2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Color Image Segmentation Based on Decision-Theoretic Rough Set Model and Fuzzy C-Means Algorithm

Min Guo and Lin Shang

***Abstract*— This paper proposes an approach which combines the Decision Theoretic Rough Set model (DTRS) and Fuzzy C-Means(FCM) algorithm to perform color image segmentation. The FCM algorithm has the limitation that it requires the initialization of cluster centroids and the number of clusters. In this paper, the DTRS model is applied to color image segmentation for the purpose of clustering validity analysis which could overcome the defect of the FCM algorithm. Firstly, we adopt the Turbopixel algorithm to split the color image into many small regions called superpixels for presegmentation. Based on color image color histogram feature extraction we use Bhattacharyya coefficient to measure the similarity between superpixels, which is in preparation for clustering validity analysis. It is our focus that we will obtain cluster centroids and the number of clusters using FCM. Our approach is according to the hierarchical clustering validity analysis algorithm using DTRS model. Finally, the FCM algorithm is utilized to achieve the result of color image segmentation. Experimental results show that the DTRS-based preprocessing approach can obtain better segmentation results than other improved FCM approaches such as ant colony algorithm or histogram thresholding approach.**

I. INTRODUCTION

As an important branch of data mining, clustering has been widely used in many fields, such as pattern recognition, image processing, medical diagnosis etc.. Clustering is the process of dividing a set of unlabeled objects into multiple clusters such that objects in the same cluster are more similar to each other than those in different clusters. The clustering strategies can be split into two broad types: the hard clustering and the soft clustering. The hard clustering requires each object to belong to just one certain cluster. However, in fuzzy clustering, the fuzzy set theory [1] is added into the traditional clustering methods and objects can belong to more than one cluster with different membership degrees, which is more in accordance with real situations because there are always no clear boundaries between objects.

Fuzzy C-Means (FCM), as the most popular algorithm of the soft clustering, has been extensively used to make compact and well separated clusters. It is based on the minimization of an objective function and iterates through the update of membership degrees and cluster centroids. One of the most frequently used aspects with FCM is image segmentation [2], [3], [4] because it can retain more image information than the hard clustering algorithm. Although the FCM algorithm works better than the hard clustering, there still exist unavoidable defects that cluster centroids and the number of clusters should be decided in advance. A good initialization could bring on ideal segmentation results. However, an unsuitable initialization may result in poor quality of image segmentation. In view of the great effect of these two parameters on clustering, what we mainly focus on in this paper is how to decide the cluster centroids and the number of clusters for better clustering with FCM applied in color image segmentation.

On the whole, a good Cluster Validity Index (CVI) is useful to determine cluster centroids and the number of clusters [5] and the Decision Theoretic Rough Set (DTRS) model can construct a good CVI. The DTRS model [6], [7], [8] was proposed by Yao et al. in the early 1990s, which has brought new insights into the probabilistic approaches to rough sets [9] and offered a better understanding of classification. It could achieve the minimal risk costs by introducing the Bayesian theory. Recently, the DTRS model has been widely used in various fields such as information filtering [10], attribute reduction [11], investment decision-making [12], spam email filtering [13]. In terms of clustering, Lingras et al. describe how to develop a cluster validity index from the DTRS model by adjusting loss functions in [14]. Yu et al. propose a new hierarchical method through extending the DTRS model for clustering validity analysis in [5]. These proposed measures are just applied to the synthetic data set or information tables on the UCI database [15]. For being applied in the real world, we propose a DTRS-based clustering validity analysis method according to the algorithm in [5] for the preprocessing step when using FCM in color image segmentation.

Recently, some preprocessing techniques have been adopted to conduct clustering validity analysis for overcoming the drawbacks of FCM in color image segmentation. In [16], the ant colony-Fuzzy C-Means hybrid algorithm (AFHA) is introduced, which used the Ant System (AS) algorithm for intelligent initialization. To increase the computational efficiency of AFHA, the improved ant colony-Fuzzy C-Means hybrid algorithm (IAFHA) in [16] is proposed for further elevation. However, algorithms in [16] do not seem to be satisfactory due to redundant parameters. In [17], a novel histogram Fuzzy C-Means hybrid (HTFCM) approach was presented by applying the histogram thresholding technique to obtain all possible uniform regions in color images. In [18], Khan et al. employed the Self Organizing Map (SOM) to automatically determine the ideal number of clusters. Both involve the fussy process of histogram curve smoothing.

Our work starts from these mentioned problems like

Min Guo and Lin Shang are from State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China (email: guomin-nj@gmail.com, shanglin@nju.edu.cn).

This work was supported by the National Natural Science Foundation of China (Nos. 61170180, 61035003).

redundant parameters and complicated process of clustering validity analysis. We only introduce two parameters to implement our algorithm. One is in our preprocessing step, which decides the number of superpixels we should segment the color image into and another is used to merge the small size cluster into its closest cluster of large size in the postprocessing step, which is in case of over-segmentation. These two parameters will control our segmentation result which can be more convenient in terms of parameter tuning compared with the algorithm in [3]. Also, our approach for clustering validity analysis is a merging process based on the extended DTRS model. The process is a simple gradual way from bottom to up without fussy process of histogram curve smoothing based on several fuzzy rules.

The general process of our approach can be summarized as follows. Thinking of the computational complexity, we presegment a color image into a set of compact regions called superpixels at first. The algorithm we adopt in this paper is Turbopixel Algorithm presented in [19]. Then, we build the similarity measure between superpixels. Next, we apply the extended DTRS model mentioned in [5] for clustering validity analysis. In case of over-segmentation, additional measure is taken for improvement based on the algorithm in [5]. Finally, we use FCM for color image segmentation. We also employ some indices for comparison.

The rest of this paper is organized as follows. The backgrounds are given in Section II, including the FCM algorithm and the extended DTRS model. Section III presents our new proposed DTRS-based preprocessing method to determine cluster centroids and the number of clusters before carrying out FCM for color image segmentation. Experiments and analysis are provided in Section IV.. Finally, conclusion is made in the last part.

II. Backgrounds

A. The Fuzzy C-Means Algorithm

Fuzzy clustering is a class of algorithms for clustering which allows data items to belong to more than one cluster with different membership degrees between 0 and 1. Fuzzy C-Means (FCM) is a major technique of fuzzy clustering and it was first introduced by Dunn [20] and modified by Bezdek [21]. It is an iterative method based on optimization of the weighted squared error function J_m [21]

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ji}^m \parallel x_i - c_j \parallel^2 \tag{1}$$

where N is the number of data items, c is the number of clusters with $2 \leq c < N$, m is a fuzzy factor with $m \geq 1$, x_i is the i^{th} data item in the d-dimensional vector space, c_j is the centroid of the j^{th} cluster, u_{ji} is the membership degree of x_i in the j^{th} cluster, $\parallel * \parallel$ is a distance measure between data item x_i and cluster centroid c_j. The detailed FCM algorithm can be described as follows:

Algorithm 1 Fuzzy C-Means

1: Set values for iteration terminating threshold ε, cluster number c and fuzzy factor m.
2: Set the loop counter $q = 0$.
3: Randomly initialize cluster centroid matrix $C^{(q)}$ with $c_k(k = 1, 2, ..., c)$.
4: Calculate the membership matrix $U^{(q)}$ according to $C^{(q)}$ with the following equation:

$$u_{ji} = \frac{1}{\sum_{k=1}^{c} (\frac{d_{ji}}{d_{ki}})^{2/m-1}}. \tag{2}$$

Notice, if $d_{ji} = 0$, then set $u_{ji} = 1$ and $u_{li} = 0(l = 1, ..., N \ and \ l \neq j)$.
5: Calculate $C^{(q+1)}$ according to $U^{(q)}$ with the following equation:

$$c_j = \frac{\sum_{i=1}^{N} u_{ji}^m x_i}{\sum_{i=1}^{N} u_{ji}^m}. \tag{3}$$

6: Update $U^{(q+1)}$ according to $C^{(q+1)}$ with Eq. (2).
7: If $max\{U^{(q+1)} - U^{(q)}\} \leq \varepsilon$, then stop, otherwise, set $q = q + 1$ and go to step 4.

B. The Extended Decision-Theoretical Rough Set Model(DTRS)

Traditional rough set theory is proposed by Pawlak [9] in 1982 and it uses upper and lower approximations to describe a rough set. However, the Pawlak rough set model does not consider fault tolerance of decision rules. Thus, Yao et al. proposed the decision-theoretical rough set model which makes use of bayesian risk decision minimization theory, introducing two threshold values α and β [6], [8]. In terms of clustering based on DTRS, Yu et al. in [5] extend some definitions about the DTRS model.

Let $\Omega = \{C, \neg C\}$ be the set of two complementary states, where C and $\neg C$ respectively represents the state in which two objects x_i and x_j belong to the same cluster and belong to different clusters, and $A = \{a_P, a_N\}$ be the set of two actions, where a_P and a_N respectively represents the action in which two objects x_i and x_j are allocated to the same cluster and allocated to different clusters, and $P(C|(x_i, x_j))$ represents the probability of allocating this pair of objects to the state C, and $\lambda_{a_j w_i}$ be the risk cost after making decision as a_j when in state w_i.

Obviously, $P(C|(x_i, x_j))$ should be in proportion to the similarity between object x_i and x_j, denoted as $sim(x_i, x_j)$. For constructing the relationship between $P(C|(x_i, x_j))$ and $sim(x_i, x_j)$, the threshold val is introduced. Namely, if $sim(x_i, x_j) = val$, $P(C|(x_i, x_j)) = 0.5$. In this paper, we choose the following equation to solve the value of val:

$$val = \frac{1}{N^2} \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} sim(x_i, x_j) \tag{4}$$

Then $P(C|(x_i, x_j))$ could be calculated as follows:

$$P(C|(x_i,x_j)) = \begin{cases} 0.5 + \frac{sim(x_i,x_j)}{2-2val} & sim(x_i,x_j) \geq val \\ 0.5 - \frac{val-sim(x_i,x_j)}{2val} & sim(x_i,x_j) < val. \end{cases} \quad (5)$$

$$P(\neg C|(x_i,x_j)) = 1 - P(C|(x_i,x_j)). \quad (6)$$

Next, the expected losses can be achieved according to the following equations:

$$\begin{aligned} R(a_P|(x_i,x_j)) &= \lambda_{PP} P(C|(x_i,x_j)) + \lambda_{PN} P(\neg C|(x_i,x_j)) \\ R(a_N|(x_i,x_j)) &= \lambda_{NP} P(C|(x_i,x_j)) + \lambda_{NN} P(\neg C|(x_i,x_j)) \end{aligned} \quad (7)$$

Here, we just consider a particular kind of cost functions, namely the cost is 0 if the two objects are divided into the same cluster, and the cost is 1 if they are divided into different clusters and vice versa. It can be expressed as:

$$\begin{aligned} \lambda_{PP} = 0, \lambda_{PN} = 1 \\ \lambda_{NP} = 1, \lambda_{NN} = 0 \end{aligned} \quad (8)$$

Eq. (7) can be expressed as:

$$\begin{aligned} R(a_P|(x_i,x_j)) = P(\neg C|(x_i,x_j)) \\ R(a_N|(x_i,x_j)) = P(C|(x_i,x_j)) \end{aligned} \quad (9)$$

So, for any pair of objects (x_i, x_j), the risk cost under the state of clustering result CS_t is

$$R(CS_t|(x_i,x_j)) = \begin{cases} P(\neg C|(x_i,x_j)) & (x_i,x_j) \in C \\ P(C|(x_i,x_j)) & (x_i,x_j) \in \neg C \end{cases} \quad (10)$$

where we can calculate P according to Eq. (5) and (6). Finally, the evaluate function we perform cluster validity analysis is defined as:

$$R(CS_t) = \sum_{i=1}^{n} \sum_{j=1}^{n} R(CS_t|(x_i,x_j)) \quad (11)$$

According to Eq. (11), we can calculate the cost of a clustering result CS_t.

III. Our Method for Color Image Segmentation

In this section, we present our method to segment the color images based on the extended DTRS model and FCM.

A. Descriptions of Subtle Preparations

We will present some specific descriptions about our work including the pre-processing, feature extraction and similarity measure. As a matter of fact, in the process of deciding the cluster number and cluster centroids, we will calculate the similarity matrix between pixels, which will lead to high computational complexity and be time consuming. To solve this problem, we presegment the image into superpixels according to the method in [19]. Then these superpixels become the smallest units we will discuss in the next parts. A superpixel after presegmentation is considered as a set of pixels that have the same color. Therefore, features we extract from images are color histograms converted from RGB color space into Hue-Saturation-Value (HSV) color space, which will comply better with the similar judgment of the image color. Here, we discrete each dimension of HSV into 8 bins, thus there are totally 512 bins in the color histogram. To calculate the similarity matrix of superpixels, we utilize Bhattacharyya Coefficient which is an absolute similarity measure when using color histograms. After brief introduction, the overall process of our proposed method can be seen in Fig. 1..

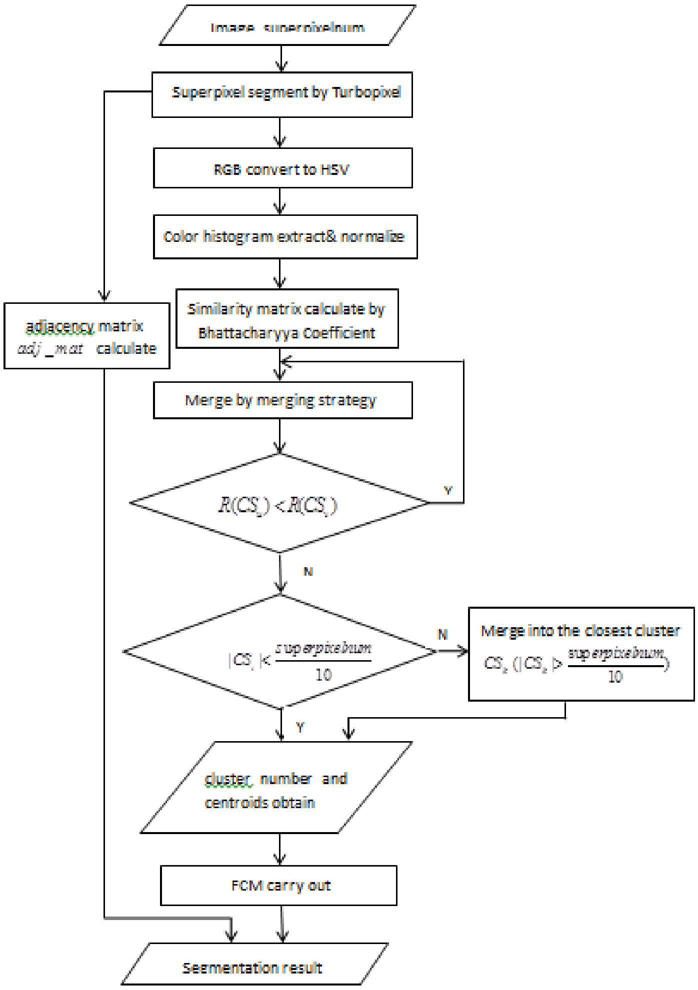

Fig. 1. Overall process of color image segmentation

B. Overall Process of Our Proposed Method

As shown in Fig. 1., our input are an image and the number of superpixels that we set. The Turbopixel algorithm is carried out to segment the image into a set of small compact regions, namely superpixels. Next, as described in Section A., we extract the color histogram feature by HSV color space and conduct normalization. After these work, we will adopt the extend DTRS model [5] to perform clustering validity analysis mainly in the next three steps. At the initialization step, each superpixel is set to be a separate cluster and we can obtain the risk cost $R(CS_v)$ according to loss functions. The next step is an iteration process. We choose the most similar superpixel clusters to merge into one cluster on the basis of similarity matrix. Then, a new risk cost $R(CS_u)$

is calculated. If the new risk cost $R(CS_u)$ is less than the former one $R(CS_v)$, this iteration will continue. The whole algorithm is a hierarchical method which is about merging one by one based on [5]. Details and modifications of this process will be presented in Section C.. Afterwards, we will take measure in case of over-segmentation, which we can merge those clusters containing few superpixels into their closest ones. Finally, the FCM algorithm will be executed and we will show the ultimate segmented images through the clustering result and the adjacency matrix of superpixels.

C. Detailed Analysis

In our method, there are generally three steps to accomplish the color image segmentation, which is pre-segmentation, cluster validity analysis based on DTRS and FCM clustering.

(1) Presegmentation. The Turbopixel algorithm [19] is based on the level set technique [22]. Firstly, place K seeds, namely the initialized superpixels dispersed in the image so that distances between superpixel neighbours are all about $\sqrt{\frac{N}{K}}$, where N is the total number of pixels in the image. Also an iterative equation that derived from the curve evolving along its normal vector direction is applied until any point on the evolving boundary reaches the edge of the narrow band. It is worth noting that the number of initialized seeds is user-specified, which leaves more space for users. And another key point is that the true number of superpixels after carrying out this algorithm will be a little more than the user-specified number. In our method, the larger K is, the higher computational complexity it will suffer from and the smaller K is, the worse result it may lead to. Thus, we have tried different values of K in our experiments for different effects.

(2) Cluster validity analysis based on DTRS. This is the key point which we focus on. In [5], the ANDC-DTRS algorithm is used to process data sets in UCI database [15]. To begin with, the similarity matrix between objects should be calculated. The number of objects on each data set in the experiment is no more than 1200. However, this is not applied to the image data, due to tens of thousands of pixels contained in an image. Thus, step (1) is our first measure we have taken to improve this algorithm and make it more suitable for image data. The similarity calculating method in the original algorithm is according to the Euclidean distance metric equation. In our method, we change this magnanimity method into Bhattacharyya Coefficient measure as defined in the following equation

$$BC(p,q) = \sum_{i=1}^{d} \sqrt{p_i(x)q_i(x)}, \tag{12}$$

where d is the number of bins in an color histogram, $p_i(x)$ represents the value of the color histogram in the i^{th} dimension. This is also different from the original ANDC-DTRS algorithm [5]. Next merging steps we used are the same way with the algorithm derived by the extended DTRS model which merges until the risk cost is minimal. In fact, Yu et al. in [5] has proven that we can merge those two clusters whose function defined as:

$$f(C_p, C_q) = \frac{1}{|C_p||C_q|} \sum_{x_i \in C_i} \sum_{x_j \in C_j} P(C|(x_i, x_j)) \tag{13}$$

is the maximal, which can be used as the merging policy. We can merge gradually until f is less than or equal to 0.5, which can be used as the merging termination condition. So what we are mainly according to is equation (13). After the merging process, the number of clusters is achieved and also cluster centroids can be attained by averaging feature values of each cluster.

Our algorithm may split the image into too many regions due to the presegmentation step. Therefore, it is necessary for us to solve the problem of over-segmentation. We incorporate those clusters whose quantities of superpixels are less than the threshold into its closest clusters whose quantities of superpixels are more than the threshold. However, we cannot ensure that this can produce visual effects because they may not adjacent. Note that Bhattacharyya Distance is defined as follows

$$D_B(p,q) = -ln(BC(p,q)), \tag{14}$$

which is utilized to measure the distance between clusters and also the threshold is user-specified. This is just the additional work we add into the original ANDC-DTRS algorithm due to the specific character of the image segmentation. The complete algorithm presented in our method called DTRS_FCM can be seen in Algorithm 2.

(3) FCM clustering. As the number of clusters and cluster centroids are obtained from the previous two steps, the FCM algorithm shown in Algorithm 1 is adopted naturally for clustering. For showing the segmented image with boundaries, it is necessary to combine the clustering result after finishing executing FCM with the superpixels adjacency matrix acquired from the first presegmentation step.

As shown in Algorithm 2, step (1:) and step (2:) are preprocessing steps before cluster validity analysis including pre-segmentation and feature extraction. On the basis of the extended DTRS model, we analyze cluster validity for color images From step (3:) to step (9:) which is our main work proposed in this paper. Step (10:) presents the clustering method for color image segmentation. The final step is used to show results visually.

On the whole, our approach focuses on how to perform clustering validity analysis with the extended DTRS model for the color image segmentation. The cluster merging process is a gradual way. To make DTRS adaptable for the image data, the preprocessing step is added. Besides, we provide the postprocessing step in case of over-segmentation. With these simple operations, our algorithm will show generate reasonable segmentation results based on our experiments in Section IV..

IV. Experiments and Analysis

To evaluate our approach for clustering validity analysis based on the extended DTRS model, we will conduct the

Algorithm 2 DTRS_FCM

Input:

An color image, I;
Superpixelnum, K;
Threshold, t;

Output:

Segmented Image I' with boundaries;

1: Presegment Image I into a set of superpixels with Turbopixel algorithm. Set the initial number of superpixels to K. Let the true number of superpixels be K'.
2: Extract and normalize the color histogram of color image I under HSV color space, which contains 512 bins.
3: Calculate the similarity matrix of superpixels according to the feature extracted in $(2:)$ using Eq. (12).
4: Assume that each superpixel is a separate cluster and the initial result $CS_0 = \{\{x_1\}, \{x_2\}, ..., \{x_n\}\}$.
5: If all the elements in the similarity matrix are equal, then CS_0 is the final result and n is the number of clusters and cluster centroids are all themselves; else let $CS_v = CS_0$, go to step $(6:)$.
6: Calculate val according to Eq. (4) and possibility matrix $P(C|(x_i.x_j))$ according to Eq. (5), (6); Then matrix f in terms of current clustering result CS_v can be attained on the basis of matrix P and Eq. (13).
7: Find the maximal element f_{max} in matrix f. If $f_{max} < 0.5$, then end the algorithm; $|CS_v|$ is the number of clusters and average feature values of each clusters in CS_v are cluster centroids; else, go to step $(8:)$.
8: Merge the two clusters corresponding to the value f_{max} and update matrix f, go to step $(7:)$.
9: For any x_i in CS_v, if $|x_i| \leq t$, then merge x_i into its closest cluster $x_s(|x_s| > t)$ according to Eq. (14).
10: Executing FCM algorithm to segment color image I.
11: Show Image I' with boundaries.

experiments to varied threshold values for different segmentation results.

A. Experiment Setup

The experiments are performed under the environment of Intel(R) Core(TM) i3-2120 CPU 3.30GHZ and Windows 7. In this paper, we take 10 images from the UC-Berkeley Image Segmentation Dataset [23] to carry out detailed test and make comparison with methods proposed in [16], [17].

In our proposed method, we have two thresholds. One is the number of superpixels K required in pre-segmentation and another is the cluster size t to decide whether it should be merged into other cluster in case of over-segmentation. Both of them may control the segmentation results so that we implement Algorithm 2 with several times of parameter tuning. For the number of superpixels, K is assigned with 200, 400, 600 respectively. For the postprecessing threshold, t is assigned with $t = K/10$. Moreover, we also fix the parameters as $K = 400$, $t = 20$ to observe the influence generated by the selection of t.

In this paper, we make use of two benchmarks. One of the commonly used benchmark is Bezdek's partition coefficient [24], where the evaluation function is defined as follows:

$$V_{PC} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} u_{ji}^2}{N}. \tag{15}$$

The properties of this evaluation model were studied in [24], [25]. The evaluate function is used to measure the fuzziness of a clustering result and ranges from 0 to 1. The larger the V_{PC} value is, the better the clustering result will be.

Another is the Xie-Beni function [26]:

$$V_{XB} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} u_{ji}^2 \|x_i - c_j\|^2}{N min_{\forall j \neq k}\{\|c_j - c_k\|^2\}}. \tag{16}$$

According to Xie and Beni, a better clustering result should produce small V_{XB} value.

B. Experiment Results

We adopt the parameter settings as described in Section A. and carry out Algorithm 2. Experiment results on these ten images are shown in Fig. 2. visually. Also the results of cluster numbers, V_{PC} values and V_{XB} values are provided in TABLE I, TABLE II and TABLE III.

C. Experiment Analysis

As can be seen from Fig. 2., DTRS_FCM we propose can generally perform the image segmentation well according to the color features. However, there still exist differences with different parameter settings. When K is small, namely $K = 200$, we can see that the segmentation results are relatively rough and just general boundaries are produced. For example, in Image House, the white eaves cannot be segmented out from the roof and in Image House3, the green trees cannot be separated exactly. This situation will be improved a lot when the value of K increases which has been shown when $K = 400$ and $K = 600$. Thus, the larger K is, the more meticulous the segmentation results will be.

The large value of K will raise the problem of over-segmentation and lead to high time complexity. In terms of execution time, it tends to rise with K increasing because it takes amount of time for the Turbopixel algorithm to split the image into superpixels especially K is large. Thus, it is not appropriate and necessary to make K too large. For showing the over-segmentation problem, we can see from the third and the fifth column where they share the same value of K and own different values of t. Obviously, when t is relatively smaller in the fifth column, clusters that could be merged are less. For example, in Image Church1, Church2, Church3, clusters about the sky do not be merged together, which is not in accordance with our visual perception. However, This is improved in the third column which can be seen clearly in Image Church2, Church3 and House1. Thus, proper proportion between K and t should be taken. Attempt in our experiment shall show that $t = K/20$ is suitable, but it is not definitely applied to all the color images.

In TABLE I, we have listed the number of clusters under different situations of parameter settings. Obviously, larger propotions between K and t tends to produce more clusters. It is worth to note that we cannot assure the post-processing step will lead to great influence in vision because clusters to be merged may not be adjacent to each other. Thus, clusters will still be separated even if they are the closest according to the color histogram feature. On top of that, the V_{PC} values of the AS, the AFHA, the IAFHA, the HTFCM and our proposed DTRS_FCM methods are presented in TABLE II. It is evident to see that DTRS_FCM can always produce the best general distribution even if under different parameter values. The V_{XB} values of various approaches are tabulated in TABLE III. Our method on this benchmark may not perform so stable as V_{PC} likes and also it partially depends on our selection of parameter values. However, it could produce comparable results by contrast to some extent.

V. Conclusions

In this paper, we have combined the decision theoretic rough set model (DTRS) and Fuzzy C-Means(FCM) algorithm to perform color image segmentation. Our proposed algorithm called DTRS_FCM could decide the number of clusters and cluster centroids and pass these two values to carry out FCM algorithm, showing the usefulness of the extended DTRS model in clustering validity analysis. It has made improvements on FCM for no need of random initialization. Extensive experiments in which segmentation results are shown and comparison with V_{PC} and V_{XB} are made have demonstrated the feasibility of our method.

References

[1] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[2] Y. W. Lim and S. U. Lee, "On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques," *Pattern Recognition*, vol. 23, no. 9, pp. 935–952, 1990.

[3] 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, no. 1, pp. 57–68, 1999.

[4] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation," *computerized medical imaging and graphics*, vol. 30, no. 1, pp. 9–15, 2006.

[5] H. Yu, Z. Liu, and G. Wang, "Automatically determining the number of clusters using decision-theoretic rough set," in *Rough Sets and Knowledge Technology*. Springer, 2011, pp. 504–513.

[6] Y. Yao and S. K. M. Wong, "A decision theoretic framework for approximating concepts," *International Journal of Man-machine Studies*, vol. 37, no. 6, pp. 793–809, 1992.

[7] S. French, *Decision theory: an introduction to the mathematics of rationality*. Halsted Press, 1986.

[8] Y. Yao, "Decision-theoretic rough set models," in *Rough Sets and Knowledge Technology*. Springer, 2007, pp. 1–12.

[9] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[10] W.-Q. Zhao, Y.-L. Zhu, and W. Gao, "Information filtering model based on decision-theoretic rough set theory." *Jisuanji Gongcheng yu Yingyong(Computer Engineering and Applications)*, vol. 43, no. 7, pp. 185–187, 2007.

[11] X. Jia, W. Liao, Z. Tang, and L. Shang, "Minimum cost attribute reduction in decision-theoretic rough set models," *Information Sciences*, 2012.

[12] D. Liu, Y. Yao, and T. Li, "Three-way investment decisions with decision-theoretic rough sets," *International Journal of Computational Intelligence Systems*, vol. 4, no. 1, pp. 66–74, 2011.

[13] X. Jia, K. Zheng, W. Li, T. Liu, and L. Shang, "Three-way decisions solution to filter spam email: an empirical study," in *Rough Sets and Current Trends in Computing*. Springer, 2012, pp. 287–296.

[14] P. Lingras, M. Chen, and D. Miao, "Rough cluster quality index based on decision theory," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 7, pp. 1014–1026, 2009.

[15] C. L. Blake and C. J. Merz, "Uci repository of machine learning databases [http://www. ics. uci. edu/˜ mlearn/mlrepository. html]. irvine, ca: University of california," *Department of Information and Computer Science*, vol. 460, 1998.

[16] Z. Yu, O. C. Au, R. Zou, W. Yu, and J. Tian, "An adaptive unsupervised approach toward pixel clustering and color image segmentation," *Pattern Recognition*, vol. 43, no. 5, pp. 1889–1906, 2010.

[17] K. Siang Tan and N. A. Mat Isa, "Color image segmentation using histogram thresholding–fuzzy c-means hybrid approach," *Pattern Recognition*, vol. 44, no. 1, pp. 1–15, 2011.

[18] A. Khan, M. A. Jaffar, and T.-S. Choi, "Som and fuzzy based color image segmentation," *Multimedia Tools and Applications*, pp. 1–14, 2013.

[19] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 12, pp. 2290–2297, 2009.

[20] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[21] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981.

[22] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.

[23] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 416–423.

[24] J. C. Bezdek?, "Cluster validity with fuzzy sets," 1973.

[25] J. C. Bezdek, "Mathematical models for systematics and taxonomy," in *Proceedings of eighth international conference on numerical taxonomy*, vol. 3, 1975, pp. 143–166.

[26] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, no. 8, pp. 841–847, 1991.

Fig. 2. Image segmentation tests on DTRS_FCM algorithm with different parameter values. First column: original images. Second column: $K = 200, t = 20$. Third column: $K = 400, t = 40$. Fourth column: $K = 600, t = 60$. Fifth column: $K = 400, t = 20$. Test images from the first row to the tenth row are House, Church1, Church2, Church3, House1, House3, Sun Flower, Building, Beach, Star Fish, respectively.

TABLE I

NUMBER OF CLUSTERS PRODUCED BY DIFFERENT ALGORITHMS

Images	DTRS_FCM			
	K=200 t=20	K=400 t=40	K=600 t=60	K=400 t=20
House	3	3	4	6
Church1	4	6	4	7
Church2	3	2	3	5
Church3	4	4	3	6
House1	5	4	4	9
House3	3	3	3	4
Sun Flower	2	3	3	5
Building	5	3	3	10
Beach	3	5	4	8
Star Fish	3	4	3	5

TABLE II

V_{PC} TEST FOR DIFFERENT ALGORITHMS

Images	Algorithms							
	AS	AFHA	IAFHA	HTFCM	DTRS_FCM			
					K=200 t=20	K=400 t=40	K=600 t=60	K=400 t=20
House	0.742	0.736	0.729	0.804	0.876	0.875	0.879	**0.897**
Church1	0.694	0.713	0.746	0.761	0.848	0.888	0.864	**0.897**
Church2	0.780	0.753	0.771	0.814	0.921	0.927	**0.929**	0.920
Church3	0.595	0.586	0.586	0.646	0.758	0.780	**0.794**	0.775
House1	0.601	0.636	0.637	0.758	0.7544	0.773	0.767	**0.779**
House3	0.656	0.668	0.670	0.749	0.863	0.860	0.860	**0.868**
Sun Flower	0.616	0.631	0.621	0.671	**0.889**	0.813	0.804	0.755
Building	0.452	0.498	0.472	0.486	0.675	**0.715**	0.702	0.595
Beach	0.593	0.587	0.603	0.7975	0.798	0.798	0.781	**0.836**
Star Fish	0.489	0.497	0.505	0.546	0.793	0.757	**0.810**	0.725

TABLE III

V_{XB} TEST FOR DIFFERENT ALGORITHMS

Images	Algorithms							
	AS	AFHA	IAFHA	HTFCM	DTRS_FCM			
					K=200 t=20	K=400 t=40	K=600 t=60	K=400 t=20
House	0.120	0.118	0.086	0.088	0.158	0.149	0.170	**0.085**
Church1	0.135	0.136	0.117	0.107	0.153	0.137	0.195	**0.101**
Church2	0.256	0.216	0.283	0.254	0.067	0.071	**0.058**	0.107
Church3	0.270	0.214	0.272	**0.140**	0.333	0.300	0.417	0.178
House1	0.269	0.267	0.219	0.228	**0.178**	0.191	0.220	0.248
House3	0.218	0.182	0.140	0.143	0.119	0.176	0.173	**0.107**
Sun Flower	0.234	0.258	0.177	**0.134**	0.206	0.352	0.390	0.342
Building	0.251	**0.248**	0.269	0.288	0.966	1.053	1.114	0.620
Beach	0.237	0.231	0.189	0.253	0.320	0.218	0.342	**0.097**
Star Fish	0.273	0.276	0.279	**0.170**	0.268	0.629	0.254	0.456