

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220240003>

# A New Region-Based Active Contour Model with Skewness Wavelet Energy for Segmentation of SAR Images

Article in IEICE Transactions on Information and Systems · July 2010

DOI: 10.1587/transinf.E93.D.1690 · Source: DBLP

CITATIONS

2

READS

248

3 authors, including:



**Gholamreza Akbarizadeh**

Shahid Chamran University of Ahvaz

93 PUBLICATIONS 1,457 CITATIONS

[SEE PROFILE](#)



**Shahriar B. Shokouhi**

Iran University of Science and Technology

102 PUBLICATIONS 689 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



paper work 1 Iris Recognition System based on Canny and LoG Edge Detection Methods [View project](#)



A new unsupervised method for feature selection and learning of fused data [View project](#)

# A Two-Phase Algorithm Based on Kurtosis Curvelet Energy and Unsupervised Spectral Regression for Segmentation of SAR Images

Zeinab Tirandaz and Gholamreza Akbarizadeh

**Abstract**—Texture-based segmentation of synthetic aperture radar (SAR) image is a difficult task in remote sensing applications because it must address the problem of speckle noise. Several methods have been proposed for this purpose based on clustering, but suffer from long run times, computational complexity, and high-memory consumption. The proposed technique consists of two phases for SAR image segmentation. A new algorithm for parameter estimation based on curvelet coefficient energy (KCE) to design an optimum kernel function and an unsupervised spectral regression (USR) method have been proposed in phases 1 and 2, respectively, for SAR image segmentation. Eigen-decomposition is not required in USR, which decreases run times over other methods. The proposed algorithm uses a single-stage curvelet to extract the texture feature. Then, a new term is introduced based on the kurtosis feature value of the curvelet coefficients energy of the SAR image. Finally, the level set method is used to outline the boundaries between textures. Subimages are then extracted from the textures. After the Gabor filter is applied and the features are extracted, they are learnt using USR and clustered using a  $k$ -means algorithm. It is demonstrated that the clustering results based on the learned features will be improved significantly. SAR image segmentation is performed using the  $k$ -means after applying the Gabor filter bank and feature extraction. The results of segmentation are compared with Nyström and parallel sparse spectral clustering (PSSC). The proposed method was shown to be more accurate and had a shorter run time than either Nyström or PSSC.

**Index Terms**—Clustering, Gabor filter bank, kurtosis curvelet energy, synthetic aperture radar (SAR), texture extraction, unsupervised spectral regression (USR).

## I. INTRODUCTION

IN APPLICATIONS such as environmental surveillance systems, target recognition and tracking, mapping of land, underground resources, and development of military systems where an imaging system is required, these systems must create images with high resolution over wide areas. It is also required to provide imaging under adverse climatic conditions.

Also, it is needed to provide high-quality imaging in both night and day. Synthetic aperture radar (SAR) can meet all

these needs. SAR allots unique radar frequencies for reflection of waves from different targets. Microwave radiation is used at a frequency range of 1–10 GHz. The atmosphere is transparent to these frequencies, cloud cover is permeable, and it does not influence imaging. SAR systems are active systems, providing its own source of illumination. Hence, despite optic systems, it does not need sunlight. Therefore, there is no limitation on time of day and environmental conditions for SAR imaging at microwave frequencies [1]–[3].

Applications for SAR images include those in resources, environment, archeology, military, and remote sensing. SAR imaging systems operate under all environmental conditions throughout the day at high spatial resolution. The main disadvantage of SAR is that processing the images is difficult and they cannot be processed using common methods [4]. This drawback hinges upon a condition known as speckle noise, which exists in all SAR images [5]–[8]. Speckle noise is multiplicative noise with a standard deviation equal to the reflectivity of the image pixels across the area of the SAR image. Speckle noise corrupts intensity of the pixels. Therefore, determining accurate objects' boundaries is difficult in SAR images.

Texture is an early sign for recognition in humans. Texture-based segmentation in SAR images plays an important role in analyzing and recognizing texture. This method allows recognition of areas with identical textural features and specifies the boundaries between areas with different textural features. Complexity and interrelation are major problems in texture-based segmentation. Computed features capable of extracting significant physical specifications of images can resolve such problems. Contrary to feature extraction in optical images, this is a difficult task in SAR images due to the influence of the speckle noise. The best features of the images should first be extracted [9]; thus, feature extraction in the SAR images should be performed carefully.

Akbarizadeh [1] used statistical-based kurtosis wavelet energy for feature extraction in texture recognition of SAR images. The technique detects the boundaries of the texture using wavelet transform coefficients energy and fourth-order cumulant of these coefficients. This method improves the accuracy in segmentation when applied to SAR images without removing speckle noise, but it cannot be applied to the SAR image until the last stage. This generates high computational complexity and texture characterization requiring a large number of coefficients. Wavelet transform is used as much as possible, depending on the size of SAR image; as the size of the

Manuscript received September 03, 2014; revised February 06, 2015, June 03, 2015, and August 19, 2015; accepted October 06, 2015. Date of publication November 10, 2015; date of current version February 22, 2016. This work was supported by the Shahid Chamran University, Ahvaz, Iran, as an M.Sc. thesis under Grant 94/3/02/31579. (Corresponding author: Gholamreza Akbarizadeh.)

The authors are with the Department of Electrical Engineering, Faculty of Engineering, Shahid Chamran University, Ahvaz 61357-83151, Iran (e-mail: g.akbari@scu.ac.ir).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTARS.2015.2492552

image increases, the number of stages and coefficients required to define texture increase [1]. The proposed wavelet-based method is used on high-resolution and large-size SAR images. This means that it requires a large number of coefficients to define texture.

Average [10], median, and order statistics have been developed to remove speckle noise. Wavelet transforms have been used extensively for filtering in the transform area [11], [12]. For a one-dimensional (1-D) signal, wavelet transform uses an optimal display of piecewise smooth signals with unique parts; however, for two-dimensional (2-D) signals such as images, the use of the wavelet transform to eliminate noise is not suitable because it changes the geometry of the objects [13]. Although wavelet transform can be implemented to a range of image processing problems, its drawbacks include random edges and curves that are unsuitable for examining singular points on edges. This has prompted the development of curvelet and ridgelet transforms [12].

Donoho and Candès [14], [15] proposed a curvelet transform as a geometric transform. This transform is better in displaying signals with unique curves and excessive points at high dimensions. The first generation of curvelet transform was used to remove speckle noise where the edges of the image are represented as direct lines; however, the edges of SAR images are curved [16]. Second-generation curvelet transforms can represent curve-like edges, making the coefficients of this transform useful for SAR image edges.

In this paper, with respect to advantages of curvelet over wavelet mentioned earlier, some incentives were created for placing curvelet coefficients in the proposed kernel in [1]. This substitution of curvelet coefficients for wavelets in phase I allows better extraction of the features, which improves segmentation, and considerably decreases the computational complexity. This phase uses a parameter estimation algorithm based on curvelet coefficient energy (KCE) to design an optimum kernel function. The curvelet transform is applied in only one step to the SAR image and internal layer coefficients. These coefficients are extracted as texture-describing features. A kernel function is then formed based on the KCE. Finally, the boundaries of the layers are determined using the estimated function of the KCE.

Phase II uses multichannel filtering to extract textural features. Multichannel filtering is appropriate for texture analysis because the spatial-frequency components of different textures differ. An advantage of multichannel filtering is that it allows decomposition of an input image into many filtered images with textural data and then it extracts features from them. A Gabor filter bank was used to describe the channels. An important feature of Gabor filters is their ability to use joint localization or resolution in both spatial and frequency domains simultaneously [17]. The parameters for the Gabor filter were selected carefully to provide maximum compatibility with the human visual system (HVS).

The extracted features must be learnt to improve the discrimination power and the performance of the algorithm. Some methods for learning features are capable of dimensionality reduction. This is used in information processing in machine learning, information retrieval, pattern recognition, machine vision, and target recognition. Principle

component analysis (PCA) [18], linear discriminant analysis (LDA), locality preserving projection (LPP), neighborhood preserving analysis, and marginal fisher analysis are among these approaches [18], [19].

PCA decreases the number of dimension in data from  $m$  to  $d$  ( $d \leq m$ ). It works in linear space but does not assure that the samples are in linear space. This has prompted the use of manifold learning approaches that can learn in all spaces. Isomap, Laplacian Eigenmap, and locally linear embedding (LLE) are common manifold learning approaches. Laplacian Eigenmap, LLE, and Isomap reduce dimensions so that a specific maximum interpoint relationship is preserved. The local geometry of the data is preserved in LLE and Laplacian Eigenmap, which maps nearby points of learning to near points in dimensionally reduced representation. Isomap is a global method that preserves the geometry of data at all scales. It maps nearby points and faraway points on a manifold to nearby points and faraway points in dimensionally reduced space, respectively. The major constraint in these methods is that they do not present a general valid mapping function between manifold space and dimensionally reduced space both ON and OFF the training data. These algorithms operate only for learning samples. Other approaches have been proposed to solve out-of-sample extension problems [18].

Bengio *et al.* [20] proposed a single framework for LLE, Isomap, and Laplacian extensions. It considers these algorithms to be eigen-functions for data-dependent kernel learning. This method is called Nyström formula. The Nyström formula [20], [21] creates a function for new data. The kernel function of the sample values must be calculated for all learning samples to obtain the results of embedding nonobserved samples, but this is not always possible. This problem can be addressed by embedding a linear function or by reproducing the kernel Hilbert space when it is minimized. In this paper, a linear function called linear expansion of graph embedding (LGE) was selected [18].

Methods for clustering include the  $k$ -means, Nyström, parallel-based spectral clustering, etc. Gou *et al.* [22] proposed the parallel sparse spectral clustering (PSSC) method that decreases dimensionality to decrease run time and computational complexity. The main problem with Nyström and PSSC is that they use eigen-decomposition, which requires costly calculations in terms of time and memory. This paper uses spectral regression (SR), a strong tool for dimension-reduction and learning, to decrease dimensions. This approach uses data embedded in the eigen-vector of the affinity matrix to reduce the number of dimensions. SR learns in a regression framework, which avoids the calculation of density matrices eigen-decomposition. Different types of regularizing can be used because regression is applied as a building block. For example, L1-norm regularized can be used to design sparse projection [18], [23]. SR can serve in supervised, unsupervised, and semisupervised methods. It can be also applied to unlabeled and labeled data to find the internal structure of data [18], [19]. SR was unsupervised in this paper. A combination of Gabor filter banks and unsupervised SR (USR) is proposed for optimal clustering in phase II.

The boundaries of different areas were specified using the method proposed in phase I. Subimages of these areas were

extracted so that each belongs to a single type of texture. Since a texture-based dataset has not been proposed for SAR images, the dataset was created for the relevant SAR image using the algorithm of phase I and the extracted subimages. The texture-based features were then extracted from each subimage after application of the Gabor filter bank [24]. Next, features were learnt by USR [18], [19], [25]–[31]. Finally,  $k$ -means was used for clustering and the clustering of the features was compared with that of nonlearned ones.

The algorithm provided a favorable response for the subimages, indicating its ability to cluster despite being affected by speckle noise and caused degradation and blurred the texture content of the image. It also provided an appropriate response for datasets of optical image texture. Clustering of unlearned and learned features was compared for different datasets of UCI and extracted subimages to study the effect of learning features on algorithm performance. In 1987, UCI Machine Learning Repository was created by Aha *et al.* at UC Irvine as an ftp archive. The dataset is a set encompassing data generators, domain theories, and databases, which are used for the experimental analysis of machine learning [32]. The results showed that learning features improved their accuracy, mutual information, and recognition performance, which led to better performance of the algorithm.

Image segmentation was also done by extracting features from the image, learning them and employing  $k$ -means. Segmentation was then compared with Nyström [21] and PSSC [22]. The experiment results were carried out using simulated and real SAR images. The performance of the proposed algorithm was also compared with other methods based on color, intensity, texture, and a combination of these features.

The results show that the proposed algorithm is an appropriate tool for segmentation and description of different textures on SAR images and produced lower segmentation error than other methods. Although the PSSC method is parallel and the proposed method is serial, PSSC has a higher computational complexity and longer running time in MATLAB.

This paper is organized as follows. Section II describes the Gabor filter bank and Section III describes the curvelet for feature extraction. Section IV explains the basic principles of USR. Section V discusses the two phases of the proposed approach. The extraction of features from the SAR image using KCE, clustering methodology, feature extraction, and USR for segmentation is defined. Section VI describes the results of experiments using the proposed algorithm on simulated and real SAR images and datasets. Section VII presents the conclusion.

## II. GABOR FILTER BANK

Malik and Perona [33] presented a texture perception-based model for the early stages of HVS. In these systems, individual symmetrical filters were used for modeling. Jain and Farrokhnia [17] proposed an efficient technique for texture segmentation inspired by multichannel. Once the image is processed using multiresolution, the filter banks can decompose the image into textural features. According to these textural features, images can be classified [9].

Multichannel filtering follows HSV features, which are complicated. The receiver cells behind the eyeball trigger the

electrical pulses through nervous pathways to the visual cortex. Various types of neurons provide the signals using different types of processing in this visual cortex. Hubel and Wiesel [34] showed that simple cells are sensitive to angles with bandwidths of approximately  $30^\circ$ . Campbell and Kulikowski [35] demonstrated that in addition to angles, humans are sensitive to spatial frequencies. The band width in simple cells is 1 octave [9].

Rao and Lohse [36] studied the angles and frequency sensitivity of human vision. They showed that people use repetition, directionality, and complexity to interpret texture. Repetition and directionality represent frequency and direction, respectively. Complexity depends on texture consistency. For example, sinusoidal texture has little complexity, while a texture that follows no certain pattern has higher complexity. These three characteristics should be considered when an algorithm is intended to determine texture [9].

Simplicity, local optimization in spatial or in frequency domains, and the ability to simulate 2-D behavior of simple cells in the visual cortex are key features that make the Gabor filter a good option for image processing. Simulation is implemented by separating frequencies and directions. Two main stages should be considered when extracting features using multichannel filters. The function, number, frequencies, orientation, and space of the filters should first be selected precisely. Next, features should be extracted from the raw filters to improve the set of features.

Several methods can be applied by Gabor filters to analyze texture. Bovik *et al.* proposed supervised methods to choose the filter locus that are all based on the power spectral density of the texture in question. For heavily directed textures, the important spectral peak along the dominant angular direction is used as the location of the filter. Choosing the initial lower frequency determines the periodic textures. Two larger maxima have been suggested for nondirected textures, but since multiple peaks are required for determination, this method is difficult [9].

Dunn and Higgins suggested a method for optimal selection of the filter parameters based on the texture samples identified. Their method is supervised and identifies the borders between textures using the least number of filters. A specific filter that separates two classes of texture optimally is used for segmentation. The optimal filter represents the outstanding textural features of a class that contradict the textural data for another class. The use of a single filter to determine a pair of textures may be difficult because each texture may be influenced by spatial variables or include several outstanding elements.

It is evident the preferred method is the one that can determine texture features automatically. A filter bank is used to determine the unique peaks belonging to a texture. These peaks include sparse filters for all spatial and frequency dimensions. In this case, relatively uniform coverage of spatial-frequency dimensions occurs, which avoids the problem of choosing only a central frequency-dependent texture.

In addition to the explanations on the advantages of Gabor filter bank, which is true for all images, the advantages of using Gabor filter bank are explained in texture and SAR images segmentation as follows.

- 1) As mentioned earlier, Gabor filter banks represent images in spatial and spatial-frequency optimally and simultaneously. They do this by extracting low-level



information of common uncertainty of these two dimensions. In this case, features of nonstationary texture specifications can be extracted [38], [39].

- 2) Selection of orientation in Gabor filters makes possible process of textures in different directions and enhances power of discrimination of various textures. Gabor filters encode texture images into several orientation channels and narrow frequencies. Importance of this matter is that each texture of the image is specified by a dedicated distinctive narrow frequency-spatial range [39].
- 3) Textural analysis of SAR images is generally difficult because textural images have the texels in assorted sizes. Gabor filter bank arrangement makes the multiresolution presentation possible. This way, image details can be achieved at different sizes [38].
- 4) Analysis using Gabor filter bank enables us to extract textural information as *local* and *global* from an image.
- 5) Gabor filters are capable to extract multispectral textural features in the presence of noise. They are resistant to noise considerably. These features have made them become efficient in analyzing and processing SAR images whose quality has been lowered by speckle noise [37], [39].

### III. CURVELET

Discrete wavelet transform (DWT) algorithms have been proposed for image analysis. Although there are many applications for conventional wavelets in all fields, their shift-variant is a weakness that can create problems in image analysis [40]. The transform works well only for point singularities, ignores the geometrical features of structures, and does not extract regularity or the sequence of the edges. Compression, structure extraction, and denoising are not appropriate for calculation of features of line and curve singularities [41].

In 1999, Candès and Donoho proposed an anisotropic wavelet transform called a ridgelet. Prior to the ridgelet, all transforms were 1-D and were transformed into 2-D through a set of data. A ridgelet, by contrast, is a natural 2-D transform. This transform acts optimally to represent direct singular lines. The best function of the ridgelet is achieved for an image that has uniformly straight and parallel transforms lines; however, this hypothesis is inadequate for remote sensing images. Moreover, direct singular lines rarely occur in actual applications, which is the most substantial weak point of this transformation method [41].

Candès and Donoho proposed a curvelet transform to extract edges of curves more effectively and broadly than wavelet transform in 2002. The process for representing images curves initially analyzes the image using separate and uncommon scales which is then analyzed by regional ridgelet transform. This is referred to as the first-generation curvelet. Both the ridgelet and curvelet contain spatial and frequency information, but they also provide angular data. Like transforms based on wavelets, they are multiscale features.

A curvelet is a directional multiscale transform. Multiscale transforms imitate the behavior of the brain's visual cortex to see images. Despite the advantages for the first generation,

this type of transform suffers several disadvantages. First, reconstruction of an image from coefficients requires several complicated stages, including estimation of curvelet parameters such as scale, location, and angle. Moreover, hyperbolic expansion will result in an asymmetrical relationship between length and width (width equals length squared). Second, in this spatial domain, transform zoning should be implemented. Using overlapping windows prevents the images appearing in mosaic form. Such segmentation in spatial domain increases the repetition factor. The increased repetition rate of high volume adds data complicity; moreover, since the ridgelet geometry is not completely clear, this method is limited.

Candès and Donoho next developed the curvelet into a relatively simple framework and transform, i.e., a second generation, based on frequency segmentation. In comparison with the first-generation curvelet, it was less redundant, simpler, and faster. The second generation of curvelet uses simple mathematical analysis to form a simpler and more natural representation of a three-component structure (location, angle, and scale). The second generation uses a very thin frame which significantly decreases the repetition factor. Unlike the first generation, the second generation does not use discrete ridgelet transform (DRT). They showed that the second-generation curvelet is an efficient tool for different applications. Mathematically, the power of the second generation is its ability to formulate strong theorems of approximation and operator theories. Curvelet transform is successful for extracting and representing curved-like edges; therefore, 2-D curvelet transform always makes possible sparse representation of an object with singularities and smooth curves [41]–[43].

### IV. UNSUPERVISED SPECTRAL REGRESSION

SR is based on spectral graph analysis and regression [44]. This method provides a graph whose *vertices* and *edges* are input data and two-by-two similarity of the vertices, respectively. A matrix is made of the two-by-two similarities, which is called affinity matrix. An affinity graph is embedded for the labeled and unlabeled points to learn their responses, i.e., responses are achieved for any labeled and unlabeled data to discover input data structure. When the responses are obtained, a projective function is achieved. Then, the normal regression is applied to learn the embedded function, i.e., regression is applied on the projective function. This method is made up of the following two major steps.

- 1) Graph analysis by which a projective function is generated;
- 2) Applying the regression on the generated projective function. Therefore, it is called SR. This method solves the problem of optimizing the linear graph and decreases calculation complexity.

LLP, Isomap, and Laplacian Eigenmap can be placed within a graph framework. The manifold learning algorithms encode the internal structural of the data in the graph weight matrix; optimization of the structural data approaches the same eigenproblem, and 1-D mapping is selected. The mapping function of  $x_i$  is  $y_i$  where  $i = 1, 2, \dots, n$  and a general graph is embedded [18], [19], [30].

Suppose that graph  $G$  has  $n$  vertices where each vertex represents a piece of data. The affinity matrix  $W$  is a  $n \times n$  matrix and  $W_{ij}$  is the weight of the common edge between the  $i$  and  $j$  vertices.  $W$  and  $G$  can be defined to determine the statistical or geometrical features of the dataset. Embedding provides representativeness of a reduced dimension vector of each vertex, which preserves the similarity between two vertices. Similarity is measured by edge weight. If  $y = [y_1, y_2, \dots, y_n]^T$  is the mapping of the graph vertices on a real line, the optimal value of  $y$  can be derived by minimizing the following equation [18], [19], [30]:

$$\sum_{i,j} (y_i - y_j)^2 \cdot W_{i,j}. \quad (1)$$

Under suitable conditions, if  $i$  and  $j$  are mapped with a large value for  $W_{ij}$ , serious compensation will be imposed. Minimizing (1) assures that, if  $i$  and  $j$  are close,  $y_i$  and  $y_j$  will also be close. The simple algebraic equation is [18], [19], [25]–[30]

$$\sum_{i,j} (y_i - y_j)^2 \cdot W_{i,j} = 2y^T Ly \quad (2)$$

where  $L$  is a Laplacian graph with  $L = D - W$ ,  $D$  is a diagonal matrix producing  $D_{ii} = \sum_j W_{ji}$ ,  $y = \{y_0, y_1, \dots, y_d\}$ ,  $d + 1$  is the eigen-vector for the maximum eigen-value  $\lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_d$  when  $\lambda_0 = 1$ , and  $y_0$  is a vector in which all elements equal 1. To obtain such a vector, the eigen-values should first be obtained. Choose the largest  $d + 1$  eigen-values and then select the eigen-vector corresponding to each eigen-value. Under these conditions, (2) is [18], [30]

$$y^* = \arg \min_{y^T Dy = 1} y^T Ly = \arg \min \frac{y^T Ly}{y^T Dy} = \arg \max \frac{y^T Wy}{y^T Dy} \quad (3)$$

and  $y^T Dy = 1$  eliminates the voluntary scaling factor. Three manifold learning algorithms are formed by selecting different values for  $W$  and  $D$ . Next, select a linear function for  $y_i = f(X_i) = a^T X_i$  with  $y_i = X^T a$  and  $X = [X_1, \dots, X_n] \in R^{m \times n}$ , (3) can be rewritten as [23]–[31]

$$a^* = \arg \max \frac{y^T Wy}{y^T Dy} = \arg \max \frac{a^T X W X^T a}{a^T X D X^T a}. \quad (4)$$

The optimal value of  $a$  is the eigen-vector for the maximum eigen-value of [18], [19], [25]–[31]

$$X W X^T a = \lambda X D X^T a. \quad (5)$$

This method is known as LGE and can be applied to LLE, Isomap, and Laplacian Eigenmap. By selecting different values for  $W$ , the LGE framework approaches well-known dimensionality reduction methods like LDA, LPP, and neighborhood preserving embedding (NPE).

According to Table I, by selecting different  $W$ 's, the dimensionality reduction methods that will be implemented are as follows.

High calculation costs limit the application of subspace learning algorithms, but SR has solved this problem [18], [19], [30]. The solution to the eigen-problem in (5) is

$$W y = \lambda D y. \quad (6)$$

*Theory 1:* If  $y$  is the eigen-vector of (2) with an eigen-value of  $\lambda$  and if  $y = X^T a$ , then  $a$  is the eigen-vector of the eigen-problem of (5) with the same eigen-value ( $\lambda$ ). Instead of using the eigen-problem in (5), the linear projective function can be derived in two steps [18], [19], [30]:

- 1) solving the eigen-problem of (2) for getting  $y$  [18], [19], [30];
- 2) obtain  $a$  that satisfies  $y = X^T a$  [18], [19], [30].

Under real conditions,  $a$  probably does not exist; therefore, the best solution is to find the value for  $a$ , which is most suitable for the least-squares problem [18], [19], [30]

$$a = \arg \min_a \sum_{i=1}^m (a^T X_i - y_i)^2 \quad (7)$$

where  $y_i$  is the  $i$ th element of  $y$ . If the number of samples is less than the number of features, the minimization of (7) is ill-posed and there may be an unlimited number of solutions. The most well-known solution is to apply a cost (compensation) to norm  $a$  [18], [19], [30]

$$a = \arg \min_a \left( \sum_{i=1}^m (a^T X_i - y_i)^2 + \alpha \|a\|^2 \right). \quad (8)$$

This method is called regularization and has been used frequently.  $\alpha \geq 0$  controls shrinkage. Equation (8) can be written as [18], [19], [30]

$$a = \arg \min_a \left( \sum_{i=1}^m (X^T a - y)^T (X^T a - y) + \alpha a^T a \right). \quad (9)$$

Equation (9) with respect to  $a$  [18], [19], [30] is

$$\begin{aligned} (X X^T + \alpha I) a &= X y \\ \Rightarrow a &= (X X^T + \alpha I)^{-1} X y. \end{aligned} \quad (10)$$

When  $\alpha > 0$ , regularization solutions do not satisfy minimum linear equation  $y = X^T a$ ,  $a$  is not the eigen-vector of the eigen-problem of (5) and (10) is the most accurate solution of the eigen-problem of (5), especially under the terms of theory 2 as follows:

*Theory 2:* If  $y$  is the eigen-vector of the eigen-problem in (6) and if  $y$  is limited by the row vectors of  $x$ , then the projective function of the value for  $a$  from (10) will be the eigen-vector of the eigen-problem of (5) until  $a$  decreases and approaches zero [18], [19], [30]. When the number of features is higher than the number of samples, the sample vectors are linearly independent and, e.g.,  $\text{rank}(X) = m$ . In this case, the stronger solution is shown in the following deduction [18], [19], [30].

*Deduction:* If the sample vectors are linearly independent and  $\text{rank}(X) = m$ , all projective functions calculated in (10) will be given vectors of the eigen-problem in (5) until  $a$  approaches zero [18], [19], [30].

## V. PROPOSED METHOD

*Phase I:* Fig. 1 is a schematic of the procedure used to determine the boundaries between textures (phase I) [45].

TABLE I  
DIFFERENT W'S AND THEIR CORRESPONDING DIMENSIONALITY REDUCTION METHODS

	NPE	LPP	LDA
Assumptions	We have $c$ classes. The $t$ th class has $m_t$ samples. $m_1 + \dots + m_c = m$	$N_k(x_i)$ is the $k$ th nearest neighbor of $x_i$	$N_k(x_i)$ is the $k$ th nearest neighbor of $x_i$ . $M$ is the $m \times m$ local reconstruction coefficient matrix. The $i$ th row of $M$ is $\begin{cases} M_{ij}=0, & \text{if } x_j \notin N_k(x_i) \\ \min \ x_i - \sum_{j \in N_k(x_i)} M_{ij} x_j\ ^2, & \sum_{j \in N_k(x_i)} M_{ij} = 1, \text{ otherwise} \end{cases}$
$W$	$\begin{cases} 1/m_t, & \text{if } x_i \text{ and } x_j \text{ belong to the same class (t-th class)} \\ 0, & \text{otherwise} \end{cases}$	$\begin{cases} e^{\ x_i - x_j\ ^2}, & \text{if } x_i \in N_k(x_j) \text{ or } x_j \in N_k(x_i) \\ 0, & \text{otherwise} \end{cases}$	$M + M^T - M^T M$

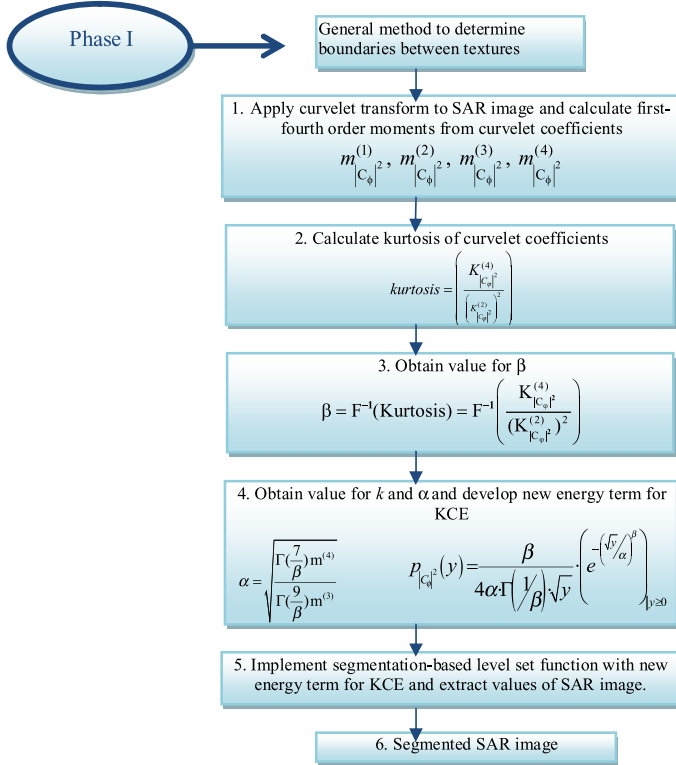


Fig. 1. Schematic of phase I of the proposed method for determining boundaries between different textures in SAR image.

#### A. Feature Extraction for Segmentation of SAR Image

The fourth-order cumulant of random variable ( $|C_\phi|^2$ ) as a function of moments is [1]

$$K_{|C_\phi|^2}^{(4)} = m_{|C_\phi|^2}^{(4)} - 4m_{|C_\phi|^2}^{(1)} \cdot m_{|C_\phi|^2}^{(3)} + 6\left(m_{|C_\phi|^2}^{(1)}\right)^2 \cdot m_{|C_\phi|^2}^{(2)} - 3\left(m_{|C_\phi|^2}^{(1)}\right)^4 \quad (11)$$

where  $m_{|C_\phi|^2}^{(i)}$  is the  $i$ th moment of random variable  $|C_\phi|^2$ . Normalization for higher order statistics is also applied.

The normalized fourth-order statistic (kurtosis) is defined as [1]

$$kurtosis = \left( \frac{K_{|C_\phi|^2}^{(4)}}{\left(K_{|C_\phi|^2}^{(2)}\right)^2} \right). \quad (12)$$

In (12), kurtosis is the fourth-order statistic that is normalized by the squared covariance. The kurtosis statistic denotes the rate for decreasing the slope of the probability density function curve for the histogram of a SAR image. In this study, application of the kurtosis normalized cumulant from the KCE is suggested instead of wavelets as a feature of texture for boundary detection of SAR images. This selection will provide more statistical information and will have better function.

#### B. Fast Discrete Curvelet Transform

The fast discrete curvelet transform from function  $f$  is

$$c(j, l, k) := \langle f, \varphi_{j,l,k} \rangle \quad (13)$$

where  $\varphi_{j,l,k}$  is the curvelet function;  $j$ ,  $l$ , and  $k$  are the parameters of scale, angle and location, respectively. The input for this transform is  $f[t_1, t_2]$ , ( $0 \leq t_1, t_2 \leq n$ ) in the Cartesian coordinate system and the discrete curvelet transform is [46]

$$c^D(j, l, k) := \sum_{0 \leq t_1, t_2 \leq n} f[t_1, t_2] \varphi_{j,l,k}^D[t_1, t_2]. \quad (14)$$

#### C. Display Texture of SAR Image Using KCE

To specify the texture at each area in a SAR image using KCE, consider the coefficients of each layer separately and calculate their energies to obtain a value for kurtosis. Since the coefficients of the innermost layer provide information for the low-frequency image and report an approximation of the main image, they will be larger than the other coefficients; thus, most of the main data for the image is reflected using these coefficients. So, only the energy of coefficients for the innermost layer has been calculated and the value of kurtosis is obtained.

The obtained value indicates a statistical feature that will be used to specify the texture. Empirical experiments on images

have shown that the features of texture in each image can be obtained using the distribution of curvelet coefficients in each sub-band. The distribution of the absolute value of these coefficients will be a generalized Gaussian as follows [47]:

$$p(C_\phi; \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|C_\phi|/\alpha)^\beta} \quad (15)$$

where  $\alpha$ ,  $\beta$ , and  $\Gamma(\cdot)$  are the variance, reduced rate of the generalized Gaussian, and the Gamma function, respectively. In fact,  $\alpha$  models the PDF peak width and it is called the scale parameter;  $\beta$  is inversely proportional to the peak reduction rate and it is the shape parameter; and  $\Gamma(\cdot)$  can be calculated as [47]

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt, \quad z > 0. \quad (16)$$

The energy distribution of the curvelet coefficients in each layer of the image follows a generalized Gaussian form as follows:

$$p_{|C_\phi|^2}(y) = \frac{\beta}{4\alpha\Gamma(1/\beta) \cdot \sqrt{y}} \cdot \left( e^{-(\sqrt{y}/\alpha)^\beta} \right)_{y \geq 0} \quad (17)$$

where  $\alpha$  and  $\beta$  are parameters for texture. As (17) is true for any layer individually, it can be said that this equation also expresses energy distribution of internal layer coefficients. The fourth-order moment from the energy distribution of curvelet coefficient  $m_{|C_\phi|^2}^{(4)}$  to define the features of texture based on kurtosis is

$$\begin{aligned} m_{|C_\phi|^2}^{(4)} &= E\{|C_\phi|^8\} = \int_0^\infty (C_\phi)^8 p(C_\phi; \alpha, \beta) d(C_\phi) \\ &= \frac{\alpha^8}{2\Gamma(\frac{1}{\beta})} \cdot \Gamma\left(\frac{9}{\beta}\right). \end{aligned} \quad (18)$$

Next,  $\alpha$  can be obtained as

$$\alpha = \sqrt{\frac{\Gamma(\frac{7}{\beta}) \times m_{|C_\phi|^2}^{(4)}}{\Gamma(\frac{9}{\beta}) \times m_{|C_\phi|^2}^{(3)}}}. \quad (19)$$

The formula for function  $F(\beta) = \text{kurtosis}(|C_\phi|^2)$  can be obtained by replacing the first- to fourth-order moments  $m_{|C_\phi|^2}^{(1)}$ ,  $m_{|C_\phi|^2}^{(2)}$ ,  $m_{|C_\phi|^2}^{(3)}$ , and  $m_{|C_\phi|^2}^{(4)}$  with their corresponding values as shown in (20) at the bottom of the page.

The value of segmentation parameter  $\beta$  can be obtained as

$$\beta = F^{-1}(\text{Kurtosis}) = F^{-1}\left(\frac{K_{|C_\phi|^2}^{(4)}}{(K_{|C_\phi|^2}^{(2)})^2}\right). \quad (21)$$

This study uses KCE to detect different areas in the SAR images. Equation (17) calculates the energy distribution for the curvelet coefficients with  $\alpha$  and  $\beta$ . If these two parameters are estimated properly, (17) will also be estimated properly and will be considered to be a kernel in the level set function.

It is evident that the value of  $\alpha$  in (19) is obtained from the third- and fourth-order moments of the curvelet coefficients energies. The challenge is to estimate the value of  $\beta$  more accurately. This is done using (21) by applying the kurtosis of the energy for the curvelet coefficient, which is a new feature of KCE. In other words, KCE is an estimator for segmentation parameter that can also be a feature of texture. In the experiments proposed in this study, KCE is a stronger estimator than those proposed to define texture in SAR images.

#### D. Calculate Formula of Level Set Function Using KCE

Active contour models (snakes) are based on curve evaluations for all general segmentation operations on images [48]–[50]. Some models are grounded based on the intensity of the pixels. Intensity inhomogeneity always occurs in SAR images and has not been addressed by existing methods. This feature is used in a level set function to apply segmentation to each area within a SAR image. Assume  $x \in r$  as a pixel and  $I(x) : r \rightarrow \mathbb{R}^1$  as a vector in a SAR image, where 1 is the dimension of vector  $I(x)$ . The function is defined as

$$\begin{aligned} F^{\text{KCE}}(L_1, L_2, \dots, L_k, f_1, f_2) \\ = \varepsilon_x^{\text{KCE}}(L_1, L_2, \dots, L_k, f_1, f_2) + \mu\rho(L). \end{aligned} \quad (22)$$

In (22),  $\mu$  is a fixed value,  $\varepsilon_x^{\text{KCE}}$  is the energy for KCE,  $f_1$  and  $f_2$  are functions that minimize  $\varepsilon_x^{\text{KCE}}$ ,  $\rho(L)$  is the deviation of the level set function, and  $L$  is a distance function. The subscripts 1,  $\dots$ ,  $k$  refer to each of the  $k$  disjoint regions in which the image is segmented. The value of kurtosis from the KCE has been suggested as the contour energy for  $\varepsilon_x^{\text{KCE}}$  in (22). It can be said that KCE energy in  $\varepsilon_x^{\text{KCE}}$  follows the form of kurtosis energy distribution associated with curvelet coefficients in each layer. KCE energy for each  $x \in r$  can be obtained as

$$\begin{aligned} F(\beta) &= \frac{m_{|C_\phi|^2}^{(4)} - 4m_{|C_\phi|^2}^{(1)} \cdot m_{|C_\phi|^2}^{(3)} + 6\left(m_{|C_\phi|^2}^{(1)}\right)^2 \cdot m_{|C_\phi|^2}^{(2)} - 3\left(m_{|C_\phi|^2}^{(1)}\right)^4}{\left(m_{|C_\phi|^2}^{(2)} - \left(m_{|C_\phi|^2}^{(1)}\right)^2\right)^2} \\ &= \frac{8\Gamma^3\left(\frac{1}{\beta}\right)\Gamma\left(\frac{9}{\beta}\right) - 16\Gamma^2\left(\frac{1}{\beta}\right)\Gamma\left(\frac{7}{\beta}\right)\Gamma\left(\frac{3}{\beta}\right) + 12\Gamma\left(\frac{1}{\beta}\right)\Gamma\left(\frac{5}{\beta}\right)\Gamma^2\left(\frac{3}{\beta}\right) - 3\Gamma^4\left(\frac{3}{\beta}\right)}{4\Gamma^2\left(\frac{1}{\beta}\right)\Gamma^2\left(\frac{5}{\beta}\right) - 4\Gamma\left(\frac{1}{\beta}\right)\Gamma^2\left(\frac{3}{\beta}\right)\Gamma\left(\frac{5}{\beta}\right) + \Gamma^4\left(\frac{3}{\beta}\right)} \end{aligned} \quad (20)$$



$$\begin{aligned} \varepsilon_x^{\text{KCE}}(\mathbf{C}, \mathbf{k}_1(x), \mathbf{k}_2(x)) \\ = \gamma_1 \int_{\text{inside contour}(\mathbf{C})} \mathbf{K}(x-y) |\mathbf{I}(y) - \mathbf{k}_1(x)|^2 dy \\ + \gamma_2 \int_{\text{outside contour}(\mathbf{C})} \mathbf{K}(x-y) |\mathbf{I}(y) - \mathbf{k}_2(x)|^2 dy. \end{aligned} \quad (23)$$

The equation associated with the level set function based on active contour  $\mathbf{C}$  is an initial contour at the area for image  $r$ . This contour divides the image into areas inside and outside the contour ( $\gamma_1$  and  $\gamma_2$ ), which are fixed numbers, and  $\mathbf{K}$  is the kernel for the level set function. KCE is assumed to be a kernel and  $\mathbf{k}_1(x)$  and  $\mathbf{k}_2(x)$  are functions that cause the textures of the image to conform to pixel  $x$ ;  $\mathbf{k}_1(x)$  is the texture of the image in the area inside the contour and  $\mathbf{k}_2(x)$  is the texture of the image in the area outside the contour. Pixel  $x$  is the central point of the equation and the energy is the KCE around the central point around  $x$ . Kernel function  $\mathbf{K}(x)$  is extracted in (17) and must be used in the KCE energy function.

#### E. Analysis of Estimating Methods

There are several methods available for estimating  $\beta$ . Do and Vetterli [51] presented a method based on maximum-likelihood (ML) estimator and Newton-Raphson (NR) algorithm iteration for estimating  $\beta$ . The estimates were compared with those obtained by kurtosis to demonstrate the superiority of using kurtosis over ML. Since the Do and Vetterli method [51] is based on the distribution of the absolute wavelet coefficients, it was compared with Akbarizadeh algorithm [1] which estimates  $\beta$  using kurtosis.

The method presented by Do and Vetterli [51] is similar to (15), where the absolute values of the wavelet coefficients have a generalized Gaussian model as

$$p(C_\phi; \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|C_\phi|/\alpha)^\beta}.$$

The generalized Gaussian model includes special case Gaussian and Laplacian PDFs corresponding to  $\beta = 2$  and  $\beta = 1$ , respectively.

Assume that the likelihood function for  $C_\phi = (C_{\phi_1}, \dots, C_{\phi_L})$  samples has independent values defined as

$$L(C_\phi; \alpha, \beta) = \log \prod_{i=1}^L p(C_{\phi_i}; \alpha, \beta). \quad (24)$$

Estimate  $\alpha$  and  $\beta$  using the maximum likelihood (MLE) as follows:

$$\frac{\partial L(C_\phi; \alpha, \beta)}{\partial \alpha} = -\frac{L}{\alpha} + \sum_{i=1}^L \frac{\beta |C_{\phi_i}|^\beta \alpha^{-\beta}}{\alpha} = 0 \quad (25)$$

$$\begin{aligned} \frac{\partial L(C_\phi; \alpha, \beta)}{\partial \beta} &= \frac{L}{\beta} + \frac{L\Psi(1/\beta)}{\beta^2} \\ &\quad - \sum_i \left( \frac{|C_{\phi_i}|^\beta}{\alpha} \right) \log \left( \frac{|C_{\phi_i}|}{\alpha} \right) = 0 \end{aligned} \quad (26)$$

where  $\Psi(\cdot)$  is the Digamma function defined  $\Psi(z) = \frac{\Gamma'(z)}{\Gamma(z)}$ .

Determine the value of  $\beta > 0$ . The unique, positive, and real solution of (25) is

$$\hat{\alpha} = \left( \frac{\beta}{L} \sum_{i=1}^L |C_{\phi_i}|^\beta \right)^{1/\beta}. \quad (27)$$

Substitute the above value into (26) to obtain  $\beta$ . The solution to the transcendental equation is

$$\begin{aligned} 1 + \frac{\Psi(1/\hat{\beta})}{\hat{\beta}} - \frac{\sum_{i=1}^L |C_{\phi_i}|^{\hat{\beta}} \log |C_{\phi_i}|}{\sum |C_{\phi_i}|^{\hat{\beta}}} \\ + \frac{\log \left( \frac{\hat{\beta}}{L} \sum_{i=1}^L |C_{\phi_i}|^{\hat{\beta}} \right)}{\hat{\beta}} = 0. \end{aligned} \quad (28)$$

$\hat{\beta}$  is obtained using the iterative NR process. Experiments show that solutions with an accuracy of  $10^{-6}$  can be obtained after three iterations. The left side of (28) is a function of  $\hat{\beta}$ , which is named  $g(\hat{\beta})$ . So,  $g(\beta)$  can be defined by

$$\begin{aligned} g(\beta) &= 1 + \frac{\Psi(1/\beta)}{\beta} - \frac{\sum_{i=1}^L |C_{\phi_i}|^\beta \log |C_{\phi_i}|}{\sum |C_{\phi_i}|^\beta} \\ &\quad + \frac{\log \left( \frac{\beta}{L} \sum_{i=1}^L |C_{\phi_i}|^\beta \right)}{\beta}. \end{aligned} \quad (29)$$

The next iterations to obtain  $g(\beta)$  and  $\beta_{k+1}$  are based on  $\beta_k$  from the following equation:

$$\beta_{k+1} = \beta_k - \frac{g(\beta_k)}{g'(\beta_k)}. \quad (30)$$

We have

$$\begin{aligned} g'(\beta) &= \frac{\Psi(1/\beta)}{\beta^2} - \frac{\Psi'(1/\beta)}{\beta^3} + \frac{1}{\beta^2} \\ &\quad - \frac{\sum_{i=1}^L |C_{\phi_i}|^\beta (\log |C_{\phi_i}|)^2}{\left( \sum_{i=1}^L |C_{\phi_i}|^\beta \right)^2} + \frac{\sum_{i=1}^L |C_{\phi_i}|^\beta (\log |C_{\phi_i}|)}{\left( \sum_{i=1}^L |C_{\phi_i}|^\beta \right)^2} \\ &\quad + \frac{\sum_{i=1}^L |C_{\phi_i}|^\beta (\log |C_{\phi_i}|)}{\beta \left( \sum_{i=1}^L |C_{\phi_i}|^\beta \right)} - \frac{\log \left( \frac{\beta}{L} \sum_{i=1}^L |C_{\phi_i}|^\beta \right)}{\beta^2} \end{aligned} \quad (31)$$

where  $\Psi(z)$  is the first polygamma or trigamma function. A good value for the  $g(\beta)$  root can be estimated using the matching moments of the dataset with the assumed distribution. For generalized Gaussian, the ratio of mean absolute value to standard deviation is a monotonically increasing function of  $\beta$

$$F_M(\beta) = \frac{\Gamma(2/\beta)}{\sqrt{\Gamma(1/\beta) \Gamma(3/\beta)}}. \quad (32)$$

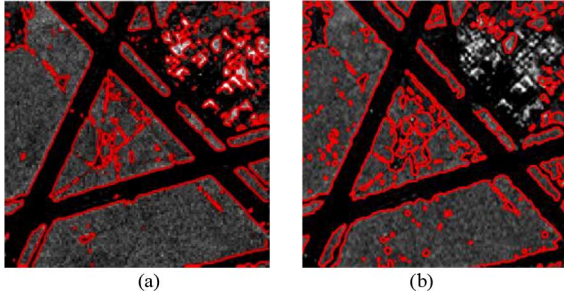


Fig. 2. Segmentation using  $\alpha$  and  $\beta$  estimates from these methods. (a) Kurtosis. (b) Maximum likelihood.

If  $m_1 = (1/L) \sum_{i=1}^L |C_{\phi_i}|$  and  $m_2 = (1/L) \sum_{i=1}^L C_{\phi_i}^2$  are estimates of the mean absolute value and variance of the dataset samples, respectively,  $\beta$  can be obtained as

$$\bar{\beta} = F_M^{-1} \left( \frac{m_1}{\sqrt{m_2}} \right). \quad (33)$$

In practical applications, this equation can be quickly solved by interpolation and the use of lookup tables with corresponding values of  $\frac{m_1}{\sqrt{m_2}}$  and  $\bar{\beta}$ . The initial estimated value for  $\beta_0 = \bar{\beta}$  can be obtained by using the ML estimator. Experiment shows that  $\hat{\beta}$  improves after three NR iterations.

In the method proposed by Akbarizadeh [1], the values of  $\alpha$  and  $\beta$  obtained using kurtosis were 40.44 and 1.537, respectively. So, the initial values for  $\alpha$  and  $\beta$  were chosen as 40 and 1, respectively. Using the equations presented for maximum likelihood and NR,  $\alpha$  and  $\beta$  were calculated to be 38.4917 and 0.9647, respectively. Segmentation was performed using these values and entering them into the level set function. Fig. 2(a) and (b) shows the segmentations obtained by estimation of  $\beta$  using [1] and [51], respectively. As seen, the results for the kurtosis estimation were significantly better than those for ML and the NR estimations.

Although the initial value of  $\beta$  for the kurtosis estimator was set according to the values obtained for  $\beta$  by the kurtosis estimation, ML did not perform well, which leads to poor segmentation compared to kurtosis. The following disadvantages of MLE are also the reasons to utilize kurtosis to estimate the parameters of  $\beta$ .

- 1) MLE is sensitive to initial value selection.
- 2) The estimation is highly biased in small samples and its optimum characteristics do not apply to small samples.
- 3) Solving likelihood equations requires multiple iterations which increases computational time.
- 4) The ML algorithm is an iterative algorithm, where the iteration must be repeated to eventually converge to a maximum value. The problem with this algorithm is that convergence may not be reached, and the loop may fall in an infinite iteration; or the iteration may stop at a local maximum, and the maximum likelihood algorithm wrongly identifies this value as the global maximum and yields an incorrect value. Therefore, convergence may

occur at the local maxima and not converge to the global maximum.

- 5) The estimation is usually nontrivial, except where the maximum likelihood follows a simple formula. In most cases, high-quality statistical software is required for estimations using maximum likelihood.
- 6) Likelihood equations perform well only for one type of distribution in a given problem. The mathematics is usually nontrivial, especially if the confidence intervals of the parameters are considered.
- 7) The variance components are underestimated.

*Phase II:* Fig. 3 is a schematic of phase II of the proposed algorithm after determining the boundaries using the level set and active contours.

In this phase, the Gabor filter bank [24] is applied to the image and the texture features are extracted. The Gabor filter is 2-D in the spatial domain ( $x$ - $y$  plane) and is calculated as [24]:

$$g_{\lambda\theta\psi\sigma\gamma}(x, y) = \exp \left( -\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2} \right) \cos \left( 2\pi \frac{x'}{\lambda} + \psi \right). \quad (34)$$

In (34),  $x' = x \cos(\theta) + y \sin(\theta)$ ,  $y' = y \cos(\theta) - x \sin(\theta)$ ,  $f = 1/\lambda$ , and  $\lambda$  is the wavelength of the sinusoidal function,  $\theta$  is the orientation of the Gabor filter, and  $\Psi$  is the phase offset in the Gabor filter.  $\Psi = 90^\circ$  is the real value and  $\Psi = 0^\circ$  is the imaginary value of the Gabor filter and they are even-symmetric and odd-symmetric filters, respectively. To meet the requirements proposed by Malik and Perona [33], we have chosen the real value of the Gabor filter.  $\sigma$  is the spread of the Gabor filter or the Gaussian standard deviation and is determined by  $\lambda$  and  $b$ . The value of  $b$  can be obtained as

$$b = \log_2 \frac{\frac{\sigma}{\lambda} \pi + \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda} \pi - \sqrt{\frac{\ln 2}{2}}}, \quad \frac{\sigma}{\lambda} = \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}} \frac{2^b + 1}{2^b - 1} \quad (35)$$

where  $b$  is the spatial-frequency bandwidth of the Gabor filter and  $\sigma/\lambda$  determines the spatial domain in simple cells. Next, choose 1 octave for the value of  $b$  as recommended by Campbell and Kulikowski [35]. The frequency values are obtained using (36) and (37). The distance between the directions of the Gabor filter can be adjusted. Here, this value is  $30^\circ$ , as recommended by Hubel and Wiesel [34].

Texture features often lie in the intermediate frequency bands because most spectral energy of natural images centers occurs at low frequencies and a small amount of energy occurs at high frequencies.  $F_L$  and  $F_H$  are low and high frequencies, respectively

$$F_L(i) = 0.25 - 2^{i-0.5}/N_c \quad (36)$$

$$F_H(i) = 0.25 + 2^{i-0.5}/N_c \quad (37)$$

where  $N_c$  is the number of columns in the image (width of image),  $i = 1, 2, \dots, \log_2^{(N_c/8)}$ . Selection of a specific frequency range for (36) and (37) reduces complexity and computational load [52]. So, we choose  $0 < F_L(i) < 0.25$ , and  $0.25 < F_H(i) < 0.5$ .

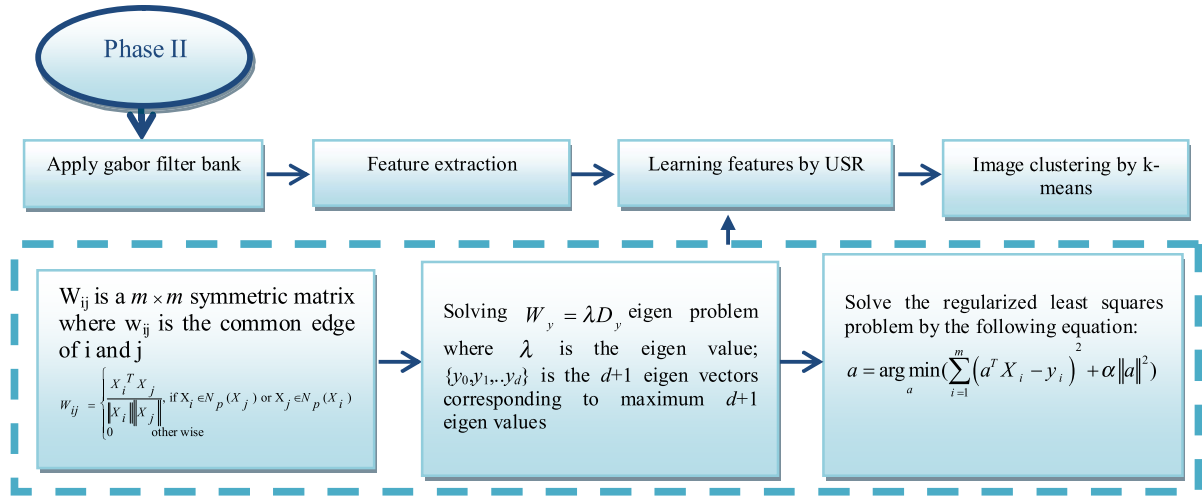


Fig. 3. Schematic of phase II of proposed algorithm: implementation of USR and clustering algorithms.

After applying the Gabor filter to the image, the feature is extracted using (38) and image smoothing is applied using (39) [24]

$$\tan(\alpha t) = \frac{1 - e^{-2\alpha t}}{1 + e^{-2\alpha t}} \quad (38)$$

$$g(x, y) = \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\} \quad (39)$$

where  $\sigma$  represents the standard deviation and  $\alpha$  is a constant value.

Gaussian smoothing improves performance of Gabor filters for texture analysis. Textures with sufficient bandwidth encounter leakage. Application of Gaussian filters of the width of the filter channels reduces leakage. These filters and channels have similar forms with a higher spatial extent [9].

After obtaining image features, the features for different filters should be combined to carry out segmentation. For  $k$  types of  $C_1, C_2, \dots, C_k$  texture, each texture forms a cluster in feature space in which each cluster is distinctive from a different texture. Then, each feature is normalized to have a mean 0 and variance 1 to avoid overlooking features with smaller value ranges while comparing features with bigger value ranges, so that they all have equal effects.

After extracting the features, they are learnt using USR [18], [19], [25]–[31]. This improves discrimination power of the extracted features. In texture segmentation, adjacent pixels most likely belong to a single texture. Therefore, spatial coordinates of pixels are added as two features in this stage. Using this method, spatial adjacency data are directly engaged in clustering. The  $k$ -means algorithm is then applied to the image.

The results of segmentation are then compared with segmentation using the Nyström and PSCC methods. USR, in addition to dimension reduction, does not require calculation of the eigen-decomposition; this decreases time and memory costs and calculation complexity and is the advantage of USR over other dimension-reduction approaches.

TABLE II  
DESCRIPTION OF DATASETS

DATASETS	Instance	Dimension	Number of clusters
Wine	178	13	3
Iris	150	4	3
Waveform	3353	21	2
Ionosphere	351	34	2
Breast Cancer W	699	9	2

## VI. EXPERIMENTAL RESULT

### A. Dataset Clustering

To evaluate the performance of the algorithms for feature learning and clustering, the  $k$ -means with unlearned features was compared with  $k$ -means with learned features on five UCI machine learning repository datasets [32]. Table II shows a description of the dataset, the number of samples in each dataset, sample size, and number of clusters in each sample.

In Table III, in the unlearned features column, the  $k$ -means algorithm was applied to the UCI datasets. In the learned features column, the features were first learnt by USR and then the  $k$ -means algorithm was applied. The results were compared for accuracy and normalized mutual information (NMI) [53], [54] in the unlearned and learned features.

Let  $x_i$  denote each feature,  $s_i$  the true labels and  $r_i$  the cluster labels generated by the algorithm. Equation (40) yields the accuracy value as

$$Accuracy = \frac{\sum_{i=1}^n \delta(s_i, \text{map}(r_i))}{n} \quad (40)$$

where  $n$  is the total number of features and  $\delta(x, y)$  is the delta function and equals 1 if  $x = y$  and zero if otherwise.  $\text{Map}(r_i)$  is the permutation mapping function to map each  $r_i$  cluster label to an equal algorithm label. The best mapping performance can be obtained using the Kuhn–Munkers algorithm, where the higher values indicate higher performance of the clustering algorithm.

TABLE III  
NMI AND ACCURACY FOR LEARNED AND UNLEARNED FEATURES FOR UCI DATASETS

Datasets	Normalized mutual information		Accuracy (100%)	
	Unlearned features	Learned features	Unlearned features	Learned features
Wine	0.4287	<b>0.6537</b>	70.22	<b>89.76</b>
Iris	0.7515	<b>0.8309</b>	89.33	<b>94.00</b>
Waveform	0.3612	<b>0.3647</b>	50.18	<b>50.36</b>
Ionosphere	0.0396	<b>0.1155</b>	60.11	<b>74.07</b>
Breast cancer W	0.0024	<b>0.7774</b>	65.52	<b>96.71</b>

If  $C$  and  $C'$  denote the cluster sets obtained by ground truth and the proposed algorithm, respectively, mutual information criterion,  $MI(C, C')$  is determined as follows:

$$MI(C, C') = \sum_{c_i \in C, c_j' \in C'} p(c_i, c_j') \cdot \log_2 \frac{p(c_i, c_j')}{p(c_i) \cdot p(c_j')} \quad (41)$$

where  $p(c_i)$ ,  $p(c_j')$  are the probability of the features belonging to clusters  $c_i$  and  $c_j'$ , respectively, and  $p(c_i, c_j')$  is the common probability. This means that the feature belongs to both class  $c_i$  and class  $c_j'$ . The normalized value of mutual information  $\overline{MI}(C, C')$  was used as follows:

$$\overline{MI}(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))} \quad (42)$$

where  $H(C)$  and  $H(C')$  are the  $C$  and  $C'$  entropies, respectively. It is evident that the value of  $\overline{MI}(C, C')$  is between 0 and 1.  $\overline{MI} = 1$  signifies that two cluster sets are equal and  $\overline{MI} = 0$  signifies that they are independent. When the clusters are compared with ground truth, values are closer to 1 which indicate that the algorithm has better performance and has performed clustering more efficiently.

Table III shows comparisons of the unlearned and learned features in UCI datasets. The results of this table show that learning the feature increased performance of the algorithm and improve the accuracy and NMI.

The proposed algorithm was compared with other algorithms to demonstrate its effectiveness for segmentation of SAR images. Its performance was also evaluated using random texture [55] and histology dataset images [55]. All images in these datasets were  $128 \times 128$  in size. The algorithms employed to evaluate the proposed method are described below.

JSEG [56] searches for segmentation that produces a uniform color distribution using region growing. EDISON [57] uses mean shift for cluster pixels for their color and location. Efficient graph-based segmentation [58] is also graph-based and is similar to normalized cut, except that the edges are assigned between adjacent pixels based on difference in intensity. The gPb-owt-ucm algorithm [59] detects a local edge by combining color and texture features. The edges and spectral clustering are used to detect contours in the images. A watershed transform finds regions using contours. In occlusion of random texture segmenter (ORTSEG) [60] method, the maximum weight deconvolution algorithm was used. No prior information was available regarding the boundaries of different regions.

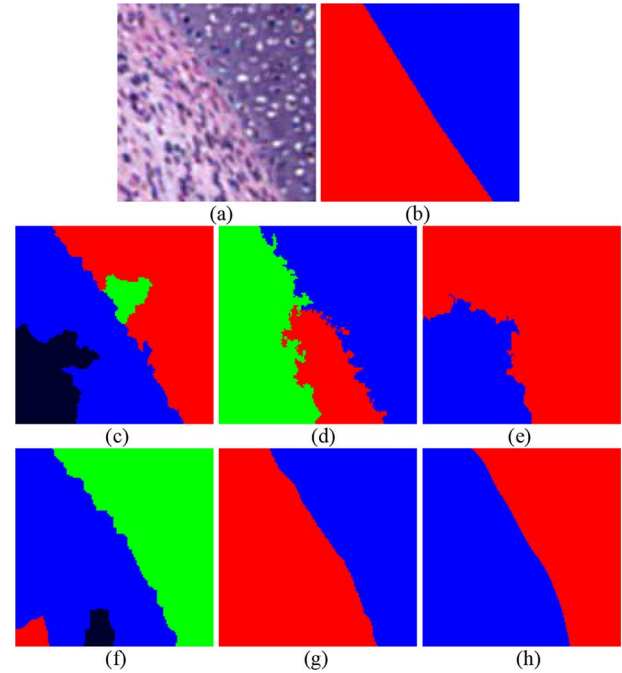


Fig. 4. Segmentation results for different methods on an image in histology dataset. (a) Input image. (b) Ground truth. (c) JSEG (error rate = 17.9%). (d) EDISON (error rate = 14.9%). (e) Efficient (error rate = 38.6%). (f) gPb-owt-ucm (error rate = 9.7%). (g) ORTSEG (error rate = 7.6%). (h) Proposed method (error rate = 0.8%).

The histology dataset consists of color H&E-stained tissue images. Ground truths were determined by a pathologist through visual inspection. From a set of 36 images and by comparing more than 20 types of tissues, 36 subimages were obtained. Each of these subimages containing only two types of tissue. The aim of this process is only the subimages that have a very accurate ground-truth label. This dataset is relatively small, because only a trained pathologist can label the images. Tissue-based labeling is very time consuming as well [55], [60], [61].

Fig. 4 shows segmentation of the image from the histology dataset using the different algorithms. Fig. 4(a) shows the dataset image and Fig. 4(b) shows the ground truth. Fig. 4(c), (d), and (f) detected the image in three and four classes. The performance of ORTSEG in segmenting the image is fairly good. Our algorithm segments the image with high accuracy and properly detects two regions and their boundary.

If there is an  $M \times N$  image size (i.e., an image with  $M$  rows and  $N$  columns), the difference between the segmented image



and ground-truth image will be obtained first to calculate the accuracy in the simulated SAR image. If  $F$  is the ground-truth image and  $G$  is the image segmented by the relevant algorithm,  $diff$  is the absolute of difference between the two images ( $diff = |F - G|$ ). We binarize the  $diff$  image. In this case, values of the image pixels are rounded to 0 or 1, and a black and white image is obtained.

Black pixels on the  $diff$  image indicate that segmentation was performed accurately and ground-truth and segmented images are equal in those pixels. The rate of misclassification pixels and the consequent rate of segmentation error can be obtained using (43) by counting the nonblack pixels in the  $diff$  image.  $Error$  denotes the error percentage. The smaller the error percentage is, the better the performance of an algorithm is, i.e.,

$$Error = \frac{L}{MN} \times 100 \quad (43)$$

where  $L$  is number of misclassification pixels and  $MN$  is the total number of pixels of an image.

The segmentation error in Fig. 4(c)–(g) was calculated as 17.9%, 14.9%, 38.6%, 9.7%, and 7.6%. The error rate was 0.8% for the proposed method, which was the lowest segmentation error of the compared methods.

In mathematics, the diagram resulted from the segmentation of a plan into several regions is called a Voroni diagram. This segmentation is based on the distance from the points located in a special place on the plan. The regions included in the segmented plan are called Voronoi cells. Random texture dataset is an artificial dataset designed to show the difficulty in the segmentation of edgeless images. To generate the images of these datasets, three fundamental locations (seeds) are randomly selected, and the ground truth is determined using the Voronoi cell map. The Veroni cells are formed using three random specified points. A random distribution of colors is considered for each region. Each pixel is selected independent of the distribution. Since in these images, each pixel is independent of its neighborhood, the images have a persistence length of 1. The resulting image can be segmented by eye, but has no discernable edge. The high frequencies in the images may be detected as edges. The dataset includes 25 subimages, each of which has eight unique colors and is  $128 \times 128$  [55], [60].

The results of these methods and those obtained by the proposed method on the random texture dataset were compared. Fig. 5(a) shows the dataset image and Fig. 5(b) shows the ground truth. Fig. 5(c)–(g) shows the methods with which the proposed method was compared. As seen, the gPb-owt-ucm and Efficient methods detected the image in one and two classes, respectively. JSEG and EDISON had difficulty and failed to segment the image. The proposed method performed well in detecting the three classes and was successful in segmenting them. The accuracy of the ORTSEG method is slightly higher, but since the segmentation error of the proposed method in Fig. 4 was much lower than ORTSEG, we can conclude that the proposed algorithm has the best performance among the five other algorithms. The error rates of the methods shown in Fig. 5(c)–(h) were 3.8%, 30.8%, 39.8%, 65.3%, 1.33%, and 1.94%, respectively.

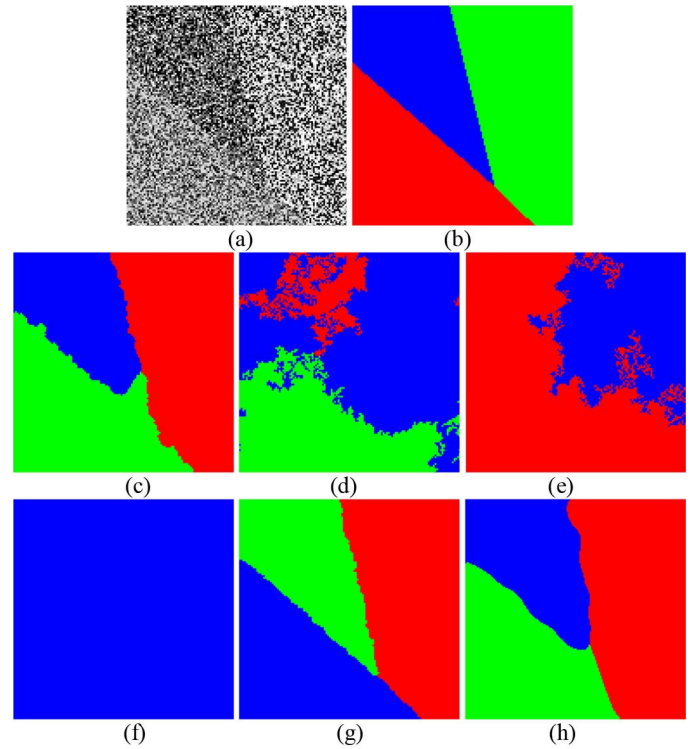


Fig. 5. Segmentation results for different method for an image from a random texture dataset. (a) Input image. (b) Ground truth. (c) JSEG (error rate = 3.8%). (d) EDISON (error rate = 30.8%). (e) Efficient (error rate = 39.8%). (f) gPb-owt-ucm (error rate = 65.3%). (g) ORTSEG (error rate = 1.3%). (h) Proposed method (error rate = 1.94%).

### B. Segmentation of Simulated SAR Image

*Phase I:* In order to objectively evaluate the performance of the proposed method, an experiment on a simulated three-look SAR image was done. The three-look noisy image, as shown in Fig. 6(a), is generated by averaging three independent realizations of speckle. Fig. 6(b) is the clear main image that is the ground truth used to calculate the error rates for segmentation with different algorithms.

The SCE, KWE, and KCE algorithms were used to apply segmentation on SAR images and the results were compared. The segmentation results obtained by SCE and KWE, as shown in Fig. 6(c) and (d), respectively, indicate that segmentation using KWE was better than SCE. The error rate of KWE decreased from 69.31% to 52.23% and the error rate of KCE decreased to 19.57%. It can be concluded that KWE performed better than SCE and KCE performed better than KWE.

The results from observation are also important. Segmentation using SCE, as shown in Fig. 6(c), caused severe noise in the areas outside the triangle on the image, indicating that many pixels were misclassified. Fig. 6(d) shows that the KWE method was more resistant to noise than SCE. All results show that the misclassified pixels were in areas outside the triangle and the areas inside the triangle were correctly segmented. As seen, KCE had less segmentation error compared to SCE and KWE.

*Phase II:* The ground-truth image in Figs. 7 and 8 is a regional ground truth and is not based on object and clutter. In

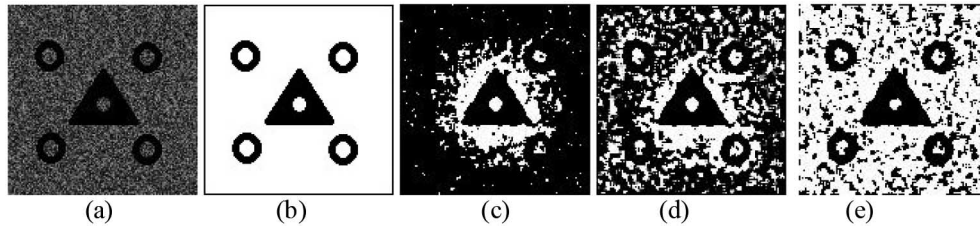


Fig. 6. Segmentation results on simulated SAR image. (a) Three-look simulated SAR image ( $128 \times 128$ ). (b) Ground truth-image. (c) SCE model (error rate: 69.31%; mis-segmented pixels: 11355) [62]. (d) KWE model (error rate: 52.23%; mis-segmented pixels: 8558). (e) KCE model (error rate: 19.57%; mis-segmented pixels: 3208).

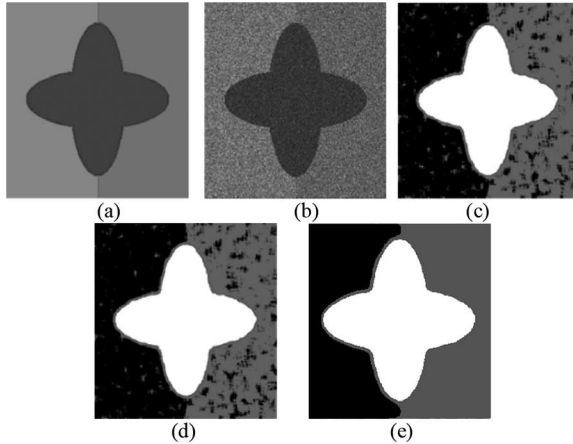


Fig. 7. Segmentation results on simulated SAR image. (a) Ground-truth image. (b) Two-look simulated SAR image ( $256 \times 256$  pixels). (c) Nyström (error rate: 6.92%). (d) PSCC (error rate: 4.63%). (e) Proposed method (error rate: 0.59%).

this case, all regions are considered, and segmentation is carried out based on the regions.

The image, as shown in Fig. 7(a), shows a 3-class ground-truth image and Fig. 7(b) shows the corresponding two-look noisy image [22]. Fig. 7(c)–(e) shows segmentation of the image using Nyström, PSCC, and the proposed method, respectively. As k-means algorithm was used in the final stage of the three algorithms of Nyström, PSCC, and the proposed method, various results were obtained in different runs of each algorithm. In order to consider equal conditions for all, we run each algorithm 10 times and select the best segmentation presented by each algorithm. The images shown for each method were the best visual result of ten runs of the algorithms. For Nyström, the number of sampling points was 100 and the scaling parameter was  $\sigma = 0.1$ . For PSCC, the number of nearest neighbors was 100 and  $\sigma = 0.05$ .

The estimated error rates in all three approaches were calculated using (43) and compared. The error rates were 6.92%, 4.63%, and 0.59% for Nyström, PSCC, and the proposed method, respectively, which indicates that the proposed algorithm is a better approach for segmentation of the simulated image.

The segmentations obtained by Nyström and PSCC clustering, as shown in Fig. 7(c) and (d), respectively, show a gray area within the black spots and a black area within the gray spots. The white area has been clustered properly and noise

has disturbed the black and gray areas. According to Fig. 7(a), we can see that all three algorithms have white areas that are correctly recognized. So, all approaches are accompanied by white strains. The proposed method, as shown in Fig. 7(e), provided good segmentation for all three areas and had fewer mis-segmented pixels in the black and gray areas. The proposed algorithm was useful for reorganization of the areas and the resulting segmented image was very similar to the ground-truth image.

The proposed algorithm was compared with several fuzzy-based clustering algorithms and kernel graph cuts [63], which is a graph-based method. KWFLICM algorithm [64] is a fuzzy algorithm based on the tradeoff between weighted fuzzy factors and a kernel metric. ILKFCM algorithm [65] has a fuzzy kernel that includes spatial information and pixel intensities. Using this kernel, spatial information and intensity are considered simultaneously for all neighboring pixels. Wavelets are used to represent texture information. NLEP-FCM [66] is a robust nonlocal FCM method which retains borders.

The best method among the segmentation methods compared in the images in Figs. 4 and 5, i.e., the ORTSEG method, was selected and applied on the simulated SAR image and compared with fuzzy techniques as well as our proposed method.

Fig. 8 compares the four fuzzy algorithms with ORTSEG and the proposed method. Fig. 8(a) is ground truth ( $244 \times 244$ ). Fig. 8(b) is the corresponding eight-look noisy image. Fig. 8(c)–(g) shows the results of image segmentation using the different algorithms. The segmented image in Fig. 8(c) includes noise. In Fig. 8(d) for the NLEP-FCM method, the word SAR was not properly obtained and details were removed from the image. Fig. 8(f) shows the results of the graph-cuts method where the object boundaries are troubled and the word SAR is not properly obtained. ORTSEG does not perform well on the boundaries of the objects in the image and the word SAR has cuts.

ILKFCM showed better results than other methods. The boundaries of the objects and word SAR were properly segmented; however, there are black spots in the boundaries of the background. Object boundaries are sharper in the proposed method and there is no misclassification in the background boundaries. The error rates in Fig. 8(c)–(g) were 0.62%, 0.14%, 0.02%, 1.65%, 13.8%, and 0.011%, respectively. This indicates that the proposed method performed better than other algorithms.

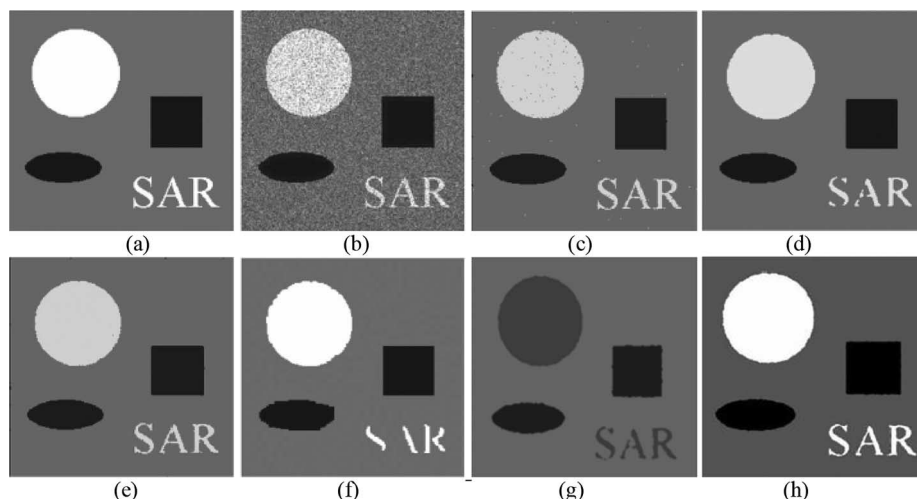


Fig. 8. Segmentation results on simulated SAR image. (a) Ground truth. (b) Noisy image. (c) KWFLICM (error rate = 62%). (d) NLEP-FCM (error rate = 14%). (e) LKFCM (error rate = 0.02%). (f) Kernel graph cuts (error rate = 1.65%). (g) ORTSEG (error rate = 13.8%). (h) Proposed method (error rate = 0.011%).

### C. Segmentation of Real SAR Image

*Phase I:* In this Phase, the evaluation of the segmentation result is based on visual inspection of the segmented images, because in this phase, the detailed ground truths corresponding to the SAR images being segmented are generally absent.

The image, as shown in Fig. 9(a), is a part of a Ku-band real SAR image with 3-m spatial resolution in the area of China Lake Airport in California. The image size is  $250 \times 251$  pixels [22]. This image consists of three types of land cover: runway, vacancy area, and airport buildings. To evaluate the performance of the proposed method, SCE, KWE, and KCE methods were selected for comparison. The segmentations obtained by SCE, KWE, and KCE are shown in Fig. 9(b)–(d), respectively. Fig. 9(e)–(g) shows magnified portions of selected areas.

The segmentation obtained by the SCE is shown in Fig. 9(b) and indicates that SCE had difficulty with segmentation of the runway areas (lower left); the boundary between the vacancy area and runway is also not defined properly. The segmentation obtained by KWE, as shown in Fig. 9(c), shows improved uniformity in runway regions. However, there is serious mis-segmentation in the vacancy region. The proposed method better identifies the runway area details in the triangular area, whereas other two methods show many mis-segmented pixels in this area.

Fig. 9(d) shows the segmentation obtained by KCE and Fig. 9(g) shows a magnified image of Fig. 9(c). The best results for segmentation with improved uniformity of the areas on the runway and in the vacancy area (wide coverage and uniformity of the runway) were obtained by KCE, as shown in Fig. 9(c) and (g). KCE recognized all three types of land covers properly and the boundaries of the areas were determined properly.

*Phase II:* In phase II, after the Gabor filter bank is applied, texture features are extracted and learned using USR. Finally k-means clustering was applied. Fig. 10(a) is a version of Fig. 9(a) that better shows the comparison with similar methods. The image is  $234 \times 208$  pixels, including three types of land cover: the vacancy area, airport building, and runway. Fig. 10(b) shows the ground-truth image [22]. The best visual

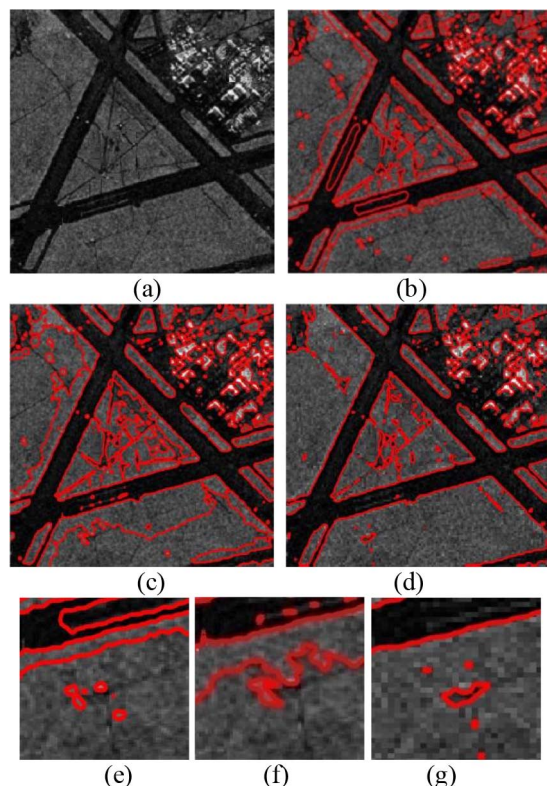


Fig. 9. Segmentation. (a) Ku-band SAR image of China Lake Airport, California at 3-m resolution ( $250 \times 251$ ). (b) SCE. (c) KWE. (d) KCE. (e) [respectively, (f) and (g)] Zoom of an area extracted from (b) [respectively, (c) and (d)].

result in ten runs of the algorithms was used for the comparisons conducted in this section. For PSSC, the number of nearest numbers was 100 and the scaling parameter  $\sigma$  was 0.05. For Nyström, the number of sampling points was 100 and  $\sigma$  is 0.2.

Fig. 10(c)–(e) shows the segmentation results using Nyström, PSSC, and the proposed method.

The parts denoted by circles in Fig. 10(f) include details that the Nyström and PSSC methods were unable to detect as



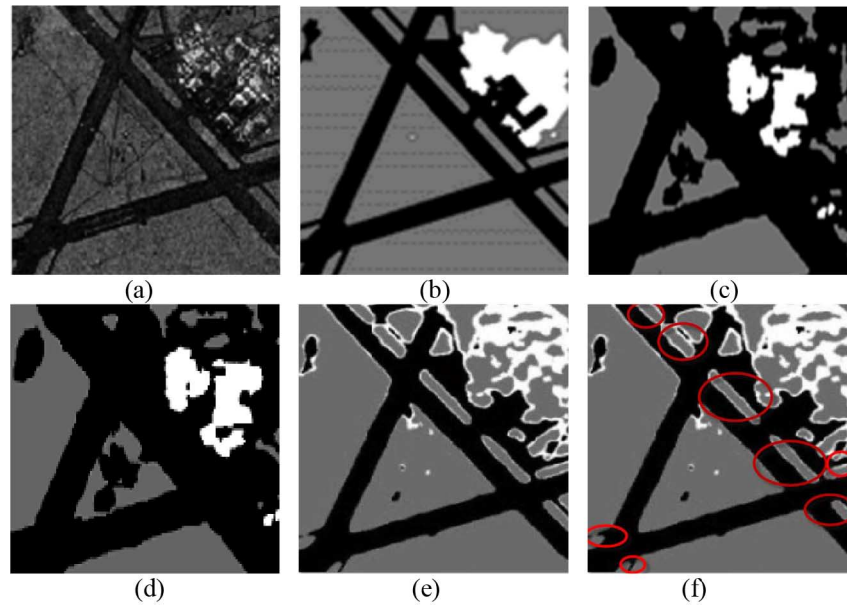


Fig. 10. Segmentation. (a) Ku-band real SAR image of China Lake Airport, California with 3-m resolution ( $234 \times 208$ ). (b) Ground truth [22]. (c) Nyström [22]. (d) PSSC [22]. (e) Proposed method. (f) Spots not detected by the other methods.

compared with ground truth. These areas have been mistakenly identified as runway. In addition, many black spots are seen in the Nyström and PSSC images in the triangular section, which should be gray, as shown for ground truth. The two algorithms mistakenly detected these vacancy areas as part of the runway. In the proposed method, the spots are much smaller and less frequent in this area; the proposed algorithm detected this area much better and more smoothly than the other two methods. The upper corner on the left side contains a small area of runway that the proposed algorithm better segmented and it is more similar to ground truth than the other methods.

Each method has some deficiencies. The white areas (airport buildings) were segmented better using Nyström and PSSC methods. In this area, the proposed method does not outperform the Nyström and PSSC methods. The proposed method was more suitable than other methods for separating the vacancy area from the airport. In these areas, it has fewer misclassifications than other methods.

Another experiment is carried out on a four-look C-band RadarSAT-2 satellite SAR image with HV polarization in the area of Gulangya Iset in China, as shown in Fig. 11(a). The 2-class (binary) image is  $400 \times 400$  pixels [22]. The aim was to find the boundary between the island and the water. Fig. 11(b) shows the ground-truth image. Fig. 11(c)–(e) shows the segmentation results of Nyström, PSSC, and the proposed method, respectively. The spots marked with circles in Fig. 11(f) were mis-segmented in PSSC.

Broken parts can be seen in the island area in Fig. 11(c) for Nyström; these parts were mistakenly classified as water. The segmentation result obtained by PSSC, as shown in Fig. 11(d), has some misclassifications in the island area, which was segmented correctly using the proposed method. In Fig. 11(f), the upper circles show areas of water that were misclassified as part of island. In the lower circles, as in the ground-truth image, the

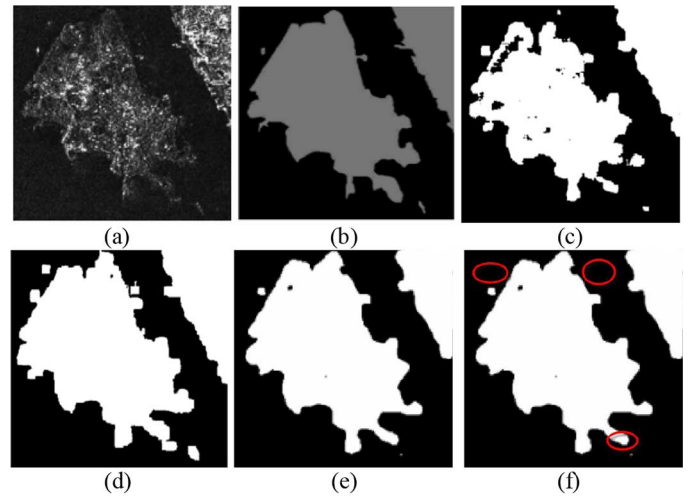


Fig. 11. Segmentation. (a) C-band SAR image of Gulangya Iset, China. (b) Ground-truth image [22]. (c) Nyström. (d) PSSC [22]. (e) Proposed method. (f) Spots misclassified by PSSC.

area is smooth, although PSSC showed this region in broken form. In the proposed method, the island and water are separated completely and were detected correctly. The segmentation carried out by the proposed method was the best and the performance of Nyström was the worst.

Fig. 12(a) shows a three-look X-band real SAR image of  $512 \times 512$  pixels. The subimage is a lake district in Switzerland taken by the Space Radar Laboratory Mission in 1994. This image consists of a lake, an urban area, and a mountainous area. The right and left sections are mountainous areas, the lake is on the top right side, and the urban area is at the bottom [22]. Fig. 12(b) shows the ground-truth image of the regions. Fig. 12(c) shows the results of segmentation using Nyström. Fig. 12(d) shows the results obtained using PSSC for



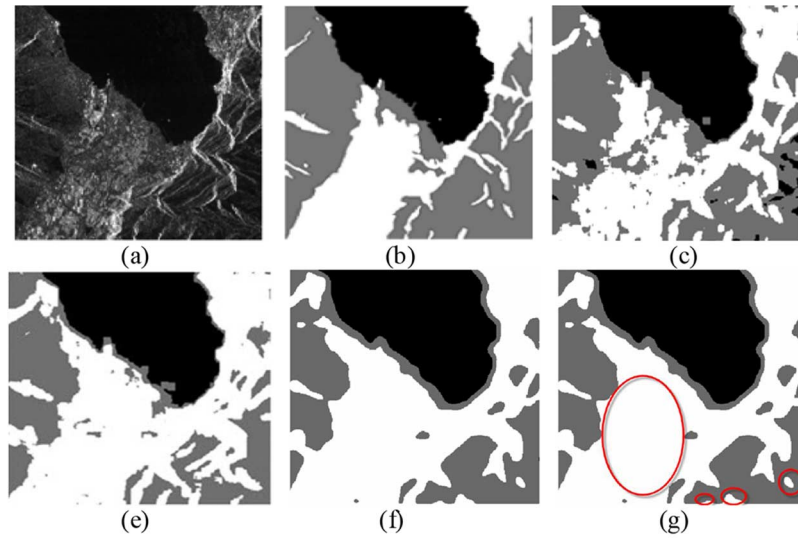


Fig. 12. Segmentation. (a) X-band SAR subimage of a lake district in Switzerland. (b) Ground-truth image [22]. (c) Nyström [22]. (d) PSSC [22]. (e) Proposed method. (f) Spots misclassified by PSSC.

TABLE IV  
REAL SAR IMAGE SEGMENTATION RESULTS

Accuracy (100%)			
	Nyström	PSCC	Proposed method
Fig.10(a)	75.39	77.49	<b>94.80</b>
Fig.11(a)	91.81	93.42	<b>96.55</b>
Fig.12(a)	75.81	84.06	<b>92.10</b>

segmentation. Fig. 12(e) shows segmentation results of using proposed method.

Fig. 12(f) shows the results of PSSC with some circles highlighting the differently classified areas. The areas specified by circles in Fig. 12(f) are those mistakenly segmented by PSSC. Fig. 12(c) shows broken parts on the island area that have been misclassified as water. Urban and mountain areas have been mistakenly detected in many areas. PSSC improved segmentation in mountain and urban areas. However, some confusion can be observed. The circle on the left, compared with ground truth, is in an urban area and is smooth. PSSC and Nyström did not segment this area correctly, but the proposed method detected it perfectly. The proposed algorithm was more successful than PSSC in detecting spatial details in urban areas. Fig. 12(e) is a sample of the details in the middle circles and on the right-hand side.

The ground-truth maps of Figs. 10–12 are labeled by referring the images from Google Map and the ground-truth maps from [22]. These ground truths consider the majority terrain type in each class. Accuracy values were assigned to the three real SAR images and were compared with other methods to better compare the proposed algorithm with the Nyström and PSSC methods. They prove the superiority of the proposed method. Equation (42) was used to obtain accuracy and Table IV shows results of the comparison. The results show that the proposed algorithm was more successful than the Nyström and PSSC methods for segmenting all real SAR images and

indicates the accuracy of the proposed method for the real SAR image which was higher than that in other algorithms.

#### D. Segmentation Evaluation

In phase I, the proposed method was used to extract the boundaries of the regions to obtain the clustering accuracy in the subimages extracted from the image. The boundaries between regions cannot be easily detected. When selecting subimages, the subimage may belong to more than one region, which causes errors in clustering and in accuracy and mutual information values. Since there is no dataset for texture in the SAR images, different total of subimages were extracted from these regions so that each subimage belonged to a specific type of texture. The algorithm of the first phase and the subimages were used to create a dataset of respective SAR images.

When image boundaries are determined by active contours, different segments of the image are easily separated. In order to show that the algorithm has better performance after learning the features, different numbers of subimages were extracted from China Lake airport in California, and the accuracy and mutual information in the clustering of the subimages whose extracted features had been learned were compared with those whose extracted features that had not been learned.

The 14, 21, 30, 33, and 42  $16 \times 16$  subimages were derived from the SAR image of China Lake, California. Each subimage belonged to a specific texture. A Gabor filter [24] was applied and the texture features were extracted. The features were learned by USR,  $k$ -means was used for clustering, and the results were compared with that of nonlearned features. The  $k$ -means algorithm can perform in original feature space or in reduced feature space [31]. The difference between clustering for learned and unlearned features in feature space was analyzed.

A comparison between the clustering of the subimages with unlearned features and those with learned features is presented in Table V. In this table, the real label of each subimage is

TABLE V  
NMI AND ACCURACY FOR LEARNED AND UNLEARNED FEATURES FOR SUBIMAGES

Subimages	Normalized mutual information		Accuracy (100%)	
	Unlearned subimages	Learned subimages	Unlearned subimages	Learned subimages
14 subimages	0.1700	<b>0.1870</b>	50.00	<b>57.14</b>
21 subimages	0.3894	<b>0.4004</b>	65.00	<b>70.00</b>
29 subimages	0.3326	<b>0.3723</b>	65.52	<b>68.97</b>
30 subimages	0.3287	<b>0.3516</b>	63.33	<b>63.33</b>
33 subimages	0.3219	<b>0.3354</b>	57.58	<b>57.58</b>
42 subimages	0.0568	<b>0.0760</b>	40.48	<b>42.86</b>

TABLE VI  
NMI AND ACCURACY FOR LEARNED AND UNLEARNED FEATURES FOR SUBIMAGES (RAND INDEX)

Subimages	Normalized mutual information		Accuracy (100%)	
	Unlearned subimages	Learned subimages	Unlearned subimages	Learned subimages
14 subimages	0.1537	<b>0.2087</b>	42.86	<b>57.14</b>
21 subimages	0.0761	<b>0.0883</b>	42.86	<b>70.00</b>
29 subimages	0.0825	<b>0.1252</b>	48.28	<b>68.97</b>
30 subimages	0.0452	<b>0.0902</b>	40.00	<b>63.33</b>
33 subimages	0.0371	<b>0.0640</b>	42.42	<b>57.58</b>
42 subimages	0.0331	<b>0.0760</b>	40.48	<b>42.86</b>

specified. Table VI shows a similar comparison, except that the labels of the subimages were considered random.

The subimages were first decomposed using the Gabor filter bank. The features were extracted and the  $k$ -means algorithm was applied to the unlearned features and USR and  $k$ -means were applied to the learned features.

The results in Tables V and VI represent the average accuracy and mutual information in ten runs of the algorithm. The results show that at least one of the two values of accuracy or mutual information increases after learning the extracted features. In other words, the proposed method improves through learning the features independent of the number of selected subimages and they also show that the algorithm yields a desirable solution despite being affected by speckle noise and destruction of their textured features. This confirms the power of the algorithm for clustering.

Two dimensions of the features are randomly selected to plot the features extracted from the 21 subimages in 2-D space. Because the images were from three classes, the features corresponded to the three regions. The dimensions are selected randomly to show that feature learning improved clustering, regardless of which dimensions of features are selected. Fig. 13(a) shows the use of  $k$ -means to classify the extracted features into three classes. As seen, clustering was not good and the features of the different classes are not differentiable. Fig. 13(b) shows the clustering, when  $k$ -means was applied to the features learned using USR. It can be seen that the classes are recognizable and clustering has improved.

#### E. Complexity Analysis

Combined addition and multiplication was used to calculate the complexity of the operation. This operation is called flam. If  $n$  is the number of data samples, each data sample has  $d$

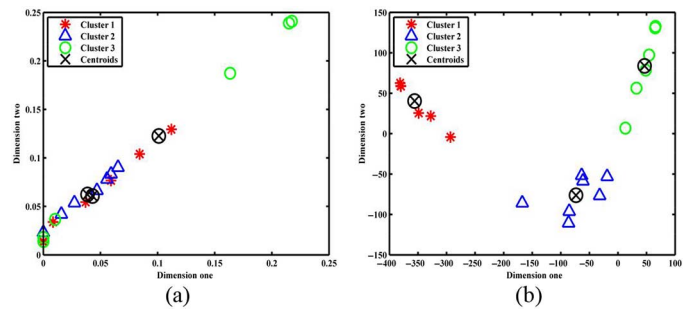


Fig. 13. (a) Unlearned feature clustering. (b) Learned feature clustering.

dimension,  $h$  is the number of sampling points for Nyström, and  $k$  is the number of clusters. Complexity is obtained using the Spectral clustering algorithm [22]. Spectral clustering consists of three parts:  $k$ -means clustering, eigen-decomposition, and construction of a similarity matrix. Construction of the similarity matrix by approximation using the Nyström method made it necessary to only calculate the product of the  $h$  randomly selected points and the other points. The Graph construction and eigen-decomposition require  $O(hnd)$  and  $O(h^3)$  complexity, respectively. Therefore, the total complexity was  $O(hnd) + O(h^3)$  [22].

The  $k$ -means algorithm contains four steps: 1) random initialization; 2) assignment of each sample to the nearest center; 3) obtain the center of the clusters by averaging each cluster; and 4) if the center of the clusters does not vary, the cluster had been selected properly; otherwise, returned to step 2 [22], [24]. Since the algorithm iterated  $T$  times, the complexity in the Nyström was  $O(ndkT)$  [22]. The Nyström approximation requires  $O(hn)$  to store the kernel matrix in memory space [22].

The  $n$  samples in PSCC were divided equally between  $p$  processors. The Sparse method calculated only the similarity

TABLE VII  
COMPLEXITY

Time complexity (flam)						Memory cost
Algorithms	Unsupervised spectral regression (USR)				Total time complexity	
	Graph construction	Response generation	Embedding functions	K-means clustering	$O(n^2(s+\log n))+O(dk_1(c_1+\delta))+O(2dk_1ns)+O(5dk_{2m})+O(ndkT)$	$O(ns+np+nd+md)$
Proposed method	$O(n^2(s+\log n))$	$O(dk_1(c_1+\delta))$	$O(2dk_1ns)+O(5dk_{2m})$	$O(ndkT)$		
	Construct the similarity matrix	Perform eigen-decomposition				
Nyström	$O(hnd)$	$O(h^3)+O(hnd)$		$O(ndkT)$	$O(hnd)+O(h^3)+O(hnd)+O(ndkT)$	$O(hn)$
PSCC	$O(n^2/d)+O(n\log t/p)$	$O(l^3)+O(nl)+O(nt) \times O(h-k)$		$O(ndkT)$	$O(n^2/d)+O(n\log t/p)+O(l^3)+O(nl)+O(nt) \times O(h-k)+O(ndkT)$	$O(nt)$

between each sample and the  $t$  samples closest to it, according to bottleneck problem. Its required calculation time and memory were  $O(n^2 \log t)$  and  $O(nt)$ , respectively [22]. Constructing the parallel similarity matrix had a time complexity of  $O\left(n^2 d/p + n^2 \log t/p\right)$  [22]. The MATLAB, including the FORTRAN routine of ARPACK, was used to establish eigen-decomposition in a sparse matrix [22], [67]. The instability of PSCC is only caused by initialization of the ARPACK eigen-decomposition on large-scale sparse matrix and  $k$ -means clustering. After Arnoldi factorization of the  $l$  times to find the  $k$  eigen-values for eigen-decomposition, the time complexity was  $O(l^3) + O(nt) \times O(h-k) + O(nt) + O(nl)$  and space complexity was  $O(nt) + O(nl)$  [22].

SR utilizes the regularized least square for finding the projective functions, which is an essential step for supervised and unsupervised cases. If  $m$  in (8) is not too large, the regularized least-square problem in (8) can be directly solved using linear (10). Computing  $XX^T$  has a complexity of  $O(1/2nm^2)$  [18]. The  $\alpha I + XX^T$  matrix is positive and finite; therefore, the application of Gaussian elimination requires flam as  $O(1/6m^3)$  to solve linear equations in (10) [18], [68].

For high-dimensional data, the regularized least-square problem in (8) can be solved using the LSQR iterative algorithm. This algorithm was used to solve large-scale sparse linear equations. In each iteration, LSQR must calculate the generated vectors  $X_p$  and  $X_q^T$  [18]. The rest of the work load, costed  $O(3n) + O(5m)$  for LSQR, in each iteration; therefore, LSQR had a complexity of  $O(2mn) + O(3n) + O(5m)$ , in each iteration. If it required  $k_2$  iterations to stop, the time cost was  $O(k_2(2mn + 3n + 5m))$ . The time cost for  $d$  projective functions was  $O(dk_2(2mn + 3n + 5m))$  [18]. In addition to matrix  $X$ , LSQR required as much memory as  $O(n) + O(2m)$  [18]. The memory cost was  $O(mn + n + 2m + dm)$  in this step, where  $O(dm)$  is the memory cost required to store the Projective functions [18].

For the unsupervised case, matrix  $W$  is the  $c_1$  nearest-neighborhood graph and has about  $nc_1$  nonzero inputs to calculate the  $d$  eigen-vectors related to the  $d$ -largest eigen-value [18]. The computational cost is  $O(dk_1n(c_1 + 8))$ , where  $k_1$  is the number of iterations in the Lanczos algorithm. The memory required for this step is the memory required to store matrix  $W$  and  $d$  eigen-vectors [18].

Table VII shows the time and memory costs. For simplicity, only the most costly parts are presented [18].

In Table VII, parameter  $n$  represents the number of data which determines the number of the filtered images resulting from applying the Gabor filter bank on the original image and.  $u, s, d, c_1, k_1$ , and  $k_2$  are the number of features, the average of the nonzero features for a sample ( $s \leq n$ ), dimension of the projective function, number of nearest neighbors, number of iterations of the Lanczos algorithm and number of iterations of the LSQR in the algorithm, respectively [18].

Table VII compares segmentation complexity of the proposed algorithm and the Nyström and PSCC algorithms. Both the PSCC and Nyström algorithms require the additional cubic complexity cost to solve the eigen-problem. SR does not require a solution to eigen-decomposition, which saves a considerable amount of time and memory.

It can be concluded that SR can be used to explore the sparseness of the data matrix, which saves both time and memory. SR can be applied to the data as long as the memory can include matrix  $X$ . Even if matrix  $X$  is so large that the memory cannot include it, SR can be applied to the data using a reasonable I/O disk [18]. SR has this capability because, in each LSQR iteration, only vectors  $X_p$  and  $X_q^T$  in SR must be calculated, which can be done by applying the  $X$  and  $X^T$  values stored on the disk [18].

In SR, the time complexity is linear in Responses Generation and Embedding Functions, but LDA and LPP have cubic complexities. Assuming that Graph constructing is similar in all



TABLE VIII  
RUNNING TIME IN MATLAB

	Proposed method				PSCC				Nyström			
	Gabor filter bank+ feature extraction (s)	Feature learning (USR) (s)	Clustering (s)	Total (s)	tNN time (s)	EVD time (s)	KM time (s)	Total (s)	tNN time (s)	EVD time (s)	KM time (s)	Total (s)
2-class	0.96	0.11	0.29	1.36	40.72	18.16	0.13	59.01	0.34	1.54	0.16	2.04
3-class	1.01	0.13	0.34	1.48	40.70	22.79	0.19	63.68	0.36	1.62	0.22	2.20
4-class	1.05	0.14	0.37	1.56	41.74	22.62	0.28	64.64	0.40	2.61	0.30	2.26
5-class	1.12	0.17	0.40	1.69	40.64	21.30	0.39	62.33	0.84	3.29	0.40	2.40

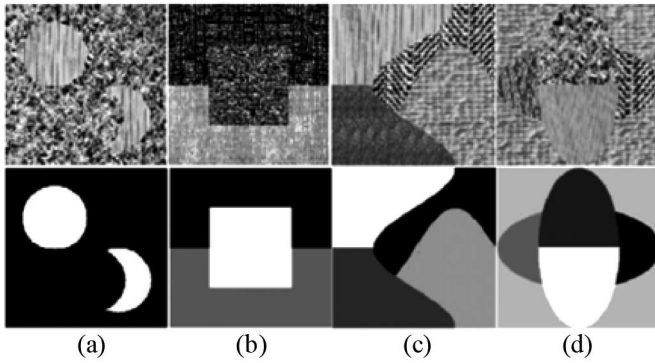


Fig. 14. Images for (a) 2 class; (b) 3 class; (c) 4 class; and (d) 5 class [22].

methods and ignoring it, it can be said that SR reduces time complexity from cubic to linear, comparing to these methods [18].

#### F. Running Time in MATLAB

Table VIII compares the times required for the program to run in MATLAB to segment an image using the proposed method, Nyström, and PSCC. The timings were compared for 2-class, 3-class, 4-class, and 5-class images of  $256 \times 256$  pixels in size [22], which are shown in Fig. 14(a)–(d), respectively. The run time per iteration was tested on a PC with core i5 CPU 1.6 GHz, 4-GB installed memory and Windows 7 operating system. The Nyström and PSCC running times were calculated for  $t$ -nearest neighbors using the sparse similarity matrix (tNN\_time),  $k$ -means clustering (KM\_time), and the time required to compute eigen-value decomposition (EVD\_time). The clustering times in Table VIII for the proposed method include the time required to add the spatial adjacency of the pixels and applying the  $k$ -means algorithm to the images. In Nyström, the computational cost of the similarity matrix depends on the number of random samples. Larger numbers of sampling points increased the eigen-decomposition computational time and  $O(h^3)$  became the dominant term in Table VII.

The contrast similarity matrix was dominant for determining the run time of the PSCC algorithm. For Nyström, eigen-decomposition primarily determined run time. Obtaining the eigen-vectors in PSCC required multiple Arnoldi iterations, and the number of iterations depended on the number of nearest neighbors.

When  $h$  is large enough, Nyström will yield stable and suitable solutions; however, higher numbers of samples may deteriorate the clustering accuracy because it increases the amount of noisy data. Higher numbers of nearest neighbors increase the clustering accuracy of PSCC and lower numbers yield weaker results. Since the relationships between points are not included in algorithms with lower numbers of nearest neighbors, the accuracy of these algorithms decreased. Generally, when the number of nearest neighbors was selected properly, the spectral clustering algorithm of PSCC showed relatively better performance than Nyström.

The time PSCC spent on sparse eigen-decomposition and  $k$ -means clustering is much lower than the time spent on constructing the similarity matrix. The time for eigen-value decomposition for the 2-class, 3-class, 4-class, and 5-class images in PSCC was 18.16, 22.79, 22.62, and 21.30 s, respectively. These calculations were not required in USR, which saved significant time over Nyström and PSCC. In the proposed algorithm, feature learning had the shortest run time; however, Tables III, V, and VI and Fig. 13 show that the performance of clustering improved considerably. These results show the superiority of USR in term of computational cost.

#### VII. CONCLUSION

Phase I proposed a new algorithm based on kurtosis of the energy of the curvelet coefficients for segmentation of textures in SAR images. The mathematical equations were extracted and implemented on SAR images using MATLAB. A new parameter estimation technique (KCE) was proposed to describe the texture of each area. KCE was shown to produce better results for SAR image segmentation than the SCE segmentation model. KCE was also better than KWE because it extracts more statistical data from the textures of the image. This requires a higher statistical order for kurtosis and is a unique feature of curvelet transform for recognizing edges of curves.

The texture of the SAR image can be defined using KCE with fewer coefficients than KWE. The results of experimentation indicate that KCE performs better than SCE or KWE for segmentation of SAR images for different sensors. Kurtosis was compared with ML for estimation of  $\beta$  and showed that using kurtosis has better performance than the ML estimator. The results showed that the estimation from kurtosis improved segmentation and lacked the disadvantages, such as sensitivity



to initial dose, high computational time, and converge to the local maxima, which can be found in the ML technique.

Phase II used a Gabor filter bank, a multichannel filtering-based analysis, to extract textural features. Gabor filters, which are joint spatial/spatial-frequency representations of texture, and HVS most-friendly parameters were selected for localization, which decreased the computational load and complexity. The images were decomposed into several filtered images and the feature was extracted. The extracted features were learnt using USR to improve their discrimination power and the  $k$ -means algorithm was used to cluster the images.

The proposed algorithm was then compared with other algorithms. In Phase II, the algorithm was compared for clustering with the results from the Nyström and PSCC algorithms. Comparisons were made using simulated and real SAR images. The results indicated that the accuracy of the proposed method for simulated images and three real SAR images was higher than that from the other algorithms. On the other hand, the time has been improved in contrast to two other algorithms. PSCC is a parallel method and the proposed method is a serial method. Since the latter does not require calculation of eigen-decomposition, it has a lower computational load which decreases time and memory computational complexity. It is also applicable to large-size data and can use different regularizers.

The proposed algorithm was compared with algorithms based on color, texture, intensity, and a combination of these features on images of the random texture, histology datasets, and simulated SAR image. The results showed that the proposed algorithm performed better than the other algorithms. The results were compared for NMI and accuracy for learned and unlearned features to understand the effect of feature learning on improvement of the algorithm. The comparison was repeated between different subimages extracted from the segmented image in the first phase. The comparisons showed that the learning feature improved clustering performance with or without noise.

## REFERENCES

- [1] 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.
- [2] M. Liao, T. Balz, L. Zhang, Y. Pei, and H. Jiang, "Analyzing TerraSAR-X and Cosmo-SkyMed high-resolution SAR data of urban areas," in *Proc. ISPRS Workshop HR Earth Imag. Geospatia Inf.*, Hannover, Germany, 2009, pp. 2–5.
- [3] M. R. Inggs and R. T. Lord, "Applications of satellite imaging radar," South African Institute of Electrical Engineers (SAIEE), South Africa, 2000.
- [4] F. Peng, H. Li, B. Cai, D. Deng, and Y. Liang, "Study on transmitting mode and imaging algorithm of MIMO-SAR," *Intelligent Science Intelligent Data Engineering*. New York, NY, USA: Springer, 2012, vol. 7202, pp. 745–752.
- [5] A. Alonso-Gonzalez, C. Lopez-Martinez, and P. Salembier, "Filtering and segmentation of polarimetric SAR data based on binary partition trees," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 2, pp. 593–605, Feb. 2012.
- [6] S. Parrilli, M. Poderico, C. Vincenzo, and L. Verdoliva, "A nonlocal SAR image denoising algorithm based on LLMMSE wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 2, pp. 606–616, Feb. 2012.
- [7] T. Esch, A. Schenk, T. Ullmann, M. Thiel, A. Roth, and S. Dech, "Characterization of land cover types in TerraSAR-X images by combined analysis of speckle statistics and intensity information," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 1911–1925, Jun. 2011.
- [8] K. Ersahin, I. G. Cumming, and R. K. Ward, "Segmentation and classification of polarimetric SAR data using spectral graph partitioning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 1, pp. 164–174, Jan. 2010.
- [9] D. Clausi and M. Ed Jernigan, "Designing Gabor filters for optimal texture separability," *Pattern Recognit.*, vol. 33, no. 11, pp. 1835–1849, Nov. 2000.
- [10] G. Changlai, "Image-denoising method based on wavelet transform and mean filtering," *Opto-Electron. Eng.*, vol. 34, no. 1, pp. 72–75, 2007.
- [11] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C2 singularities," *Commun. Pure Appl. Math.*, vol. 57, no. 2, pp. 219–266, 2004.
- [12] J. L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, Jun. 2002.
- [13] F. Colonna and G. Easley, "Generalized discrete radon transforms and their use in the Ridgelet transform," *J. Math. Imag. Vis.*, vol. 23, pp. 145–165, Sep. 2005.
- [14] E. J. Candès, "Ridgelets: Theory and applications," Dept. Statistics, Stanford Univ., Stanford, CA, USA, 1998.
- [15] J. L. Starck, E. Candès, and D. L. Donoho, "Astronomical image representation by the curvelet transform," *Astron. Astrophys. (A & A)*, Tech. Rep. DAPNIA-2002-66, vol. 398, no. 2, pp. 785–800, Feb. 2003.
- [16] M. O. Ulfarsson, J. R. Sveinsson, and J. A. Benediktsson, "Speckle reduction of SAR images in the curvelet domain," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS'02)*, Toronto, Canada, Jun. 2002, vol. 1, pp. 315–317.
- [17] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognit.*, vol. 24, no. 12, pp. 1167–1186, Nov. 1991.
- [18] D. Cai, "Spectral regression: A regression framework for efficient regularized subspace learning," Ph.D. dissertation, Comput. Sci. Dept., Urbana-Champaign (UIUC) Univ., Champaign, IL, USA, 2009.
- [19] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *Proc. 11th IEEE Int. Conf. Comput. Vis. (ICCV'07)*, Rio de Janeiro, Brazil, 2007, pp. 1–8.
- [20] Y. Bengio, J. F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for LLE isomap, MDS, eigenmaps, and spectral clustering," in *Advances Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2003, pp. 177–184.
- [21] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.
- [22] S. Gou, X. Zhuang, H. Zhu, and T. Yu, "Parallel sparse spectral clustering for SAR image segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 4, pp. 1949–1963, Aug. 2013.
- [23] D. Cai, X. He, and J. Han, "Spectral regression: A unified approach for sparse subspace learning," in *Proc. 17th IEEE Int. Conf. Data Mining (ICDM'07)*, 2007, pp. 73–82.
- [24] S. Naotoshi, (2006). *Texture Segmentation Using Gabor Filters, ENEE731 Project 1* [Online]. Available: <http://note.sonots.com/index.php?SciSoftware%2FGaborTextureSegmentation>
- [25] D. Cai, X. He, and J. Han, "Isometric projection," in *Proc. Nat. Conf. Artif. Intell.*, 2007, vol. 22, pp. 528–533.
- [26] D. Cai, X. He, and J. Han, "Spectral regression: A unified subspace learning framework for content-based image retrieval," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 403–412.
- [27] D. Cai, X. He, and J. Han, "Efficient kernel discriminant analysis via spectral regression," in *Proc. 7th IEEE Int. Conf. Data Mining (ICDM'07)*, 2007, pp. 427–432.
- [28] D. Cai, X. He, and J. Han, "SRDA: An efficient algorithm for large-scale discriminant analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 1–12, Jan. 2008.
- [29] D. Cai, X. He, and J. Han, "Speed up kernel discriminant analysis," *VLDB J.*, vol. 20, no. 1, pp. 21–33, 2011.
- [30] D. Cai, X. He, W. V. Zhang, and J. Han, "Regularized locality preserving indexing via spectral regression," in *Proc. 16th ACM Conf. Inf. Knowl. Manage.*, 2007, pp. 741–750.
- [31] Q. Wang and K. L. Boyer, "Feature learning by multidimensional scaling and its applications in object recognition," in *Proc. 26th Conf. Graph. Patterns Images (SIBGRAPI'13)*, 2013, pp. 8–15.
- [32] A. Asuncion and D. J. Newman, (2007). *UCI Machine Learning Repository*, Univ. California, Sch. Inf. Comp. Sci., Irvine, CA, USA [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>

- [33] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *J. Opt. Soc. Amer. A*, vol. 7, no. 5, pp. 923–932, May 1990.
- [34] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture in two nonstriate visual areas 18 and 19 of the cat," *J. Neurophysiol.*, vol. 7, no. 5, pp. 229–289, Mar. 1965.
- [35] F. W. Campbell and J. J. Kulikowski, "Orientational selectivity of the human visual system," *J. Physiol.*, vol. 187, no. 2, pp. 437–445, Nov. 1966.
- [36] A. R. Rao and G. L. Lohse, "Identifying high level features of texture perception," *CVGIP Graph. Models Image Process.*, vol. 55, no. 3, pp. 218–233, May 1993.
- [37] K. Kayabol and J. Zerubia, "Unsupervised amplitude and texture classification of SAR images with multinomial latent model," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 561–572, Feb. 2013.
- [38] P. P. Raghu and B. Yegnanarayana, "Multispectral image classification using Gabor filters and stochastic relaxation neural network," *Neural Netw.*, vol. 10, no. 3, pp. 561–572, 1997.
- [39] P. P. Raghu and B. Yegnanarayana, "Segmentation of Gabor-filtered textures using deterministic relaxation," *IEEE Trans. Image Process.*, vol. 5, no. 12, pp. 1625–1636, Dec. 1996.
- [40] J. Nunez *et al.*, "Multiresolution-based image fusion with additive wavelet decomposition," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1204–1211, May 1999.
- [41] G. Plonka and J. Ma, "The curvelet transform," *IEEE Signal Process. Mag.*, vol. 27, no. 2, pp. 118–133, Mar. 2010.
- [42] G. Zhou, Y. Cui, Y. Chen, J. Yang, H. Rashvand, and Y. Yamaguchi, "Linear feature detection in polarimetric SAR, curvelet transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 4, pp. 1453–1463, Apr. 2011.
- [43] E. Uslu and S. Albayrak, "Curvelet-based synthetic aperture radar image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 6, pp. 1071–1075, Jun. 2014.
- [44] Z. Tirandaz and G. Akbarizadeh, "Unsupervised texture-based SAR image segmentation using spectral regression and Gabor filter bank," *J. Indian Soc. Remote Sens.*, pp. 1–10, 2015.
- [45] G. Akbarizadeh and Z. Tirandaz, "Segmentation parameter estimation algorithm based on curvelet transform coefficients energy for feature extraction and texture description of SAR images," *Electron. Ind. Quart.*, vol. 5, no. 1, pp. 53–63, May 2014 (in Persian).
- [46] J. Shen, Q. Li, and G. Erlebacher, "Hybrid no-reference natural image quality assessment of noisy, blurry, jpeg2000, and jpeg images," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2089–2098, Aug. 2011.
- [47] F. Gómez and E. Romero, "Rotation invariant texture characterization using a curvelet based descriptor," *Pattern Recognit. Lett.*, vol. 32, pp. 2178–2186, Oct. 2011.
- [48] C. Li, C. Kao, J. Gore, and Z. Ding, "Minimization of region-scalable fitting energy for image segmentation," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1940–1949, Oct. 2008.
- [49] O. Michailovich, Y. Rath, and A. Tannenbaum, "Image segmentation using active contours driven by the Bhattacharyya gradient flow," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2787–2801, Nov. 2007.
- [50] C. Li, C. Xu, C. Gui, and M. D. Fox, "Level set evolution without re-initialization: A new variational formulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR'05)*, 2005, vol. 1, pp. 430–436.
- [51] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized gaussian density and Kullback–Leibler distance," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 146–158, Feb. 2002.
- [52] J. Zhang, T. Tan, and L. Ma, "Invariant texture segmentation via circular gabor filter," in *Proc. 16th Int. Conf. Pattern Recognit. (ICPR'02)*, 2002, vol. 2, pp. 901–904.
- [53] M. Omran, A. P. Engelbrecht, and A. Salman, "Particle swarm optimization method for image clustering," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 19, no. 3, pp. 297–321, May 2005.
- [54] P. Hanchuan, L. Fulmi, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
- [55] M. T. McCann, D. G. Mixon, M. Fickus, C. A. Castro, J. A. Ozolek, and J. Kováček, (2013). *Images as Occlusions of Textures: A Framework for Segmentation* [Online]. Available: [http://www.jelena.ece.cmu.edu/repository/tr/13\\_McCannMFCOK/13\\_McCannMFC%OK.html](http://www.jelena.ece.cmu.edu/repository/tr/13_McCannMFCOK/13_McCannMFC%OK.html)
- [56] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.
- [57] C. M. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Quebec, QC, Canada, 2002, vol. 4, pp. 150–155.
- [58] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
- [59] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
- [60] M. T. McCann, D. G. Mixon, M. C. Fickus, C. A. Castro, J. A. Ozolek, and J. Kováček, "Images as occlusions of textures: A framework for segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2033–2046, May 2014.
- [61] J. A. Ozolek and C. A. Castro, *Teratomas Derived from Embryonic Stem Cells as Models for Embryonic Development, Disease, and Tumorigenesis*. Vellore, India: INTECH Open Access Publisher, 2011.
- [62] X. Zhang, L. Jiao, F. Liu, L. Bo, and M. Gong, "Spectral clustering ensemble applied to SAR image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2126–2136, Jul. 2008.
- [63] M. B. Salah, A. Mitiche, and I. B. Ayed, "Multiregion image segmentation by parametric kernel graph cuts," *IEEE Trans. Image Process.*, vol. 20, no. 2, pp. 545–557, Feb. 2011.
- [64] M. Gong, Y. Liang, J. Shi, W. Ma, and J. Ma, "Fuzzy c-means clustering with local information and kernel metric for image segmentation," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 573–584, Feb. 2013.
- [65] D. Xian, 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.
- [66] J. Feng, L. C. Jiao, X. Zhang, M. Gong, and T. Sun, "Robust non-local fuzzy c-means algorithm with edge preservation for SAR image segmentation," *Signal Process.*, vol. 93, no. 2, pp. 487–499, Jun. 2013.
- [67] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK User's Guide*. Philadelphia, PA, USA: SIAM, 1998.
- [68] G. W. Stewart, *Matrix Algorithms Vol. 1: Basic Decompositions*. Philadelphia, PA, USA: SIAM, 1998.



**Zeinab Tirandaz** was born in Ahvaz, Iran, on June 24, 1989. She received the B.S. degree in biomedical engineering from Islamic Azad University of Dezfoul (IAUD), Dezfoul, Iran, in 2011. She is currently pursuing the M.S. degree in electrical and electronics engineering at Shahid Chamran University of Ahvaz (SCU), Ahvaz, Iran.

Her research interests include synthetic aperture radar image analysis and understanding, machine vision, image processing, and machine learning.



**Gholamreza Akbarizadeh** was born in Shiraz, Iran, on July 14, 1981. He received the B.S. degree from Khajeh-Nassir Tousei University of Technology (KNTU), Tehran, Iran, in 2003, and the M.S. and Ph.D. degrees from Iran University of Science and Technology, Tehran, Iran, in 2005 and 2011, respectively, all in electrical and electronics engineering.

From 2003 to 2011, he has worked with the DSP R&D Research Laboratory, Tehran, Iran, as a Senior Researcher. He is currently an Assistant Professor and Faculty Member with the Department of Electrical Engineering, Shahid Chamran University (SCU), Ahvaz, Iran. He has authored more than 50 published papers in different international electrical engineering conferences and journals.

Dr. Akbarizadeh is a member of some international scientific societies such as the IEEE and also the Board of Iranian Society of Machine Vision and Image Processing (ISMVIP). He is the Reviewer of some international journals such as the IEEE TGRS, the IEEE JSTARS, the IEEE GRSL, Measurement, Information Fusion, and NCA.

His research interests include machine vision, image processing, remote sensing analysis, and geographic information system (GIS) techniques for earth surface mapping.