Chinese Journal of Aeronautics, (2014),27(1): 102–109

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

A salient edges detection algorithm of multi-sensor images and its rapid calculation based on PFCM kernel clustering

Xu Guili *, Zhao Yan, Guo Ruipeng, Wang Biao, Tian Yupeng, Li Kaiyu

College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Received 25 January 2013; revised 26 April 2013; accepted 24 July 2013
Available online 8 December 2013

KEYWORDS

Data reduction;
Edge detection;
Fuzzy clustering;
Kernel clustering;
Possibility fuzzy C-means (PFCM)

Abstract Multi-sensor image matching based on salient edges has broad prospect in applications, but it is difficult to extract salient edges of real multi-sensor images with noises fast and accurately by using common algorithms. According to the analysis of the features of salient edges, a novel salient edges detection algorithm and its rapid calculation are proposed based on possibility fuzzy C-means (PFCM) kernel clustering using two-dimensional vectors composed of the values of gray and texture. PFCM clustering can overcome the shortcomings that fuzzy C-means (FCM) clustering is sensitive to noises and possibility C-means (PCM) clustering tends to find identical clusters. On this basis, a method is proposed to improve real-time performance by compressing data sets based on the idea of data reduction in the field of mathematical analysis. In addition, the idea that kernel-space is linearly separable is used to enhance robustness further. Experimental results show that this method extracts salient edges for real multi-sensor images with noises more accurately than the algorithm based on force fields and the FCM algorithm; and the proposed method is on average about 56 times faster than the PFCM algorithm in real time and has better robustness.

1. Introduction

Multi-sensor image matching has been one of the key technologies in such fields as remote sensing and navigation guidance.[1] However, a lot of difficulties still exist[2] so it is hard to get the common features of multi-sensor images in such aspects as gray, intensity, and color, due to the large differences of imaging principles (such as optical images and synthetic aperture radar (SAR) images) and the different bands (such as optical images and infrared radiation (IR) images). We find that the salient edge of an object is a common feature between images, according to the analysis of typical multi-sensor images and their imaging principles. A salient edge can show the position and shape of a target and provide the important feature for object recognition.

Common edge detection methods include gradient methods,[3] force field conversion,[4] image segmentation, edge extraction methods, etc. However, no method can extract salient edges accurately and quickly for real IR images and SAR

* Corresponding author. Tel.: +86 25 84892284.
E-mail address: guilixu2002@163.com (G. Xu).

Peer review under responsibility of Editorial Committe of CJA.

1000-9361 © 2014 Production and hosting by Elsevier Ltd. on behalf of CSAA & BUAA. Open access under CC BY-NC-ND license.
http://dx.doi.org/10.1016/j.cja.2013.12.001

images which are mixed with noises and have poor contrast between background and target.

Image segmentation methods include such methods as threshold-based, region-based, edge-based, and clustering-based.[5] Among them, methods based on clustering algorithms, especially fuzzy clustering algorithms,[6] can overcome uncertainty and ambiguity for IR images and SAR images to a certain extent. Image segmentation algorithms based on clustering analysis consist of two parts: calculating a feature space and clustering with the feature space. In aspect of selecting the feature space, the gray scale is one of the most basic features of an image, but it cannot obtain satisfactory segmentation accuracy by only using the gray feature. According to imaging characteristics of multi-sensor images, the texture information[7] should be employed to make up for the deficiency in order to improve segmentation accuracy. Common texture descriptors include first-order texture features such as entropy, standard deviation, maximum difference, etc.,[8] and two-order texture features based on gray level co-occurrence matrix (GLCM) such as energy, contrast, autocorrelation, etc.[9]

In aspect of clustering algorithms, hard clustering methods have been extended gradually to fuzzy clustering methods, where the fuzzy C-means (FCM) method proposed by Dunn[10] and popularized by Bezdek[11] has been used most widely and successfully. Many improved algorithms based on FCM have been proposed: Pal and Bezdek put forward a fast algorithm called "eFFCM" for "large image" segmentation[12]; a fast clustering algorithm (geFFCM) fit for large data sets was proposed by Bezdek and Hathaway.[13] These methods cannot improve clustering accuracy further, though they are able to reduce clustering time, because FCM uses probabilistic constraints that the memberships of a data point across classes sum to 1, i.e., the influence of each data point on the clustering is the same, which makes a noise or an outlier be divided into an error cluster due to its large membership. A possibility C-means (PCM) clustering algorithm was proposed by Krishnapuram and Keller[14] by abandoning the constraint of FCM, which makes a noise or an outlier have little influence on the clustering due to its very small membership, but PCM tends to find identical clusters and is sensitive to initialization.

The possibility fuzzy C-means (PFCM) clustering algorithm[15] proposed by Pal et al., which is a combination of FCM and PCM, can overcome the shortcomings that FCM is sensitive to noises and PCM tends to find identical clusters. However, the segmentation accuracy of the PFCM algorithm still needs to be improved for real multi-sensor images with large noises. In addition, in order to reduce the computational complexity, Wu and Zhou[16] put forward an improved PFCM clustering algorithm that the second iteration was not required by optimizing the parameters of PFCM. Its real-time performance is still poor for images which contain tens of thousands of pixels although its computational complexity has been decreased.

For the problems that real-time performance and segmentation accuracy need to be improved for multi-sensor images with noises, we propose a novel salient edges detection algorithm and its rapid calculation based on PFCM kernel clustering by using two-dimensional vectors composed of gray and texture. Firstly, the two-dimensional feature space is constructed by each pixel's gray and texture features. Secondly, a method is proposed to improve real-time performance by compressing data sets based on the idea of data reduction in the field of mathematical analysis. Thirdly, the Euclidean distance in the kernel space instead of the sample space is employed to enhance the robustness by mapping the sample space of reduced data into the Gaussian kernel space. Lastly, image segmentation is carried out using PFCM clustering algorithm in kernel space and then salient edges are extracted by the Canny operator.

2. The KPFCM clustering algorithm of salient edges detection for multi-sensor images

The PFCM clustering algorithm can overcome the shortcomings to a certain extent that FCM is sensitive to noises and PCM tends to find identical clusters. However, in the case of large noises, the segmentation accuracy of the PFCM algorithm still needs to be improved. Therefore, in order to solve this problem, in the field of pattern recognition, Han et al.[17] proposed the idea of kernel clustering on the basis of the PFCM algorithm, which was KPFCM. Let data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ be a sample space, and $\boldsymbol{x}_k \in \boldsymbol{X}$, $k = 1, 2, \ldots, n$ be a sample vector in the sample space, that is, this sample space is composed of n samples. Map the data of the sample space into the Gaussian kernel space H by using nonlinear mapping ϕ. Get the data of the kernel space, that is $\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \ldots, \phi(\boldsymbol{x}_n)$. The dot product of two vectors in the sample space can be expressed by Gaussian kernel in the Gaussian kernel space. Gaussian kernel:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)) \tag{1}$$

In the space of kernel, the objective function of the PFCM algorithm can be formulated as:

$$J(\boldsymbol{T}, \boldsymbol{U}, \boldsymbol{V}) = \sum_{i=1}^{c}\sum_{j=1}^{n}(au_{ij}^m + bt_{ij}^p)D_{ij}^2 + \frac{\delta^2}{m^2 c}\sum_{i=1}^{c}\sum_{j=1}^{n}(1 - t_{ij})^p \tag{2}$$

where $\boldsymbol{T}$ is the possibility membership matrix; $\boldsymbol{U}$ is the fuzzy membership matrix; $\boldsymbol{V}$ is the cluster center matrix; t_{ij} is the possibility membership values of $\boldsymbol{x}_j$ in class i; p is the weighting exponent of t_{ij}; u_{ij} is the fuzzy membership values of $\boldsymbol{x}_j$ in class i; m is the weighting exponent of u_{ij}; c is the number of clusters; n is the number of data points; a, b define the relative importance of the fuzzy membership and possibility values in the objective function; D_{ij} is the distance from the jth data $\boldsymbol{x}_j$ to the ith cluster center $\boldsymbol{v}_i$; δ^2 is the sample variance; $D(x_j, \bar{x})$ is the distance from the jth data $\boldsymbol{x}_j$ to average value $\bar{\boldsymbol{x}}$.

$$D_{ij}^2 = D^2(\boldsymbol{x}_j, \boldsymbol{v}_i) = ||\phi(\boldsymbol{x}_j) - \phi(\boldsymbol{v}_i)||^2 = K(\boldsymbol{x}_j, \boldsymbol{x}_j) + K(\boldsymbol{v}_i, \boldsymbol{v}_i) - 2K(\boldsymbol{x}_j, \boldsymbol{v}_i) = 2(1 - K(\boldsymbol{x}_j, \boldsymbol{v}_i)) = 2(1 - \exp(-\beta||\boldsymbol{x}_j - \boldsymbol{v}_i||^2)) \tag{3}$$

$$\begin{cases} \delta^2 = \dfrac{\sum_{j=1}^{n} D^2(\boldsymbol{x}_j, \bar{\boldsymbol{x}})}{n} \\ \bar{\boldsymbol{x}} = \dfrac{\sum_{j=1}^{n} \boldsymbol{x}_j}{n} \end{cases} \tag{4}$$

For Eq. (2), $\min_{(t,u,v)} J(\boldsymbol{T}, \boldsymbol{U}, \boldsymbol{V})$ is optimized and the following updated equations are obtained:

$$t_{ij} = \left[1 + \left(\frac{bm^2 c}{\delta^2} D_{ij}^2\right)^{\frac{1}{p-1}}\right]^{-1}; \quad \forall i,j \tag{5}$$

$$u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2}\right)^{\frac{1}{m-1}}\right]^{-1}; \quad \forall i,j \tag{6}$$

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{n} (au_{ij}^m + bt_{ij}^p) \mathrm{e}^{-\beta \|\boldsymbol{x}_j - \boldsymbol{v}_i\|^2} \boldsymbol{x}_j}{\sum_{j=1}^{n} (au_{ij}^m + bt_{ij}^p) \mathrm{e}^{-\beta \|\boldsymbol{x}_j - \boldsymbol{v}_i\|^2}}; \quad \forall i \tag{7}$$

In the process of iterations, set $\varepsilon > 0$, r represents the iterative times, and $\boldsymbol{V}^{(r)}$ is the cluster center matrix after r times of iterations. If $\|\boldsymbol{V}^{(r)} - \boldsymbol{V}^{(r-1)}\| < \varepsilon$, then the iterations will end. Otherwise, iterations will continue until the maximum iterative time is reached.

The KPFCM algorithm maps the data of the sample space into the Gaussian kernel space by using nonlinear mapping, and then carries out PFCM clustering in the kernel space. The kernel space is infinite-dimensional, so a limited sample is linearly separable in the kernel space. In addition, the KPFCM algorithm uses an Euclidean distance in the kernel space instead of the sample space. In essence, it is a non-Euclidean distance instead of a Euclidean distance. Wu and Yang[18] proposed a non-Euclidean distance and verified its robustness on the basis of robust statistical viewpoint and influence function.

Therefore, we will introduce the idea of the kernel space in the field of pattern recognition into the algorithm of salient edge detection in order to improve the robustness. However, every iteration in this algorithm has to be calculated with the whole data set, which is unacceptably time-consuming for an image that contains tens of thousands of pixels. Therefore, we will research the rapid calculation of salient edges detection for multi-sensor images, on the basis of the KPFCM clustering algorithm.

3. The rapid algorithm of salient edges detection for multi-sensor images

3.1. Construct feature space

Imaging principle of an optical image is active imaging depending on the reflectivity of different materials. An optical image can reflect texture details of object surface well because of its high brightness and contrast. Imaging principle of an IR image is different from that of an optical image. It forms according to a gray value which is transformed by material IR received by an infrared detector. The overall grayscale distribution of an IR image is more concentrated due to small difference of temperature distribution of an actual object. An SAR image is microwave imaging by scattering from different objects. The main factors of affecting scattering are dielectric constant, surface roughness, etc., and different materials have different surface roughness, namely different textures when reflected in the image. Therefore, we give the definition of "salient edge": a significant edge which forms between different materials.

According to the above imaging principle of multi-sensor images and the definition of salient edge, gray scale is one of the most basic features to represent images. However, the issue that different objects have similar gray would appear if only using the gray feature, especially for IR images and SAR images which have poor contrast between background and target. Therefore, it will give more satisfactory segmentation results if combining gray scale and texture.

3.1.1. Analysis of gray feature

The average gray level is employed to describe the gray distribution of an image based on the characteristics of multi-sensor images. Calculate all pixels' average gray value in the neighborhood of a central pixel. By comprehensive consideration of time and quality, the size of neighborhood is set to be 9×9.

$$m' = \sum_{i=0}^{L-1} z_i p(z_i) \tag{8}$$

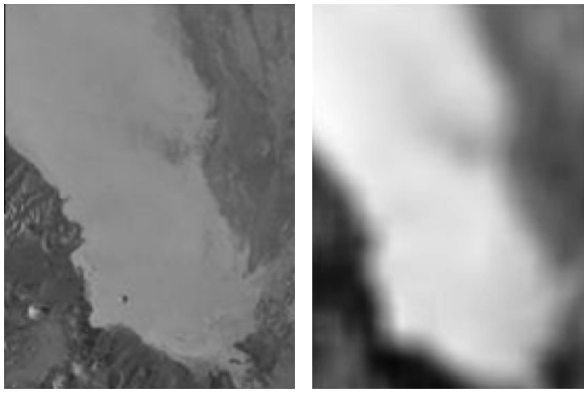
where L is the number of gray levels in the neighborhood; z_i represents the value of brightness, which belongs to the ith gray level, $z_i \in [0, 255]$; $p(z_i)$ is the gray histogram of the neighborhood, and $p(z_i) = n_i/N$, in which n_i is the number of pixels whose value of brightness is z_i and N represents the number of all pixels in the neighborhood; m' is all pixels' average gray value in the neighborhood. Taking an orginal SAR_1 image for example, we respectively calculate the average gray value for each pixel in its neighborhood by Eq. (8), which is the corresponding pixel's new gray value. All the new gray values constitute the grayscale average feature map, as shown in Fig. 1. It can reflect the characteristics of the original image.

3.1.2. Analysis of texture feature

Textures can reflect the spatial distribution of image grayscale and keep a better balance between the macroscopic and microscopic information in an image, compared with other image features. Common texture descriptors include first-order texture features and two-order texture features based on GLCM.

(1) First-order texture statistical features

(A) Entropy is a statistical measurement of image grayscale randomness and represents the texture information of an image. It is expressed as:



(a) Original image (SAR_1) (b) Grayscale average feature map

Fig. 1 An original image (SAR_1) and its grayscale average feature map.

$$E = -\sum_{i=0}^{L-1} p(z_i)\log_2 p(z_i) \tag{9}$$

If an image does not contain any texture, the value of entropy will be close to 0; if an image is filled with textures, the value of entropy will be larger.

(B) Standard deviation is the standard to measure the dispersion degree of the data distribution. It is expressed as:

$$\sigma = \sqrt{\sum_{i=0}^{L-1} (z_i - m)^2 p(z_i)} \tag{10}$$

The above two descriptors are used to calculate texture feature maps. Similarly, the values of E and σ are calculated in each pixel's neighborhood, respectively. All the obtained values form the entropy image and the standard deviation image, which are shown in Fig. 2 and reveal the characteristics of the original image.

(2) Two-order texture statistical features

GLCM is defined as the joint probability distribution of simultaneous occurrence of any two pixels in an image. $f(x, y)$ is a two-dimensional digital image. Its size is $M \times N$ and the number of gray levels is N_g. GLCM is expressed as

$$\begin{aligned} p(i,j) = g\{&(x_1, y_1), (x_2, y_2) \in M \times N | f(x_1, y_1) \\ &= i, f(x_2, y_2) = j\} \end{aligned} \tag{11}$$

which is normalized as follows:

$$P(i,j) = p(i,j)/S \tag{12}$$

where $g(\boldsymbol{X})$ represents the number of elements in the set $\boldsymbol{X}$, and S is the total number of pixel pairs. Three common texture features based on GLCM are as follows:

(A) Entropy is a measurement of content randomness and the amount of information.

$$W_1 = -\sum_i \sum_j P(i,j)\log_2[P(i,j)] \tag{13}$$

(B) Angle second moment reflects the uniformity and smoothness of an image.

$$W_2 = \sum_i \sum_j \{P(i,j)\}^2 \tag{14}$$

(C) Contrast refers to the clarity of texture.

$$W_3 = \sum_i \sum_j (i-j)^2 P(i,j) \tag{15}$$

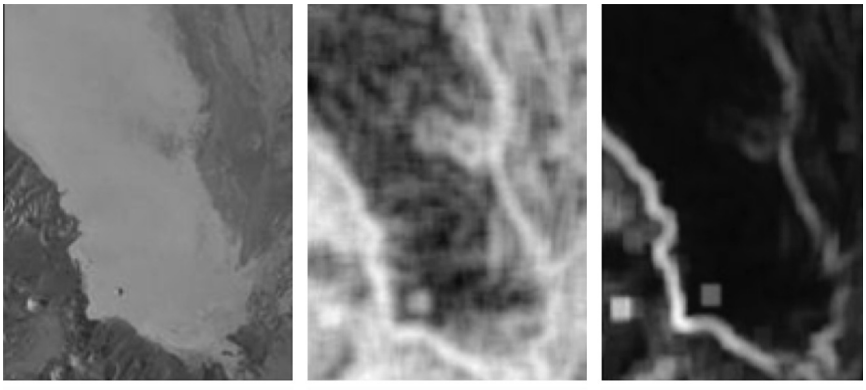
Texture feature maps based on GLCM are shown in Fig. 3. It can also reflect the characteristics of the original image.

(3) Analysis and research of two kinds of texture features

According to the descriptions of texture features in the above two subsections, we analyze from three aspects. Firstly, calculation principle aspect: first-order texture features are based on statistical properties of regional luminance histogram. Two-order texture features are on the basis of joint probability distribution of simultaneous occurrence of any two pixels in an image. It incurs large amount of calculation. Second, feature images aspect: we can see from Fig. 2 that target and background can be separated clearly, but some detail textures exist in target and background of two-order texture feature maps (see Fig. 3). The entire external contour of foreground, instead of details, is what we want to get, according to the definition of salient edge, so first-order texture feature images are relatively in line with the expected results. Thirdly, real-time performance aspect: taking entropy features for example, their edge detection times are shown in Table 1. Based on the above three points, first-order texture features will be used in this paper. In addition, we can see from Fig. 2(b) and (c) that the entropy image can retain the entire area of foreground but the standard deviation image can only retain edges and some details. Therefore, we will ultimately select first-order entropy as the texture feature.

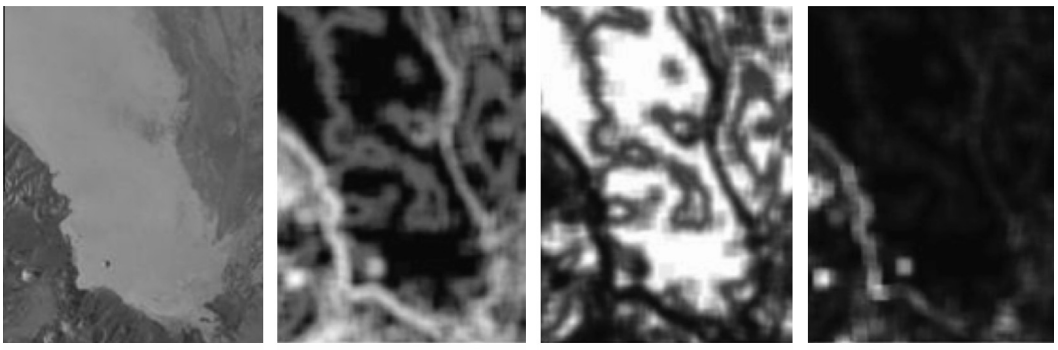
3.2. Research on the fast algorithm of salient edges detection for multi-sensor images based on KPFCM (FKPFCM)

We will carry out clustering with the two-dimensional feature space constructed above. On the basis of PFCM clustering in the kernel space, a method is proposed to improve real-time performance by compressing data sets based on the idea of data reduction[19] in the field of mathematical analysis. Here, in the process of image segmentation, we try to compress the two-dimensional feature space by reducing the number of data points for iterations from n to s (s is far less than n) and keep the clustering well, so as to decrease time of every iteration.The same feature vectors should belong to the same category in the data set $\boldsymbol{X}$. Therefore, the algorithm of data reduction can be described as: there are s kinds of feature vectors in the data set $\boldsymbol{X}$, that is, the new data set $\boldsymbol{X}' = \{\boldsymbol{x}'_1, \boldsymbol{x}'_2, \ldots, \boldsymbol{x}'_s\}$. The



(a) Original image (SAR_1) (b) Entropy image (c) Standard deviation image

Fig. 2 An original image (SAR_1) and its first-order texture feature maps.



(a) Original image (SAR_1) (b) Entropy image (c) Angle second moment image (d) Contrast image

Fig. 3 An original image (SAR_1) and its two-order texture feature maps.

Table 1 Detection time comparison of two kinds of texture features.

Image	Detection time(s)	
	First-order texture feature (entropy)	Two-order texture feature (entropy)
SAR_1	0.154414	25.353053

number of data points corresponding to every kind of feature vector is $H = \{h_1, h_2, \ldots, h_s\}$. Taking an SAR_1 image for example, it contains 237×306 pixels. Each pixel corresponds to a two-dimensional vector which is composed of the values of gray average and first-order entropy, so the data set $\boldsymbol{X}$ contains 72522 two-dimensional vectors. Now the values of gray average and first-order entropy are both normalized to 0–255 and every 10 is for a group, i.e., we substitute 5 for the numbers values of which are in 0–9, substitute 15 for the numbers values of which are in 10–19, and so on. Then the data set $\boldsymbol{X}'$ will at most contain 26×26 kinds of feature vectors, as shown in Fig. 4 (The first number of each two-dimensional vectors is the value of gray average and the second number is the value of first-order entropy in Fig. 4). Next, count the number of pixels corresponding to every kind of feature vector. Then 343 kinds of feature vectors are left except for those vectors to which the number of pixels corresponding is 0. Therefore, the number of data participating in iterations can be reduced from 72522 to 343 in every iteration process, which improves the speed of clustering greatly.

Formula is as follows:

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{s} h_j (a u_{ij}^m + b t_{ij}^p) \mathrm{e}^{-\beta \|\boldsymbol{x}'_j - \boldsymbol{v}_i\|^2} \boldsymbol{x}'_j}{\sum_{j=1}^{s} h_j (a u_{ij}^m + b t_{ij}^p) \mathrm{e}^{-\beta \|\boldsymbol{x}'_j - \boldsymbol{v}_i\|^2}}; \quad \forall i \tag{16}$$

The iterative formulas of t_{ij} and u_{ij} are the same as in Eqs. (5) and (6), respectively, but they should be computed using the new data set.

[5, 5]	[5, 15]	⋯	[5, 255]
[15, 5]	[15, 15]	⋯	[15, 255]
⋯	⋯	⋯	⋯
[255, 5]	[255, 15]	⋯	[255, 255]

Fig. 4 The largest new data set after data reduction.

Then the algorithm described below is called the FKPFCM algorithm:

(1) Initialization: fix c, m, p, β, a, b, and set $\varepsilon > 0$, the initial iterative times $r = 1$, and the maximum iterative times $r_{\max}$.
(2) Compute δ^2 using Eq. (4).
(3) Employ the FCM algorithm and get the cluster centers as the initial cluster centers $\boldsymbol{V}^{(0)}$.
(4) Get the new data set using the algorithm of data reduction.
(5) Repeat the calculation below with the new data set: update $\boldsymbol{T}^{(r)}$, $\boldsymbol{U}^{(r)}$, and $\boldsymbol{V}^{(r)}$ using Eqs. (5), (6) and (16), respectively. Until $\|\boldsymbol{V}^{(r)} - \boldsymbol{V}^{(r-1)}\| < \varepsilon$ or $r >= r_{\max}$.

The Canny operator is used to detect the edge, on the basis of image segmentation.

4. Experimental results and analysis

In order to verify the validity of the algorithm, we designed experiments from two aspects: the robustness and the real-time performance.

4.1. The validation of robustness

We carried out image segmentation and edge detection using the algorithm based on force field transformation (see Ref. [4]), the FCM algorithm, the PFCM algorithm, and the FKPFCM algorithm proposed in this paper for optical images, IR images, SAR images, and their corresponding actual images with noises, respectively. The experimental results are shown in Figs. 5–7. The first row of each figure is the experimental results for noise-free images and the second row of each figure is the experimental results for the simulated actual images with noises.

We can see from the figures that, the algorithm of Ref. [4] can extract salient edges of optical images and IR images, but there exist some breakpoints in the detection results (see Figs. 5(b) and 6(b)). In addition, the experimental results are poor for SAR images which have ample textures (see Fig. 7(b)). Because the composition and separation of mechanics in the model just uses grey information, the FCM algorithm has poor segmentation accuracy when the species of background or target is not single (see Fig. 5(c)) and is sensitive to noises (see Fig. 7(c)). Because FCM uses probabilistic constraints that the memberships of a data point across classes sum to 1, which makes a noise or an outlier be divided into an error cluster due to its large membership, the PFCM algorithm can retain relatively complete target area and extract the salient edge, but some error clustering may exist for images with large noises, because the Euclidean distance in the sample space cannot distinguish data points well, so the obtained target areas are deformed or inaccurate. For example, at the places of labels 1, 2, 3, 4 in Fig. 5(d), the segmentation of the horse's leg is incomplete. At the places of labels 1, 2 in Fig. 6(d), the rockery in the middle of the image and the water in the pond are not separated well. The deformation exists at the places of labels 1, 2 in Fig. 7(d). The proposed method can realize image segmentation and edge detection effectively for optical images, IR images, SAR images, and their corresponding actual images with noises, because it employs the Euclidean distance in the kernel space instead of the sample space on the basis of the PFCM algorithm. As shown at the places of labels in Figs. 5–7, this algorithm has better segmentation accuracy compared with PFCM; it is basically not affected by noises and has better robustness, by comparing the experimental results of noise-free images with the results of their noisy images.

The experimental results of three other images are presented in Figs. 8–10, respectively, to further verify the effectiveness of the proposed method and the conclusions above.

At the places of labels 1, 2, 3, 4 in Fig. 8(d), there exists some error clustering on the path; at the places of labels 1, 2, 3, 4 in Fig. 9(d), the more transitional part between path and lawn is detected; at the places of labels 1, 2, 3 in Fig. 10(d), there exists more inaccurate clustering. According to the comparison of the experimental results above, apparently the proposed method can realize image segmentation and edge detection effectively for optical images, IR images, and SAR images, and basically it is not affected by noises.

In order to show the effectiveness of the proposed method more intuitively, we adopted the evaluation system of edge detection proposed by Grigorescu et al.[20] to make quantitative evaluation for the above experimental results. The precision of edge detection is defined as below:

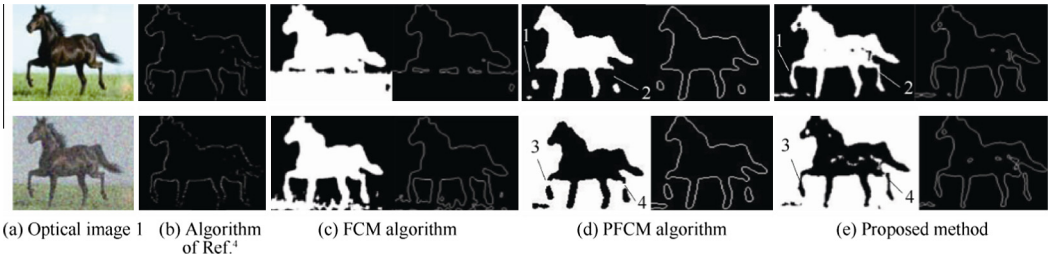


Fig. 5 Experimental results of different algorithms for optical image 1.

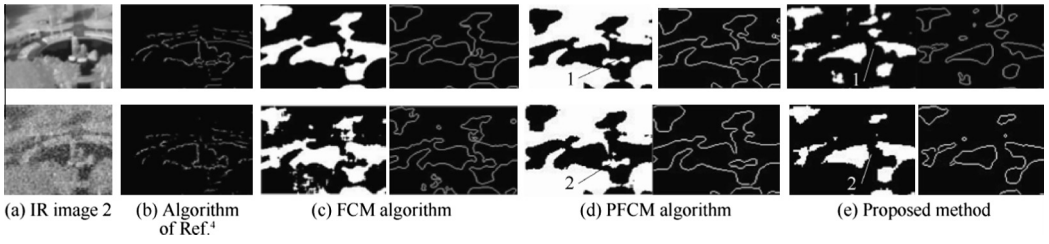


Fig. 6 Experimental results of different algorithms for IR image 2.

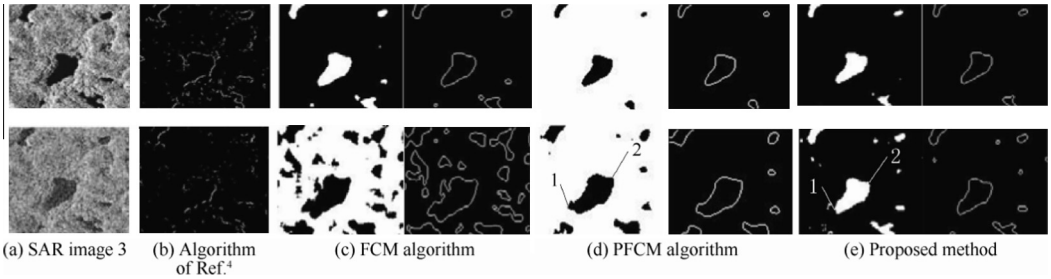


Fig. 7 Experimental results of different algorithms for SAR image 3.

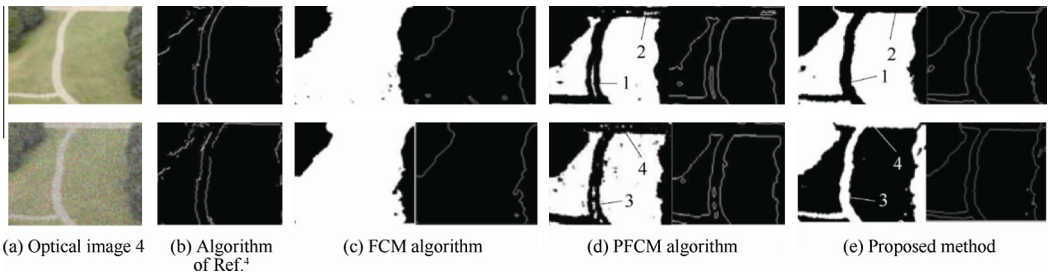


Fig. 8 Experimental results of different algorithms for optical image 4.

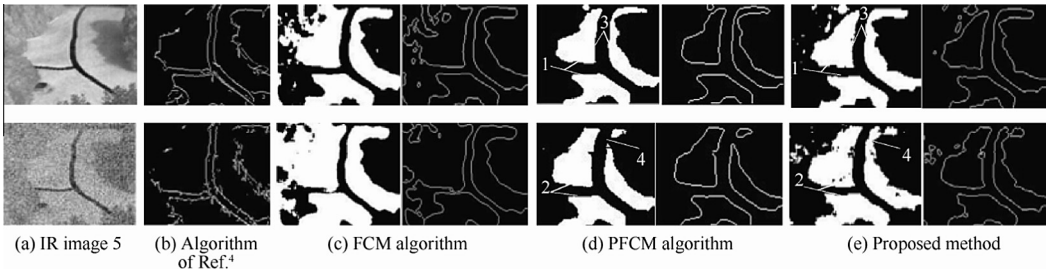


Fig. 9 Experimental results of different algorithms for IR image 5.

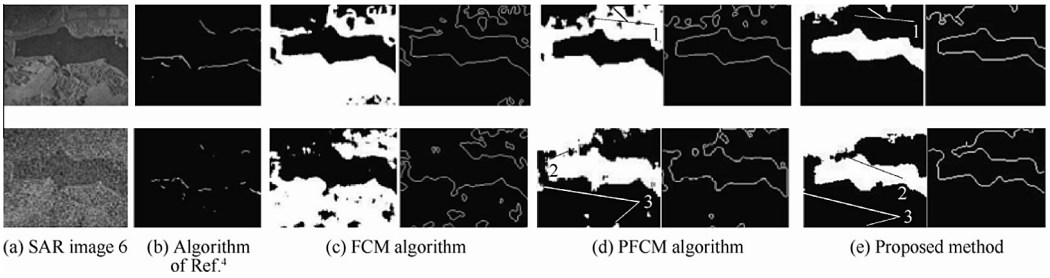


Fig. 10 Experimental results of different algorithms for SAR image 6.

Table 2 The edge detection precisions of different algorithms for images and their corresponding noisy images.

No. of multi-sensor images	The precision for noise-free image/the precision for its corresponding image with noises			
	Algorithm of Ref. [4]	FCM algorithm	PFCM algorithm	Proposed method
Optical image 1	0.7659/0.6825	0.4800/0.6859	0.8394/0.8128	0.8628/0.8469
IR image 2	0.4026/0.3912	0.6053/0.5655	0.5435/0.5612	0.6667/0.6188
SAR image 3	0.2084/0.2023	0.8166/0.3192	0.8252/0.6792	0.8821/0.7897
Optical image 4	0.6099/0.5929	0.3853/0.3999	0.8152/0.8089	0.8580/0.8453
IR image 5	0.6484/0.6019	0.7758/0.5865	0.8478/0.8425	0.8482/0.8135
SAR image 6	0.3843/0.2106	0.5141/0.4465	0.7030/0.5597	0.8672/0.5719

$$P = \mathrm{card}(A)/(\mathrm{card}(A) + \mathrm{card}(A_l) + \mathrm{card}(A_f)) \quad (17)$$

where A represents the collection of pixels detected correctly, A_l is the collection of pixels which are the correct target pixels but undetected, A_f represents the collection of pixels detected falsely, and card(X) indicates the number of pixels in the set $\boldsymbol{X}$. The precision P of different algorithms are shown in Table 2.

Table 2 quantitatively shows that the proposed method has better robustness than other algorithms for optical images, IR images, SAR images, and their corresponding actual images with noises.

4.2. The validation of real-time performance

The edge detection times of different algorithms are shown in Table 3.

Table 3 shows that the algorithm of Ref. [4], FCM, and the proposed algorithm all have good real-time performance, but the edge detection results of the proposed algorithm are better than those of others. In addition, the proposed method is on average about 56 times faster than the PFCM algorithm in real time and even more if the original image gets larger. Therefore, the proposed method can extract salient edges of

Table 3 The comparison of the edge detection times of different algorithms.

No. of multi-sensor images	Detection time(s)			
	Algorithm of Ref. [4]	FCM algorithm	PFCM algorithm	Proposed method
Optical image 1	3.37	0.72	86.12	1.79
IR image 2	3.17	0.55	89.32	1.66
SAR image 3	1.66	0.69	92.66	1.17
Optical image 4	4.17	0.50	66.79	1.44
IR image 5	1.09	0.61	98.41	1.49
SAR image 6	4.18	0.40	74.33	1.65

real multi-sensor images mixed with noises fast and accurately according to the validation of the robustness and real-time.

5. Conclusions

(1) According to the research of the salient edge formation, PFCM clustering, and the idea of data reduction in the field of mathematical analysis, a novel salient edge detection algorithm for multi-sensor images and its rapid calculation is proposed.
(2) The values of gray average and first-order entropy are selected to construct a two-dimensional feature space, based on the analysis of imaging principle of multi-sensor images and the contrast between first-order and two-order texture features.
(3) On the basis of PFCM, the idea of kernel clustering is used to enhance robustness further; in addition, a novel method is proposed to improve real-time performance based on the idea of data reduction.
(4) Experimental results show that the proposed algorithm of salient edge detection is more accurate than the algorithm based on force field and the FCM algorithm; it is on average about 56 times faster than the PFCM algorithm in real time and has better robustness.

Acknowledgements

This project was supported by the Aeronautical Science Foundation of China (Grant No. 20100152003), the National Natural Foundation of China (Grant Nos. 60974105, 61074161), and the Fundamental Research Funds for the Central Universities (No. NZ2012307).

References

1. Li Z. Research on key technologies of multi-sensor images matching [dissertation]. Changsha: College of Aerospace and Material Engineering, National University of Defense Technology; 2011 [Chinese].
2. Papari G, Campisi P, Petkov N, Neri A. A biologically motivated multiresolution approach to contour detection. *EURASIP J Appl Signal Processing* 2007;**2007**(1):119.
3. Sen D, Pal SK. Gradient histogram: thresholding in a region of interest for edge detection. *Image Vis Comput* 2010;**28**(4):677–95.
4. Cao CD, Xu GL, Chen X, Leng XF, Li KY, Ye YQ. Image edge detection algorithm based on force field. *Acta Aeronautica et Astronautica Sinica* 2011;**32**(5):891–9 [Chinese].
5. Xu XZ, Ding SF, Shi ZH, Jia WK. New theories and methods of image segmentation. *Acta Electronica Sinica* 2010;**38**(2A):76–82 [Chinese].
6. Feng J, Jiao LC, Zhang XR, Gong MG, Sun T. Robust non-local fuzzy c-means algorithm with edge preservation for SAR image segmentation. *Signal Processing* 2013;**93**(2):487–99.
7. Liu GH, Yang JY. Image retrieval based on the texton co-occurrence matrix. *Pattern Recogn* 2008;**41**(12):3521–7.
8. Zhang FL. Research on contour-based image matching technology [dissertation]. Nanjing: College of Automation Engineering, Nanjing University of Aeronautics and Astronautics; 2010 [Chinese].
9. Haralick RM, Shangmugam K, Dinstein I. Textural feature for image classification. *IEEE Transactions on Systems, Man and Cybernetics* 1973;**SMC-3**(6):610–21.
10. Dunn JC. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J Cyber Syst* 1973;**3**(3):32–57.
11. Bezdek JC. *Pattern recognition with fuzzy objective function algorithms*. New York: Kluwer Academic Publishers; 1981.
12. Pal NR, Bezdek JC. Complexity reduction for "large image" processing. *IEEE Trans Syst Man and Cybern Part B* 2002;**32**(5):598–611.
13. Bezdek JC, Hathaway RJ. Progressive sampling schemes for approximate clustering in very large datasets. In: *Proceedings of 2004 IEEE international conference on fuzzy systems*; 2004 Jul 25–29; Budapest, Hungary.
14. Krishnapuram R, Keller JM. A possibilistic approach to clustering. *IEEE Trans Fuzzy Syst* 1993;**1**(2):98–110.
15. Pal NR, Pal K, Bezdek JC. A possibilistic fuzzy c-means clustering algorithm. *IEEE Trans Fuzzy Syst* 2005;**13**(4):517–30.
16. Wu XH, Zhou JJ. A novel possibilistic fuzzy C-means clustering. *Acta Electronica Sinica* 2008;**36**(10):1996–2000 [Chinese].
17. Han XD, Xia ZX, Liu B, Zhou Y. Kernel-based fast improved possibilistic C-means clustering method. *Comput Eng Appl* 2011;**47**(6):176–80 in Chinese.
18. Wu KL, Yang MS. Alternative C-means clustering algorithm. *Pattern Recogn* 2002;**35**(10):2267–78.
19. Eschrich S, Ke JW, Hall LO, Goldgof DB. Fast accurate fuzzy clustering through data reduction. *IEEE Trans Fuzzy Syst* 2003;**11**(2):262–70.
20. Grigorescu C, Petkov N, Westenberg MA. Contour detection based on nonclassical receptive field inhibition. *IEEE Trans Image Processing* 2003;**12**(7):729–39.

Xu Guili is a professor in the College of Automation Engineering at Nanjing University of Aeronautics and Astronautics, Nanjing, China. He received his Ph.D. degree from Jiangsu University in 2002. His current research interests are photoelectric measure, computer vision, and intelligent system.