Particle swarm optimization method for image clustering

Article *in* International Journal of Pattern Recognition and Artificial Intelligence · May 2005

DOI: 10.1142/S0218001405004083 · Source: DBLP

Mahamed Omran
Gulf University for Science and Technology (Kuwait)

Andries Engelbrecht
University of Pretoria

Ayed A. Salman
Kuwait University

Particle Swarm Optimization Method for Image Clustering

M Omran[a], AP Engelbrecht[a], A Salman[b]

[a]Department of Computer Science, University of Pretoria, South Africa

mjomran@engineer.com, engel@driesie.cs.up.ac.za

[b]Computer Engineering Department, Kuwait University, Kuwait

ayed@eng.kuniv.edu.kw

Abbreviated title: PSO for Image Clustering

Abstract

An image clustering method that is based on the particle swarm optimizer (PSO) is developed in this paper. The algorithm finds the centroids of a user specified number of clusters, where each cluster groups together similar image primitives. To illustrate its wide applicability, the proposed image classifier has been applied to synthetic, MRI and satellite images. Experimental results show that the PSO image classifier performs better than *state-of-the-art* image classifiers (namely, K-means, Fuzzy C-means, K-Harmonic means and Genetic Algorithms) in all measured criteria. The influence of different values of PSO control parameters on performance is also illustrated.

Keywords: Image Clustering, Particle Swarm Optimization, Pattern Recognition, Remote Sensing, Spectral Domain

1 Introduction

Image classification is the process of identifying groups of similar image primitives [38]. These image primitives can be pixels, regions, line elements, etc. depending on the problem encountered. Many basic image processing techniques such as quantization, segmentation and coarsening can be viewed as different instances of the clustering problem [38].

There are two main approaches to image classification: supervised and unsupervised. In the supervised approach, the number and the numerical characteristics (e.g. mean and variance) of the classes in the image are known in advance (by the analyst) and used in the training step, which is followed by a classification step. There are several popular supervised algorithms such as the minimum-distance-to-mean, parallelepiped and the Gaussian maximum likelihood classifiers [46]. For unsupervised approaches, classes are unknown and the approach starts by partitioning the image data into groups (or clusters), according to a similarity measure, which can be compared by an analyst to available reference data [29]. Therefore, unsupervised classification is also referred to as a clustering problem. In general, the unsupervised approach has several advantages over the supervised approach [10], namely

- For unsupervised approaches, there is no need for an analyst to specify in advance all the classes in the image data set. The clustering algorithm will automatically find distinct classes, which dramatically reduce the work of the analyst.
- The characteristics of the objects being classified can vary with time; the unsupervised approach is an excellent way to monitor these changes.
- Some characteristics of objects may not be known in advance. Unsupervised approaches will automatically flag these characteristics.

The focus in this paper is on the unsupervised approach (i.e. image clustering). There are several algorithms that belong to this class of algorithms. These algorithms can be categorized into two groups: hierarchical and partitional [15, 28]. For hierarchical clustering, the output is a tree showing a sequence of clustering with each cluster being a partition of the data set [28]. These type of algorithms have the following advantages:

- The number of clusters need not be specified *a priori*.
- They are independent of the initial condition.

However, they suffer from the following drawbacks:

- They are static, i.e. pixels assigned to a cluster can not move to another cluster.
- They may fail to separate overlapping clusters due to lack of information about the global shape or size of the clusters [15].

On the other hand, partitional clustering algorithms partition the data set into a specified number of clusters. These algorithms try to minimize certain criteria (e.g. a squared error function). Therefore, they can be treated as an optimization problem. The advantages of the hierarchical algorithms are the disadvantages of the partitional algorithms and *vice versa*.

The most widely used partitional algorithm is the iterative K-means approach. For K-means clustering, the algorithm starts with K cluster centers or *centroids* (initial values for the centroids are randomly selected or derived from *a priori* information). Then, each pixel in the image is assigned to the closest cluster (i.e. closest centroid). Finally, the centroids are recalculated according to the associated pixels. This process is repeated until convergence [14]. There are three drawbacks of this algorithm, namely

- the algorithm is data-dependent;

- it is a greedy algorithm that depends on the initial condition, which may cause the algorithm to converge to suboptimal solutions; and
- the user needs to specify the number of classes in advance [10].

ISODATA is an enhancement proposed by Ball and Hall [4] that operates on the same concept of the K-means algorithm with the addition of the possibility of merging classes and splitting elongated classes. A recent alternative approach to ISODATA is SYNERACT [20]. This approach combines both K-means and hierarchical descending approaches to overcome the three drawbacks mentioned above. Three concepts are used by SYNERACT:

- a hyperplane to split a cluster into two smaller clusters and compute their centroids,
- iterative clustering to assign the pixels into the available clusters, and
- a binary tree to store the clusters generated from the splitting process.

According to Huang [20], SYNERACT is faster than ISODATA, and almost as accurate as ISODATA. Furthermore, it does not require the user to specify the number of clusters and initial location of the centroids in advance. Another improvement to the K-means algorithm is proposed by Rosenberger and Chehdi [41], which automatically finds the number of clusters in the image set by using intermediate results.

A fuzzy version of K-means, called Fuzzy C-means (FCM), was proposed by Bezdek [28, 5]. FCM is based on a fuzzy extension of the least-square error criterion. The advantage of FCM over K-means is that, while in K-means the pixels are assigned to one and only one cluster (i.e. hard or *crisp* clustering), in FCM each pixel belongs to each cluster with some degree of membership (i.e. fuzzy clustering). This is more suitable for real applications where there are some overlap between the clusters in the data set. Gath and Geva proposed an algorithm based on a combination of fuzzy C-means and fuzzy maximum likelihood estimation [16]. Based on fuzzy clustering, Lorette *et al* optimizes an objective function which takes into consideration the entropy of the image partition using an iterative schema [30]. These approaches developed by Gath *et al* and Lorette *et al* do not require the user to specify the number of clusters in advance – they are fully unsupervised.

Another popular clustering algorithm is the Expectation-Maximization (EM) [7, 34, 40]. EM is a general algorithm for parameter estimation in the presence of some unknown data [17]. EM partitions the data set into clusters by determining a mixture of Gaussians to fit the data set. Each Gaussian has a mean and covariance matrix [1]. Zhang *et al* [51] used EM along with a hidden Markov random field (HMRF) for the segmentation of brain magnetic resonance (MR) images.

Results from [17, 18] showed that K-means performs comparably to EM. Furthermore, Aldrin *et al* [1] stated that EM fails on high-dimensional data sets due to numerical precision problems. They also observed that Gaussians often collapsed to delta functions [1].

Recently, Zhang *et al* [50, 49] proposed a novel algorithm called K-harmonic means (KHM) with promising results. In KHM, the harmonic mean of the distance of each cluster center to every data point is computed. The cluster centers are then updated accordingly. KHM is less sensitive to initial conditions (contrary to K-means) and does not have the problem of collapsing Gaussians exhibited by EM [1]. Experiments conducted by Zhang *et al* [50, 49] and Hamerly *el al* [17, 18] showed that KHM outperforms K-means, FCM (according to Hamerly [17]) and EM. Hamerly *et al* [17, 18] proposed a variation of KHM, called Hybrid 2 (H2), which uses the soft membership function of KHM and the constant weight function of K-means (refer to section 3.2.1) with encouraging results.

Another category of unsupervised partitional algorithms includes the non-iterative algorithms. The most widely used non-iterative algorithm is MacQueen's K-means algorithm [31]. This algorithm works in two phases as follows: the first phase finds the centroids of the classes, and the second classifies the image pixels. Competitive Learning (CL) updates the centroids sequentially by moving the closest centroid toward the pixel being classified [42]. These algorithms suffer the drawback of being dependent on the order in which the data points are presented. To overcome this problem, data points are presented in a random order [10]. Another common non-iterative approach for unsupervised clustering depends on image texture. These algorithms work by moving a window (e.g. 3×3 window) over the image, calculating the variance of the pixels within this window. If the variance is less than a pre-specified threshold, the mean of the pixels within this window is considered as a new centroid. This process is repeated until a pre-specified maximum number of classes is reached. The closest centroids are then merged until the entire image is analyzed. The final centroids resulting from this algorithm are used to classify the image [29]. In general, iterative algorithms are more effective than non-iterative algorithms, since they are less dependent on the order in which data points are presented.

Furthermore, Artificial Neural Networks (ANN) have been successfully applied to image clustering [32, 36]. Self-Organizing Maps (SOM) [27] has been used by Evangelou *et al* as a data mining tool in image clustering [12]. Different data mining techniques have been used by Antonie *et al* to detect tumors in digital mammography [2]. On the other hand, due to its ability to handle large search spaces with minimum information about the objective function, GAs have been used to develop GA-based clustering algorithms [13, 33]. Ramos and Muge used K-means clustering as a guide for a GA to search for the most appropriate clusters [39]. Furthermore, Scheunders proposed a GA-based K-means clustering algorithm for color quantization [42]. Vafaie *et al* and Bala *et al* used GAs for image feature selection. Recently, an approach to simulate the human visual system by modeling the blurring effect of lateral retinal interconnections based on scale space theory has been proposed by [28]. Finally, an approach that performs grouping directly on the histogram data without requiring data re-representation was proposed in [38].

Omran *et al* introduced a new PSO-based image clustering algorithm in [35]. This paper explores this algorithm in more detail, and presents an improved fitness function. A second PSO-based algorithm is implemented and compared with the standard PSO approach. The influence of PSO control parameters on performance is illustrated. It is shown in the paper that the PSO-based image clustering approaches perform better than *state-of-the-art* clustering approaches.

The rest of the paper is organized as follows: An overview of different PSO variations is presented in section 2. A detailed description of the algorithms used in the study is presented in section 3. Section 4 presents experimental results to illustrate the efficiency of the algorithm and studies the effects of the user defined parameters on the final product. Section 5 concludes the paper, and outlines future research.

2 Particle Swarm Optimization

Particle swarm optimizers (PSO) are population-based optimization algorithms modeled after the simulation of social behavior of birds in a flock [21, 25]. The PSO maintains a swarm of candidate solutions to the optimization problem under consideration. Each candidate solution is referred to as a particle. If the optimization problem has n variables, then each particle represents an n-dimensional point in the search space. The quality, or fitness, of a particle is measured using a fitness function. The fitness function quantifies how close a particle is to the optimal solution.

Each particle is flown through the search space, having its position adjusted based on its distance from its own personal best position and the distance from the best particle of the swarm [44]. The performance of each particle, i.e. how close the particle is from the global optimum, is measured using a fitness function which depends on the optimization problem.

Each particle i maintains the following information:

- $\mathbf{x}_i$, the *current position* of the particle;
- $\mathbf{v}_i$, the *current velocity* of the particle; and
- $\mathbf{y}_i$, the *personal best position* of the particle.

The personal best position associated with a particle i is the best position that the particle has visited so far, i.e. a position that yielded the highest fitness value for that particle. The personal best position therefor serves as a kind of memory. If f denotes the objective function, then the personal best of a particle at a time step t is updated as:

$$\mathbf{y}_i(t+1) = \begin{cases} \mathbf{y}_i(t) & \text{if } f(\mathbf{x}_i(t+1)) \geq f(\mathbf{y}_i(t)) \\ \mathbf{x}_i(t+1) & \text{if } f(\mathbf{x}_i(t+1)) < f(\mathbf{y}_i(t)) \end{cases} \tag{1}$$

One of the unique principles of PSO is that of information exchange between members of a swarm. This information exchange is used to determine the best particle(s), or position(s), in the swarm so that other particles can adjust their position toward the best ones. The first social topologies that were developed are the star and ring topologies [24]. The star topology allows each particle to communicate with every other particle. The effect of the star topology is that the best particle in the swarm is determined, and all particles move towards this global best particle. The resulting algorithm is generally referred to as the *gbest* PSO. The ring topology, on the other hand, defines overlapping neighborhoods of particles. Particles in a neighborhood communicate to identify the best in that neighborhood. All particles in a neighborhood then adjust toward to neighborhood best, or local best particle. The resulting algorithm is generally referred to as the *lbest* PSO. Recently, Kennedy and Mendes [26] investigated more complex social topologies, of which the Von Neumann topology showed to be an efficient alternative [37]. For the Von Neumann topology, neighborhoods are formed on a 2-dimensional lattice.

For the *gbest* PSO model, where the best particle is determined from the entire swarm, the best particle is

$$\hat{\mathbf{y}}(t) \in \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_s\} = \min\{f(\mathbf{y}_0(t)), f(\mathbf{y}_1(t)), \ldots, f(\mathbf{y}_s(t))\} \tag{2}$$

where s is the total number of particles in the swarm. For the *lbest* PSO model, neighborhoods are defined as

$$N_j = \{\mathbf{y}_{i-l}(t), \mathbf{y}_{i-l+1}(t), \ldots, \mathbf{y}_{i-1}(t), \mathbf{y}_i(t), \mathbf{y}_{i+1}(t), \ldots, \mathbf{y}_{i+l-1}(t), \mathbf{y}_{i+1}(t)\} \tag{3}$$

and the best particle in neighborhood N_j is

$$\hat{\mathbf{y}}_j(t+1) \in N_j \mid f(\hat{\mathbf{y}}_j(t+1)) = \min\{f(y_i)\}, \forall y_i \in N_j \tag{4}$$

Neighborhoods are usually determined using particle indices [24], although topological neighborhoods have also been used [45]. The *gbest* PSO is simply a special case of *lbest* with $l = s$; that is, the neighborhood is the entire swarm. While the *lbest* PSO has larger diversity than the *gbest* PSO, it is slower than *gbest* PSO. The rest of this paper concentrates on the faster *gbest* PSO.

For each iteration of a *gbest* PSO algorithm, $\mathbf{v}_i$ and $\mathbf{x}_i$ are updated as follows:

$$\begin{aligned} \mathbf{v}_i(t+1) &= w\mathbf{v}_i(t) + c_1\mathbf{r}_1(t)(\mathbf{y}_i(t) - \mathbf{x}_i(t)) + c_2\mathbf{r}_2(t)(\hat{\mathbf{y}}(t) - \mathbf{x}_i(t)) \qquad (5) \\ \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \qquad (6) \end{aligned}$$

where w is the inertia weight [43], c_1 and c_2 are the acceleration constants and $\mathbf{r}_1(t), \mathbf{r}_2(t)$ are vectors with their elements sampled from a uniform distribution, $U(0, 1)$. Equation (5) consists of three components, namely:

- The *inertia term*, which serves as a memory of previous velocities. The inertia weight controls the impact of the previous velocity: a large inertia weight favors exploration, while a small inertia weight favors exploitation [44].

- The *cognitive component*, $\mathbf{y}_i(t) - \mathbf{x}_i$, which represents the particle's own experience as to where the best solution is. The cognitive component serves as a memory of previous best positions for each particle.

- The *social component*, $\hat{\mathbf{y}}(t) - \mathbf{x}_i(t)$, which represents the belief of the entire swarm as to where the best solution is.

The performance of the PSO is sensitive to the values of the parameters w, c_1 and c_2. While several suggestions (based on empirical studies) for good values can be found in the literature, theoretical studies have been done to provide bounds on these values. For example, Van den Bergh *et al* showed that, if

$$w > \frac{1}{2}(c_1 + c_2), \quad w < 1 \tag{7}$$

then the PSO exhibits convergent behavior [47, 48]. If the above condition is not satisfied, the PSO exhibit cyclic or divergent behavior.

To ensure that adjustments to particle velocities are not too large (which may cause the particles to leave the confines of the search space), velocity updates are usually clamped. Velocity clamping is, however, problem dependent.

The PSO algorithm performs repeated applications of the update equations above until a specified number of iterations has been exceeded, or until velocity updates are close to zero. The quality of particles is measured using a fitness function which reflects the optimality of the corresponding solution.

The original versions of PSO as given above, suffer when $\mathbf{x}_i = \mathbf{y}_i = \hat{\mathbf{y}}$, since the velocity update equation will depend only on the term $w\mathbf{v}_i(t)$. If this condition persists for a number of iterations, $w\mathbf{v}_i(t) \to 0$, which means that the global best particle stagnates. As a result, all particles converge on this global best position, with no guarantee that it is an optimum [47]. To overcome this problem, a new version of PSO with guaranteed local convergence was introduced by Van den Bergh *et al*, namely GCPSO [48, 47]. In GCPSO, the global best particle with index τ is updated using a different velocity update, namely

$$v_{\tau,j}(t+1) = -x_{\tau,j}(t) + \hat{y}_j(t) + wv_{\tau,j}(t) + \rho(t)(1 - 2r_{2,j}(t)) \tag{8}$$

which results in a position update of

$$x_{\tau,j}(t+1) \quad = \quad \hat{y}_j(t) + wv_{\tau,j}(t) + \rho(t)(1 - 2r_{2,j}(t)) \tag{9}$$

The term $-\mathbf{x}_\tau$ 'resets' the particle's position to the global best position $\hat{\mathbf{y}}$, $w\mathbf{v}_\tau$ signifies a search direction, and $\rho(t)(1 - 2\mathbf{r}_2(t)))$ adds a random search term to the equation. The constant $\rho(t)$ defines the area in which a better solution is searched.

The value of $\rho(0)$ is initialized to 1.0, with $\rho(t+1)$ defined as

$$\rho(t+1) = \begin{cases} 2\rho(t) & \text{if } \#successes > s_c \\ 0.5\rho(t) & \text{if } \#failures > f_c \\ \rho(t) & \text{otherwise} \end{cases} \tag{10}$$

A 'failure' occurs when $f(\hat{\mathbf{y}}(t)) \geq f(\hat{\mathbf{y}}(t-1))$ (in the case of a minimization problem) and the variable $\#failures$ is subsequently incremented (i.e. no apparent progress has been made). A success then occurs when $f(\hat{\mathbf{y}}(t)) < f(\hat{\mathbf{y}}(t-1))$. Van den Bergh *et al* suggest learning the control threshold values f_c and s_c dynamically [48, 47]. That is,

$$s_c(t+1) = \begin{cases} s_c(t)+1 & \text{if } \#failures(t+1) > f_c \\ s_c(t) & \text{otherwise} \end{cases} \tag{11}$$

$$f_c(t+1) = \begin{cases} f_c(t)+1 & \text{if } \#successes(t+1) > s_c \\ f_c(t) & \text{otherwise} \end{cases} \tag{12}$$

This arrangement ensures that it is harder to reach a success state when multiple failures have been encountered, and likewise, when the algorithm starts to exhibit overly confident convergent behavior, it is forced to randomly search a smaller region of the search space surrounding the global best position. For equation (10) to be well defined, the following rules should be implemented:

$$\#successes(t+1) > \#successes(t) \Rightarrow \#failures(t+1) = 0$$

$$\#failures(t+1) > \#failures(t) \Rightarrow \#successes(t+1) = 0$$

Van den Bergh *et al* suggest repeating the algorithm until ρ becomes sufficiently small, or until stopping criteria are met. Stopping the algorithm once ρ reaches a lower bound is not advised, as it does not necessarily indicate that all particles have converged – other particles may still be exploring different parts of the search space.

It is important to note that, for the GCPSO algorithm, all particles except for the global best particle, still follow equations (5) and (6). Only the global best particle follows the new velocity and position update equations.

PSO is generally considered as an evolutionary algorithm, such as genetic algorithms (GA) and evolutionary programs (EP) [11]. In the terminology of evolutionary algorithms, a swarm is equivalent to a population, and a particle is the same as an individual (chromosome). However, the PSO differs from evolutionary algorithms in that no reproduction occurs, and that particles have a memory of previously found best solutions. Furthermore, changes in particle positions are based on the exchange of socially acquired information as dictated by the social topology used.

3 Image Clustering

This section defines the terminology used throughout the rest of the paper. A measure is given to quantify the quality of image clustering algorithms, after which an overview of the clustering algorithms used in the paper (namely, K-means, FCM, KHM and H2) is presented. The PSO-based image clustering algorithms are then introduced.

For the purpose of this paper, let

- N_b denote the number of spectral bands of the image set
- N_p denote the number of image pixels
- N_c denote the number of spectral classes (as provided by the user)
- $\mathbf{z}_p$ denote the N_b components of pixel p
- $\mathbf{m}_j$ denote the mean of cluster j

For the rest of this paper, image clustering is with regard to the spectral domain (i.e. pixel values).

3.1 Measure of Quality

Different measures can be used to express the quality of image clustering algorithms. The most general measure of performance is the quantization error, defined as

$$J_e = \frac{\sum_{j=1}^{N_c} [\sum_{\forall \mathbf{z}_p \in C_j} d(\mathbf{z}_p, \mathbf{m}_j)]/|C_j|}{N_c} \tag{13}$$

where

$$d(\mathbf{z}_p, \mathbf{m}_j) = \sqrt{\sum_{k=1}^{N_b} (z_{pk} - m_{jk})^2} \tag{14}$$

3.2 General Iterative Clustering

Using the general form of iterative clustering used by Hamerly *el al* [17, 18], the steps of a clustering algorithm are:

1. Randomly initialize the N_c cluster means
2. **Repeat**
   (a) for each pixel, $\mathbf{z}_p$, in the image, compute its membership $u(\mathbf{m}_j|\mathbf{z}_p)$ to each centroid $\mathbf{m}_j$ and its weight $w(\mathbf{z}_p)$ (i.e. how much influence pixel $\mathbf{z}_p$ has in recomputing the centroids in the next iteration; $w(\mathbf{z}_p) > 0$ [18]).

   (b) recalculate the N_c cluster means, using

$$\mathbf{m}_j = \frac{\sum_{\forall \mathbf{z}_p} u(\mathbf{m}_j|\mathbf{z}_p) w(\mathbf{z}_p) \mathbf{z}_p}{\sum_{\forall \mathbf{z}_p} u(\mathbf{m}_j|\mathbf{z}_p) w(\mathbf{z}_p)} \quad \text{for } j = 1, \cdots, N_c \tag{15}$$

until a stopping criterion is satisfied.

Different stopping criteria can be used, for example:

- stop when the change in centroid values are smaller than a user-specified value
- stop when the quantization error is small enough
- stop when a maximum number of iterations has been exceeded.

This paper terminates the clustering process after a fixed number of iterations have been reached to allow for a fair comparison of clustering algorithms.

In the next subsections, specific iterative clustering algorithms are described by defining the membership and weight functions in equation (15).

3.2.1 The K-means algorithm

The membership and weight functions for K-means are defined as

$$u(\mathbf{m}_j|\mathbf{z}_p) = \begin{cases} 1 & \text{if arg min}_j \; \|\mathbf{z}_p - \mathbf{m}_j\|^2 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

and

$$w(\mathbf{z}_p) = 1 \tag{17}$$

3.2.2 The FCM algorithm

The membership and weight functions for FCM are defined as

$$u(\mathbf{m}_j|\mathbf{z}_p) = \frac{||\mathbf{z}_p - \mathbf{m}_j||^{-2/(q-1)}}{\sum_{j=1}^{N_c} ||\mathbf{z}_p - \mathbf{m}_j||^{-2/(q-1)}} \tag{18}$$

and

$$w(\mathbf{z}_p) = 1 \tag{19}$$

where q is the fuzziness exponent and $q \geq 1$.

3.2.3 The KHM algorithm

The membership and weight functions for KHM are defined as

$$u(\mathbf{m}_j|\mathbf{z}_p) = \frac{||\mathbf{z}_p - \mathbf{m}_j||^{-p-2}}{\sum_{j=1}^{N_c} ||\mathbf{z}_p - \mathbf{m}_j||^{-p-2}} \tag{20}$$

and

$$w(\mathbf{z}_p) = \frac{\sum_{j=1}^{N_c} ||\mathbf{z}_p - \mathbf{m}_j||^{-p-2}}{(\sum_{j=1}^{N_c} ||\mathbf{z}_p - \mathbf{m}_j||^{-p})^2} \tag{21}$$

where p is an input parameter; typically $p \geq 2$.

3.2.4 The H2 algorithm

The membership and weight functions for H2 are defined as

$$u(\mathbf{m}_j|\mathbf{z}_p) = \frac{||\mathbf{z}_p - \mathbf{m}_j||^{-p-2}}{\sum_{j=1}^{N_c} ||\mathbf{z}_p - \mathbf{m}_j||^{-p-2}} \tag{22}$$

and

$$w(\mathbf{z}_p) = 1 \tag{23}$$

where p is an input parameter; typically $p \geq 2$.

The iterative nature of K-means, FCM, KHM and H2 algorithms makes them computationally expensive. Also, due to the greedy nature of K-means and FCM, these algorithms are susceptible to local minima (refer to section 4 for an illustration of this).

3.3 PSO-Based Image Clustering

In the context of image clustering, a single particle represents the N_c cluster means. That is, each particle $\mathbf{x}_i$ is constructed as $\mathbf{x}_i = (\mathbf{m}_{i1}, \cdots, \mathbf{m}_{ij}, \cdots, \mathbf{m}_{iN_c})$ where $\mathbf{m}_{ij}$ refers to the j-th cluster centroid vector of the i-th particle. Therefore, a swarm represents a number of candidate image clusterings.

Omran *et al* introduced a *gbest* PSO image clustering algorithm in [35], where the quality of each particle is measured using

$$f(\mathbf{x}_i, \mathbf{Z}_i) = w_1 \overline{d}_{max}(\mathbf{Z}_i, \mathbf{x}_i) + w_2(z_{max} - d_{min}(\mathbf{x}_i)) \tag{24}$$

where z_{max} is the maximum pixel value in the image set (i.e. $z_{max} = 2^b - 1$ for an b-bit image); $\mathbf{Z}_i$ is a matrix representing the assignment of pixels to clusters of particle i. Each element z_{ijp} indicates if pixel $\mathbf{z}_p$ belongs to cluster C_{ij} of particle i. The constants w_1 and w_2 are user defined constants. Also,

$$\overline{d}_{max}(\mathbf{Z}_i, \mathbf{x}_i) = \max_{j=1,\cdots,N_c} \{ \sum_{\forall \mathbf{z}_p \in C_{ij}} d(\mathbf{z}_p, \mathbf{m}_{ij}) / |C_{ij}| \}$$

is the maximum average Euclidean distance of particles to their associated classes, and

$$d_{min}(\mathbf{x}_i) = \min_{\forall j_1, j_2, j_1 \neq j_2} \{ d(\mathbf{m}_{ij_1}, \mathbf{m}_{ij_2}) \}$$

is the minimum Euclidean distance between any pair of clusters. In the above, $|C_{ij}|$ is the cardinality of the set C_{ij}.

This fitness function has as objective to simultaneously

- minimize the intra-distance between pixels and their cluster means, as quantified by $\overline{d}_{max}(\mathbf{Z}_i, \mathbf{x}_i)$, and
- maximize the inter-distance between any pair of clusters, as quantified by $d_{min}(\mathbf{x}_i)$.

The fitness function is thus a multi-objective problem. Approaches to solve multi-objective problems have been developed mostly for evolutionary computation approaches [8]. Recently, an approach to multi-objective optimization using PSO has been developed [9]. Since the focus of this paper is to illustrate the applicability of PSO to image clustering, and not on multi-objective optimization, this paper uses a simple approach to cope with multiple objectives. Different priorities are assigned to the sub-objectives through appropriate initialization of the values of w_1 and w_2.

Experimental results in [35] have shown the PSO image classifier to improve on the performance of the K-means algorithm. This paper proposes a different fitness function, i.e.

$$f(\mathbf{x}_i, \mathbf{Z}_i) = w_1 \overline{d}_{max}(\mathbf{Z}_i, \mathbf{x}_i) + w_2(z_{max} - d_{min}(\mathbf{x}_i)) + w_3 J_{e,i} \tag{25}$$

which simply adds to the fitness function an additional sub-objective to also minimize the quantization error.

The PSO image clustering algorithm is summarized below:

```
1. Initialize each particle to contain $N_c$ randomly selected cluster means
2. For $t = 1$ to $t_{max}$
    (a) For each particle $i$
        i. For each pixel $\mathbf{z}_p$
            • calculate $d(\mathbf{z}_p, \mathbf{m}_{ij})$ for all clusters $C_{ij}$
            • assign $\mathbf{z}_p$ to $C_{ij}$ where
              $d(\mathbf{z}_p, \mathbf{m}_{ij}) = \min_{\forall c=1,\cdots,N_c}\{d(\mathbf{z}_p, \mathbf{m}_{ic})\}$
        ii. Calculate the fitness, $f(\mathbf{x}_i(t), \mathbf{Z}_i)$
    (b) Find the global best solution $\hat{\mathbf{y}}(t)$
    (c) Update the cluster centroids using equations (5) and (6)
```

In addition to the new fitness function, this paper also investigates the performance of the GCPSO for image clustering to see if any improvement can be obtained, and adds comparisons with K-means, FCM, KHM, H2 and a GA clustering algorithm. For comparison reasons, another population-based approach has been used, namely GA. The above algorithm of PSO has been used for a GA-based image classifier except for step (c) which has been replaced by the GA operators: elitism, tournament selection, uniform crossover and random mutation.

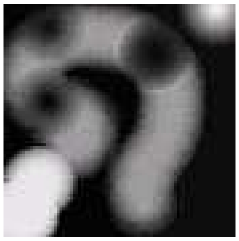


Figure 1: Synthesized Image

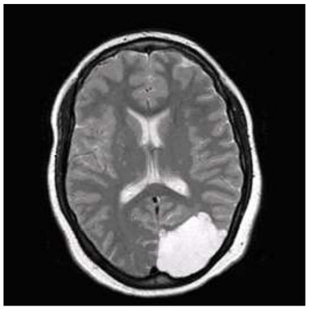


Figure 2: MRI Image of Human Brain

An advantage of using PSO is that a parallel search for an optimal clustering is performed. This population-based search approach reduces the effect of initial conditions, compared to K-means (especially for relatively large swarm sizes).

4 Experimental Results

The PSO-based image clustering algorithms have been applied to three types of imagery data, namely synthesized, MRI and LANDSAT 5 MSS (79 m GSD) images. These data sets have been selected to test the algorithm, and to compare it with other algorithms, on a range of problem types, as listed below:

- **Synthesized Image**: Figure 1 shows a 100 × 100 8-bit gray scale image created to specifically show that the PSO algorithm does not get trapped in the local minimum. The image was created using two types of brushes,

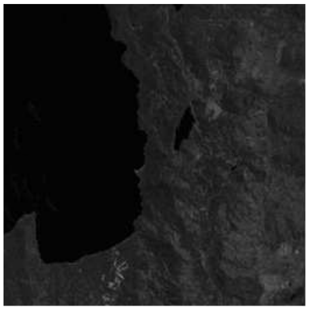


Figure 3: Band 4 of the Landsat MSS test image of Lake Tahoe

one brighter than the other.

- **MRI Image**: Figure 2 shows a 300 × 300 8-bit gray scale image of a human brain, intensionally chosen for its importance in medical image processing.

- **Remotely Sensed Imagery Data**: Figure 3 shows band 4 of the four-channel multi-spectral test image set of the Lake Tahoe region of the US. Each channel comprises of a 300 × 300 8-bit per pixel (remapped from the original 6-bit) image. The test data are one of the North America Landscape Characterization (NALC) Landsat multi-spectral scanner data sets obtained from the US Geological Survey (USGS).

The rest of this section is organized as follows: Section 4.1 illustrates that the basic PSO can be used successfully as image classifier, using the original fitness function as defined in equation (24). Section 4.2 illustrates the performance under the new fitness function. Results of the *gbest* PSO are compared with that of GCPSO in section 4.3, using the new fitness function. Section 4.4 investigates the influence of the different PSO control parameters. The performance of PSO using the new fitness function is compared with *state-of-the-art* clustering approaches in Section 4.5.

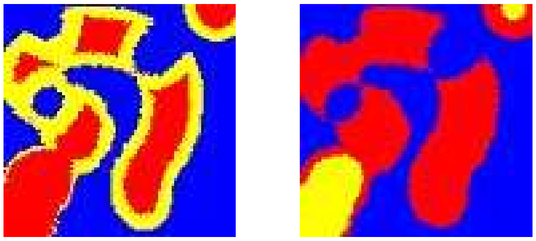
The results reported in this section are averages and standard deviations over 20 simulations. All comparisons are made with reference to $J_e, \overline{d}_{max}$ and d_{min}.

4.1 *gbest* PSO versus K-Means

This section presents a short graphic summary of the results in [35] to illustrate the applicability of PSO to image clustering. Figure 4(a) illustrates the thematic map image of the synthesized image for K-means, while figure 4(b) illustrates the thematic map obtained from the PSO algorithm. These figures clearly illustrates that K-means was trapped in a local minimum, and could not classify the clusters correctly. PSO, on the other hand, was not trapped in this local minimum. The thematic maps for the MRI and Tahoe images are given in figures 5 and 6. Table 1 presents a summary of the results using 10 particles, $w_1 = w_2 = 0.5, c_1 = c_2 = 1.49$ and $w = 0.72$ (these values were found to ensure good convergence [48]). Velocities were clamped at $V_{max} = 4$.

4.2 Improved Fitness Function

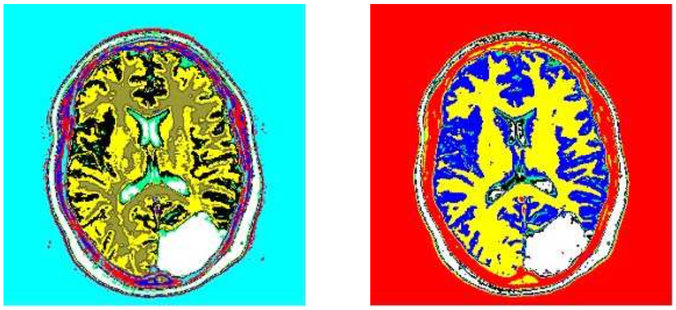
This section compares the results of the *gbest* PSO above with results using the new fitness function as defined in equation (25). For this purpose, 50 particles were trained for 100 iterations, $V_{max} = 5$, $w = 0.72$ and $c_1 = c_2 = 1.49$. For fitness function (24), $w_1 = w_2 = 0.5$ and for fitness function (25), $w_1 = w_2 = 0.3, w_3 = 0.4$ were used for the synthetic image, $w_1 = 0.2, w_2 = 0.5, w_3 = 0.3$ were used for the MRI image and $w_1 = w_2 = w_3 = 0.333$ were used for the Tahoe image. A total number of clusters of 3, 8 and 4 was used respectively for the synthetic, MRI and Tahoe images.



(a) K-means

(b) PSO

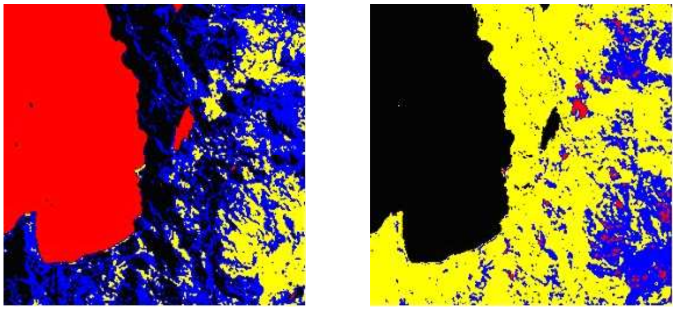
Figure 4: Thematic Maps for the Synthesized Image



(a) K-means

(b) PSO

Figure 5: Thematic Maps for the MRI Image



(a) K-means

(b) PSO

Figure 6: Thematic Maps for the Lake Tahoe Image

Table 1: Comparison between K-means and PSO

Image		$\mathbf{J}_e$	$\overline{\mathbf{d}}_{max}$	$\mathbf{d}_{min}$	**Function Evaluations**
Synthesized	K-means	20.466	28.676	77.146	10000
	PSO	24.413	27.156	98.680	10000
MRI	K-means	7.370	13.214	9.934	5000
	PSO	11.030	12.777	37.466	5000
Tahoe	K-means	7.281	11.877	17.674	1000
	PSO	12.344	13.760	27.920	1000

Table 2: 2-Component versus 3-Component Fitness Function

	2-Component Fitness Function			**3-Component Fitness Function**		
Problem	J_e	$\overline{d}_{max}$	d_{min}	J_e	$\overline{d}_{max}$	d_{min}
Synthetic	24.452958 ± 0.209681	27.157489 ± 0.017690	98.678918 ± 0.023706	17.112672 ± 0.548096	24.781384 ± 0.270409	92.767925 ± 4.043086
MRI	8.535903 ± 0.584300	10.129168 ± 1.261646	28.744714 ± 2.948953	7.225384 ± 0.552381	12.205947 ± 2.506827	22.935685 ± 8.310654
Tahoe	7.214660 ± 2.392860	9.035649 ± 3.362867	25.777255 ± 9.601882	3.556281 ± 0.139881	4.688270 ± 0.259919	14.986923 ± 0.425077

Table 2 compares the results for the two fitness functions. The new fitness function succeeded in significant improvements in the quantization error, J_e. The new fitness function also achieved significant improvement in minimizing the intra-cluster distances for the synthetic and Tahoe images, thus resulting in more compact clusters, and only marginally worse for the MRI image. These improvements were at the cost of loosing on maximization of the inter-cluster distances.

Due to the improved performance on the quantization error and intra-cluster distances, the rest of this paper uses the 3-component fitness function as defined in equation (25).

4.3 *gbest* PSO versus GCPSO

This section compares the performance of the *gbest* PSO with the GCPSO. This is done for a low $V_{max} = 5$ and a high $V_{max} = 255$. All other parameters are as for section 4.2. Table 3 shows no significant difference in the performance between PSO and GCPSO. It is, however, important to note that too much clamping of the velocity updates have a

Table 3: PSO versus GCPSO

Problem	**PSO**			**GCPSO**		
$V_{max} = 5$	J_e	$\overline{d}_{max}$	d_{min}	J_e	$\overline{d}_{max}$	d_{min}
Synthetic	17.112672 ± 0.548096	24.781384 ± 0.270409	92.767925 ± 4.043086	17.116036 ± 0.547317	24.826868 ± 0.237154	92.845323 ± 4.056681
MRI	7.225384 ± 0.552381	12.205947 ± 2.506827	22.935786 ± 8.310654	7.239264 ± 0.475250	12.438016 ± 2.437064	23.377287 ± 6.722787
Tahoe	3.556281 ± 0.139881	4.688270 ± 0.259919	14.986923 ± 0.425077	3.542732 ± 0.109415	4.672483 ± 0.129913	15.007491 ± 0.621020
$V_{max} = 255$						
Synthetic	17.004993 ± 0.086698	24.615665 ± 0.143658	93.478081 ± 0.276109	17.000393 ± 0.022893	24.672107 ± 0.174457	93.588530 ± 0.400137
MRI	7.640622 ± 0.514184	10.621452 ± 1.284735	24.948486 ± 3.446673	7.694498 ± 0.591383	10.543233 ± 1.038114	25.355967 ± 3.945929
Tahoe	3.523967 ± 0.172424	4.681492 ± 0.110739	14.664859 ± 1.177861	3.609807 ± 0.188862	4.757948 ± 0.227090	15.282949 ± 1.018218

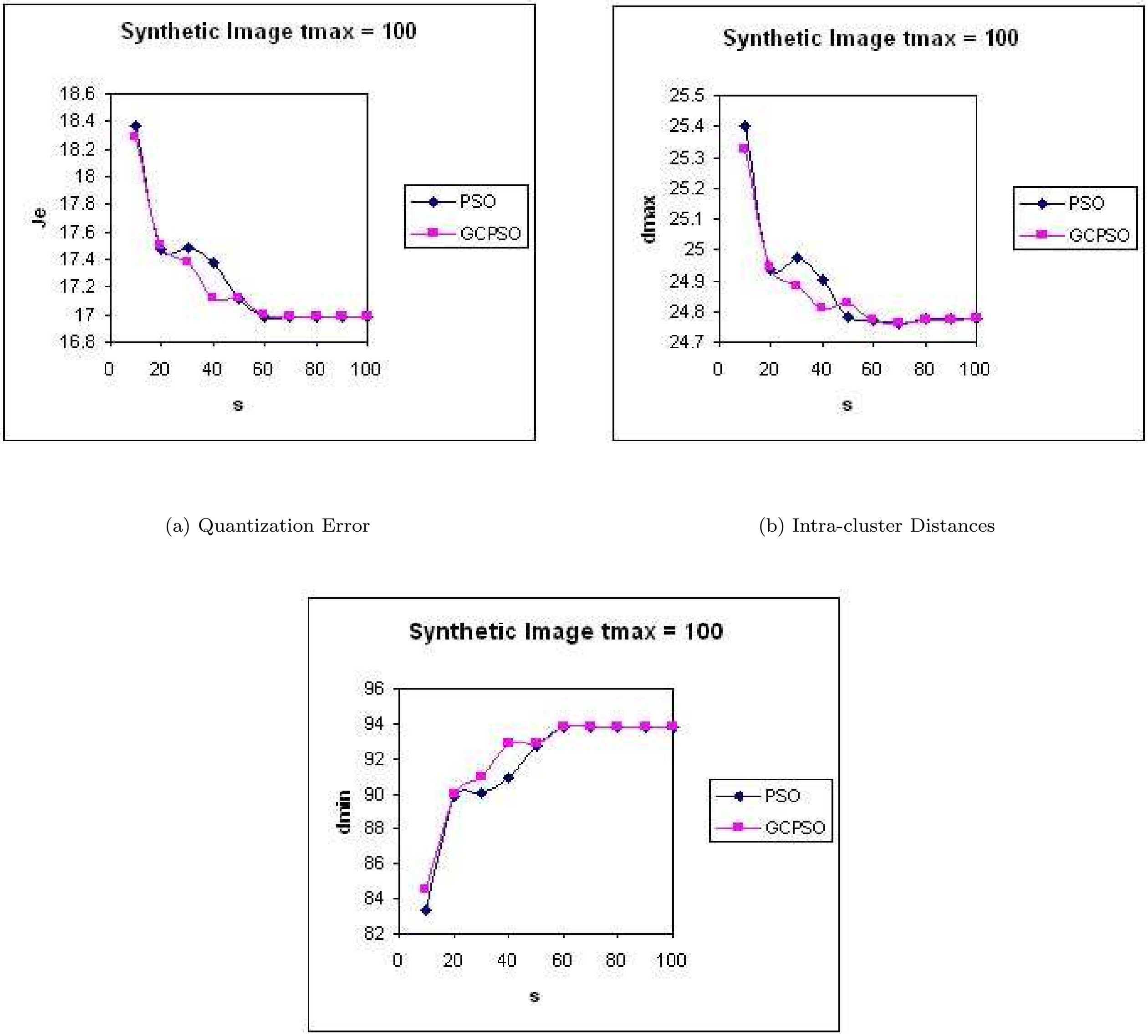
negative influence on performance. Better results were obtained, for both the PSO and GCPSO, with a large value of V_{max}.

4.4 Influence of PSO parameters

The PSO have a number of parameters that have an influence on the performance of the algorithm. These parameters include V_{max}, the number of particles, the inertia weight and the acceleration constants. Additionally, the PSO-based image classifier adds a weight to each sub-objective. This section investigates the influence of different values of these parameters.

Velocity Clamping

Table 3 shows that less clamping of velocity updates is more beneficial. This allows larger jumps of particles in the search space.



(a) Quantization Error

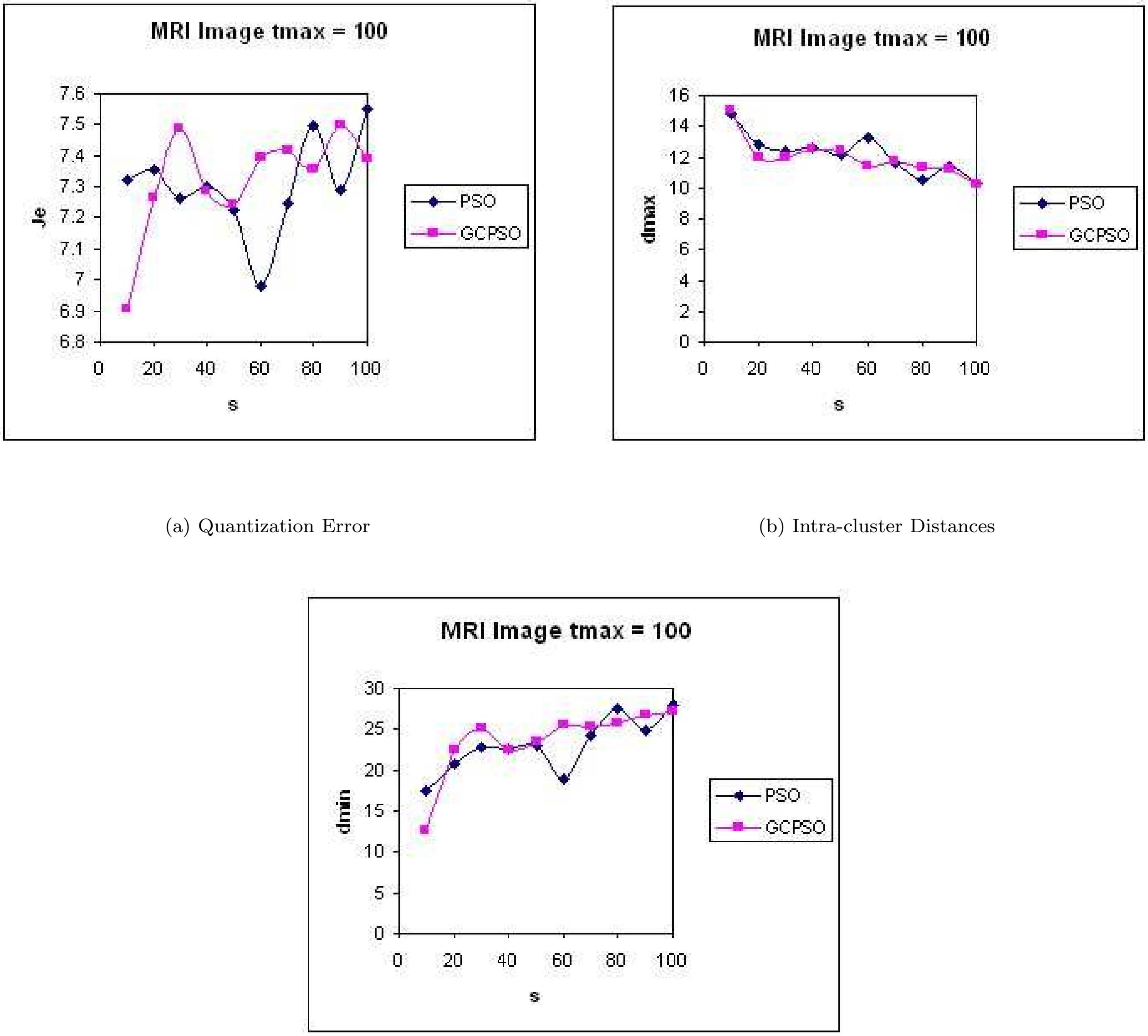
(b) Intra-cluster Distances

(c) Inter-cluster Distances

Figure 7: Effect of swarm size on synthetic image

Swarm Size

To investigate the effect of different swarm sizes on performance, both the PSO and GCPSO have been executed using 10 to 100 particles. All parameters are as for section 4.2. Figure 7 shows the effect of the swarm size, s, on the synthetic image. It is clear from the figure that increasing the number of particles improves the performance of both algorithms. The same conclusion can be drawn for the MRI image as illustrated in figure 8. An increase in the number of particles increases diversity, thereby limiting the effects of initial conditions and reducing the possibility of being trapped in local minima.



(a) Quantization Error

(b) Intra-cluster Distances

(c) Inter-cluster Distances

Figure 8: Effect of swarm size on MRI image

Table 4: Effect of inertia weight on the synthetic image

	PSO			**GCPSO**		
w	J_e	$\overline{d}_{max}$	d_{min}	J_e	$\overline{d}_{max}$	d_{min}
0.0	16.983429 ± 0.017011	24.581799 ± 0.165103	93.435221 ± 0.308601	16.986386 ± 0.016265	24.649368 ± 0.138223	93.559275 ± 0.254670
0.1	16.982362 ± 0.016074	24.645884 ± 0.137442	93.543795 ± 0.256700	16.985079 ± 0.016995	24.637893 ± 0.138894	93.538635 ± 0.257167
0.5	16.985826 ± 0.014711	24.664421 ± 0.144252	93.595394 ± 0.246110	16.987470 ± 0.028402	24.662973 ± 0.163768	93.581237 ± 0.281366
0.72	16.992102 ± 0.021756	24.670338 ± 0.150542	93.606400 ± 0.258548	16.995967 ± 0.039686	24.722414 ± 0.144572	93.680765 ± 0.253954
0.9	16.993759 ± 0.014680	24.650337 ± 0.140005	93.569595 ± 0.252781	17.040990 ± 0.168017	24.633802 ± 0.352785	93.495340 ± 0.584424
1.4 to 0.8	17.824495 ± 0.594291	24.433770 ± 1.558219	92.625088 ± 2.031224	17.481146 ± 0.504740	24.684407 ± 1.010815	93.223498 ± 1.490217

Inertia Weight

Given that all parameters are fixed at the values given in section 4.2, the inertia weight w was set to different values for both PSO and GCPSO. In addition, a dynamic inertia weight was used with an initial $w = 1.4$, which linearly decreased to 0.8. The initial large value of w favors exploration in the early stages, and exploitation in the later stages. Tables 4 and 5 summarize the results for the synthetic and MRI images respectively. The results illustrate no significant difference in performance, meaning that for the two images, the PSO-based clustering algorithms are insensitive to the value of the inertia weight (provided that c_1 and c_2 are selected such that equation (7) is not violated).

Acceleration Coefficients

Given that all parameters are fixed at the values given in section 4.2, the influence of different values for the acceleration coefficients, c_1 and c_2, were evaluated for the synthetic and MRI images. Tables 6 and 7 summarize these results. For these choices of the acceleration coefficients, no single choice is superior to the others. While these tables indicate an independence to the value of the acceleration coefficients, it is important to note that convergence depends on the relationship between the inertia weight and the acceleration coefficients, as derived in [48, 47] (also refer to equation (7).

Table 5: Effect of inertia weight on the MRI image

	PSO			**GCPSO**		
w	J_e	$\overline{d}_{max}$	d_{min}	J_e	$\overline{d}_{max}$	d_{min}
0.0	7.538669 ± 0.312044	9.824915 ± 0.696940	28.212823 ± 2.300930	7.497944 ± 0.262656	9.731746 ± 0.608752	28.365827 ± 1.882164
0.1	7.511522 ± 0.281967	10.307791 ± 1.624499	27.150801 ± 3.227550	7.309289 ± 0.452103	10.228958 ± 1.354945	26.362349 ± 3.238452
0.5	7.612079 ± 0.524669	10.515242 ± 1.103493	26.996556 ± 2.161969	7.466388 ± 0.492750	10.348044 ± 1.454050	26.790056 ± 2.830860
0.72	7.574454 ± 0.382172	10.150214 ± 1.123441	27.393498 ± 3.260418	7.467591 ± 0.396310	10.184191 ± 0.955129	26.596493 ± 3.208689
0.9	7.847689 ± 0.529134	10.779765 ± 1.134843	26.268057 ± 3.595596	7.598518 ± 0.516938	10.916945 ± 1.534848	25.417859 ± 3.174232
1.4 to 0.8	8.354957 ± 0.686190	13.593536 ± 2.035889	21.625623 ± 4.507230	8.168068 ± 0.709875	12.722139 ± 1.850957	21.169304 ± 4.732452

Table 6: Effect of acceleration coefficients on the synthetic image

	PSO			**GCPSO**		
w	J_e	$\overline{d}_{max}$	d_{min}	J_e	$\overline{d}_{max}$	d_{min}
$c_1 = 0.7$ $c_2 = 1.4$	16.989197 ± 0.011786	24.726716 ± 0.101239	93.698591 ± 0.184244	16.989355 ± 0.012473	24.708151 ± 0.120168	93.667144 ± 0.207355
$c_1 = 1.4$ $c_2 = 0.7$	16.991884 ± 0.016970	24.700627 ± 0.125603	93.658673 ± 0.208500	16.993095 ± 0.040042	24.685461 ± 0.165669	93.619162 ± 0.279258
$c_1 = 1.49$ $c_2 = 1.49$	16.987582 ± 0.009272	24.710933 ± 0.122622	93.672792± 0.206395	16.995967 ± 0.039686	24.722414 ± 0.144572	93.680765 ± 0.253954

Table 7: Effect of acceleration coefficients on the MRI image

	PSO			GCPSO		
w	J_e	$\overline{d}_{max}$	d_{min}	J_e	$\overline{d}_{max}$	d_{min}
$c_1 = 0.7$ $c_2 = 1.4$	7.599324 ± 0.289702	10.145010 ± 1.353091	26.977217 ± 3.467738	7.530823 ± 0.477134	10.201762 ± 0.986726	26.425638 ± 3.248949
$c_1 = 1.4$ $c_2 = 0.7$	7.528712 ± 0.439470	10.238989 ± 1.484245	27.747333 ± 2.850575	7.476468 ± 0.459432	10.159019 ± 1.085977	27.001444 ± 3.360799
$c_1 = 1.49$ $c_2 = 1.49$	7.499845 ± 0.416682	10.203905 ± 0.951100	26.629647 ± 2.652593	7.467591 ± 0.396310	10.184191 ± 0.955129	26.596493 ± 3.208689

Sub-objective Weight Values

Tables 8 and 9 summarize the effects of different values of the weights, w_1, w_2 and w_3, of the sub-objectives for the synthetic and MRI images respectively. Table 9 shows that weights of $w_1 = 0.1, w_2 = 0.1$ and $w_3 = 0.8$ result in the smallest quantization error, shortest intra-distances and largest inter-distances. It is not so clear which sub-objective weight value combination is best for the synthetic image. To eliminate tuning of these weight values, an alternative multi-objective approach can be followed [8, 9].

4.5 *gbest* PSO versus *state-of-the-art* clustering algorithms

This section compares the performance of the *gbest* PSO with K-means, FCM, KHM, H2 and a GA clustering algorithm. This is done for a high $V_{max} = 255$. All other parameters are as for section 4.2. For FCM, q was set to 2 since it is the commonly used value [19]. For KHM and H2, p was set to 2.5 and 4 respectively since these values produced the best results according to our preliminary tests. For the GA, a tournament size of 2 was used, a uniform crossover probability of 0.8 with mixing ratio of 0.5, and a mutation probability of 0.05. Only the best individual survived to the next generation. The results are summarized in table 10. These results are also averages over 20 simulation runs. Table 10 shows that PSO generally outperformed K-means, FCM, KHM and H2 in d_{min} and $\overline{d}_{max}$; while performing comparably with respect to J_e. The PSO and GA showed similar performance, with no significant difference.

These results show that the PSO image clustering algorithm is a viable alternative that merits further investigation.

5 Conclusions

This paper presented a new approach to image clustering using particle swarm optimization (PSO). The PSO clustering algorithm has as objective to simultaneously minimize the quantization error and intra-cluster distances, and to

Table 8: Effect of sub-objective weight values on synthetic image

			PSO			**GCPSO**		
w_1	w_2	w_3	J_e	$\overline{d}_{max}$	d_{min}	J_e	$\overline{d}_{max}$	d_{min}
0.3	0.3	0.3	17.068816 ± 0.157375	24.672006 ± 0.572276	93.594977 ± 0.984724	17.010279 ± 0.059817	24.742272 ± 0.258118	93.711385 ± 0.437418
0.8	0.1	0.1	17.590421 ± 0.353375	21.766287 ± 0.127098	88.892284 ± 0.143159	17.514336 ± 0.025242	21.724623 ± 0.018983	88.879342 ± 0.062452
0.1	0.8	0.1	18.827495 ± 0.558357	27.623976 ± 0.427120	97.719446 ± 0.202744	18.827120 ± 0.688529	27.522388 ± 0.282601	97.768398 ± 0.266885
0.1	0.1	0.8	16.962755 ± 0.003149	24.495742 ± 0.089611	93.228826 ± 0.135893	16.983721 ± 0.122501	24.546881 ± 0.434417	92.576271 ± 4.357444
0.1	0.45	0.45	17.550448 ± 0.184982	26.707924 ± 0.692239	96.020559 ± 0.757185	17.557817 ± 0.226305	26.598435 ± 0.907974	95.888089 ± 1.152158
0.45	0.45	0.1	18.134349 ± 0.669151	26.489043 ± 0.982256	96.461779 ± 1.495491	18.294904 ± 0.525467	26.795286 ± 0.800436	96.922471 ± 1.225336
0.45	0.1	0.45	17.219807 ± 0.110357	22.631958 ± 0.522369	90.152811 ± 0.887423	17.201690 ± 0.093969	22.701159 ± 0.469470	90.289690 ± 0.828522

Table 9: Effect of sub-objective weight values on MRI image

			PSO			GCPSO		
w_1	w_2	w_3	J_e	$\overline{d}_{max}$	d_{min}	J_e	$\overline{d}_{max}$	d_{min}
0.3	0.3	0.3	7.239181 ± 0.576141	10.235431 ± 1.201349	24.705469 ± 3.364803	7.194243 ± 0.573013	10.403608 ± 1.290794	23.814072 ± 3.748753
0.8	0.1	0.1	7.364818 ± 0.667141	9.683816 ± 0.865521	24.021787 ± 3.136552	7.248268 ± 0.474639	9.327774 ± 0.654454	23.103375 ± 4.970816
0.1	0.8	0.1	8.336001 ± 0.599431	11.256763 ± 1.908606	31.106734 ± 1.009284	8.468620 ± 0.626883	11.430190 ± 1.901736	30.712733 ± 1.336578
0.1	0.1	0.8	6.160486 ± 0.241060	15.282308 ± 2.300023	2.342706 ± 5.062570	6.088302 ± 0.328147	15.571290 ± 2.410393	1.659674 ± 4.381048
0.1	0.45	0.45	7.359711 0.423120	10.826327 1.229358	24.536828 ± 3.934388	7.303304 ± 0.439635	11.602263 ± 1.975870	22.939088 ± 3.614108
0.45	0.45	0.1	8.001817 ± 0.391616	9.885342 ± 0.803478	28.057459 ± 1.947362	7.901145 ± 0.420714	9.657340 ± 0.947210	29.236420 ± 1.741987
0.45	0.1	0.45	6.498429 ± 0.277205	11.392347 ± 2.178743	12.119429 ± 8.274427	6.402205 ± 0.363938	10.939902 ± 2.301587	14.422413 ± 6.916785

Table 10: Comparison between K-means, FCM, KHM, H2, GA and PSO

Image		$\mathbf{J}_e$	$\overline{\mathbf{d}}_{max}$	$\mathbf{d}_{min}$	**Function Evaluations**
Synthesized	K-means	20.21225 ± 0.937836	28.04049 ± 2.777938	78.4975 ± 7.062871	5000
	FCM	20.73192 ± 0.650023	28.55921 ± 2.221067	82.4341 ± 4.404686	5000
	KHM	20.16857 ± 0.0	23.36242 ± 0.0	86.3076 ± 0.000008	5000
	H2	20.13642 ± 0.793973	26.68694 ± 3.011022	81.8341 ± 6.022036	5000
	GA	17.00400 ± 0.035146	24.60302 ± 0.11527	93.4922 ± 0.2567	5000
	PSO	16.98891 ± 0.023937	24.69606 ± 0.130334	93.6322 ± 0.248234	5000
MRI	K-means	7.3703 ± 0.042809	13.21437 ± 0.761599	9.9343 ± 7.30853	5000
	FCM	7.205987 ± 0.166418	10.85174 ± 0.960273	19.5178 ± 2.014138	5000
	KHM	7.53071 ± 0.129073	10.65599 ± 0.295526	24.2708 ± 2.04944	5000
	H2	7.264114 ± 0.149919	10.92659 ± 0.737545	20.5435 ± 1.871984	5000
	GA	7.038909 ± 0.508953	9.811888 ± 0.419176	25.9542 ± 2.993480	5000
	PSO	7.594520 ± 0.449454	10.18609 ± 1.237529	26.7059 ± 3.008073	5000
Tahoe	K-means	3.280730 ± 0.095188	5.234911 ± 0.312988	9.40262 ± 2.823284	5000
	FCM	3.164670 ± 0.000004	4.999294 ± 0.000009	10.9706 ± 0.000015	5000
	KHM	3.830761 ± 0.000001	6.141770 ± 0.0	13.7684 ± 0.000002	5000
	H2	3.197610 ± 0.000003	5.058015 ± 0.000007	11.0529 ± 0.000012	5000
	GA	3.472897 ± 0.151868	4.645980 ± 0.105467	14.4469 ± 0.857770	5000
	PSO	3.523967 ± 0.172424	4.681492 ± 0.110739	14.6649 ± 1.177861	5000

maximize the inter-cluster distances. Both a *gbest* PSO and GCPSO algorithm have been evaluated. The *gbest* PSO clustering algorithm was further compared against K-means, FCM, KHM, H2 and a GA. In general, the PSO algorithm produced better results with reference to inter- and intra-cluster distances, while having quantization errors comparable to the other algorithms.

Although the fitness function used by the PSO-approach contains multiple objectives, no special multi-objective optimization techniques have been used. Further research will investigate the use of a PSO multi-objective approach, which may produce better results. A strategy will also be developed to dynamically determine the optimal number of clusters.

6 Acknowledgments

We are grateful to G. Hamerly and C. Elkan for providing us with MATLAB code for KHM.

References

[1] N. Alldrin, A. Smith, D. Turnbull, "Clustering with EM and K-means," Unpublished Manuscript, http://louis.ucsd.edu/ nalldrin/research/cse253_wi03.pdf, 15 Nov 2003.

[2] M. Antonie, O. Zaiane, A. Coman, "Application of Data Mining Techniques for Medical Image Classification," Proceedings of the 2nd International Workshop on Multimedia Mining, San Francisco, USA, pp. 94-101, 2001.

[3] J. Bala, J. Juang, H. Vafaie, K. DeJong, W. Wechsler, "Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification," International Joint Conference on Artificial Intelligence, Montreal, August, 1995.

[4] G. Ball, D. Hall, "A Clustering Technique for Summarizing Multivariate Data," Behavioral Science, vol. 12, pp. 153-155, 1967.

[5] J. Bezdek, "A Convergence Theorem for the fuzzy ISODATA Clustering Algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, pp. 1-8, 1980.

[6] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, 1981.

[7] C. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

[8] C.A. Coello-Coello, "An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design", PhD Thesis, Tulane University, 1996.

[9] C.A. Coello-Coello, M.S. Lechuga, “MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization”, IEEE Congress on Evolutionary Computation, Vol. 2, pp. 1051-1056, 2002.

[10] E. Davies, Machine Vision: Theory, Algorithms, Practicalities, Academic Press, 2nd Edition, 1997.

[11] A.P. Engelbrecht, “Computational Intelligence: An Introduction,” John Wiley, 2002.

[12] I. Evangelou, D. Hadjimitsis, A. Lazakidou, C. Clayton, “Data Mining and Knowledge Discovery in Complex Image Data using Artificial Neural Networks,” Proceedings of the Workshop on Complex Reasoning on Geographical Data (CRGD), 7th International Conference on Logic Programming - ICLP’01 and 7th International Conference on Principles and Practice of Constraint Programming - CP’01, Paphos, Cyprus, pp. 39-48, 2001.

[13] E. Falkenauer, Genetic Algorithms and Grouping Problems, John Wiley & Sons Publishing, 1998.

[14] E. Forgy, “Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification,” Biometrics, vol. 21, pp. 768-769, 1965.

[15] H. Frigui, R. Krishnapuram, “A Robust Competitive Clustering Algorithm with Applications in Computer Vision,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp. 450-465, 1999.

[16] I. Gath, A. Geva, “Unsupervised Optimal Fuzzy Clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp. 773-781, 1989.

[17] G. Hamerly, Learning structure and concepts in data using data clustering, PhD Thesis, University of California, San Diego, 2003.

[18] G. Hamerly, C. Elkan, “Alternatives to the K-means algorithm that find better clusterings,” Proceedings of the ACM Conference on Information and Knowledge Management, pp.600-607, 2002.

[19] F. Hoppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis, Methods for Classification, Data Analysis and Image Recognition. John Wiley & Sons Ltd, 1999.

[20] K. Huang, “A Synergistic Automatic Clustering Technique (Syneract) for Multispectral Image Analysis,” Photogrammetric Engineering and Remote Sensing, vol. 1, no. 1, pp. 33-40, 2002.

[21] J. Kennedy, R. Eberhart, “Particle Swarm Optimization,” Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942-1948, Perth, Australia, 1995.

[22] J. Kennedy, R. Eberhart, “A Discrete Binary Version of the Particle Swarm Algorithm,” Proceedings of the Conference on Systems, Man, and Cybernetics, pp. 4104-4109, 1997.

[23] J. Kennedy, W. Spears, “Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator,” Proceedings of the International Conference on Evolutionary Computation, pp. 78-83, Anchorage, Alaska, 1998.

[24] J. Kennedy, “Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance,” Proceedings of the Congress on Evolutionary Computation, pp. 1931-1938, 1999.

[25] J. Kennedy, R. Eberhart, Swarm Intelligence, Morgan Kaufmann, 2001.

[26] J. Kennedy, R. Mendes, “Population Structure and Particle Performance,” Proceedings of the IEEE Congress on Evolutionary Computation, Honolulu, Hawaii, 2002.

[27] T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30, Springer-Verlag, 1995.

[28] Y. Leung, J. Zhang, Z. Xu, “Clustering by Space-Space Filtering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1396-1410, 2000.

[29] T. Lillesand, R. Kiefer, Remote Sensing and Image Interpretation, John Wiley & Sons Publishing, 1994.

[30] A. Lorette, X. Descombes, J. Zerubia, “Fully Unsupervised Fuzzy Clustering with Entropy Criterion,” IEEE Proceedings of the International Conference on Pattern Recognition, pp. 986-989, 2000.

[31] J. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observations,” Proceedings 5th Berkeley Symposium on Mathematics, Statistics and Probability, vol. 1, pp. 281-297, 1967.

[32] P. Mather, Computer Processing of Remotely Sensed Images, John Wiley & Sons Publishing, 1999.

[33] U. Maulik, S. Bandyopadhyay, “Genetic Algorithm-Based Clustering Technique,” Pattern Recognition, vol. 33, pp. 1455-1465, 2000.

[34] G. McLachlan, T. Krishnan, The EM algorithm and Extensions, John Wiley & Sons, Inc., 1997.

[35] M. Omran, A. Salman, A.P. Engelbrecht, “Image Classification using Particle Swarm Optimization,” Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, Singapore, 2002.

[36] D. Paola, R. Schowenderdt, “A Review and Analysis of Back Propagation Neural Networks for Classification of Remotely Sensed Multi-Spectral Imagery,” International Journal of Remote Sensing, vol. 16, pp. 3033-3058, 1995.

[37] E.S. Peer, F. van den Bergh, A.P. Engelbrecht, “Using Neighborhoods with the Guaranteed Convergence PSO,” Proceedings of the IEEE Swarm Intelligence Symposium, Indianapolis, pp. 235-242, 2003.

[38] J. Puzicha, T. Hofmann, J.M. Buhmann, "Histogram Clustering for Unsupervised Image Segmentation," IEEE Proceedings of the Computer Vision and Pattern Recognition, pp. 602-608, 2000.

[39] V. Ramos, F. Muge, "Map Segmentation by Colour Cube Genetic K-Mean Clustering," Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science, Vol. 1923, pp. 319-323, Springer-Verlag, Lisbon, Portugal, September 2000.

[40] R. Rendner, H. Walker, "Mixture Densities, Maximum Likelihood and The EM Algorithm," SIAM Review, Vol. 26, No. 2, 1984.

[41] C. Rosenberger, K. Chehdi, "Unsupervised Clustering Method with Optimal Estimation of the Number of Clusters: Application to Image Segmentation," IEEE Proceedings of the International Conference on Pattern Recognition, pp. 656-659, 2000.

[42] P. Scheunders, "A Genetic C-means Clustering Algorithm Applied to Image Quantization," Pattern Recognition, vol. 30, no. 6, 1997.

[43] Y. Shi, R. Eberhart, "A Modified Particle Swarm Optimizer," IEEE International Conference of Evolutionary Computation, Anchorage, Alaska, May 1998.

[44] Y. Shi, R. Eberhart, "Parameter Selection in Particle Swarm Optimization," Evolutionary Programming VII: Proceedings of EP 98, pp. 591-600, 1998.

[45] P. Suganthan, "Particle Swarm Optimizer with Neighborhood Optimizer," Proceedings of the Congress on Evolutionary Computation, pp. 1958-1962, 1999.

[46] H Vafaie, K. De Jong, "Genetic Algorithms as a Tool for Feature Selection in Machine Learning," Fourth International Conference on Tools with Artificial Intelligence (ICTAI '92), Arlington, Virginia, USA, pp. 200-203, 1992.

[47] F. van den Bergh, A.P. Engelbrecht, "A New Locally Convergent Particle Swarm Optimizer," Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, Hammamet, Tunisia, 2002.

[48] F. van den Bergh, An Analysis of Particle Swarm Optimizers, PhD Thesis, University of Pretoria, 2002.

[49] B. Zhang, "Generalized K-Harmonic Means - Boosting in Unsupervised Learning," Technical Report HPL-2000-137). Hewlett-Packard Labs, 2000.

[50] B. Zhang, M. Hsu, U. Dayal, "K-Harmonic Means - A Data Clustering Algorithm," Technical Report HPL-1999-124). Hewlett-Packard Labs, 1999.

[51] Y. Zhang, M. Brady, S. Smith, “Segmentation of Brain MR Images Through a Hidden Markov Random Field Model and the Expectation-Maximization Algorithm,” IEEE Transactions on Medical Imaging, vol. 20, no. 1, pp. 45-57, 2001.