Application of Rosette Pattern for Clustering and Determining the Number of Clusters

Ali SADR, Amir Keyvan MOMTAZ
Electrical Engineering Department, Iran University of Science & Technology, Narmak, Tehran, Iran
Sadr@iust.ac.ir, Akmomtaz@ee.iust.ac.ir

***Abstract*— Clustering is one of the most important research topics which has many practical applications such as medical imaging and Non-Destructive Testing (NDT). Most clustering algorithms like K-means, fuzzy C-Means (FCM) and their derivatives require the number of clusters as one of the initializing parameters.**

This paper proposes an algorithm for image clustering with no need to any initializing parameter. In this state-of-the-art, an image is sampled based on a rosette pattern and according to the pattern characteristics, the extracted samples are clustered and then the number of clusters is determined. The centroids of classes are computed by means of a method based on calculation of distribution function. Based on different data sets, the results show that the algorithm improves the capability of the clustering by a minimum of 62.26% and 87.62% in comparison with FCM and K-means algorithms, respectively. Moreover, in dealing with high resolution data sets, the efficiency of the algorithm in clusters detection and run time improvement increases considerably.

***Index Terms*—Clustering, Fuzzy C-means (FCM), Pattern Recognition, Rosette Pattern, Validity Index.**

I. INTRODUCTION

Clustering is an unsupervised pattern classification technique which groups a set of objects into clusters based on their similarity. There are two fundamental purposes in any clustering scenario: The clustering algorithms can be widely classified into hierarchical or partitional groups [1]. Hierarchical clustering algorithms recursively find clusters either in agglomerative mode or in divisive mode. The Single-link [2], average-link [2] and Complete-link [3] algorithms are the samples of Hierarchical clustering algorithms. In partitional clustering algorithms, clustering produces a single partition of the data set which aims to optimize a certain cluster criterion function. The K-means algorithm [4] and Gaussian mixture model (GMM) [5] are the most well known examples of partitional clustering.

Based on this classification, different kinds of clustering algorithms have been reported in the literature each reflecting a different point of view. In these algorithms, it is usually assumed that the number of clusters c is known. For situations where no prior knowledge of c is available, determining the number of clusters automatically would be a difficult task in clustering algorithms.

One of the most widely used clustering algorithms is K-means and its derivatives. Since in these approaches, the clustering criterion is based on Euclidean distance between the samples and calculated centers of clusters, inappropriate value selection of the number of clusters or merging and splitting parameters may deteriorate the performance of the algorithms. Therefore, clustering results would not be satisfactory and they completely depend on parameters initialization. In the past few decades, the basic K-means Algorithm has been extended in many ways [6]-[13]. All these extensions introduce some additional algorithmic parameters that must be specified by the user [14].

One of the extensions of K-means algorithm is Fuzzy C-Means (FCM) clustering. FCM is a data clustering technique wherein each data point belongs to a cluster to some degree that is specified by a membership grade [15]-[16]. Like K-means algorithm, in FCM and its derivatives, it is assumed that the number of clusters is a user defined parameter and therefore they do not propose a mechanism for finding a correct number of clusters. This normally prepares by the use of validity functions.

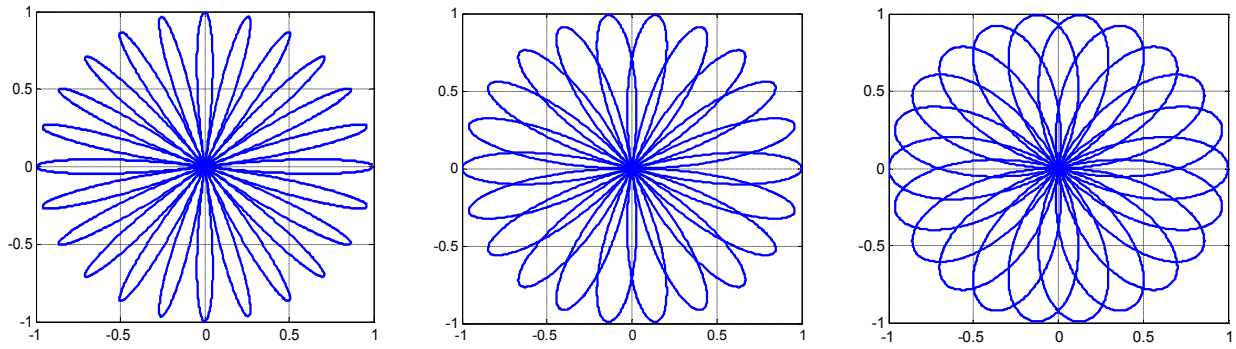
In advanced versions of FCM derivatives [17]-[22], the number of clusters (c) is considered between two predefined values (C_{min} & C_{max}) that represent, respectively, the minimal and maximal numbers of clusters. The best value of c is chosen based on the validity indices that have been proposed in the literature. Their major drawback is high computational cost [23]. In addition, most of the indices use the Euclidean distances in their computation. Therefore, they are able to characterize only compact clusters [24].

Spectral clustering [25]-[26] represents the data points as nodes in a weighted graph. The edges connecting the nodes are weighted by their pair-wise similarity. The fundamental idea is dividing the nodes into two subsets A and B such that the cut size is minimized [14]. Various versions of spectral clustering have been proposed in the literature [26]-[29]. Although spectral clustering has been widely used in several areas such as information retrieval and computer vision [26], when the number of data instances is large, the algorithm requires considerable time and memory to find and store the eigenvectors of a Laplacian matrix [29].

Gaussian mixture model-based (GMM) classifiers are commonly used in image clustering due to the analytical tractability and robustness of the Gaussian distribution [30]. When the underlying mixture component distributions are not necessarily Gaussian, however, there is no guarantee that the Gauss mixture model-based clustering algorithms will be able to capture the mixture components [31].

Other clustering methods are based on neural networks [32]-[33]. The neural networks have the capability of constructing an arbitrary nonlinear mapping from multiple input data to multiple output data through learning sample input versus output relations and estimating appropriate data. Although these networks are executed rapidly, their training is too long [34].

Digital Object Identifier 10.4316/AECE.2011.03013



(a) N_1=13, N_2=11, ΔN=2 (b) N_1=13, N_2=9, ΔN=4 (c) N_1=15, N_2=7, ΔN=8

Figure 1. Rosette patterns with different N_1 & N_2

First time, the rosette pattern was used in infrared seekers for tracking planes [35]. A rosette scan infrared seeker is a tracker that a single or double infrared detector scans the total field of view and detects the heat radiated from the target. Flares are false target that planes release them to keep themselves safe against thermal tracking missiles. In the processing unit of missiles, in order to distinguish real target from the false targets according to data collected by the rosette scan infrared seeker, several algorithms based on image processing such as k-means, ISODATA and ALCA have been proposed [36]-[38]. In addition to common problems in algorithms like k-means and ISODATA, since the number of clusters is not fixed, multiple clusters are recognized as a class. Furthermore, the algorithms require considerable processing time and necessitate parameters modification during the tracking procedure.

Up to present time, as far as the authors are aware, there has been no report on application of rosette pattern in image clustering and determining the number of clusters automatically. The principle object of this paper is to sample, cluster and determine the number of clusters in images by the use of rosette pattern. In comparison with previous clustering methods, the proposed algorithm has some advantages: there is no need to any parameters initialization; by variation of the number of clusters, the performance of the algorithm will not deteriorate. In addition, the run time and clustering efficiency are enhanced. In the proposed method, a sample image is scanned by the rosette pattern. Based on the rosette pattern characteristics, scanned samples are mapped to a linear plane. The converted samples are clustered and the number of clusters is determined. Finally, the clustered samples are remapped to the main plane. To compute the accurate centroid of classes, a method based on calculation of weight function for each point on the rosette pattern is introduced.

In this paper, in order to evaluate the performance of the proposed algorithm, different data sets are considered as exemplary case studies and clustering results by the proposed algorithm are compared with K-means and FCM. However, it will be understood by those skilled in the relevant arts that it is possible to use the proposed approach in the other data sets without departing from the scope of the concept.

II. THEORETICAL BACKGROUNDS

A rosette scan pattern is formed by having two optical elements rotate to the opposite direction according to certain rules with a constant rate [36]. The loci of the rosette pattern at an arbitrary time t can be expressed by (1) [38].

$$x(t)=\frac{1}{2}\delta(\cos 2\pi f_1 t+\cos 2\pi f_2 t)$$
$$y(t)=\frac{1}{2}\delta(\sin 2\pi f_1 t-\sin 2\pi f_2 t) \quad (1)$$

where f_1 and f_2 are two rotational frequencies of prisms and the radius of the rosette pattern is determined by δ. For simplicity, δ is considered as 1. Based on (1), coordinates of a point in a two dimensional array can only be achieved by parameter t. The values of f_1 and f_2 determine the rosette pattern parameters such as scan speed, total number of petals and the petal width. If f_1/f_2 is a rational number, and f_1 and f_2 have the greatest common divisor f such that $N_1=f_1/f$ and $N_2=f_2/f$ are both positive integers, the pattern is closed. Moreover, N_1 and N_2 are the smallest integers satisfying

$$\frac{N_2}{N_1}=\frac{f_2}{f_1} \quad (2)$$

The rosette period, T, is $1/f=N_1/f_1=N_2/f_2$. The number of petals in the rosette pattern is represented by

$$N=N_1+N_2 \quad (3)$$

The parameter representing the width of the rosette pattern petals is

$$\Delta N=N_1-N_2 \quad (4)$$

The value of ΔN determines the overlapping rate of petals and the width of each petal increases by increasing ΔN. If ΔN is getting smaller, the width of leaves gets narrower and for ΔN=2, petals will not cross each other. Figure 1 illustrates different rosette patterns with varying N_1 and N_2 parameters.

The rosette pattern is a function of f_1, f_2 and t. Since the values of f_1 and f_2 are fixed, the pattern position in each point is individually a function of t. Figure 2 shows the image scanning by the rosette pattern. If total number of samples of the rosette pattern is considered as N_T, then N_T is calculated by (5).

$$N_T=2.N.N_P \quad (5)$$

where N is total number of petals and N_P is number of samples in each half of the petal.

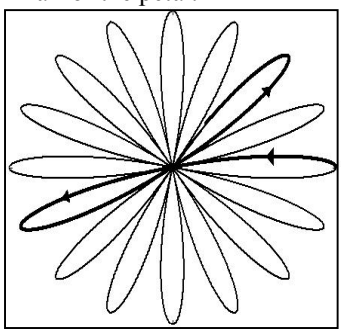


Figure 2. Image scanning by the rosette pattern

To produce the rosette pattern with the specific total number of samples (N_T), it is necessary to select the sample time (Δt) as follows:

$$\Delta t = \frac{1}{f.N_T} = \frac{1}{2.f.N.N_p} \tag{6}$$

III. Proposed Method

Contrary to conventional clustering methods, in the proposed algorithm, there is no need to initialize any parameter. Therefore, in the case of unsuitable initialization, the efficiency of the algorithm will not deteriorate. Moreover, the clustering algorithm based on the rosette pattern is reliable, fast and capable to determine the accurate number of clusters truly.

This paper offers an algorithm based on rosette pattern leading to improve image sampling and clustering. Scanning the images by the rosette pattern causes the number of sampled pixels to reduce. Furthermore, the coordinates of each sample can be expressed by a single variable (t).

3.1 Clustering Algorithm Based on Rosette Pattern (CABRP)

The general strategy in the Clustering Algorithm Based on Rosette Pattern (CABRP) is mapping the samples into a two dimensional space and determining partial clusters. The algorithm consists of three parts. The first part is sampling the image pixels. In the second part, i.e. clustering, to decrease the nonlinear property of the rosette pattern, the samples are mapped to a linear space. The characteristics of the space are determined by the rosette pattern. In the linear plane, all adjacent samples are considered as a partial cluster. Two partial clusters are merged if they have adjacency to each other. Finally, the clustered samples are remapped to the main plane. The steps of the proposed algorithm are explained in the subsequent sections.

3.1.1 Image Scanning & Sampling

At first, the original image is sampled according to the rosette pattern equation. In the proposed algorithm, the rosette pattern is produced by a Matlab™ program. According to equation (1) and increasing t from zero to rosette period (T), the grayscale of each pixel of original image is sampled and coordinates are stored in a two dimensional array. In figure 3, sampling an image by the rosette pattern (N_1=39 & N_2=37) is illustrated.

In order to cover the area of the image precisely, the number of rosette petals (N) should be selected large enough. However, if the number of petals is chosen too large, the speed of the algorithm will reduced. Therefore, N=76 can be a suitable selection.

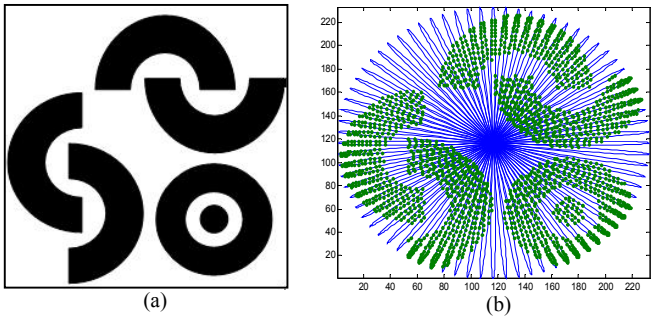


Figure 3. Sampling an image by the rosette pattern, a) original image b) the scanned image

Furthermore, the corner areas of the image are not covered by the rosette pattern. Hence, the image is embedded in the rosette pattern similar to figure 4.

Based on equation (1) and figure 1, the radius of the rosette pattern is limited to 1. For an image with the size of $m \times n$, factor δ in (1) is defined αm if $m>n$. Otherwise, αn. Here, α stands for a scale factor to place the original image in the center of the rosette pattern.

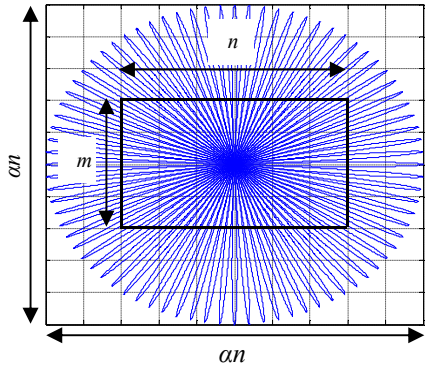


Figure 4. Embedding the image in the rosette pattern

The grayscales of image pixels (I) are sampled by the rosette pattern based on (7):

$$I(t)=\text{pixel}(i,j), \text{ if } i<m \text{ and } j<n \tag{7}$$

where t stands for time parameter.

Since the resolution of the rosette pattern in the center is more than the extremes of the pattern, the performance of images sampling will improve. Figure 5 illustrates the sampling an embedded image in the rosette pattern. The rosette pattern parameters, f_1, f_2 and Δt are 3900Hz, 3700Hz and 0.25μs respectively. Consequently, the values of N and ΔN are computed 76 and 2.

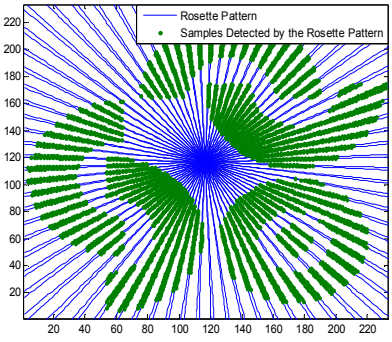


Figure 5. Sampling an embedded image in the rosette pattern

It is seen that all areas of an image are covered by the rosette pattern. Since the size of the image is 233×233, the size of the rosette pattern has been changed to 233×233. Here, the value of α is considered 3.

3.1.2 Clustering

The improved version of clustering algorithm proposed by Jahng *et al*. [40] is used for the clustering. Contrary to the algorithm proposed by Jahng *et al*., the used method has the applicability to grayscale images and covers all areas of images.

In order to decrease the nonlinear properties of the rosette pattern, the samples are mapped to a two dimensional linear space. For this purpose, each rosette petal is divided into halves along the central line of each petal including the central point of the rosette pattern and the outer end point of the petal. The number of each half of the petal increases from i=1 to i=2N one by one counterclockwise. Rosette petals numbering is illustrated in figure 6.

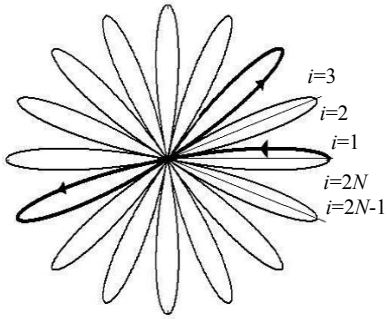


Figure 6. Petals numbering in the rosette pattern

The value of i is defined as

$$i = \begin{cases} \left[\frac{\theta(t)N}{\pi}\right] + 1 & \text{If } \theta(t) \geq 0 \\ \left[\frac{\theta(t)N}{\pi} + 2\pi\right] + 1 & \text{If } \theta(t) \prec 0 \end{cases} \quad (8)$$

The value of $\theta(t)$ is obtained as follows:

$$\theta(t) = \pi(f_1 - f_2)t \quad (9)$$

Each row of half of the petal (j) is composed of Np samples (j=1, 2, ..., Np) that Np is obtained as (6). The values of i and j correspond to row number and column number of a two dimensional plane. If j begins with the sampling point at the center of a petal and ends at the outer end of the petal, the value of j is defined as

$$j = (1 + D).N_p - N_j + 1 \quad (10)$$

If j begins with the sampling point at the outer end of a petal and ends at the center of the petal, the value of j is defined as follows:

$$j = N_j - N_p.D \quad (11)$$

In the foregoing equations (10) and (11), D equals

$$D = \left[\frac{N_j}{N_P}\right] \;\&\; Nj = 1, 2, \ldots, N_T \quad (12)$$

Following the aforementioned stage, the scanned pixels are converted into the two dimensional plane and the grayscale values of the pixels are transferred to the corresponding position on the two dimensional plane based on equations (8) to (12). Figure 7 shows the converted samples to the linear space corresponding to figure 5. In figure 7, based on the value of α and rosette pattern parameters, the values of N and N_P are 76 and 88 respectively.

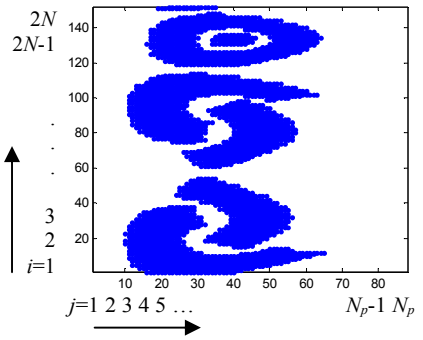


Figure 7. Converted pixels from figure 5 to the linear space

In this step, the clustering method is applied to the samples in the two dimensional plane. At first, partial clusters are defined. For this purpose, all of the continuous data in a row with the same grayscale values are considered as a partial cluster. It is possible to find more than one partial cluster in a row. Details of defining partial clusters are shown in figure 8. The number of gray levels is considered as m.

By defining the partial clusters located in each two consecutive sequential row, their adjacency is checked pair wise. The process is applied to all of the partial clusters. If a partial cluster doesn't have any adjacency to other partial clusters, it is considered as an independent class. At the end of this step, the number of classes is equaled to the number of partial clusters.

1	0	0	gr1	gr1	0		0	0
2	0	0	gr1	0	0		gr2	0
3	0	gr1	gr1	0	0		gr2	gr2
4	gr1	gr1	gr1	0	0		0	0
5	gr1	gr1	0	gr1	gr1		0	0
.								
.								
.								
2N-1	0	0	grm	grm	0		0	0
2N	0	0	grm	grm	grm		0	0
	1	**2**	**3**	**4**	**5**	**…**	**N_P-1**	**N_P**

Figure 8. Determining partial clusters

It is remarkable that in the rosette pattern, the first row (i=1) and the last row (i=2N) are adjacent to each other even though in the two dimensional plane, they are located apart. Therefore, the adjacency between these two partial clusters is examined.

3.1.3 Remapping the clustered samples

Based on the equations (8) to (12), the clustered samples are remapped to the corresponding position in the main plane. So, the purpose of the clustering is obtained. The clustering results of figure 7 are depicted in figure 9.

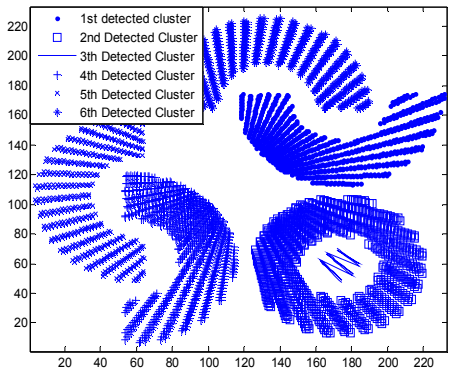


Figure 9. Results of clustering by CABRP for the sample image

3.2 Calculation of the centroid

To calculate the center of each class in the rosette pattern, two methods are proposed based on an averaging method and distribution function [37], [39].

In the averaging method, the positions of all samples of each class are stored in the memory. The stored data is averaged and the result is set as the center of the class. Because of the nonlinearity of the rosette pattern, the number of scan lines passing over the classes is not uniform. There are more scan lines at the center than the extremes of the pattern. Consequently, the computed centroid leans to the center of the rosette pattern.

In the second method, i.e. weighting method, the distribution function is used to compensate the error. This function describes the line density in the rosette pattern. For each sample in the rosette pattern, weight function will be set using the line density. The Weight function value would be greater where the line density is less. The centroid of each cluster is obtained as

$$\hat{x} = \frac{\sum_{i=1}^{m} w_i x_i}{\sum_{i=1}^{m} w_i}, \hat{y} = \frac{\sum_{i=1}^{m} w_i y_i}{\sum_{i=1}^{m} w_i} \quad (13)$$

where w_i and m are the value of ith element of weight matrix and length of weight matrix respectively. Because of the symmetry property of the rosette pattern, the distribution function is calculated only between two neighboring petals. The area between two neighboring petals is divided into L angle directions and N_P radius parts. Therefore, there are $L \times N_P$ points in the area for which a weight function is considered, where N_P is the number of samples for each half of the rosette petal. When, the distance between two petals is divided into L slices, the equation of nth line would be:

$$y_n = \tan(\frac{2\pi}{N}).\frac{n}{L}.x \quad (14)$$

Figure 9 shows the division of two adjacent petals in the rosette pattern (N_1=13, N_2=9) using equation (14). To calculate the distribution function, a class with the radius of 0.1 (the radius of the rosette pattern is normalized to 1) is set to the center of the rosette pattern and moved along the mentioned lines in equation (14) with a defined step (here is $1/N_P$) from r=0 to r=1, where $r = \sqrt{x^2 + y^2}$. When one scan frame of the rosette pattern is finished, the total number of pixels for the relevant class in each position is calculated. Then the weight function is defined as a reciprocal of the distribution function.

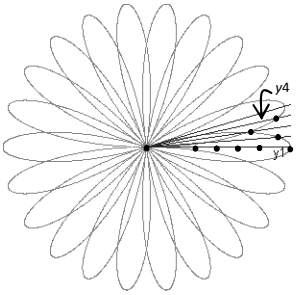


Figure 9. Dividing two neighboring petals

Figure 10 shows the distribution function of the total number of pixels of the corresponding class. The rosette pattern parameters are N_1=39, N_2=37.

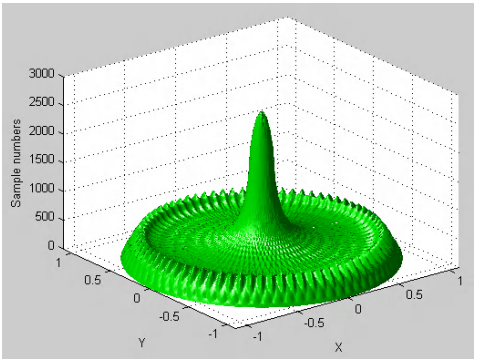


Figure 10. Distribution of the total number of pixels of a class with radius of 0.1.

In order to show the priority of weighting method to averaging method, the centroid of a widespread class is calculated by two methods. Results are demonstrated in figure 11. The centroid computed by weighting method is much closer to original centroid in comparison to centroid computed by averaging method. Since all pixels of the class are not scanned by the rosette pattern, the original centroid and the computed centroid by the weighting method will not be exactly the same.

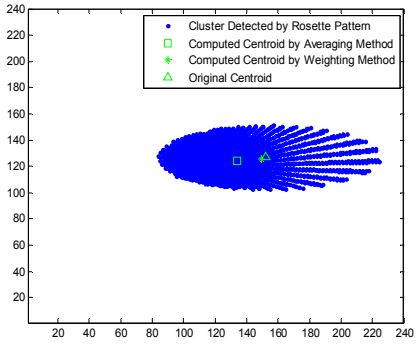


Figure 11. Comparison of calculated centroid by averaging and weighting methods

IV. Simulation Results

In this section, the performance of CABRP is presented and compared with FCM and K-means algorithms. The simulation results are reported for eight data sets. Results are concluded based on the following assumption: in the FCM algorithm, no theoretically rule for choosing m exists. Usually m=2 is chosen [40]. The test for the convergence is performed using ε = 0.001 and distance function ||.|| is defined as Euclidean distance. Moreover, the number of

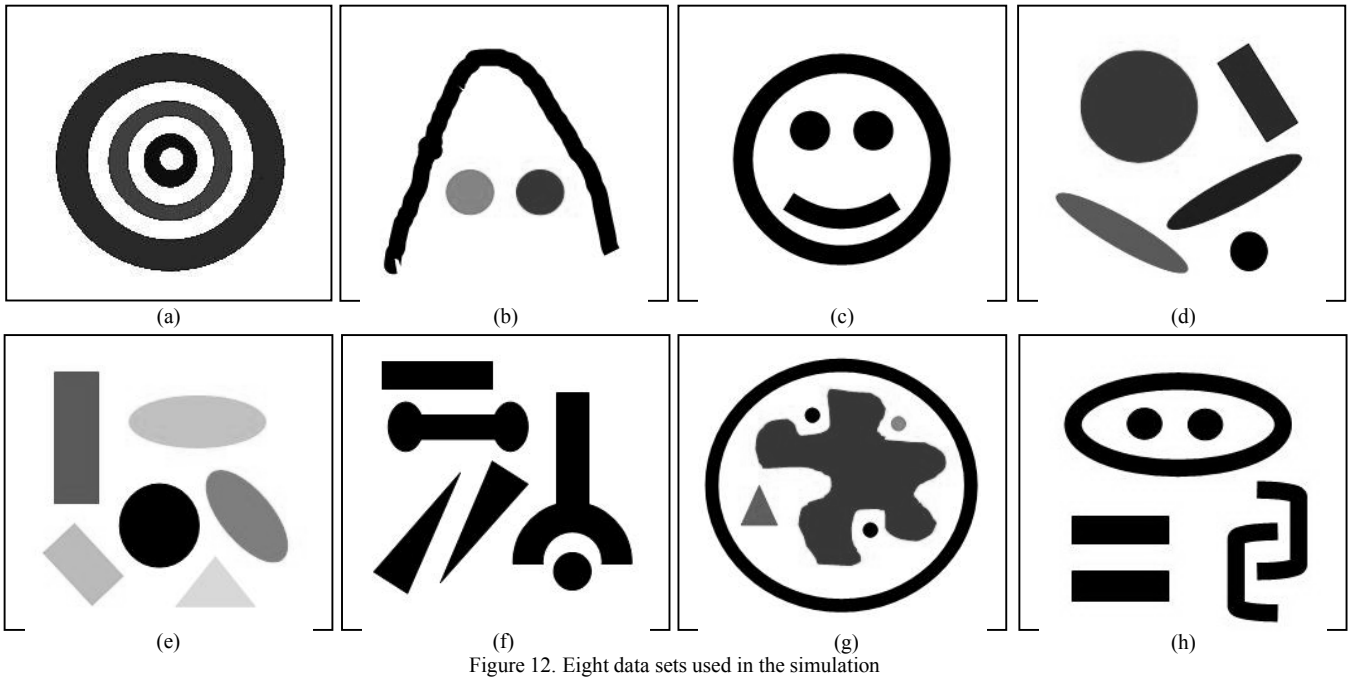


Figure 12. Eight data sets used in the simulation

clusters is initially set. For determination of the optimal number of clusters, the validity indices V_{PC}, V_{PE}, V_{Xie} and V_{FS} are compared with CABRP. For CABRP, in order to achieve the proper resolution and speed, the values of f_1, f_2 and Δt are set to 3900Hz, 3700Hz and 0.25μs. Consequently, using equations (2) to (6), total number of petals (N), overlapping rate (ΔN) and total number of samples (N_T) are computed as 76, 2 and 40,000 respectively. The Rosette pattern parameters are considered fixed for different data sets. In K-means algorithm, only the number of clusters should be initialized.

4.1 Data Sets

The results are released based on eight data sets; the data sets are two dimensional generated arrays and their characteristics are given in table I. The number of clusters varies from three to seven. Data set 1 and data set 7 have the minimum and maximum number of samples respectively. Figure 12 depicts the data sets.

TABLE I. NUMBER OF SAMPLES FOR EACH DATABASE

	C1	C2	C3	C4	C5	C6	C7	Total
DS1	5102	1088	1086	-	-	-	-	7276
DS2	11100	3407	1110	-	-	-	-	15617
DS3	7773	1455	802	801	-	-	-	10831
DS4	2307	6496	2311	2203	730	-	-	14047
DS5	2001	3780	3341	3594	1329	2847	-	16892
DS6	1790	2073	2985	1896	5320	754	-	14818
DS7	6503	481	10539	112	113	112	-	17860
DS8	1800	1650	4911	522	517	1533	1550	12483

4.2 Validation of the algorithm

In order to evaluate the performance of the algorithms four criteria are used: error between cluster prototype and component mean, accuracy of the optimal number of clusters, time cost and stability across different runs. Since in K-means algorithm, seed points are selected randomly and output varies significantly across different runs, the calculated results are average of 20 times runs.

4.2.1. Error between cluster prototype and component mean

Considering the fact that the component means are known for the data sets, the error between cluster prototype and component mean is used to evaluate the performance of the algorithms. The error is defined as equation (15). In the following equation, v_i (i=1,…,N_C) are the cluster centers, N_C is the number of clusters determined by the algorithm, m_j (j=1,…,C) are the component means and C is the true number of component.

$$Error = \sum_{i=1}^{Nc} \min_{1\le j\le C} \left\| v_i - m_j \right\| \tag{15}$$

Table II lists the results for CABRP, FCM and K-means algorithms. For k-means and FCM, the number of clusters is initially set for each data set separately. The use of the weight function improves CABRP clusters detection rate by a minimum of 62.26% and 87.62% in comparison with FCM and K-means algorithms respectively.

TABLE II. ERROR BETWEEN CLUSTER PROTOTYPE AND COMPONENT MEAN FOR THE EIGHT DATA SETS

	CABRP With the Weight Function	CABRP Without the Weight Function	FCM	K-means
DS1	1.586	8.876	100.3	100.3
DS2	1.255	2.135	155.0	155.1
DS3	0.501	3.990	144.4	135.9
DS4	2.329	19.85	57.41	58.66
DS5	7.560	28.75	20.03	61.09
DS6	11.23	39.98	133.1	134.0
DS7	3.522	10.31	188.7	186.3
DS8	3.880	3.79	168.2	159.8

According to table II, it is concluded that FCM and k-means are not able to cluster the data sets truly, while by the use of CABRP, all the clusters are detected accurately. However, the calculated errors for CABRP with weight function are not zero. The reason is that all clusters pixels are not covered by the rosette pattern.

In figure 13, the clustering results for data set 8 by CABRP, FCM and K-means algorithms are demonstrated.

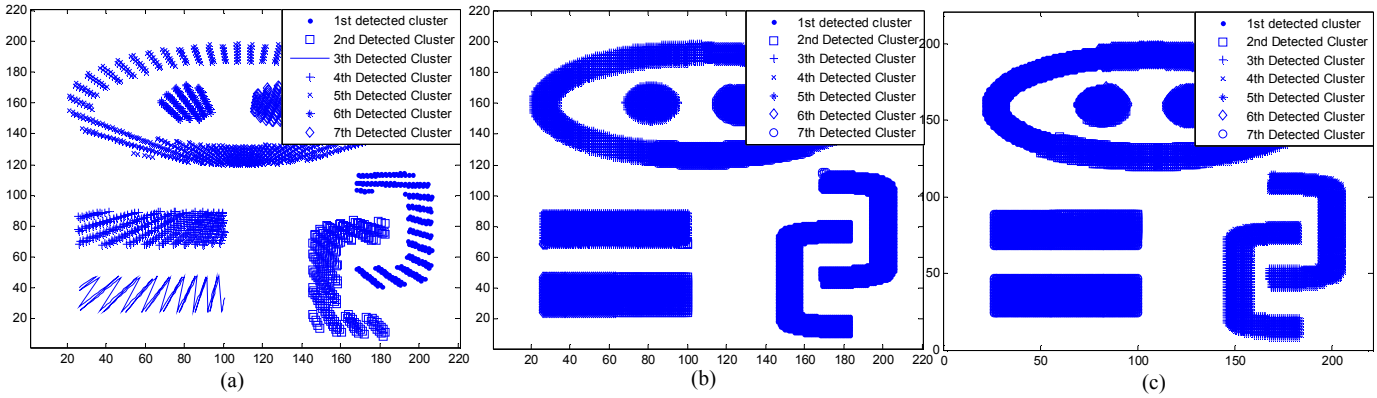


Figure 13. Clustering results for dataset3 (a) by CABRP (b) K-means and (c) FCM

Contrary to CABRP, FCM and K-means algorithms cannot distinct even a cluster truly.

4.2.2 Accuracy of the optimal number of clusters

The major goal of each clustering algorithm is to find the true number of clusters. For FCM, the optimal number of clusters is determined by variation of N_C between N_{Cmin}=2 to $N_{Cmax}=\sqrt{n}$ based on Bezdek's suggestion [41] and calculation of the validity indices. Here, the existing well-known validity indices V_{PC}, V_{PE}, V_{Xie} and V_{FS} are used. However, in CABRP, results of clustering lead to determine the optimal number of clusters.

The optimal number of clusters determined by the algorithms (N_{Copt}) is presented in table III. CABRP shows the best identification ability for the data bases, reaching a 100% accuracy rate. Unlike CABRP, none of the validity indices are able to determine the correct number of clusters even for a data set.

TABLE III. THE OPTIMAL NUMBER OF CLUSTERS BY CABRP AND FCM

	Number of Component (*C*)	CABRP	FCM			
			V_{PC}	V_{PE}	V_{Xie}	V_{FS}
DS1	3	3	2	2	9	12
DS2	3	3	2	2	7	12
DS3	4	4	2	2	11	13
DS4	5	5	2	2	4	9
DS5	6	6	2	2	4	8
DS6	6	6	2	2	11	11
DS7	6	6	2	2	7	13
DS8	7	7	2	2	4	12
Accuracy rate		8/8	0/8	0/8	0/8	0/8

4.2.3 Time cost

In this part, the proposed algorithm is evaluated by the required convergence run time. All the algorithms are executed on a VIAO FZ244 2.00 GHz PC with 2 GB memory running Windows Vista™ Home Premium.

The run time's comparison of CABRP and four validity indices for eight data sets is shown in table IV. CABRP has the best performance among the other methods particularly when the number of samples increases. On the other hand, in dealing with clusters with large number of data vectors, CABRP has much better performance in comparison with FCM algorithm. The run times of validity indices rarely come less than 50 second while for CABRP, its run time do not exceed 0.029 second. On average, in determining the optimal number of clusters, CABRP is 99.97% faster than FCM.

TABLE IV. THE RUN TIMES (*SEC.*) COMPARISON OF CABRP AND FCM FOR THE DATA SETS

	CABRP	FCM			
		V_{PC}	V_{PE}	V_{Xie}	V_{FS}
DS1	0.016	69.74	69.71	73.86	75.71
DS2	0.029	49.76	60.41	47.56	47.45
DS3	0.015	114.9	117.7	100.2	122.2
DS4	0.014	50.23	47.02	45.03	57.43
DS5	0.020	41.79	45.10	42.37	44.56
DS6	0.023	46.17	49.61	47.98	45.23
DS7	0.025	132.5	130.6	136.5	135.8
DS8	0.016	36.97	36.80	38.84	37.33

4.2.4 Stability across different runs

Since in algorithms like K-means and its derivatives, the cluster centers are selected randomly, the outputs vary significantly across different runs. In CABRP, the clustering criterion is based on the neighborhood properties and no parameters related to clusters are initialized. Therefore, the clustering results are just dependent on samples position and the algorithm is completely stable in different runs.

V. CONCLUSIONS

This paper investigates an algorithm for image clustering and determining the number of clusters. The proposed method for clustering is based on image scanning using the rosette pattern and clustering the sampled pixels according to the rosette pattern characteristics. Appropriate method for better image scanning is presented and a clustering algorithm which is able to apply to grayscale images is used. Contrary to the most clustering algorithms requiring parameters initialization, the CABRP results merely depend on samples' position giving complete stability in different runs.

Experiments on eight data sets with different number of clusters and size prove that CABRP improves clusters detection rate by at least 62.26% and 87.62% compared with FCM and K-means algorithms. Furthermore, the proposed approach has proved to be very efficient in terms of run time improvement by 99.97% compared with FCM algorithm. However, this algorithm may be more efficient in dealing with high resolution data sets. Also, experiments on data sets show that CABRP is able to yield the accurate number of clusters and performs much better than all other tested methods.

REFERENCES

[1] A. K. Jain, R. C. Dubes, *Algorithms for clustering data*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[2] P. H. A. Sneath, R. R. Sokal, *Numerical taxonomy*, Freeman, San Francisco, London, 1973.

[3] B. King, Step-wise clustering procedures, *J. Am. Statist. Assoc.* vol. 69, pp. 86–101, 1967.

[4] J. MacQueen, "Some methods for classification and analysis of multivariate observations," Fifth Berkeley Symposium on Mathematics, Statistics and Probability, University of California Press, pp. 281–297, 1967.

[5] B. S. Everitt and D. J. Hand, *Finite mixture distributions*, London, U.K.: Chapman and Hall, 1981.

[6] G. H. Ball, D. I. Hall, "ISODATA- A novel method of data analysis and classification," Stanford Res. Inst., California, 1965.

[7] E. W. Forgy, "Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications," *Biometrics*, vol 21, pp. 768–769, 1965.

[8] S. Eschrich, K. Jingwei, L. O. Hall, D. B. Goldgof, "Fast accurate fuzzy clustering through data reduction," *IEEE Trans. Fuzzy Systems*, vol. 11, no. 2, pp. 262–270, 2003.

[9] M. Steinbach, G. Karypis, V. Kumar, "A comparison of document clustering techniques," KDD Workshop on Text Mining, 2000.

[10] D. Pelleg, A. Moore, "Accelerating exact k-means algorithms with geometric reasoning," Proc. Fifth Internat. Conf. on Knowledge Discovery in Databases, AAAI Press, pp. 277–281, 1999.

[11] P. S. Bradley, U. Fayyad, C. Reina, "Scaling clustering algorithms to large databases," Proc. 4th KDD.1998.

[12] D. Pelleg, A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," 17th Int. Conf. on Machine Learning. pp. 727–734, 2000.

[13] L. Kaufman, P. J. Rousseeuw, "Finding groups in data: An introduction to cluster analysis," Wiley series in Probability and Statistics, 2005.

[14] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, pp. 651–666, 2010.

[15] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cyberne*, vol. 3, pp. 32–57, 1973.

[16] J. C. Bezdek, Pattern recognition with fuzzy objective function algorithms, Plenum Press, New York, 1981.

[17] H. Sun, S. Wang, Q. Jiang, "FCM-based model selection algorithms for determining the number of clusters," *pattern recognition society*, vol. 37, no. 10, pp. 2027–2037, 2004.

[18] A. Baraldi, P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition- part I," *IEEE Trans. Syst. Man, Cybern. B,* vol. 29, no. 6, pp. 778–785, 1999.

[19] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 2, pp. 133-155, March 2009.

[20] U. Maulik, S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, 2002.

[21] M. K. Pakhira, U. Maulik, and S. Bandyopadhyay, "Validity index for crisp and fuzzy clusters," *Pattern Recognition*, vol. 37, no. 3, pp. 487–501, 2004.

[22] S. M. Pan and K. S. Cheng, "Evolution-based tabu search approach to automatic clustering," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 5, pp. 827-838, Sep. 2007.

[23] Y. Wang, C. Li, and Y. Zuo, "A Selection model for optimal fuzzy clustering algorithm and number of clusters based on competitive comprehensive fuzzy evaluation," *IEEE Tran. on Fuzzy Systems*, vol. 17, (3), pp. 568–577, 2009.

[24] S. Saha and S. Bandyopadhyay, "Performance evaluation of some symmetry-based cluster validity indexes," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 4, pp. 420-425, Jul. 2009.

[25] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214-225, Feb. 2004.

[26] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[27] S. X. Yu, J. Shi, "Multiclass spectral clustering," Proc. Int. Conf. on Computer Vision, pp. 313–319, 2003.

[28] M. Belkin, P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in Neural Information Processing Systems*, vol. 14, pp. 585–591, 2002.

[29] W. Y. Chen, Y. Song, H. Bai, C J. Lin, and E. Y. Chang, "Parallel spectral clustering in distributed systems", *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 33, no. xx, 2011, to be published.

[30] R. M. Gray, J. C. Young, and A. K. Aiyer, "Minimum discrimination information clustering: modeling and quantization with Gauss mixtures," Proc. Int. Conf. Image Processing, vol. 3, pp. 14-17, 2001.

[31] K. M. Ozonat and R. M. Gray, "Guass mixture image classification for the linear image transforms," IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 5, pp. v/337 - v/340, 2005.

[32] R. P., Lippman, "An introduction to computing with neural nets," *ASSP Magazine, IEEE*, vol. 4, no.2, pp. 4–22, 1987.

[33] S. Liu, C. Ume, and A. Achari, "Defects pattern recognition for flip-chip solder joint quality inspection with laser ultrasound and interferometer," *IEEE transactions on electronics packing manufacturing*, vol. 27, no. 1, pp. 59–66, 2004.

[34] S. Haykin, *Neural Networks- A comprehensive foundation*, New Jersey: Prentice Hall, 1999.

[35] S. G., Jahng, H. K., Hong, and J. S. Choi, "Dynamic simulation of the rosette scanning infrared seeker and an IRCCM using the moment technique," *Optical Engineering*, vol. 38, no. 5, pp. 921–928, 1999.

[36] S. G. Jahng, H. K. Hong, and J. S. Choi, "Simulation of rosette infrared seeker and counter-countermeasure using K-means algorithm," *IEICE Tran. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E82-A, no. 6, pp. 987-993, 1999.

[37] S. G. Jahng, H. K. Hong, D. S. Seo, and J. S. Choi, "New infrared counter-countermeasure technique using an iterative self-organizing data algorithm for the rosette scanning infrared seeker," *Optical Engineering*, vol. 39, no. 9, pp. 2397-2404, 2000.

[38] S. G. Jahng, H. K. Hong, J. S. Choi, "Clustering method for rosette scan images," *US patent*, number 6,807,307 B2, Oct. 19, 2004.

[39] S. B. Shokouhi, A. K. Momtaz, H. Soltanizadeh, "The new weighting and clustering methods for the rosette pattern," *WSEAS Transactions on information science & applications*, vol. 2, no. 9, pp. 1250-1257, 2005.

[40] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Norwell, USA: Kluwer Academic publishers, 1996.

[41] J. C. Bezdek, *Pattern recognition in handbook of fuzzy computation*, IOP Publishing Ltd., Boston, MA, 1998.