Journal of
Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

Independent feature subspace iterative optimization based fuzzy clustering for synthetic aperture radar image segmentation

Hang Yu
Luping Xu
Dongzhu Feng
Xiaochuan He

SPIE.

Independent feature subspace iterative optimization based fuzzy clustering for synthetic aperture radar image segmentation

Hang Yu, Luping Xu, Dongzhu Feng,* and Xiaochuan He
Xidian University, School of Aerospace Science and Technology, 2 South Taibai Road,
Xi'an 710071, China

Abstract. Synthetic aperture radar (SAR) image segmentation is investigated from feature extraction to algorithm design, which is characterized by two aspects: (1) multiple heterogeneous features are extracted to describe SAR images and the corresponding similarity measures are developed independently to avoid the mutual influences between different features in order to enhance the discriminability of the final similarity between objects. (2) A method called fuzzy clustering based on independent subspace iterative optimization (FCISIO) is proposed. FCISIO integrates multiple features into an objective function which is then iteratively optimized in each feature subspace to obtain final segmentation results. This strategy can protect the distribution structures of the data points in each feature subspace, which realizes an effective way to integrate multiple features of different properties. In order to improve the computation speed and the accuracy of feature description for FCISIO, we design a region merging algorithm before FCISIO which can use many kinds of information to quickly merge regions inside the true segments. Experiments on synthetic and real SAR images show that the proposed method is effective and robust and can obtain good segmentation results with a very short running time. [DOI: 10.1117/1.JRS.9.095060]

Keywords: multifeatures ensemble; image segmentation; clustering algorithm; synthetic aperture radar image.

Paper 15143 received Feb. 20, 2015; accepted for publication Aug. 10, 2015; published online Sep. 11, 2015.

1 Introduction

With the advantages of all weather, all time, and strong permeability, synthetic aperture radar (SAR) images have found wide applications, such as resource management, environmental, archaeology, military, and so on. As a fundamental step for many image-processing tasks such as SAR image understanding and content-based image retrieval, SAR image segmentation is a process that decomposes an image into disjoint regions of different characteristics. Effective segmentation of SAR images can not only reduce computation cost but also increase the efficiency for further SAR image processing tasks. Compared with other kinds of images (such as optical images, infrared images, and so on), SAR images own some specific properties. For example, (1) due to the coherent processing of backscattered signals from multiple distributed targets, SAR images contain speckle noise, which leads to random changes of pixels' brightness.[1] (2) The exhibition of the same terrain target in SAR images is often nonstationary[2] and has complex variation which, on the other side, make different objects have a similar appearance. (3) Different radars or the same radar with different parameters will produce different SAR images even for the same scene. (4) SAR images contain tremendous information. These properties of SAR images bring great challenges to the segmentation task.

On one side, we need effective and efficient features to accurately describe SAR images. Actually, there are many kinds of information in SAR images. First, due to unique geometric

*Address all correspondence to: Dongzhu Feng, E-mail: dongzhufengnet@163.com structures, moisture levels, and canopy surface roughness, different land covers in SAR images will produce different brightnesses and textures. Second, there are obvious edges in SAR images, for example, the coast and riversides in natural scenes, and roads and buildings in man-made scenes. In addition, the objects in SAR images are also spatially related. Thus, many kinds of features are extracted for SAR images, such as the brightness after denoising,[3,4] texture,[5–9] and edges.[10–13] These features can describe some properties of SAR images, but no single feature can completely and accurately characterize miscellaneous objects in SAR images. The pixels' brightnesses always change randomly due to the speckle noise, different terrain surfaces often show similar textures, and the speckle noise and large-scale texture often create positive false edges. Therefore, it is necessary to integrate multiple heterogeneous features to accurately describe SAR images. Some work has been done on this subject. Deng and Clausi[14] adopted image brightness to discriminate ice from water and further used texture to identify distinct ice types. Yu and Clausi[15] combined brightness and edges into the traditional Markov random field context model to obtain more accurate segmentation. The texture and spatial information have been fused in kernel density estimation to segment SAR images.[16] However, how to integrate heterogeneous features of different structures is still a difficult task and the related work is rare.

On the other side, effective and efficient segmentation schemes are also necessary to manage different kinds of features. Researchers have proposed many kinds of segmentation algorithms for SAR images, which include threshold methods,[3,17–19] spectral clustering (SC) algorithms,[20,21] statistic model-based methods,[1,14,15,22–25] artificial intelligence methods,[26–30] support vector machine (SVM),[6,31] region growing methods,[15,32–35] and so on. Among these algorithms, cluster-based algorithms form one popular and representative family, whose main idea is to group pixels in such a way that the pixels in the same group are more similar to each other than those in other groups. The key point in these algorithms is to define an objective function (or a criterion) that computes the overall similarity (or dissimilarity) of clusters (segments), which thus decides the final image segmentation. One direct solution to improve the accuracy and robustness of the objective function is to extract more information from SAR images. Such considerations have driven the emergence of a large amount of literatures[5–9,20,27,29,36,37] concerning the texture classification of SAR images. Clausi[9] carefully compared and integrated different texture features into the classification task of SAR ice images. Kandaswamy et al.[8] proposed a statistical occupancy model to analyze the efficiency of different texture features in SAR image classification. However, so far as we know, little research has been done to combine texture and brightness together to describe SAR images, which are two different and complementary features for SAR image interpretation. This is mainly because of the different structures of texture and brightness, which will be further discussed in Sec. 2.2. Another solution for improvement of the objective function is to take advantage of some data mining technologies on the extracted features so as to accurately compute the similarity (or dissimilarity) between operation elements (e.g., pixels). Using this train of thought, Zhang et al.[20] applied SC based on eigenvector decomposition to SAR image segmentation, which can recognize the clusters of unusual shapes and obtain the global optimal solutions in a relaxed continuous domain. In order to accurately characterize the structure of clusters, Yang et al.[29] adopted two conflicting and complementary objective functions, and they proposed a multiobjective optimization algorithm for texture classification of SAR images. Some machine learning techniques, such as SVM,[6,31,37] have also been modified for SAR image texture classification.

Based on the consideration above, we study the problem of SAR image segmentation from the feature extraction to the algorithm design in this paper. First, we extract multiple heterogeneous features to accurately describe SAR images, which include brightness features, texture features, and edge features. Then in order to maximize the discriminability of features, three kinds of similarity measures are independently developed based on the properties of the corresponding features. The three similarity measures can accurately compute the similarity from different aspects. Second, a method called fuzzy clustering based on independent subspace iterative optimization (FCISIO) is proposed to manage the multiple heterogeneous features. The designed objective function of FCISIO is iteratively optimized in each independent subspace to avoid the mutual influence between different kinds of features. In FCISIO, we adopt over-segmented regions as the operation elements to reduce the influence of speckle noise and the computation cost. In order to further improve the computation speed and the accuracy of feature description for FCISIO, we design a region merging algorithm before FCISIO, which can make use of more information to quickly merge regions inside the true segments. The proposed method is compared with four state-of-the-art segmentation algorithms by experiments on synthetic and real SAR images, which show that the extracted features are robust and accurate to describe SAR images, and the proposed algorithm can obtain a balance between the computation speed and the segmentation accuracy.

The remainder of this paper is organized as follows. Section 2 describes the multifeatures' extraction and similarity measures for SAR images. The proposed segmentation method will be presented in Sec. 3. In Sec. 4, the proposed method is validated by experiments on synthetic and real SAR images. Summary and future works comprise Sec. 5.

2 Multifeatures Extraction and Similarity Measures for Synthetic Aperture Radar Images

Without loss of generality, let $\mathbf{S} = \{(r, c)|1 \leq r \leq R, 1 \leq c \leq C\}$ denote the discrete rectangular lattice, and the input SAR images are defined as $\mathbf{I} = \{I(r, c)|(r, c) \in \mathbf{S}\}$. Then the segmentation problem is formulated to decomposes the input image into L disjoint regions, satisfying

1. $s_i = \{(r, c)|l(r, c) = i, 1 \leq i \leq L, (r, c) \in \mathbf{S}\}$,
2. $\bigcup_{i=1}^{L} s_i = \mathbf{S}$,
3. $\forall\, i \neq j,\ s_i \cap s_j = \varnothing$.

Here, $l(r, c)$ represents the label of pixel (r, c) in the segmentation result. The number L of true segments is assumed to be known beforehand in this paper.

2.1 *Preprocessing to Obtain Oversegmented Regions*

SAR images suffer from speckle noise, which make the pixel-based methods run very slowly and produce errors. To cope with these problems, oversegmented regions are adopted as the operation elements instead of pixels in this paper. An oversegmented region consists of a group of pixels that are consistent inside according to some constraints such as brightness, texture, and so on. There are many algorithms that can produce oversegmented regions, such as the mean shift algorithm,[38] the normalized cuts algorithm,[39] the graph-based segmentation algorithm,[40] and the watershed segmentation algorithm.[41] In this paper, a level-set method called TurboPixels[42] is utilized to oversegment an input image into N_S regions $\{s_i\}_{i=1}^{N_S}$, where all the pixels in one region have the same label and will be treated as a whole in the proposed algorithm. Five examples by TurboPixels are illustrated in Fig. 1 for real SAR images. Note that these oversegmented regions preserve most edges of the images and are approximately uniform in size and shape, which makes them very suitable for further processing task of SAR images.

2.2 *Multifeatures Extraction for Synthetic Aperture Radar Images*

Multiple heterogeneous features are extracted for SAR images in this section. The first kind of feature is brightness, which reflects the microwave scattering properties of the area on earth in

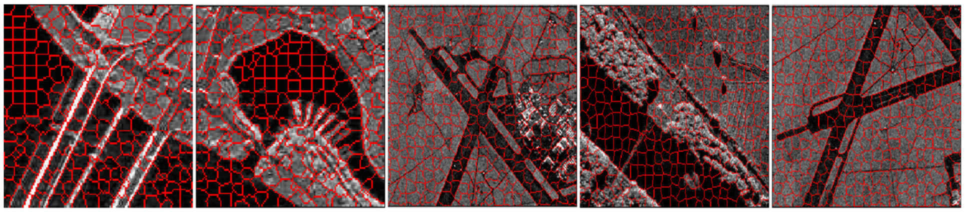

Fig. 1 The oversegmented regions of five real synthetic aperture radar (SAR) images overlapping the original images.

SAR images. The brightness feature $f^B(r,c)$, $(r,c) \in \mathbf{S}$, is obtained by linearly normalizing the SAR image $I(r,c)$ to [0,1] such as the example shown in Fig. 2(a).

The second kind of feature is texture, which describes the characteristic structure of a terrain surface in a finite sampling window. Texture is an important tool for identifying land covers in SAR image, and many methods have been proposed to extract the texture features of SAR images, such as gray level co-occurrence matrix,[43] wavelet transforms,[44] and Markov random field (MRF).[9] Because of the relationship with the visual perception of human beings and the optimal joint localization in both space and frequency domains, Gabor filters have been widely used to describe texture in images.[9,43] In this paper, a heuristically designed Gabor filter bank[45] is utilized to produce multiscale texture features of SAR images. The method in Ref. 45 first filters the input image using a bank of Gabor filters with different scales ω and orientations θ, then the normalized magnitudes of Gabor filter responses are computed to produce texture images. Finally, the texture feature for pixel $(r,c) \in \mathbf{S}$ is made up of the values of all texture images at (r,c) in the form of a row vector denoted as $\mathbf{f}^T(r,c) = [f_1^T(r,c), f_2^T(r,c), \cdots, f_{N_\omega \times N_\theta}^T(r,c)]$, where $f_i^T(r,c)$ stands for the value of the i'th texture image at pixel (r,c), $i = 1,2,\cdots,N_\omega \times N_\theta$. N_ω and N_θ are the number of scales and orientations in the Gabor filter bank, respectively.

The parameters of the Gabor filter bank play an important role in extraction of texture features. In this paper, the parameters are set by an unsupervised self-adapted way as follows. Since a Gabor filter is nonorthogonal basis set, in order to reduce redundancy, the half-peak magnitude supports of the filter responses in the frequency spectrum are made to touch each other.[46] Based on this consideration, the scale ω of the Gabor filter bank is equal to $\sqrt{2}/4, \sqrt{2}/8, \ldots, \sqrt{2}/2^{N_\Delta - 4}$, where $N_\Delta = \lfloor(\log_2[\min(R,C)])\rfloor$, $\lfloor\bullet\rfloor$ denotes the largest integer that is no larger than its argument. The orientation θ is set as 0, 30, 60, 90, 120, and 150 deg. The study[46] shows that this strategy for setting θ and ω is in agreement with the characteristic of the human visual system. Figure 2(b) illustrates four isotropic texture images of different scales with respect to Fig. 2(a), from which we can notice that the multiscale texture images can effectively characterize different textures in the real SAR image.

Especially noteworthy is that the brightness feature and the texture feature are two important and complementary features for SAR image interpretation, which describe SAR images from different aspects. The terrain objects with different microwave scattering properties show obvious different brightnesses in SAR images, e.g., the farm area and the river area in Fig. 2(a). However, different objects sometimes display similar brightness, e.g., some forest area and the farm area in Fig. 2(a). In this situation, the texture feature can make a clear separation just as illustrated in Fig. 2(b). On the other side, the brightness can also complement the texture. Take the farm area and the river area in Fig. 2 for instance, which have similar texture features but apparently different brightness features. Therefore, it is quite necessary to integrate the brightness and the texture together to characterize SAR images. However, most of the existing segmentation methods for SAR images rely on either brightness or texture, and only a few recent

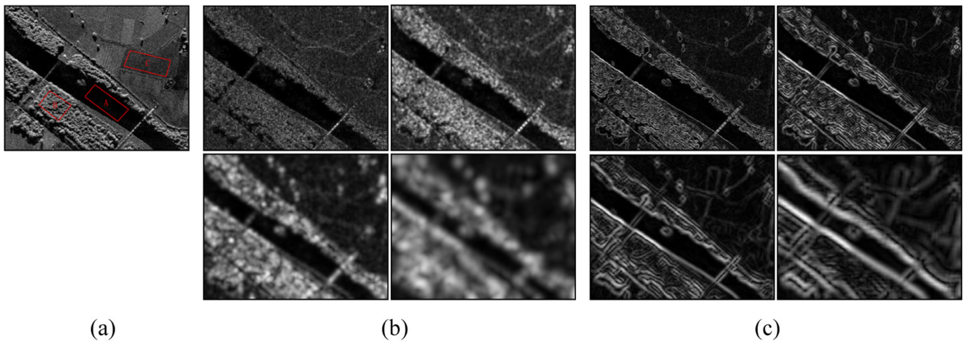

Fig. 2 Multifeatures extraction for a real SAR image. (a) A brightness SAR image where the three areas with red borders denote three kinds of terrain surfaces: (A) river area, (B) forest area, and (C) farm area. (b) Four Gabor texture images of different scales for (a). (c) The multiscale edge images for (a).

approaches[35,47] incorporate brightness and texture simultaneously into the same framework. One major reason is the different structures (such as dimensions and ranges) of the brightness feature and the texture feature.

Besides brightness and texture, SAR images are full of edges such as coast and riversides in natural scenes and the bridges and roads in man-made scenes. Comparing with brightness and texture, edges can not only accurately locate boundaries between objects, but can also be particularly helpful to characterize the gradually changing regions of the same class which often invalidate the appearance-based features. However, due to speckle noise and the coexistence of multiscale objects, it is not as easy to accurately extract edges in SAR images as in ordinary images. In this paper, a multiscale edge detector[35] is utilized to extract the edge features, as shown below. The input image is first convolved with a bank of Prewitt kernels of different scales to obtain the multiscale edge images, which can be written as below.

$$\mathbf{G}_{\lambda}^{r} = \mathbf{I} \otimes \begin{bmatrix} \begin{array}{ccc|c|ccc} -1 & \cdots & -1 & 0 & 1 & \cdots & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & \cdots & -1 & 0 & 1 & \cdots & 1 \end{array} \end{bmatrix}_{\lambda\times(2\lambda+1)},$$

$$\mathbf{G}_{\lambda}^{c} = \mathbf{I} \otimes \begin{bmatrix} \begin{array}{ccc} -1 & \cdots & -1 \\ \vdots & \ddots & \vdots \\ -1 & \cdots & -1 \\ \hdashline 0 & \cdots & 0 \\ \hdashline 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{array} \end{bmatrix}_{(2\lambda+1)\times\lambda}, \tag{1}$$

where λ is the scale parameter of the Prewitt kernel, $\otimes$ denotes the convolution operation, and $\mathbf{G}_{\lambda}^{r}$ and $\mathbf{G}_{\lambda}^{c}$ are the vertical and horizontal edge images at a given scale λ, respectively. When λ increases, the size of the Prewitt kernel and the scale of the extracted edges will increase. The basic kernel type is not limited to Prewitt, and other derivative filters (such as Roberts filter and Sobel filter[48]) can also be options that experimentally result in similar results. Next, the orientation information is eliminated by the computation of the normalized magnitude of the horizontal and the vertical edge feature images. Then the edge feature for pixel $(r,c)\in\mathbf{S}$ is composed of the values of all edge images at (r,c) and is denoted as $\mathbf{f}^{E}(r,c)=[f_{1}^{E}(r,c), f_{2}^{E}(r,c), \cdots, f_{N_{\lambda}}^{E}(r,c)]$, where $f_{i}^{E}(r,c)$ stands for the value of the i'th edge image at pixel (r,c), $i=1,2,\cdots,N_{\lambda}$. N_{λ} is the scale number of Prewitt kernels, which is set the same as the scales of the Gabor filter bank above. An example of multiscale edge images corresponding to Fig. 2(a) is shown in Fig. 2(c). It can be clearly noticed that various edges in the SAR image have been effectively found by the multiscale edge detector, which yields another cue about the terrain surface in SAR images.

After obtaining the pixel-level features, we next extend them to region-level following an average strategy: for the brightness and texture features, the feature of any oversegmented region is defined as the average of the corresponding features of all pixels in the region; for the edge feature, the edge feature of any pair of adjacent regions is defined as the average of the edge features of all pixels along the boundary between them. This averaging strategy can improve the accuracy and robustness of the feature description and reduce the negative influence of speckle noise. For any two adjacent regions s_m and s_n, $m\neq n$, m, $n=1,2,\cdots,N_S$, the brightness feature, the texture feature, and the edge feature are computed by

$$f_{m}^{sB} = \frac{\sum_{(r,c)\in s_m} f^{B}(r,c)}{|s_m|}, \quad \mathbf{f}_{m}^{sT} = \frac{\sum_{(r,c)\in s_m} \mathbf{f}^{T}(r,c)}{|s_m|}, \quad \mathbf{f}_{mn}^{sE} = \frac{\sum_{(r,c)\in\mathbf{B}(s_m,s_n)} \mathbf{f}^{E}(r,c)}{|\mathbf{B}(s_m,s_n)|}, \tag{2}$$

where f_{m}^{sB} and $\mathbf{f}_{m}^{sT}$, respectively, denote the brightness feature and the texture feature of s_m, $\mathbf{f}_{mn}^{sE}$ represents the edge feature between s_m and s_n, $|\bullet|$ computes the cardinality of the argument, and $\mathbf{B}(s_m,s_n)$ stands for the set of pixels on the boundary between s_m and s_n.

2.3 *Similarity Measures*

Because the multiple heterogeneous features are generated by different methods, they will have different properties, e.g., different dimensions, different ranges, and different numbers of involved operation elements, which require separate measures of the similarity according to each kind of feature. Otherwise, if all dimensions from different features are treated equally, such as concatenating all features into a single vector and measuring the similarity by Euclidean distance, the feature with a high dimension or with a large range will become dominant. The edge feature involves two operation elements and is not straightforward to combine with the brightness feature and texture feature in such a scheme. In this section, three measures based on the properties of the different features extracted above are introduced to compute the brightness-based similarity, the texture-based similarity, and the edge-based similarity between oversegmented regions, which will be further integrated in the proposed algorithm in the next section.

Since the brightness feature is one-dimension, the brightness-based similarity is computed directly by the difference of the brightness features between oversegmented regions, which are denoted as below:

$$d_{mn}^{sB} = \|f_m^{sB} - f_n^{sB}\|_1, \tag{3}$$

where f_m^{sB} and f_n^{sB}, $m \neq n$, m, $n = 1,2, \cdots, N_S$ represent the brightness features of s_m and s_n, respectively. $\| \bullet \|_1$ computes the 1-norm distance of the arguments.

The texture feature is an $N_\omega \times N_\theta$-dimension vector, any value of which represents the response of the operation element to the Gabor filter with a specific scale and orientation. In SAR images, the texture responses of land covers show a great diversity in appearance as shown in Fig. 2(b), which mainly lies in two aspects: (1) the same types of objects are more sensitive to some scale or orientation than to others and (2) different objects usually have different response ranges. Take Fig. 3, for example, which plots the distributions of the texture features of three kinds of terrain surfaces shown in Fig. 2(a), where the scale and orientation increase from left to right along the horizontal axis. Figure 3(a) is for the river area, Fig. 3(b) is for the forest area, and Fig. 3(c) is for the farm area. Note that the water area has very low texture responses and can be considered as textureless; the texture of the forest area has obvious orientational selectivity; and the selected farm area is more sensitive to small scale filters than to the large ones. The texture features belonging to the three areas are different not only in range, but also in distribution pattern. Therefore, the texture-based similarity cannot simply be

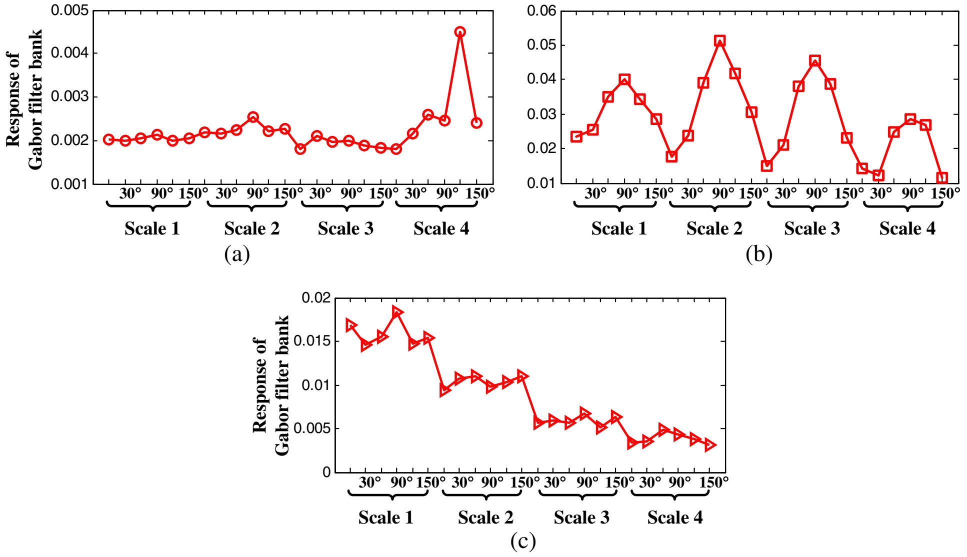

Fig. 3 The distribution of the texture features of three kinds of terrain surfaces shown in Fig. 2(a): (a) the river area, (b) the forest area, and (c) the farm area.

computed as the sum of the element-to-element distance between the texture features, even if a normalization operation is performed beforehand. For this reason, the texture-based similarity in this paper is computed by the χ^2 distance that is used to assess whether or not an observed frequency distribution differs from another distribution, which can be written as below:

$$d_{mn}^{sT} = \frac{1}{2} \sum_{k=1}^{N_\omega \times N_\theta} \frac{(f_{m,k}^{sT} - f_{n,k}^{sT})^2}{(f_{m,k}^{sT} + f_{n,k}^{sT})}, \tag{4}$$

where $f_{m,k}^{sT}$, $k = 1,2, \cdots, N_\omega \times N_\theta$, $m = 1,2, \cdots, N_S$ is a component of the texture feature $\mathbf{f}_m^{sT}$.

The edge feature is an N_λ-dimension vector. Similar to the texture feature, each value in this vector represents the corresponding response of an edge with a fixed scale. When the scale of the edge feature is very small, we can obtain many tiny edges such as shown in the upper left subfigure of Fig. 2(c), which also includes some false positive edges. This is mainly caused by speckle noise and the large-scale texture. As the scale of edge feature increases, the false positive edges gradually disappear such as shown in the upper right and bottom left subfigures of Fig. 2(c). This is because the Prewitt kernel of a large size can, in effect, reduce the influence of small local changes in images. But, when the scale of edge filter reaches too large of a value, the output responses will be blurred so as to reduce the accuracy of the edge location such as is shown in the bottom right subfigure of Fig. 2(c). In order to reduce the false positive edges and increase the accuracy of the edge location at the same time, a minimum operator is used to fuse the multiscale edge features to obtain the edge-based similarity, which can be written as below:

$$d_{mn}^{sE} = \min(f_{mn,1}^{sE}, f_{mn,2}^{sE}, \cdots, f_{mn,N_\lambda}^{sE}), \tag{5}$$

where $f_{mn,k}^{sE}$, $i = 1,2, \cdots, N_\lambda$, $m \neq n$, and $m, n = 1,2, \cdots, N_S$ stands for the k'th edge feature of any adjacent pair of s_m and s_n. $\min(\bullet)$ denotes the minimum operator, which is a T-norm operator in fuzzy sets and possibility theory,[49] performing a severe behavior: an edge will finally be deemed to exist, if and only if it is believed to exist in all channels of the edge feature.

In order to make the three kinds of similarity measures have an equal effect on assessing the similarity between oversegmented regions, a normalization operation is applied to linearly transform each type of similarity to [0,1], which results in three consistent similarity measures from different aspects. For any two regions s_m and s_n, $i = 1,2, \cdots, N_\lambda$, $m \neq n$, m, $n = 1,2, \cdots, N_S$, the normalization operation is performed as below:

$$\begin{cases} \tilde{d}_{mn}^{sB} = \frac{d_{mn}^{sB} - \min(\mathbf{D}^{sB})}{\max(\mathbf{D}^{sB}) - \min(\mathbf{D}^{sB})} \\ \tilde{d}_{mn}^{sT} = \frac{d_{mn}^{sT} - \min(\mathbf{D}^{sT})}{\max(\mathbf{D}^{sT}) - \min(\mathbf{D}^{sT})} \\ \tilde{d}_{mn}^{sE} = \frac{d_{mn}^{sE} - \min(\mathbf{D}^{sE})}{\max(\mathbf{D}^{sE}) - \min(\mathbf{D}^{sE})} \end{cases}, \tag{6}$$

where $\mathbf{D}^{sB}$, $\mathbf{D}^{sT}$, and $\mathbf{D}^{sE}$ represent the sets of the brightness-based similarity, the texture-based similarity, and the edge-based similarity between all pairs of oversegmented regions. Note that different features play different roles for different SAR images. For example, some images have more edge information, while some have more texture information. Therefore, $\mathbf{D}^{sB}$, $\mathbf{D}^{sT}$, and $\mathbf{D}^{sE}$ should include similarities of all pairs of oversegmented regions of all SAR images which can accurately rank the importance of each similarity. However, it is impractical to accurately compute $\mathbf{D}^{sB}$, $\mathbf{D}^{sT}$, and $\mathbf{D}^{sE}$. To deal with this problem, we use all pairs of oversegmented regions of all synthetic and real SAR images in this paper to make an approximation. $\max(\bullet)$ denotes the maximum operator. For convenience of expression, we still use d_{mn}^{sB}, d_{mn}^{sT}, and d_{mn}^{sE} to denote the normalized similarities $\tilde{d}_{mn}^{sB}$, $\tilde{d}_{mn}^{sT}$, $\tilde{d}_{mn}^{sE}$ in the following sections.

An example of the three kinds of similarity measures is illustrated in Fig. 4, where Fig. 4(a) is a real SAR image and Figs. 4(b)–4(d) illustrate the brightness-based similarity, the texture-based similarity, the edge-based similarity, respectively. In Figs. 4(b)–4(d), the values on each boundary denote the corresponding similarity between the two involved adjacent oversegmented regions. Note that each kind of similarity measure can accurately compute the similarity between adjacent regions from its corresponding aspect, e.g., the brightness-based similarity can clearly separate the runway area from other areas, the texture-based similarity can effectively distinguish

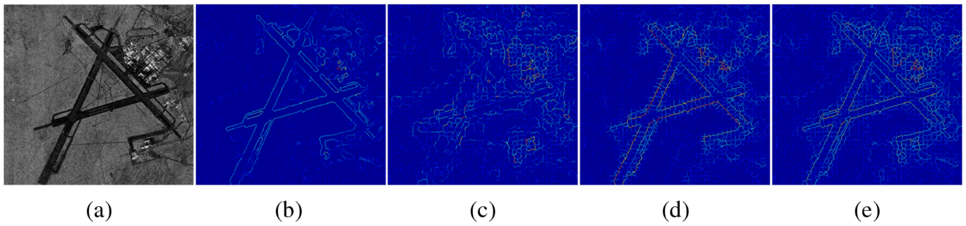

Fig. 4 Illustration of the three kinds of similarity measures and the fusion effect by Eq. (7): (a) a real SAR image; (b) the brightness-based similarity; (c) the texture-based similarity; (d) the edge-based similarity; and (e) the final ensemble similarity by Eq. (7). Among (b)–(e), the values on each boundary denote the corresponding similarity between the two involved adjacent oversegmented regions.

the building area, and the edge-based similarity can accurately locate the true boundary between different objects. However, no single similar measure can distinguish all kinds of objects, which confirms our previous view that these features are different and are complementary for describing the objects in SAR images.

3 Unsupervised Segmentation Algorithm

The purpose of the segmentation algorithm is to reduce the number N_S of the initial oversegmented regions to the number L of true segments. To realize this aim, we divide the segmentation process into two stages: (1) a fast region merging stage and (2) a fine region classification stage.

3.1 *Region Merging Stage*

Although the adoption of oversegmented regions can greatly reduce the number of operation elements and improve the robustness of features, there is still much space to further cut down their number, because what we are really concerned about in the segmentation problem is the oversegmented regions along the boundary between true segments. Many oversegmented regions inside true segments are actually not relevant to the quality of the segmentation results. In addition, when the scales of objects in SAR images are much larger than the size of the oversegmented regions, the features of the regions cannot effectively capture the characteristic structure of the underlying image contents and may lead to inaccurate descriptions. This will cause misclassification of global clustering methods. In order to speed up the segmentation process and improve the segmentation accuracy, a fast region merging algorithm is first performed on the initial oversegmented regions which can merge regions at a very high speed in order to cut down the computation burden for the further fine region classification stage.

The basic assumption of the region merging algorithm is that the more similar two adjacent oversegmented regions are, the more probably they will be in the same segment and can be merged. Since it is performed in the image domain, the region merging algorithm can naturally make use of all three kinds of similarities as defined in Sec. 2. Therefore, the merging criterion here is to globally merge the most similar adjacent oversegmented regions based on brightness, texture, and edge.

For the sake of explanation, let $\mathbf{G} = (\mathbf{S}, \mathbf{E})$ be a region adjacency graph (RAG), where each node $s_m \in \mathbf{S}$, $m = 1,2, \cdots, N_S$, represent an oversegmented region and each edge $e_{mn} \in \mathbf{E}$, $m \neq n$, m, $n = 1,2, \cdots, N_S$, represent a link between the adjacent pair of s_m and s_n. When we say s_m and s_n are adjacent, it means there is a common boundary between s_m and s_n. Since the three kinds of similarities originate from different aspects, it cannot be determined which one is more important than the others. For any two adjacent regions s_m and s_n, the similarity criterion in the region merging algorithm is defined by the product of the three kinds of similarities as follows:

$$d_{mn}^{sRMS} = d_{mn}^{sB} \times d_{mn}^{sT} \times d_{mn}^{sE}, \quad e_{mn} \in \mathbf{E}, \tag{7}$$

where d_{mn}^{sB}, d_{mn}^{sT}, and d_{mn}^{sE} are the brightness-based similarity, the texture-based similarity, and the edge-based similarity of s_m and s_n, respectively. In Eq. (7), we utilize the product operator to integrate the three kinds of similarities, which is a T-norm operator in fuzzy sets and possibility theory,[49] performing a strict judgment: two regions can be considered to be similar only when they simultaneously have high similarity according to the three kinds of similarities. An example of the fusion effect by Eq. (7) can be seen in Fig. 4(e). By the subjective observation, the final ensemble similarity can take the advantages of the three kinds of similarities to separate the different objects more accurately than any single similarity: the saliency of similarity along the boundary between true segments has been improved and that inside the segments has been suppressed.

Once the similarity criterion is obtained, a merging sequence can be defined in nondecreasing order. Then a greedy search will find the most similar region pair $\operatorname{argmin}_{e_{mn}\in\mathbf{E}}(d_{mn}^{sRMS})$ and merge them. This process will not terminate until the merging condition below is satisfied.

$$\begin{cases} \min_{e_{mn}\in\mathbf{E}}(d_{mn}^{sRMS}) \le \alpha \\ N_S' \ge L \end{cases}, \tag{8}$$

where α is the halt condition of the region merging algorithm, determining whether or not the regions are similar. N_S' denotes the number of the remaining regions in the image domain. The flowchart of the region merging algorithm is shown in Table 1. It is assumed by Eq. (8) that if $(d_{mn}^{sRMS}) \le \alpha$, then s_m and s_n are similar and can be merged together, and vice versa. The optimal value of α for each input image cannot be determined *a priori*, since it depends on the similarity between the true segments. This is related to the content of the image scene, which is unknown to us. For simple scenes, the similarity between different true segments is large, while for complex scenes, it is relatively small. Therefore, α will be selected by experiment. In this paper, α is set as $\alpha = 90\% \times \max_{e_{mn}\in\mathbf{E}}(d_{mn}^{sRMS})$ for SAR images, and as $\alpha = 80\% \times \max_{e_{mn}\in\mathbf{E}}(d_{mn}^{sRMS})$ for real SAR images. That is, the region merging algorithm will merge nearly 90% off the pairs of oversegmented regions for SAR images and 80% of the pairs for real SAR images. It is worth noticing that it is better to set α a little too small rather than too large because if α is too large, there will be too many regions being merged, which will result in irreversible mistakes. If α is a little too small, the further merging classification algorithm can reduce the large number of regions to L, although with more computation cost.

One obvious advantage of the region merging algorithm is that it can take as much information as possible to determine the similarity between oversegmented regions. The region merging algorithm will globally merge the two regions that are not only similar in brightness and texture, but also have no distinct boundary. What is more, the spatial information has also been employed by an underlying assumption that the closer two regions are, the more probable that they will be in the same segment. This can reduce the ambiguity from the appearance of regions and guarantee that the regions inside the true segments can be merged with priority. In addition, as the sizes of the regions increase, the regions will contain complete patterns of objects, which can improve the accuracy of features to describe the objects and aid in the further classification of FCISIO. This process of the region merging algorithm can be illustrated by an example shown in Fig. 5, where Fig. 5(a) is an SAR image, Fig. 5(b) shows the oversegmented regions after the preprocessing step, and Figs. 5(c)–5(h) are the intermediate results during the region merging algorithm. It can be clearly noticed that the oversegmented regions inside the true segments are first merged, whereas the oversegmented regions along the boundary are not merged.

Table 1 Flowchart of the region merging algorithm.

Step 1. A greed search is performed to find RAG: $\mathbf{G} = (\mathbf{S}, \mathbf{E})$;

Step 2. Compute the similarity criterion by Eq. (7), and obtain the merging sequence;

Step 3. Find the most similar pair of adjacent regions. If the merging condition Eq. (8) is satisfied, then merge them and turn to **Step 4**; Otherwise stop the merging process and output the results.

Step 4. Update **G** and the similarity criterion, and turn to **Step 3**.

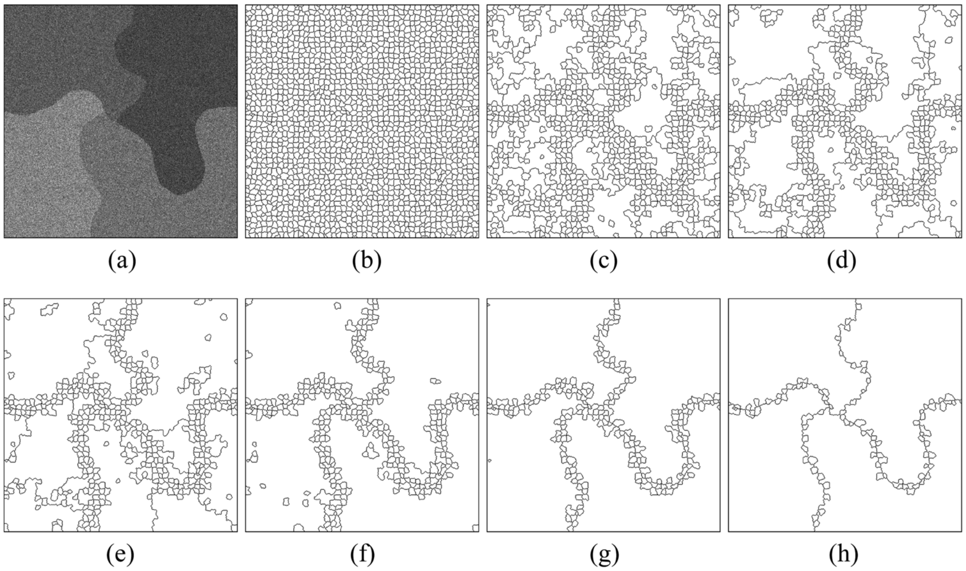

Fig. 5 A SAR image and the intermediate segmentation results of the region merging stage: (a) the SAR image; (b) the oversegmented regions obtained by the preprocessing step; (c)–(h) the intermediate segmentation results during the region merging stage.

3.2 *Region Classification Stage*

After the region merging stage, the number of oversegmented regions is still larger than the number L of true segments. The extra regions come from two sources: (1) the disconnected regions of the same class and (2) the ambiguous regions that are difficultly merged in the region merging stage, e.g., some small regions along the boundary between true segments. In this section, we design a method called fuzzy clustering based on an independent subspace iterative optimization to incorporate multiple heterogeneous features and further classify the remaining oversegmented regions into L.

The purpose of clustering is to group similar data points in the same cluster and dissimilar data points in different clusters, which is decided by a predefined objective function. Therefore, the key point in the fuzzy clustering algorithm is the fuzzy objective function. The most studied fuzzy c-means (FCM) algorithm, which was proposed by Dunn[50] and later extended by Bezdek,[51] computes the weighted sum of the squared intraclass distance as the objective function. The analytical iterative optimization equation of the objective function can be conveniently obtained by adopting the Euclidean distance as the similarity measure. With its simple implementation and high efficiency, FCM has been widely used, but its drawbacks are also obvious: the distance measure adopted in FCM is not robust and accurate enough. As we know, the Euclidean distance can effectively characterize the cluster of a spherical shape or hyperspherical shape, but is ineffective for the data points with nonspherical and complex shape distributions. If the separation boundaries between clusters are nonlinear, FCM will not work satisfactorily. To deal with this problem, Chen and Zhang[52] mapped the original data points into a higher dimensional Hilbert space using a kernel function, which could induce a class of non-Euclidean distance measures. This method can improve the algorithm's robustness to noise. But in this method, all dimensions of data are equally treated whether in the original space or in the transferred space, which is not consistent with the practical situation because input data is usually made up of values observed from different sources. Based on this consideration, Chen et al.[53] later extended their single-kernel method to a multiple-kernel one where different kernels were defined for different information channels. As demonstrated in Ref. 53, the multiple-kernel fuzzy clustering algorithm can obtain better segmentation results than the single-kernel method, especially for the texture images. However, it can also be noticed that the results of the kernel-based methods are very sensitive to the selection of the kernel function and its related parameters, which were set by the "trial-and-error" method and is still an "open problem."[52] This problem also limits the use of complex and high-dimension features. The essence of the multiple-kernel method[53] is to concatenate all information channels together forming a complete high-dimension feature space and then to change the proportions of different information in measuring the distance between objects. This strategy is similar to the variable weighting clustering algorithm[54] where a weight is assigned to measure the importance of the corresponding dimension and finally decides its proportion in the distance measure. However, we argue that this strategy is not very reasonable and tractable. Each kind of information (or feature) characterizes the objects from its own specific aspect. In each subspace, the corresponding feature can effectively describe the related property of objects. For example, the brightness can accurately separate the objects with different brightnesses [such as the river area in Fig. 2(a)], and the texture can accurately distinguish the objects with different texture properties [such as the forest area in Fig. 2(b)]. But because the input data points have different distribution structures in the different subspaces, if we concatenate all kinds of information together forming a complete high-dimension space, they will affect each other and become indistinguishable. Therefore, as all the features are concatenated together, the discriminability in each specific aspect is actually weakened. From another point of view, different kinds of features require different methods to compute the corresponding distance (similarity) just as mentioned in Sec. 2.3. It is not reasonable to concatenate all features together and compute the final distance by the weighted sum of the element-to-element differences in all feature dimensions. However, the similarity measure adopted by the existed fuzzy clustering algorithm cannot solve this problem.

To cope with the drawbacks analyzed above, we construct the following fuzzy objective function to be minimized:

$$J^s(\mathbf{U},\mathbf{V}^s)=\sum_{m=1}^{N_S'}\sum_{n=1}^{L}|s_m|u_{mn}^2(d_{mn}^{sB})^2(d_{mn}^{sT})^2,\quad s.t.\sum_{n=1}^{L}u_{mn}=1, \tag{9}$$

where $u_{mn}\in\mathbf{U}$ is the degree of membership of the m'th data point in the n'th cluster $\mathbf{v}_n^s$, and $|\bullet|$ computes the cardinality of the argument. Equation (9) first independently computes the brightness-based similarity d_{mn}^{sB} and the texture-based similarity d_{mn}^{sT} between oversegmented region s_m and the n'th cluster $\mathbf{v}_n^s$, $m=1,2,\cdots,N_S'$, $n=1,2,\cdots,L$. N_S' is the number of the remaining oversegmented regions after the region merging stage. Then the product operator is performed to integrate the two kinds of similarities together as the final distance. It can be noticed that the different kinds of features will not affect each other which will preserve the distribution structures of the data points in each subspace. Only when the data point is close to the center simultaneously in both the brightness-based subspace and the texture-based subspace, it will obtain a short final distance to the center. This strategy can enhance the discriminability of the similarity measure between data points.

Different from the simple data clustering problem, the data points here represent the corresponding regions of different sizes. Since a large region is derived from the merging of many small regions, it is unreasonable to equally treat the large regions and the small regions. In addition, the feature vectors of small regions are more likely to be influenced by speckle noise than those of large regions. Therefore, the sizes of regions are introduced to let the large regions have more weight in Eq. (9).

Similar to that in the conventional FCM, we apply the alternating minimization (AM) approach to obtain the iterative optimization equation of Eq. (9). AM optimizes the objective function over one group of variables at each subiteration, while keeping all the other variables fixed, which provides a useful framework for the derivation of an iterative optimization algorithm. According to Eq. (9), we divide the features into two groups: the brightness feature and the texture feature. The objective function $J^s(\mathbf{U},v^{sB},\mathbf{v}^{sT})$ will be alternatively minimized in the brightness-based subspace and the texture-based subspace, respectively.

Introducing the Lagrange term of the constraint u_{mn} $(m=1,2,\cdots,N_S', n=1,2,\cdots,L)$ into the objective function Eq. (9), we have

$$Q^s(\mathbf{U},v^{sB},\mathbf{v}^{sT})=\sum_{m=1}^{N_S'}\sum_{n=1}^{L}|s_m|u_{mn}^2(d_{mn}^{sB})^2(d_{mn}^{sT})^2+\lambda\left(\sum_{n=1}^{L}u_{mn}-1\right). \tag{10}$$

If $(u_{mn}, v_n^{sB}, \mathbf{v}_n^{sT})$ is to minimize Q^s, its gradient in all sets of variables must vanish. Thus,

$$\frac{\partial Q}{\partial \lambda} = 0, \tag{11}$$

$$\frac{\partial Q}{\partial u_{mn}} = 0, \tag{12}$$

$$\frac{\partial Q}{\partial v_n^{sB}} = 0, \tag{13}$$

$$\frac{\partial Q}{\partial \mathbf{v}_n^{sT}} = 0. \tag{14}$$

From Eqs. (11) and (12), we obtain the updating rule of the weights u_{mn}:

$$u_{mn} = \left[\frac{(d_{mn}^{sB})^2 (d_{mn}^{sT})^2}{\sum_{l=1}^{L} (d_{ml}^{sB})^2 (d_{ml}^{sT})^2} \right]^{-1}. \tag{15}$$

Substituting $(d_{mn}^{sB})^2 = (f_m^{sB} - v_n^{sB})^2$ into Eq. (13), we obtain the updated cluster center in the brightness-based subspace:

$$v_n^{sB} = \frac{\sum_{m=1}^{N_S'} u_{mn}^2 |s_m| (d_{mn}^{sT})^2 f_n^{sB}}{\sum_{m=1}^{N_S'} |s_m| u_{mn}^2 (d_{mn}^{sT})^2}. \tag{16}$$

When obtaining the updated cluster center in the texture-based subspace, we let $(d_{mn}^{sT})^2 = \beta_{mn} \| \mathbf{f}_m^{sT} - \mathbf{v}_n^{sT} \|_2^2$, where $\| \bullet \|_2$ stands for the 2-norm distance of the arguments. Then from Eq. (14), we obtain

$$\mathbf{v}_n^{sT} = \frac{\sum_{m=1}^{N_S'} u_{mn}^2 |s_m| \beta_{mn} (d_{mn}^{sB})^2 \mathbf{f}_n^{sT}}{\sum_{m=1}^{N_S'} |s_m| u_{mn}^2 \beta_{mn} (d_{mn}^{sB})^2}, \tag{17}$$

where the parameter β_{mn} can be approximately computed by the ratio of the texture-based similarity and the square of the 2-norm distance between the data point and the cluster center at the last iteration. As the algorithm approaches the local minimum, the differences between the approximate value and the true value about β_{mn} will decrease to zero.

Iteratively performing Eqs. (15)–(17) will lead to the local minimum of J^s. The objective function will be iteratively optimized along the direction of the gradient steepest descent in each independent subspace, which can at the same time allow for a high convergence speed and preserve the distribution structures of the data points in each subspace. The flowchart of the region merging algorithm is shown in Table 2.

Table 2 Flowchart of fuzzy clustering based on independent subspace iterative optimization (FCISIO).

Step 1. Set the number L of the true segments, and the stopping condition K;

Step 2. Initialize randomly the fuzzy partition matrix $[u_{mn}]$;

Step 3. Set the loop counter $k = 0$;

Step 4. Compute the cluster centers using Eqs. (16) and (17).

Step 5. Calculate the fuzzy partition matrix using Eq. (15).

Step 6. If $k > K$ then stop and output the results; otherwise set $k = k + 1$, and go to **Step 4**.

4 Experiments

In this section, synthetic and real SAR images are used to test the proposed FCISIO and four other comparing algorithms, which are

1. Classical FCM algorithm;
2. SC algorithm[20] which can automatically obtain a tight structure of clusters by mapping initial data points into a new space;
3. Multilevel logistic (C-MLL) model based segmentation algorithm,[15] which is a classical MRF model. C-MLL is based on RAG and designs a spatial context model to penalize the boundary between different segmented regions.
4. Context-based hierarchical unequally merging for SAR image segmentation (CHUMSIS),[35] which models the context of oversegmented regions by three rules inspired from the Gestalt laws and then designs two different strategies to combine the rules for the region merging. The rules in CHUMSIS are different in essence from the fuzzy operators adopted in this paper. Since CHUMSIS only relies on spatial region merging, the disconnected region of the same class will be considered as two segments by CHUMSIS. Therefore, for convenience of illustration, the final segmentation of CHUMSIS will be presented by the boundaries between segments.

For the sake of fairness, all five algorithms begin with the same initial oversegmented regions. Although FCM and SC can directly operate on pixels, the adoption of oversegmented regions can greatly improve the accuracy of feature description and the algorithms' computation speeds. In addition, the oversegmented regions[42] used in this paper are more coherent inside and can preserve image edges better than the oversegmented regions produced by the watershed segmentation in Ref. 15, which will not decrease the performance of C-MLL. The brightness feature and the texture feature extracted in Sec. 2.2 are concatenated together as the input of FCM and SC, and the variables in each dimension have been linearly transformed into [0,1] just as in Ref. 20. C-MLL adopts the mean brightness of the oversegmented regions in the feature model. The related parameters of the algorithms are set as follows. Since SC is highly sensitive to the scaling parameter σ, SC is performed with each parameter in the interval $(0,1]$ with a step length of 0.05, and the best segmentation result is selected as the final result for the corresponding input SAR image. It is found by experiments that this interval is sufficient because the segmentation results of SC have been stable when σ is increasing to 1. The parameters of CHUMSIS are set according to Ref. 35.

Considering the influence of random initialization, we independently run each algorithm for the same input SAR image 20 times and select the best result as the final segmentation result. For the SAR images, the percentages of correctly labeled pixels are computed as the scores to compare the five methods. For the real SAR images, the performance of the methods is visually assessed. All the experiments are conducted on a machine with Pentium® 4 and with 3.2 GHz and 1 GB memory, and the operation system of Window XP SP3. The algorithms are implemented in MATLAB 7.9.0(2009b).

4.1 *Synthetic Aperture Radar Images Without Texture*

In this section, we first use four groups of SAR images, as shown in Fig. 6, to quantitatively test the five methods. The original SAR images are shown in the first column, where Fig. 6(a) is a three-class image without texture, Fig. 6(b) is a four-class image without texture, Fig. 6(c) is a three-class image with texture, and Fig. 6(d) is a four-class image with texture. The textures are selected from the Brodatz Texture database. All the SAR images are 512×512 pixels in size. Because the speckle noise can be modeled as a multiplicative Nakagami distribution, each group of SAR images is created by adding multiplicative Nakagami distribution noise of different looks to the original synthetic image. In each row, the 2-look, 5-look, and 10-look synthetic images are shown on the right of the corresponding original synthetic image. In Fig. 6(a), the gray levels of three classes are [96, 144, 160]. In Fig. 6(b), the gray levels of four classes are [128, 144, 160, 176]. Note that the gray levels in the original images are much closer, and the speckle noise causes significant overlapping of the intensity ranges among different regions, especially for

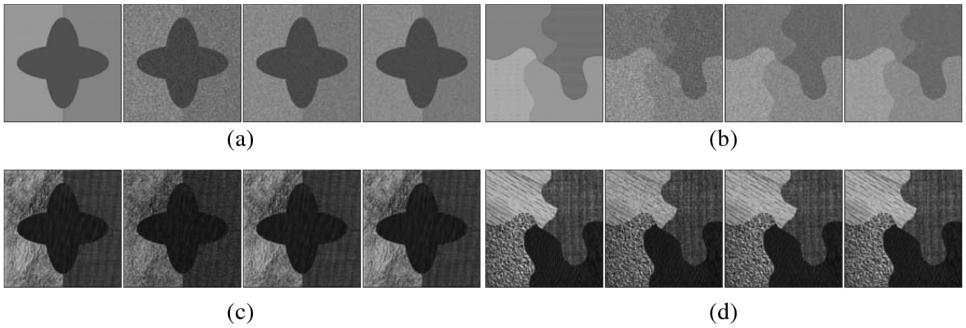

Fig. 6 Four groups of SAR images. The original synthetic images are shown on the first column, and the corresponding 2-look, 5-look and 10-look SAR images are illustrated on their right, respectively. (a) Three-class SAR images without texture, the gray levels of which are [96, 144, 160]. (b) Four-class SAR images without texture, the gray levels of which are [128, 144, 160, 176]. (c) Three-class SAR images with texture. (d) Four-class SAR images with texture.

those two-look SAR images. In Fig. 6(c), it can be seen that the left part and the right part have different texture characteristics and similar mean brightnesses in local regions. The brightness in the left part gradually increases from top to bottom. The middle part is used to roughly simulate the ocean scenes in real SAR images, which often have regular waves. In Fig. 6(d), the four classes have different texture properties and obvious different mean brightnesses in local regions. They are used to simulate different terrain surfaces, such as mountain area (the class on the top left), farm area (the class on the top right), forest area (the class on the bottom left), and river area (the class on the bottom right).

The segmentation results are shown in Fig. 7, where Fig. 7(a) is the three-class images without texture, Fig. 7(b) is the four-class images without texture, Fig. 7(c) is the three-class images with texture, and Fig. 7(d) is the four-class images with texture. In each subfigure, the number of look increases from top to bottom, and the results from left to right are by FCM, SC, C-MLL, CHUMSIS, the proposed region merging algorithm, and FCISIO, respectively. It can be noticed from the first two columns in Figs. 7(a) and 7(b) that FCM and SC cannot produce satisfactory segmentation results for the images without texture, especially for those with more classes and heavy noise. On the contrary, FCM and SC perform quite well on the images with texture, as shown in the first two columns in Figs. 7(c) and 7(d). That is mainly because of the low discriminability of the concatenated brightness and texture features. On one side, when the brightness feature is concatenated with the texture feature, the discriminability of the brightness feature is weakened by the texture feature, which results in the unsatisfactory results by FCM and SC in Figs. 7(a) and 7(b). On the other side, since the dimension of the texture feature is higher than that of the brightness feature, the texture feature becomes dominant for classification, which explains that FCM and SC can obtain better results on the images with texture in Figs. 7(c) and 7(d). Relying on the transformation of the distribution structure of data points, SC obtains improved results but still produces obvious regional errors. Therefore, it can be concluded that it is not appropriate for the strategy to concatenate features and process them as a whole in the complete high-dimension space.

C-MLL produces results with obviously better region consistency than FCM and SC, as shown in the third column in Figs. 7(a)–7(d), especially for the images without texture. That is mainly because of the context model adopted by C-MLL. However, due to the absence of texture information, C-MLL is not successful at segmenting the images with texture, especially for the four-class ones. CHUMSIS obtains better results than FCM, SC, and C-MLL, which is due to the utilization of multiple features. But since it is only operated in the image domain, CHUMSIS will produce extra segments. When the images are complex such as the texture images with heavy noise (e.g., two-look image with texture), the performance of CHUMSIS will fall down.

It can be noticed in the fifth column in Fig. 7 that the proposed region merging algorithm has merged most of the oversegmented regions inside the true segments. The successful segmentations have been given by FCISIO in the last column in Fig. 7. The results by FCISIO are clear of

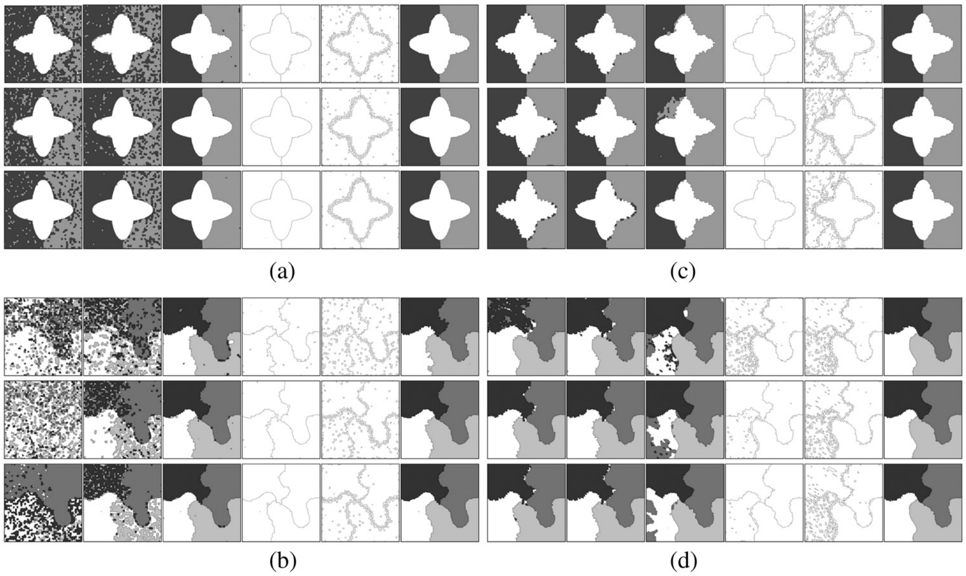

Fig. 7 The experiment results on the SAR images: (a) the three-class images without texture; (b) the four-class images without texture; (c) the three-class image with texture; and (d) the four-class image with texture. In each subfigure, the number of look increases from top to bottom, and the results from left to right are by fuzzy c-means (FCM), spectral clustering (SC), multilevel logistic (C-MLL), context-based hierarchical unequally merging for SAR image segmentation (CHUMSIS), the proposed region merging algorithm and fuzzy clustering based on independent subspace iterative optimization (FCISIO), respectively.

any regional errors with only minor errors along the boundary. This implies that the strategy to integrate brightness features, texture features, and edge features is effective and helpful for the SAR images. Note that there are some teeth-like errors along the boundary, which is due to the initial oversegmented regions. When the brightnesses of the two adjacent classes are close and the noise is heavy, the boundary between the two classes will be unclear. As a result, the initial oversegmented regions cannot accurately capture the boundary between true segments and will produce such teeth-like errors. Against such errors, FCISIO can deal with the oversegmented regions along the boundary better than the compared algorithms, especially for the images with more classes and heavier speckle noise.

Table 3 summarizes the accuracies of the five algorithms for SAR images, among which the best result for the same input image is in bold. Note that as the noise is reduced, the performance of FCM and SC will increase. C-MLL can give good segmentation results for most of those images, but is not stable, especially when the number of classes is large and the speckle noise is heavy. CHUMSIS performs very well on the images without texture, but is not suitable for the images with texture. By contrast, FCISIO is quite stable and obtains the best results.

4.2 *Real SAR Images*

In this section, we use two real Ku-Band SAR images from Sandia National Laboratories to further test the five methods. The images are processed in real-time by the radar carried by the Sandia Twin Otter aircraft. The results of the five algorithms and the intermediate segmentation results of FCISIO are shown, respectively.

Figure 8(a) shows the first real SAR image that captures the area of China Lake Airport in California with a 3-m resolution and 522 × 446 pixel size. There are three classes in this SAR image: runway (dark), buildings (bright), and farms (gray). Note from Fig. 8(a) that the small buildings are randomly distributed in the farms, and some areas of farm are quite similar to the buildings or the runway by subjective observation, which increases the difficulty in segmenting this image. The SAR image is first oversegmented into 1600 regions, as shown in Fig. 8(b). Figures 8(c)–8(f) illustrate the segmentation results of FCM, SC, C-MLL, and CHUMSIS, respectively. Figures 8(g) and 8(h) show the intermediate segmentation after the region merging

Table 3 Summary of segmentation results for synthetic aperture radar (SAR) images.

		FCM	SC	C-MLL	CHUMSIS	FCISIO
SAR images without texture						
Three-class	2-Look	0.78264	0.82857	0.98560	0.97456	**0.98967**
	5-Look	0.82132	0.86392	0.99304	0.99246	**0.99329**
	10-Look	0.87952	0.89481	0.99349	0.99402	**0.99423**
Four-class	2-Look	0.36703	0.56526	0.96154	0.95013	**0.97471**
	5-Look	0.26171	0.73410	0.98032	0.98056	**0.98357**
	10-Look	0.45784	0.78416	0.98357	0.98312	**0.98609**
SAR images with texture						
Three-class	2-Look	0.97890	0.97953	0.97079	0.98117	**0.98565**
	5-Look	0.97616	0.97974	0.93331	0.97668	**0.98648**
	10-Look	0.97306	0.97872	0.97531	0.97564	**0.98621**
Four-class	2-Look	0.92422	0.97062	0.91089	0.88042	**0.98652**
	5-Look	0.97716	0.97870	0.88743	0.95775	**0.98769**
	10-Look	0.97482	0.97498	0.92056	0.97127	**0.98714**

Note: FCM, fuzzy c means algorithm; SC, spectral clustering algorithm; C-MLL, multilevel logistic model; CHUMSIS, context-based hierarchical unequally merging algorithm for SAR image segmentation.

stage in the proposed method and the final segmentation result by FCISIO. Due to the inappropriate similarity measure, FCM misclassifies the runway area and the building area. Compared with FCM, SC can improve the classification accuracy a bit, but a runway track has been misclassified as a building area and the boundaries between objects are still not precise. Although C-MLL can accurately segment out most of the runway due to the obvious different brightness, as shown in Fig. 8(e), it cannot separate the building area from some farm area on the right part of

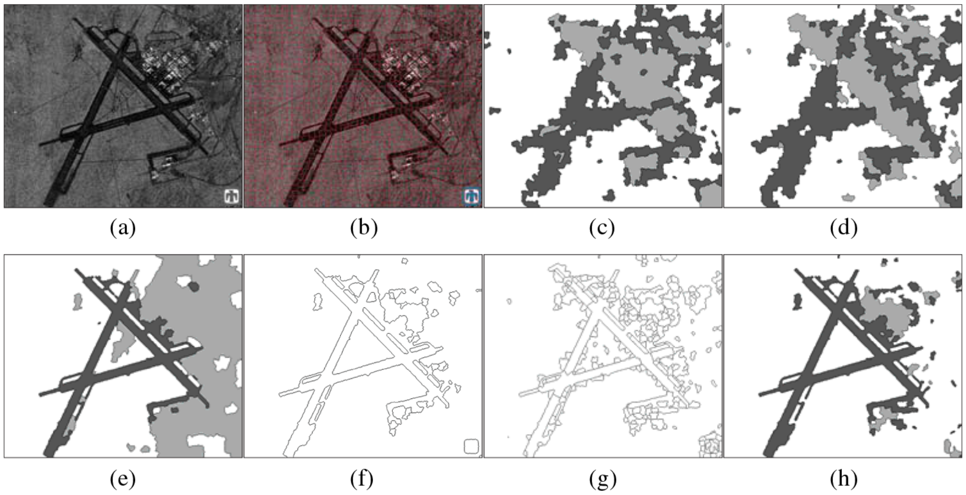

Fig. 8 Segmentation results of the first real SAR image. (a) A Ku-SAR image of the area of China Lake Airport in California with a 3-m resolution and a 522×446 pixels size. (b) The oversegmented regions obtained in the preprocessing step overlapped the original image. (c), (d), (e), (f) and (h) The segmentation results by FCM, SC, C-MLL, CHUMSIS and FCISIO, respectively. (g) The intermediate segmentation after the region merging stage.

the image. That is because such a farm area has large intraclass variations, which make the building area and the farm area difficult to be separated by only the brightness feature. CHUMSIS can obtain a much better segmentation result, as shown in Fig. 8(f), which supports our previous conclusion that edge features and a region merging mechanism can effectively improve the accuracy of segmentation results for the images with large intraclass variations. In Fig. 8(h), the boundaries between different objects have been accurately located, and the small buildings have also been exactly segmented out, which indicates that the proposed FCISIO can deal with both the large intraclass variations and the small objects very well. Figure 8(g) validates the effectiveness of a region merging algorithm for the real SAR image, where the oversegmented regions within the true segments have been merged.

Figure 9(a) illustrates the second real SAR image that captures the area of the Rio Grande River near Albuquerque, New Mexico, with a 1-m resolution and a 600 × 432 pixels size. This SAR image has three classes: river (dark), farm (with small-scale texture), and forest (with large-scale texture). This image is much more complex than Fig. 8(a): the brightness of the farm varies greatly, and some farm areas are similar to the forest in brightness; the forest has a large-scale texture that results in large shadow areas similar to the river; the small objects like trees are randomly distributed in the farms. So, large intraclass variations, little interclass difference, and the existence of multiscale objects make this image particularly difficult for segmentation. The initial number of oversegmented regions is equal to 1900, as shown in Fig. 9(b). Figures 9(c)–9(f) illustrate the segmentation results of FCM, SC, C-MLL, and CHUMSIS, respectively. Figures 9(g) and 9(h) show the results after the region merging stage and FCISIO. It can be noticed that FCM cannot effectively separate the three classes and misclassifies the river area and the farm area. C-MLL and SC can segment out primary target areas of the three classes and make a good segmentation for the regions with obvious different brightnesses, such as the river. However, because of the similar brightness, the results by C-MLL and SC will lose some details, such as trees and shadows. By visual assessment, CHUMSIS and FCISIO obtain much better results, which is mainly because of the adoption of the texture feature. FCISIO can preserve more details and make a better balance between the details and region consistency than SC. Figure 9(g) clearly illustrates that the oversegmented regions without ambiguity have been merged during the region merging stage.

4.3 *Complexity Analysis*

Since the SAR images have the same size, we summarize the average running time of the five comparing algorithms for all SAR images in Table 4. The running time for C-MLL is for 30

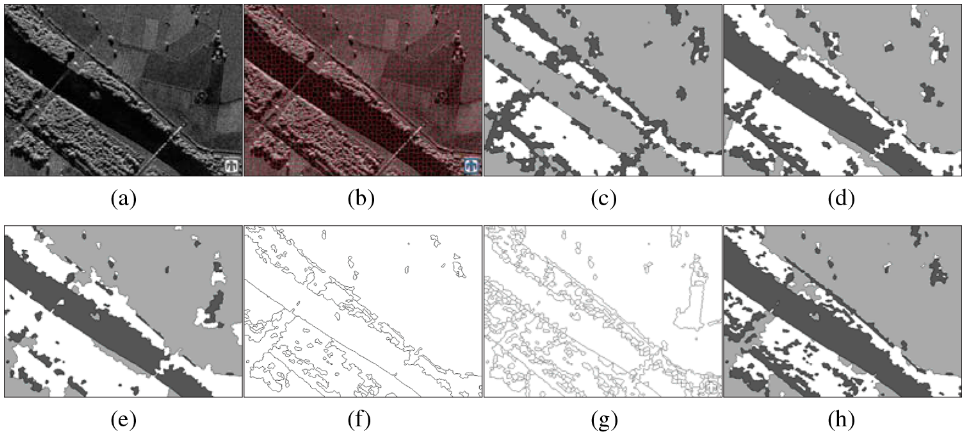

Fig. 9 Segmentation results of the second real SAR image. (a) A Ku-SAR image in the area of Rio Grande River near Albuquerque, New Mexico, with a 1-m resolution and a 600 × 432 pixels size. (b) The oversegmented regions obtained in the preprocessing step overlapped the original image. (c), (d), (e), (f) and (h) The segmentation results by FCM, SC, C-MLL, CHUMSIS, and FCISIO, respectively. (g) The intermediate segmentation after the region merging stage.

Table 4 Summary of the average running time for the five algorithms.

	FCM	SC	C-MLL	CHUMSIS	FCISIO
Preprocessing time (s)	43.09				
Segmenting time (s)	28.95	55.51	4249.63	69.77	89.23
Total running time (s)	72.04	98.60	4292.72	112.86	132.32

100 iterations for FCM and SC, 30 iterations for C-MLL, and 100 iterations for FCISIO.

iterations and that for FCM and FCISIO is for 100 iterations. Note that FCM spends the shortest time due to the simple scheme. C-MLL costs the most time, because the oversegmented regions' labels and the model's parameters have to be updated in each iteration. On the contrary, the region merging algorithm is deterministic and does not need such an iterative optimization process, which saves a lot of time. It is worth noticing that although FCISIO is more complicated than FCM, the running time of FCISIO is less than that of FCM, because the proposed region merging stage has greatly reduced the computation burden for FCISIO. Considering the high accuracy obtained by FCISIO, it can be concluded that the proposed method can produce a good balance between the computation speed and the segmentation accuracy.

Considering the influence of implementation, the computational complexity for FCISIO is next briefly analyzed. The preprocessing method need $O(N_I)$[42] to obtain the oversegmented regions, where N_I denotes the pixel number of the input image. It costs $O[N_I(N_\omega N_\theta + N_\lambda + 1)]$ to extract the features and $O(N_\omega N_\theta N_s^2)$ to compute the similarity between oversegmented regions. Because $(N_\omega N_\theta + N_\lambda + 1)$ is a constant, the running time to extract features is linear with the pixel number of the input image. Since the region merging stage is based on a RAG model, it will at most cost $O[N_e' \log(N_e')]$[40] to merge two oversegmented regions, and $O[M(N_\omega N_\theta + N_\lambda + 1)]$ to make an update, where N_e' represents the number of the edges in the RAG at a certain iteration and M (usually <5) is the number of adjacent regions to the merged regions. Therefore, the computation complexity of the region merging algorithm is $O[N_e' \log(N_e')]$ for one iteration cycle. The computational complexity of FCISIO is $O(N_S' LK)$, where N_S' is the number of the remaining oversegmented regions after the region merging stage, L is the number of the final segments, and K is the number of iterations.

5 Conclusion

The task of reducing the number of pixels to the number of true segments which is used throughout this paper, consists of three steps: (1) Obtain oversegmented regions in the preprocessing step; (2) merge oversegmented regions inside true segments by the region merging algorithm where multiple information is employed; (3) classify the remaining oversegmented regions into final segments by FCISIO. From step (1) to step (3), the number of the involved operation elements decreases, while the computational cost on every operation element increases. The strategy of this framework is to spend a little time in segmenting the elements without ambiguity such as the pixels or oversegmented regions inside true segments, while spending much more time in partitioning the elements with ambiguity such as those along the boundary between true segments. This process is plausibly similar to that of human beings, in which people always pay more attention to details such as the boundary between objects than the components inside objects. Therefore, this two-stage method can improve the segmentation accuracy and reduce the running time at the same time.

It is worth emphasizing that we investigate SAR image segmentation entirely from the feature extraction to the algorithm design, where we extract features of different properties for SAR images, design three similarity measures according to different kinds of features, and finally integrate the different similarities in the proposed algorithm. FCISIO realizes an effective way to integrate features with different properties: on one side, the objective function with a multifeatures ensemble is designed to avoid the influence between different kinds of features and preserve the distribution structures in each subspace. On the other side, the optimization process will iteratively search in each subspace of the corresponding feature along the direction of the gradient steepest descent so as to accelerate the convergence speed of the algorithm. The comparative experiments on the synthetic and real SAR images have proven that the proposed method can achieve the best performance with a very short running time and is stable and effectiveness for various kinds of SAR images.

Acknowledgments

All of the authors would like to thank the reviewers for their useful comments and suggestions on our manuscript. This work was supported by the National Natural Science Foundation of China under Grant Nos. 61172138, 61203202, and 61501352, China Postdoctoral Science Foundation under Grant No. 2014M562376, and Fundamental Research Funds for the Central Universities under Grant No. JB141304.

References

1. Y. Li, J. Li, and M. A. Chapman, "Segmentation of SAR intensity imagery with a voronoi tessellation, Bayesian inference, and reversible jump MCMC algorithm," *IEEE Trans. Geosci. Remote Sens.* **48**(4), 1872–1881 (2010).
2. P. Zhang et al., "Unsupervised multi-class segmentation of SAR images using fuzzy triplet Markov fields model," *Pattern Recognit.* **45**(11), 4018–4033 (2012).
3. Z. Y. Bai, Z. F. Liu, and P. K. He, "SAR image target segmentation based on entropy maximization and morphology," *J. Syst. Eng. Electron.* **15**(4), 484–487 (2004).
4. D. M. Smith, "Speckle reduction and segmentation of synthetic aperture radar images," *Int. J. Remote Sens.* **17**(11), 2043–2058 (1996).
5. L. J. Du, "Texture segmentation of SAR images using localized spatial filtering," in *IEEE International Geoscience and Remote Sensing Symposium*, pp. 1983–1986 (1990).
6. Z. Liu, X. Fan, and F. Lv, "SAR image segmentation using contourlet and support vector machine," in *Fifth Int. Conf. on Natural Computation*, pp. 250–254 (2009).
7. H. B. Kekre, S. Gharge, and T. K. Sarode, "SAR image segmentation using vector quantization technique on entropy images," *Int. J. Comput. Sci. Inf. Secur.* **7**(3), 276–282 (2010).
8. U. Kandaswamy, D. A. Adjeroh, and M. C. Lee, "Efficient texture analysis of SAR imagery," *IEEE Trans. Geosci. Remote Sens.* **43**(9), 2075–2083 (2005).
9. D. A. Clausi, "Comparison and fusion of co-occurrence, Gabor and MRF texture features for classification of SAR sea-ice imagery," *Atmos. Ocean* **39**(3), 183–194 (2001).
10. J. A. Karvonen and M. Hallikainen, "Sea ice SAR classification based on edge features," in *IEEE Int. Geoscience and Remote Sensing Symposium*, pp. 129–132 (2009).
11. R. Touzi, A. Lopes, and P. Bousquet, "A statistical and geometrical edge detector for SAR images," *IEEE Trans. Geosci. Remote Sens.* **26**(6), 764–773 (1988).
12. O. Germain and P. Réfrégier, "Edge location in SAR images: Performance of the likelihood ratio filter and accuracy improvement with an active contour approach," *IEEE Trans. Image Process.* **10**(1), 72–78 (2001).
13. R. Fjørtoft et al., "An optimum multiedge detector for SAR image segmentation," *IEEE Trans. Geosci. Remote Sens.* **36**(3), 793–802 (1998).
14. H. Deng and D. A. Clausi, "Unsupervised segmentation of synthetic aperture radar sea ice imagery using a novel Markov random field model," *IEEE Trans. Geosci. Remote Sens.* **43**(3), 528–538 (2005).
15. Q. Yu and D. A. Clausi, "IRGS: image segmentation using edge penalties and region growing," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(12), 2126–2139 (2008).
16. D. M. Zhang, M. S. Fu, and B. Luo, "SAR image segmentation using kernel density estimation on region adjacency graph," in *2nd Asia-Pacific Conf. on Synthetic Aperture Radar*, pp. 668–671 (2009).
17. J. Xue, Y. Zhang, and X. Lin, "Rayleigh-distribution based minimum error thresholding for SAR images," *J. Electron.* **16**(4), 336–342 (1999).
18. A. E. Zaart et al., "Segmentation of SAR images," *Pattern Recognit.* **35**(3), 713–724 (2002).

19. C. M. Han et al., “A method to segment SAR images based on histogram,” in *IEEE Int. Geoscience and Remote Sensing Symposium*, pp. 3694–3696 (2005).
20. X. R. Zhang et al., “Spectral clustering ensemble applied to SAR image segmentation,” *IEEE Trans. Geosci. Remote Sens.* **46**(7), 2126–2136 (2008).
21. H. Q. Liu, F. Zhao, and L. C. Jiao, “Fuzzy spectral clustering with robust spatial information for image segmentation,” *Appl. Soft Comput.* **12**(11), 3636–3647 (2012).
22. R. Fjørtoft et al., “Unsupervised classification of radar images using hidden Markov chains and hidden Markov random fields,” *IEEE Trans. Geosci. Remote Sens.* **41**(3), 675–686 (2003).
23. S. Derrode and W. Pieczynski, “SAR image segmentation using generalized pairwise Markov chains,” *Proc. SPIE* **4885**, 80–91 (2002).
24. G. S. Xia, C. He, and H. Sun, “Integration of synthetic aperture radar image segmentation method using Markov random field on region adjacency graph,” *IET Radar Sonar Navig.* **1**(5), 348–353 (2007).
25. Y. F. Cao, H. Sun, and X. Xu, “An unsupervised segmentation method based on MPM for SAR images,” *IEEE Geosci. Remote Sens. Lett.* **2**(1), 55–58 (2005).
26. J. A. Karvonen, “Baltic sea ice SAR segmentation and classification using modified pulse-coupled neural networks,” *IEEE Trans. Geosci. Remote Sens.* **42**(7), 1566–1574 (2004).
27. R. C. Liu et al., “A multiobjective immune clustering ensemble technique applied to unsupervised SAR image segmentation,” in *2010 Int. Conf. on Image and Video Retrieval*, pp. 158–165 (2010).
28. J. J. Quan, X. B. Wen, and X. Q. Xu, “Multiscale probabilistic neural network method for SAR image segmentation,” *Appl. Math. Comput.* **205**(2), 578–583 (2008).
29. D. D. Yang et al., “Artificial immune multi-objective SAR image segmentation with fused complementary features,” *Inf. Sci.* **181**(13), 2797–2812 (2011).
30. D. D. Yang et al., “An efficient automatic SAR image segmentation framework in AIS using kernel clustering index and histogram statistics,” *Appl. Soft Comput.* **16**, 63–79 (2014).
31. P. Han et al., “An iterative segmentation algorithm of SAR image based on support vector machine,” in *2nd Asia-Pacific Conf. on Synthetic Aperture Radar*, pp. 676–679 (2009).
32. E. A. Carvalho et al., “SAR imagery segmentation by statistical region growing and hierarchical merging,” *Digital Signal Process.* **20**(5), 1365–1378 (2010).
33. W. Li et al., “Watershed-based hierarchical SAR image segmentation,” *Int. J. Remote Sens.* **20**(17), 3377–3390 (1999).
34. F. Galland, N. Bertaux, and P. Réfrégier, “Minimum description length synthetic aperture radar image segmentation,” *IEEE Trans. Image Process.* **12**(9), 995–1006 (2003).
35. H. Yu et al., “Context-based hierarchical unequal merging for SAR image segmentation,” *IEEE Trans. Geosci. Remote Sens.* **51**(2), 995–1009 (2013).
36. B. Ogo, V. Haese-Coat, and J. Ronsin, “SAR image segmentation by mathematical morphology and texture analysis,” in *IEEE Int. Geoscience and Remote Sensing Symposium*, pp. 717–719 (1996).
37. G. Akbarizadeh, “A new statistical-based kurtosis wavelet energy feature for texture recognition of SAR images,” *IEEE Trans. Geosci. Remote Sens.* **50**(11), 4358–4368 (2012).
38. D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002).
39. J. B. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000).
40. P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vision* **59**(2), 167–181 (2004).
41. F. Meyer, “An overview of morphological segmentation,” *Int. J. Pattern Recognit. Artif. Intell.* **15**(7), 1089–1118 (2001).
42. A. Levinshtein et al., “TurboPixels: fast superpixels using geometric flows,” *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(12), 2290–2297 (2009).
43. D. A. Clausi and H. Deng, “Design-based texture feature fusion using Gabor filters and co-occurrence probabilities,” *IEEE Trans. Image Process.* **14**(7), 925–936 (2005).
44. T. Randen and J. H. Husoy, “Filtering for texture classification: a comparative study,” *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(4), 291–310 (1999).

45. A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognit.* **24**(12), 1167–1186 (1991).
46. D. A. Clausi and M. Jernigan, "Designing Gabor filters for optimal texture separability," *Pattern Recognit.* **33**(11), 1835–1849 (2000).
47. K. Kayabol and J. Zerubia, "Unsupervised amplitude and texture classification of SAR images with multinomial latent model," *IEEE Trans. Image Process.* **22**(2), 561–572 (2013).
48. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed., Prentice Hall, London (2002).
49. I. Bloch, "Information combination operators for data fusion: a comparative review with classification," *IEEE Trans. Syst., Man, Cybern. A* **26**(1), 52–67 (1996).
50. J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.* **3**(3), 32–57 (1973).
51. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, New York (1981).
52. S. Chen and D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," *IEEE Trans. Syst., Man, Cybern. B* **34**(4), 1907–1916 (2004).
53. L. Chen, C. L. P. Chen, and M. Z. Lu, "A multiple-kernel fuzzy c-means algorithm for image segmentation," *IEEE Trans. Syst., Man, Cybern. B* **41**(5), 1263–1274 (2011).
54. J. Z. Huang et al., "Automated variable weighting in k-means type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(5), 657–668 (2005).

Hang Yu received his BS degree in 2005 and his PhD in 2014 at the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China. Then he joined in the School of Aerospace Science and Technology at Xidian University. His current interests include synthetic aperture radar image analysis and understanding, pattern recognition, computer vision, and evolutionary computation.

Luping Xu received his PhD in signal and information processing from Xidian University in 1996. Since 2000, he has been a professor at Xidian University. His main research interests are target detection, spread spectrum, satellite and mobile communication, and SAR image processing.

Dongzhu Feng received his PhD in control science and engineering from Northwestern Polytechnical University in 2006. She then joined in the School of Electronic Engineering at Xidian University. Now she is an associate professor in the School of Aerospace Science and Technology at Xidian University. Her current interests include image analysis and understanding, machine learning, computer vision, and compressive sensing and its applications.

Xiaochuan He received his BS degree in 2006 at the School of Electronic Engineering of Xidian University. He joined the School of Electronic Engineering of Xidian University in 2006 and is now working at the School of Aerospace Science and Technology at Xidian University. His current interests include computer vision and image processing.