

SAR Image Segmentation Based on Fisher Vector Superpixel Generation and Label Revision

Ronghua Shang , Senior Member, IEEE, Jiansheng Chen , Jie Feng , Member, IEEE, Yangyang Li , Senior Member, IEEE, Licheng Jiao , Fellow, IEEE, and Rustam Stolkin , Member, IEEE

Abstract—This article addresses the problem of superpixel-bases segmentation of synthetic aperture radar (SAR) images. Most superpixel segmentation methods have difficulties in segmenting adjacent regions with similar gray values, due to only considering spatial and gray information. To solve this problem and improve segmentation accuracy, this article proposes an SAR image segmentation method based on Fisher vector superpixel generation and label revision. First, the Fisher vector is obtained by processing the Gaussian mixture function. By introducing the Fisher vector, a distance formula is constructed for superpixel segmentation. Therefore, the adjacent regions with similar gray values can be segmented effectively in the generated superpixel map. Second, the superpixels are clustered using the K-means algorithm to obtain the initial label map. Then, with extracted edge information as constraints, the pixel labels obtained by K-means are repaired pixel by pixel to get the middle label map, according to the number and gray value difference of labels. This overcomes the influence of noise generated by K-means. Finally, the middle label map is relabeled using the region growth algorithm to divide pixel blocks. Isolated pixel blocks surrounded by similar labels are corrected, based on the gray mean difference. The final label result has a better segmentation accuracy. Experiments on synthetic SAR images and real images demonstrate that the proposed algorithm achieves higher segmentation accuracy than six state-of-the-art clustering algorithms for SAR image segmentation.

Index Terms—Fisher vector, label revision, superpixel, synthetic aperture radar (SAR), unsupervised segmentation.

I. INTRODUCTION

RADAR-BASED machine vision is an important application of image processing [1], [2]. Machine intelligence and artificial intelligence are two main types of intelligence.

Manuscript received 8 March 2022; revised 14 June 2022 and 11 August 2022; accepted 24 October 2022. Date of publication 2 November 2022; date of current version 14 November 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62176200 and Grant 61871306, in part by the 111 Project, in part by the Natural Science Basic Research Program of Shaanxi under Grant 2022JC-45 and Grant 2022JQ-616, in part by the Open Research Projects of Zhejiang Lab under Grant 2021KG0AB03, in part by the National Key R&D Program of China, the Guangdong Provincial Key Laboratory under Grant 2020B121201001, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515110686. (Corresponding author: Ronghua Shang.)

Ronghua Shang, Jiansheng Chen, Jie Feng, Yangyang Li, and Licheng Jiao are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: rhshang@mail.xidian.edu.cn; jschen69@163.com; jiefeng0109@163.com; yyli@xidian.edu.cn; lchjiao@mail.xidian.edu.cn).

Rustam Stolkin is with the Extreme Robotics Lab, University of Birmingham, B15 2SQ Birmingham, U.K. (e-mail: r.stolkin@cs.bham.ac.uk).

Digital Object Identifier 10.1109/JSTARS.2022.3218983

In machine vision useful for developing electronic systems, machine intelligence is mainly used [3]. Synthetic aperture radar (SAR) image segmentation can be applied to military target detection, ocean monitoring, and crop estimation. Automatic SAR image segmentation has attracted increasing attention in the literature [4], [5]. As a critical step in the processing of SAR images, there are difficulties in segmenting SAR images correctly due to multiplicative speckle [6].

Well-known algorithms for SAR image segmentation include a series of methods such as threshold segmentation [7], segmentation based on edge detection [8], segmentation methods based on clustering [9], and segmentation using neural networks [10]. With the rise of neural networks, some researchers have begun to apply deep learning to applications such as targets classification [11], video processing [12], object detection [13], etc. In [12], Davari et al. used faster R-CNN to perform object detection on power devices in each frame in the video, and used color thresholding to identify corona discharges in the frame. Later, the ratio of corona to equipment area was used to determine the fault degree of power equipment. The algorithm could automatically identify early faults in distribution lines, and has strong practicability. Zalpour et al. [14] used deep learning to perform object detection on oil tanks. First, an improved faster R-CNN was used to extract the target of interest, and then a convolutional neural network (CNN) was used to extract high-order features. The algorithm used deep learning to have a high prediction accuracy for oil tank detection. Geng et al. [15] proposed a semisupervised depth joint distributed adaptive network model using transfer learning. It could match the original and target region's joint distribution probabilities and achieve high classification accuracy. While these algorithms generate satisfactory results, they also require significant time for training the network model. Feng et al. [16] designed a simple sampling method to train a semisupervised CNN, which reduced the running time. However, the method of deep learning relies heavily on training data. Furthermore, the training process often consumes large computing resources. Due to the limited open-source training datasets, some scholars still use traditional methods for SAR image segmentation.

In traditional SAR image segmentation, in order to obtain high segmentation accuracy, edge detection can be introduced as reference information. Its fundamental premise is to locate the gray value transition in an image. Traditional edge detection algorithms use a rectangular window to obtain pixel ratios. To reduce the influence of speckle on gray ratios, Shui et al. [17]

applied a Gaussian Gamma window to mitigate false edge information. Ganugapati et al. [18] used ratio of averages detector (ROA) to obtain the edge information of SAR image. Based on the mean ratio of ROA, the influence of noise can be weakened, and an accurate edge map can be obtained. Xiang et al. [19] proposed to use the sketch edge map to refine the segmentation of the generated superpixels further. Then the statistical region merging (SRM) framework was proposed to merge superpixels to obtain segmentation results quickly. Experimental results showed that this algorithm had high computational efficiency and a good segmentation effect. Jing et al. [20] introduced a new image-filtering method to maintain edges while smoothing homogeneous regions and obtained the edge strength map (ESM) using a Gaussian smoothing window. The NMS and double thresholding methods exploit the edge segmentation map. To get accurate edge information, Shang et al. [21] proposed a superpixel boundary-based edge description algorithm for SAR image segmentation (SpBED). This algorithm used the Gabor window and three edge detectors, the ROA, cross-correlation-based (CC) detector, and gradient detector, to find the edge information. Three types of edge information were fused interactively to reduce the effect of noise and obtain smoother edge detection. The edge map obtained was more stable because of its ability to identify gray values with high contrast. Edge detection is one of the important processes in SAR image segmentation since it provides information that can be used to achieve high segmentation accuracy.

Segmenting SAR images by clustering is a common approach. For the relationship between each pixel and adjacent pixels, if they are similar in color, texture, or gray level, they should be merged into the same label. Traditional methods include K-means, fuzzy C-means clustering (FCM), seed region growth method, etc. For K-means, it can cluster quickly, but it also has inherent disadvantages, which will produce noise spots. Each pixel is classified based on its membership matrix by FCM. Gong et al. [22] proposed a new Markov random field (MRF) energy function and added an additional term to modify the membership in FCM. FCM iteratively updates each pixel's membership, resulting in slow segmentation speed. To overcome this problem, Szilagyi et al. [23] directly calculated the gray histogram of the image to achieve fast segmentation. However, the algorithm proposed by Szilagyi must manually provide a parameter to balance noise and preserve image details, and this method has defects in boundary information preservation. Jing et al. [24] proposed to use the density peak (DP) algorithm and the knee point to select the number of clusters automatically. Then the improved K-means clustering was used for generated superpixels. The algorithm did not require clustering parameters and had high segmentation accuracy. Ji et al. [25] suggested a nonlocal FCM method for SAR image segmentation based on the between-cluster separation measure (NSFCM). To limit the influence of speckle, the program exploited nonlocal spatial information. In addition, the goal function included a fuzzy between-cluster variation term. Experimental results showed that the algorithm was better for image segmentation with some compact classes in feature space. However, this algorithm was prone to segmentation errors at edges. Xiang et al. [26]

proposed combining kernel FCM with pixel intensity and location information (ILKFCM). In addition, the energy measure of SAR image wavelet decomposition was used to represent texture information, which made the algorithm more robust. However, each step of the fuzzy factor requires iterations, so ILKFCM takes a long time to run. Based on the fact that the Gamma distribution resembles the probability distribution of speckles, Zhao et al. [27] introduced the Gamma distribution into the distance formula of FCM (Gamma-FCM), showing that the shape parameters were derived, and updating the membership of each pixel iteratively. Experiment results showed that each class in the image conformed to a Gamma distribution. However, due to the classification of each pixel separately, the algorithm was prone to noise. Compared with the region segmentation algorithms, the implementation of FCM and its improved methods are relatively simple, but they have some disadvantages. The segmented images have noise classification errors, and the FCM point-by-point iterative calculations are time-consuming.

Due to the noise problems of FCM clustering mentioned above, region-based segmentation has also been applied to SAR image segmentation. Classical methods include the Simple Linear Iterative Clustering algorithm (SLIC), watershed algorithm, meanshift, etc. Kurtosis wavelet energy (KWE) was proposed by Akbarizadeh as a high-order feature that can extract more statistical information for SAR image [28]. Akbarizadeh combined the KWE feature, wavelet energy feature, and gray values into a normalized feature vector which was used to train the SVM classifier. Experiments showed that the algorithm was effective for the classification of different textures in SAR images. In addition, Tirandaz et al. [29] proposed to use kurtosis curvelet energy (KCE) to design the optimal kernel function. The boundaries of each layer were determined by using the estimation function of KCE. Since KWE and KCE are two efficient estimating methods, they can be applied to superpixel segmentation. Zou et al. [30] introduced a local clustering scheme combining spatial proximity and data similarity and used generalized Gamma distribution to model SAR images accurately. Lei et al. [31] developed a method based on superpixel and fast fuzzy C-means clustering for color image segmentation (SFFCM). SFFCM utilized generated superpixels to simplify the images and obtain image histograms. Then the color images were clustered based on the fuzzy C-means of the histogram, which reduced the running time. Due to satisfactory performance in color image segmentation, SFFCM was extended to gray image segmentation. However, due to the lack of edge information, it was insensitive to adjacent regions with approximate gray values. Jing et al. [32] introduced a new superpixel generation method and clustered the superpixels using a shrinkage expansion strategy instead of K-means. This algorithm could obtain superpixels with low computational cost and high edge adhesion. Wang et al. [33] accurately detected ship targets at a low signal-to-clutter ratio by combining local contrast of Fisher vectors (LCFVs) with superpixels. The algorithm can segment different target regions of the SAR image accurately.

Because SLIC is simple to implement and can resist speckle, this algorithm is often used for SAR image segmentation. Tirandaz et al. [34] proposed to incorporate improved SLIC

results under the constraints of feature and edge information. The label map produced by K-means was then optimized using Hidden Markov random field-expectation maximization (HMRP-EM) and zero padding weighted neighborhood filter bank (ZPWNTFB). The final segmentation was obtained by combining the above results. The experimental results showed that it could effectively resist noise and achieve high accuracy. Ghaffari et al. [35] used robust FCM clustering to classify SAR images into homogeneous and nonhomogeneous regions. The superpixel was generated by SLIC and the fast weighted conditional random fields (FWCRF) algorithm was used to mark the image to obtain the higher segmentation accuracy. A potential area for improvement is that the L0 smoothing method used by the algorithm makes regions with similar gray values have closer values, hence two regions with similar gray levels cannot be effectively segmented. Shang et al. [21] used SLIC and boundary information to generate superpixels and obtained the segmentation results through K-means. Experimental results showed the segmentation result could maintain the boundary information. However, SpBED ignores the noise produced by K-means. The effect of speckle on image segmentation can be avoided to some extent through the use of superpixels in region segmentation. However, it is still a challenge to accurately segment regions with insignificant gray contrast.

To reduce the effect of speckle on segmentation, some scholars use region segmentation to segment SAR images, but traditional methods cannot effectively segment adjacent regions with similar gray values. To solve this problem, an SAR image segmentation method based on Fisher vector superpixel generation and label revision (FVSGLR) is proposed. First, the Gaussian mixture model (GMM) is estimated by maximum likelihood, and then the variables are derived to obtain the mixture parameter set $\{\omega_k, \mu_k, \sigma_k\}$. Then the third-order information of the Fisher vector is acquired by normalizing and regularizing the variables in the mixture parameter set. A superpixel generation algorithm based on Fisher vector (FV-SLIC) is proposed. The third-order information is introduced into the superpixel distance formula, and the labels of all pixels are iteratively updated to obtain the superpixel result map. Since the distance formula considers more dimension information, the above steps can effectively distinguish adjacent regions with similar gray values. Second, according to the edge information generated by previously proposed algorithm SpBED, the superpixel result map is segmented again to obtain finer superpixels. The edge information is integrated into the superpixels, and then the small superpixels are fused. Third, K-means is used to cluster the fused superpixels, and the Canny algorithm is utilized to obtain the edge result image. Fixed window label revision based on label and gray information (LRLG) is proposed to eliminate the noise points generated by K-means. The edge result image is combined with the edge information obtained by SpBED to get the final edge information. Under the condition of edge limitation, the label is updated by using the gray value and the number of labels. Finally, the region growth algorithm is implemented to find isolated pixel blocks with no boundaries in the homogeneous region. Using the isolated pixel blocks label revision (IPBLR) algorithm, the isolated pixel blocks that meet the fusion conditions are fused

into the neighborhood label, and the final segmentation result is obtained.

The main contributions of this article are as follows.

- 1) The superpixel distance formula introduces the Fisher vector third-order information to update the label of pixels iteratively. It can effectively segment the adjacent regions with similar gray values.
- 2) LRLG, under the constraints of edge information, can revise the noise points by using gray information and the number of labels. This method can eliminate the noise points generated by K-means.
- 3) IPBLR, using region growth algorithm to calibrate isolated pixel blocks in the homogeneous region, can solve the label error in pixel blocks caused by superpixel segmentation.

The rest of this article is organized as follows. Section II discusses in detail the proposed FVSGLR algorithm. Section III analyzes the results of the experimental results performed by each algorithm on synthetic and real images. Finally, Section IV concludes this article.

II. PROPOSED METHOD

To achieve a higher segmentation accuracy, and avoid generating noise points, this article proposes an SAR image segmentation method based on FVSGLR. The block diagram of the proposed FVSGLR algorithm is shown in Fig. 1.

As shown in Fig. 1, first, the Fisher vector is introduced into the superpixel distance formula, and SLIC superpixel segmentation is used. This method can effectively segment the adjacent regions with similar gray values. The ESM obtained by edge detection is introduced into the superpixel result image for resegmentation and fusion to obtain the superpixel. Second, to get the Initial result, K-means is used to cluster superpixels. To eliminate the noise points generated by K-means, under the given edge information constraint, the target pixel label is repaired by a label revision algorithm based on label and gray information (LRLG) to obtain a revised result. Finally, to solve the pixel blocks with label errors caused by superpixel segmentation, a region growth algorithm is used to find isolated pixel blocks in the revised result. The isolated pixel blocks that meet the fusion conditions are fused into neighborhood labels by the IPBLR algorithm to obtain the final result. The remainder of this section focuses on superpixel generation based on the Fisher vector and edge constraints, fixed window label revision based on label and gray information, and revision of isolated pixel block labels.

A. Superpixel Generation Based on Fisher Vector and Edge Limitation

The edge detection technique locates the pixel position with the maximum difference in the gray area. The SAR images also have adjacent regions with small differences in gray value. In order to segment the SAR image correctly, it is necessary to detect weak edges, which can be obtained by generating superpixels.

Traditionally, SLIC, watershed algorithm, or ecological methods were used to over-segment images. However, when the

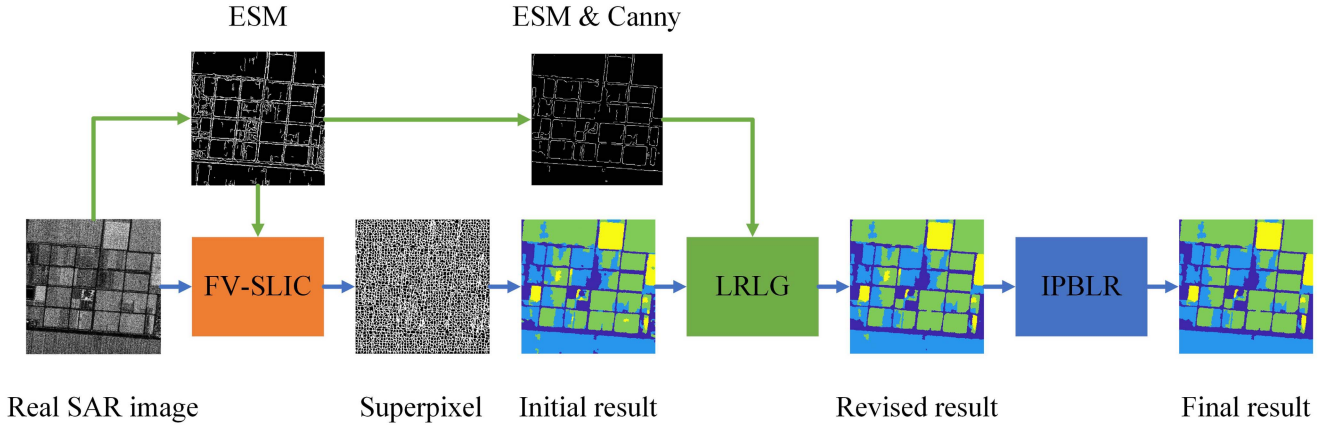


Fig. 1. Block diagram of the proposed FVSLR algorithm.

difference of gray values in the segmentation area is too small, it is impracticable to segment effectively in this way. Therefore, the Fisher vector is introduced to introduce additional information into the distance similarity formula when superpixel segmentation is performed on ground objects. Parameter set $\{\omega_k, \mu_k, \sigma_k\}$ is constructed by deriving the variables of a GMM

$$f_G(x) = \sum_{k=1}^K \omega_k f_k(x) \quad (1)$$

where K denotes the gaussian function's total number. The k th Gaussian function is $f_k(x)$. $\sum_{k=1}^K \omega_k = 1$, where the weight of the k th Gaussian function is represented by ω_k . Then the parameter set in the GMM are normalized separately to get the Fisher vector $\alpha_p^{(\omega, \mu, \sigma)}$

$$\alpha_p^{(\omega, \mu, \sigma)} = [\alpha_p^{\omega_1}, \dots, \alpha_p^{\omega_K}, \alpha_p^{\mu_1}, \dots, \alpha_p^{\mu_K}, \alpha_p^{\sigma_1}, \dots, \alpha_p^{\sigma_K}]^T \quad (2)$$

$$\alpha_p^{\omega_k} = \frac{1}{\sqrt{\beta_k}} (\varphi_{p,k} - \beta_k)$$

$$\alpha_p^{\mu_k} = \frac{x_p - \mu_k}{\sigma_k \sqrt{\beta_k}} \varphi_{p,k}$$

$$\alpha_p^{\sigma_k} = \frac{\varphi_{p,k}}{\sqrt{2\beta_k}} \left(\frac{(x_p - \mu_k)^2}{\sigma_k^2} - 1 \right) \quad (3)$$

where $\beta_k = \frac{\exp(\omega_k)}{\sum_{k=1}^K \exp(\omega_k)}$, $\varphi_{p,k} = \frac{\omega_k f_k(x_p)}{\sum_{k=1}^K \omega_k f_k(x_p)}$. ω_k , μ_k , and σ_k represent the weight, mean, and standard deviation of the k th GMM, respectively. $\alpha_p^\omega = [\alpha_p^{\omega_1}, \dots, \alpha_p^{\omega_K}]^T$, $\alpha_p^\mu = [\alpha_p^{\mu_1}, \dots, \alpha_p^{\mu_K}]^T$, $\alpha_p^\sigma = [\alpha_p^{\sigma_1}, \dots, \alpha_p^{\sigma_K}]^T$. α_p^ω , α_p^μ , α_p^σ are updated by their signed inner product square root and l_2 -normalization, respectively.

Before generating superpixels, to prevent noise from affecting the segmentation, Gaussian smoothing is applied to the input image. SLIC is used to segment the image due to its simple implementation and good segmentation effect. The size of the superpixels is S_p . Next, each seed point interval S is $\sqrt{S_p}$. Each initial center point falls at a lower gradient in the 3×3 neighborhood to prevent falling into the edge and affecting the

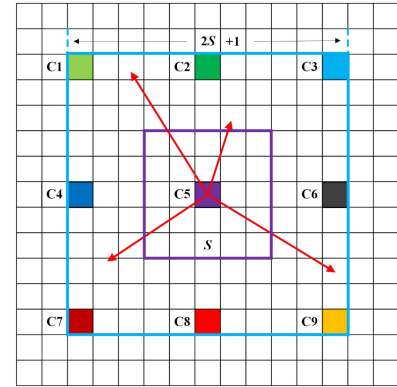


Fig. 2. Exploration range of superpixel centroid.

segmentation. The similarity detection range of superpixel seed points is shown in Fig. 2. As shown in Fig. 2, each superpixel center point calculates the distance similarity within its exploration range. The purple superpixel seed judges the similarity of each point within the range $(2S+1) \times (2S+1)$ with itself as the center, where S is the step of each superpixel seed. Each superpixel seed point is assigned a label. If the point in the range has the shortest set distance from the seed point, it will be assigned as the label of this superpixel seed. As analyzed above, this process is performed for each superpixel seed point.

Each pixel point is finally marked with the label of the superpixel center with the shortest set distance. The superpixel center's gray value and coordinates are then modified. The update process will not stop until the number of iterations is reached. The Fisher vector obtains three-order information by deriving the weights, means, and standard deviations in a GMM with maximum likelihood estimation. α_i^ω , α_i^μ , α_i^σ denote the weight, expectation, and standard deviation vector of the Fisher vector for the i th pixel. The distance similarity formula generated by superpixels is then as follows:

$$d_{ij} = \frac{S_p}{\theta} (I_s^i - I_s^j)^2 + E_{ij} + \|\alpha_i^{(\omega, \mu, \sigma)} - \alpha_j^{(\omega, \mu, \sigma)}\|_2 \quad (4)$$

$$E_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2 \quad (5)$$

Algorithm 1: Description of FV-SLIC Algorithm.

Input: SAR image I , Gaussian smoothing σ , Gaussian window G_w , the balance parameter θ , the number of Gaussian mixture function K , the size of the superpixel S_p ;

Input: initial label map;

- 1: Use Gaussian filter function to smooth image I to get I_s ;
- 2: Use formula (1) Gaussian mixture function for maximum likelihood estimation of I_s to get the best parameter set;
- 3: Get the Fisher vector by normalizing and regularizing;
- 4: Use the SLIC method and fomula (4) to segment superpixel;
- 5: Post-processing with *seledge* information;
- 6: Under the constraints of *seledge*, smooth the gray values of superpixels;
- 7: Use K-means to cluster the smoothed superpixels.
- 8: Get the initial label map.

where the similarity distance between the i th and j th points is denoted by d_{ij} , with lower values indicating higher similarity. x_i and y_i represent the horizontal and vertical coordinates of the i th point. $i, j \in \{1, 2, \dots, R * C\}$, where the input image's row and column are denoted by R and C . θ is the balance parameter. I_s^i represents the gray value of the i th pixel in the smoothed image.

The edge information after over-segmentation is obtained by generating superpixels based on the Fisher vector. Then edge information *selEdge* is generated in SpBED, which is used to segment the superpixels once more to get smaller superpixels. The region growth algorithm is then used to mark labels of strong edge pixels as superpixel labels with the closest gray values. After processing the superpixels, small blocks of superpixels are merged into adjacent superpixel blocks with the smallest pixel mean difference without strong edges between them. Finally, the superpixels are clustered by K-means to generate the initial result map. The description of the FV-SLIC algorithm is shown in Algorithm 1.

B. Fixed Window Label Revision Based on Label and Gray Information

K-means has its inherent shortcomings and tends to fall into local optimality. After the clustering mentioned above of superpixel blocks, a window may have multiple missegmented labels due to incorrect classification of some superpixels. In addition, K-means generates noise points, especially in the boundary. Fig. 3 demonstrates the steps involved in the label revision process.

The revision process is shown in Fig. 3. The red box in Fig. 3(a) shows that there are purple and blue pixel points in the yellow and green borders. The purple and blue points represent noise points. The bottom of Fig. 3(b) shows the repair process of the purple pixel group. Among them, there is a boundary between yellow and green pixels. Under the fixed window, the number of purple pixels is first counted. Then the region growing

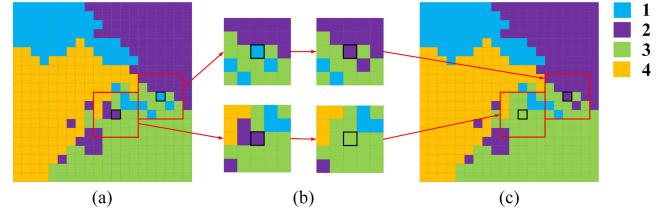


Fig. 3. Image revision process. (a) Error label. (b) fixed window label revision process. (c) Fixed window label revision result.

algorithm is used to count the number of pixel groups adjacent to the purple pixels which have no boundaries. The ones that meet the conditions and have the largest number are the green pixel groups. The difference between the gray mean of the green pixel group and the purple pixel group is less than the set threshold, so the purple pixel is marked as the green pixel group. Fig. 3(c) shows the result after restoring the purple and blue pixel points. The specific process of the above label revision algorithm is as follows.

The final edge map *edge* is created by applying the Canny algorithm to the initial label map and then combining it with the strong edges *selEdge*. The algorithm searches for nearby pixels with the same label starting at the center point. Those whose number is less than the quantity threshold T_n are regarded as the blocks to be processed, and the label is $label_i$. It will otherwise be moved. The pixels of other objects are accumulated under the edge limit. A vector is used to find the pixel block with the maximum label value $label_{max}$ in the fixed window. The gray mean difference between the $label_i$ and $label_{max}$ is judged. If it is less than the threshold T_{gray} , $label_i$ is changed to $label_{max}$. Otherwise, the label is assigned to the $label_j$ that is a different label and the neighborhood's nearest pixel value. This process is defined as follows:

$$label_{max} = \arg \max(num(label_i)) \quad \forall i \in L \quad (6)$$

$$gray_{max} = g_{avg}(label_{max}) \quad (7)$$

$$gray_i = g_{avg}(label_i) \quad (8)$$

$$T_{gray} = \frac{g_{max} - g_{min}}{P} \quad (9)$$

$$label_i = \begin{cases} label_{max}, & \text{if } abs(gray_{max} - gray_i) < T_{gray} \\ label_j, & \text{else if } abs(gray_i - gray_j) = M \end{cases} \quad (10)$$

where L denotes the total number of label sets under the fixed window, $label_i$ denotes the i th label, and $num(label_i)$ denotes the number of i th labels. $label_{max}$ represents the maximum number of labels for the fixed window. $g_{avg}(label_i)$ is the i th label's gray mean value. $gray_j$ is the mean gray of the j th neighborhood pixel group of label i , and T_{gray} is the set pixel value difference threshold. g_{max} , g_{min} represent the maximum and minimal gray value in the image. P is the number of clusters. M is the minimum difference between the i th pixel block and the surrounding pixel block.

After restoring the image by the above algorithm LRLG, the boundary information *edge* is combined with the resultant map.

Algorithm 2: Description of LRLG Algorithm.

Input: initial label map, $seledge$, the edge information $canny$ obtained by the Canny algorithm, the fixed window w_g , quantity threshold T_n , gray threshold T_{gray} ;
Input: middle label map;
1: Merge the edge information $canny$ with $seledge$ to get $edge$;
2: **while** pixel in initial label map has not checked **do**
3: Under the edge limit, perform window sliding to find the same pixel as the center pixel label, if its amount $< T_n$, set it as the block to be processed, and $label=label_i$;
4: Continue to find the $label_{max}$ with the largest number of labels in the fixed window;
5: Using formula (7) and (8) to calculate the mean gray value of $label_{max}$ and the $label_i$;
6: **if** $abs(gray_{max} - gray_i) < T_{gray}$ **then**
7: $label_i = label_{max}$;
8: **else**
9: $label_i$ is covered with the smallest difference between the gray values of the pixel groups adjacent to $label_i$;
10: **end if**
11: **end while**
12: Get the middle label map.

For the case that the boundary will have isolated points, each pixel will be checked to see whether its four neighborhoods are not of the same label. If the condition is met, the label with the largest number in the eight neighborhoods of the edge pixel is found to cover the boundary point

$$label_{(x,y)_m} = \arg \max(num(label_{(x,y)_{n_8}})) \quad (11)$$

$$label_{(x,y)} = \begin{cases} label_{(x,y)_m}, & \text{if } label_{(x,y)} \neq label_{(x,y)_{n_4}} \\ label_{(x,y)}, & \text{else} \end{cases} \quad (12)$$

where $label(x, y)$ denotes the label at (x, y) in the initial result map, and $label_{(x,y)_{n_4}}$ represents the label value in the four neighborhoods of point (x, y) . $label_{(x,y)_m}$ denotes the label with the largest number of labels in the eight neighborhoods of point (x, y) . The description of LRLG algorithm is shown in Algorithm 2.

C. Isolated Pixel Block Labels Revision

The speckle in some homogeneous regions results in a significant difference between the generated new superpixels and the neighborhood superpixel blocks, leading to K-means clustering errors. In this article, a correction strategy based on is also presented to resolve this issue. Algorithm 3 shows the description of the IPBLR algorithm.

It can be seen from Algorithm 3 that the region growth algorithm is used to reassign the labels of the middle map. For each separated label pixel block, the labels are reassigned. The bigger location of the global pixel block is then determined to see

Algorithm 3: Description of IPBLR Algorithm.

Input: middle label map, quantity threshold T_m , the threshold of gray mean difference T ;
Input: final label map;
1: Use the region growth algorithm for the middle label map to reassign the pixel labels;
2: **while** the pixel of middle label map has not checked **do**
3: Find isolated blocks of pixels smaller than a threshold T_m , and $label = label_i$;
4: Check whether the isolated pixel blocks $label_i$ has boundary;
5: Using formula (13), update the value of $label_i$;
6: **end while**
7: Get the final label map.

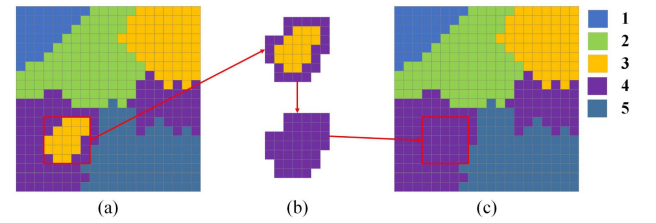


Fig. 4. Correction process of Isolated pixel block. (a) Isolated pixel block in homogeneous area. (b) Label revision process. (c) Restoration results.

if this region has just one sort of neighborhood. The pixel block will not be handled if it is on the border. The gray difference between the target and neighboring pixel blocks is computed otherwise. If the difference is less than the threshold T , the pixel block will be fused into the neighborhood to get the final label result. The process is shown in Fig. 4.

It can be seen from Fig. 4 that there are yellow isolated pixel blocks in the purple pixel blocks. The yellow pixel blocks are transformed into purple using the region growth algorithm and formula 13. Each pixel block is relabeled using the region growth algorithm. The yellow pixel group whose number of pixel blocks is less than the threshold is found by setting the threshold T_m . The adjacent pixel blocks of the yellow pixel block in the red box only have purple pixel groups, and the yellow pixel block is not at the boundary. The yellow pixel block is corrected to the purple pixel block according to formula 13. Because the difference between the pixel mean value of the yellow pixel block and the purple pixel group is smaller than the set threshold T , which meets the criteria in formula 13. This process is defined as

$$label_i = \begin{cases} label_j, & \text{if } num(label_i) < T_m \\ & \text{and } adjacency[i] = 1 \\ & \text{and } gray_i - gray_j < T \\ label_i, & \text{else} \end{cases} \quad (13)$$

$$T = \frac{2 \times (g_{max} - g_{min})}{P} \quad (14)$$

where j represents the different neighborhood subscripts of the i th pixel block. The i th pixel block does not contain boundary pixels. g_{avg} denotes the gray mean value of pixels. $adjacency[i]$

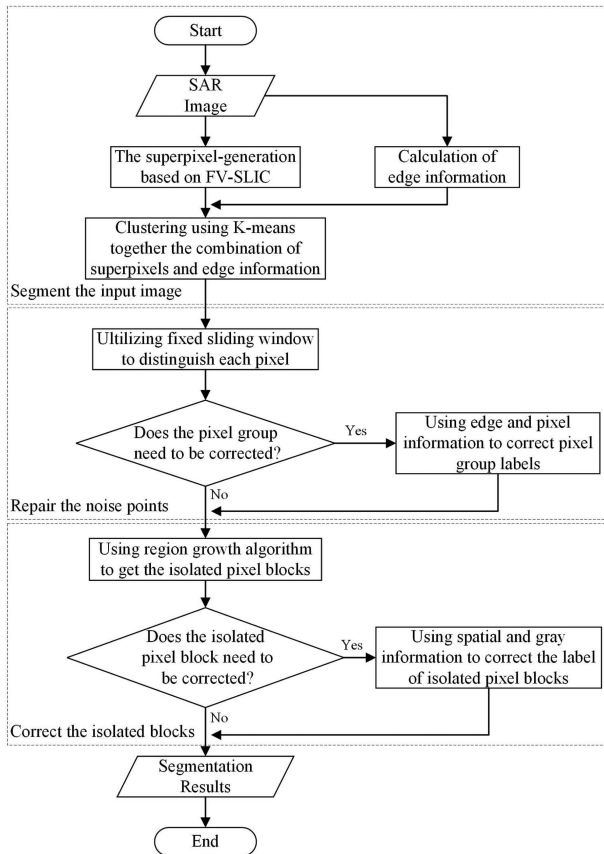


Fig. 5. Flowchart of the proposed FVSGLR algorithm.

represents how many neighboring pixel blocks the i th pixel block has. The T_m value indicates the threshold value for identifying tiny blocks. The threshold of difference of gray values between two pixel blocks is represented by T .

D. Flowchart and Pseudocode of the Algorithm

To better understand the FVSGLR algorithm, the flowchart of the proposed FVSGLR algorithm is shown in Fig. 5. As shown in Fig. 5, the proposed FVSGLR algorithm has three main steps. First, the similarity formula of SLIC introduces the third-order information of the Fisher vector. The improved SLIC is used to generate superpixels on the Gaussian smoothed image. The superpixels are further divided using the extracted edge information. The initial label result is obtained by using K-means. Second, each point in the initial label result is traversed by a fixed sliding window. Under the limitation of edge information, label revision is performed on the noise points generated by K-means by referring to the gray value and number of labels. The revised label result is obtained. Third, the region growth algorithm is used to find the isolated pixel blocks. The spatial and gray information is used to correct the labels of the eligible isolated pixel blocks. The segmentation result is obtained.

To achieve high segmentation accuracy, dealing with speckle is a key issue. After introducing the algorithm, how the proposed FVSGLR algorithm handles speckle during image segmentation is described below. First, the proposed FVSGLR uses the

Algorithm 4: The Framework of the Proposed FVSGLR Algorithm.

Input: SAR image I , superpixel initial size S_p , the number of cluster P , the number of Gaussian mixture function K , maximum iteration count $Iter$;

Input: the result map of segmentation S ;

- 1: Bias derivative of the weights, means, and standard deviations of the log-likelihood global GMM; Get the third-order information of the Fisher vector;
- 2: **for** $i < Iter$ **do**
- 3: With FV-SLIC algorithm generates superpixels;
- 4: $i++$;
- 5: **end for**
- 6: Create a Gabor function edge detection template, and use three edge detection algorithms to extract edge;
- 7: Use the nonmaximum suppression algorithm and double threshold method to obtain the final edge $selEdge$;
- 8: Integrate $selEdge$ into superpixels;
- 9: **while** Finish smoothing superpixels **do**
- 10: **if** Adjacent superpixels exist $selEdge$ **then**
- 11: Continue;
- 12: **else**
- 13: Use gray information to smooth each other;
- 14: **end if**
- 15: **end while**
- 16: Use K-means to cluster superpixels to obtain Initial result;
- 17: **while** Initial result map exists noise pixels **do**
- 18: Use LRLG algorithm to revise noise points;
- 19: **end while**
- 20: Get the Revised result;
- 21: **while** Pixel blocks meet (13) **do**
- 22: Use IPBLR algorithm to revise the isolated pixel blocks;
- 23: **end while**
- 24: Get the Final result S .

Gaussian kernel function to perform simple filtering on the input image I . Second, using superpixel generation, the pixels with similar characteristics are formed into sub-regions, and the gray values in the sub-regions are set as the average value. This method averages the influence of speckle on different pixels. Finally, due to speckle, superpixels will have missegmented regions, resulting in K-means clustering errors. The proposed FVSGLR algorithm uses a region growth algorithm to find isolated pixel blocks and uses gray and spatial information to correct the labels of eligible pixel blocks. Through the above method, the influence of speckle on image segmentation can be effectively reduced. When all steps are completed, the framework of the proposed FVSGLR algorithm is shown in Algorithm 4.

E. Computational Complexity Analysis

Suppose the input image size is $n \times m$. The size of the multiscale gabor window function is $scale \times scale$, the direction is

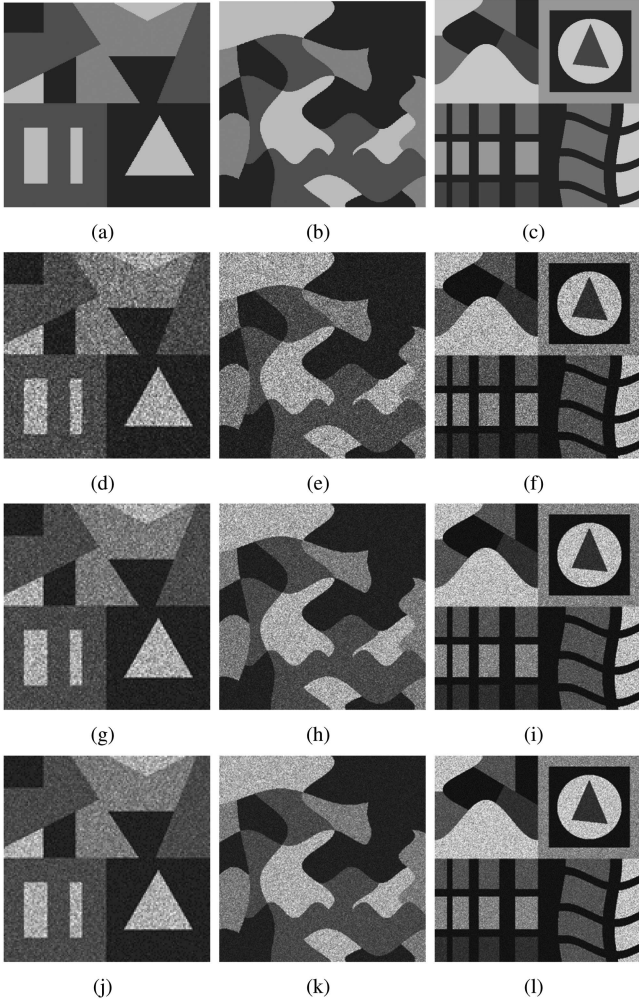


Fig. 6. Synthetic SAR images. (a) SI1. (b) SI2. (c) SI3. (d)–(g) and (j) 2-, 4-, and 6-look of SI1. (e)–(h) and (k) 2-, 4-, and 6-look of SI2. (f)–(i) and (l) 2-, 4-, and 6-look of SI3.

dirnum, and the total product is recorded as N . The number of superpixels to be fused is K . The main time consumption of the algorithm in this article can be divided into three parts, the first is edge information generation, the second is superpixel generation and subsequent processing, and the third is label revision. The time complexity of edge information generation is $O(N \times n \times m)$. The time complexity of superpixel generation and subsequent processing is $O(K \times n \times m)$, and the label revision is $O(n \times m)$. Since superpixel generation needs to perform $iter$ times, and superpixel fusion uses 8 neighborhood exploration, $iter \times K \times 8$ can reach 10^4 , so the overall computational complexity of the algorithm in this article is $O(K \times n \times m)$.

III. EXPERIMENTAL SETUP AND ANALYSIS OF RESULTS

A. Experimental Configuration

In the same environment, performance of the proposed FVS-GLR is compared against that of six state-of-the-art algorithms, on the both synthetic SAR images and real SAR images. The

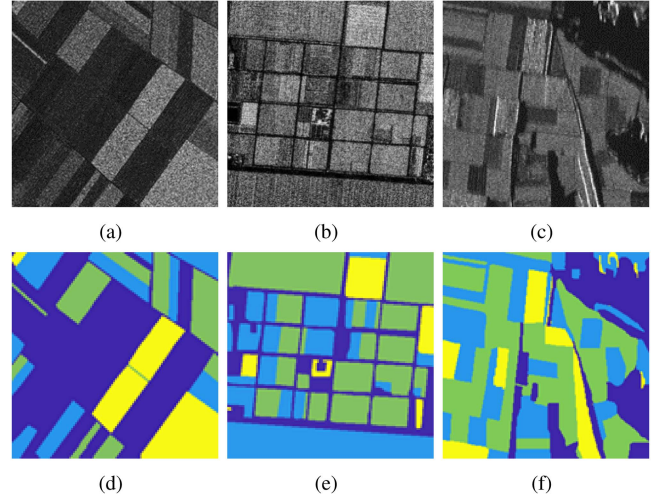


Fig. 7. Real SAR images. (a) Noerdlinger image. (b) Maricopa image. (c) Xian image. (d) Ground truth of Noerdlinger. (e) Ground truth of Maricopa. (f) Ground truth of Xian.

proposed FVSGLR algorithm and the comparison algorithm are implemented using MATLAB programming. All the algorithms run in the following environment: CPU Intel Core i5-4590 CPU @ 3.30GHZ 3.30GHZ, 8 G RAM, Win10 64-b operating system, and MATLAB2021a.

B. Experimental Images

Three sets of synthetic SAR images are selected for the dataset. They are synthetic image 1 (SI1), synthetic image 2 (SI2), and synthetic image 3 (SI3), as shown in Fig. 6.

The three sets of synthetic SAR images are generated by simulating the effect of coherent speckle noise on noiseless images. The generated synthetic SAR images are 2-, 4-, and 6-look of synthetic images, respectively. The first set of synthetic SAR images SI1 is shown in Fig. 6(d)–(g) and (j), and the size of this synthetic SAR image is 256×256 , which contains four classes. The SI2 image has a size of 384×384 and consists mainly of curves. It can be segmented into four types of targets with different gray values, as shown in Fig. 6(e)–(h) and (k). The synthetic SAR image SI3 has a size of 512×512 and consists of both curves and lines. It can be segmented into five classes, as shown in Fig. 6(f)–(i) and (l). The SI3 image is more difficult to segment because it has more corner points, and the gray values between targets are more similar. Since both SI2 and SI3 contain corner-point targets, they are possible to test the algorithm to segment small corner-point targets.

In this section, a set of real SAR images is also selected as a test image. They are Noerdlinger, Maricopa, and Xian image, as shown in Fig. 7.

Fig. 7(a) is an SAR image named Noerdlinger with the original resolution of 1 m, HH polarization, and in X-band. It was captured by TerraSAR-X, situated in the middle of the Swabian Jura in southwestern Germany, which is 256×256 in size and can be segmented into four different types of farmland areas. Fig. 7(b) shows the second real SAR image named Maricopa, imaged in the Ku-band and VV polarization, located at the Maricopa

Agricultural Center near Arizona. The size is 350×350 and the resolution is 1. The image contains four types of targets such as farmland, roads, and water. Fig. 7(c) shows Xian's real SAR image, taken in the X-band by TerraSAR viewing Xi'an, China, at a resolution of 1 m. Xian image is an eight-look SAR image. The size of image Xian is 256×256 and can be divided into four regions, including three kinds of farmland and water. Fig. 7(e)–(f) are the ground truth images for the three real SAR images, which are manually annotated to directly and objectively compare the proposed FVSGLR algorithm and contrastive algorithms.

C. Comparison Algorithms and Evaluation Metrics

In this section, six better algorithms in recent years are used as comparison algorithms. SFFCM, FWCRF, and SpBED are SAR image segmentation based on superpixel, and ILKFCM, NS-FCM, and Gamma-FCM are SAR image segmentation based on FCM. This article takes three evaluation metrics for comparison: Segmentation accuracy (SA), the consistency test coefficient (Kappa) and the intuitive segmentation effect. The formula of SA and Kappa are as follows:

$$SA = \frac{\sum_{i=1}^P L_i \cap G_i}{\sum_{i=1}^P G_i} \quad (15)$$

$$Kappa = \frac{SA - pe}{1 - pe} \quad (16)$$

$$pe = \left(\sum_{i=1}^P num(L_i) \times num(G_i) \right) / (R \times C)^2 \quad (17)$$

where L_i denotes i th label pixels in the segmentation result. G_i represents i th label pixels in the ground truth. $num(L_i)$ is the num of i th label pixels in the segmentation result.

D. Parameter Setting and Analysis

In this article, the initial setting range of the number of pixels S_p contained in the superpixel is [10, 200]. Experimental results suggest that the best pixel range contained in superpixel is [20, 90]. Gaussian smoothing σ is 3.1. The number of Gaussian mixture function K is 7. Gaussian window G_w is 5. Set the θ to 15. The sliding fixed window W_g is set to 19. T_n is the quantity threshold for finding the blocks to be processed in the sliding fixed window W_g . T_m is the threshold for finding isolated pixel blocks. R is the number of rows of the input image. To select the effective values of T_n and T_m , the relevant experiments of SA on Maricopa with T_n and T_m are performed in this algorithm. Table VI shows the SA of Maricopa with the different values of T_n and T_m .

As can be seen from Table VI, the value range of T_m is $[R \times 0.8, R \times 1.3]$ and the interval is $R \times 0.1$. The value of T_n starts from 10 and increases to 170 in increments of 20. From the Table VI, SA increases with the increase of T_n when the value of T_m remains unchanged, and then gradually becomes stable. With a fixed T_n , SA increases with increasing T_m and then plateaus. The larger T_n and T_m are, the more pixel groups need to be processed, and the more time is cost, so the proposed

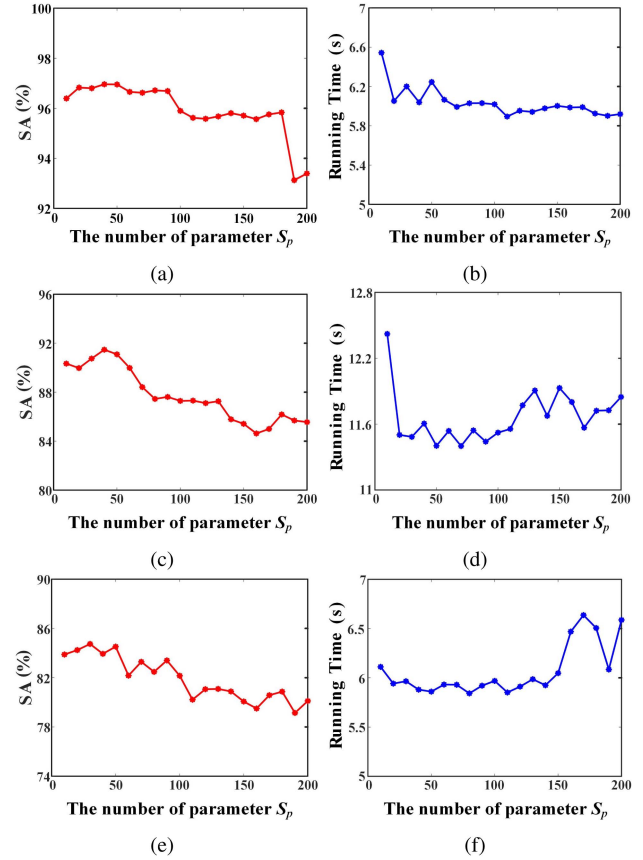


Fig. 8. Impact of parameter S_p on SA and time. (a) SA of Noerdlinger. (b) Time of Noerdlinger. (c) SA of Maricopa. (d) Time of Maricopa. (e) SA of Xian. (f) Time of Xian.

FVSGLR algorithm takes the initial values of T_n and T_m when SA tends to be stable in the Table VI. For the Maricopa image, the proposed FVSGLR algorithm set $T_n=130$ and $T_m=R$. For other images, when SA takes the best result, the sizes of T_n and T_m are similar to the current values.

To test the effective range of S_p , the algorithm performs image segmentation on the three real SAR images. The SA and running time of different S_p are shown in Fig. 8.

The number of parameters S_p starts from 10 and increases to 200 in increments of 10. As shown in Fig. 8(a)–(c) and (e), when S_p is 10, the SA is relatively low. Because there are too few pixels in each superpixel, the gray value of adjacent pixel groups is significantly different, and the cluster will appear noise. The optimal range for S_p is [10, 90]. As the superpixel contains more and more pixels, the superpixel will contain many misclassified pixels, resulting in misclassification.

As shown in Fig. 8(b)–(d) and (f), the algorithm's running time decreases as S_p increases. The reason is that when the initial S_p is small, the number of superpixels is large. The runtime is long due to a large number of iterations while creating the superpixel. As the number of superpixels decreases, the running time decreases. When the number is reduced further, the detection range of each superpixel center point becomes larger, causing the running time to increase again. Combining SA and running time, when

TABLE I
SA OF MARICOPA WITH THE DIFFERENT VALUES OF T_n AND T_m

$T_n \backslash T_m$	$R \times 0.8$	$R \times 0.9$	$R \times 1$	$R \times 1.1$	$R \times 1.2$	$R \times 1.3$
10	90.39	90.32	90.28	90.63	90.28	90.57
30	90.56	90.54	90.44	90.82	90.73	90.73
50	90.68	90.82	90.67	90.97	90.94	90.96
70	90.74	90.74	90.99	90.99	91.00	91.00
90	90.99	90.99	91.24	91.24	91.24	91.24
110	91.26	91.24	91.50	91.50	91.52	91.50
130	91.25	91.47	91.73	91.73	91.72	91.72
150	91.44	91.44	91.43	91.70	91.70	91.70
170	91.59	91.16	91.52	91.52	91.52	91.70

S_p is in the range [20, 90], the algorithm works well on the real datasets.

E. Experimental Results and Analysis

1) *Results and Analysis of Synthetic Images:* The SA and kappa coefficient of each algorithm on the 2-, 4-, and 6-look SII synthetic SAR images are shown in Table I.

As shown in Table I, the accuracy of each comparison algorithm is above 90%, among which the SpBED algorithm and Gamma-FCM algorithm can achieve higher SA. With the increase of noise, the accuracy of SFFCM and FWCRF gradually decreases and fluctuates greatly, indicating that the two algorithms are susceptible to noise and are less robust. ILKFCM, NSFCM, SpBED, and the proposed FVSGLR algorithm have little volatility and good stability. By the two evaluation indexes of SA and Kappa, the proposed FVSGLR algorithm achieves a better image segmentation.

The experimental results of each algorithm on 2-look SII synthetic image are shown in Fig. 9.

As shown in Fig. 9(d), FWCRF's result has many pixel point segmentation errors, resulting in poor image intuitive segmentation. Fig. 9(e) shows that SFFCM cannot be utilized with boundary information, resulting in regions with low grayscale differences that cannot be segmented accurately. As shown in Fig. 9(f) and (g), ILKFCM and NSFCM can perform accurate segmentation in homogeneous regions but cannot effectively classify the boundary information, resulting in classification errors. The SpBED and the proposed FVSGLR algorithms can segment the synthetic image accurately. Still, there are individual small spots in the homogeneous region in Fig. 9(h), which leads to slight inferiority of the result compared with the proposed FVSGLR algorithm. The proposed FVSGLR algorithm uses the boundary information to segment them and correct the pixel groups classified in the homogeneous regions, and a better segmentation effect can be seen in Fig. 9(c).

The synthetic image of each look of SI2 are used as the segmentation images. The SA and Kappa are calculated for the algorithm of this article, and the six comparison algorithms, as shown in Table II.

It can be seen from Table II that for the synthetic images, the accuracy of each comparison algorithm is above 90%. Among them, ILKFCM and Gamma-FCM algorithms have lower segmentation accuracy due to the inability to accurately classify

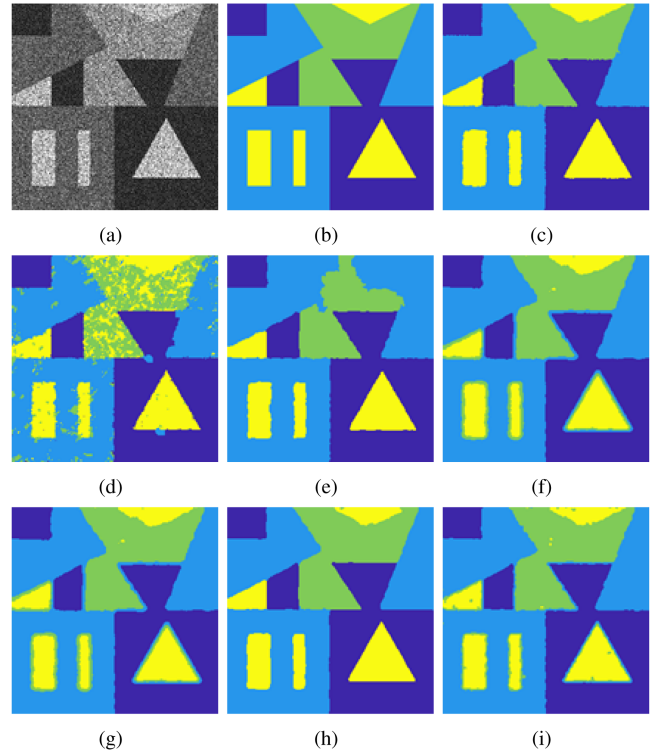


Fig. 9. Segmentation results on 2-look SII synthetic image. (a) 2-look SII. (b) Ground truth. (c) Proposed FVSGLR. (d) FWCRF. (e) SFFCM. (f) ILKFCM. (g) NSFCM. (h) SpBED. (i) Gamma-FCM.

TABLE II
EXPERIMENTAL RESULTS ON SII SYNTHETIC SAR IMAGES

Algorithm	2-look		4-look		6-look	
	SA(%)	Kappa	SA(%)	Kappa	SA(%)	Kappa
FWCRF [35]	90.81	0.871	92.23	0.890	93.12	0.902
SFFCM [31]	91.45	0.877	98.57	0.980	96.73	0.954
ILKFCM [26]	93.02	0.901	93.35	0.906	93.23	0.904
NSFCM [25]	92.86	0.897	93.32	0.904	94.22	0.917
SpBED [21]	98.73	0.984	98.90	0.984	98.99	0.986
Gamma-FCM [27]	94.76	0.926	94.55	0.923	95.00	0.929
Proposed FVSGLR	98.95	0.985	99.24	0.989	99.29	0.990

boundary pixels and the presence of speckle during segmentation, respectively. Lower segmentation accuracy is obtained when the FWCRF algorithm is executed for the 2-look of SI2. NSFCM, SpBED, and the proposed FVSGLR algorithm have good stability with slight fluctuation. In addition, Table II shows that proposed FVSGLR's segmentation accuracy and Kappa are the highest.

The experimental results of each algorithm on the 2-look of SI2 are shown in Fig. 10.

As shown in Fig. 10(d), FWCRF has a large number of noises, so this algorithm has the lowest accuracy among the comparison algorithms. As shown in Fig. 10(g), the NSFCM algorithm does not retain edge information due to the smoothing operation, which finally leads to the wrong classification of edge pixels. As shown in Fig. 10(h), SpBED can segment the synthetic image accurately. Still, due to the drawback of K-means' tendency to produce local optima, it leads to individual small patches of

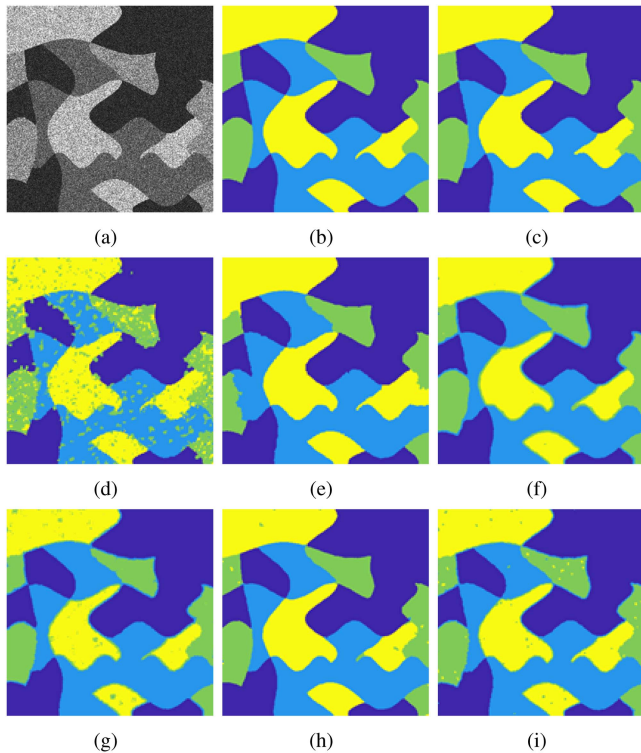


Fig. 10. Segmentation results on 2-look SI2 synthetic image. (a) 2-look SI2. (b) Ground truth. (c) Proposed FVSGLR. (d) FWCRF. (e) SFFCM. (f) ILKFCM. (g) NSFCM. (h) SpBED. (i) Gamma-FCM.

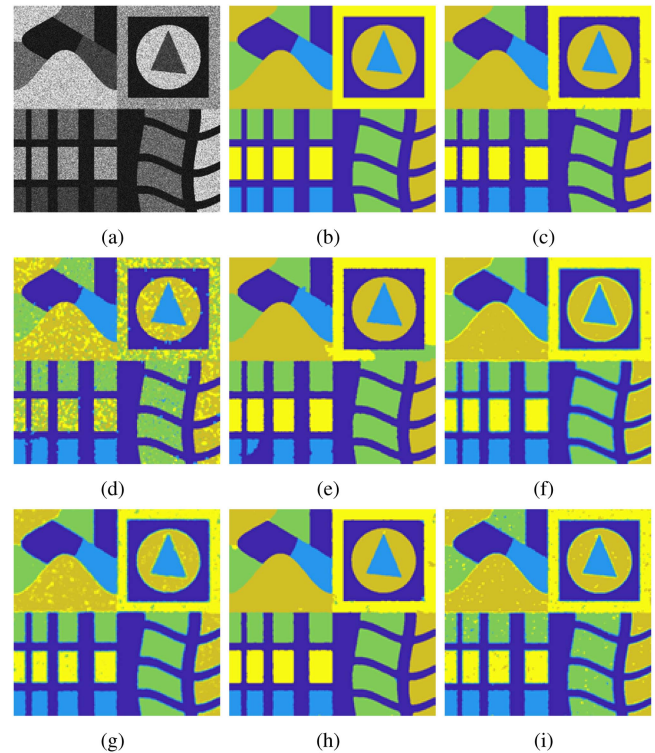


Fig. 11. Segmentation results on 2-look SI3 synthetic image. (a) 2-look SI3. (b) Ground truth. (c) Proposed FVSGLR. (d) FWCRF. (e) SFFCM. (f) ILKFCM. (g) NSFCM. (h) SpBED. (i) Gamma-FCM.

TABLE III
EXPERIMENTAL RESULTS ON SI2 SYNTHETIC SAR IMAGES

Algorithm	2-look		4-look		6-look	
	SA(%)	Kappa	SA(%)	Kappa	SA(%)	Kappa
FWCRF [35]	90.74	0.872	94.77	0.927	97.96	0.972
SFFCM [31]	95.61	0.939	97.77	0.969	98.91	0.985
ILKFCM [26]	94.35	0.922	94.63	0.926	94.59	0.926
NSFCM [25]	93.54	0.911	94.25	0.920	94.77	0.928
SpBED [21]	98.77	0.983	99.00	0.986	99.09	0.987
Gamma-FCM [27]	95.61	0.940	95.70	0.941	95.83	0.943
Proposed FVSGLR	99.25	0.990	99.36	0.991	99.42	0.992

homogeneous regions in Fig. 10(h). Compared with NSFCM, the proposed FVSGLR algorithm in Fig. 10(c) has no noise in the homogeneous region, and the boundary segmentation is accurate, indicating the effectiveness of this algorithm's segmentation.

The synthetic image of each SI3 look is used as the segmentation image. The SA and Kappa are calculated by the algorithm of this article and six comparison algorithms, as shown in Table III.

From Table III, as the number of looks of SI3 decreases, some comparison algorithms can no longer guarantee segmentation results above 90%, such as the ILKFCM and NSFCM algorithms. The segmentation accuracy of ILKFCM and NSFCM algorithms are low due to the inability to accurately classify the boundary pixels, and the presence of speckle during segmentation. With the increase of noise, FWCRF has many noises in the 2-look of SI3, and the algorithm is not robust. SpBED and the proposed FVSGLR algorithm are not volatile and have good stability.

Furthermore, the proposed FVSGLR algorithm can reduce the problem of pixel group segmentation errors and obtains higher segmentation accuracy.

The experimental results of each algorithm on the 2-look of SI3 are shown in Fig. 11.

Five classes are represented in Fig. 11(b). The difference of the adjacent pixels is minor, and the image's size is 512×512 . By observing Fig. 11(g) and (i), it is found that NSFCM and Gamma-FCM have noises, resulting in poor visual segmentation appearance of the image. Fig. 11(f) shows that ILKFCM cannot segment the boundary accurately. As shown in Fig. 11(e), the SFFCM lacks the use of boundary information, resulting in the fusion of large areas with low difference values. As shown in Fig. 11(h), SpBED can accurately classify boundary pixels, but due to the use of the SLIC algorithm to generate superpixels, there are some superpixel segmentation errors in segmentation process, resulting in the final clustering error, such as the area with a small block classification error at the boundary edge. Compared to ILKFCM, the boundary segmentation of the proposed FVSGLR algorithm is smooth, indicating the effectiveness of the algorithm segmentation.

2) *Results and Analysis of Real Images*: Using three real SAR images, the SA and Kappa are calculated by the proposed FVSGLR and comparison algorithms, as shown in Table IV.

From Table IV, due to the two relatively noisy images of Maricopa and Xian in the real SAR image, the difference of segmentation accuracy of each comparison algorithm is huge. In the Maricopa image, the Gamma-FCM algorithm is different

TABLE IV
EXPERIMENTAL RESULTS ON SI3 SYNTHETIC SAR IMAGES

Algorithm	2-look		4-look		6-look	
	SA(%)	Kappa	SA(%)	Kappa	SA(%)	Kappa
FWCRF [35]	80.97	0.742	92.94	0.907	96.35	0.952
SFFCM [31]	95.97	0.947	98.28	0.977	98.42	0.979
ILKFCM [26]	88.67	0.853	88.89	0.856	89.09	0.859
NSFCM [25]	88.71	0.852	91.01	0.882	92.22	0.898
SpBED [21]	98.56	0.981	98.87	0.985	98.99	0.987
Gamma-FCM [27]	90.55	0.877	91.39	0.888	91.39	0.888
Proposed FVSGLR	99.12	0.988	99.32	0.991	99.41	0.992

TABLE V
EXPERIMENTAL RESULTS ON REAL SAR IMAGES

Algorithm	Noerdlinger		Maricopa		Xian	
	SA(%)	Kappa	SA(%)	Kappa	SA(%)	Kappa
FWCRF [35]	77.16	0.637	73.88	0.629	67.08	0.534
SFFCM [31]	87.81	0.806	80.82	0.722	60.74	0.490
ILKFCM [26]	87.79	0.814	76.45	0.666	67.16	0.548
NSFCM [25]	87.63	0.805	79.66	0.707	67.04	0.558
SpBED [21]	95.93	0.936	88.51	0.883	83.26	0.761
Gamma-FCM [27]	88.45	0.821	80.79	0.728	78.32	0.689
Proposed FVSGLR	97.02	0.953	91.73	0.880	84.52	0.779

from the proposed FVSGLR algorithm because the Gamma-FCM algorithm only considers the affiliation of each point and does not consider the local information of each pixel point. After generating superpixels, the proposed FVSGLR uses local information to revise the pixel labels, which can better suppress the influence of speckle. Due to the irregular shape and high noise of Xian SAR image, FWCRF, SFFCM, ILKFCM, and NSFCM can not segment the image well, which indicates that these algorithms have weak ability to segment the image with high noise and irregular shape. For the Noerdlinger image, the accuracy of all algorithms is higher except for the FWCRF algorithm. Compared with the Maricopa and Xian SAR images, the Noerdlinger image is easier to segment because of its more regular shape. By observing the two evaluation indexes of SA and Kappa, the proposed FVSGLR algorithm achieves a better segmentation effect on the real SAR image segmentation.

The experimental results of each algorithm on the real SAR image Noerdlinger are shown in Fig. 12.

As shown in Fig. 12(d), FWCRF can maintain the boundary information, but there are spots in the figure, such as green spots in the yellow area. Fig. 12(e) shows that SFFCM lacks boundary information, and large tracts of farmland are segmented incorrectly. The green farmland is wrongly segmented into dark blue farmland. As shown in Fig. 12(g), the edge in the NSFCM segmentation result is smooth. Still, when smoothing the image, the edge information is not retained, and the outer layer of the boundary is incorrectly classified. The outer edge of each farmland is wrapped by a layer of blue pixels. Fig. 12(f) shows that ILKFCM has pixel group classification errors in the homogeneous region. Fig. 12(h) shows that the SpBED algorithm uses boundary information to segment the image and classify the boundary pixels accurately. There are green roads in yellow farmland. Compared with the SFFCM algorithm, the proposed FVSGLR has a smooth boundary segmentation, indicating the effectiveness of the algorithm.

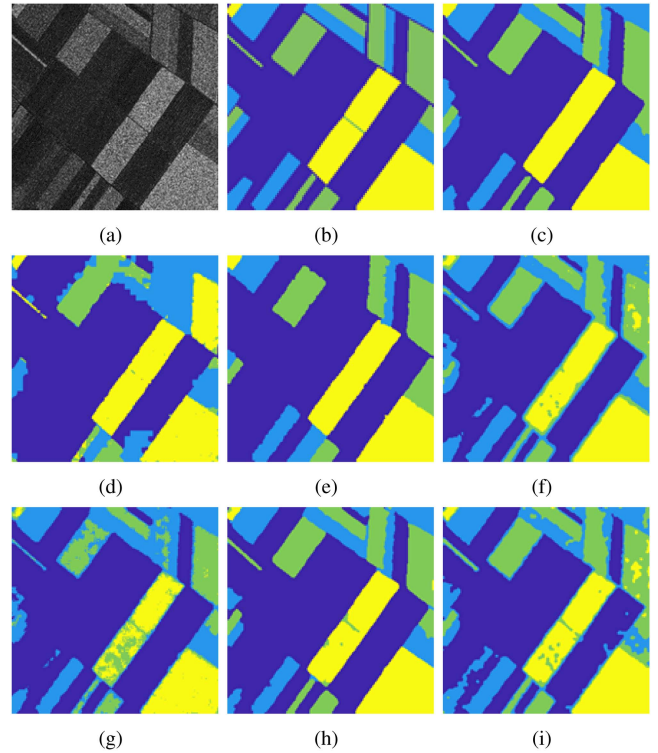


Fig. 12. Segmentation results on the Noerdlinger image. (a) Noerdlinger image. (b) Ground truth. (c) Proposed FVSGLR. (d) FWCRF. (e) SFFCM. (f) ILKFCM. (g) NSFCM. (h) SpBED. (i) Gamma-FCM.

The experimental results of each algorithm on the real SAR image Maricopa are shown in Fig. 13.

As shown in Fig. 13(g), the edges in the NSFCM map are often covered with a layer of blue pixel clusters, mainly because there is no edge information involved. Fig. 13(e) shows that the SFFCM can segment accurately within the homogeneous region without speckle effect. Fig. 13(f) shows that ILKFCM can segment individual targets. Still, there are often pixel group classification errors in homogeneous regions, such as dark blue pixel blocks appearing in the lower light blue pixel groups. As shown in Fig. 13(h), the SpBED boundary segmentation is accurate, but there are often small blocks of misclassified areas in the boundary edges and homogeneous regions. Compared with the SFFCM and SpBED, the proposed FVSGLR in Fig. 13(c) has no misclassified pixel groups in the homogeneous region and keeps the boundary information, indicating the effectiveness of the algorithm.

The experimental results of each algorithm on the real SAR image Xian are shown in Fig. 14.

Fig. 14(f) shows that ILKFCM has pixel group classification errors in homogeneous regions and more errors in target boundary segmentation, indicating that the algorithm cannot segment perfectly for smaller regions. As shown in Fig. 14(g) and (i), NSFCM and Gamma-FCM can maintain the boundary information but often misclassify in regions with low contrast of gray values. As shown in Fig. 14(e), SFFCM cannot use the boundary information to segment small areas of the whole image. Although there is no noise in homogeneous areas, large

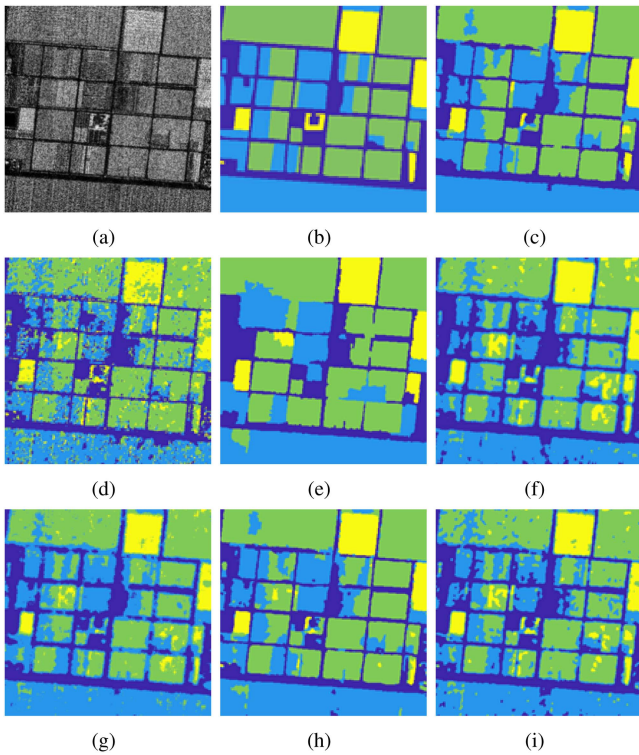


Fig. 13. Segmentation results on the Maricopa image. (a) Maricopa image. (b) Ground truth. (c) Proposed FVSGLR. (d) FWCRF. (e) SFFCM. (f) ILKFCM. (g) NSFCM. (h) SpBED. (i) Gamma-FCM.

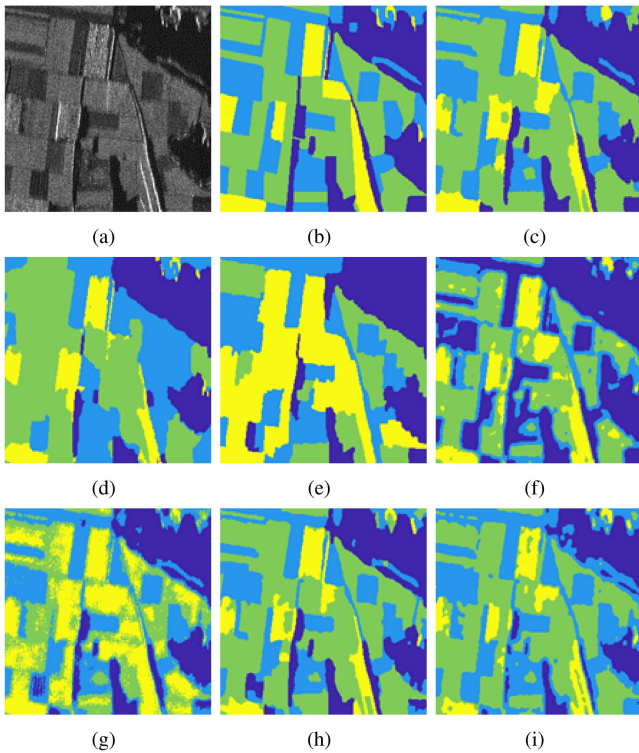


Fig. 14. Segmentation results on the Xian image. (a) Xian image. (b) Ground truth. (c) Proposed FVSGLR. (d) FWCRF. (e) SFFCM. (f) ILKFCM. (g) NSFCM. (h) SpBED. (i) Gamma-FCM.

TABLE VI
RUNNING TIME ON REAL SAR IMAGES

Algorithm	Noerdlinger	Maricopa	Xian
	Time(s)	Time(s)	Time(s)
FWCRF [35]	5.59	9.3	7.5
SFFCM [31]	0.34	0.56	0.4
ILKFCM [26]	83.66	165.57	86.71
NSFCM [25]	21.7	11.46	16.29
SpBED [21]	4.7	9.16	4.92
Gamma-FCM [27]	7.74	14.66	7.69
Proposed FVSGLR	5.90	11.77	6.12

areas are incorrectly segmented into green pixel blocks. As shown in Fig. 14(d), after FWCRF uses the smoothing method, it makes the values of adjacent regions smooth each other, and regions with low values of gray value differences are incorrectly confused for classification, leading to significant errors in the results. Fig. 14(h) shows that SpBED has irregular pixel groups included in the image, which decreases the accuracy, such as the blue region containing green pixel groups. By comparing the segmentation results of the proposed FVSGLR algorithm against six state-of-the-art comparison algorithms, the proposed FVSGLR also has some erroneous pixel groups being segmented, but there is no overall noise.

F. Comparison of Running Time

To verify the effectiveness of the proposed FVSGLR, running times of six algorithms are compared. In this section, the real SAR image is taken as the experimental object. The running time of each algorithm for each real image is shown in Table V.

It can be seen from Table V that SFFCM has the minimum running time. The superpixels are obtained through multiscale morphological gradient reconstruction (MMGR) operation and Watershed Transform. The SFFCM algorithm uses histogram parameters to perform FCM clustering on superpixel images. Since each pixel is not discriminatively classified, the running time is significantly reduced. The reason for the long running time of ILKFCM is that the fuzzy factor needs to be calculated in each iteration, and the calculation of the kernel distance of the wavelet features also increases the running time. In addition, the proposed FVSGLR algorithm runs almost the same time as other algorithms, which proves that the running time of the proposed FVSGLR algorithm is within a reasonable range.

IV. CONCLUSION

To overcome the problem of low gray contrast regions in SAR images, this article proposes a segmentation method based on FVSGLR. In this article, the Fisher vector is obtained by deriving the parameter set in a Gaussian mixture model, and then the Fisher vector is introduced into the distance similarity formula in SLIC. This similarity metric can effectively segment adjacent regions with similar gray values by introducing third-order information. Next, K-means is used to cluster the segmented superpixels, and fixed window label revision is used to revise the noise in the K-means cluster. Finally, to find isolated pixel

blocks, a region growth algorithm is used. The isolated pixel blocks that meet the fusion condition are incorporated into the neighborhood pixel blocks, which improves the segmentation accuracy of the proposed FVSLR algorithm. For performance evaluation, three sets of synthetic images of different sizes, and three real SAR images are used, and six recent state-of-the-art segmentation algorithms are compared against our proposed FVSLR algorithm. By comparing the segmentation accuracy, Kappa coefficients, and visually intuitive comparison results, the proposed FVSLR demonstrates improved accuracy and ability to preserve edge information. On the Maricopa dataset, the accuracy of the proposed FVSLR algorithm reaches over 91%, and the corresponding Kappa coefficients are significantly higher than other comparison algorithms. The proposed FVSLR algorithm can effectively segment adjacent pixel blocks with similar gray values by using edge information. For the noise points in the image, the label can be revised by using the boundary information and gray value. However, the label revision algorithm may ignore the details of the complex texture image when correcting the label. In the future, we will study how to improve the label revision algorithm to apply to complex texture images.

REFERENCES

- [1] R. Shang, C. Chen, G. Wang, L. Jiao, M. A. Okoth, and R. Stolkin, "A thumbnail-based hierarchical fuzzy clustering algorithm for SAR image segmentation," *Signal Process.*, vol. 171, Jun. 2020, Art. no. 107518.
- [2] X. Lu, Y. Yuan, and X. Zheng, "Joint dictionary learning for multispectral change detection," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 884–897, Apr. 2017.
- [3] Z. Michalewicz and M. Michalewicz, "Machine intelligence, adaptive business intelligence, and natural intelligence," *IEEE Comput. Intell. Mag.*, vol. 3, no. 1, pp. 54–63, Feb. 2008.
- [4] A. Zhang, X. Yang, S. Fang, and J. Ai, "Region level SAR image classification using deep features and spatial constraints," *ISPRS J. Photogramm. Remote Sens.*, vol. 163, pp. 36–48, May 2020.
- [5] R. Shang et al., "SAR image segmentation based on constrained smoothing and hierarchical label correction," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5102216.
- [6] J. Gu, L. Jiao, S. Yang, F. Liu, B. Hou, and Z. Zhao, "A multi-Kernel joint sparse graph for SAR image segmentation," *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 9, no. 3, pp. 1265–1285, Mar. 2016.
- [7] J. S. Lee and I. Jurkevich, "Segmentation of SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 27, no. 6, pp. 674–680, Nov. 1989.
- [8] Q. Wei and D. Feng, "Antistretch edge detector for SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1382–1386, Sep. 2018.
- [9] L. Jia, M. Li, P. Zhang, Y. Wu, and H. Zhu, "SAR image change detection based on multiple Kernel k-means clustering with local-neighborhood information," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 6, pp. 856–860, Jun. 2016.
- [10] H. Jing, X. Sun, Z. Wang, K. Chen, W. Diao, and K. Fu, "Fine building segmentation in high-resolution SAR images via selective pyramid dilated network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6608–6623, Apr. 2021.
- [11] R. Shang, J. Wang, L. Jiao, R. Stolkin, B. Hou, and Y. Li, "SAR targets classification based on deep memory convolution neural networks and transfer parameters," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 8, pp. 2834–2846, Aug. 2018.
- [12] N. Davari, G. Akbarizadeh, and E. Mashhour, "Intelligent diagnosis of incipient fault in power distribution lines based on corona detection in UV-visible videos," *IEEE Trans. Power Del.*, vol. 36, no. 6, pp. 3640–3648, Dec. 2021.
- [13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [14] M. Zalpour, G. Akbarizadeh, and N. Alaei-Sheini, "A new approach for oil tank detection using deep learning features with control false alarm rate in high-resolution satellite imagery," *Int. J. Remote Sens.*, vol. 41, no. 6, pp. 2239–2262, Mar. 2020.
- [15] J. Geng, X. Deng, X. Ma, and W. Jiang, "Transfer learning for SAR image classification via deep joint distribution adaptation networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5377–5392, Aug. 2020.
- [16] J. Feng et al., "Deep reinforcement learning for semisupervised hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–19, 2022.
- [17] P. Shui and D. Cheng, "Edge detector of SAR images using Gaussian-gamma-shaped bi-windows," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 5, pp. 846–850, Sep. 2012.
- [18] S. Ganugapati and C. Moloney, "A ratio edge detector for speckled images based on maximum strength edge pruning," in *Proc. Int. Conf. Image Process.*, Oct. 1995, vol. 2, pp. 165–168.
- [19] D. Xiang et al., "Fast pixel-superpixel region merging for SAR image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9319–9335, Nov. 2021.
- [20] W. Jing, T. Jin, and D. Xiang, "SAR image edge detection with recurrent guidance filter," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 6, pp. 1064–1068, Jun. 2021.
- [21] R. Shang, J. Lin, L. Jiao, X. Yang, and Y. Li, "Superpixel boundary-based edge description algorithm for SAR image segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1972–1985, May 2020.
- [22] M. Gong, L. Su, M. Jia, and W. Chen, "Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 1, pp. 98–109, Feb. 2014.
- [23] L. Szilágyi, Z. Benyo, S. M. Szilágyi, and H. Adam, "MR brain image segmentation using an enhanced fuzzy c-means algorithm," in *Proc. 25th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2003, vol. 1, pp. 724–726.
- [24] W. Jing, T. Jin, and D. Xiang, "Fast Superpixel-based clustering algorithm for SAR image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, Jan. 2022.
- [25] J. Ji and K. Wang, "A robust nonlocal fuzzy clustering algorithm with between-cluster separation measure for SAR image segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 12, pp. 4929–4936, Dec. 2014.
- [26] D. Xiang, T. Tang, C. Hu, Y. Li, and Y. Su, "A Kernel clustering algorithm with fuzzy factor: Application to SAR image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 7, pp. 1290–1294, Jul. 2014.
- [27] X. Zhao, H. Wang, J. Wu, Z. Peng, and X. Li, "A gamma distribution-based fuzzy clustering approach for large area SAR image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 11, pp. 1986–1990, Nov. 2021.
- [28] G. Akbarizadeh, "A new statistical-based kurtosis wavelet energy feature for texture recognition of SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4358–4368, Nov. 2012.
- [29] Z. Tirandaz and G. Akbarizadeh, "A two-phase algorithm based on kurtosis curvelet energy and unsupervised spectral regression for segmentation of SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 3, pp. 1244–1264, Mar. 2016.
- [30] H. Zou, X. Qin, H. Kang, S. Zhou, and K. Ji, "A pdf-based slic superpixel algorithm for SAR images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 6823–6826.
- [31] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng, and A. K. Nandi, "Superpixel-based fast fuzzy c-means clustering for color image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 9, pp. 1753–1766, Sep. 2018.
- [32] W. Jing, T. Jin, and D. Xiang, "Content-sensitive superpixel generation for SAR images with edge penalty and contraction–expansion search strategy," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5210715.
- [33] X. Wang, G. Li, X. Zhang, and Y. He, "Ship detection in SAR images via local contrast of fisher vectors," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6467–6479, Sep. 2020.
- [34] Z. Tirandaz, G. Akbarizadeh, and H. Kaabi, "Polsar image segmentation based on feature extraction and data compression using weighted neighborhood filter bank and hidden Markov random field-expectation maximization," *Measurement*, vol. 153, Mar. 2020, Art. no. 107432.
- [35] R. Ghaffari, M. Golpardaz, M. S. Helfroush, and H. Danyali, "A fast, weighted CRF algorithm based on a two-step superpixel generation for SAR image segmentation," *Int. J. Remote Sens.*, vol. 41, no. 9, pp. 3535–3557, 2020.



Ronghua Shang (Senior Member, IEEE) received the B.S. degree in information and computation science and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2003 and 2008, respectively.

She is currently a Professor with Xidian University. Her current research interests include evolutionary computation, image processing, and data mining.



Jiansheng Chen received the B.E. degree in computer science and technology from the School of Information Science and Engineering, Henan University of Technology, Henan, China, in 2020. He is currently working toward the M.S. degree in electronic information with the School of Artificial Intelligence, Xidian University, Xi'an, China.

His research interests include image processing and deep learning.



Jie Feng (Member, IEEE) received the B.S. degree in electronic information engineering from Chang'an University, Xi'an, China, in 2008, and the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 2014.

She is currently an Associate Professor with the Laboratory of Intelligent Perception and Image Understanding, Xidian University. Her current interests include remote sensing image processing, deep learning, and machine learning.



Yangyang Li (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science and technology, and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2001, 2004, and 2007, respectively.

She is currently a Professor with the School of Artificial Intelligence, Xidian University. Her research interests include quantum-inspired evolutionary computation, artificial immune systems, and deep learning.



Licheng Jiao (Fellow, IEEE) received the B.S. degree in electronic engineering from Shanghai Jiaotong University, Shanghai, China, in 1982, the M.S. and Ph.D. degrees in electronic engineering from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

From 1990 to 1991, he was a Postdoctoral Fellow with the National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an, China, where he has been a Professor with the School of Electronic Engineering since 1992. He is currently the Director with the Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University. He is in charge of about 40 important scientific research projects, and authored/coauthored more than 20 monographs and 100 papers in international journals and conferences. His research interests include image processing, natural computation, machine learning, and intelligent information processing.

Dr. Jiao is a Member of the IEEE Xi'an Section Execution Committee and the Chairman of awards and recognition committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, the Councilor of the Chinese Institute of Electronics, the Committee Member of the Chinese Committee of Neural Networks, and an expert of academic degrees committee of the state council.



Rustam Stolkin (Member, IEEE) received the M.Eng. degree in engineering science from the University of Oxford, Oxford, U.K., in 1998, and the Ph.D. degree in computer vision from University College London, London, U.K., in 2004.

His research interests include computer vision and image processing, machine learning and AI, robotic grasping and manipulation, and human-robot interaction.

Dr. Stolkin is the Chair of Robotics with the University of Birmingham, U.K.; Royal Society Industry Fellow; Chair of the Expert Group on Robotic and Remote Systems for the OECD's Nuclear Energy Agency; and founded the U.K. National Centre for Nuclear Robotics in 2017.