Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 18 (2008) 455–461
Progress in Natural Science
www.elsevier.com/locate/pnsc

Artificial immune kernel clustering network for unsupervised image segmentation

Wenlong Huang *, Licheng Jiao

Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China

Received 10 August 2007; received in revised form 29 October 2007; accepted 30 October 2007

Abstract

An immune kernel clustering network (IKCN) is proposed based on the combination of the artificial immune network and the support vector domain description (SVDD) for the unsupervised image segmentation. In the network, a new antibody neighborhood and an adaptive learning coefficient, which is inspired by the long-term memory in cerebral cortices are presented. Starting from IKCN algorithm, we divide the image feature sets into subsets by the antibodies, and then map each subset into a high dimensional feature space by a mercer kernel, where each antibody neighborhood is represented as a support vector hypersphere. The clustering results of the local support vector hyperspheres are combined to yield a global clustering solution by the minimal spanning tree (MST), where a predefined number of clustering is not needed. We compare the proposed methods with two common clustering algorithms for the artificial synthetic data set and several image data sets, including the synthetic texture images and the SAR images, and encouraging experimental results are obtained.

Keywords: Artificial immune network; Kernel mapping; Nonsubsampled contourlet transform; Unsupervised image segmentation

1. Introduction

Tax and Duin [1] and Schőlkopf et al. [2] proposed a kernel method, also known as the support vector domain description, to characterize the support of a high dimensional distribution. Intuitively, the support vector domain description computes the smallest hypersphere in feature space enclosing the image of the input data. In this paper, we introduce the support vector domain description (SVDD) into a novel structural adaptation artificial immune network for image segmentation. As the total number of pixels in the original image is usually huge, which cannot be directly used as antigens (train pattern), we firstly segment the original image into regions using the watershed segmentation algorithm. Then the mean features energy of each watershed region is calculated, which is regarded as the antigens of the immune kernel clustering network (IKCN). A set of features, including the nonsubsampled contourlet transform (NSCT) [3] and the gray level co-occurrence matrix (GLCM) [4], are extracted from the image. Starting from IKCN, the input image features are firstly divided into subsets by the antibodies, and then each subset is mapped into a hypersphere in a high dimensional feature space by a mercer kernel. Finally, the clustering results of the local support vector hyperspheres are combined to yield a global clustering solution by MST [5] in graph theory, which can automatically cluster the antibody obtained in the output space without a predefined number of clustering. Such an immune kernel approach can deal with problems with unevenly distributed samples. Another advantage of the proposed method is that it can simplify the computation of support vector domain description as well as facilitate a parameter tuning task. Furthermore, noise patterns can be easily detected by the boundary curves

* Corresponding author. Tel.: +86 29 88209788; fax: +86 29 88201023.
E-mail address: yykjhuang@yahoo.com.cn (W. Huang).

1002-0071/$ - see front matter © 2007 National Natural Science Foundation of China and Chinese Academy of Sciences. Published by Elsevier Limited and Science in China Press. All rights reserved.
doi:10.1016/j.pnsc.2007.10.012

obtained from support vectors for each antibody, which is represented as a support vector hypersphere.

2. Mathematical preliminaries

2.1. Watershed segmentation

In this work, we use the well-known watershed segmentation [6] to partition an image into nonoverlapping regions. Pixels in a watershed region are homogeneous in the feature space. We then introduce the basic concept of watershed segmentation as follows:

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

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

If the watershed regions are too large, one region may contain more than one focused subject in the image. If the number of regions is too small, texture feature in the region may not be homogeneous, and the computational complexity will increase. In our design, we adopted two parameters: $r = 4$ and $h = 4$. Using this setting, the number of regions was about 1400 in each 256×256 image.

2.2. Feature extraction

We introduce the features used in our work, including the nonsubsampled contourlet transform and the gray level co-occurrence matrix. As for the NSCT features, the original image was firstly transformed into multi-channel images by using a NSCT decomposition, and then texture features have been extracted by moving a window, i.e. for any position in the feature images, mean deviation is estimated in its neighborhood. In this study, we used a small window (e.g. 15×15) to estimate "texture energy feature" from the transformed multi-channel images. As for the GLCM features, we use the entropy, energy, contrast and correlation features for the displacement is 1 and the window's size is 9. The feature vectors from NSCT had a set of 18 features (the resolution level is 3 and the frequency direction is 6), while the feature vector from GLCM had a set of 12 features.

2.3. Support vector domain description

Let $D = (x_i \in R^n, i = 1,2,\cdots,m) \subseteq X$, with $X \subseteq \Re^n$. Using a nonlinear transformation Φ from X to some high-dimensional feature space, it looks for the smallest enclosing hypersphere of radius R. This is described by the constraints

$$\min_{R,a,\xi_i} R^2 + C\sum_i \xi_i \tag{2}$$

Subject to the constraint $\|\Phi(x_i) - a\|^2 \leqslant R^2 + \xi_i$ where $\|\cdot\|$ is the Euclidean norm, a is the center of the hypersphere and $\xi_i \geqslant 0$ is the slack variable, and C is a constant controlling the penalty of noise. The Lagrangian is introduced to find the smallest sphere of radius R

$$L = R^2 - \sum_{i=1}^{m}\left(R^2 + \xi_i - \|\Phi(x_i) - a\|^2\right)\beta_i - \sum_{i=1}^{m}\xi_i\mu_i + C\sum_{i=1}^{m}\xi_i \tag{3}$$

Using (2) and (3), we may turn the constrained minimization of Lagrangian into the Wolfe dual form

$$\max_{\beta_i} W = \sum_{i=1}^{m}\Phi(x_i)^2\beta_i - \sum_{i=1}^{m}\sum_{j=1}^{m}\beta_i\beta_j\Phi(x_i)\cdot\Phi(x_j)$$
$$\text{subject to} \quad \sum_{i=1}^{m}\beta_i = 1 \quad \text{and} \quad 0 \leqslant \beta_i \leqslant C, \ \ i = 1,2,\ldots,m \tag{4}$$

The image of a point x_i with $\xi_i > 0$ and $\beta_i > 0$ lies outside the feature space hypersphere. A point with $\mu_i = 0$ is called a bounded support vector (BSV). A point x_i with $\xi_i = 0$ and $0 < \beta_i < C$ implies that its image $\Phi(x_i)$ lies on the surface of the feature space hypersphere. Such a point will be referred to as a support vector (SV). We compute the dot products $\Phi(x_i)\cdot\Phi(x_j)$ in (4) by an appropriate mercer kernel $G(x_i, x_j)$, and the Gaussian kernel is used in this study.

$$\Phi(x_i)\cdot\Phi(x_j) = G(x_i, x_j) = \mathrm{e}^{-\|x_i - x_j\|^2/\sigma^2}, \quad \sigma \in \Re \tag{5}$$

3. Immune kernel clustering network

The IKCN makes use of several features of the immune response, such as the clonal expansion of the most stimulated cells, death of the non-stimulated cells and the affinity maturation of the repertoire [7]. The network does not have a predefined number of antibodies, which will be determined dynamically based on immune principles. Finally, a MST is used to automatically determine the final number of clusters. The model can adaptively map input data into the antibody output space, which has a better adaptive network structure.

3.1. IKCN algorithm

The IKCN algorithm is summarized in the pseudocode presented below.

Step 1: Initialize randomly the antibodies in the network and define the parameters: $\eta(1)$, α, β, ε and σ. The number of initial antibodies could be set between $0.005N$ and $0.01N$, where N is the number of input data points.

Step 2: While not reached the convergence criterion do:

2.1. For each input pattern do:

(1) present all the antigens to the network;

(2) calculate the Euclidean distance between the antigens and the antibodies;

(3) calculate the neighborhood of each antibody and find the winner antibody;
(4) calculate the support vector domain description of each antibody and eliminate the outliers of it;
(5) update the antibodies using the intra-points of their SV hyperspheres.

2.2. If the current iteration is a multiple of β, then clone the winner antibody if necessary.
2.3. If the concentration level of a given antibody is smaller than a given threshold that can be set to be $0.01N$, then it is pruned from the network.

Step 3: Use the MST criterion proposed to segment the antibodies at the network output.

The convergence criterion used checks the stability of the network topology (the number of the antibodies). It is assumed that the network topology has reached stability if there was no variation in the number of antibodies during the last 5β iterations.

3.2. Antibody neighborhood

An antigen is recognized involving the finding of the most similar antibody to the given antigen, which is expressed through

$$Ag_j^{Ab_i} = \{Ab_i | \max_i(1/\|Ag_j - Ab_i\|),\ Ab_i \in Ab_{\text{set}}(k)\} \tag{6}$$

$Ag_j^{Ab_i}$ shows that the jth antigen is recognized by the ith antibody. The neighborhood of the ith antibody $Ab_{i\_}Ag_{\text{N}}$ is expressed through

$$Ab_{i\_}Ag_{\text{N}} = \{Ag_j^{Ab_q} | q = i, Ag_j \in Ag_{\text{set}}\} \tag{7}$$

where $Ab_{\text{set}}(k)$ is the current antibody set, and Ag_{set} is the antigen set.

3.3. Antibody competitiveness rule

The realization of antibody competitiveness is to choose the most stimulated antibody, i.e. the winner antibody Ab_{w}

$$Ab_{\text{w}} = \{Ab_i | \max(Ab_i^{\text{concentration}}),\ Ab_i \in Ab_{\text{set}}(k)\} \tag{8}$$

where $Ab_i^{\text{concentration}} = \text{sumnum}(Ab_{i\_}Ag_{\text{N}})$ is the concentration of the ith antibody, sumnum$(\cdot)$ is a count operator.

3.4. Antibody clone operator

In IKCN, network growing is inspired by the clonal selection principle, where the most stimulated antibody is selected for cloning. This process is described by

$$\begin{aligned} &\text{if round}(k, \beta) = 0, \text{ and } Af_{Ag_1}^{Ab_{\text{w}}} > \varepsilon, \text{ then} \\ &\quad Ab_{\text{set}}(k) = Ab_{\text{set}}(k) + Ab_{\text{w}} \\ &\text{else} \quad Ab_{\text{set}}(k) = Ab_{\text{set}}(k) \end{aligned} \tag{9}$$

where $Af_{Ag_1}^{Ab_{\text{w}}} = 1/\|Ab_{\text{w}} - Ag_1\|$, and Ag_1 is the antigen with the lowest affinity to Ab_{w}.

3.5. Antibody death operator

Antibody death operator is to realize the network pruning policy, which is defined as follows: if an antibody has its concentration level less than a presented value, for example one, longer than a specified length of time, then it can be deleted from the network.

3.6. Antibody network learning rule

The learning rule in IKCN is similar to that of SOFM neural networks [8]

$$Ab_i(k+1) = Ab_i(k) + \eta(k)(Ab_{i\_}Ag_{\text{N}}(p) - Ab_i(k)) \tag{10}$$

where $\eta(k)$ is the learning coefficient, $Ab_{i\_}Ag_{\text{N}}(p)$ is the pth antigen that is recognized by the ith antibody, and the antigen is enclosed in its SV hyperspheres. Eq. (10) shows the antibody updating rule used. In this step, the support vector domain description of each antibody is calculated firstly as described in Section 2. Antibodies are constantly moved in the direction of the recognized antigens, and one antibody is only adjusted by those antigens in its SV hyperspheres, where the BSVs are regarded as outliers and are eliminated. Thus the recognition bound of the antibody can be arbitrary shape rather than a sphere and the noise patterns can be easily detected.

3.6.1. Learning coefficient

In the above learning rule, the amount of the change is guided by the learning coefficient $\eta(k)$. The learning coefficient $\eta(k)$ is set as a large enough value in the beginning of learning, then after α iterations, it is exponentially decreased by a factor σ and is adaptively adjusted with each antibody affinity. This process is described by

$$\eta(k) = \eta(k)\exp(-\sigma(k-\alpha)) \tag{11}$$

$$\eta_{Ab_{i\_}Ag_{\text{N}}(p)}^{Ab_i} = \eta(k)(Af_{Ab_{i\_}Ag_{\text{N}}(p)}^{Ab_i} - Af_{\max}^{Ab_i})/(Af_{\min}^{Ab_i} - Af_{\max}^{Ab_i}) \tag{12}$$

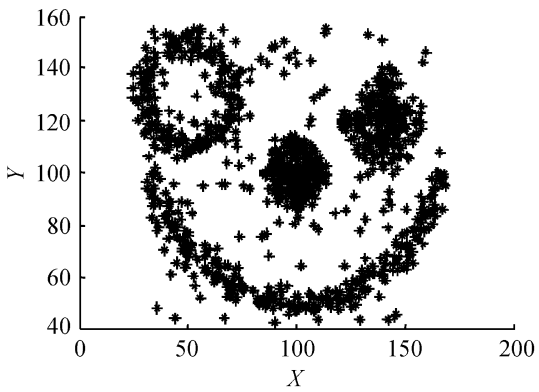

Fig. 1. Four different prototypes noise data (with 20% noise).

where $Af^{Ab_i}_{Ab_i-Ag_N(p)} = 1/\|Ab_i-Ag_N(p) - Ab_i\|$, $Af^{Ab_i}_{\max} = \max\{Af^{Ab_i}_{Ag(p)}, Ag(p) \in Ab_i-Ag_N\}$, and $Af^{Ab_i}_{\min} = \min\{Af^{Ab_i}_{Ag(p)}, Ag(p) \in Ab_i-Ag_N\}$.

3.6.2. Long-term memory coefficient

Based on the biological aspects [9], if an input pattern is something new then the pattern can be stored as long-term memory. With this strategy, we design a long-term memory $y_i(k)$ by utilizing the affinity of antibodies, which is expressed as

$$y_i(k) = \exp\left(-\frac{\|Ab_i(k) - Ab_d(k)\|^2}{2\delta(e_m)^2}\right) \tag{13}$$

Here, $\|Ab_i(k) - Ab_d(k)\|^2$ is the distance between the ith antibody Ab_i and the antibody Ab_d, which is the nearest antibody to Ab_i in the current antibody set; $\delta(e_m)$ is an emotion signal, for simplicity, it is set to be a constant one in this study. If $\|Ab_i(k) - Ab_d(k)\|^2$ is relatively large, then the antibody Ab_i may be something new and will have a chance to recognize more antigens, so it has a tendency to be stored as long-term memory, otherwise the input pattern will disappear or be replaced by the other antibodies.

With the adjusted Learning coefficient $\eta(k)$ and long-term memory coefficient $y_i(k)$, we have the new complete antibody learning rule given as

$$Ab_i(k+1) = Ab_i(k) + \eta^{Ab_i}_{Ab_i-Ag_N(p)} y_i(k)(Ab_i-Ag_N(p) - Ab_i(k)). \tag{14}$$

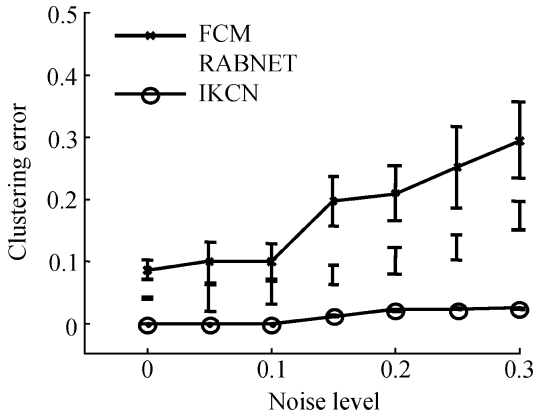

Fig. 2. The clustering error for the four different prototypes noise data (the mean and the standard deviation).

3.7. Defining the number of clusters

We propose the use of the MST, which defines a neighborhood relationship among antibodies and determines the optimal number of clusters. An inconsistent edge may be

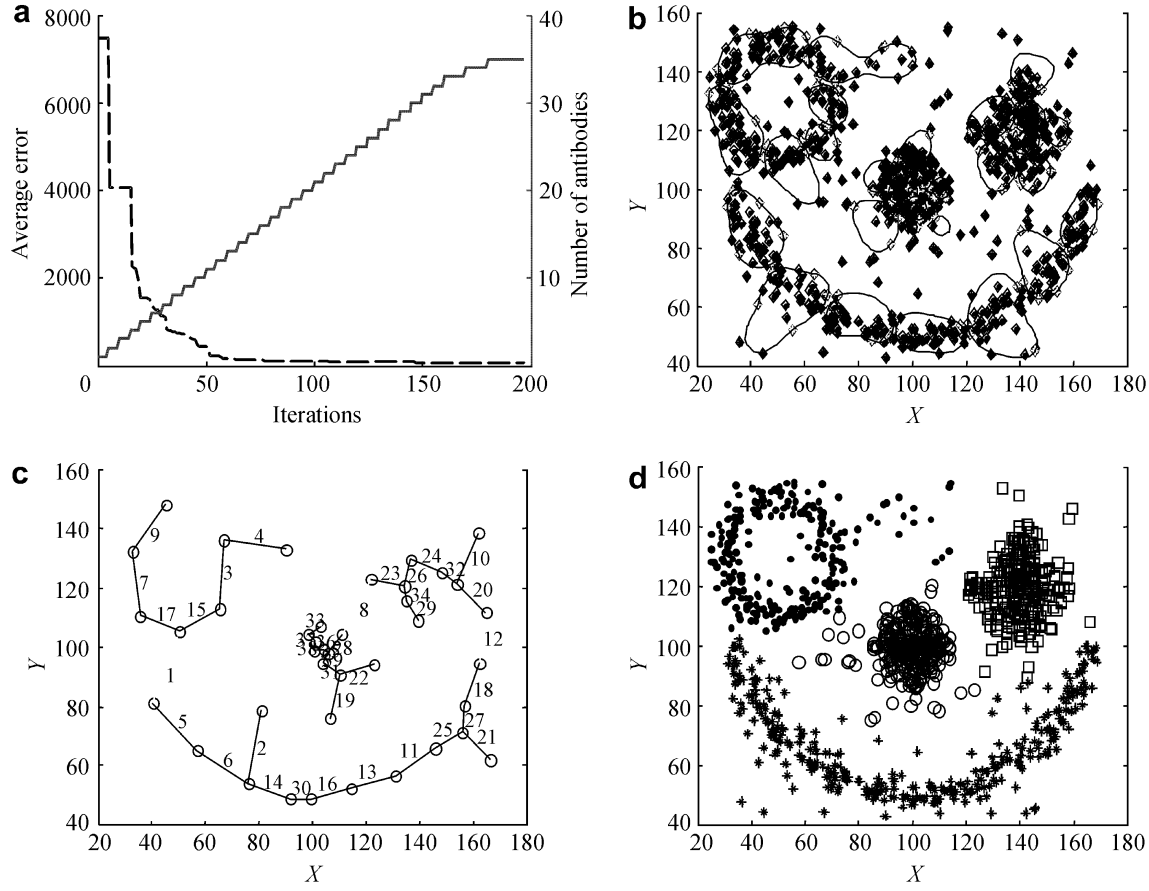

Fig. 3. Four different prototypes noise data obtained from our algorithm (with 20% noise).

determined as follows: if the length of an edge is greater than the average plus two standard deviations, then this edge is considered inconsistent.

4. Experimental results

Our algorithm has been tried on both the artificial synthetic data set and the image data sets. We have compared the performance of our proposed algorithms with two clustering algorithms: fuzzy C mean clustering (FCM) and RABNET [7]. One of the choice of the parameters used to run IKCN may be: $\beta = 2$, $\alpha = 30$, $\eta(1) = 0.2$, $\sigma = 0.02$, $\varepsilon = 0.05$.

4.1. Artificial synthetic data

In this experiment, we adopt a set of artificial synthetic data as shown in Fig. 1, which consists of 1000 points in 2D plane belonging to four different prototypes. This database is primarily used to compare the robustness of algorithms to noise. We added uniform a noise of increasing magnitude from 0.05 to 0.3 in steps of 0.05 to the original data set. In order to evaluate the robustness of the algorithms, all experiments were run 30 times on the data set and different values of the parameters were used in each run.

The obtained statistical clustering error plot is shown in Fig. 2. We can see that with the increase of the noise, our algorithm can successfully identify four clusters, while FCM and RABNET are too sensitive to noise. The number of antibodies and average error between the antibodies and the antigens with the iterations, the antibody support vector domain description, the antibody minimal spanning tree result and the final clustering result with 20% noise are shown in Fig. 3. In Fig. 3(a), the dashed lines correspond to the average error and the solid lines indicate the network size evolution. From Fig. 3(b), we can see that the support vector domain description reduces the effect of outliers (denoted as the diamond), so our algorithm can deal with outliers, making it robust with respect to noise in the data.

4.2. Image data

Experimental results on images containing various synthetic and SAR texture images have been obtained. To ensure unbiased comparison, the texture features were kept the same for both the algorithms. All synthetic textured

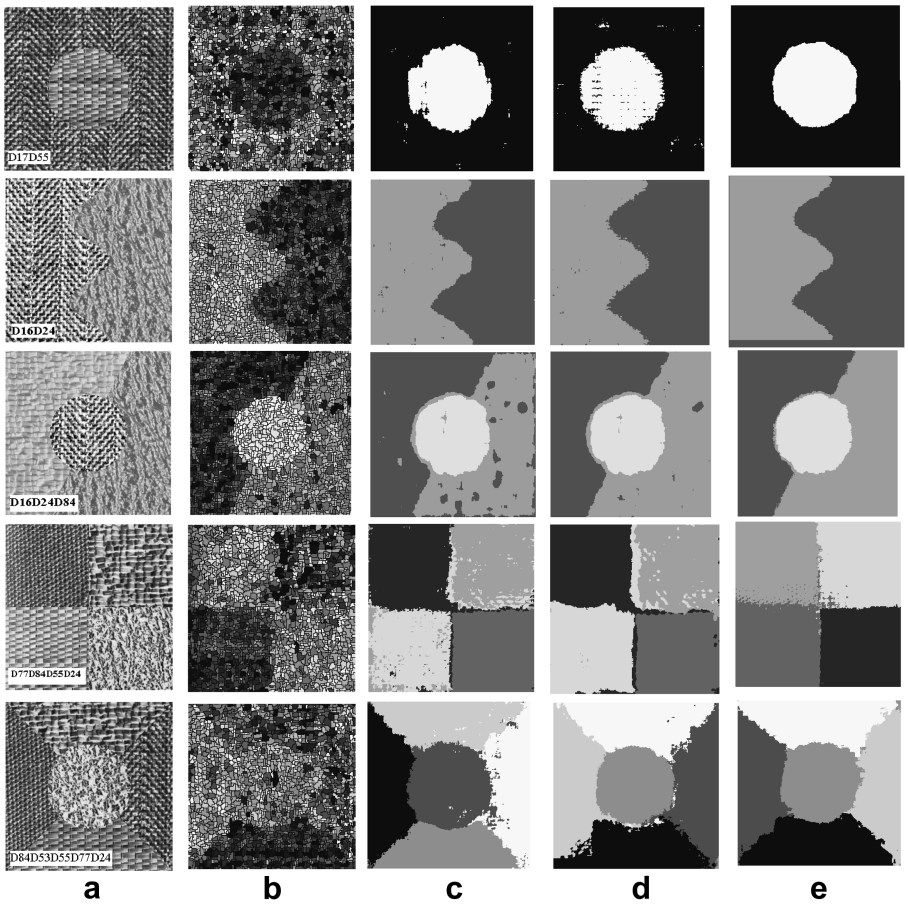

Fig. 4. The segmentation results of the synthesized images. (a) The original images; (b) watershed segmentation results; (c) FCM segmentation results; (d) RABNET segmentation results; (e) IKCN segmentation results.

images are from the Brodatz album. Iteration average error number of antibodies.

4.2.1. Synthetic images

Texture mosaic images were made by combining several different Brodatz textures. Two two-textures images D17D55 and D16D24, one three-textures image D16D24D84 and one four-textures image D77D84D55D24, one five-textures image D84D53D55D77D24, all of size 256×256, were used for the comparative study with other texture segmentation methods.

The original images and the watershed segmentation results are illustrated in Fig. 4. The segmentation results of FCM, RABNET and IKCN are illustrated in Fig. 4(c)–(e), respectively. We can see that the FCM and RABNET segmentation results have some noise, while the result used by IKCN algorithm is less speckled and smoother. The segmentation results with segmentation correctness measures are shown in Table 1. It is obvious that IKCN segmentation result is better than the other methods. As for the computational complexity, for a 256×256 image, FCM had lowest the computational time of about 21 s while IKCN had a higher computational time than FCM, which is about 32 s, and RABNET had the highest computational time of about 43 s. The test program has been implemented in MATLAB 7.0 and run on an Intel(R) Core(TM)2 1.86 GHz CPU computer.

Table 1
Comparison of correctness between FCM, RABNET and IKCN segmentation algorithms

	FCM (%)	RABNET (%)	IKCN (%)
D17D55	97.13	97.86	98.95
D16D24	97.95	98.85	99.34
D16D24D84	97.65	98.73	99.25
D77D84D55D24	89.96	92.46	96.85
D84D53D55D77D24	93.78	94.09	95.02

4.2.2. SAR images

The SAR image is generated by the coherent processing of the scattering signals, which results in a scene texture with an undesired multiplicative speckle noise. The speckle reduces drastically the ability to distinguish and classify the features in SAR images. So the processing of the SAR texture image is generally more involved. Fig. 5 shows results of various algorithms applied to three SAR amplitude images. Fig. 5(a) is the original SAR images. Fig. 5(b) is the resulting segmented images using watershed algorithm. Fig. 5(c)–(e) are the results of segmentation by FCM, RABNET and IKCN, respectively. It is obvious that IKCN outperforms FCM and RABNET either in the accuracy of texture classification or in the boundary localization or in the noise smoothness. Moreover, it can be clearly seen that our segmentation results are much closer to the ground truth.

5. Conclusion

In this paper, we have proposed a new image segmentation algorithm inspired by the artificial immune network and the support vector domain description of a data set. The novel artificial immune kernel network has a better adaptive network structure. With the antigens got from each watershed region, the immune kernel network can automatically cluster the antibody obtained in the output space without a predefined number of clustering. The new algorithm can deal with problems with unevenly distributed samples and it can simplify the computation of support vector domain description as well as facilitate a parameter tuning task. Our algorithm, whose empirical

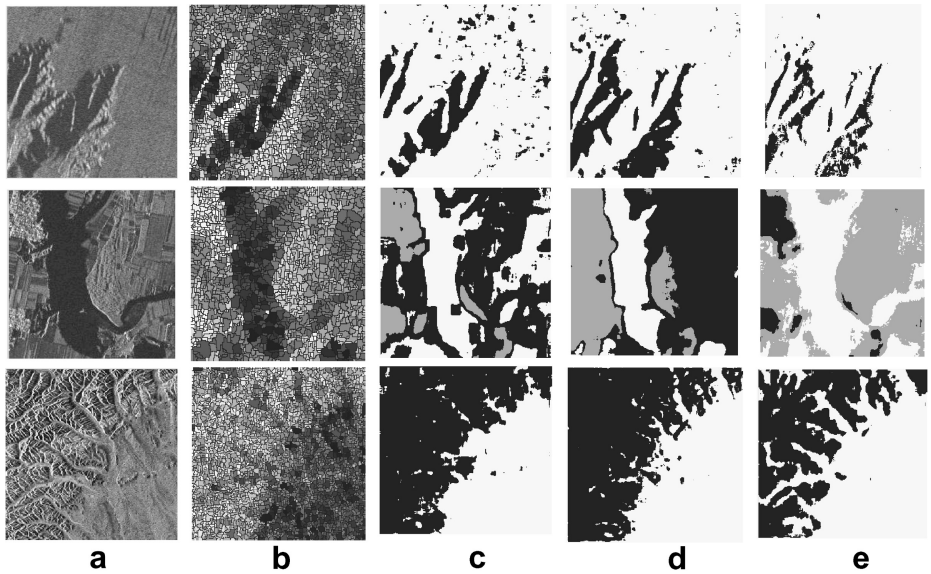

Fig. 5. The segmentation results of the SAR images. (a) The original images; (b) watershed segmentation results; (c) FCM segmentation results; (d) RABNET segmentation results; (e) IKCN segmentation results.

performance has been consistently verified, compares favorably against popular clustering algorithms, like FCM and RABNET, on the synthetic data set and the SAR image data sets. Results have shown that the proposed method actually reduces the effect of outliers and it is a good tradeoff between quality and computing time.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 60472084 and 60372050) and the National Program on Key Basic Research Projects of China (Grant No. 2001CB309403).

References

[1] Tax DMJ, Duin RPW. Support vector domain description. Pattern Recogn Lett 1999;20:1191–9.

[2] Schőlkopf B, Williamson RC, Smola AJ, et al. Support vector method for novelty detection. In: Advances in neural information processing systems 12: proceedings of the 1999 conference, USA: MIT Press; 2000. p. 582–88.

[3] Cunha AL, Zhou J, Do MN. The nonsubsampled contourlet transform: theory, design, and applications. IEEE Trans Image Process 2006;15(10):3089–101.

[4] Gotlieb CC, Kreyszig HE. Texture descriptors based on cooccurrence matrices. Comput Vision Graph 1990;51:70–86.

[5] Leclerc B. Minimum spanning trees for tree metrics: abridgements and adjustments. J Classif 1995;12:207–41.

[6] Vincent L, Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans Pattern Anal 1991;13(6):583–98.

[7] Knidel H, de Castro LN, Von Zuben FJ, et al. RABNET: a real-valued antibody network for data clustering. In: GECCO'05, Washington, DC, USA, June 25–29, 2005. p. 371–2.

[8] Kohonen T. The self-organizing map. Neurocomputing 1998;21:1–6.

[9] Lanprecht R, LeDoux J. Structural plasticity and memory. Nat Rev Neurosci 2004;5:45–54.