International Journal of Computational Intelligence in Control
Vol. 12 No. 2 (July-December, 2020)

The Pyramid Fuzzy C-means Algorithm

Dan E. Tamir*[a] and Abraham Kandel[b]

[a]Computer Science, Texas State University, San Marcos, Texas

[b]Computer Science and Engineering, University of South Florida, Tampa Florida
**Corresponding author: Dan E. Tamir, dt19@txstate.edu*

Abstract: *Researchers have observed that multistage clustering can accelerate convergence and improve clustering quality. A two-stage and two-phase fuzzy C-means (FCM) algorithms have been reported. A pyramid multistage approach, however, has not been applied to FCM. This paper describes pyramid FCM clustering, where in the first stage the FCM uses an arbitrary partition matrix applied to a low resolution input sample. Next, the resultant partition matrix is used to seed the following, higher resolution stage where the sample size is doubled. The process of seeding higher resolution FCM using the results of lower resolution FCM, continues until the entire data is clustered. The utility and validity of the traditional FCM, two-stage FCM, two-phase FCM, and pyramid FCM are tested through fuzzy clustering of synthetic data and natural color images. The pyramid FCM outperforms the two-stage and two-phase multistage variants and obtains a speedup of ~3X, while maintaining the same or slightly better clustering quality than the traditional FCM. The two-stage and two-phase multistage variants achieve a speedup of about 2X with slight degradation in performance. Furthermore, all the multistage variants can be used to identify several local optimum solutions at the same time the traditional C-means identifies one solution.*

Keywords: *Pattern Recognition, Clustering, Cluster Validity, Quantization, Fuzzy Clustering, Fuzzy C-means.*

1. INTRODUCTION

Numerous clustering, optimization, and classification algorithms, apply a training-phase where the system adjusts parameters using a subset of the data, referred to as the training-set. Examples of such algorithms include K-means, fuzzy C-means (FCM), ISODATA, Kohonen Neural networks, and simulated annealing [12, 16, 27, 33, 44]. Researchers observed that a multistage based training-procedure can accelerate convergence and improve the quality of the training as well as the quality of the classification/decision phases of many of these algorithms [2, 22, 26, 38]. Research reports show that the pyramid K-means clustering algorithm and multi-resolution Kohonen neural networks yield two-to-four times convergence speedup [41, 42].

The FCM algorithm provides a soft (fuzzy) assignment of patterns to clusters. The assignment is represented by a partition matrix. The algorithm starts with an initial partition matrix and attempts to improve the partition according to a given quality criterion. Seeding the FCM is done by selecting an initial partition matrix. Alternatively and equivalently the seeding can be accomplished by selecting initial cluster centers. Improvements to FCM due to a two-stage approach are reported in [2]. A two-phase framework where the first phase includes linear multistage sampling with no replacement has been reported in [10]. There are no reports, however, on the performance of pyramid FCM which is another form of multistage FCM.

In multistage FCM, clustering of low resolution data-samples is used to seed FCM with a higher resolution sample. The objective of this procedure is to reduce the computational cost and improve the quality of the clustering process. The two-stage algorithm reported in [2] starts with a low resolution data-sample, and clusters this data. The initial partition matrix for this stage is chosen through commonly used methods reported in the literature [6, 25]. The clustering performed in the low resolution stage is used to seed FCM with the entire training-data. The two-phase approach of [10] refines the two-stage algorithm. The first phase applies several stages where data is accumulated in a linear fashion through sampling with no replacement. As data is accumulated, the results of clustering are refined. The first phase terminates after clustering a portion of the entire data. The second phase uses the results of the first phase to cluster the entire data.

The pyramid multistage FCM is a refinement of the two-stage and the two-phase approaches; which accumulates training-data exponentially. The proposed method involves a number of stages. The first stage starts with a low resolution sample. For each successive stage, the data-set is re-sampled with replacement with twice the resolution of the previous stage. The final partition matrix obtained in stage I is used as the initial partition matrix for stage $I + 1$. The process is repeated until the sample set is equal to the original data-set.

This paper describes the pyramid multistage FCM procedure and provides an extensive comparative study comparing the pyramid FCM to the two-stage/two-phase FCM. The FCM algorithm and the multistage variants are tested through clustering of a set of synthetic data and quantization of natural color images. The validity of clustering is assessed using a Binomial Monte Carlo analysis [24]. In the average case, the FCM obtains a speedup of 3X over the traditional FCM, while maintaining the same or a slightly better quantization quality. On the other hand, the comparison study shows that the pyramid FCM outperforms the two-stage and the two-phase FCM variants which achieve a speedup of about 2X with slight quality degradation. Moreover, the two-stage, two-phase, and the pyramid FCM variants can be used to identify several local optimum solutions at the same time the traditional FCM identifies one solution.

The rest of the paper is organized in the following way. Section 2 reviews related research and presents two multistage FCM algorithm variants the two-stage and the two-phase. Section 3 introduces the theoretical background of the FCM algorithm, lists metrics used to assess the quality of clustering, and provides the details of the traditional, and pyramid implementations of the algorithm. Section 4 describes a set of experiments conducted to assess the performance and validity of FCM, two-stage, two-phase, and pyramid FCM. Finally, section 5 includes conclusions and proposals for further research.

2. REVIEW OF RELATED RESEARCH

2.1. Iterative Optimization Clustering Techniques

Clustering is a widely-used data classification method applied in numerous research fields including image segmentation, vector quantization, data mining, and data compression [11, 13, 30, 32, 40, 45]. K-means is one of the most commonly used clustering algorithms, and the LBG vector quantization (VQ) algorithm with unknown probability distribution of the sources, which is a variant of K-means, is utilized in many applications [30, 33]. The LBG algorithm has been intensively researched. Some of these research results which are relevant to K-means and fuzzy C-means are reviewed next.

Lloyd proposes an iterative optimization method for quantizer design; which assumes that the distribution of the data is unknown and attempts to identify the optimal quantizer [31]. This approach is equivalent to 1-means (that is K-means; with $k = 1$). While Lloyd's method yields optimal minimum mean square error (MMSE) for the design of one dimensional quantizer, its extension to multi-dimensional data quantizer (i.e., vector quantization) with unknown distributions is not guaranteed to yield optimal results [31]. Consequently, K-means with $k > 1$ is not guaranteed to reach a global optimum.

Linde, Buzo, and Gray (LBG) method for vector quantization (VQ) with unknown underlying distribution generalizes Lloyd's iterative method and sets a VQ design procedure that is based on K-means [30]. The LBG VQ procedure is currently the most commonly used/researched VQ approach. Garey has shown that the LBG VQ converges in a finite number of iterations, yet it is NP complete [17]. Thus, finding the global minimum solution or proving that a given solution is optimal is an intractable problem. Another problem with K-means is that the number of clusters (k) is fixed and has to be set in advance of executing the algorithm. ISODATA is a generalization of K-means which allows splitting, merging, and eliminating clusters dynamically [4, 6]. This might lead to better clustering (better local optimum) and eliminate the need to set k in advance. ISODATA, however, is computationally expensive and is not guaranteed to converge [44].

Several clustering algorithms and combinatorial optimization techniques, such as genetic algorithms and simulated annealing, have been devised in order to enforce the clustering algorithm out of local minima [1, 12]. These schemes, however, require long convergence time, especially for large clustering problems. Fuzzy C-means (FCM) and fuzzy ISODATA generalize the crisp K-means and ISODATA. The FCM clustering algorithm is of special interest since it is more likely to converge to a global optimum than many other clustering algorithms including K-means. This is due to the fact that the cluster assignment is "soft" [7, 25]. On the other hand, the FCM attempt to "skip" local optima may bear the price of numerous soft iterations and can cause an increase in computation time. FCM is used in many applications of pattern recognition, clustering, classification, compression, and quantization including signal and image processing applications such as speech coding, speech recognition, edge detection, image segmentation, and color-map generation [5, 7, 10, 20, 25, 35, 39, 46]. Thus, improving the convergence time of the FCM is of special importance.

2.2. Accelerating Clustering Convergence Rate

Multistage processing is a well known procedure used for reducing the computational time of several applications; specifically, image processing procedures. This method uses a sequence of reduced resolution versions of the data to execute an image processing task. Results of execution at a low resolution stage are used to initialize the next, higher resolution, stage. For example, Coleman proposes an algorithm for image segmentation using K-means clustering [11]. Hsiao, have applied Coleman's technique for texture segmentation [22]. He has used a $\frac{1}{16}$-sample of the image to identify k. Huang and Zhu have applied the Coleman algorithm to DCT based segmentation and color separation respectively [23, 47]. Like Hsiao, they have used -of the image-pixels to set up the parameters of the final clustering algorithm; where the final clustering is performed on the entire image. They have found that the final cluster-centers obtained in the training-stage are very close to the final cluster-centers obtained from clustering the entire image. This lends itself to a two-stage K-means procedure that uses one low resolution sample to initiate the parameters of the actual clustering. Pyramid processing is a generalization of the two-stage approach where the resolution of samples is growing exponentially; each execution stage doubles the number of samples.

Additional applications of multistage architectures are reported in the literature [10, 21]. Rosenfeld surveys the area and proposes methods for producing the multistage snapshots of an image [38]. Kasif shows that multistage linking is a special case of ISODATA [26], and Tilton uses multistage for clustering remote sensing data [43]. Tamir introduces a pyramid multistage method to non supervised training in the context of K-means, and neural networks. He has shown that the pyramid approach significantly accelerates the convergence of these procedures [41, 42].

Several papers deal with accelerating the convergence of FCM [9, 21, 28]. Altman has implemented a two-stage FCM algorithm [2]. The first stage operates on a random sample of the data, and the second stage uses the cluster centers obtained in the first stage to cluster the entire set. Instead, we use a multistage pyramid approach with multiple stages; each stage operates on higher resolution data where the resolution grows in an exponential fashion. Our method is compared to Altman's two-stage approach, and we show a significant improvement in performance. Cheng improves the method proposed by Altman and has investigated a two phase approach. The first phase implements a linear multistage algorithm which operates on small random slices of the data. Each slice contains Δ% of the data. The algorithm finds the cluster centers of the first slice (say S_1), then use these centers as initial centers for clustering a sample that contains the first slice and an additional slice (S_2) obtained through random sampling. After running the multistage phase for n stages, the final centers for the combination of slices $\{S_1, S_2, ... S_n\}$ which contain $n\Lambda$% of the entire data are obtained. Next, in the second phase, these centers are used to cluster the entire data. The research reported in this paper, however, extends this method, and rather than using two phases and linear multistage sampling with no replacement; we use a pyramid sampling (i.e., exponential growth in the sampling). Another difference is that our method uses sampling with replacement which is less susceptible to bias. Results presented in this paper show that the pyramid approach outperforms Cheng's two-stage approach. Other approaches for improving the convergence rate of clustering include data reduction techniques and data sampling using hypothesis testing [14, 34].

A related research effort deals with clustering of very large data-sets which are too big to fit available memory. One approach to this problem is using incremental algorithms [18, 29]. Several of these algorithms load a slice of the data, where the size of a slice is constraint by available memory, and cluster this slice [8, 15]. Results of clustering current slices (e.g., centers, partition matrices, dispersion, etc.) are used in the process of clustering upcoming slices. Hore has proposed a slice based single-pass FCM algorithm for large data-sets [21]. The proposed method lumps data that has been clustered in previous slices into a set of weighted points and uses the weighted points along with fresh slices to commence with the clustering of the entire set in one path [21]. Another approach for clustering large data-sets is to sample, rather than slice, the data [34].

It is interesting to note that K-means, FCM, Neural Networks (e.g., Kohonen Neural networks), and many other iterative optimization algorithms have two main modes of operation, the batch mode and the parallel-update mode. For example, in the batch mode execution of FCM, each iteration considers every pattern individually, and the centers are updated with respect to every pattern considered. The parallel-update mode, which is less computationally expensive and the predominantly used mode in most current applications, assigns all the patterns to the relevant clusters and then updates the centers. In this context, the slice approach which is used for large data-sets can be considered as a hybrid of batch and parallel-update.

This brings the issue of parallel processing of clustering algorithms. Several ways to partition and distribute the clustering task have been considered [3, 21, 36, 37, 43, 45]. One potential way it to assign a set of samples or a slice of data to each processor and eventually merge the cluster centers obtained from each processor into one set of centers. We plan to address this problem as a future research subject.

3. THE FUZZY C-MEANS AND MULTISTAGE FUZZY C-MEANS CLUSTERING ALGORITHM

The fuzzy C-means algorithm (FCM) is a generalization of the crisp K-means clustering. Actually, the generalization is quite intuitive. In the K-means algorithm, set membership is crisp. Hence, each pattern belongs to exactly one cluster. In the FCM, set membership is fuzzy and each pattern belongs to each cluster with some degree of membership. The following section formalizes this notation.

Let $X = \{x_1, x_2, ..., x_m\}$, where $x_i \in R^n$, be a set of m, n-dimentsional vectors representing the data to be clustered into c clusters $S = \{S_1, S_2, ..., S_c\}$ with cluster centers $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$. Under the FCM each element x_j belongs to every cluster S_i with some degree of membership u_{ij}. Hence, the matrix $U = [u_{ij}]$, referred to as the partition matrix, represents the fuzzy cluster assignment of each vector x_j to each cluster S_i. The goal of FCM is to identify a partition matrix U, such that U optimizes a given objective function. A commonly used FCM objective function is defined to be:

[EQ-4]

Where $q > 1$ is weighting exponent. In this research q is set to 2.

The most common measures for FCM clustering quality are: (1) The value of objective function, (2) the partition coefficients, (3) the classification entropy, (4) measures of deviation of the partition matrix from a matrix obtained with uniformly distributed data, and (5) measures of induced fuzziness [6, 7, 24]. It should be noted that some of the quality criteria are derived from distortion measures. Hence, in this case the goal is to minimize distortion, and high quality means low distortion. In other words, the quality can be considered as the inverse of distortion. Measures 1 trough 5, assume that the end result of the clustering is soft. Nevertheless, in many cases, it is desirable to obtain "hard clustering" assignment to be used for vector quantization, image segmentation, or other classification applications. In these cases two additional quality criteria can be considered: 6) the rate distortion function, and 7) the dispersion matrix [24, 44]. Of all these measures, 1, 6, and 7 are most commonly used. In specific, metric 1, the functional J_q can be interpreted as a generalized distortion measure which is the weighted sum of the squared distances from all the points in the cluster domain to their assigned cluster center. The weights are the fuzzy membership values [7, 25]. Hence, this metric is proportional to the inverse of the quality of FCM. Lower distortion denotes higher quality. Metrics 6 and 7 are further elaborated in the next section.

In general, the rate distortion function is used when the FCM is utilized for quantization. In this case, after convergence, the matrix $U = [u_{ij}]$ is defuzzified; e.g., by using a nearest neighbor assignment. The compression rate of fuzzy C-means is fixed by the selection of c. Hence, the rate distortion quality-measure boils down to the MMSE; given by:

$$D = \frac{1}{m}\sum_{i=1}^{c}\sum_{x_j \in \omega_i} \| x_j - \omega_i \| \qquad \text{[EQ-2]}$$

Again, lower distortion denotes higher quality. When the clustering is used for classification, a quality criteria that measure the density of cluster as well as the relative distance between clusters can be used to estimate the recognition accuracy. In this case a dispersion measure can be used. To elaborate: Let $S = \{S_1, S_2, ..., S_c\}$ be the set of clusters obtained through "hard clustering," and let $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$ be the set of the corresponding cluster centers, then, W_i the Within Dispersion Matrix of the cluster S_i is defined to be the covariance matrix of the set of elements that belong to S_i. The within dispersion matrix of S, (W), is a given function of the entire set of the within dispersion matrices of the individual clusters. For example, the elements of W can be the averages of the compatible elements of W_i for $1 \le i \le c$. The Between dispersion matrix of S, (B), is the covariance matrix of Ω. The quality of the clustering can be expressed as a function of the within dispersion matrix W and the between dispersion matrix B. A commonly used dispersion function is [24]:

$$D = tr(W) / tr(B) \qquad \text{[EQ-3]}$$

where $tr(M)$ is the trace of the matrix M.

3.1. The Fuzzy C-means Algorithm

The FCM consists of two main phases; setting/updating the membership of vectors in clusters and setting/updating cluster centers. Some variants of FCM start with a set of centers which induces a partition matrix [7, 25]. In this case, seeding the algorithm relates to the initial selection of centers. Other variants initialize a partition matrix which induces initial centers [6]. Hence, seeding these FCM variants; amounts to initializing the partition matrix. The two approaches are virtually equivalent choosing one over the second is just a matter of convenience related to the format of data and the form of the application. We are using the second approach where the seeding relates to selecting the initial partition matrix. Hence, in the seeding step, the membership matrix is initialized. In the next iterations, the cluster centers are calculated and the partition matrix is updated. Finally, the value of the objective function for the current classification is calculated. The algorithm terminates when a limit on the number of iterations is reached or a "short circuit condition" is met. A commonly used termination condition halts the algorithm when the derivative of the distortion function is small. Because the C-means algorithm is sensitive to the seeding method, a variety of procedures have been proposed for selecting seed points [1, 3]. The following paragraphs include formal definition of the algorithm as well as pseudo-code.

Given a set of vectors $X = \{x_1, x_2, ..., x_m\}$, where $x_i \in R^n$ and an initial partition matrix $U^{(0)}$, the FCM is an iterative algorithm for partitioning a set of vectors into c clusters $S = \{S_1, S_2, ..., S_c\}$, with cluster centers $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$. In iteration l the algorithm uses the cluster centers $\Omega^{(1)} = \{\omega_1^{(l)}, \omega_2^{(l)}, \cdots, \omega_c^{(l)}\}$ induced by the partition matrix $U^{(l)}$ to re-partition the data-set and obtain a new partition matrix $U^{(l+1)}$. Cluster centers at iteration l are computed according to:

$$\omega_i^{(l)} = \left(\sum_{j=1}^{m}\left(u_{ij}^{(l)}\right)^{q-1} \cdot x_j\right) / \left(\sum_{j=1}^{m}\left(u_{ij}^{(l)}\right)^{q-1}\right) \qquad \text{[EQ-4]}$$

The matrix $U^{(l+1)} = [u_{ij}^{(l+1)}]$ is calculated according to

$$U^{(l+1)} = [u_{ij}^{(l+1)}] = \sum_{l=1}^{c}\left(\frac{\| x_j - \omega_i^{(l)} \|}{\| x_j - \omega_l^{(l)} \|}\right)^{-\frac{2}{q-1}} \qquad \text{[EQ-5]}$$

The process of center induction, data partition, and matrix update continues until a given termination condition which relates to an optimization criteria or limit on the number of iterations is met. The following is a commonly used criterion[30]:

$$\left| \frac{J_q^{(l-1)} - J_q^{(l)}}{J_q^{(l-1)}} \right| < \varepsilon \qquad \text{[EQ-6]}$$

The following is a pseudo code of the algorithm.

Algorithm-1 baseline FCM:

1. Parameters:
   - *a.* $X = \{x_1, x_2, ..., x_m\}$, ($x_j \in R^n$)- a set of vectors.
   - *b.* m - the number of vectors
   - c. c - the number of partitions
   - *d.* q – a weighting exponent ($q > 1$)
   - *e.* $U^{(k)} = [u_{ij}^{(k)}]$ – the partition matrix at iteration k
   - f. $\Omega^{(k)} = \{\omega_1^{(k)}, \omega_2^{(k)}, ..., \omega_c^{(k)}\}$ - the set of clustering centers at iteration k
   - *g.* N- the maximum number of iterations
   - *h.* $J_q^{(k)}$ – the objective-function's value at iteration k
2. Set $k = 0$, choose an initial partition matrix $U^{(0)}$
3. In iteration $k \geq 0$ let $\Omega^{(k)} = \{\omega_1^{(k)}, \omega_2^{(k)}, ..., \omega_c^{(k)}\}$ be the induced clustering centers computed by equation 4.
   - a. Set $U^{(k+1)} = [u_{ij}^{(k+1)}]$ according to equation 5.
   - b. Compute $J^{(k+1)}$ according to equation 1.
   - c. Set $k = k + 1$.
4. Stop if $k = N$; or if $k > 1$, and equation 6 holds for a small ε such as $\varepsilon = 10^{-6}$. Otherwise, go to (2).

The idea behind the multistage methods reported in the next section is that an estimate of the partition matrix and the location of the cluster-centers can be obtained by clustering a sample of the data. There is however, a trade-off that relates to the sample size. A small sample is expected to produce a fast yet less reliable estimation of the cluster-centers. This leads to a multistage approach, which involves several stages of sampling (with replacement) of the data and estimating the membership matrix for the next stage. The size of the first sample should be as small as possible. On the other hand it should be statistically significant [24]. Each of the stages includes more objects from the data and sets the initial partition matrix of stage *I* according to the final partition matrix of stage *I* – 1.

3.2. The Pyramid FCM and Other Multistage FCM Algorithm

The pyramid procedure consists of the following stages: In the first stage, FCM is applied to an initial under-sampling of the data. The partition matrix for the first stage is initialized by any of the traditional methods for partition matrix initialization e.g., random initialization or uniform distribution initialization [6, 7]. FCM is performed on the sample until it converges. In stage *I* the original data-set is re-sampled with twice the resolution of stage *I* – 1. The membership matrix obtained in stage *I* – 1 is used to initiate the C-means clustering of the stage *I*.

The procedure is repeated until a resolution of 1: 1 is reached. At this stage, the C-means algorithm is performed on the entire input data and the resultant membership matrix, which implies cluster assignments and centers, is reported.

Every stage of the pyramid C-means increases the resolution by a factor of two. Since the estimation of the membership matrix for the clustering in every stage (excluding the first stage) is based on previous stage results, and utilizes an increasing sample of the data, it is expected to be closer to the final value of the membership matrix obtained from the complete data-set. This decreases the total number of weighted iterations (iterations are weighted by the number of data elements executed) required for reaching stability. In addition, since the under-samplings are smaller than the original data-set, each low-resolution stage of the FCM requires fewer computations. Therefore, it is expected that the under-sampling method would decrease the computational cost of FCM clustering.

For the pyramid execution we assume the availability of a procedure *f cm*() which implements the base-line FCM algorithm (Algorithm 1) described above. Let: $Pcm(c, X^{(I)}, U_{in}^{(I)}) = U_{out}^{(I)}$ be a procedure with the following inputs (1) the number of clusters (c), (2) an input set of vectors ($X^{(I)}$), and (3) an input partition matrix of $X^{(I)}$, ($U_{in}^{(I)}$). The function implements the algorithm describes in section 3.1 and outputs the final partition matrix ($U_{out}^{(I)}$). In this notation, (I) stands for stage *I* in the multistage algorithm. The pyramid algorithm, using *f cm*() is described next.

The two-stage and two-phase algorithms, described in section 2.2, are similar to the pyramid FCM algorithm. The main differences are the number of stages, the down sampling resolution at each stage, the exponential versus linear accumulation of samples, and the sampling method (i.e., sampling with, or without, replacement).

4. EXPERIMENTS AND RESULTS

4.1. Experimental Setup

The FCM, two-stage FCM, two-phase FCM, and pyramid FCM are applied numerous times to different data-sets, using

Algorithm-2 Pyramid FCM

1. Parameters:
   - *a.* X-the entire set of input vectors.
   - *b.* c - the number of partitions
   - *c.* I - the stage number
   - *d.* $X^{(I)}$ - the set of input vectors at stage I.
   - *e.* $|X^{(I)}|$ - the cardinality of the set $X^{(I)}$
   - *f.* $R^{(I)}$– an under-sampling index at stage I.
   - *g.* z– an initial under sampling rate (e.g., 128).
   - *h.* $U_{in}^{(I)}$ - the initial partition matrix at stage I
   - *i.* - the final partition matrix at stage I.
   - *j.* $J_q^{(I)}$ – the final clustering quality at stage I.
2. Set $I = 0$, set $R^{(0)} = z$, set $X^{(0)}$ to be the set obtained from X by randomly choosing every $R^{(0)}$ element from X, and choose an initial partition matrix
3. At stage $I \geq 0$:
   - a. Call *f cm*
   - b. Compute $\Omega^{(I)}$ using $U_{out}^{(I)}$ according to equation 4
   - c. Set
   - d. Set $R^{(I+1)} = \min(|X|, 2 \cdot R^{(I)})$
   - e. Set $X^{(I+1)}$ to be the set obtained from X by randomly choosing every $R^{(I+1)}$ element from X
   - f. Set $I = I + 1$
4. If $R^{(I)} < |X|$ then got to stage (2).
5. If $R^{(I)} = |X|$ then
   - a. call f cm (
   - b. Compute Ω using according to equation 4
   - c. Output Ω and
   - d. Stop

different parameters. Two sets of data are used for the experiments performed; the first set includes synthetic data with known centers and known distribution. The second set consists of the Red, Green, and Blue (RGB) components of color images used for color quantization. The experiments compare and contrast the performance of the FCM and the multistage variants. Three types of output data/results are collected: (1) Execution time and solution quality (i.e., inverse of distortion), (2) the results of a Binomial Monte Carlo validity testing of assertions concerning the execution time and quality, and (3) Records of convergences (i.e., distortion per iteration).

The execution time is approximated through a weighted number of iterations. Since there is almost no overhead in the down-sampling procedure, then a single multistage FCM iteration applied to a sample which contains $\frac{1}{z}$ of the data points takes about the same amount of time as of a traditional FCM iterations performed on the entire data-set. Thus, for the multistage FCM, the weighted number of total iterations N_p is given by: Where, p is the total number of stages, $N^{(I)}$ is the number of iterations at stage I of the pyramid algorithm, and $R^{(I)}$ is the sampling rate at stage I. The termination condition in all the experiments is a fixed number of iterations (set to 150) or a "short circuit" related to a negligible change in the first derivative of the distortion [30]. In most of the experiments the "short circuit" is encountered before the maximum number of iterations is reached.

The objective function (J_q) affects the partition matrix obtained by the FCM, the MMSE of hard clustering, the rate distortion function, and the dispersion. For this reason, in this research, we use the objective function (EQ-1) to assess the quality of FCM procedures. Since J_q is a distortion measure, then low values of J_q denote high clustering quality.

A Binomial Monte Carlo validity testing of assertions concerning the results of execution time and distortion is implemented. The results of each experiment are translated into "success" or "failure" of basic assertions or hypothesis related to validity. Analysis of the execution time and solution quality is accomplished by computing histograms showing the distribution of execution time and distortion.

4.1.1. Synthetic Data

A set of C ($4 \leq C \leq 32$) random cluster centers with m vectors per cluster ($4096 \leq m \leq 16384$) is generated. The vectors within a cluster are distributed with variances of 0.01 to 0.05 according to a normal distribution around the center. These patterns are used for the experiments with the synthetic data. The first level of under-sampling used in the experiments varies from $\frac{1}{128}$ to . This supplies an average of 128-512 elements per cluster and is above the number of elements per partition which are required in order to guarantee a tight and acceptable 95% confidence interval. In this research, an acceptable confidence interval lies above 0.5 [24].

4.1.2. Monte Carlo Analysis of Experiments with Synthetic Data

An extensive set of experiments using synthetic data is conducted and performance is recorded. In addition, the experiments are designed to enable Monte Carlo analysis for validation of assertions about the relations between the traditional FCM and the multistage FCM variants (two-stage, two-phase, and pyramid). Overall, thousands of experiments are performed. In general, each set of experiments is used for accepting or rejecting one assertion. The validation of an assertion is done using Binomial Monte Carlo procedure. Under this approach, the "success" of an experiment is defined in tandem with an assertion. The estimated probability of success in a set of experiments and the exact 95% confidence interval are used to assess the significance of the results which implies the validity of the assertion. Some of the assertions, however, may require hundreds of experiments in order to ensure a tight and acceptable 95% confidence interval.

Generally, each set of experiments is divided into groups of 100 experiments per group. The difference between experiments within a group is due to the fact that the initial partition matrices are randomly initiated, the sub sampling is random, and due to the fact that initial sub-sampling of the multistage FCM variants is selecting different sub-samples. Within a set of experiments only one element of the experiment is changing. For example, one set of experiments is used to compare the performance of the multistage FCM variants to traditional FCM where the number of clusters synthetically generated is 16, and the number of centers sought by the FCM procedure, is 8.

The following assertions are examples of assertions that are validated or rejected through the Monte Carlo experiments:

1. INS> IS; where INS denotes the number of weighted iterations with the traditional FCM, where no sampling is applied, and IS denotes the number of weighted iterations under one of the multistage FCM variants. Hence, if this assertion is true then the number of weighted iterations required for convergence of the multistage FCM variant is smaller than the number of weighted iterations required for convergence of the traditional FCM
2. DNS < DS: where D stands for distortion (. Hence, if this assertion is true, then the distortion obtained by the traditional FCM is smaller than the distortion produced by the multistage FCM variant. Thus, the quality of the traditional FCM clustering is higher than the clustering quality of the multistage variant.

4.1.3. Color Quantization

The problem of color quantization can be stated in the following way: given an image with N different colors, choose *C<<N* colors such that the resulting C-color image is the least distorted version of the original image [19, 40, 47]. Color quantization can be implemented by applying the FCM clustering procedure to the image-pixels where each pixel represents a vector in some color representation system. Nevertheless, the FCM produces a fuzzy assignment of clusters to centers, while the quantization requires a crisp assignment of patterns to colors. For this end, the final partition matrix is defuzzified and each pattern is assigned to one cluster e.g., the nearest cluster. For example, the clustering can be performed on the three-dimensional vectors formed by the red, green, and blue (RGB) color components of each pixel in the image. After clustering and defuzzification, each three-dimensional vector (pixel) is represented by the cluster-number to which the vector belongs, and the cluster centers are stored in a color-map. The C-value image along with the color-map is a compressed representation of the N-colors, original image. The compressed image can be used to reconstruct the original three-dimensional data-set by replacing each cluster-number by the centroid associated with the cluster. In the case of k=64 with 8 bit per color component, the original 24 bit per pixel image is represented by a 6 bits per pixel image along with a small color map. Hence, about 4 times compression is achieved.

4.1.4. Monte Carlo Analysis of Experiments with Color Quantization

Several RGB images are used to assess convergence rate and the distortion (J_q) of the traditional FCM and the FCM multistage variants. As in the case of synthetic data, we have performed an extensive set of experiments to enable a Monte Carlo analysis of assertions about the algorithms. For example, one set of tests is performed on the image Lena with *C=8* (*C* is the number of clusters sought). Another set of experiments uses *C=16*. Similarly, *C=8* and *C=16* are used with other images too.

4.2. Experimental Results

4.2.1. Experiments with Synthetic Data

This section provides detailed results of **one** set of experiments, and general results of the entire set of experiments where the FCM, two-stage FCM, two-phase FCM, and pyramid FCM (these last three methods are referred to as the multistage FCM variants) have been applied to 2-dimensional synthetic data. The data is generated by randomly selecting 16 centers within the unit square [0 1; 0, 1] and randomly distributing 16384 samples, with two dimensional normal distribution, around these centers. The variance for each center is randomly selected to be in the range (0.01, 0.05). As a result, the boundaries between clusters are not crisp. The initial quantization level is $\frac{1}{32}$.

The traditional FCM and its multistage variants, with C = 16, are run 100 times with different sets of random patterns, produced as described above, and different random selection of initial cluster centers. In each run the maximum number of iterations is 150. A short circuit termination condition stops the run if the change in the derivative of the distortion measure (J_q) is below a small threshold.

Figure 1 shows a histogram with the distribution of the number of weighted iterations for the traditional FCM and for each of the FCM multistage variants. The figure shows that the number of weighted iterations obtained in most of the runs of the pyramid FCM are located below 35 (i.e., these runs converge in less than 35 weighted iterations) while most of the runs of the traditional FCM require more than 75 iterations before convergence. The number of weighted iterations for the two-stage and two-phase FCM are centered in the middle of the histogram with no significant difference between these two variants. The two-phase, however, slightly outperforms the two-stage variant. Consequently, Figure 1 demonstrates that pyramid FCM has higher potential for speed up than two-phase FCM which has higher potential for speedup than the two-stage FCM. Overall, the same trend is apparent in the results of all of the other sets of data, where the distribution of number of iterations for the traditional FCM is at the high end of the histogram, the distribution for the pyramid variant is at the low end of the histogram, and the distribution of the other two variants is at the mid-range.

Figure 2a shows the average speedup relative to the traditional FCM obtained with every multistage variant of the FCM. The figure demonstrates that pyramid FCM provides a speed-up of about 2.75X, while two-phase FCM provides a speedup of 2.4X, and two-stage FCM provides a speedup of 2X. Again, the pyramid FCM outperforms the other multistage variants. Overall, the same trend is apparent in the results of all of the other sets of data where pyramid FCM provides a speedup of 2X to 3.5X , two-stage FCM provides a speedup that is close to, and generally slightly lower than 2X, and the two-phase algorithm provides a speedup that is slightly above 2X.

Figure 2b shows the average values of the distortion obtained with the traditional FCM and its multistage variants. The quality of the traditional FCM and different multistage variants is very similar. The quality is assessed via the FCM objective function: Since J_q is a distortion measure, then low values of J_q denote high clustering quality. In this specific case the pyramid approach outperforms the rest of the multistage variants, but the difference is not significant (< 2%). Overall, after running numerous experiments and obtaining about the same quality (distortion) from many different experiments we conclude that almost all of the runs provide a solution that is very close to the global optimum. Hence, in this case, we cannot demonstrate a significant improvement in quality due to the multistage approach.

Figure 3 shows the convergence rate for the traditional FCM and the multistage variants for one out of the 100 experiments. The x-axis represents the number of weighted iterations required for convergence and the y-axis shows the distortion in each of the weighted iterations. The discontinuities in the curves of the multistage variants are due to a "jump" in distortion that occurs when moving from one stage to the next where centers from previous stage are used as seed for the next stage. Overall, the pyramid approach has the best convergence rate and converges to the lowest value. A similar trend is observed in the rest of the experiments.

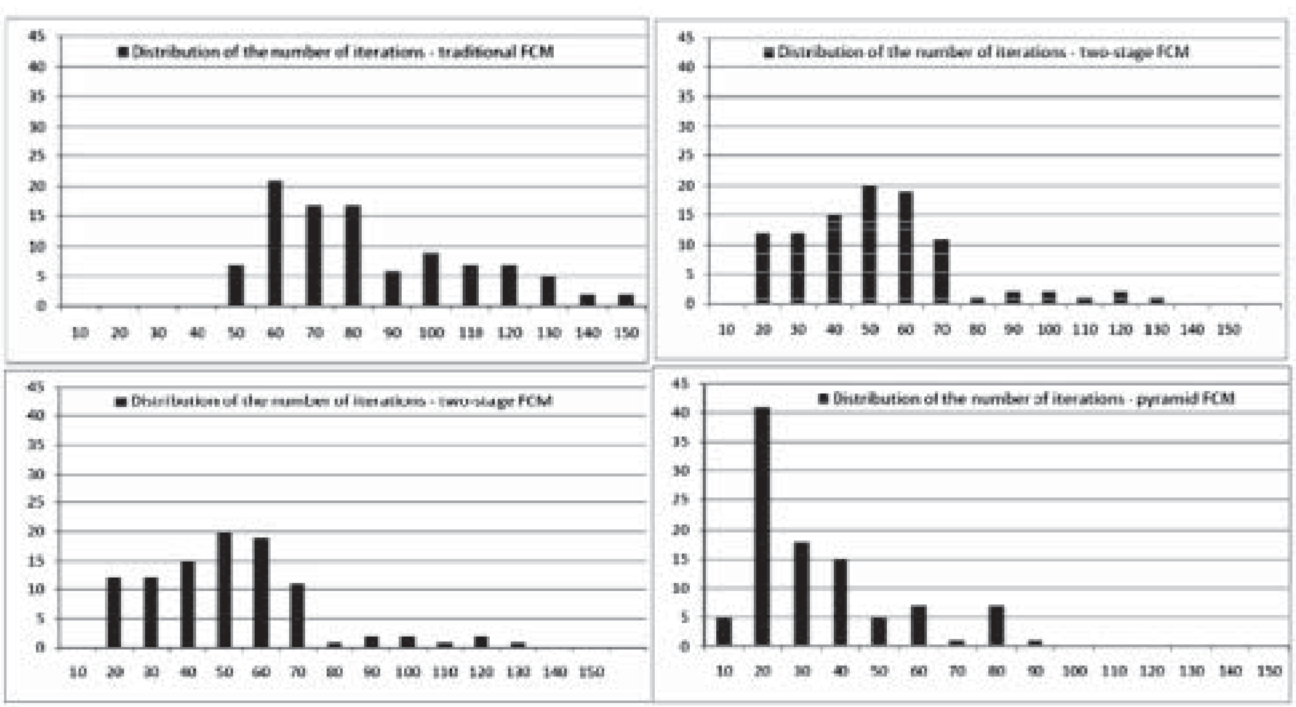

Figure 1: Histograms of the Distribution of the Number of Iterations (Synthetic Data)

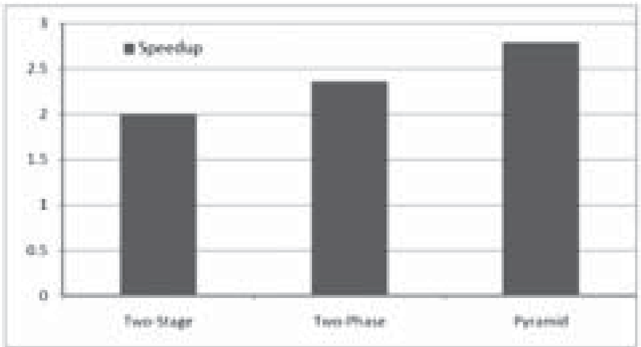

Figure 2a: Average Speedup (Synthetic Data)

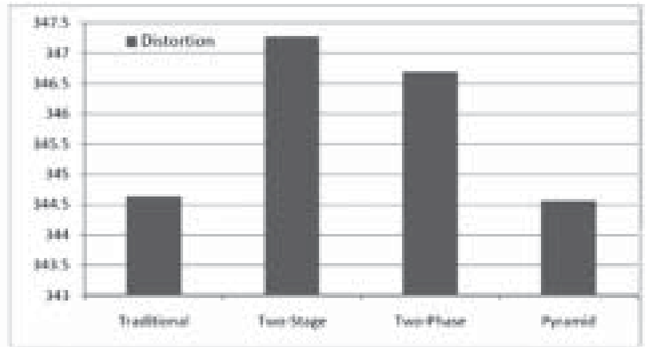

Figure 2b: Average Distortion (Synthetic Data)

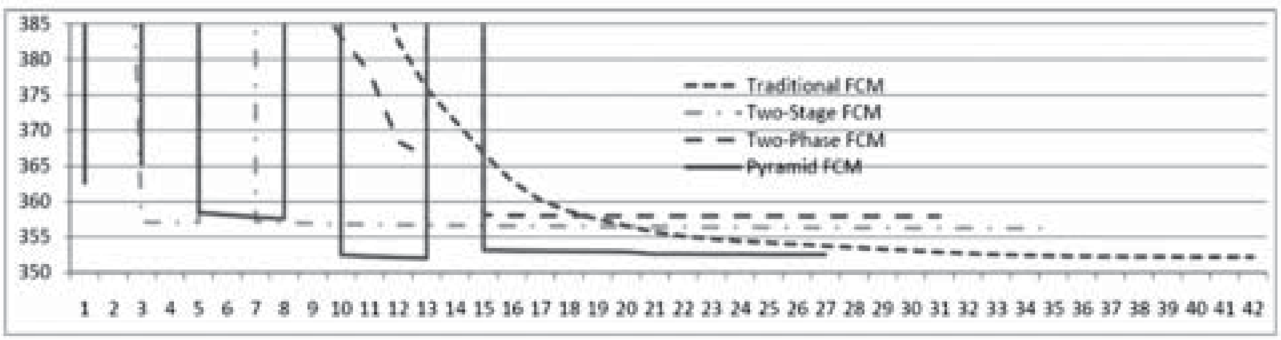

Figure 3: Convergence Rate of the FCM Algorithms (Synthetic Data)

Table 1 summarizes the results of Monte Carlo validation for this set of experiments. The assertion tested for the 100 members of this set of experiment and analyzed using exact estimation for the confidence intervals of binomial distribution are: (1) *INS* > *IS*; where *INS* denotes the number of weighted iterations with no sampling, 2) *INS* > 2 × *IS*, 3) *INS* > 3 × *IS*, 4) *INS* > 4 × *IS*, 5) *DNS* > *DS*; where *D* stands for distortion (J_q), and 6) *DNS* < *DS*. The table shows the number of successes in the binomial tests as well as the exact 95% confidence interval. It can be observed that in the case of pyramid FCM, assertions 1, and 2, hold while assertions 3, 4, 5, and 6 fail (a part of the confidence interval is below 0.5).

This further validates the speedup results shown in figure 2a and the distortion results shown in figure 2b. In addition, the table shows that assertions 1 is the only assertions that hold true for the two-stage and two-phase approaches, and the rest of the assertions fail. Again, this is consistent with the results depicted in figures 2a and 2b. In specific, the fact that both assertion 5 and 6 fail for all of the experiments shows that there is no significant difference in the distortion obtained from the different algorithms. Similar and consistent results are obtained with other sets of experiments. The conclusion from the observations obtained from experiments with synthetic data is that the pyramid FCM can be used to extend the speedup obtained with the other multistage variants, while providing the same or slightly better quality.

4.2.2. Experiments with Color Quantization

This section provides detailed results of **one** set of experiments, and general results of the entire set of experiments where the FCM, two-stage FCM, two-phase FCM, and pyramid FCM have been applied to color Images. The image used for this set is an RGB version of Lena with resolution of 512 × 512 pixels. The initial quantization level

Table 1
Summary of Monte Carlo Validation Tests (Synthetic Data)

Assertion / FCM Variant	*INS > IS*	*INS > 2 × IS*	*INS > 3 × IS*	*INS > 4 × IS*	*DNS > DS*	*DNS < DS*
Two-Stage	86 [0.78,0.92]	58 [0.48,0.68]	41 [0.31,0.51]	27 [0.19,0.39]	47 [0.37,0.57]	53 [0.43,0.63]
Two-Phase	99 [0.95,0.99]	57 [0.47,0.67]	25 [0.17,0.35]	9 [0.00,0.16]	46 [0.36,0.56]	54 [0.44,0.64]
Pyramid	95 [0.89,0.98]	76 [0.76,0.84]	58 [0.48,0.68]	51 [0.41,0.61]	57 [0.47,0.67]	43 [0.33,0.53]

is $\frac{1}{32}$. The traditional FCM and its multistage variants (two-stage FCM, two-phase FCM, and pyramid FCM), with, C = 16 are run 100 times with different random selection of initial cluster centers. In each run the maximum number of iterations is 150. A short circuit termination condition stops the run if the change in the derivative of the distortion measure (J_q) is below a small threshold.

Figure 4 shows a histogram with the distribution of the number of weighted iterations for the traditional FCM and for all the FCM multistage variants. The figure shows that the number of weighted iterations obtained in most of the runs of the pyramid FCM is located below 35 while most of the runs of the traditional FCM require more than 100 iterations before convergence. The number of weighted iterations for the two-stage and two-phase FCM is centered in the middle of the histogram with no significant difference between the two multistage variants. As in the case of synthetic data, the two-phase variant slightly outperforms the two-stage variant. Overall, the same trend is apparent in the results of all of the other sets of images.

Figure 5a shows the average speedup relative to the traditional FCM obtained with every multistage variant of the FCM. The figure demonstrates that pyramid FCM provides a speed-up of about 3.1X while two-phase FCM provides a speedup of 2.6X and two-stage FCM provides a speedup of 1.9X. Again, the pyramid FCM outperforms the other FCM multistage variants. Overall, the same trend is apparent in the results of other sets of images where pyramid FCM provides a speedup of 2X to 4X, two-stage FCM provides a speedup that is between 1.5X to 3X, and the two-phase algorithm provides a speedup that is slightly better than the two-stage speedup. Figure 5b shows the average values of the solution quality obtained with the traditional FCM and its multistage variants. The quality (inverse of distortion) of the traditional FCM and different multi stage variants is very similar with a difference of less than 1% between variants. Nevertheless, the pyramid approach outperforms the rest of the approaches. Again, after running numerous experiments with different images and obtaining about the same quality from many different experiments (when applied to the same image) we conclude that almost all of the runs provide a solution that is very close to the global optimum, and we cannot demonstrate a significant improvement in quality due to the multistage approach.

Figure 6 shows the convergence rate for the different multistage variants for one out of the 100 experiments. The x-axis shows the number of weighted iterations required for convergence and the y-axis shows the distortion in each of the weighted iterations. The discontinuities in the curves of the multistage variants are due to a "jump" in distortion that occurs when moving from one stage to the next; where centers from previous stage are used as seed for the next stage. Overall, the pyramid approach has the best convergence rate and converges to the lowest value. A similar trend is observed in the rest of the experiments in this and other sets of experiments.

Table 2 summarizes the results of Monte Carlo validation for this set of experiments. The assertion tested

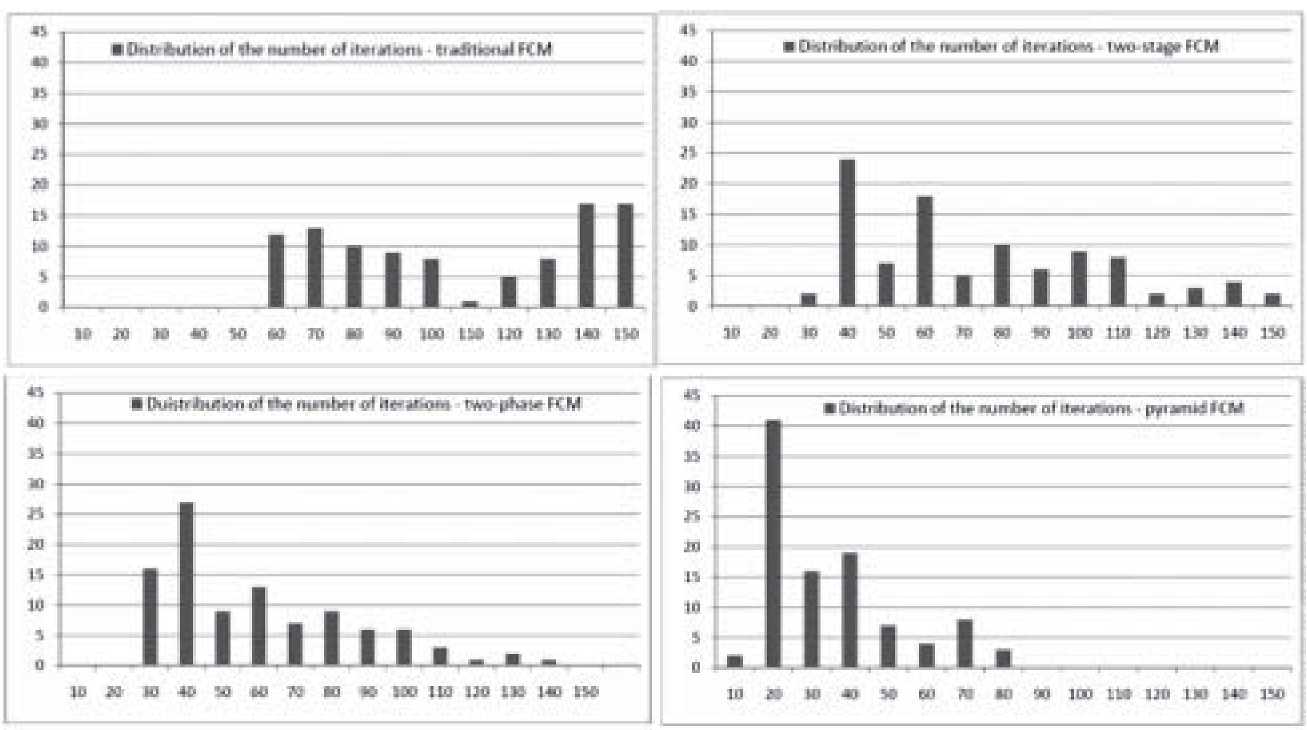

Figure 4: Histograms of the Distribution of the Number of Iterations (Lena)

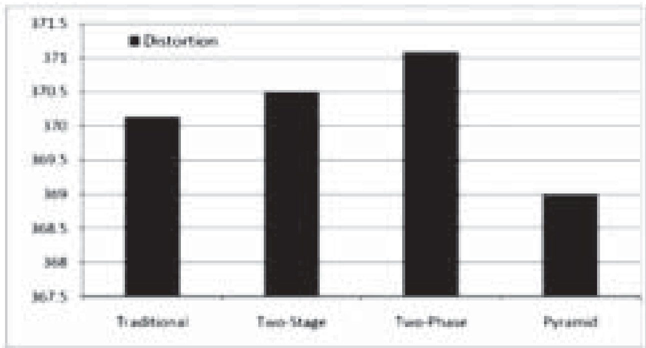

Figure 5a: Average Speedup (Lena)

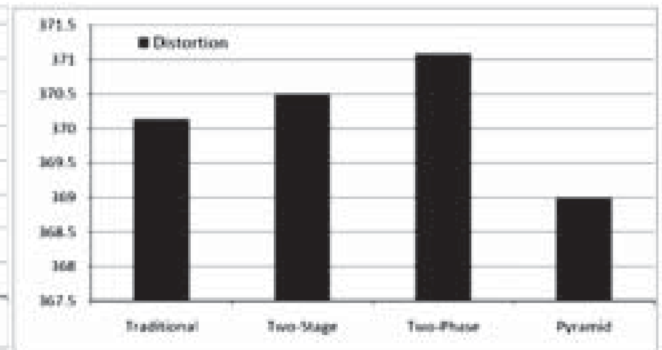

Figure 5b: Average Distortion (Lena)

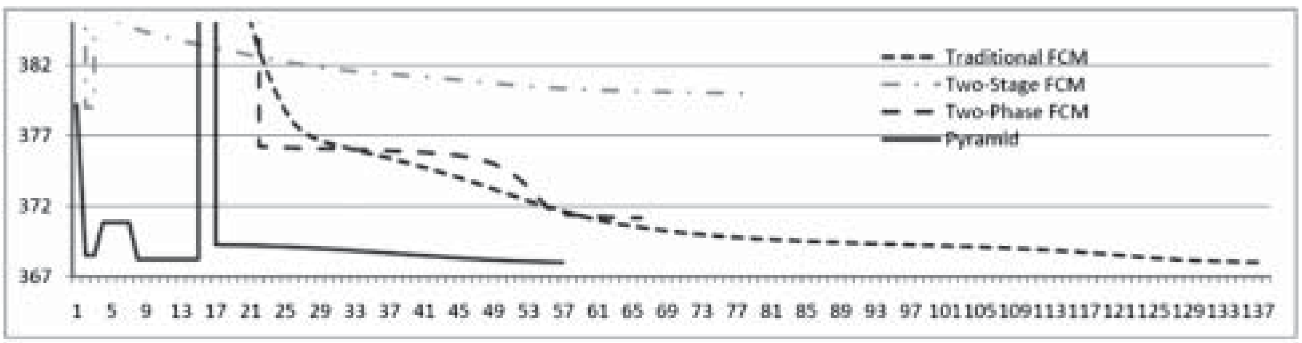

Figure 6: Convergence Rate of the FCM Algorithms (Lena)

Table 2
Summary of Monte Carlo Validation Tests (Lena)

Assertion / FCM Variant	*INS > IS*	*INS > 2 × IS*	*INS > 3 × IS*	*INS > 4 × IS*	*DNS > DS*	*DNS < DS*
Two-Stage	87 [0.78,0.93]	60 [0.50,0.70]	41 [0.31,0.52]	37 [0.78,0.47]	43 [0.33,0.53]	57 [0.65,0.83]
Two-Phase	93 [0.86,0.97]	53 [0.43,0.63]	24 [0.16,0.34]	10 [0.05,0.18]	56 [0.46,0.66]	44 [0.34,0.54]
Pyramid	97 [0.91,0.99]	82 [0.73,0.89]	69 [0.59,0.78]	58 [0.47,0.68]	69 [0.59,0.78]	31 [0.22,0.41]

for the 100 members of the set of experiment and analyzed using exact confidence levels for binomial distribution are: 1) *INS* > *IS*, 2) *INS* > 2 × *IS*, 3) *INS* > 3 × *IS*, 4) *INS* > 4 × *IS*, 5) *DNS* > *DS*, and 6) *DNS* < *DS*. The table shows the number of successes in the binomial tests as well as the exact 95% confidence interval. It can be observed that in the case of pyramid FCM, assertions 1, 2, 3, and 5, hold while assertions 4 and 6 fail (a part of the exact confidence interval is below 0.5). In addition, the table shows that assertions 1 and 2 are the only assertions that hold true for the two-stage approach, and assertion 1 is the only assertion that holds true for the two-phase approach. Again, this is consistent with the results depicted in figures 5a and 5b. Similar and consistent results are obtained with other sets of experiments. Finally, the conclusion for the observations obtained from experiments with color image data is that the pyramid FCM can be used to extend the speedup obtained with the other multistage variants, while providing the same or slightly better quality.

5. CONCLUSIONS

The quality of clustering and the computational cost of the FCM depend on the initialization of the partition matrix. The multistage FCM variants set the matrix using data-samples. Consequently, these methods can provide improvement in convergence rate. The results show that the multistage FCM can be tuned to improve convergence rate without significant impact on quality. Of the multistage FCM variants, the pyramid approach is the most cost effective. We were not able to demonstrate a significant improvement in clustering quality. Nevertheless, in some cases, it is conceivable that the saving in time due to the multistage framework can be translated into better quality via multiple runs of the multistage procedure.

We plan to expand this research and investigate a general framework for multistage approach in optimization algorithms including fuzzy ISODATA and expectation maximization. In addition, we plan to investigate a hybrid of the linear sampling of the two-phase approach with the exponential sampling of the pyramid FCM variant. Furthermore, we plan to investigate the utility of methods for merging the results of different runs of clustering in parallel multistage FCM and large data FCM applications. Finally, we plan to use the validation technique developed in this research to compare the performance of algorithm from different domains; for example, K-means versus FCM.

REFERENCES

[1] Alippi, C. and Cucchiara, R., Cluster Partitioning in Image Analysis Classification: A Genetic Algorithm Approach. *Proceedings of the Computer Systems and Software Engineering*, 1992: 139-44.

[2] Altman, D., Efficient Fuzzy Clustering of Multi-spectral Images, in *IEEE FUZZ*. 1999.

[3] Anderson, D. T., Luke, R. H., and Keller, J. M., Speedup of Fuzzy Clustering Through Stream Processing on Graphics Processing Units. *IEEE Transactions on Fuzzy Systems*, 2008, **16**(14): 1101-1106.

[4] Ball, G. H. and Hall, J. D., *ISODATA, A Novel Method of Data Analysis and Pattern Classification*, in *International Communication Conference*. 1966: Philadelphia.

[5] Bensaid, A., *et al.*, *Partially Supervised Clustering for Image Segmentation*, Pattern Recognition. 1996, **29**(5): 859-871.

[6] Bezdek, J. C., A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1980, **2**(1): 1-8.

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

[8] Bradley, P. S., Usama, F., and Corey, R. C. *Scaling Clustering Algorithms to Large Database*. in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. 1998.

[9] Cannon, R. J., Dave, J., and Bezdek, J. C., Efficient Implementation of the Fuzzy C-Means Clustering Algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1986, **8**: 248-255.

[10] Cheng, T. W., Goldgof, D. B., and Hall, L. O., Fast Fuzzy Clustering. *Fuzzy Sets and Systems*, 1998, **93**: 49-56.

[11] Coleman, G. and Andrews, H., Image Segmentation by Clustering. *Proceedings of the IEEE*, 1979, **67**(5): 773-785.

[12] El-Gamal, A. A., Using Stimulated Annealing to Design Good Codes. *IEEE Transactions on Information Theory*, 1987, **33**(1): 116-123.

[13] Equitz, W. H., A New Vector Quantization Clustering Algorithm. *IEEE Transactions on Acoustics Speech and Signal Processing*, 1989, **37**: 1568-1575.

[14] Eschrich, S., *et al.*, Fast Accurate Fuzzy Clustering through Data Reduction. *IEEE Transactions on Fuzzy Systems*, 2003, **11**(2): 262-270.

[15] Farnstrom, F., Lewis, J., and Elkan, C. *Scalability for Clustering Algorithms Revisited*. in *ACM SIGKDD Explorations*. 2000.

[16] Fraser, A., Simulation of Genetic Systems by Automatic Digital Computers. *Information Theory*, 1957: 484-491.

[17] Garey, R. M., Johnson, S., and Witsenhausen, H. S., The Complexity of the Generalized Lloyd-Max Problem. *IEEE Transactions on Information Theory*, 1972, **28**: 255-256.

[18] Gupta, C. and Grossman, R. *GenIc: A Single Generalized Incremental Algorithm for Clustering*. in *The Fourth (SIAM) International Conference on Data Engineering*. 2004.

[19] Heckbert, P., *Color Image Quantization for Frame Buffer Display*. ACM Transactions on Computer Graphics, 1982. **16**(3): 297-307.

[20] Höppner, F., *et al.*, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. 1999, New York: Wiley.

[21] Hore, P., Hall, L. W., and Goldgof, D. B., *Speedup of Fuzzy Clustering Through Stream Processing on Graphics Processing Units*, in *IEEE International Conference on Fuzzy Systems*. 2007 London.

[22] Hsiao, Y. J. and Sawchuk, A. A., *Unsupervised Textured Image Segmentation Using Feature Smoothing and Probabilistic Relaxation Techniques.* Computer Vision, Graphics, and Image Processing, 1989, **48**: 1-21.

[23] Huang, J. F., *Image Segmentation Using Discrete Cosine Transform and K-Means Clustering Algorithm*, in *Computer Science*. 1991, Florida Institute of Technology: Melbourne, Florida.

[24] Jain, A. K. and Dubes, R. C., *Algorithms for Clustering Data*. 1988, Englewood Cliffs: Prentice Hall.

[25] Kandel, A., *Fuzzy Mathematical Techniques with Applications*. 1987, New York: Addison Wesley.

[26] Kasif, S. and Rosenfeld, A., Pyramid Linking is a Special Case of ISODATA. *IEEE Transactions on Systems, Man, and Cybernetics*, 1983. **13**(1): 84-85.

[27] Kohonen, T., *et al.*, *Artificial Networks*. 1991, Amsterdam: Elsevier.

[28] Kolen, J. F. and Hutcheson, T., Reducing the Time Complexity of the Fuzzy C-Means Algorithm. *IEEE Transactions on Fuzzy Systems*, 2002, **3**: 263-267.

[29] Lin, J., *et al.*, *Iterative Incremental Clustering of Time Series.* Advances in Database Technology - EDBT 2004, 2004: 106-122.

[30] Linde, Y., Buzo, A., and Gray, R., An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, 1980, **28**(1): 84-95.

[31] Lloyd, S. P., Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 1982, **25**(2): 129-137.

[32] Lung, H. V. and Kim, J. M., *A Generalized Spatial Fuzzy C-means Algorithm for Medical Image Segmentation*, in *Proceedings of the 18th International Conference on Fuzzy Systems*. 2009: Jeju Island, Korea.

[33] Macqueen, J., *Some Methods for Classification and Analysis of Multivariate Data*. 5th Berkeley Symposium on Probability and Statistics, 1967.

[34] Pal, N. R. and Bezdek, J. C., Complexity Reduction for "Large Image" Processing. *IEEE Transactions on Systems, Man, and Cybernetics*, 2002. **32**(5): 598-611.

[35] Pal, S.K. and Majumdar, D., *Fuzzy Mathematical Approach to Pattern Recognition*. 1986, New York: Wiley.

[36] Petrosini, A. and Verde, M. *P-AFLC: a Parallel Scalable Fuzzy Clustering Algorithm*. in *International Conference on Pattern Recognition*. 2004.

[37] Rahmi, S., *et al.*, *A Parallel Mean Algorithm for Image Segmentation*. Fuzzy Information, 2004. **1**: 234-237.

[38] Rosenfeld, A., *Pyramids: Multi-resolution Image Analysis*. *3rd Scandinavian Conference on Image Analysis*, 1983: 23-28.

[39] Stauton, R. C. and Ma, L., A Modified Fuzzy C means Image Segmentation Algorithm for use with Uneven Patterns. *Pattern Recognition*, 2007, **40**: 2005-3011.

[40] Swagatam, D. and Amit, K., Automatic Image Pixel Clustering with an Improved Differential Evolution. *Applications of Soft Computing*, 2009, **9**(1): 226-236.

[41] Tamir, D.E., *Cluster Validity of Multi Resolution Competitive Neural Networks. International Conference on Artificial Intelligence*, 2007.

[42] Tamir, D. E., Park, C., and Yoo, B., *The Validity of Pyramid K-means*, in *SPIE Conference on Optics and Photonics / Optical Engineering and applications*. 2007: San Diego.

[43] Tilton, J. C., *Multi-resolution Spatially Constrained Clustering of Remotely Sensed Data on the Massively Parallel Processor. IGARSS Symposium*, 1984: 661-666.

[44] Tou, J. and Gonzalez, R.C., *Pattern Recognition Principles*. 1974, London: Addison-Wesley.

[45] Xu, X., Jager, J., and Kriegel, H.P., A Fast Parallel Clustering Algorithm for Large Spatial Databases. *Data Mining and Knowledge Discovery*, 1999, **3**(3): 263-290.

[46] Yager, R. R., *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*. 1987, Ney York: Wiley.

[47] Zhu, Y., *Image Segmentation for Color Separation* in *Computer Science*. 1991, Florida Institute of Technology: Melbourne, Florida.