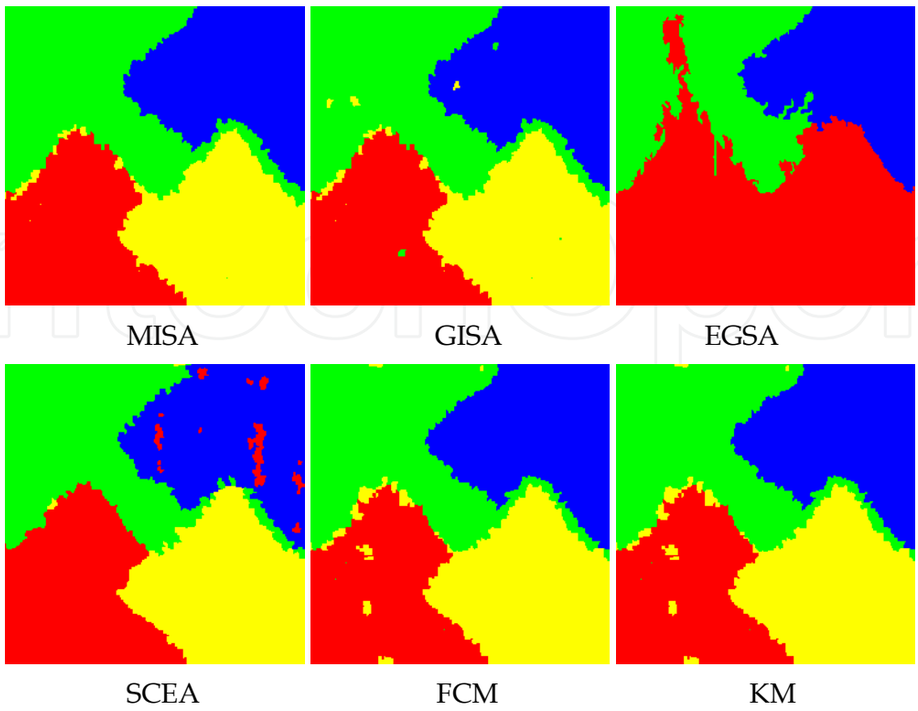
(c) Segmentation results of Image3



(d) Segmentation results of Image4



(e) Segmentation results of Image5



(f) Segmentation results of Image6

Fig. 10. Typical results obtained from the compared algorithms on the artificial texture images

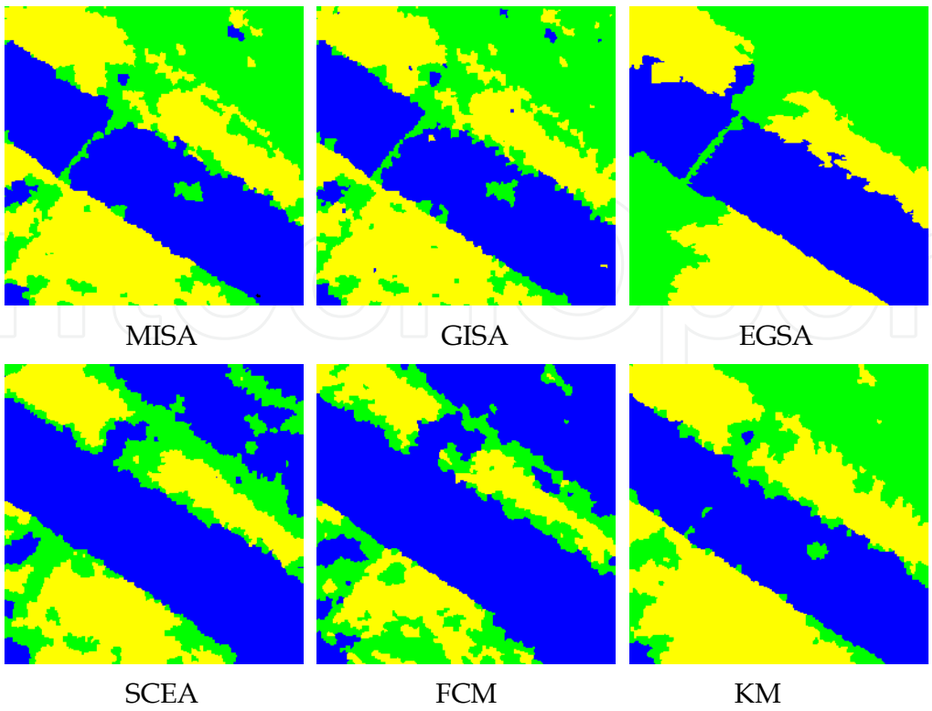
2.4.3 Results on remote sensing Images

Fig. 11 shows the typical results on remote sensing images. For each test image, the six segmentation images arranged from upper left to lower right are typical results of MISA, GISA, EGSA, SCEA, FCM and KM respectively. Gray level and wavelet features are used. For these remote sensing images, there is no true partitioning for reference, thus numerical results could not be obtained here.

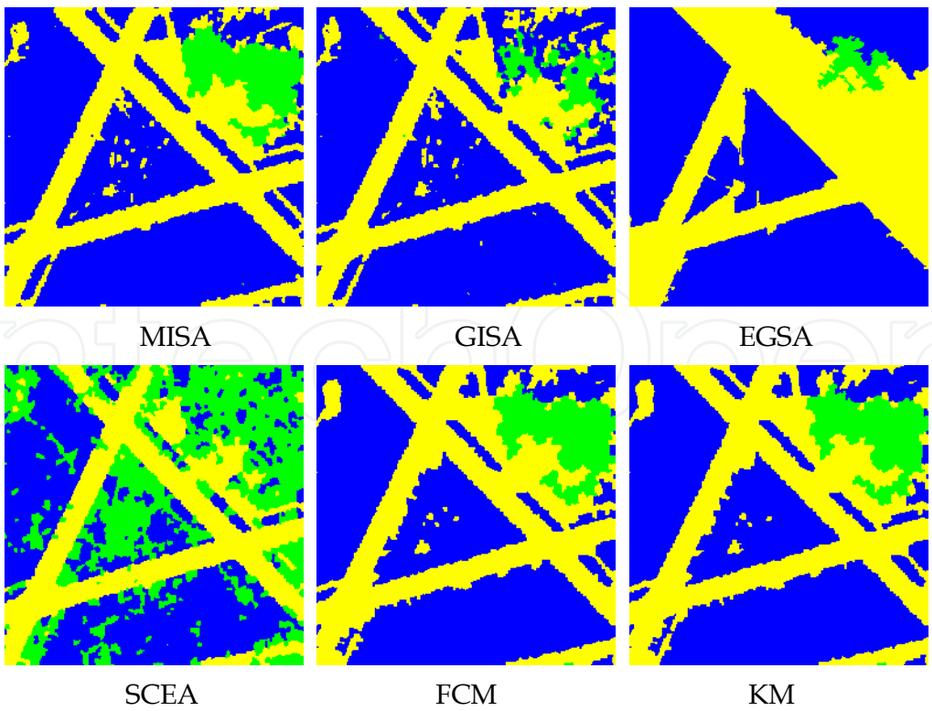
For Image7, MISA makes the river bank clearer, and generates more homogeneous lower crops part than GISA does. But both MISA and GISA lose the crop line on the upper left corner. EGSA lose the crop part in the lower left part and the island in the river, but generates a clear bridge. SCEA and FCM might confuse the vegetations with the river. KM loses most of the bridge and other detailed information.

For Image8, MISA gets very clear edges and integrates the building area, leaving some speckles in the center which are the trails of thin roads there. In the GISA results, some buildings are lost in the background. GISA produces some speckles because 200 generations are not enough for it to converge. EGSA and SCEA produce unacceptable results which mix up major cluster information. Both FCM and KM generate coarse edges of the main roads and lose detailed information of other narrow roads in the center and upper right.

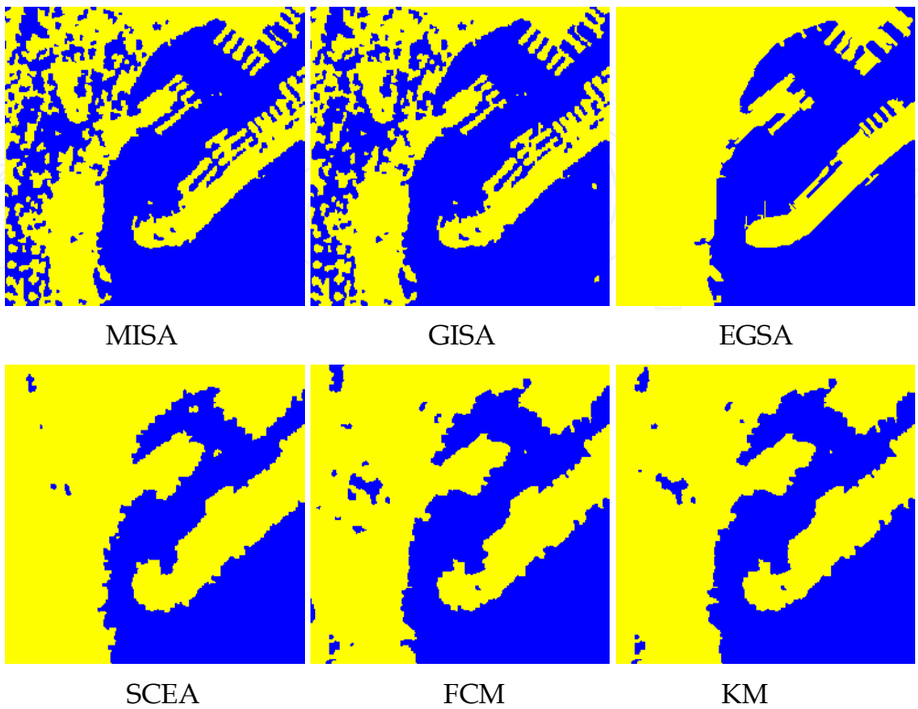
For Image9, MISA and GISA can distinguish details on the port area but generate some speckles on the land area. EGSA doesn't confuse the land area on the left with water, but misses detailed information on the port area. SCEA, FCM and KM have worse results when compared with MISA on the port area and compared with EGSA on the land area.



(a) Segmentation results of Image7



(b) Segmentation results of Image8



(c) Segmentation results of Image9

Fig. 11. Typical results obtained from the compared algorithms on the remote sensing images

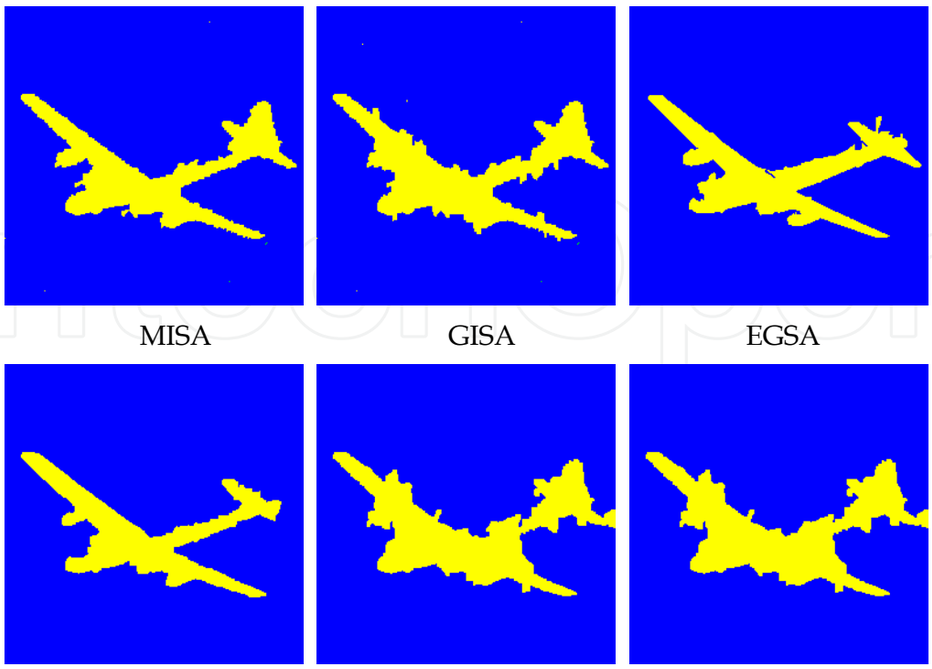
2.4.4 Results on natural images

Fig. 12 shows the typical results on natural images. For each image, the six segmentation images arranged from upper left to lower right are typical results of MISA, GISA, EGSA, SCEA, FCM and KM respectively. Gray level and wavelet features are used for experiments here.

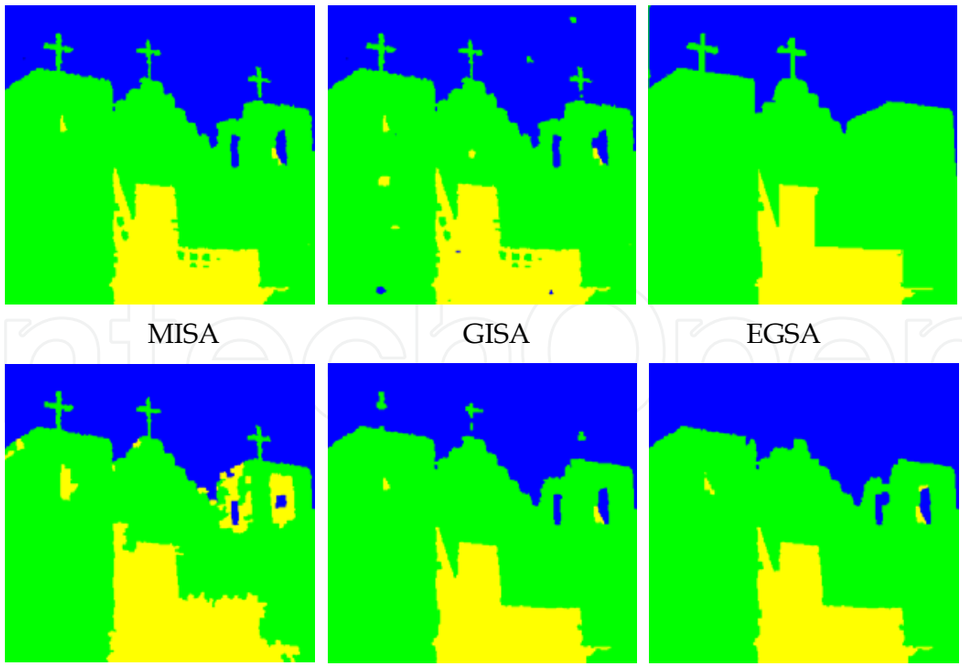
Image10 is segmented into three clusters; Image11 and Image12 are segmented into two clusters. Apparently, MISA produces the best results. GISA also gets better results than FCM and KM. EGSA, SCEA, FCM and KM lose a lot of details and even confuse very large parts in segmenting Image12.

2.4.5 Sensitivity in relation to parameters

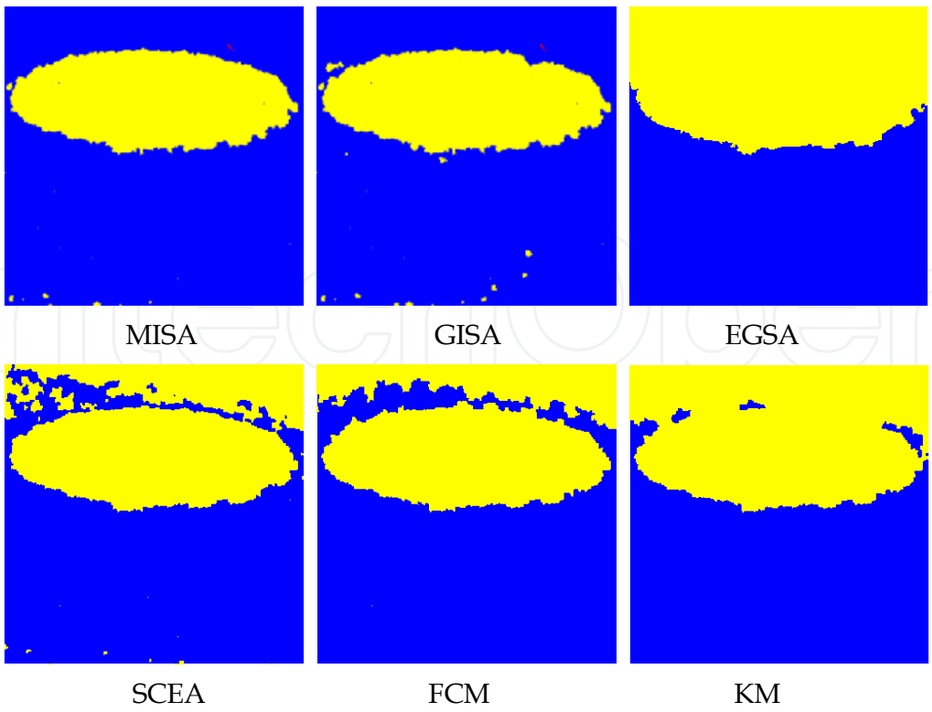
Fig.13 shows the convergence curves of MISA and GISA got from experiments on the artificial texture Image4, the remote sensing Image9, and the natural Image12. Each experiment is repeated 30 times for statistical evaluation, and the points in those figures are the average values. We can see that MISA converges within no more than 50 generations on all the experiments, while GISA can not converge within 100 generations.



(a) Segmentation results of Image10



(b) Segmentation results of Image11



(c) Segmentation results of Image12

Fig. 12. Typical results obtained from the compared algorithms on the natural images

Fig. 14 shows the curves which show the influence of learning intensity. We make use of correct ratios of the six artificial texture images to exhibit the effect of learning intensity. The experiments are repeated 30 times, and we plot their statistical average values. Based on Fig. 14, we can see that the correct ratio increases as the learning intensity is reinforced before at about 0.2. After that，the curves fluctuate but do not increase apparently. This demonstrates that learning intensity should be adjusted properly as a small value is not enough for good convergency while a big one will lead to wasting of computation.

3. Image texture classification using a manifold distance based evolutionary clustering cethod

3.1 Introduction

Image classification or segmentation based on texture features using unsupervised approaches has been a challenge 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 categorized into four major categories [53, 54]: statistical, geometrical, model-based, and signal processing. Among them, gray-level co-occurrence features, first proposed by Haralick, Shanmugam and Dinstein [55], are one of the most common features used in 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

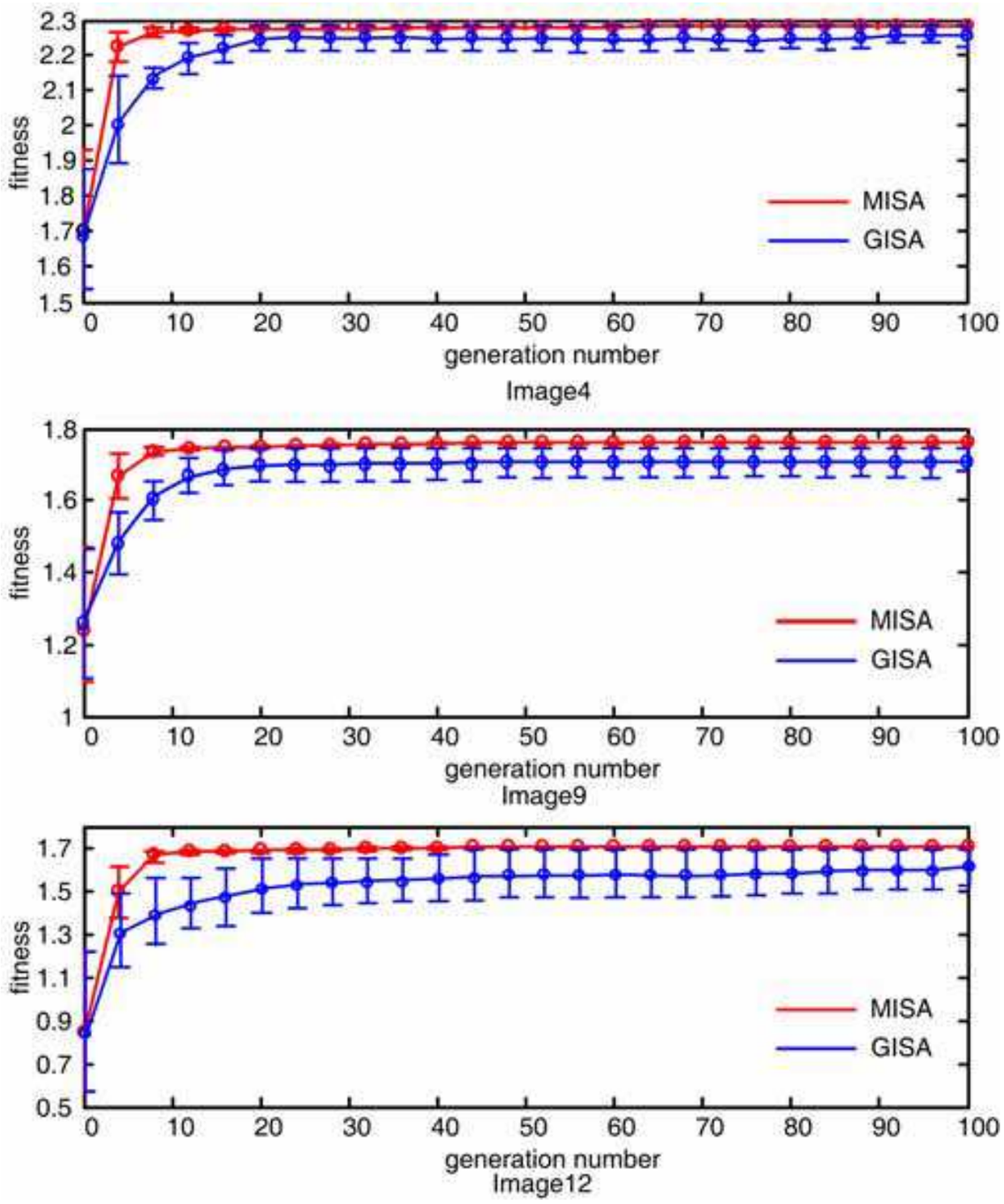


Fig. 13. The convergence curves of MISA and GISA

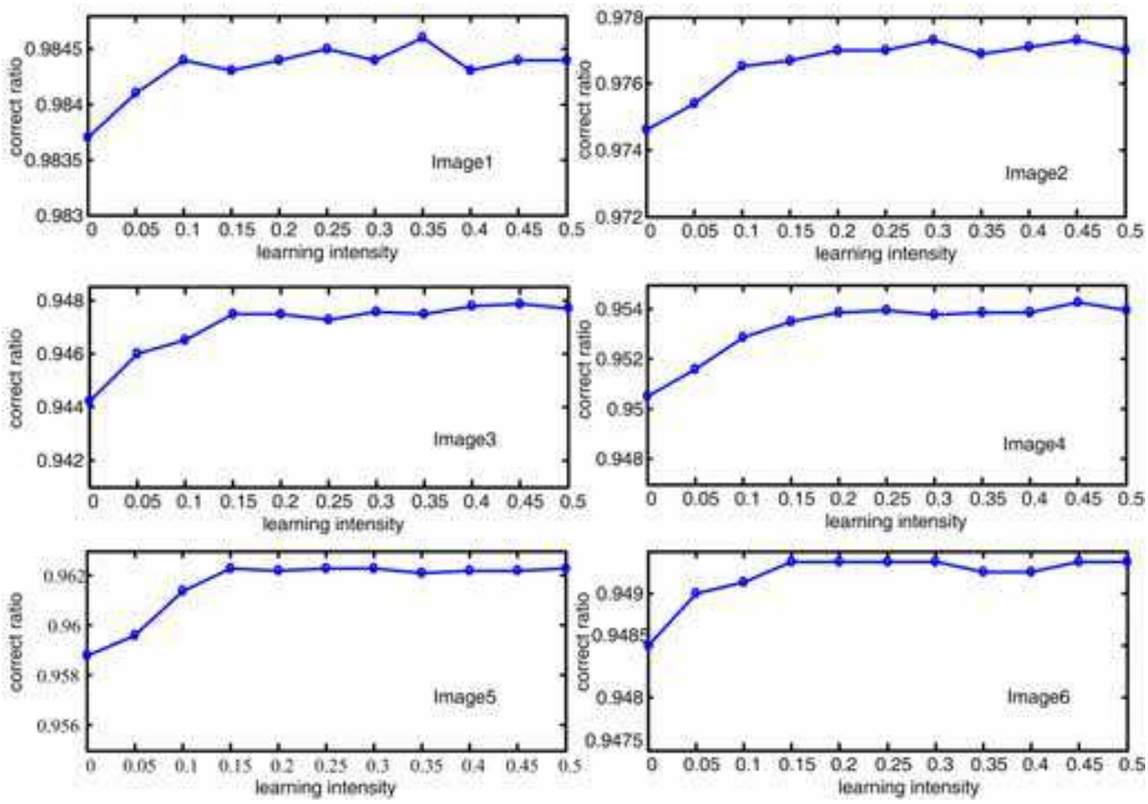


Fig. 14. The influence of learning intensity versus correct ratio

used to assign class labels to the feature vectors. These approaches can be categorized into two groups [56, 57]: hierarchical clustering and partitional clustering, where partitional clustering approaches, such as K-Means Algorithm [58], 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 Algorithm (GA), Evolutionary Strategy (ES), Evolutionary Programming (EP), etc., have been used for clustering tasks commonly in literature [59~62]. A variety of EA representations for clustering solutions have been explored, such as the straightforward encoding with each gene coding for the cluster membership of the corresponding data item, and the locus-based adjacency representation [62]. Many researchers [59~61] have chosen to use a more indirect approach that borrows from the K-Means algorithm: the representation codes for cluster center only, and each data item is subsequently assigned to a cluster representative according to an appointed dissimilarity measure. The most popular dissimilarity measure is the Euclidean distance. By using Euclidean distance as a measure of dissimilarity, these evolutionary clustering methods as well as the K-Means algorithm have a good performance on the data set with compact super-sphere distributions, but tends to fail in the data set 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 [63] 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 present a symmetrical structure with respect to the cluster center. Therefore, it is suitable to clustering data sets with clear symmetrical structure. Charalampidis [64] 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 solve 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 data item is assigned to a cluster representative according to a novel dissimilarity measure which can 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 will be validated by comparing with the K-Means algorithm, a modified K-Means algorithm using the manifold distance-based dissimilarity measure [65], and the genetic algorithm-based clustering technique proposed by Maulik and Bandyopadhyay [60] in solving seven benchmark clustering problems of artificial data sets, three artificial texture image classification problems and two Synthetic Aperture Radar (SAR) images classification problems.

The remainder of this section is organized as follows: Section 2.2 describes the novel manifold distance-based dissimilarity measure. Section 2.3 describes the evolutionary clustering algorithm based on the novel dissimilarity measure. In Section 2.4, we summary and evaluate the experimental results.

3.2 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 location or global characteristics. Through a large mount of observation, we have found the following two consistency characteristics of data clustering.

a. Local consistency refers that data points close in location will have a high affinity.
b. Global consistency refers that data points locating 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 results in the classical Euclidian distance metric can only reflect the local consistency, but fail to describe the global consistency. We can illustrate this problem by the following example. As shown in Fig. 15, 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 Euclidian distance metric, however, point *a* is much closer to point *f* than to *e*, thus without reflecting the global consistency. Hence for complicated real-world problems, simply using Euclidean distance metric as a dissimilarity measure can not fully reflect the characteristics of data clustering.

Here, we want to design a novel dissimilarity measure with the ability of reflecting both the local and global consistency. As an example, we can observe from the data distribution in Fig. 15 that data points in the same cluster tend to lie in the same manifold.

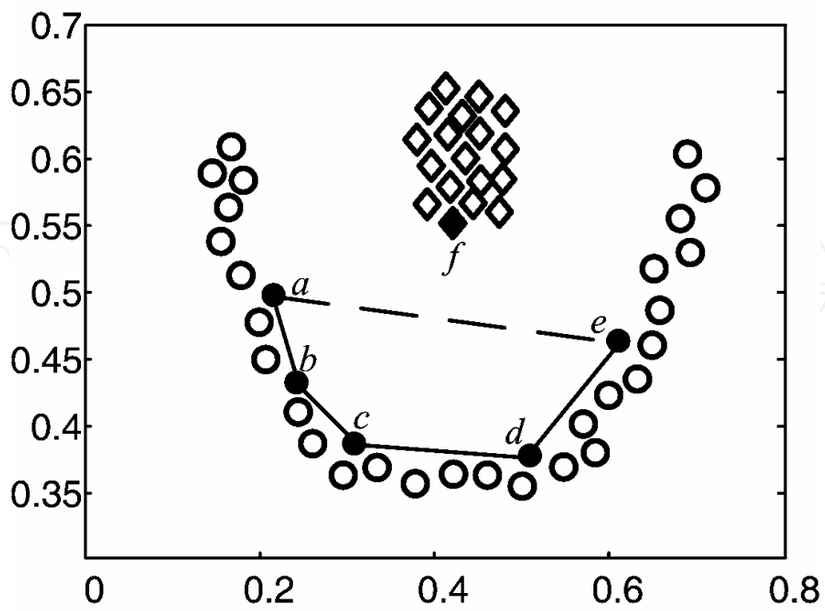


Fig. 15. An illustration of that the Euclidian distance metric can not reflect the global consistency.

At first, 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 a 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 semi-supervised learning [66, 67]. The aim of using this kind of measure is to elongate the paths that cross different manifolds, and simultaneously shorten those that not cross.

To formalize this intuitive notion of dissimilarity, we need first define a so-called manifold length of line segment. We have found a property that a distance measure describing the global consistency of clustering does not always satisfy the triangle inequality under the Euclidean metric. As shown in Fig. 15, to describe the global consistency, it is required that the length of the path connected by shorter edges is smaller than that of the direct connected path, i.e. $\overline{ab}+\overline{bc}+\overline{cd}+\overline{de}<\overline{ae}$. In other words, a direct connected 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 line segment (x_i,x_j) is defined as

$$L(x_i,x_j)\triangleq\rho^{dist(x_i,x_j)}-1 \tag{7}$$

where $dist\left(x_i,x_j\right)$ is the Euclidean distance between x_i and x_j; $\rho>1$ is the flexing factor. Obviously, the manifold length of line segment possesses the property mentioned above, thus can be utilized to describe the 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 segment, we define a new distance metric, called 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 $\mathbf{P}_{i,j}$ 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

$$D(x_i,x_j) \triangleq \min_{p \in \mathbf{P}_{i,j}} \sum_{k=1}^{|p|-1} L(p_k,p_{k+1}) . \tag{8}$$

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 distance among data points in different manifolds and simultaneously shortening the distance among data points in the same manifold.

3.3 Evolutionary clustering based on the manifold distance

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

3.3.1 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 number of K cluster representatives. The length of a chromosome is K words, where the first gene represents the first cluster, the second gene 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 is 100 and the number of clusters being considered is 5. Then the individual (6, 19, 91, 38, 64) represents that the 6-th, 19-th, 91-st, 38-th, and 64-th 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 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 [68] because it is unbiased with respect to the ordering of genes and can generate any combination of alleles from the two parents [62, 69]. An example of the operation of uniform crossover on the encoding employed is shown in example 2.

Example 2. Let the two parent individuals are (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 is repeat, we keep it unchanged.
Each individual undergoes mutation with probability p_m as example 3.
Example 3. Let the size of the data set is 100 and the number of clusters being considered is 5. Then the individual (6, 19, 91, 38, 64) can mutate to (6, $19+\lfloor(100-19)\times random+1\rfloor$, 91, 38, 64) or (6, $19-\lfloor(19-1)\times random+1\rfloor$, 91, 38, 64) equiprobably when the second gene is chosen to mutate, *random* denotes a uniformly distributed random number in the range [0,1).

3.3.2 Objective function

Each data item 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 6-th, 19-th, 91-st, 38-th, and 64-th points represent the five clusters, respectively. For the first point, we compute the manifold distance between it and the 6-th, 19-th, 91-st, 38-th, and 64-th points, respectively. If the manifold distance between the first point and the 38-th point is the minimum one, then the first point is assigned to the cluster represented by the 38-th point. All the points are assigned in this way.
Subsequently, the objective function is computed as follows:

$$Dev(C)=\sum_{C_k\in C}\sum_{i\in C_k} D(i,\mu_k) \tag{9}$$

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 *i*-th data item of cluster C_k and μ_k.

3.3.3 Manifold evolutionary clustering algorithm

In MEC, the processes of fitness computation, roulette wheel selection with elitism [70], crossover and mutation are executed for a maximum number of generations Gmax. The best individual in the last generation provides the solution to the clustering problem. The main loop of MEC is as follows.

Algorithm 2.1. 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 number 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.

3.4 Experimental study

3.4.1 Experimental setup

In order to validate the performance of MEC, we first apply MEC to seven benchmark clustering problems of artificial data sets. The results will be compared with the K-Means algorithm (KM) [58], a modified K-Means algorithm using the manifold distance-based dissimilarity measure (DSKM) [65], and the genetic algorithm-based clustering technique (GAC) proposed by Maulik and Bandyopadhyay [60]. In all the algorithms, the desired number of clusters is set to be known in advance.

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

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

In the image classification experiments (the second and third experiments), we will use the gray-level co-occurrence matrix (GLCM) [55] 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 [54], 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 co-occurrence data, namely, interpixel orientation, interpixel distance, the number of gray levels and window size. Typically, interpixel orientation is set to 0^o, 45^o, 90^o, 135^o since this is easiest to implement. Short interpixel distances have typically achieved the best success, so interpixel distance is 1 will be used. This combination of offset and orientation has characterized SAR texture well [54]. The role of varying the values of the number of gray levels and windows size with respect to GLCM statistics has been presented in many references [54, 71]. After their analysis and fine-tune experiments, in this study, we set the number of gray levels is 16 and the window size is 13×13.

The parameter settings used for MEC and GAC in our experimental study are given in Table 3. For DSKM and KM, the maximum iterative number is set to 500, and the stop threshold is 10^{-10}.

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

Table 3. Parameter settings for MEC and GAC

In the first two experiments, the true partitioning is known, we will evaluate the performance using two external measures, the Adjusted Rand Index [62, 72, 73] and the Clustering Error [65].

The Adjusted Rand Index [72] is a generalization of the Rand Index [74] which takes two partitioning as the input and count the number of pair-wise co-assignments of data items between the two partitioning. Given a set of N points $S=\{p_1,p_2,\cdots,p_N\}$, suppose $U=\{u_1,u_2,\cdots,u_K\}$ and $V=\{v_1,v_2,\cdots,v_K\}$ represent two different partitions of the points in S such that $\bigcup_{i=1}^{K}u_i=\bigcup_{j=1}^{K}v_j=S$ and $u_i\cap u_{i'}=v_j\cap v_{j'}=\varnothing$ for $1\le i\ne i'\le K, 1\le j\ne j'\le K$.

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 $\frac{a+d}{a+b+c+d}$. 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 the expected value of the Rand index of two random partitions does not take a constant value (say zero). The Adjusted Rand Index proposed by Hubert and Arabie [72] assumes the generalized hypergeometric distribution as the model of randomness, i.e. the U and V partitions are picked at random such 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 $n_{i\bullet}$ and $n_{\bullet j}$ be the number of points in class u_i and class v_j respectively. Under the generalized hypergeometric model, it can be shown that:

$$E\left[\sum_{i,j}\binom{n_{ij}}{2}\right]=\left[\sum_{i}\binom{n_{i\bullet}}{2}\bullet\sum_{j}\binom{n_{\bullet j}}{2}\right]/\binom{n}{2} \tag{10}$$

Then the Adjusted Rand Index is given as

$$R(U,V)=\frac{\sum_{i,j}\binom{n_{ij}}{2}-\left[\sum_{i}\binom{n_{i\bullet}}{2}\bullet\sum_{j}\binom{n_{\bullet j}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_{i}\binom{n_{i\bullet}}{2}+\sum_{j}\binom{n_{\bullet j}}{2}\right]-\left[\sum_{i}\binom{n_{i\bullet}}{2}\bullet\sum_{j}\binom{n_{\bullet j}}{2}\right]/\binom{n}{2}} \tag{11}$$

The Adjusted Rand Index return values in the interval [0, 1] and is to be maximized.

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

$$CE(U,V)=\frac{1}{N}\sum_{i=1}^{K}\sum_{\substack{j=1\\ i\ne j}}^{K}Confusion(i,j), \tag{12}$$

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

3.4.2 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 data sets can be seen in Fig. 16. We perform 30 independent runs on each problem. The average results of the two metrics, Clustering Error and Adjusted Rand Index, are shown in Table 4.

Problem	Clustering Error				Adjusted Rand Index			
	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

Table 4. Results of MEC, GAC, DSKM and KM on artificial data sets where the results in bold are the best ones

From Table 4, 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 desired clustering for the two spheroid data sets, i.e. Size5 and Square4. This is due to that 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 clusters, which indicate the manifold distance metric are very suitable to measure complicated clustering structure. When comparisons are made between MEC and DSKM, MEC obtained the true clustering on the Long1, Spiral, Sticks, Line-blobs and Three-circles in all the 30 runs, but DSKM can not do it on the Line-blobs and Three-circles. Further more, 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 as the K-Means algorithm after cluster assignment which reducing the ability of reflecting the global consistency. MEC made up this drawback by evolutionary searching the cluster representatives from a combinatorial optimization viewpoint. In order to show the performance visually, the typical simulation results on the eight data sets obtained from MEC are shown in Fig. 16.

3.4.3 Implementation results on artificial texture image classification

Image1 is a simple 256×256 bipartite image (Fig. 17(a)). The original image contains two textures, cork and cotton, selected from the Brodatz texture images [75]. Fig. 17(b) represents the true partitioning of Image1. Image2 also contains two textures as shown in Fig. 17(c), and Fig. 17(d) represents its true partitioning. Image3 is a more complicated texture synthesized image with 4 classes, and Fig. 17(e) and (f) represent the original image and the true partitioning, respectively.

We perform 30 independent runs on each problem. The average results of the two metrics, Clustering Error and Adjusted Rand Index, are shown in Table 5. Fig. 18 to Fig. 20 are the typical implementation results obtained from the four algorithms, MEC, GAC, DSKM and KM, in clustering the three texture images, respectively.

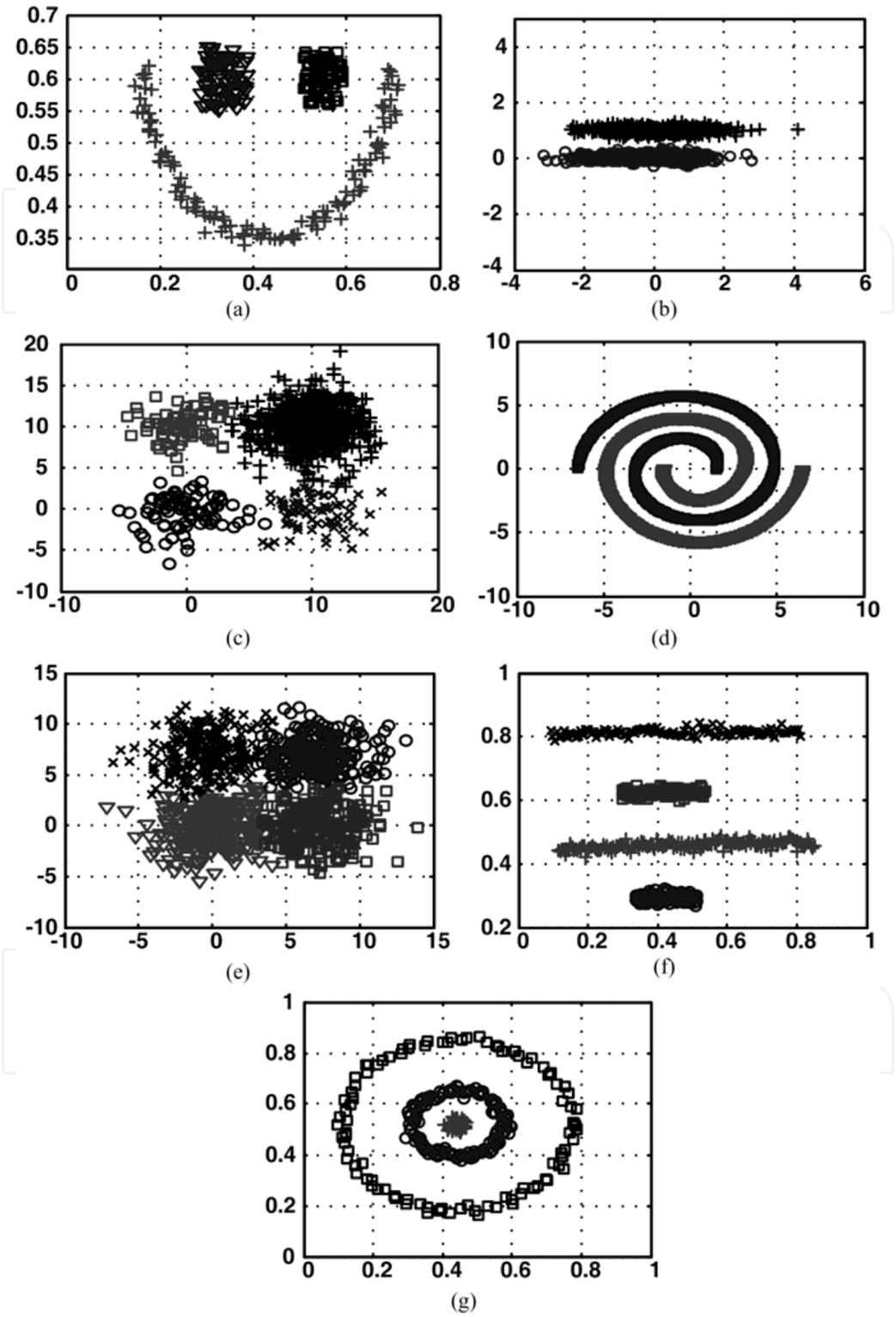


Fig. 16. The typical implementation results on the artificial data sets obtained from MEC. (a) Line- blobs; (b) Long1; (c) Size5; (d) Spiral; (e) Square4; (f) Sticks; (G) Three-circles.

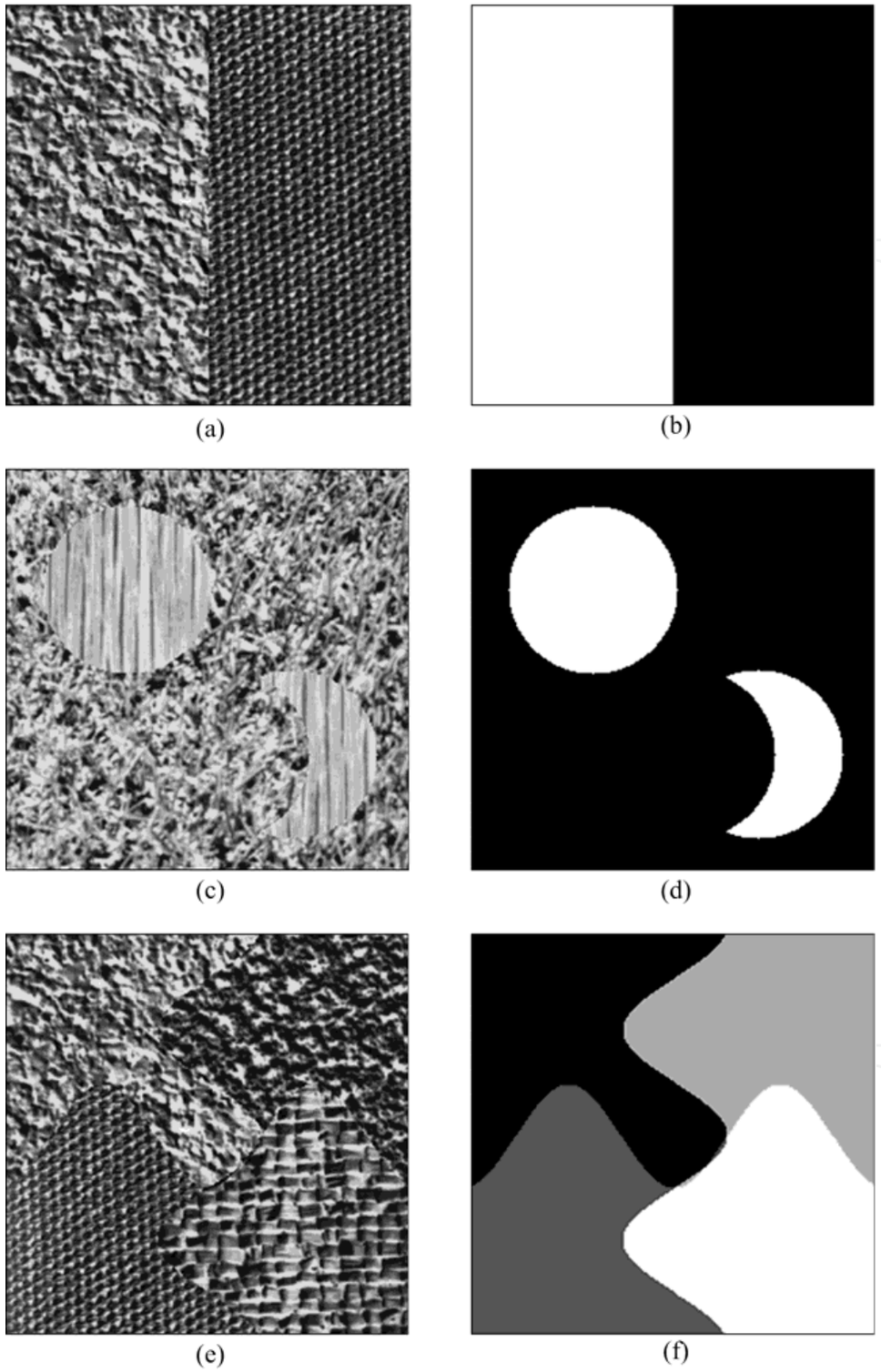


Fig. 17. Artificial texture images and their true partitioning. (a) Original Image1; (b) True partitioning of Image1; (c) Original Image2; (d) True partitioning of Image2; (e) Original Image3; (f) True partitioning of Image3.

Problem	Clustering Error				Adjusted Rand Index			
	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

Table 5. Results of MEC, GAC, DSKM and KM on artificial texture image classification where the results in bold are the best ones

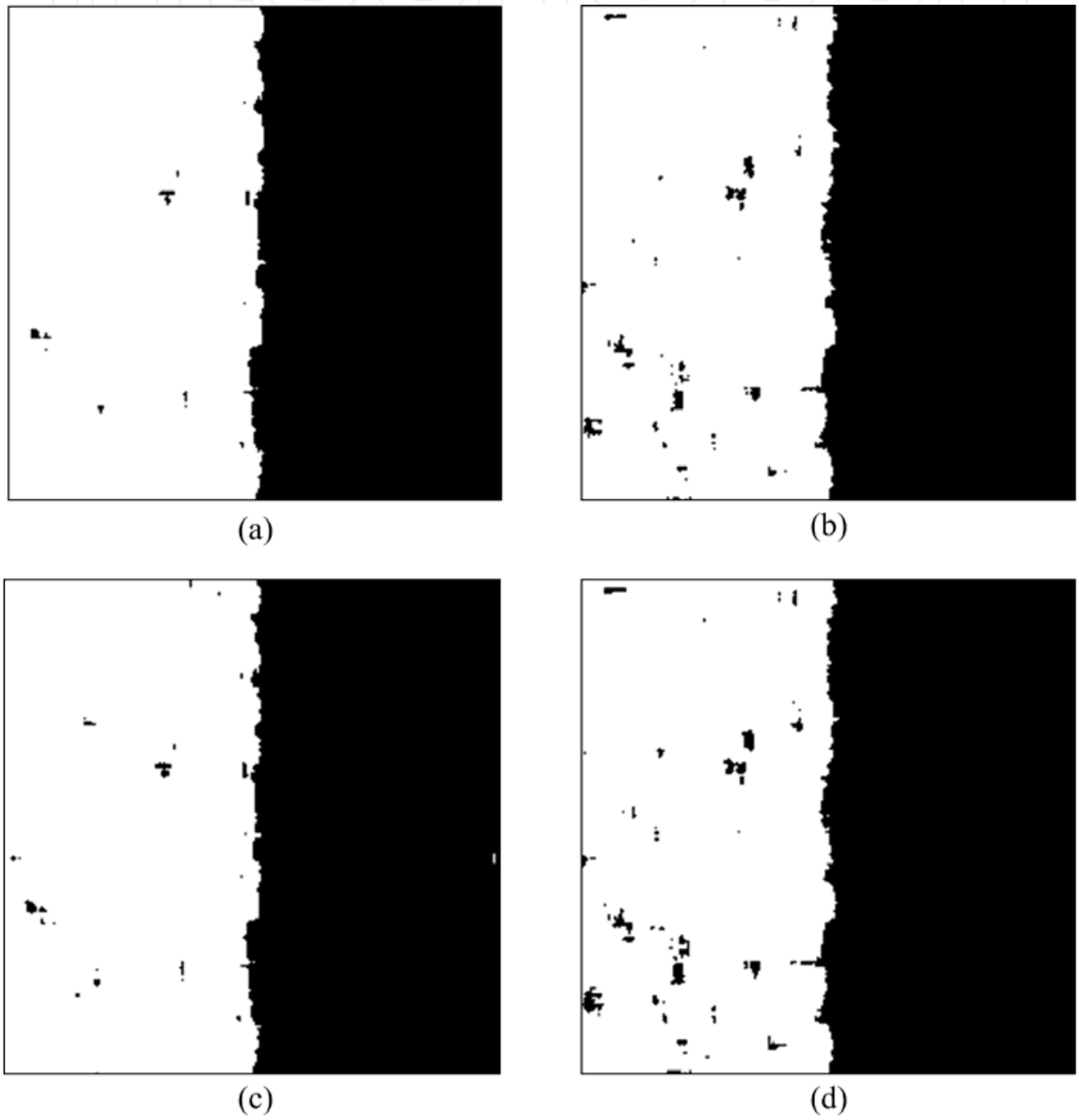


Fig. 18. The typical implementation results obtained from (a) MEC, (b) GAC, (c) DSKM and (d) KM in clustering Image1.

As shown in Table 5, all the average values of 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 the Image1. The values of Cluster Error and Adjusted Rand Index and Fig. 18 also show that the results obtained from MEC and DSKM are much better than the results of GAC and KM because both MEC and DSKM assign data items according to the manifold distance while GAC and KM assign data items according to Euclidian distance. However, the computational cost of the manifold distance is much larger than that of Euclidian distance. MEC and DSKM have similar results in clustering Image1.

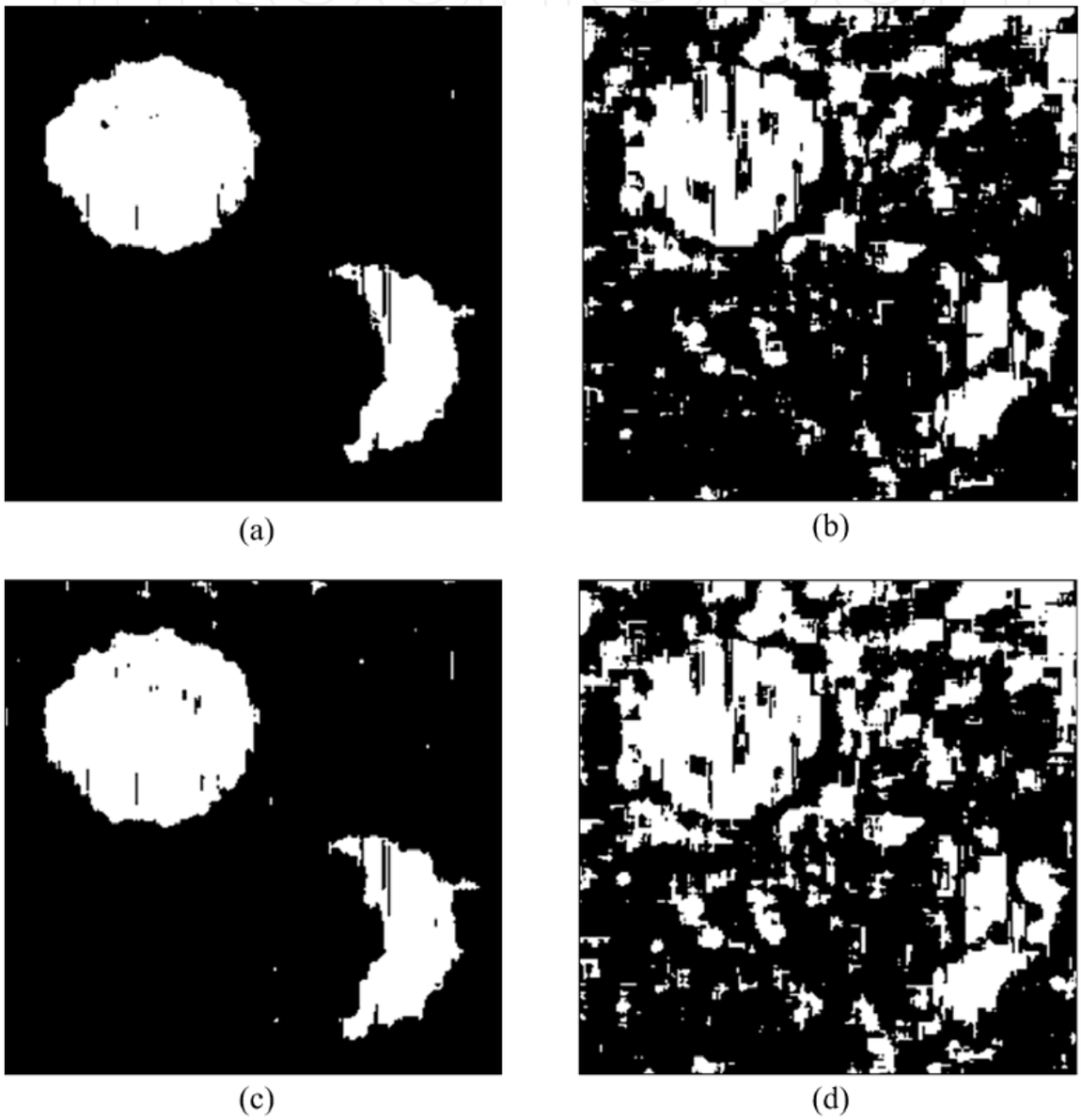


Fig. 19. The typical implementation results obtained from (a) MEC, (b) GAC, (c) DSKM and (d) KM in clustering Image2.

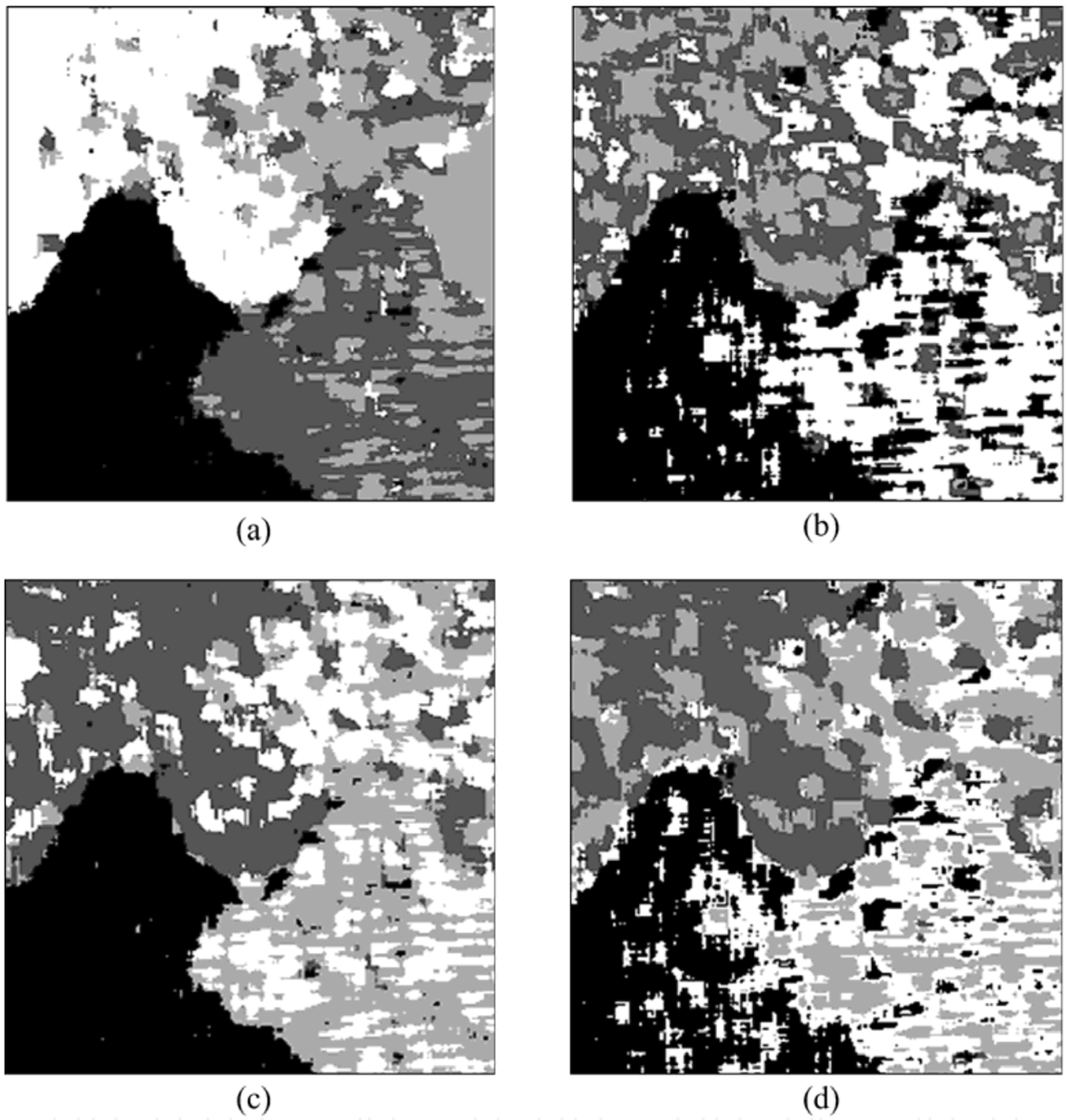


Fig. 20. The typical implementation results obtained from (a) MEC, (b) GAC, (c) DSKM and (d) KM in clustering Image3.

In clustering Image2, the average value of Cluster Error obtained from MEC is much smaller than the results obtained from GAC, DSKM and KM, and the average value of Adjusted Rand Index of MEC is obviously greater than the results obtained from GAC, DSKM and KM. So MEC does best in this problem. Fig. 19 also shows that the MEC result and DSKM result are obviously better than the GAC result and KM result, and the MEC result is better than the DSKM result. MEC segment the two textures better than DSKM may be due to MEC search 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 the ability of reflecting the global consistency.

In clustering the more complicated texture image Image3, all the average values of Cluster Error are greater than 12%, so none of the four algorithms can segment the image very well based on GLCM features. However, Table 5 and Fig. 20 show that MEC does much better than the other three algorithms.

3.4.4 Implementation results on remote sensing image classification

The first image, as shown in Fig. 21(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. 21(b), is a 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. Fig. 22 and Fig. 23 shows the clustering results obtained from the MEC, DSKM, GAC and KM in clustering these two SAR image respectively.

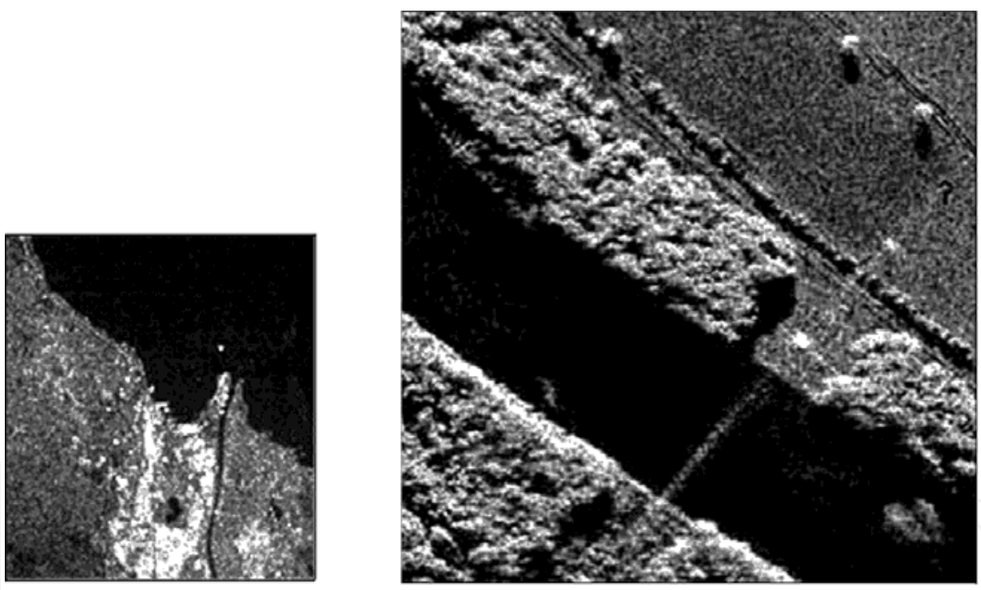


Fig. 21. Original SAR images. (a) X-band SAR image; (b) Ku-band SAR image.

Fig. 22 shows that all methods are readily able to perform the classification of the X-band SAR image. Fig. 22(b) and (d) show that many mountainous regions in the bottom left are recognized as lake by KM and GAC. Fig.22 (a) and (c) show that MEC can recognize these regions and DSKM can obviously reduce these error recognitions. Meanwhile, KM confuses many mountainous regions in the top left with city seriously. MEC reduce these errors mostly. Generally speaking, the MEC method outputs relatively better partitioning.

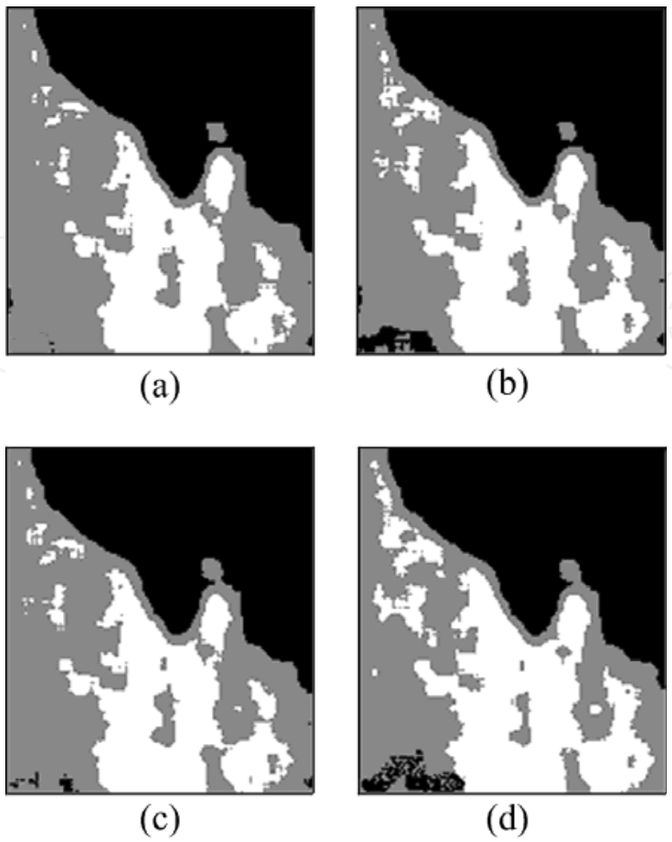


Fig. 22. Implementation results obtained from (a) MEC, (b) GAC, (c) DSKM and (d) KM in clustering the X-band SAR image.

Fig. 23 shows that the MEC, GAC, DSKM and KM generate different results, and all the methods do not perform as well as 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 river. Relatively, MEC and DSKM generate better partitioning of the river region. In distinguishing the vegetation and crop, the partitioning of GAC and KM appear more discontinuous than the results of MEC and DSKM method. GAC and KM tend to confuse the vegetation with crop along the river, delineating the crop more than it should. However, MEC and DSKM tend to identify the vegetation in the bottom left as the river, due to the nature of the gray-level of the leads in that region. DSKM also tends to confuse the vegetation with crop in the region along the river and 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.

3.4.5 Robustness and computing time

In order to compare the robustness of these methods, we follow the criteria used by [76]. In detail, the relative performance of the algorithm m on a particular data set is represented by the ratio b_m of its mean value of Adjusted Rand Index R_m and the highest mean value of Adjusted Rand Index among all the compared methods:

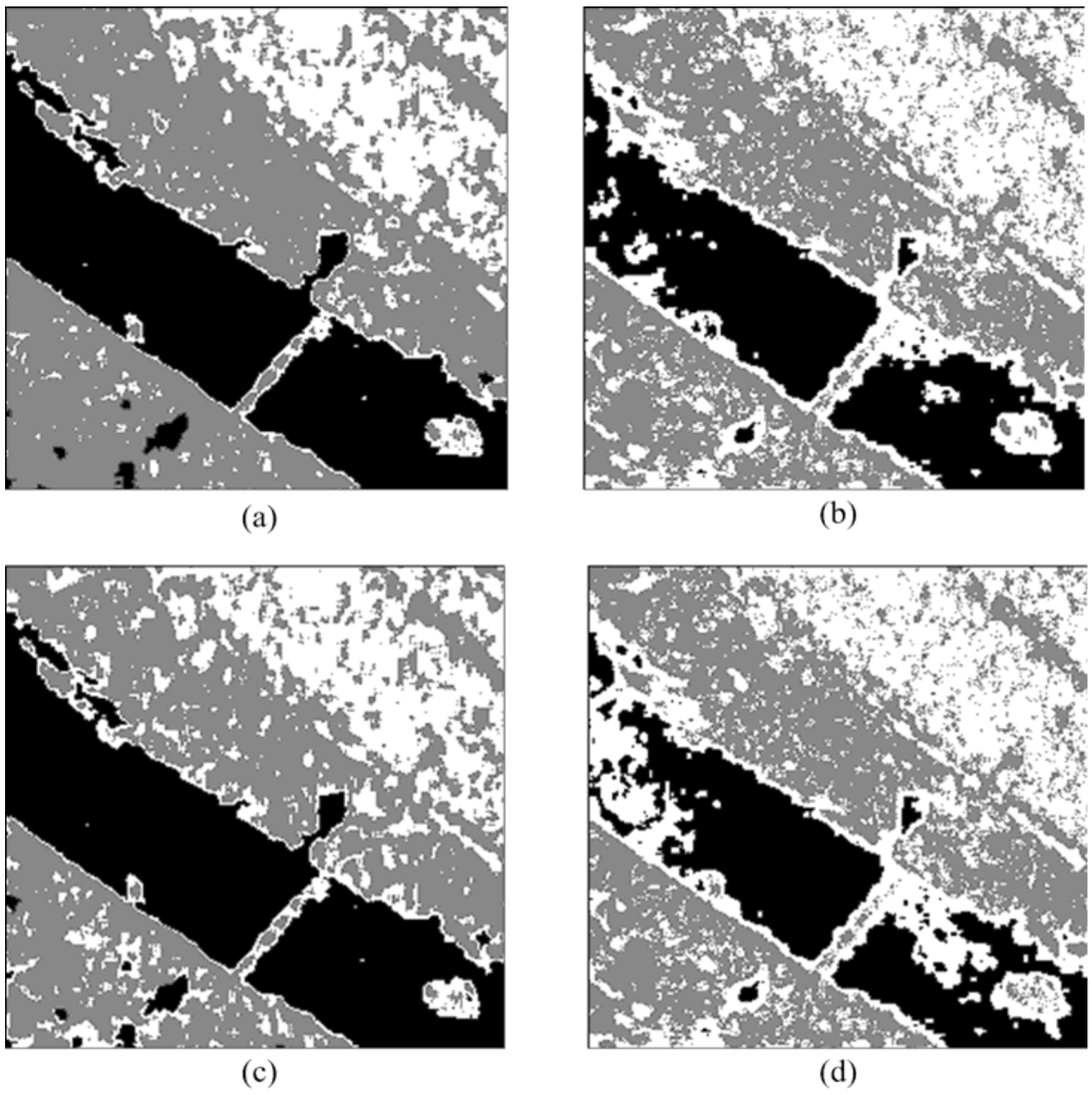


Fig. 23. Implementation results obtained from (a) MEC, (b) GAC, (c) DSKM and (d) KM in clustering the Ku-band SAR image.

$$b_m = \frac{R_m}{\max_k R_k} \tag{13}$$

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 measurement of the robustness of the method m. A large value of the sum indicates good robustness.

Fig. 24 shows the distribution of b_m of each method over the ten problems. For each method, the 10 values of b_m are stacked and the sum is given on top of the stack. Fig. 24 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 denotes MEC performs very well in different situations. Thus MEC is the most robust method among the compared methods.

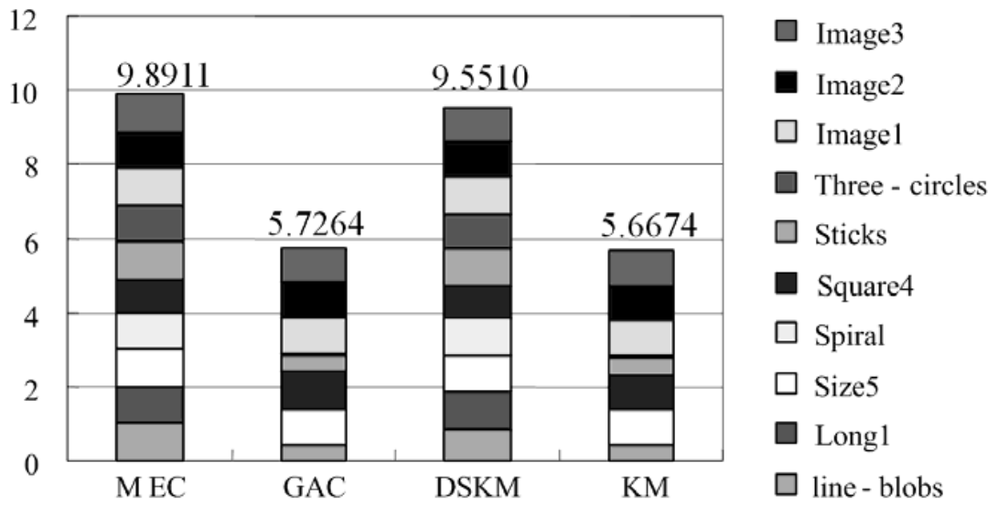


Fig. 24. Robustness of the compared algorithms

Fig. 25 illustrates the sum of the computing time of the four algorithms in solving the twelve problems at an IBM IntelliStation M Pro 6233. From Fig. 25, 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.

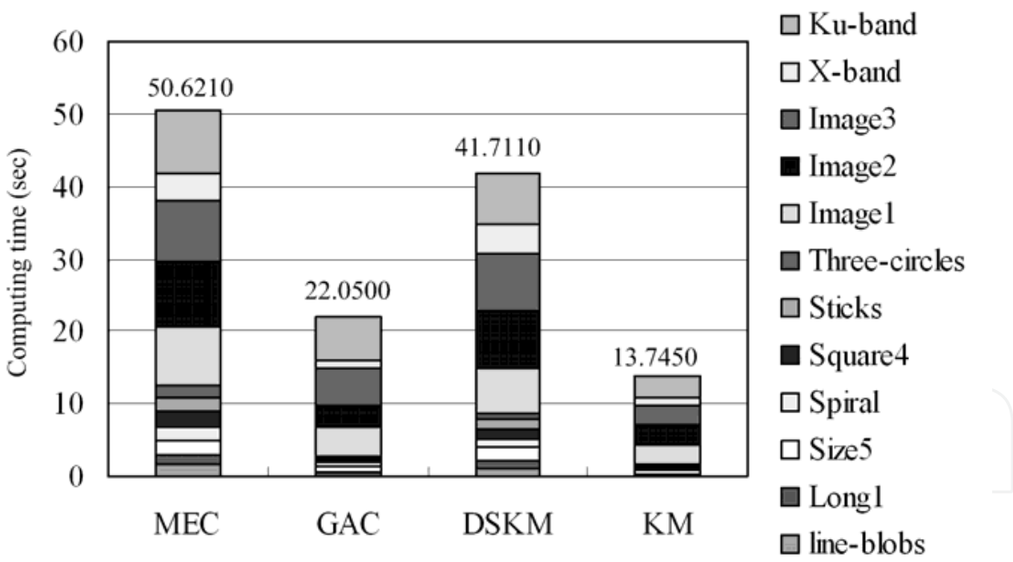


Fig. 25. Computing time of the compared algorithms.

4. Concluding remarks

In the first method, we proposed a novel image segmentation approach based on memetic algorithm called MISA. MISA applies the idea of clustering to achieve image segmentation task. In preprocessing phase, gray-level co-occurrence matrix and wavelet decomposition are used for feature extraction. The watershed segmentation is employed to segment images into non-overlap small regions. MISA tries to find the optimal combination of the watershed regions

under the criteria of interclass variance in feature space by a memetic algorithm. In MISA, after implementing cluster-based crossover and mutation, the individual learning procedure moves exocentric regions in current cluster to the one they should belong to according to the distance between these regions and cluster centers in feature space. Then, tournament selection and elitism strategy are used for producing the next generation. If stop criterion is satisfied, the segmentation result is outputted directly using the best individual in population.

In order to evaluate the new algorithm, six artificial texture images, three remote sensing images and three natural images are employed in experiments. The EGSA, SCEA, GISA, FCM and KM are our compared algorithms. We exhibit typical segmentation results, convergence curves for all kinds of images, and numerical results on artificial texture images for which the true partitioning is known. Experimental results showed that MISA outperformed GISA on most of the tested images. The only difference between GISA and MISA lies in the individual learning strategy. Thus the new improvement of MISA could benefit from the learning operator. The comparisons between MISA and the compared algorithms showed that MISA was an effective image segmentation approach.

Image segmentation remains a challenging problem. The main contribution of this study is to make substantial progress through the introduction of memetic computing methods to solving this problem. This study also shows that MAs provide useful computational tools. However we only designed one local search technique for image segmentation problem in this method. We will try to design more individual learning methods to cooperate together as well as higher order learning strategy in our future work.

In this study, we have attempted to illustrate the power of MA for segmenting the three kinds of images, namely texture images, remote sensing images and natural images. In fact, it is difficult or impossible to design an always powerful general-purpose algorithm. Thus, applying the proposed algorithm for practical applications such as magnetic resonance imaging (MRI) image segmentation and synthetic aperture radar (SAR) image segmentation with domain-specific knowledge is also planned in our future work.

In the second method, we proposed the manifold evolutionary clustering using a novel representation method and a manifold distance-based dissimilarity measure to solve 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 made up the drawbacks of DSKM by evolutionary searching the 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 computing method of the manifold distance is also our future work.

5. References

[1] M. Wertheimer, “Laws of Organization in Perceptual Forms(partial translation),” A Sourcebook of Gestalt Psycychology, W.B.Ellis, ed., pp 71-88, Harcourt, Brace, 1938.

[2] J. Shi, J. Malik. Normalized cuts and image segmentation.IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI, vol. 22, no. 8, pp 888-905, 2000.

[3] L. G. Shapiro and G. C. Stockman. Computer Vision. New Jersey, Prentice-Hall, ISBN 0-13-030796-3, pp 279-325, 2001.
[4] K.V. Mardia and T.J. Hainsworth. A Spatial Thresholding Method for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 6, pp. 919-927, 1988.
[5] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.12, no. 7, pp. 629-639, 1990.
[6] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos. Hybrid Image Segmentation Using Watersheds and Fast Region Merging. IEEE Transactions on Image Processing, vol. 7, no. 12, pp. 1684-1699, 1998.
[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. International Journal of Computer Vision, vol. 59, no. 2, pp. 167-181, 2004.
[8] J. A. Hartigan and M. A.Wong. A K-Means clustering algorithm. Applied Statistics, vol. 28, pp. 100-108, 1979.
[9] B. Bhanu, S. Lee, J. Ming. Adaptive image segmentation using a genetic algorithm. IEEE Transactions on Systems, Man and Cybernetics, vol. 25, no. 12, pp. 1543-1567, 1995.
[10] S. M. Bhandarkar, H. Zhang. Image segmentation using evolutionary computation. IEEE Transactions on Evolutionary Computation, vol. 3, no. 1, pp. 1-21, 1999.
[11] P. Andrey. Selectionist relaxation: Genetic algorithms applied to image segmentation. Image and Vision Computing, vol. 17, no. 3-4, pp. 175-187, 1999.
[12] C. J. Veenman, M. J. T. Reinders, E. Backer. A cellular coevolutionary algorithm for image segmentation. IEEE Transactions on Image Processing, vol. 12, no. 3, pp. 304-313, 2003.
[13] M. Koppen, K. Franke, R. Vicente-Garcia. Tiny GAs for image processing applications. IEEE Computational Intelligence Magazine, vol. 1, no. 2, pp. 17-26, May 2006.
[14] K. E. Melkemi, M. Batouche, and S. Foufou. A multiagent system approach for image segmentation using genetic algorithms and extremal optimization heuristics. Pattern Recognition Letters, vol. 27, no. 11, pp. 1230-1238, 2006.
[15] Maoguo Gong, Licheng Jiao, Liefeng Bo, Ling Wang, Xiangrong Zhang. Image Texture Classification Using a Manifold Distance based Evolutionary Clustering Method. Optical Engineering, vol. 47, no. 7, pp. 077201-1–10, 2008.
[16] U. Maulik. Medical Image Segmentation Using Genetic Algorithms. IEEE Transactions on Information Technology in Biomedicine, vol. 13, no. 2, pp. 166-173, 2009.
[17] Q. C. Nguyen, Y. S. Ong and M. H. Lim. A Probabilistic Memetic Framework. IEEE Transactions on Evolutionary Computation, vol. 13, no. 3, pp. 604-623, 2009.
[18] K. W. C. Ku, M. W. Mak, and W. C. Siu. A study of the Lamarckian evolution of recurrent neural networks. IEEE Transactions on Evolutionary Computation, vol. 4, no. 1, pp. 31-42, 2000.
[19] S. A. Kazarlis, S. E. Papadakis, J. B. Theocharis, V. Petridis. Microgenetic algorithms as generalized hill-climbing operators for GA optimization. IEEE Transactions on Evolutionary Computation, vol. 5, no. 3, pp. 204-217, 2001.
[20] Y. S. Ong and A. J. Keane. Meta-lamarckian learning in memetic algorithms. IEEE Transactions on Evolutionary Computation, vol. 8, no. 2, pp. 99–110, 2004.
[21] N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Transactions on Evolutionary Computation, vol. 9, no. 5, pp. 474–488, 2005.

[22] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner. A fast adaptive memetic algorithm for online and offline control design of pmsm drives. IEEE Transactions on Systems, Man and Cybernetics - Part B, vol. 37, no. 1, pp. 28-41, 2007.
[23] M. Tang and X. Yao. A memetic algorithm for VLSI floorplanning. IEEE Transactions on Systems, Man and Cybernetics - Part B, vol. 37, no. 1, pp. 62–69, 2007.
[24] J. Tang, M. H. Lim, and Y. S. Ong. Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems. Soft Computation, vol. 11, no. 9, pp. 873-888, 2007.
[25] Z. Zhu, Y. S. Ong and M. Dash. Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework. IEEE Transactions On Systems, Man and Cybernetics - Part B, vol. 37, no. 1, pp. 70-76, Feb 2007.
[26] S. Hasan, R. Sarker, D. Essam, and D. Cornforth. Memetic algorithms for solving job-shop scheduling problems. Memetic Computing, vol. 1, no. 1, pp. 69-83, 2009.
[27] Jong-Hwan Kim, Ye-Hoon Kim, Seung-Hwan Choi, In-Won Park. Evolutionary multi-objective optimization in robot soccer system for education. IEEE Computational Intelligence Magazine, vol. 4, no. 1, pp. 31-41, February 2009 .
[28] Shinkyu Jeong, S. Hasegawa, K. Shimoyama, S. Obayashi. Development and investigation of efficient GA/PSO-HYBRID algorithm applicable to real-world design optimization. IEEE Computational Intelligence Magazine, vol. 4, no. 3, pp. 36-44, Aug. 2009.
[29] S. C. Chiam, K. C. Tan, and A. Al. Mamun. A memetic model of evolutionary PSO for computational finance applications. Expert Systems With Applications, vol. 36, no. 2, pp. 3695-3711, March 2009.
[30] K. C. Tan, S. C. Chiam, et al. Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization. European Journal of Operational Research, vol. 197, no. 2, pp. 701-713, 2009.
[31] J. H. Ang, K. C. Tan, and A. A. Mamun. An evolutionary memetic algorithm for rule extraction. Expert Systems With Applications, vol. 37, no. 2, pp. 1302-1315, March 2010.
[32] Y. S. Ong, M. H. Lim, X Chen. Research Frontier: Towards Memetic Computing. Technical Report, School of Computer Engineering, Nanyang Technological University, Singapore, December 2009.
[33] E. Fernandez, M. Grana, J. Cabello. An instantaneous memetic algorithm for illumination correction. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1, pp 1105-1110, 2004.
[34] K. Batenburg. An evolutionary algorithm for discrete tomography. Discrete Applied Mathematics, vol. 151, no. 1-3, pp 36-54, 2005.
[35] V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, T. Rossi. An enhanced memetic differential evolution in filter design for defect detection in paper production. Evolutionary Computation, vol. 16, no. 4, pp 529-555, 2008.
[36] V. D. Gesù, G. L. Bosco, F. Millonzi, C. Valenti. A Memetic Algorithm for Binary Image Reconstruction. Lecture Notes in Computer Science, vol. 4958, Springer, pp 384-395, 2008.
[37] V. D. Gesù, G. L. Bosco, F. Millonzi, C. Valenti. Discrete tomography reconstruction through a new memetic algorithm. Lecture Notes in Computer Science, vol. 4974, Springer, pp 347-352, 2008.

[38] Qingzhou Zhang, Ziqiang Wang and Dexian Zhang. Memetic Algorithm-Based Image Watermarking Scheme. Lecture Notes in Computer Science, vol. 5263, Springer, pp 845-853, 2008.
[39] J. C. Bezdek. Pattern Recognition with Fuzzy Objective Function Algoritms. Plenum Press, New York, 1981.
[40] Xiangrong Zhang, Licheng Jiao, Fang Liu, Liefeng Bo, Maoguo Gong. Spectral Clustering Ensemble Applied to Texture Features for SAR Image Segmentation. IEEE Transactions on Geoscience and Remote Sensing, vol. 46, no. 7, pp 2126-2136, 2008.
[41] R. M. Haralick, K. Shanmugam and I. Dinstein. Textural features for images classification. IEEE Transactions on Systems, Man, and Cybernetics, vol. 3, no. 6, pp. 610-621, 1973.
[42] S. Fukuda and H. Hirosawa. A wavelet-based texture feature set applied to classification of multifrequency polarimetric SAR images. IEEE Transactions on Geoscience and Remote Sensing, vol. 37, no. 8, pp. 2282-2286, 1999.
[43] D. A. Clausi and B. Yue. Comparing Cooccurrence Probabilities and Markov Random Fields for Texture Analysis of SAR Sea Ice Imagery. IEEE Transactions on Geoscience and Remote Sensing, vol. 42, no. 1, pp. 215-228, 2004.
[44] D.A. Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization," Canadian Journal of Remote Sensing, vol. 28, no. 1, pp. 45-62, 2002.
[45] D. Wang. A Multiscale Gradient Algorithm for Image Segmentation Using Watersheds. Pattern Recognition, vol. 30, no. 12, pp. 2043-2052, 1997.
[46] A. Hoeffler, U. Leysner, and J. Weidermann. Optimization of the layout of trusses combining strategies based on Mitchel's theorem and on biological principles of evolution. In: Proceedings of the Second Symposium on Structural Optimization, Milan, Italy, 1973.
[47] P. Moscato. On Evolution, Search, Optimization Algorithms and Martial Arts: Towards Memetic Algorithms. Report 826, Caltech Concurrent Computation Program, California Institute of Technology, Pasadena, 1989.
[48] L. O. Hall, I. B. Ozyurt and J. C. Bezdek. Clustering with a genetically optimized approach. IEEE Transactions on Evolutionary Computation, vol. 3, no. 2, pp. 103-112, 1999.
[49] U. Maulik and S. Bandyopadhyay. Genetic algorithm-based clustering technique. Pattern Recognition, vol. 33, no. 9, pp. 1455-1465, 2000.
[50] H. Pan, J. Zhu and D. Han. Genetic algorithms applied to multiclass clustering for gene expression data. Genomics, Proteomics & Bioinformatics, vol. 1, no. 4, pp. 279-287, 2003.
[51] Maoguo Gong, Licheng Jiao, Ling Wang, Liefeng Bo. Density-Sensitive Evolutionary Clustering. In: Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD07. Springer-Verlag, Lecture Notes in Computer Science, LNAI, vol. 4426, pp. 507-514, 2007.
[52] P. Brodatz. Textures: A Photographic Album for Artists and Designers. Dover Publications, New York, 1966.
[53] 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).
[54] D. A. Clausi and B. Yue, "Comparing Cooccurrence Probabilities and Markov Random Fields for Texture Analysis of SAR Sea Ice Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, 42(1), 215-228 (2004).

[55] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural features for images classification," *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), 610-621 (1973).
[56] H. Frigui and R. Krishnapuram, "A Robust Competitive Clustering Algorithm with Applications in Computer Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 450-465 (1999).
[57] Y. Leung, J. Zhang and Z. Xu, "Clustering by Space-Space Filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), 1396-1410 (2000).
[58] J. A. Hartigan and M. A.Wong, "A K-Means clustering algorithm," *Applied Statistics*, 28, 100-108 (1979).
[59] L. O. Hall, I. B. Ozyurt and J. C. Bezdek, "Clustering with a genetically optimized approach, " *IEEE Transactions on Evolutionary Computation*, 3(2), 103-112 (1999).
[60] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique, " *Pattern Recognition*, 33(9), 1455-1465 (2000).
[61] H. Pan, J. Zhu and D. Han, "Genetic algorithms applied to multiclass clustering for gene expression data," *Genomics, Proteomics & Bioinformatics*, 1(4), 279-287 (2003).
[62] J. Handl, J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Transactions on Evolutionary Computation*, 11(1), 56-76 (2007).
[63] M.C. Su and C. H. Chou, "A modified version of the K-Means algorithm with a distance based on cluster symmetry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6), 674-680 (2001).
[64] D. Charalampidis, "A Modified K-Means Algorithm for Circular Invariant Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), 1856-1865 (2005).
[65] L. Wang, L. F. Bo and L. C. Jiao, "A modified K-Means clustering with a density-sensitive distance metric," *Lecture Notes in Computer Science*, 4062, 544-551 (2006).
[66] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," In *Proceedings of the Eighteenth International Conference on Machine Learning*, 19-26 (2001).
[67] O. Bousquet, O. Chapelle and M. Hein, "Measure based regularization," *Advances in Neural Information Processing Systems 16 (NIPS)*, MIT Press, Cambridge, MA (2004).
[68] G. Syswerda, "Uniform crossover in genetic algorithms," In *Proceedings of the Third International Conference on Genetic Algorithms*, pp2-9, Morgan Kaufmann Publishers, San Francisco, CA (1989).
[69] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, 4, 65-85 (1994).
[70] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Massachusetts (1989).
[71] 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).
[72] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, 193-218 (1985).
[73] K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, 17(9), 763-774 (2001).
[74] W. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, 66(336), 846-850 (1971).
[75] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover Publications, New York (1966).
[76] X. Geng, D. C. Zhan and Z. H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 35(6), 1098-1107 (2005).

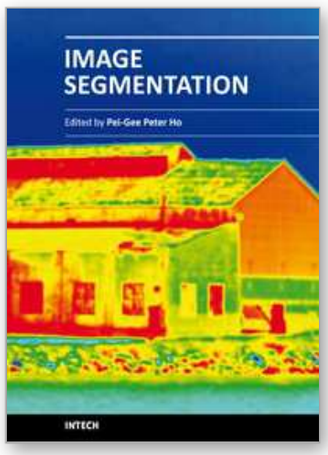


Image Segmentation

Edited by Dr. Pei-Gee Ho

ISBN 978-953-307-228-9
Hard cover, 538 pages
Publisher InTech
Published online 19, April, 2011
Published in print edition April, 2011

It was estimated that 80% of the information received by human is visual. Image processing is evolving fast and continually. During the past 10 years, there has been a significant research increase in image segmentation. To study a specific object in an image, its boundary can be highlighted by an image segmentation procedure. The objective of the image segmentation is to simplify the representation of pictures into meaningful information by partitioning into image regions. Image segmentation is a technique to locate certain objects or boundaries within an image. There are many algorithms and techniques have been developed to solve image segmentation problems, the research topics in this book such as level set, active contour, AR time series image modeling, Support Vector Machines, Pixon based image segmentations, region similarity metric based technique, statistical ANN and JSEG algorithm were written in details. This book brings together many different aspects of the current research on several fields associated to digital image segmentation. Four parts allowed gathering the 27 chapters around the following topics: Survey of Image Segmentation Algorithms, Image Segmentation methods, Image Segmentation Applications and Hardware Implementation. The readers will find the contents in this book enjoyable and get many helpful ideas and overviews on their own study.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Licheng Jiao (2011). Evolutionary-based Image Segmentation Methods, Image Segmentation, Dr. Pei-Gee Ho (Ed.), ISBN: 978-953-307-228-9, InTech, Available from: http://www.intechopen.com/books/image-segmentation/evolutionary-based-image-segmentation-methods

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen