Optik 124 (2013) 3103–3111

Contents lists available at SciVerse ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Artificial Bee Colony approach to information granulation-based fuzzy radial basis function neural networks for image fusion

Jiaqian Yu[a], Haibin Duan[a,b,*]

[a] Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, PR China
[b] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, PR China

ARTICLE INFO

Article history:
Received 6 May 2012
Accepted 14 September 2012

Keywords:
Radial basis function neural network (RBFNN)
Fuzzy C-Means (FCM)
Ordinary least square (OLS)
Artificial Bee Colony (ABC)
Image fusion

ABSTRACT

This paper mainly proposed a novel method of Artificial Bee Colony (ABC) optimized fuzzy radial basis function neural networks with information granulation (IG-FRBFNNs) for solving the image fusion problem. Image fusion is the process of combining relevant information from two or more images into a single image. The fuzzy RBF neural networks exploit the Fuzzy C-Means (FCM) clustering to form the premise part of the rules. As the consequent part of the model (being the local model representing input output relation in the corresponding sub-space), four types of polynomials are considered, with the ordinary least square (OLS) learning being exploited to estimate the values of the coefficients of the polynomial. Since the performance of the IG-FRBFNN model is directly affected by the parameters such as the fuzzification coefficient used in the FCM, the position of their centers and the values of the widths, ABC algorithm is exploited to carry out the structural and parametric optimization of the model respectively while the optimization is of multi-objective character as it is aimed at the simultaneous minimization of complexity and maximization of accuracy. Subsequently, the proposed approach can dynamically obtain optimal image fusion weights based on regional features, so as to optimize performance of image fusion. Series of experimental results are presented to verify the feasibility and effectiveness of the proposed approach.

1. Introduction

Image fusion is to integrate complementary information from multi-sensor data so that the new images are more suitable for the purpose of human visual perception and computer-processing tasks [1]. Its major application fields include night vision, target detection, navigation aid for pilots, medical imaging, remote sensing, weather forecasting and computer vision. Image fusion can be performed at signal, pixel, and feature and symbol level [2].

Image fusion techniques can be applied to numerous fields such as surveillance, remote sensing, automated machine vision and medical imaging. Security systems can be improved by detecting concealed weapon underneath person's clothing where manual screening procedures are difficult to implement. Several imaging sensors such as infrared imager or passive millimeter wave (MMW) sensors can detect concealed weapon by capturing the temperature distribution of the target to form an image. However, these sensors are unable to provide the information needed to identify the person carrying a weapon. A visual image of the same scene can supplied such information. Therefore, by fusing a visual image with the corresponding IR or MMW image result in positively identification of a person carrying a concealed weapons [3–5]. Many methods and software for image fusion have been developed. According to the stage at which the combination mechanism takes place, the existing image fusion methods can be generally grouped into three categories, namely, pixel level, feature level, and decision level [6].

The most direct image fusion method only calculates the average pixel-by-pixel of the source images, which usually leads to undesirable side effects such as reduced contrast [7]. Another basic idea is to perform a multi-resolution decomposition on each source image, then integrate all these decompositions to produce a composite representation. The fused image is finally reconstructed by performing an inverse multi-resolution transform. Examples of this approach include the Laplacian pyramid, the gradient pyramid, the ratio-of-low-pass pyramid and the morphological pyramid. More recently, the discrete wavelet transform (DWT) has also been used [8].

In this study, we proposed a method of bio-inspired scheme to solve the image fusion problem. Bio-inspired intelligence has the advantages of strong robustness, good distributed computing mechanism, and easy to combine with other methods [9]. Bio-inspired intelligent methods are nowadays popular used in various fields, such as control problems for hypersonic vehicle [10], image matching problems [11,12], and target recognition problems for

* Corresponding author at: Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, PR China. Tel.: +86 10 82317318.
E-mail address: hbduan@buaa.edu.cn (H. Duan).

0030-4026/$ – see front matter © 2012 Elsevier GmbH. All rights reserved.
http://dx.doi.org/10.1016/j.ijleo.2012.09.033

aircraft [13]. Especially, Artificial Bee Colony (ABC) algorithm has been successfully applied in these fields, which is a newly bio-inspired optimization algorithm based on the foraging behavior of a honey bee swarm, using only common control parameters such as colony size and maximum cycle number. ABC can provide a population-based search procedure in which individuals called food positions are modified by the artificial bees with time and the bee's aim is to discover the places of food sources with high nectar amount and finally the one with the highest nectar [14].

Artificial neural networks (ANNs) have been advocated as excellent classifiers in medical image analysis systems, because of their ability to define nonlinear decision surface and the special training procedures that alleviate the need for explicitly defining the parameters space [15–19]. The radial basis function (RBF) neural network architecture [20] is a special type of ANNs that has certain advantages over other network types, including simpler network configurations, faster training procedures, and better approximation capabilities [21].

In this paper, an approach based on fuzzy radial basis neural networks (FRBFNN) is introduced. FRBFNN is designed by integrating the principles of a radial basis function neural network (RBFNN) and exploiting the mechanisms of information granulation provided by the Fuzzy C-Means (FCM) [22,23]. In conventional RBFNNs, the activation function of a hidden node is typically realized in the form of some Gaussian function. The location of any RBF in the input space is uniquely specified by its center and width (spread). The number of the receptive fields has to be decided in advance. The output of the RBFNN is a weighted sum of the activation levels of the individual RBFs. The learning algorithm of the RBFNN is used to adjust the weights of the links from the hidden layer to the output layer. The salient features of RBF neural networks are as follows.

(1) They are universal approximates [24].
(2) They possess the best approximation property [25].
(3) Their learning speed is fast because of locally tuned neurons [20].
(4) They have more compact topology than other neural networks [26].

Fuzzy clustering may be used to determine the number of RBFs, as well as a position of their centers and the values of the widths.

The remainder of this paper is structured as follows. Firstly, the problem of image fusion is analyzed to motivate our work in Section 2. Then, the RBF networks used are described and the architecture proposed of IG-FRBFNN is introduced in Section 3, with a strong focus on the innovative aspects mentioned above. After that, the architecture optimization, the ABC algorithm is introduced in Section 4. Finally, series of subjective and objective experimental results are given in Section 5, which shows the feasibility and effectiveness of our proposed approaches. Our concluding remarks are contained in Section 6.

2. Image fusion method based on regional features

Image fusion can be performed at two levels, pixel level and feature level. The pixel-level fusion determines the value of each pixel based on a set of pixels from the source images [27]. The feature level fusion extracts salient features from each source image and performs the integration based on the extracted features. When applying any of these image fusion methods, it is necessary that the source images be accurately superimposed. An Excellent image fusion method should contain a good retention of the edge and texture information, as well as the maximization the information contained in the fused image.

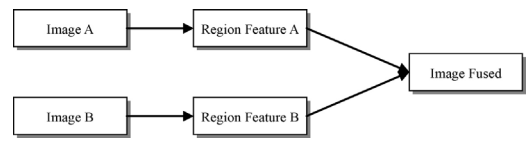

Fig. 1. Output by IG-FRBFNN.

The basic idea of region-based features for image fusion is the contrast of the regional features of the source images (including regional variance, regional energy, etc., we select characteristics of the regional variance) to dynamically select this feature prominent source image component integration the results [28]. This choice is based on regional characteristics conducted by pixel, the window size is generally chosen as 3×3 or 5×5. Let the image area to be treated x, the size of $M \times N$, the window of the center pixel of $x(i, j)$, said the regional features of the area variance:

$$Std(x) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (x(i, j) - \overline{x})^2 \tag{1}$$

where $\overline{x}$ is the gray area average (Fig. 1).

3. Information granulation based fuzzy radial basis function neural networks

In this section, an approach based on FRBFNN is introduced, which is designed by integrating the principles of a radial basis function neural network (RBFNN) and exploiting the mechanisms of information granulation provided by FCM.

The conventional RBF neural network can be considered as a mapping: $R^r \rightarrow R^s$ (see Fig. 2). A single-output RBF network consists of three different types of neurons [29]: input neurons which are used just for feeding the input data to the hidden neurons, nonlinear neurons having a Gaussian TF at the hidden layer, and a linear neuron performing a weighted sum at the output layer. The activation function of a hidden node is typically realized in the form of some Gaussian function. The location of any RBF in the input space is uniquely specified by its center and width (spread). The node is characterized by its center, which is a vector with dimension equal to the number of inputs to the node. The number of the receptive fields has to be decided in advance. The output of the RBFNN is a weighted sum of the activation levels of the individual RBFs. The learning algorithm of the RBFNN is used to adjust the weights of the links from the hidden layer to the output layer.

The key idea of an RBF neural network is to partition the input space into a number of subspaces. Accordingly, clustering algorithms, which are widely used in RBF neural network, are a logical approach to solve the problems. Fuzzy clustering may be used to determine the number of RBFs, as well as a position of their centers and the values of the widths. Fig. 3 shows the architecture of IG-RBFNN, concluding globally the following steps:

Step 1: Input the information into the network.

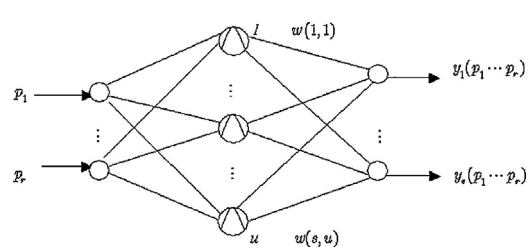

Fig. 2. Conventional RBF network.

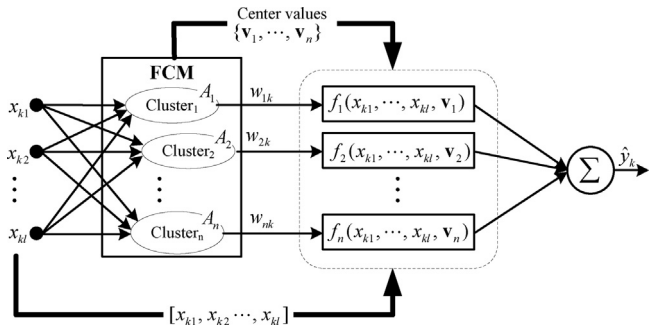

Fig. 3. The architecture of IG-RBFNN.

Step 2: The fuzzy layer, calculate the input variables belong to each fuzzy membership functions, with the use of Fuzzy C-Means.
Step 3: Calculate the fitness value belong to the fuzzy rules, and calculate the output of each fuzzy rule, based on the method of information granulation.
Step 4: Output the results of network.

3.1. Premise learning of the IG-FRBFNN with the use of Fuzzy C-Means

Consider the set $\mathbf{X}$ composed by m vectors located in a certain l-dimensional Euclidean space, that is, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$, $\mathbf{x}_k = \{\mathbf{x}_{k1}, \mathbf{x}_{k2}, \ldots, \mathbf{x}_{kl}\} \in \Re^l$, $1 \le i \le m$, $1 \le j \le l$, where m is the number of input data, l is the number of input variables. Clustering results in the assignment of the input vectors $\mathbf{x}_k \in \mathbf{X}$ into n clusters where the clusters are represented by the prototypes (center values) $\mathbf{v}_i = \{v_{i1}, v_{i2}, \ldots, v_{il}\} \in \Re^l$, $1 \le i \le n$. Let $\Re^{nl}$ denote the set of real $n \times 1$ matrices with the entries in [0,1]. Then $\mathbf{U} = [u_{ik}] \in \Re^{nl}$. The level of assignment of $\mathbf{x}_k \in \mathbf{X}$ to the ith cluster is expressed by the membership function $u_{ik} = u_i(\mathbf{x}_k)$. Fuzzy partitions $\mathbf{U}$ of $\mathbf{X}$ satisfy the condition [22]:

$$\sum_{i=1}^{n} u_{ik} = 1, \quad 1 \le k \le m \tag{2}$$

and

$$0 < \sum_{k=1}^{m} u_{ik} < m, \quad 1 \le i \le n \tag{3}$$

The FCM algorithm develops the structure in data by minimizing the following objective function [14,22]:

$$J_m = \sum_{i=1}^{n} \sum_{k=1}^{m} (u_{ik})^r d(\mathbf{x}_k, \mathbf{v}_i), \quad 1 < r < \infty \tag{4}$$

where r is the fuzzification coefficient, $d(\mathbf{x}_k, \mathbf{v}_i)$ is a distance between the input vector $\mathbf{x}_k \in \mathbf{X}$ and the prototype (centroid) $\mathbf{v}_i \in \Re^l$. Quite commonly it comes in the form of the weighed Euclidean distance computed between $\mathbf{x}_k$ and $\mathbf{v}_i$ defined as follows:

$$d(\mathbf{x}_k, \mathbf{v}_i) = ||\mathbf{x}_k - \mathbf{v}_i||^2 = \sum_{j=1}^{l} \frac{(\mathbf{x}_{kj} - \mathbf{v}_{ij})^2}{\sigma_j^2} \tag{5}$$

where σ_j^2 is the variance of the lth input variable. This type of distance equipped with the weights allows us to deal with variables of substantial variability.

Under the assumption of the form of the weighted Euclidean distance, the necessary conditions for solutions $(\mathbf{U}, \mathbf{V})$ of $\min\{J_m(\mathbf{U}, \mathbf{V})\}$ are specified as [22]:

$$u_{ik} = w_{ik} = \frac{1}{\sum_{i=1}^{n} \left(||\mathbf{x}_k - \mathbf{v}_i|| / ||\mathbf{x}_k - \mathbf{v}_j||\right)^{(2/(r-1))}}, \quad 1 \le k \le m, \quad 1 \le i \le n \tag{6}$$

and

$$v_i = \frac{\sum_{k=1}^{m} u_{ik}^r \mathbf{x}_k}{\sum_{k=1}^{m} u_{ik}^r}, \quad 1 \le i \le n \tag{7}$$

3.2. Learning of the consequent part of the IG-FRBFNN

As the consequent part of fuzzy rules of the IG-based FRBFNN model (being the local model representing input output relation in the corresponding sub-space), four types of polynomials are considered, namely constant, linear, quadratic, and modified quadratic [22]. This provides a significant level of design flexibility as each rule could come with a different type of the local model. Admit the consequent polynomials to be in of one of the following extended forms:

Type 1: Zero-order polynomial (constant type)

$$f_i(x_{k1}, \ldots, x_{kl}, \mathbf{v}_i) = a_{i0} \tag{8}$$

Type 2: First-order polynomial (linear type)

$$f_i(x_{k1}, \ldots, x_{kl}, \mathbf{v}_i) = a_{i0} + a_{i1}(x_{k1} - v_{i1}) + a_{i2}(x_{k2} - v_{i2}) + \cdots + a_{il}(x_{kl} - v_{il}) \tag{9}$$

Type 3: Second-order polynomial (quadratic type)

$$\begin{aligned} f_i(x_{k1}, \ldots, x_{kl}, \mathbf{v}_i) = {} & a_{i0} + a_{i1}(x_{k1} - v_{i1}) + a_2(x_{k2} - v_{i2}) + \cdots \\ & + a_{il}(x_{kl} - v_{il}) + a_{i(l+1)}(x_{k1} - v_{i1})^2 \\ & + a_{i(l+2)}(x_{k2} - v_{i2})^2 + \cdots + a_{i(2l)}(x_{kl} - v_{il}) \\ & + a_{i(2l+1)}(x_{k1} - v_{i1})(x_{k2} - v_{i2}) + \cdots \\ & + a_{i((l+1)(l+2)/2)}(x_{k(l-1)} - v_{i(l-1)})(x_{kl} - v_{il})^2 \end{aligned} \tag{10}$$

Type 4: Modified second-order polynomial (modified quadratic type)

$$\begin{aligned} f_i(x_{k1}, \ldots, x_{kl}, \mathbf{v}_i) = {} & a_{i0} + a_{i1}(x_{k1} - v_{i1}) + a_{i2}(x_{k2} - v_{i2}) + \cdots \\ & + a_{il}(x_{kl} - v_{il}) + a_{i(l+1)}(x_{k1} - v_{i1})(x_{k2} - v_{i2}) \\ & + \cdots + a_{i(l(l+1)/2)}(x_{k(l-1)} - v_{i(l-1)})(x_{kl} - v_{il}) \end{aligned} \tag{11}$$

The determination of the numeric output of the model, based on the activation levels of the rules, is given in the form:

$$\hat{y}_k = \sum_{i=1}^{n} w_{ik} f_i(x_{k1}, \ldots, x_{kl}, \mathbf{v}_i) \tag{12}$$

3.3. Ordinary least square (OLS)-based learning algorithm

Ordinary least square (OLS) is a well known global learning algorithm that minimizes an overall squared error JG between output of the model and the experimental data:

$$J_G = \sum_{k=1}^{m} \left(y_k - \sum_{i=1}^{n} w_{ik} f_i(\mathbf{x}_k - \mathbf{v}_i) \right)^2 \tag{13}$$

where w_{ik} is the normalized firing (activation) level of the ith rule as expressed.

The performance index JG can be expressed in a concise form as follows:

$$J_G = (\mathbf{Y} - \mathbf{Xa})^T(\mathbf{Y} - \mathbf{Xa}) \tag{14}$$

where $\mathbf{a}$ is the vector of coefficients of the polynomial, $\mathbf{Y}$ is the output vector of real data, $\mathbf{X}$ is matrix which rearranges input data, information granules (centers of each cluster) and activation level. In case all consequent polynomials are linear (first-order polynomials), $\mathbf{X}$ and $\mathbf{a}$ can be expressed as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & w_{11}(x_{11}-v_{11}) & \cdots & w_{11}(x_{l1}-v_{1l}) & \cdots & \cdots & 1 & w_{n1}(x_{11}-v_{n1}) & \cdots & w_{n1}(x_{l1}-v_{nl}) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_{1k}(x_{1k}-v_{11}) & \cdots & w_{1k}(x_{lk}-v_{1l}) & \cdots & \cdots & 1 & w_{nk}(x_{1k}-v_{n1}) & \cdots & w_{nk}(x_{lk}-v_{nl}) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_{1m}(x_{1m}-v_{11}) & \cdots & w_{1m}(x_{lm}-v_{1l}) & \cdots & \cdots & 1 & w_{nm}(x_{1m}-v_{n1}) & \cdots & w_{nm}(x_{lm}-v_{nl}) \end{bmatrix}$$

$$\mapsto \quad \text{the first fuzzy rule} \quad \hookleftarrow \qquad \mapsto \quad \text{the n'th fuzzy rule} \quad \hookleftarrow \tag{15}$$

$$\mathbf{a} = [\, a_{10} \quad a_{11} \quad \cdots \quad a_{1l} \cdots \quad \cdots a_{n0} \quad a_{n1} \quad \cdots \quad a_{nl} \,]^T \tag{16}$$

The optimal values of the coefficients of the consequent are determined in the form

$$\mathbf{a} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{17}$$

4. ABC algorithm

Since the performance of the IG-FRBFNN model is directly affected by the parameters such as the fuzzification coefficient used in the FCM, the position of their centers and the values of the widths, the ABC algorithm is exploited to carry out the structural and parametric optimization of the model respectively while the optimization is of multi-objective character as it is aimed at the simultaneous minimization of complexity and maximization of accuracy.

ABC algorithm was firstly proposed by a famous Nobel Prize winner Karlvon Frisch, simulating the self-organization simulation model of honey bees [30]. In this model, although each bee only performs one single task, yet through a variety of information communication ways between bees such as waggle dance and special odor, the entire colony can always easily find food resources that produce relative high amount nectar, hence realize its self-organizing behavior.

In order to introduce the self-organization model of forage selection that leads to the emergence of collective intelligence of honey bee swarms, first, we need to define three essential components: food sources, unemployed foragers and employed foragers [13].

(1) Food sources (A and B in Fig. 4): For the sake of simplicity, the "profitability" of a food source can be represented with a single quantity, which corresponds to the similarity value in our image fusion problem.
(2) Unemployed foragers (UF in Fig. 4): Unemployed foragers are continually looking out for a food source to exploit. There are two types of unemployed foragers: scouts (S in Fig. 2) and onlookers (R in Fig. 2).
(3) Employed foragers (EF1 and EF2 in Fig. 4): They carry with them information about this particular source, the profitability of the source and share this information with a certain probability.

After the employed foraging bee loads a portion of nectar from the food source, it returns to the hive, unloads the nectar to the food area in the hive, and converts into any kind of bees (UF or EF) in accordance with the profit of the searched food sources.

Define N_s as the total number of bees, N_e as the colony size of the employed bees and N_u as the size of unemployed bees, which satisfy that $N_s = N_e + N_u$.

Fig. 5 shows the architecture of ABC, concluding globally the following steps:

Step 1: Randomly initialize a set of feasible solutions $(X_1, \ldots, X_n)$.

$$X_i^j = X_{\min}^j + rand(0,1)(X_{\max}^j - X_{\min}^j) \tag{18}$$

Step 2: For an employed bee in the nth iteration X_n, search new solutions in the neighborhood of the current position.

$$V_i^j = X_i^j + \varphi_i^j(X_i^j + X_k^j) \tag{19}$$

Step 3: Apply the greedy selection operator to choose the better solution between searched new vector V_i and the original vector X_i.

$$P\{T_s(X_i, V_i) = V_i\} = \begin{cases} 1, f(V_i) \geq f(X_i) \\ 0, f(V_i) < f(X_i) \end{cases} \tag{20}$$

Step 4: Each unemployed bee selects an employed bee from the colony according to their fitness values and the probability distribution.

$$P\{T_{sl}(X) = X_i\} = \frac{f(X_i)}{\sum_{m=1}^{Ne} f(X_m)} \tag{21}$$

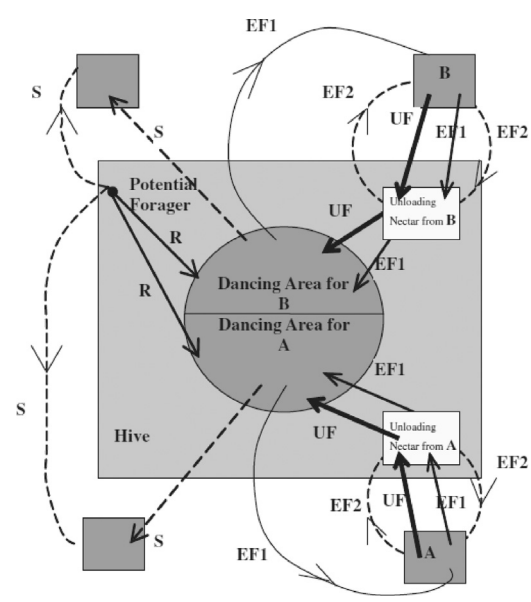

Fig. 4. The behavior of different bees.

Step 5: Repeat the steps 2 and 3 until we get the best fitness and its parameters.

Step 6: If the searching times surrounding an employed bee Bas exceeds a certain threshold Limit, but still could not find better solutions, then the location vector can be re-initialized.

$$X_i(n+1) = \begin{cases} X_{\min} + rand(0,1)(X_{\max} - X_{\min}), Bas > Limit \\ X_i(n), Bas < Limit \end{cases} \quad (22)$$

Step 7: Output the optimal fitness value f best and correlative parameters.

Fig. 5 shows the detailed learning process of ABC optimizing IG-FRBFNN, and the main idea includes the following steps:

Step 1: Image pre-processing

(1) Obtain the image, and convert it into grayscale format for further operation.

(2) Obtain the images with different noise processed from one existing image, as to represent the image gotten by different sensors. For this purpose, we applied Gaussian noise with 0.01 variance and salt and pepper noise with 0.1 noise density.

(3) Adopt the image fusion method mentioned in Section 2 to detect the regional features of each image, in which case we obtain a multi-dimension data matrix.

Step 2: Fussy C-Means

According to Eqs. (3)–(7), the sample data is divided into a certain number of categories.

Step 3: Information granulation

According to one equation of (8)–(11), we could express the consequent polynomials by OLS-based learning algorithm determining the coefficients values of the polynomials. In case that the input data can be used by network, we may normalize the data by previous calculation.

Step 4: Training of network

Execute the first training of network. In this step, the neutral network toolbox of MATLAB 7.11.0 (R2010b) is used.

Step 5: Initialization of ABC algorithm

Take the weights and thresholds of network, as the parameters to be optimized by ABC. Initialize the parameters of ABC algorithm mentioned in Section 4. The network output values, which represent the error of trained data, are taken as the objective function of ABC.

Step 6: The employed bees search around their current positions to find new solutions, and update their positions if the new fitness value is higher than the original value.

Step 7: The unemployed bees apply the roulette selection method to choose the bee individual that possesses a relatively good fitness value as the leading bee, according to the calculated fitness results of employed bees. Each recruited unemployed bee continues to search new solutions just around the leading bee's solution space, and calculate their fitness values. If the value of the new solution is better than the original value, the unemployed bee converts into an employed bee, which means update the positions of the employed bees, and continue exploring with *Bas* re-initialized as 0, or else, keep searching around, and its *Bas* value plus one.

Step 8: If the search times *Bas* is larger than certain threshold *Limit*, the employed bee gives up the solution, and research the new food resources, which is realized by reinitializing the geometric parameters and calculating the fitness value.

Step 9: Store the best solution parameters and the best fitness value.

Step 10: If $T < T_{max}$, go to Step 6. Otherwise, output the optimal parameters and optimal fitness value.

Step 11: By obtaining the best solution parameters and the best fitness value, we get also the optimized values of weights and thresholds of network, and we train the network to obtain a better result.

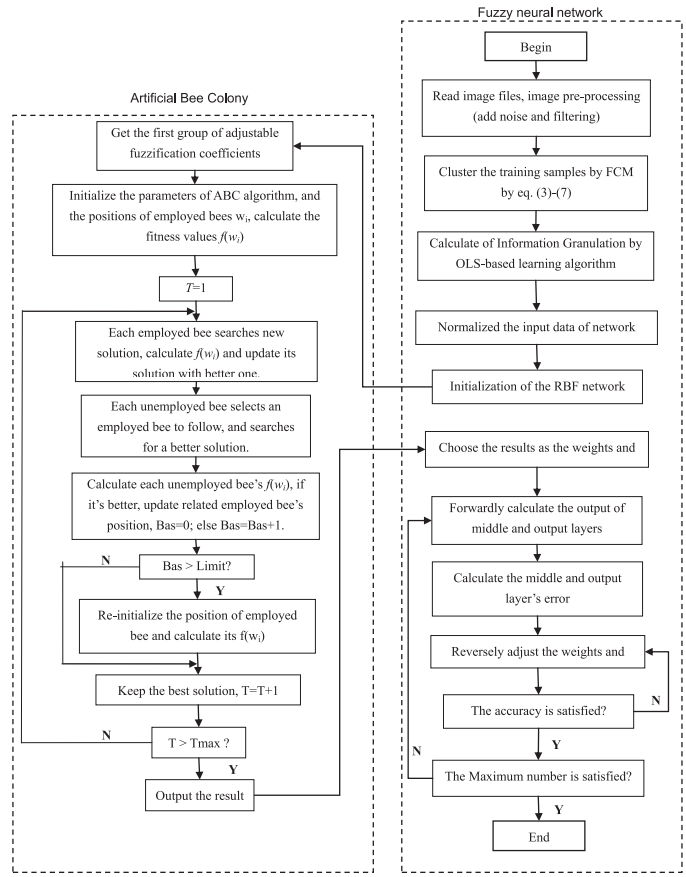

Fig. 5. Learning process of ABC optimized IG-FRBFNN.

5. Experimental results

5.1. Image pre-processing

To illustrate the concept of IG-FRBFNN and in order to evaluate the performances of the proposed image fusion method, our experiments are performed on the images with different noise processed from one existing image, which is a detail picture of the famous image of Lena Söderberg used in many image processing experiments. We processed it to get the noisy images by Gaussian noise with 0.01 variance and salt and pepper noise with 0.1 noise density as shown in Fig. 6(a) and (b).

Adopting the image fusion method mentioned in Section 2 and according to Eq.(1), we could detect the regional features of each image. The window size of the regional features is set to be 3×3 pixels. Fig. 6 shows the images in processing of experimental results by using our proposed approach. Fig. 6(c) and (d) is the calculation results of region feature of Fig. 6(a) and (d), and it is obvious that our proposed method can efficiently express the information of the pictures provided. Both the feature of the edges and the detail of the noisy point can be evidently presented.

5.2. Clustering and calculation before learning

By the two data of region feature, we obtain our sample as a two-dimensional data matrix, as shown, the *x*–*y* axes, respectively, corresponding to the same pixel from the two images of regional

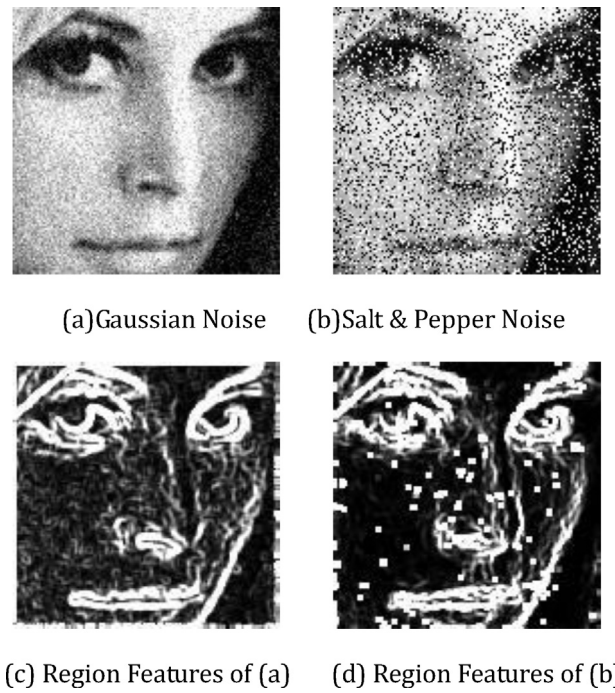

Fig. 6. Experimental results. (a) Gaussian noise, (b) salt and pepper noise, (c) region features of (a) and (d) region features of (b).

feature Fig. 6(c) and (d). Here the feature 1 means the value of pixel in Fig. 6(c) and the feature 2 in Fig. 6(d). According to the principle mentioned in Section 3.1 and Eqs. (3)–(7), the sample data is divided into three categories, which would be called high, medium and low categories.

Fig. 7 shows that FCM method can well distinguish between different data points which are in well-separated distribution of different sub-classes. At the same time, the three cluster centers calculated are marked using a black asterisk.

With considering the concert situation in our image problem and in order to simplify the question, we admit that the consequent polynomials to be the second-order polynomial (quadratic type), Eq. (10).

Calculating the coefficients of polynomial by the OLS-based learning algorithm introduced in Section 3, by one cluster after another, we have obtained the value of coefficients of the linear part of the polynomial according to Eq. (17) as shown in Table 1; the coefficient of the secondary order is randomly determined between [1,1] at current, which is optimized and trained later. Fig. 8(a)–(c) is the image displayed intuitively of the result, and Fig. 8(d) is the global result.

It is also obvious that the results of IG can be calculated performance characteristics of the target image, including edge and some local information. On the other hand, there is still a wide gap between some of the details. It also reflects the need to progress in the following neural.

Table 1
Coefficients in polynomial.

	Cluster 1	Cluster 2	Cluster 3
a_{i0}	0.575954	0.577088	0.575874
a_{i1}	0.000136723	0.000326508	−6.23214e−005
a_{i2}	−2.32009e−005	−4.8396e−005	5.2984e−005

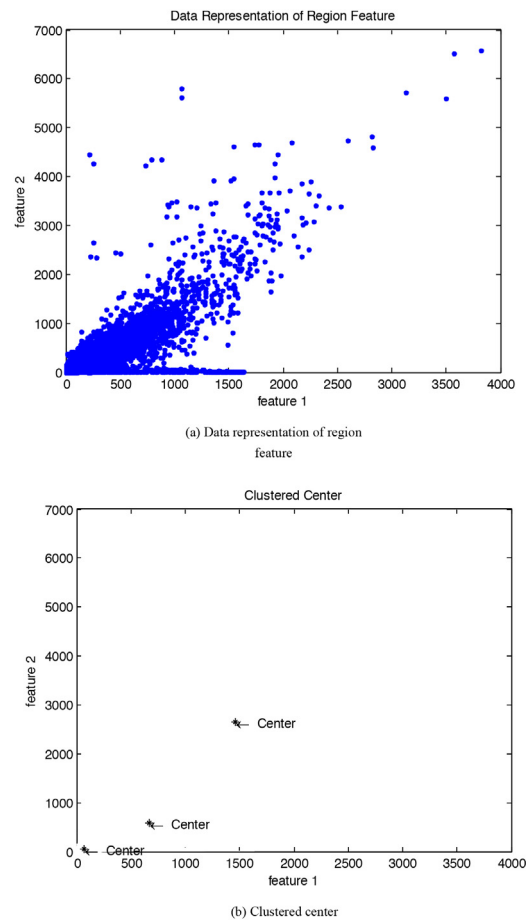

Fig. 7. Fuzzy C-Means cluster (cluster = 3). (a) Data representation of region feature and (b) clustered center.

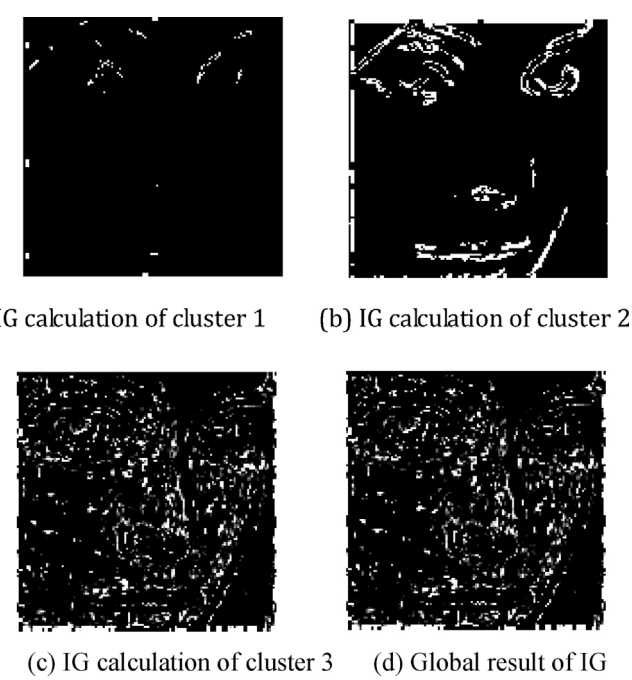

Fig. 8. Experimental results of IG calculation. (a) IG calculation of cluster 1, (b) IG calculation of cluster 2, (c) IG calculation of cluster 3 and (d) global result of IG.

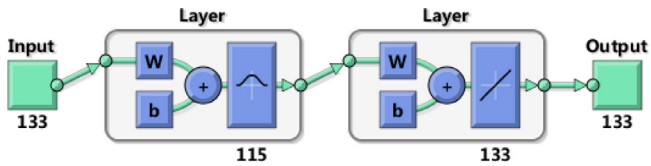

Fig. 9. Structure of RBFNN.

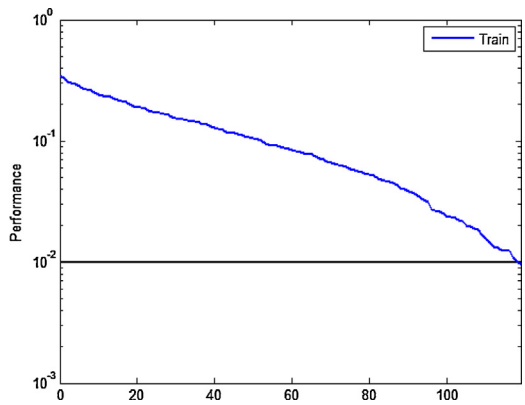

Fig. 10. Performance of RBFNN (performance = 0.00957923, goal = 0.01).

5.3. RBF neutral network training

As the image experimented is of size 133 × 133, that a set of 133 two-input data sampled uniformly and the corresponding target data from the two images is used as the training data. Another 133 input-target data from the original image (also interpreted as a priori knowledge) are also used as the testing data. The output picture is obtained by the calculation in meantime with the training of network.

By using the MATLAB radial basis function neural network toolbox, we have established the structure of the neural network shown in Fig. 9. 133 which represents the size of our input image and 115 is the number of network training in the experiment running. We set the target network error vector as 0.01. From Fig. 10 we can see that the training error of the results continues to reduce, and ultimately to achieve the target output after the error map. Table 2 presents a serious of results by several times of experiments, in which the MSE means mean squared normalized error performance function, by which we concluded that the RBF network could basically achieve the goal within a certain epochs.

5.4. Optimization by ABC algorithm

Take the weights and thresholds of the previous network, as the parameters to be optimized by ABC. Initialize the parameters of ABC algorithm mentioned in Section 4 as follows: $Ns = 50$, $Ne = Ns/2 = 25$,

Table 2
Coefficients in RBF network.

Number of test	Goal/error vector	Epochs/neurons	MSE
1	0.01	117	0.0099942
2	0.01	115	0.00978785
3	0.01	116	0.00999824
4	0.01	113	0.00970231
5	0.01	119	0.00976953
6	0.01	115	0.00983611
7	0.01	116	0.00994747
8	0.01	114	0.00969407
9	0.01	114	0.00956542
10	0.01	116	0.00955412

Fig. 11. Output by IG-FRBFNN.

the maximum cycling number $T_{\max} = 50$, and the limit of search times $Bas_{Limit} = 20$.

With artificial RBF network, the network output value is taken as the objective function of ABC. As shown in Fig. 11, ABC algorithm is used to optimize the network with the fitness is improved along with the iteration. Finally we obtain the optimal parameters (the weights and thresholds), and the output picture is obtained by the calculation in meantime with the training of network of which the sample input data is the characteristic value from the two images and the test data is from the original image (also interpreted as a priori knowledge).

By using the radial basis function neural network toolbox, we have established the structure of the neural network shown the Fig. 9, and 133 represents the size of our input image and 115 is the number of network training. We set the target network error vector as 0.01. From the figure we can see that the training error of the results continue to reduce, and ultimately to achieve the target output after the error map. Take the weights and thresholds of this network, as the parameters to be optimized by ABC. With artificial RBF network, the network output value is taken as the objective function of ABC. As shown in figure, the iterative ABC algorithm is used to optimize the network. Finally we obtain the optimal parameters (the weights and thresholds), and in the mean time the output of the network as shown Fig. 12. By visual observation of Fig. 12, it can be concluded that the proposed image fusion method provides as a whole a performance that meets the visual requirements that we may observe the clear figure of the face in the fused image.

5.5. Application in another case

In order to better validate our approach, we applied it in another case in which the images are processed from a picture (size 133 × 133) of the New Main Building of Beihang University, Beijing, China.

Fig. 12(a)–(d) is the pre-processed images and the regional features of images. Figs. 13 and 14 and Table 3 show result of FCM and IG calculation, in which the samples are also well separated by proposed method. From Fig. 15 we conclude that the RBF network could also achieve the goal (equals to 0.01) within a certain epochs (equals to 110 in this test). By observation of the curve of iteration of ABC algorithm in Fig. 12 and the fused image in Fig. 17, we conclude that the proposed approach is also effective to this type of images. From Fig. 17, we can obtain a visual observation that the figure of the building in the picture, and the contrast between the building and the background can be clearly represented (Fig. 16).

Table 3
Case 2: coefficients in polynomial.

	Cluster 1	Cluster 2	Cluster 3
a_{i0}	0.532608	0.530436	0.530568
a_{i1}	−1.6074e−005	1.85991e−005	4.76468e−005
a_{i2}	−9.18538e−005	−3.56823e−005	−3.27667e−005

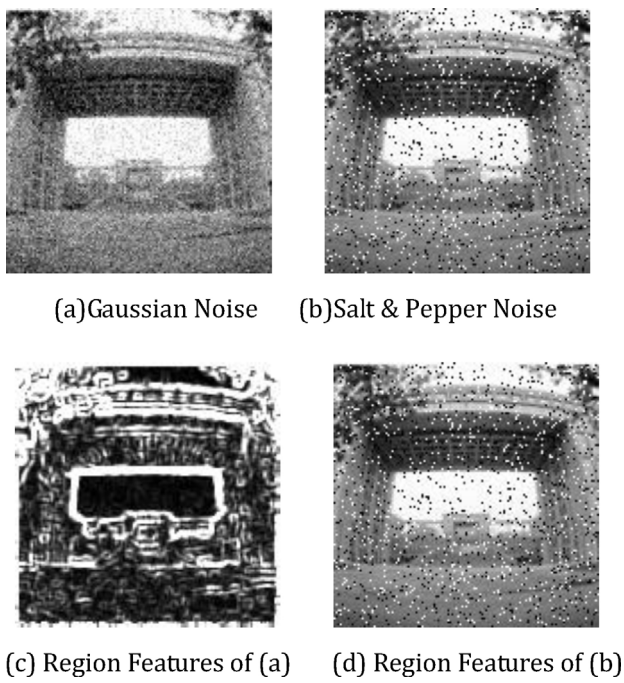

Fig. 12. Case 2: experimental results. (a) Gaussian noise, (b) salt and pepper noise, (c) region features of (a) and (d) region features of (b).

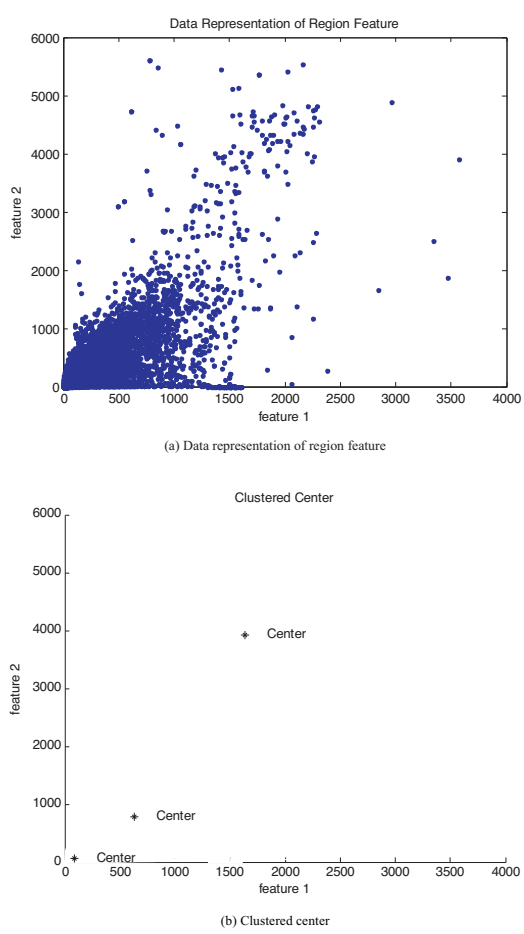

Fig. 13. Case 2: Fuzzy C-Means cluster (cluster = 3). (a) Data representation of region feature and (b) clustered center.

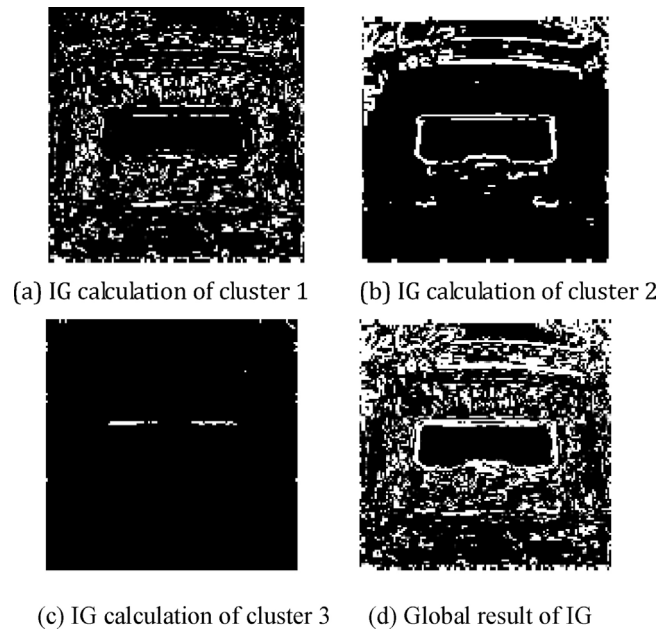

Fig. 14. Case 2: Experimental results of IG calculation. (a) IG calculation of cluster 1, (b) IG calculation of cluster 2, (c) IG calculation of cluster 3 and (d) global result of IG.

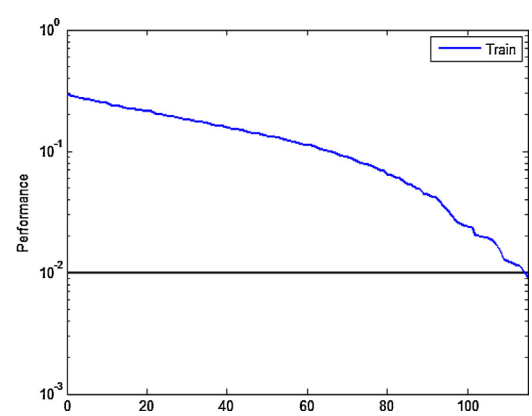

Fig. 15. Performance of RBFNN (performance = 0.00916653, goal = 0.01).

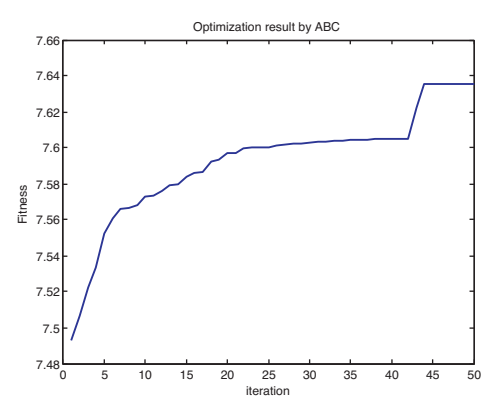

Fig. 16. Case 2: Performance of ABC algorithm.

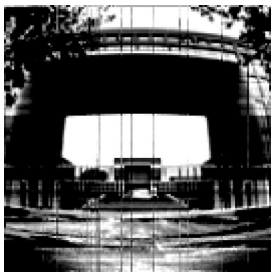

Fig. 17. Case 2: Output by IG-FRBFNN.

6. Conclusions

In this work, a novel method of ABC optimized IG-FRBFNN for solving the image fusion problem is proposed. Image fusion using the fuzzy neural network approach for surveillance is studied. The structure of IG-FRBRNN is illustrated and the fuzzy neural network approach to image fusion is presented. ABC algorithm is exploited to carry out the structural and parametric optimization of the model respectively while the optimization is of multi-objective character as it is aimed at the simultaneous minimization of complexity and maximization of accuracy. Series of experimental results are presented to verify the feasibility and effectiveness of the proposed approach.

Our future work will focus on applying our proposed hybrid model in more complicated patterns and real-world images, and implement this novel approach in parallel embedded processors. In addition, our proposed algorithm will be also applied to image enhancement and other image processing, which will be another key issue in the near future.

Acknowledgements

This work was partially supported by Natural Science Foundation of China (NSFC) under grant #61273054, #60975072 and #60604009, Program for New Century Excellent Talents in University of China under grant #NCET-10-0021, Aeronautical Foundation of China under grant #20115151019, the Fundamental Research Funds for the Central Universities of China under grant #YWF-11-03-Q-012 and YWF-10-01-A18, and Opening Foundation of State Key Laboratory of Virtual Reality Technology and Systems of China under grant # VR-2011-ZZ-01.

References

[1] Y. Luo, L. Chen, Y. Luo, SGNN to image fusion based on multi-feature clustering, in: Proceedings of IEEE International Conference on Measuring Technology and Mechatronics Automation, Changsha, 2010, pp. 329–333.
[2] A. Mumtaz, A. Majid, Genetic algorithms and its application to image fusion, in: Proceedings of International Conference on Emerging Technologies, Pakistan, 2008, pp. 6–10.
[3] Z. Xue, R.S. Blum, Concealed weapon detection using color image fusion, in: Proceedings of the 6th International Conference on Information Fusion, Gallup, NM, 2003, pp. 1389–1394.
[4] J. Yang, R.S. Blum, A statistical signal processing approach to image fusion for concealed weapon detection, in: Proceedings of 2002 Int. Conf. Image Processing, vol. 1, Rochester, NY, 2002, pp. 513–516.
[5] Z. Liu, Z. Xue, R.S. Blum, R. Laganière, Multisensory concealed weapon detection by using a multi-resolution mosaic approach, in: Proceedings of Vehicular Technology Conf. 7, 2004, pp. 1601–4597.
[6] S. Zheng, W.-Z. Shi, J. Liu, G.-X. Zhu, J.-W. Tian, Multisource image fusion method using support value transform, IEEE Trans. Image Process. 16 (7) (2007) 1831–1839.
[7] S. Li, B. Yang, Multifocus image fusion using region segmentation and spatial frequency, Image Vision Comput. 26 (7) (2008) 971–979.
[8] S. Li, J.T. Kwok, Y. Wang, Using the discrete wavelet frame transform to merge Landsat TM and SPOT panchromatic images, Inform. Fusion 3 (1) (2002) 17–23.
[9] H. Duan, S. Shao, B. Su, L. Zhang, New development thoughts on the bio-inspired intelligence based control for unmanned combat aerial vehicle, Sci. China Technol. Sci. 53 (8) (2010) 2025–2031.
[10] H. Duan, P. Li, Progress in control approaches for hypersonic vehicle, Sci. China Technol. Sci. 55 (10) (2012) 2965–2970.
[11] F. Liu, H. Duan, Y. Deng, A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching, Optik 123 (21) (2012) 1955–1960.
[12] H. Duan, C. Xu, S. Liu, S. Shao, Template matching using chaotic imperialist competitive algorithm, Pattern Recogn. Lett. 31 (13) (2010) 1868–1875.
[13] C. Xu, H. Duan, Artificial Bee Colony optimized edge potential function approach to target recognition for low-altitude aircraft, Pattern Recogn. Lett. 31 (13) (2010) 1759–1772.
[14] H. Duan, C. Xu, Z. Xing, Artificial Bee Colony optimization based quantum evolutionary for continuous optimization problems, Int. J. Neural Syst. 20 (1) (2010) 39–50.
[15] L. Bocchi, G. Coppini, J. Nori, G. Valli, Detection of single and clustered microcalcifications in mammograms using fractals models and neural networks, Med. Eng. Phys. 26 (4) (2004) 303–312.
[16] S. Joo, Y.S. Yang, W.K. Moon, H.C. Kim, Computer-aided diagnosis of solid breast nodules: use of an artificial neural network based on multiple sonographic features, IEEE Trans. Med. Image 23 (10) (2004) 1292–1300.
[17] N. Theera-Umpon, P.D. Gader, System-level training of neural networks for counting white blood cells, IEEE Trans. Syst. Man Cybern. Part C 32 (1) (2002) 48–53.
[18] G. Coppini, S. Diciotti, M. Falchini, N. Villari, G. Valli, Neural networks for computer-aided diagnosis: detection of lung nodules in chest radiograms, IEEE Trans. Inf. Technol. Biomed. 7 (4) (2003) 344–357.
[19] C.I. Christodoulou, S.K. Kakkos, A. Nicolaides, Texture based classification of atherosclerotic carotid plaques, IEEE Trans. Med. Image 22 (7) (2003) 902–912.
[20] J. Moody, C.J. Darken, Fast learning in networks of locally-tuned processing units, Neural Comput. 1 (2) (1989) 281–294.
[21] I. Maglogiannis, H. Sarimveis, C.T. Kiranoudis, A.A. Chatziioannou, N. Oikonomou, V. Aidinis, Radial basis function neural networks classification for the recognition of idiopathic pulmonary fibrosis in microscopic images, IEEE Trans. Inform. Technol. Biomed. 12 (1) (2008) 42–54.
[22] B.-J. Park, J.-N. Choi, W.-D. Kim, S.-K. Oh, Analytic design of information granulation-based fuzzy radial basis function neural networks with the aid of multiobjective particle swarm optimization, Int. J. Intell. Comput. Cybern. 5 (1) (2012) 4–35.
[23] J. Yu, H. Duan, S.-K. Oh, Fuzzy radial function neural networks approach to image fusion, Proceedings of KIIS Fall Conference 21 (2) (2011) 3–5.
[24] I.W. Sandberg, Universal approximation using radial basis functions network, Neural Comput. 3 (2) (1991) 246–257.
[25] F. Girossi, T. Poggio, Networks and the best approximation property, Biol. Cybern. 63 (3) (1990) 169–176.
[26] S. Lee, R.M. Kil, A Gaussian potential function network with hierarchically self-organizing learning, Neural Networks 4 (2) (1991) 207–224.
[27] R. Aguilar-Ponce, J.-L. Tecpanecatl-Xihuitl, A. Kumar, M. Bayoumi, Pixel-level image fusion scheme based on linear algebra, circuits and systems, in: Proceedings of 2007 IEEE International Symposium on Circuits and Systems, New Orleans, LA, 2007, pp. 2658–2661.
[28] Y.-P. Wang, J.-W. Dang, Q. Li, S. Li, Image fusion approach based on fuzzy radial basis neural networks, Comput. Eng. Appl. 43 (25) (2007) 48–50.
[29] M. Alper Selver, C. Güzelis, Semiautomatic transfer function initialization for abdominal visualization using self-generating hierarchical radial basis function networks, IEEE Trans. Vis. Comput. Graphics 15 (3) (2009) 395–409.
[30] M. Fathian, B. Amiri, A. Maroosi, Application of honey bee mating optimization algorithm on clustering, Appl. Math. Comput. 190 (2) (2007) 1502–1513.