Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure

Songcan Chen[1, 2*] and Daoqiang Zhang[1]

[1]*Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, People's Republic of China*

[2]*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100080, P.R. China*

***Abstract* ---** Fuzzy c-means clustering (FCM) with spatial constraints (FCM_S) is an effective algorithm suitable for image segmentation. Its effectiveness contributes not only to introduction of fuzziness for belongingness of each pixel but also to exploitation of spatial contextual information. Although the contextual information can raise its insensitivity to noise to some extent, FCM_S (1) still lacks enough robustness to noise and outliers and (2) is not suitable for revealing non-Euclidean structure of the input data due to the use of Euclidean distance (L_2 norm). In this paper, to overcome the above problems, we first propose two variants, FCM_S_1 and FCM_S_2, of FCM_S to aim at simplifying its computation and then extend them, including

[*] Corresponding author. Tel.:+86-25-8489-2805.

E-mail address: s.chen@nuaa.edu.cn (S.-C. Chen), daoqz@mail.com (D.-Q. Zhang).

FCM_S, to corresponding robust kernelized versions KFCM_S, KFCM_ S_1 and KFCM_ S_2 by the kernel methods. Our main motives of using the kernel methods consist in: (1) inducing a class of robust non-Euclidean distance measures for the original data space to derive new objective functions and thus clustering the non-Euclidean structures in data; (2) enhancing robustness of the original clustering algorithms to noise and outliers, and (3) still retaining computational simplicity. The experiments on the artificial and real-world datasets show that our proposed algorithms, especially with spatial constraints, are more effective.

***Index Terms* ---** Image segmentation, fuzzy C-means clustering, kernel methods, kernel-induced distance measures, robustness, spatial constraints.

I. Introduction

Image segmentation plays an important role in a variety of applications such as robot vision, object recognition, and medical imaging [1]-[3]. In the last decades, fuzzy segmentation methods, especially the fuzzy c-means algorithm (FCM) [4], have been widely used in the image segmentation [26]-[27] and such a success chiefly attributes to introduction of fuzziness for the belongingness of each image pixel, which makes the clustering methods able to retain more information from the original image than the crisp or hard segmentation methods [5]. So-called clustering is to partition a set of given observed input data vectors or image pixels into *c*-clusters so that members of the same cluster are similar to one another than to members of other clusters where the number, *c*, of clusters is usually pre-defined or set by some validity criterion or a priori knowledge.

Generally, clustering methods can be categorized into [25]: hierarchical, graph theoretic, decomposing a density function and minimizing an objective function. In this paper, we will focus on clustering methods by minimization of objective function and apply them to segment images.

Mathematically, the standard FCM objective function of partitioning a dataset $\{\mathbf{x}_k\}_{k=1}^{N} \subset R^d$ into c clusters is given by

$$J_m = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m} \| \mathbf{x}_k - \mathbf{v}_i \|^2 \tag{1a}$$

where $\|\cdot\|$ stands for the Euclidean norm. Equivalently, Eq. (1a) can, in an inner or scalar product form, be rewritten as

$$J_m = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^m (\mathbf{x}_k^T\mathbf{x}_k - 2\mathbf{x}_k^T\mathbf{v}_i + \mathbf{v}_i^T\mathbf{v}_i) \tag{1b}$$

where $\{\mathbf{v}_i\}_{i=1}^{c} (\subset R^d)$ are the centroids or prototypes of the clusters, T denotes matrix transpose, the parameter m is a weighting exponent on each fuzzy membership and the array U=[u_{ik}] is a fuzzy partition matrix satisfying

$$U \in \left\{ u_{ik} \in [0,1] \left| \sum_{i=1}^{c} u_{ik} = 1, \forall k \text{ and } 0 < \sum_{k=1}^{N} u_{ik} < N, \forall i \right. \right\}. \tag{2}$$

Although the original intensity-based FCM algorithm functions well on segmenting most noise-free images, it fails to segment images corrupted by noise, outliers and other imaging artifacts, such as the intensity inhomogeneity induced by the radio-frequency coil in magnetic resonance imaging (MRI), and thus leads to its non-robust results mainly due to (1) the use of non-robust Euclidean distance and (2) disregard of spatial contextual information in image. To deal with the first problem, some researchers adopted so-called robust distance measures, such as L_p norms ($0<p\leq1$)[28-30], to replace the L_2 norm in the FCM objective function reducing the effect of outliers on clustering results, and while many other algorithms have also been proposed to deal with the second problem by incorporating spatial information into original FCM objective function [33], [5]-[10], for example, Ahmed *et al* modified FCM objective function by incorporating spatial constraints (called as FCM_S later)[9]. However, besides the increase in computational time due to such introduction of spatial constraints, two problems with the FCM_S algorithm still suffer: one is insufficient robustness to outliers as shown in our experiments, and the other is difficult to cluster non-Euclidean structure in data such as non-spherical shape

clusters.

In this paper, we first propose two variants, FCM_S$_1$ and FCM_S$_2$, of FCM_S in [9] with an intent to simplify the computation of parameters and then extend them, together with the original FCM_S, to corresponding kernelized versions, KFCM_S, KFCM_ S$_1$ and KFCM_ S$_2$, by the kernel function substitution. Goals of adopting the kernel functions aim: (1) to induce a class of new robust distance measures for the input space and then replace non-robust L$_2$ measure to cluster data or segment images more effectively; (2) to more likely reveal inherent non-Euclidean structures in data and (3) to retain simplicity of computation.

The kernel methods [11]-[16] are one of the most researched subjects within machine learning community in the recent few years and have widely been applied to pattern recognition and function approximation. Typical examples are support vector machines [12]-[14], kernel Fisher linear discriminant analysis (KFLDA) [15], kernel principal component analysis (KPCA) [16], kernel perceptron algorithm [24], and just name a few. The fundamental idea of the kernel method is to first transform the original low-dimensional inner-product input space into a higher (possibly infinite) dimensional feature space through some nonlinear mapping where complex nonlinear problems in the original low-dimensional space can more likely be linearly treated and solved in the transformed space according to the well-known Cover's theorem [17]. However, usually such mapping into high-dimensional feature space will undoubtedly lead to an exponential increase of computational time, i.e., so-called curse of dimensionality. Fortunately, adopting kernel functions to substitute an inner product in the original space, which exactly corresponds to mapping the space into higher-dimensional feature space, is a favorable option. Therefore, the inner product form in Eq. (1b) leads us to applying the kernel methods to cluster complex data [21]-[22]. However, compared to the approaches presented in [21]-[22], a major difference of our proposed approach in this paper is that we do not adopt so-called dual representation for each centroid, i.e., a linear combination of all given dataset samples, but directly transform all the centroids in the original space, together with given data samples, into high-dimensional feature space with an (implicitly) mapping. Such a direct transformation brings us the following benefits: (1) inducing a class of robust non-Euclidean distance measures if we employ robust kernels; (2) inheriting the computational simplicity of the FCM; (3) interpreting clustering results intuitively; even (4) coping with data set with missing values easily [31]. The proposed algorithms, especially with the spatial constraints, are shown to be more robust to noise and outlier in image segmentation than the algorithms without the kernel substitution.

The rest of this paper is organized as follows. In Section II, the conventional spatial FCM algorithm (FCM_S) for image segmentation is introduced and two new low-complexity variants are derived. In Section III, we obtain a group of kernel fuzzy clustering algorithms with spatial constraints for image segmentation by first replacing the Euclidean norm in the objective functions with kernel-induced non-Euclidean distance measures and then minimizing these new objective functions. The experimental comparisons are presented in Section IV. Finally, Section V gives our conclusions and several issues for future work.

II. Fuzzy clustering with spatial constraints (FCM_S) and its Variants

In [9], an approach was proposed to increase the robustness of FCM to noise by directly modifying the objective function defined in (1) as follows

$$J_m = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^m \,||\, \mathbf{x}_k - \mathbf{v}_i \,||^2 + \frac{\alpha}{N_R}\sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^m \sum_{r\in N_k} ||\, \mathbf{x}_r - \mathbf{v}_i \,||^2 \tag{3}$$

where N_k stands for the set of neighbors falling into a window around $\mathbf{x}_k$ and N_R is its cardinality. The parameter α in the second term controls the effect of the penalty. In essence, the addition of the second term in (3), equivalently, formulates a spatial constraint and aims at keeping continuity on neighboring pixel values around $\mathbf{x}_k$. By an optimization way similar to the standard FCM algorithm, the objective function J_m can be minimized under the constraint of U as stated in (2). A necessary condition on u_{ik} for (3) to be at a local minimum is

$$u_{ik} = \frac{\left(||\, \mathbf{x}_k - \mathbf{v}_i \,||^2 + \frac{\alpha}{N_R}\sum_{r\in N_k} ||\, \mathbf{x}_r - \mathbf{v}_i \,||^2 \right)^{-1/(m-1)}}{\sum_{j=1}^{c}\left(||\, \mathbf{x}_k - \mathbf{v}_j \,||^2 + \frac{\alpha}{N_R}\sum_{r\in N_k} ||\, \mathbf{x}_r - \mathbf{v}_j \,||^2 \right)^{-1/(m-1)}} \tag{4}$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^m \left(\mathbf{x}_k + \frac{\alpha}{N_R}\sum_{r\in N_k} \mathbf{x}_r \right)}{(1+\alpha)\sum_{k=1}^{n} u_{ik}^m} \tag{5}$$

A shortcoming of (4) and (5) is that computing the neighborhood terms will take much more time than FCM. In this section, we will present two low-complexity modifications or variants to (3). Firstly, we notice that by simple manipulation, $\frac{1}{N_R}\sum_{r\in N_k} \left\| \mathbf{x}_r - \mathbf{v}_i \right\|^2$ can equivelently be written as $\frac{1}{N_R}\sum_{r\in N_k} \left\| \mathbf{x}_r - \bar{\mathbf{x}}_k \right\|^2 + \left\| \bar{\mathbf{x}}_k - \mathbf{v}_i \right\|^2$, where

$\overline{\mathbf{x}}_k$ is a mean of neighboring pixels lying within a window around $\mathbf{x}_k$. Unlike (3), $\overline{\mathbf{x}}_k$ can be computed in advance, thus the clustering time can be saved when $\frac{1}{N_R}\sum_{r\in N_k}\left\|\mathbf{x}_r-\mathbf{v}_i\right\|^2$ in (3) is replaced with $\left\|\overline{\mathbf{x}}_k-\mathbf{v}_i\right\|^2$. Hence the simplified objective functions can be rewritten as

$$J_m=\sum_{i=1}^{c}\sum_{k=1}^{N}u_{ik}^m\|\mathbf{x}_k-\mathbf{v}_i\|^2+\alpha\sum_{i=1}^{c}\sum_{k=1}^{N}u_{ik}^m\|\overline{\mathbf{x}}_k-\mathbf{v}_i\|^2 \tag{6}$$

Similarly, we can obtain the following solution by minimizing (6)

$$u_{ik}=\frac{\left(\|\mathbf{x}_k-\mathbf{v}_i\|^2+\alpha\|\overline{\mathbf{x}}_k-\mathbf{v}_i\|^2\right)^{-1/(m-1)}}{\sum_{j=1}^{c}\left(\|\mathbf{x}_k-\mathbf{v}_j\|^2+\alpha\|\overline{\mathbf{x}}_k-\mathbf{v}_j\|^2\right)^{-1/(m-1)}} \tag{7}$$

$$\mathbf{v}_i=\frac{\sum_{k=1}^{n}u_{ik}^m\left(\mathbf{x}_k+\alpha\overline{\mathbf{x}}_k\right)}{(1+\alpha)\sum_{k=1}^{n}u_{ik}^m} \tag{8}$$

Therefore (4) and (5) obtain simplification to some extent. In order to enhance robustness of clustering, $\overline{\mathbf{x}}_k$ can be considered to take as median of the neighbors within a specified window around $\mathbf{x}_k$. Similarly from (7) and (8), we can see that α controls the tradeoff between the original image and the corresponding mean- or median-filtered image. When α is set to zero, the algorithm is equivalent to the original FCM, while α approaches infinite, the algorithm acquires the same effect as the original FCM on the mean or median filtered image respectively.

For convenience of notation later, we will name the algorithm using (4) and (5) FCM_S, the ones using (7) and (8) with mean and median filtering FCM_S$_1$ and FCM_S$_2$ respectively. The above algorithms can uniformly be summarized in the following steps.

Algorithm 1:

Step 1) Fix the number c of these prototypes or clusters and then select initial prototypes (centroids) , and set $\varepsilon > 0$ to a very small value.

Step 2) For FCM_S1 and FCM_S2 only, compute the mean or median filtered image.

Step 3) Update the partition matrix using Eqs. (4) (FCM_S) or Eqs. (7) (FCM_S1 and FCM_S2).

Step 4) Update the centroids using Eqs. (5)(FCM_S) or Eqs. (8)(FCM_S1 and FCM_S2).

Repeat Steps 3)-4) until the following termination criterion is satisfied:

$$\left\| \mathbf{V}_{new} - \mathbf{V}_{old} \right\| < \varepsilon$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_c]$ are the vectors of cluster centroids.

III. Fuzzy c-means with spatial constraints based on kernel-induced distance

A. The kernel methods and kernel functions

Let $\Phi : \mathbf{x} \in X \subseteq R^d \mapsto \Phi(\mathbf{x}) \in F \subseteq R^H (d \ll H)$ be a nonlinear transformation into a higher (possibly infinite)-dimensional feature space *F*. In order to explain how to use the kernel methods, let us recall a simple example[13]. If $\mathbf{x} = \left[x_1, x_2\right]^T$ and $\Phi(\mathbf{x}) = \left[x_1^2, \sqrt{2}x_1 x_2, x_2^2\right]^T$, where x_i is the *i*th component of vector **x**. Then the inner product between Φ(**x**) and Φ(**y**) in the feature space *F* are: $\Phi(\mathbf{x})^{\mathrm{T}}$ $\Phi(\mathbf{y})$=$[(x_1)^2, \sqrt{2}\, x_1\, x_2, (x_2)^2]^T$ $[(y_1)^2, \sqrt{2}\, y_1\; y_2, (y_2)^2]$=$(\mathbf{x}^T\mathbf{y})^2$=$K(\mathbf{x}, \mathbf{y})$. Thus, in order to compute the inner products in *F*, we can use kernel representation *K*(**x**, **y**), without explicitly using transformation or mapping Φ (thus overcoming curse of dimensionality). It is a direct consequence from [11]: every linear algorithm that only uses inner products can be easily extended to a nonlinear version only through the kernels satisfying the Mercer's conditions [11]. In the following are given typical radial basis function (RBF) and polynomial kernels:

$$K(\mathbf{x},\mathbf{y}) = \exp\left(\frac{-(\sum_{i=1}^{d} |x_i - y_i|^a)^b}{\sigma^2} \right) \tag{9}$$

where d is the dimension of vector $x; a \geq 0; 1 \leq b \leq 2$. Obviously, $K(\mathbf{x},\mathbf{x}) = 1$ for all $\mathbf{x}$ and the above RBF kernels, and a polynominal with degree of p

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^p \tag{10}$$

From Eq.(1b), we can now use kernel functions to substitute the inner products there to realize an implicit mapping into some feature space so that their corresponding kernelized versions are constructed. Actually, there are two ways to kernelize FCM and FCM_S: one is to view every centroid as a mapped point in the feature space and use their dual forms, i.e., a linear combinatin of all data samples, to replace original certroids to get clustering results in the feature space but not in the original space as in [21-22]. In this way, the resulting clustering are not easily interpreted intuitively in the original space. The other [18] is to still view every centroid as a data point in the original space like given samples and directly transform them, together with the data samples, into the feature spaces and then carry out clustering. A advantage of doing so is that clustering is performed still in the original space and thus the results can easier be interpreted intuitively as will be done below. In this paper, we adopt the latter kernelizing way.

B. FCM based on kernel-induced distance (KFCM)

With the above formulations, we are now in position to construct the kernelized version of the FCM algorithm and modify its objective function with the mapping Φ as follows

$$J_m^{\Phi} = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m \| \Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i) \|^2 \tag{11}$$

Now through the kernel substitution, we have

$$\begin{aligned} \| \Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i) \|^2 &= \left(\Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i)\right)^T \left(\Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i)\right) \\ &= \Phi(\mathbf{x}_k)^T \Phi(\mathbf{x}_k) - \Phi(\mathbf{v}_i)^T \Phi(\mathbf{x}_k) - \Phi(\mathbf{x}_k)^T \Phi(\mathbf{v}_i) + \Phi(\mathbf{v}_i)^T \Phi(\mathbf{v}_i) \\ &= K(\mathbf{x}_k, \mathbf{x}_k) + K(\mathbf{v}_i, \mathbf{v}_i) - 2K(\mathbf{x}_k, \mathbf{v}_i) \end{aligned} \tag{12}$$

in this way, a new class of non-Euclidean distance measures in original data space (also a squares norm in the feature space) are obtained. Obviously, different kernels will induce different measures for the original space, which leads to a new family of clustering algorithms. In particular, if the kernel function $K(\mathbf{x}, \mathbf{y})$ is taken as the RBF in Eq.(9), Eq. (12) can be simplified to $2\left(1 - K\left(\mathbf{x}_k, \mathbf{v}_i\right)\right)$. Furthermore, for sake of conveinence of manipulation below and robustness, in this paper, we only consider the Gaussian RBF(GRBF) kernel with *a*=2 and *b*=1 in Eq. (9) (in fact, by means of the Huber's robust statistics [19], [23], we can prove that the measures based on Eqs. (9) are robust but those based on polynomials are not), then Eq. (11) can be rewritten as

$$J_m^{\Phi} = 2\sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m \left(1 - K(\mathbf{x}_k, \mathbf{v}_i)\right) \tag{13}$$

Therefore an algorithm in [23] immediately becomes a special case of ours, where the distance measure is not induced from a viewpoint of the kernel but just is a direct definition. In addition, when the parameter σ in Eq. (9) is a sufficient large positive constant, we can again obtain the FCM and its robust versions such as the L$_p$-norm-based clustering algorithms [28]-[30] because from Eq. (9), we have $K(\mathbf{x},\mathbf{y}) \approx 1-[\frac{-(\sum_{i=1}^{d}|x_i - y_i|^a)^b}{\sigma^2}]$ and thus $J_m^\Phi \approx \frac{2}{\sigma^2}\sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^m (\sum_{j=1}^{d}|x_{kj} - v_{ij}|^a)^b$ in terms of Eq. (13). Obviously if a=p and b=1/p, J_m^Φ reduces to the objective function based on the L$_p$ norm.

By an optimization way similar to the FCM, J_m^Φ can be minimized under the constraint of U same as (2). Specifically, if we take its first derivatives with respect to u_{ik} and $\mathbf{v}_i$, and zero them respectively, two necessary but not sufficient conditions for J_m^Φ to be at local minimum will be obtained as below

$$u_{ik} = \frac{\left(1 - K(\mathbf{x}_k, \mathbf{v}_i)\right)^{-1/(m-1)}}{\sum_{j=1}^{c}\left(1 - K(\mathbf{x}_k, \mathbf{x}_j)\right)^{-1/(m-1)}} \tag{14}$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^m K(\mathbf{x}_k, \mathbf{v}_i)\mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^m K(\mathbf{x}_k, \mathbf{v}_i)} \tag{15}$$

It is evident that the obtained centroids or prototypes {$\mathbf{v}_\mathrm{i}$} still lie in the original space and not in the transformed higher dimensional feature space, thus the computational simplicity is still retained. In addition, it is shown that the KFCMs resulted from Eq. (11), i.e., Eq. (13), are robust to outliers and noise according to Huber's robust statistics [19], [23]. This characteristic can also give an intuitive explanation from (15): the data point $\mathbf{x}_k$ is endowed with an additional weight $K(\mathbf{x}_k, \mathbf{v}_i)$, which measures the similarity between $\mathbf{x}_k$ and $\mathbf{v}_i$, and when $\mathbf{x}_k$ is an outlier, i.e., $\mathbf{x}_k$ is far from the other data points, $K(\mathbf{x}_k, \mathbf{v}_i)$ will be very small, so the weighted sum of data points shall be suppressed and hence result in robustness. Our experiments also confirm that new algorithms are indeed more robust to outliers and noise than FCM.

C. FCM with spatial constraints based on kernel-induced distance (KFCM_S)

In this subsection, we will construct the corresponding kernelized version of the FCM_S algorithm. Similar to the derivation of KFCM, we kernelize criterion (3) and obtain the following new objective function through the newly-induced distance measure substitution,

$$JS_m^{\Phi} = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^m \left(1 - K(\mathbf{x}_k, \mathbf{v}_i)\right) + \frac{\alpha}{N_R}\sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^m \sum_{r \in N_k} \left(1 - K(\mathbf{x}_r, \mathbf{v}_i)\right) \qquad (16)$$

where $K(\mathbf{x}, \mathbf{y})$ is still taken as GRBF, N_k, $\mathbf{x}_k$, α and N_R are defined as before.

An iterative algorithm of minimizing (16) with respect to u_{ik} and $\mathbf{v}_i$ can similarly be derived again

$$u_{ik} = \frac{\left(\left(1 - K(\mathbf{x}_k, \mathbf{v}_i)\right) + \frac{\alpha}{N_R}\sum_{r \in N_k}(1 - K(\mathbf{x}_r, \mathbf{v}_i))^m\right)^{-1/(m-1)}}{\sum_{j=1}^{c}\left(\left(1 - K(\mathbf{x}_k, \mathbf{v}_j)\right) + \frac{\alpha}{N_R}\sum_{r \in N_k}(1 - K(\mathbf{x}_r, \mathbf{v}_j))^m\right)^{-1/(m-1)}} \qquad (17)$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^m \left(K(\mathbf{x}_k, \mathbf{v}_i)\mathbf{x}_k + \frac{\alpha}{N_R}\sum_{r \in N_k} K(\mathbf{x}_r, \mathbf{v}_i)\mathbf{x}_r\right)}{\sum_{k=1}^{n} u_{ik}^m \left(K(\mathbf{x}_k, \mathbf{v}_i) + \frac{\alpha}{N_R}\sum_{r \in N_k} K(\mathbf{x}_r, \mathbf{v}_i)\right)} \qquad (18a)$$

In practical realization of the algorithm, we use Eq. (18b) below to replace (18a) so as to further reduce computation of Eq. (18a). In fact, such a simplification is not able to cause bad effective to clustering results as shown in our experiments.

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^m \left(K(\mathbf{x}_k, \mathbf{v}_i) + \frac{\alpha}{N_R} \sum_{r \in N_k} K(\mathbf{x}_r, \mathbf{v}_i) \right) \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^m \left(K(\mathbf{x}_k, \mathbf{v}_i) + \frac{\alpha}{N_R} \sum_{r \in N_k} K(\mathbf{x}_r, \mathbf{v}_i) \right)} \tag{18b}$$

Similarly, a kernelized modification to Eq. (6) is

$$JS_m^{\Phi} = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m \left(1 - K(\mathbf{x}_k, \mathbf{v}_i)\right) + \alpha \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m \left(1 - K(\overline{\mathbf{x}}_k, \mathbf{v}_i)\right) \tag{19}$$

where $\overline{\mathbf{x}}_k$ is defined as before and directly viewed a data point in the original space to be mapped by Φ and thus can be computed in advance and stored. And the objective function in (19) is minimized using the following alternate iterations

$$u_{ik} = \frac{\left(\left(1 - K(\mathbf{x}_k, \mathbf{v}_i)\right) + \alpha \left(1 - K(\overline{\mathbf{x}}_k, \mathbf{v}_i)\right) \right)^{-1/(m-1)}}{\sum_{j=1}^{c} \left(\left(1 - K(\mathbf{x}_k, \mathbf{v}_j)\right) + \alpha \left(1 - K(\overline{\mathbf{x}}_k, \mathbf{v}_j)\right) \right)^{-1/(m-1)}} \tag{20}$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^m \left(K(\mathbf{x}_k, \mathbf{v}_i)\mathbf{x}_k + \alpha K(\overline{\mathbf{x}}_k, \mathbf{v}_i)\overline{\mathbf{x}}_k \right)}{\sum_{k=1}^{n} u_{ik}^m \left(K(\mathbf{x}_k, \mathbf{v}_i) + \alpha K(\overline{\mathbf{x}}_k, \mathbf{v}_i) \right)} \tag{21}$$

In the rest of the paper, we name the algorithm using (17) and (18) as KFCM_S, the algorithms using (20) and (21) with mean or median filter as KFCM_S$_1$ and KFCM_S$_2$ respectively. The above algorithms can be summarized in the following unified steps.

Algorithm 2:

Step 1) Fix the number c of these centroids or clusters and then select initial class centroids and set $\varepsilon > 0$ to a very small value.

Step 2) For KFCM_S$_1$ and KFCM_S$_2$ only, compute the mean or median filtered image.

Step 3) Update the partition matrix using Eqs. (17) (KFCM_S) or Eqs. (20)

(KFCM_S$_1$ and KFCM_S$_2$).

Step 4) Update the centroids using Eqs. (18b) (KFCM_S) or Eqs. (21) (KFCM_S$_1$ and KFCM_S$_2$).

Repeat Steps 3)-4) until the following termination criterion is satisfied:

$$\left\|\mathbf{V}_{new} - \mathbf{V}_{old}\right\| < \varepsilon$$

where $\mathbf{V}$ are defined as before.

The optimization flowcharts described in Algorithm 1 and Algorithm 2 are called a fixed-point iteration (FPI) or alternate optimization (AO) as in FCM and [23], the iteration process will terminate to the user-specified number of iteration or local minima of the corresponding objective functions. Consequently optimal or locally optimal results can always be ensured.

IV. Experimental results

In this section, we describe the experimental results on several synthetic and real images. There are totally eight algorithms used in this section, i.e., standard FCM, KFCM, FCM_S, KFCM_S, FCM_S$_1$, KFCM_ S$_1$, FCM_S$_2$ and KFCM_ S$_2$. The kernel used in KFCM _S, KFCM _S$_1$ and KFCM _S$_2$ is the Gaussian RBF kernel. Note that the kernel width σ in Gaussian RBF has a very important effect on performances of the algorithms. However, how to choose an appropriate value for the kernel width in Gaussian RBF is still an 'open problem'. In this paper, we adopt the 'trial-and-error' technique and set the parameter $\sigma = 150$. We also find that for a wide range of σ around 150 (e.g. from 100 to 200), there seem no apparent changes in results. Thus we use this constant value in all experiments throughout the paper. In addition, we set the parameters $m = 2$, $\varepsilon = 0.001$, $N_R = 9$ (a 3x3 window centered around each pixel, used in FCM_S and KFCM_S only) in the rest experiments.

Our first experiment applies these algorithms to a synthetic test image. The image with 64x64 pixels includes two classes with two intensity values taken as 0 and 90. We test the algorithms' performance when corrupted by 'Gaussian' and ' Salt & Pepper' noises respectively and the results are shown in Table 1 and Fig. 1. Table 1 gives the segmentation accuracy (SA) of the eight algorithms on two different noisy images, where SA is defined as the sum of the total number of pixels divided by the sum of number of correctly classified pixels [9]. Fig. 1 shows the result on 'Salt & Pepper' noise corrupted image. Here the parameters $c = 2$, $\alpha = 3.8$. From Table 1 and Fig. 1, KFCM achieves nearly the same result as FCM, and neither can algorithm remove the disturbances of noises due to no spatial information used in both algorithms. On the other hand, the kernel versions with spatial constraints are superior to the corresponding classical algorithms, especially on the 'Salt & Pepper' corrupted image.

We take a set of values for α to test its effect on performance. Fig. 2 and 3 show the comparisons of classification errors of FCM_S, KFCM_S, FCM_S$_1$, KFCM_ S$_1$, FCM_S$_2$ and KFCM_ S$_2$ under different values of α on the synthetic image corrupted by 'Gaussian' and 'Salt & Pepper' noises respectively. From Fig. 2, as α increases, the numbers of misclassified pixels of all six algorithms reduce under 'Gaussian' noises, and there are no apparent changes after $\alpha = 3$. It can also be seen from Fig. 2 that the kernel version algorithms are superior to the corresponding classical algorithms. According to Fig. 3, we know that under 'Salt & Pepper' noises, the kernel version algorithms, i.e. KFCM_S, KFCM_ S_1, and KFCM_ S_2, still achieve much better performance than FCM_S, FCM_ S_1, and FCM_S_2. From Fig. 3, for KFCM_S and KFCM_ S_2, there are no apparent changes after $\alpha = 3$, implying performance to be stable, and most of these algorithms reach minima between $\alpha = 3$ and $\alpha = 4$.

Fig. 4 presents a comparison of segmentation results on a real image [20]. Fig. 4(a) is the original image corrupted by 'Gaussian' and 'Salt & Pepper' noise simultaneously. Fig. 4(b)-(i) show the results applying the eight algorithms on Fig. 4(a) respectively. We use $c = 2$ and $\alpha = 3.8$ in the experiment. In this case, the margins of the segmented image of FCM_S_1 and KFCM_S_1 are a little blurred because of the use of mean filter. It can be easily seen that KFCM_S, KFCM_S_1 and KFCM_S_2 achieve much better segmentation in the presence of hybrid noise, compared to the corresponding FCM_S, FCM_S_1 and FCM_S_2 algorithms.

Fig. 5 and 6 present a comparison of segmentation results of the eight algorithms applied on T1-weighted MR phantom [32]. We use a high-resolution T1-weighted phantom with slice thickness of 1mm, 3% noise and no intensity inhomogeneities. Two slices in the axial plane with the sequence of 121 and 91 are shown in Fig. 5(a) and Fig. 6(a) respectively. The segmentation results on two slices using the eight methods with eight classes are shown in Fig. 5(b)-(i) and 6(b)-(i) respectively. And Table 2 gives the quantitative comparison scores corresponding to Fig. 5(a) and 6(a) using eight methods for gray matter, white matter and cerebrospinal fluid, where the comparison scores are calculated as

$$s_{ij} = \frac{A_{ij} \cap A_{refj}}{A_{ij} \cup A_{refj}}$$

where A_{ij} represents the set of pixels belonging to the jth class found by the ith algorithm and A_{refj} represents the set of pixels belonging to the jth class in the reference segmented image. In the experiment, we use the parameter α =0.8 for all algorithms with spatial constraints. From Fig. 5 and 6 and Table 2, the kernel version algorithms outperform the corresponding classical algorithms.

To test the performances of the eight algorithms under other levels of noises on the simulated brain database [32], we do the following comparison experiments. Fig. 7 and 8 show the relationship between the comparison scores of the eight algorithms and noise levels on two image slices in the axial plane with the sequence of 121 and 91 respectively. Obviously, as the level of noises increases, performances of most algorithms gradually degrade, but the kernel version algorithms still surpass the corresponding classical ones.

Fig. 9 and 10 show the comparison of segmentation results on real T1-weighted MR images with artificially added 'Gaussian' and 'Salt & Pepper' noises respectively. Note that MR images typically do not suffer from 'Salt & Pepper' noise, and we add such type of noise just for the comparison of robustness to noises of different algorithms. We do not consider KFCM algorithm in this experiment, because without spatial constraints, KFCM has the similar performance as that of FCM, as shown in Fig. 1 and 2. Fig. 9(b)-(i) and 10(b)-(i) show segmentation results from applying the seven algorithms on Fig. 9(a) and Fig. 10(a) respectively. Here the parameters $c = 3$,

$\alpha = 3.8$ are used in the experiment. From Fig. 9 and 10, FCM_S and KFCM_S have bad performance in the presence of 'Gaussian' noise, while FCM_S_1, FCM_S_2, KFCM_S_1 and KFCM_S_2 achieve satisfactory results and the latter two are a little superior to the former two algorithms. On the other hand, when 'Salt & Pepper' noise is added, only KFCM_S and KFCM_S_2 work well, while the other algorithms fail to remove the effect of added noises. On the whole, KFCM_S_2 algorithm achieves better segmentation results under both noises.

Finally, Table 3 gives the comparison of the running time on two MR images (see Fig. 5(a) and 6(a)) using FCM_S, KFCM_S, FCM_S_1, KFCM_S_1, FCM_S_2 and KFCM_S_2. We artificially add 'Gaussian' and 'Salt & Pepper' noises on both MR images and perform experiments on an IBM computer with 1.7 GHz Pentium processor using MATLAB (Mathworks, Inc). From Table 3, our proposed FCM_S_1 and FCM_S_2 are much faster than the original FCM_S (typically 3-10 times faster), while their corresponding kernelized versions (KFCM_S_1 and KFCM_S_2) need no less run-time than KFCM_S in most cases, but still less than original FCM_S algorithm.

V. Conclusions

The well-known 'kernel methods' has been recently applied to unsupervised clustering [18], [21]-[22]. However, an unfavorable point of these existed kernel clustering algorithms is that the clustering prototypes lie in high dimensional feature space and hence clustering results are not easily interpreted as intuitively as in the original space. In this paper, based on one of our early works [22], we use a group of novel fuzzy clustering algorithms based on a family of kernel-induced distance measures for image segmentation. We discussed three types of spatial constraint on the objective function of original FCM to effectively segment images corrupted by noise and outliers. Synthetic and real images were used to compare the performances of these algorithms including segmentation evaluation and run-time requirements. On the whole, KFCM algorithms with spatial constraints (SC) have more robustness to noise and outliers than their counterparts without SC. In our opinions, we suggest using KFCM_S$_2$ algorithm in practice considering the tradeoff between robustness to different noises and execution speed.

The results reported in this paper show that the kernel method is an effective approach to constructing a robust image clustering algorithm. This method can also be used to improve the performance of other FCM-like algorithms based on adding some type of penalty terms to the original FCM objective function. Our further and ongoing works include clustering validity in our algorithms, adaptive determination for the clustering number and other applications, *e.g.*, gain field estimation.

Acknowledgements

We thank the anonymous reviewers' constructive comments to improve presentation of our paper. And also thank National Science Foundations of China and of Jiangsu under Grant Nos. 60271017 and BK2002092 , "QingLan" project and the Returnee's foundations of Jiangsu and of CSC for partial supports respectively.

References

[1] J.C. Bezdek, L.O. Hall, and L.P. Clarke, "Review of MR image segmentation techniques using pattern recognition," *Medical Physics*, vol.20, pp.1033-1048, 1993.

[2] D.L. Pham, C.Y. Xu, J.L. Prince, "A survey of current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, vol. 2, pp. 315-337, 2000.

[3] W.M. Wells, W.E. LGrimson, R. Kikinis and S.R. Arrdrige, "Adative segmentation of MRI data," *IEEE Trans. Medical Imaging*, vol. 15, pp.429-442, 1996.

[4] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.

[5] D.L. Pham, and J.L. Prince, "An adaptive fuzzy C-means algorithm for image segmentation in the presence of intensity inhomogeneities," *Pattern Recognition Letters,* vol. 20, pp.57-68, 1999.

[6] Y.A. Tolias and S.M. Panas, "On applying spatial constraints in fuzzy image clustering using a fuzzy rule-based system," *IEEE Signal Processing Letters*, vol. 5, pp.245-247, 1998.

[7] Y.A Tolias and S.M. Panas, "Image segmentation by a fuzzy clustering algorithm using adaptive spatially constrained membership fucntions," *IEEE Trans. on Systems, Man, and Cybernetics*-Part A, vol. 28, pp.359-369, 1998.

[8] A.W.C. Liew, S.H. Leung and W.H. Lau, "Fuzzy image clustering incorporating spatial continuity" *IEE Proc.-Vis. Image Signal Process.*, vol. 147, pp. 185-192,

2000.

[9] M.N. Ahmed, S.M. Yamany, N. Mohamed, A.A. Farag and T. Moriarty, "A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data," *IEEE Trans. on Medical Imaging*, vol. 21, pp.193-199, 2002.

[10] D.L. Pham, "Fuzzy clustering with spatial constraints," in *Proc. IEEE Intern. Conf. on Image Processing*, New York, USA, August, 2002.

[11] K.R. Muller, S. Mika, et al, "An Introduction to Kernel-based Learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, pp.181-202, 2001.

[12] N. Cristianini and J.S. Taylor, *An Introduction to SVMs and other kernel-based learning methods*. Cambridge Univ. Press, 2000.

[13] V.N. Vapnik, *Statistical learning theory*. New York: Wiley, 1998.

[14] B. Scholkopf, *Support vector learning*. R. Oldenbourg Verlay, 1997.

[15] V. Roth and V. Steinhage, "Nonlinear discriminant analysis using kernel functions," in *Advances in Neural Information Processing Systems* 12, S.A Solla, T.K. Leen, and K.-R. Muller, Eds., MIT Press, 2000, pp.568-574.

[16] B. Scholkopf, A.J. Smola, K.R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299-1319, 1998.

[17] T.M. Cover, "Geomeasureal and statistical properties of systems of linear inequalities in pattern recognition," *IEEE Trans. Electron. Comput*. Vol. EC-14, pp. 326-334, Mar. 1965.

[18] D.Q Zhang and S.C. Chen, "Kernel based fuzzy and possibilistic c-means clustering," In *Proc. Intern. Conf. on Artificial Neural Network* (ICANN '03),

Istanbul, Turkey, June, 2003, pp.122-125.

[19] P.J. Huber, *Robust statistics*. New York: Wiley, 1981.

[20] The Mathworks Inc., *Image processing toolbox*, available at htttp://www.mathworks.com.

[21] M. Girolami, "Mercer kernel-based clustering in feature space," *IEEE Trans. Neual Networks*, vol. 13, pp. 780-784, 2002.

[22] D.Q. Zhang and S.C. Chen, "Fuzzy clustering using kernel methods," in *Proc. Intern. Conf. on Control and Automation*, Xiamen, China, June, 2002, pp. 123-127.

[23] K.L Wu and M.S Yang, "Alternative c-means clustering algorithms," *Pattern Recognition*, vol. 35, pp. 2267-2278, 2002.

[24] J.H. Chen and C.S Chen, "Fuzzy kernel perceptron," *IEEE Trans. Neural Nets.*, vol.13, pp.1364-1373, 2002.

[25] J. Leski, “Towards a robust fuzzy clustering,” *Fuzzy Sets and Systems*, vol.137, no. 2, pp.215-233 2, July, 2003.

[26] J.K. Udupa and S. Samarasekera, “Fuzzy connectedness and object definition: Theory, algorithm and applications in image segmentation,” *Graphical Models Image Processing*, vol.58, no.3, pp.246-261, 1996.

[27] S.M. Yamany, A.A. Farag, and S. Hsu, “A fuzzy hyperspectral classifier for automatic target recognition (ATR) systems,” *Pattern Recognition Lett.*, vol.20, pp.1431-1438, 1999.

[28] R.J. Hathaway, J.C. Bezdek, “Generalized fuzzy c-means clustering strategies

using L$_p$ norm distance," *IEEE Trans. Fuzzy Systems*, vol.8, no.5, pp.576-572, 2000.

[29] K. Jajuga, "L$_1$ norm based fuzzy clustering," *Fuzzy Sets and Systems*, vol.39, no.1, pp.43-50, 1991.

[30] J. Leski, "An ε-insensitive approach to fuzzy clustering," *Intern. J. Appl. Math. Comp. Sci.*, vol.11, no.4, pp.993-1007, 2001.

[31] D.Q. Zhang, S.C. Chen, "Clustering incomplete data using kernel-based fuzzy c-means algorithm," *Neural Processing Lett.,* vol.18, no.3, pp.155-162, 2003.

[32] R.K.S. Kwan, A.C. Evans, G.B. Pike, "An Extensible MRI Simulator for Post-Processing Evaluation," *Visualization in Biomedical Computing (VBC'96). Lecture Notes in Computer Science*, vol. 1131, pp.135-140, Springer-Verlag, 1996.

[33] A.W.C. Liew, H. Yan, "An adaptive spatial fuzzy clustering algorithm for 3-D MR image segmentation," *IEEE Trans. Medical Imaging*, vol.22, no.9, pp. 1063-1075, 2003.

Figure Captions:

Fig. 1 Comparison of segmentation results on synthetic test image. (a) Original image with 'Salt & Pepper' noise, (b) FCM result, (c) KFCM result, (d) FCM_S result, (e) KFCM_S result, (f) FCM_S$_1$ result, (g) KFCM_S$_1$ result, (h) FCM_S$_2$ result, (i) KFCM_S$_2$ result.

Fig. 2 Comparison of classification errors on synthetic image with 'Gaussian' noise under different values of alpha.

Fig. 3 Comparison of classification errors on synthetic image with 'Salt & Pepper' noise under different values of alpha

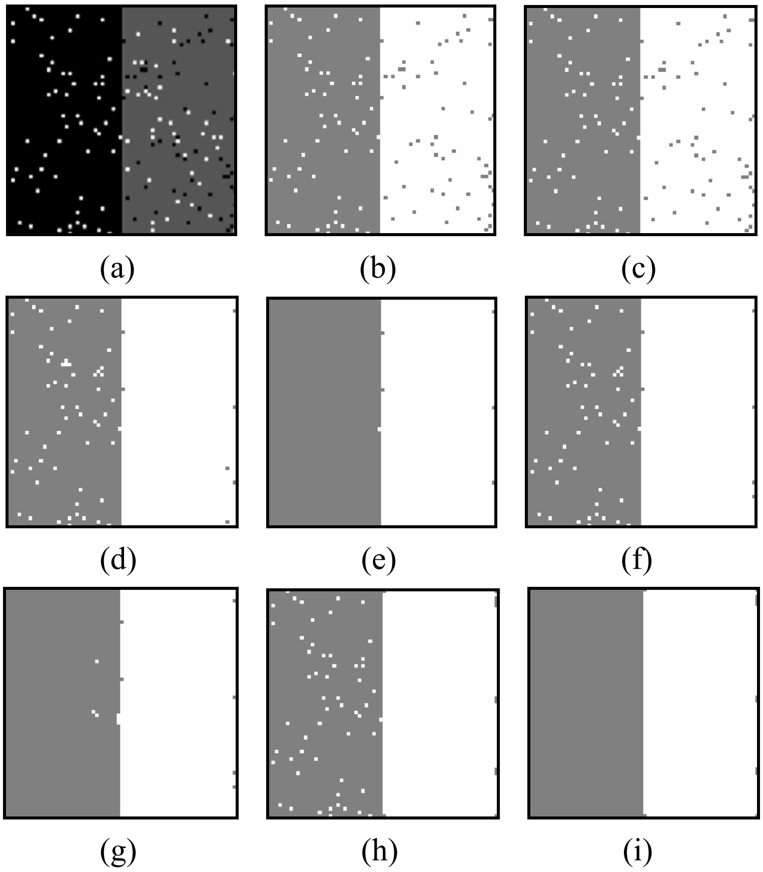
Fig. 4 Comparison of segmentation results on a real image corrupted by hybrid 'Gaussian' and 'Salt & Pepper' noise. (a) The original noisy image, (b) FCM result, (c) KFCM result, (d) FCM_S result, (e) KFCM_S result, (f) FCM_S$_1$ result, (g) KFCM_S$_1$ result, (h) FCM_S$_2$ result, (i) KFCM_S$_2$ result.

Fig. 5 Comparison of segmentation results on a simulated brain MR image. (a) Original T1-weighted image, (b) using FCM, (c) using KFCM, (d) using FCM_S, (e) using KFCM_S, (f) using FCM_S$_1$, (g) using KFCM_S$_1$, (h) using FCM_S$_2$, (i) using KFCM_S$_2$.

Fig. 6 Another simulated brain MR image example. (a) Original T1-weighted image, (b) using FCM, (c) using KFCM, (d) using FCM_S, (e) using KFCM_S, (f) using FCM_S$_1$, (g) using KFCM_S$_1$, (h) using FCM_S$_2$, (i) using KFCM_S$_2$.

List of Tables

Fig. 1



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 2

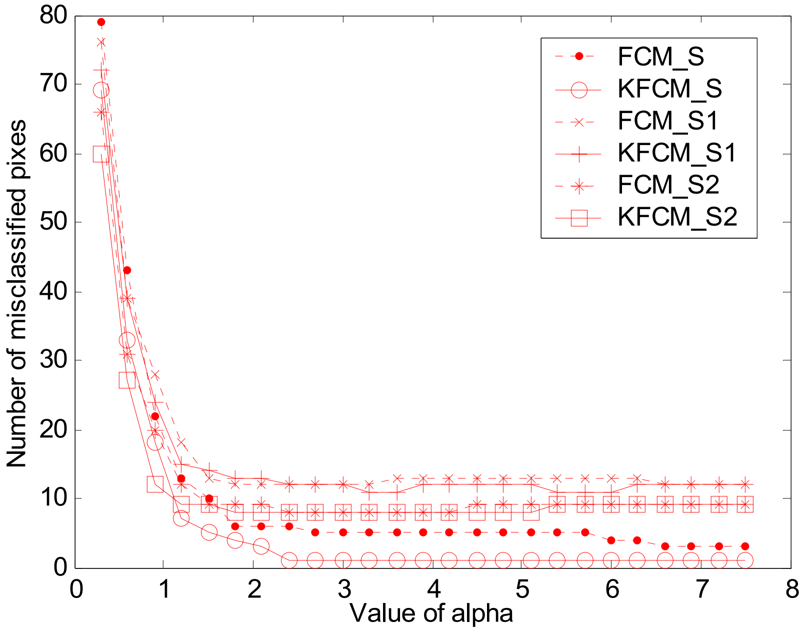

FCM_S
KFCM_S
FCM_S1
KFCM_S1
FCM_S2
KFCM_S2
Number of misclassified pixes
Value of alpha
0
10
20
30
40
50
60
70
80
0
1
2
3
4
5
6
7
8

Fig. 3

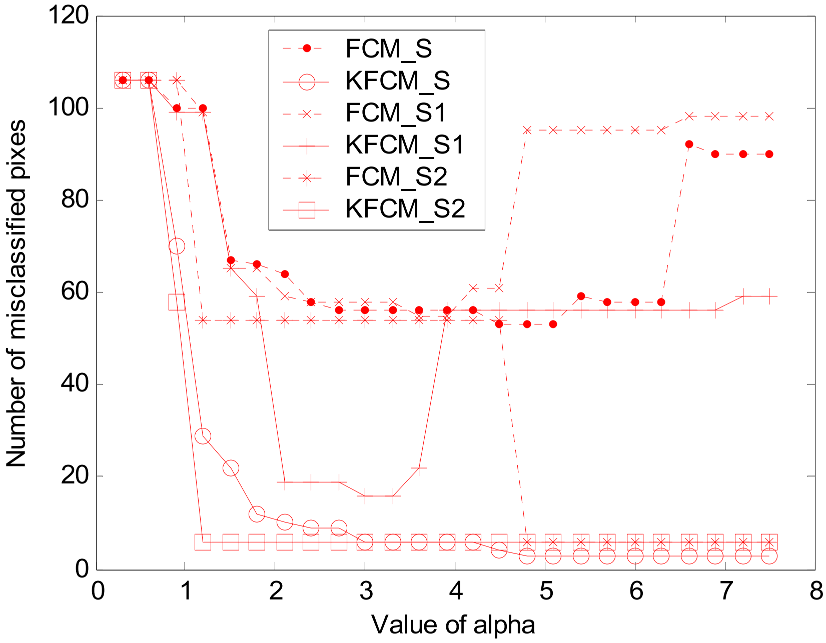
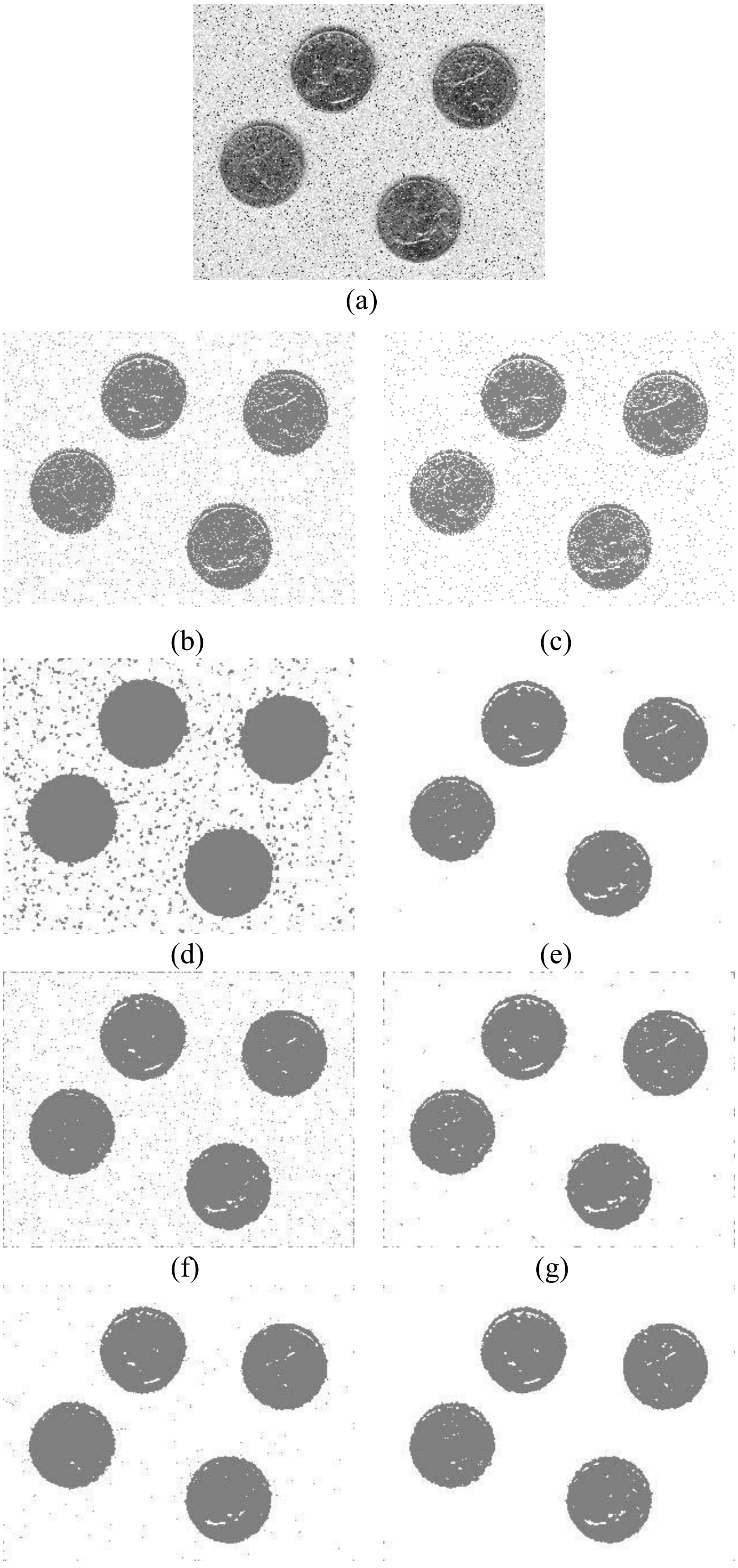

FCM_S
KFCM_S
FCM_S1
KFCM_S1
FCM_S2
KFCM_S2
Number of misclassified pixes
Value of alpha
0
20
40
60
80
100
120
0
1
2
3
4
5
6
7
8

Fig. 4



(a)

(b)

(c)

(d)

(e)

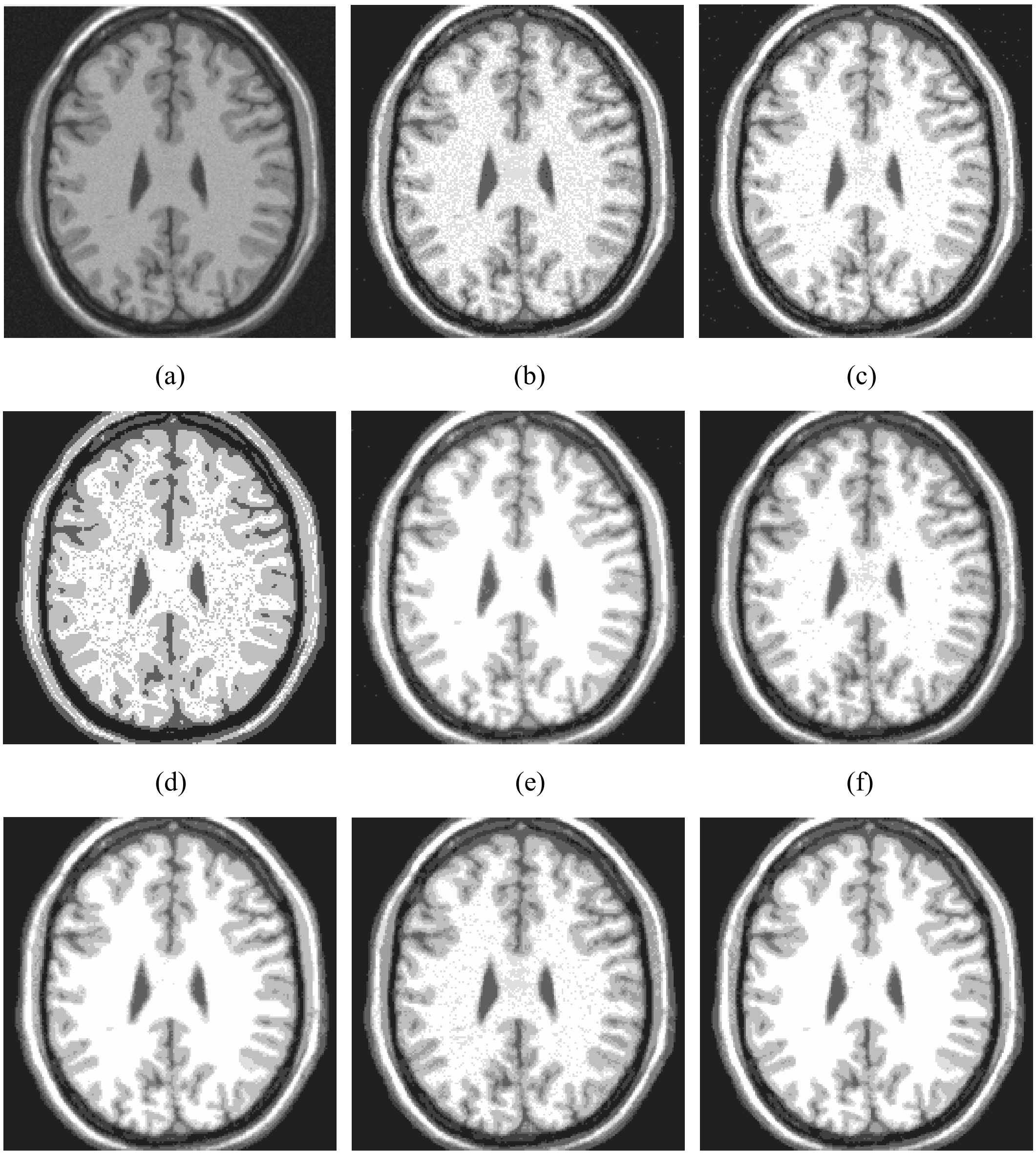
(f)

(g)

(h)

(i)

Fig. 5

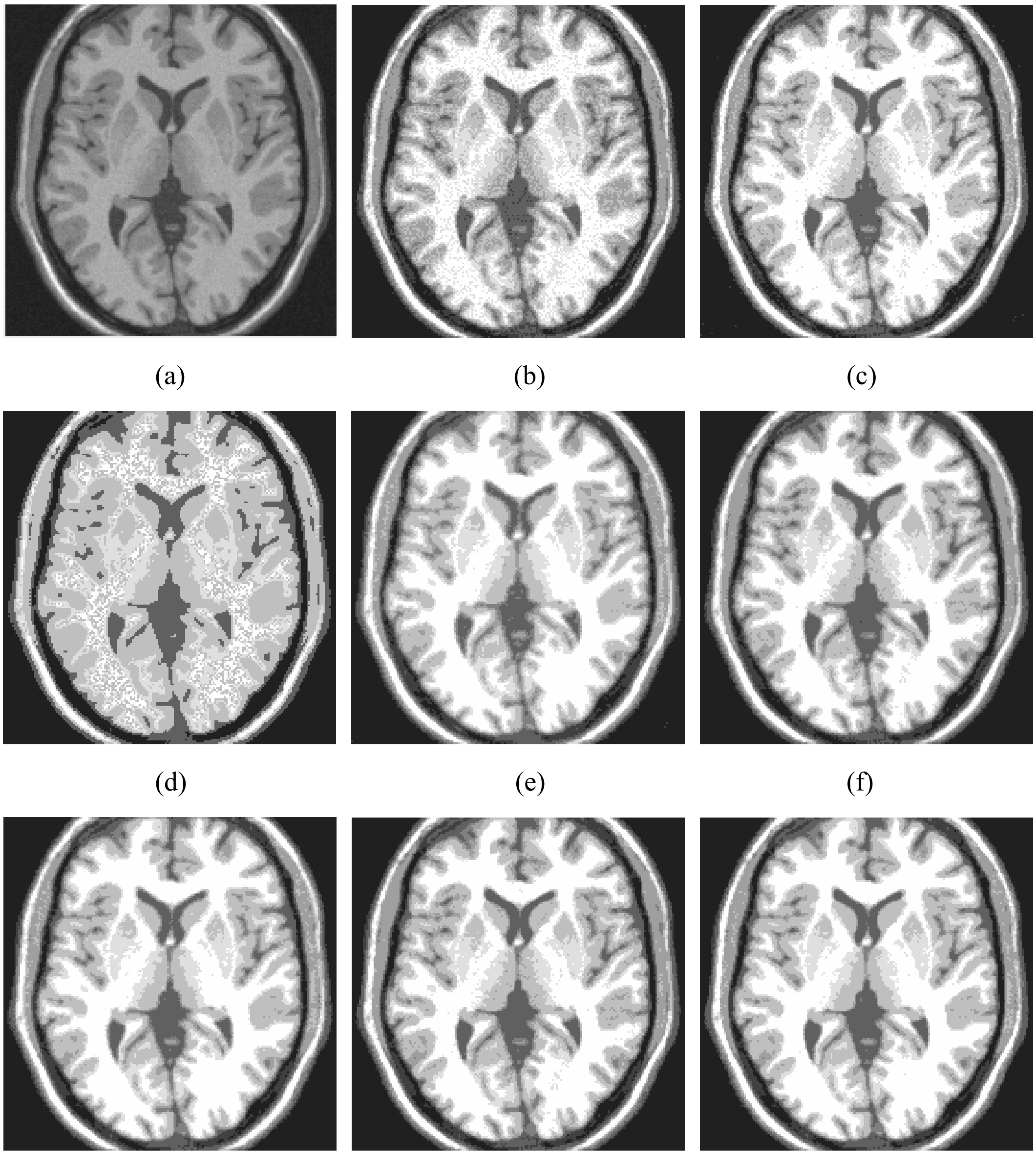


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 6

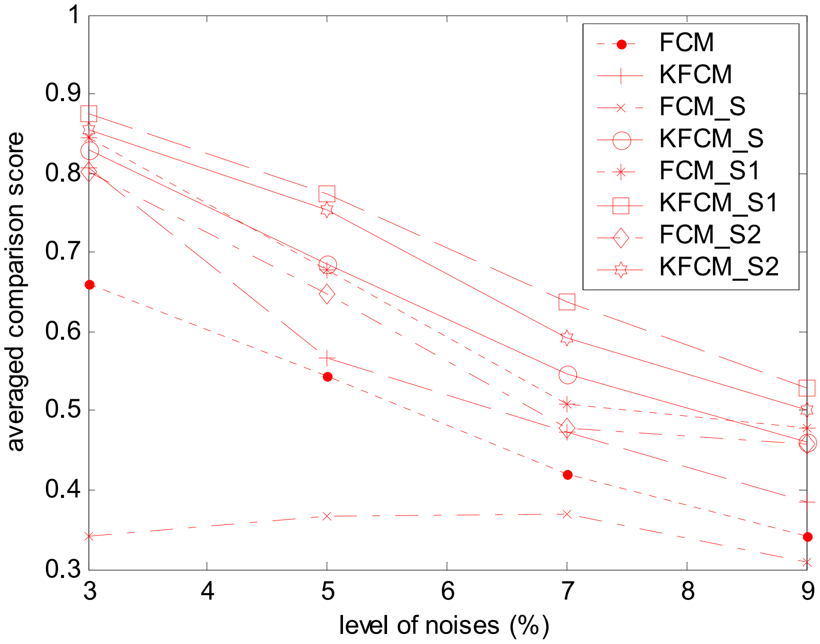


(a) (b) (c)

(d) (e) (f)

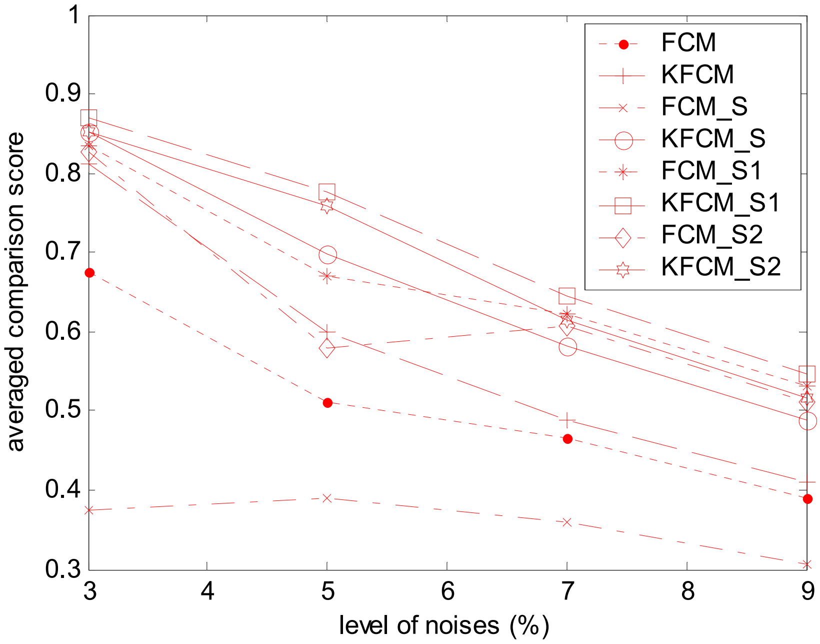
(g) (h) (i)

Fig. 7



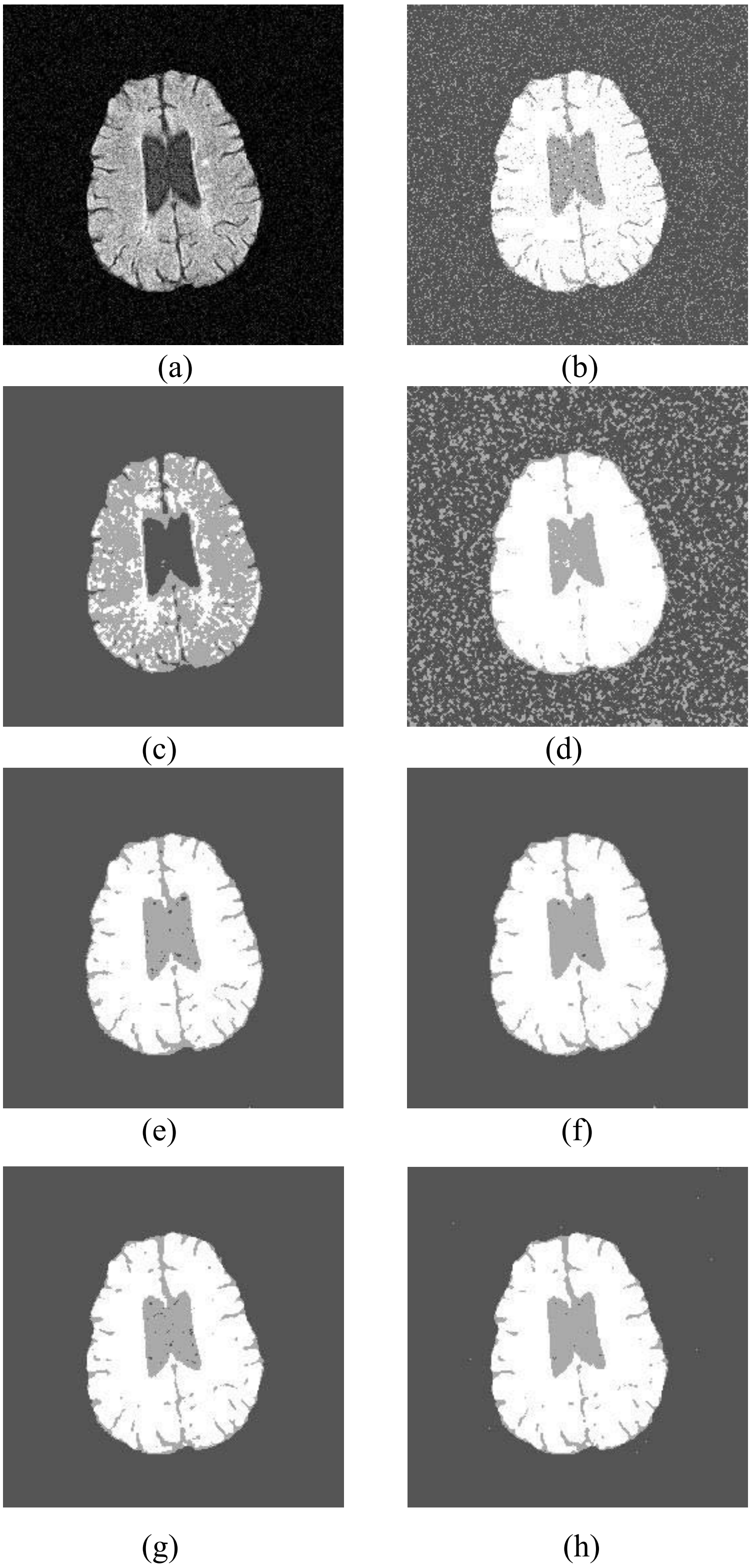
FCM
KFCM
FCM_S
KFCM_S
FCM_S1
KFCM_S1
FCM_S2
KFCM_S2
averaged comparison score
level of noises (%)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
3
4
5
6
7
8
9

Fig. 8



FCM
KFCM
FCM_S
KFCM_S
FCM_S1
KFCM_S1
FCM_S2
KFCM_S2
averaged comparison score
level of noises (%)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
3
4
5
6
7
8
9

Fig. 9



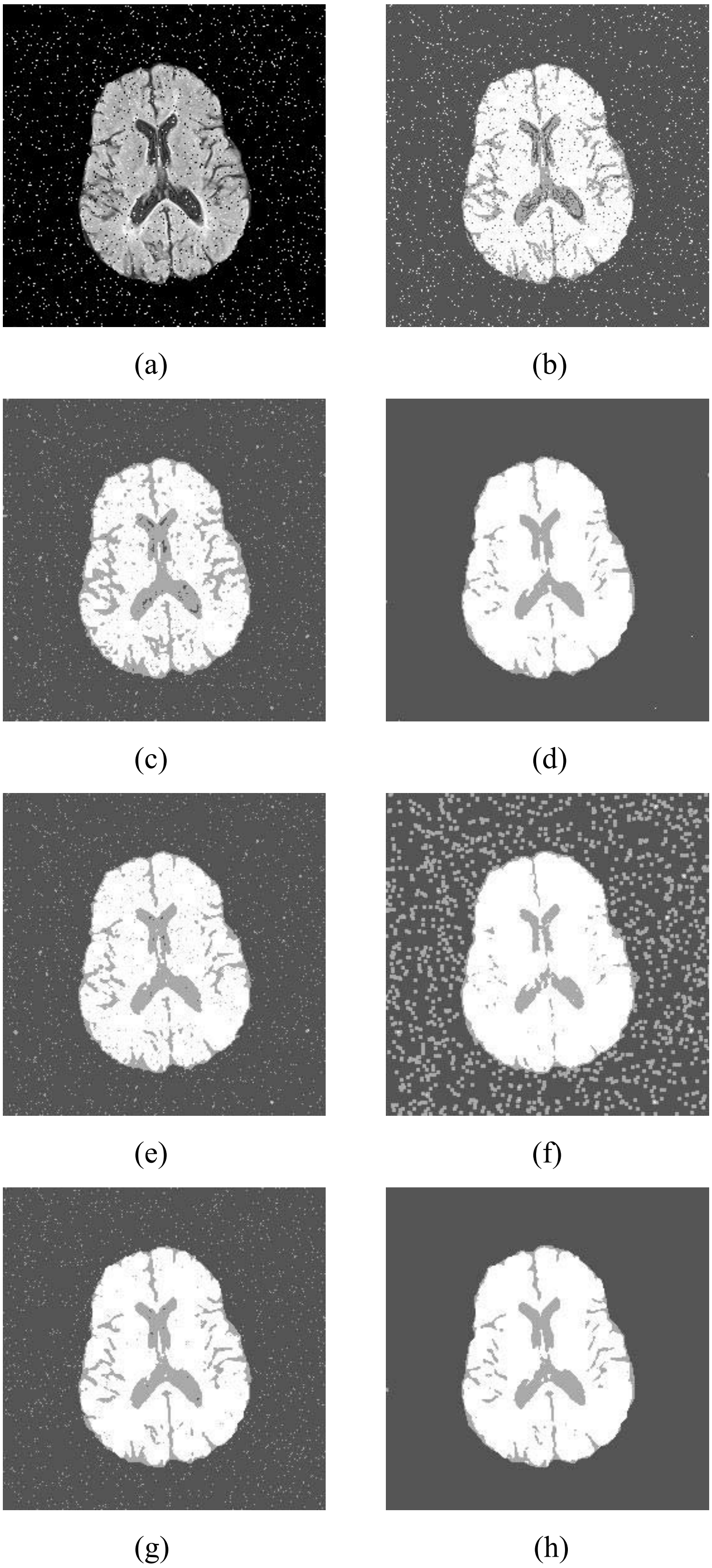
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 10



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Table 1 Segmentation accuracy (SA %) of eight methods on synthetic image

	FCM	KFCM	FCM _S	KFCM _S	FCM _S_1	KFCM _S_1	FCM _S_2	KFCM _S_2
Salt& Pepper	96.92	96.92	98.19	99.83	98.28	99.67	98.17	99.69
Gaussian	93.97	94.58	99.83	99.89	99.72	99.81	99.86	99.86

Table 2 Comparison scores of eight methods for Fig. 3(a) and 4(a)

	Fig. 3(a)			Fig. 4(a)		
FCM	0.9229	0.5412	0.5170	0.9088	0.4606	0.6576
KFCM	0.9317	0.6485	0.8422	0.9301	0.6293	0.8780
FCM_S	0.7944	0.0076	0.2254	0.7943	0.0102	0.3177
KFCM_S	0.9314	0.6592	0.8966	0.9350	0.7016	0.9173
FCM_S_1	0.9526	0.6965	0.8828	0.9492	0.6790	0.8732
KFCM_S_1	0.9557	0.7490	0.9190	0.9532	0.7380	0.9155
FCM_S_2	0.9437	0.6612	0.8001	0.9391	0.6552	0.8861
KFCM_S_2	0.9494	0.7084	0.9026	0.9443	0.6911	0.9198

Table 3 Comparisons of running time of six algorithms on two MR images (x 10^3 seconds)

		FCM_S	KFCM_S	FCM_S_1	KFCM_S_1	FCM_S_2	KFCM_S_2
Slice 1	Gaussian	1.6729	1.2056	0.2673	1.2458	0.2135	1.3797
	Salt & Pepper	1.5637	2.1205	0.5722	1.7527	0.5870	1.3920
Slice 2	Gaussian	2.4367	0.7930	0.3488	2.3074	0.2275	1.0275
	Salt & Pepper	3.4214	0.7027	0.3554	1.5369	0.3036	1.2005