Image clustering using Particle Swarm Optimization

Conference Paper · June 2011

DOI: 10.1109/CEC.2011.5949627 · Source: DBLP

Man To Wong
University of Technology Sydney

Xiangjian He
University of Technology Sydney

Wei-Chang Yeh
National Tsing Hua University

Image Clustering Using Particle Swarm Optimization

Man To Wong, Xiangjian He
Centre for Innovation in IT Services and Applications
University of Technology, Sydney
Broadway, NSW 2007, Australia
eemtwong@gmail.com, xiangjian.he@uts.edu.au

Wei-Chang Yeh
Dept. of Industrial Engineering & Engineering Management
National Tsing Hua University
Hsinchu 300, Taiwan
yeh@ieee.org

***Abstract*—This paper proposes an image clustering algorithm using Particle Swarm Optimization (PSO) with two improved fitness functions. The PSO clustering algorithm can be used to find centroids of a user specified number of clusters. Two new fitness functions are proposed in this paper. The PSO-based image clustering algorithm with the proposed fitness functions is compared to the K-means clustering. Experimental results show that the PSO-based image clustering approach, using the improved fitness functions, can perform better than K-means by generating more compact clusters and larger inter-cluster separation.**

Keywords - particle swarm optimization, image clustering, K-means clustering, partitional clustering

I. INTRODUCTION

Clustering refers to the process of grouping samples (or data) so that the samples are similar within each group. The groups are called clusters [1]. Clustering algorithms are used in many applications, such as pattern recognition, image analysis, data mining and machine learning.

Clustering algorithms can be hierarchical or partitional [12]. In hierarchical clustering, the output is a tree showing a sequence of clustering, with each cluster being a partition of the data set. On the other hand, partitional clustering algorithms attempt to decompose the data set directly into a set of disjoint clusters [12]. They try to optimize certain criteria (e.g., a square-error function). A comprehensive survey of various clustering techniques can be found in [12].

A widely used partitional clustering algorithm is the K-means clustering algorithm [2]. K-means clustering groups data vectors into a predefined number of clusters, based on the Euclidean distance as similarity measure. Data vectors within a cluster have small Euclidean distance from one another, and are associated with the centroid vector, which represents the mean of the data vectors that belong to the cluster.

The standard K-means algorithm is summarized below:

1. Randomly initialize the cluster centroid vectors.
2. For each data vector, assign the vector to the cluster with the closest cluster center, using the Euclidean distance between the data vector and the centroid.
3. Re-calculate each cluster's centroid vector, which represents the mean of the data vectors that belong to the cluster.
4. Repeat 2 and 3 until a stopping criterion is satisfied.

The K-means clustering has the following two main advantages [12]. It is easy to implement and the time complexity is only O(n) (where n is the number of data points), which makes it suitable for large data sets. However, its performance depends on initial conditions, which may cause the algorithm to converge to suboptimal solutions.

Recently particle swarm optimization (PSO) [3,4] has been applied to image clustering [8,10] and it has been shown in [8] that PSO-based image clustering can have better performance than K-means. In PSO, a swarm of individuals (called particles) is maintained, where each particle represents a candidate solution to the optimization problem. 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. The performance of each particle is measured by a fitness function which depends on the optimization problem [5,10]. In PSO-based clustering, the design of good fitness functions for PSO is important to ensure the quality of clustering. Various fitness functions have been proposed for PSO based clustering [8,9,10]. In this paper, the objective is to propose two new fitness functions for PSO clustering that can provide good quality of image clustering. PSO clustering using these improved fitness functions can provide more compact clusters and larger separation between the cluster centroids when compared to K-means clustering.

The rest of the paper is organized as follows. Section II explains the PSO algorithm. Section III describes the past related work in PSO-based image clustering. Section IV describes our approach in PSO-based clustering and explains how the two new fitness functions proposed in this paper can improve the PSO-based clustering. Experimental results and discussion using three natural images are provided in Section V. Conclusions are provided in Section VI.

II. PARTICLE SWARM OPTIMIZATION (PSO)

PSO is a population based stochastic optimization technique modeled after the social behavior of bird flocks [3,4]. In PSO, the algorithm maintains a population of particles, where each particle represents a potential solution to the optimization problem. Each particle is also assigned a randomized velocity. The particles are then flown through the problem space [4,5,6]. The aim of PSO is to find the particle position that results in the best evaluation of a given fitness function.

Each particle keeps track of the following information in the problem space: x_i, the current position of the particle; v_i, the current velocity of the particle; and y_i, the personal best position of the particle which is the best position that it has achieved so far. This position yields the best fitness value for that particle. The fitness value of this position, called *pbest*, is also stored.

There are two approaches to PSO, namely local best (*lbest*) and global best (*gbest*). The difference is in the neighborhood topology used to exchange information among the particles. For the *gbest* model, the best particle is determined from the entire swarm. For the *lbest* model, the swarm is divided into overlapping neighborhoods of particles. For each neighborhood, a best particle is determined. The *gbest* PSO is a special case of *lbest* when the neighborhood is the entire swarm. In this paper, the *gbest* model is used.

Another best value that is tracked by the global version of the PSO is the overall best value (*gbest*), obtained so far by any particle in the population. The location of this overall best value is called y_g. This location is also tracked by PSO.

The PSO changes the velocity of each particle at each time step so that it moves toward its personal best and global best locations. The algorithm for implementing the global version of PSO is as follows [7]:

1. Initialize a population of particles with random positions and velocities on a d-dimensional problem space.
2. For each particle, evaluate the desired optimization fitness function of d variables.
3. Compare particle's fitness evaluation with particle's personal best value (*pbest*). If the current fitness function value is better than *pbest* , then set the *pbest* value equal to the current value, and the *pbest* location equal to the current location in the d-dimensional space.
4. Compare fitness evaluation with the population's overall previous best value. If the current value is better than the global best value (*gbest*), then set *gbest* to the current particle's value and set the global best position y_g to the current particle's position.
5. Change the velocity and position of the particle according to Equations (1) and (2), respectively.

$$v_i(t+1) = wv_i(t) + c_1r_1(t)\big(y_i(t) - x_i(t)\big) + c_2r_2(t)(y_g(t) - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

where w is the inertia weight, c_1 and c_2 are the acceleration constants, and $r_1(t)$ and $r_2(t)$ are random numbers generated in the range between 0 and 1. Velocity updates are also clamped to prevent them from exploding, thereby causing premature convergence. The values of these parameters used in this paper are given later in the section on experimental results and discussion.

6. Loop to Step 2 until a termination criterion is met. The criterion is usually a sufficiently good fitness or a maximum number of iterations. In this paper, a maximum number of iterations is used.

III. RELATD WORK IN PSO-BASED CLUSTERING

In this paper, the following notations are used:

- N_p denotes the number of image pixels to be clustered
- N_c denotes the number of clusters to be formed
- z_p denotes the p-th pixel
- m_j denotes the mean of cluster j
- C_j denotes the subset of pixel vectors that form cluster j
- $|C_j|$ denotes the number of pixels in cluster j

In this paper, the PSO based image clustering algorithm proposed in [8] is used. A single particle represents the N_c cluster means. Each particle x_i is constructed as $x_i = (m_{i1}, \ldots, m_{ij}, \ldots, m_{iNc})$ where m_{ij} refers to the j-th cluster centroid vector of the i-th particle. The quality of each particle is measured by the fitness function. The PSO-based clustering algorithm can be summarized below:

1. Initialize each particle to contain N_c randomly selected cluster means.
2. For $t = 1$ to t_{max} (maximum number of iterations)
   (a) For each particle i
      - For each pixel z_p

        Calculate $d(z_p, m_{ij})$ for all clusters C_{ij}

        Assign z_p to C_{ij} where

        $$d(z_p, m_{ij}) = min_{\forall c=1,\ldots,N_c}\{d(z_p, m_{ic})\}$$

        $d(z_p, m_{ij})$ represents the Euclidean distance between the p-th pixel z_p and the centroid of j-th cluster of particle i.
      - Calculate the fitness function $f(x_i(t),Z)$ where Z is a matrix representing the assignment of pixels to clusters of particle i.

   (b) Update the personal best and the global best positions.

   (c) Update the cluster centroids using Equations (1) and (2).

The fitness function proposed in [8,10] uses the following three evaluation criteria: quantization error, intra-cluster distance and inter-cluster separation.

The quantization error J_e is defined below:

$$J_e = \frac{\sum_{j=1}^{N_c}[\sum_{\forall Z_p \in C_j} d(Z_p, m_j)]/|C_j|}{N_c} \quad (3)$$

where $d(z_p,m_j)$ represents the Euclidean distance between the p-th pixel z_p and the centroid of j-th cluster m_j.

The intra-cluster distance is measured by $\bar{d}_{\max}$ which is defined in [8,10] as

$$\bar{d}_{\max}(Z, x_i) = max_{j=1,\dots,N_c}\{\sum_{\forall z_p \in C_{ij}} d(z_p, m_{ij})/|C_{ij}|\}. \quad (4)$$

In Equation (4), Z is a matrix representing the assignment of pixels to clusters of particle i. A smaller value of $\bar{d}_{\max}$ means that the clusters are more compact.

Another measure of quality is the inter-cluster separation. It is measured by the minimum Euclidean distance between any pair of clusters and is defined below:

$$d_{min}(x_i) = min_{\forall j1,j2,j1 \neq j2}\{d(m_{ij_1}, m_{ij_2})\}. \quad (5)$$

The above three criteria have been used by [10] to form the fitness function as shown in Equation (6).

$$f_1(x_i, Z) = w_1\bar{d}_{max}(Z, x_i) + w_2(z_{max} - d_{min}(x_i)) + w_3J_e \quad (6)$$

where w_1, w_2 and w_3 are user defined constants and determine the relative weights of intra-cluster distance ($\bar{d}_{max}$), inter-cluster separation (d_{min}) and quantization error (J_e) in the fitness function. z_{max} is the maximum pixel value in the image set, which is 255 for 8-bit grayscale image used in this paper. One objective of the fitness function in Equation (6) is to minimize the intra-cluster distance ($\bar{d}_{max}$) and the quantization error (J_e). This will make the clusters compact. Another objective is to maximize the inter-cluster separation (d_{min}) which means the clusters are well separated.

However, a recent research paper [11] has pointed out a problem with the use of quantization error J_e as defined by Equation (3) in data clustering. In Equation (3), for every cluster, it first calculates the average distance of the pixels of a cluster to its cluster centroid. Then, it takes the average distances of all clusters and calculates another average, which is denoted by J_e. In [11], Esmin *et al.* pointed out that a cluster with just one data vector would influence the final result as much as another cluster with many data vectors. For example, suppose that one of the particle's clusters has one data vector that is very close to the centroid, and another cluster has many data vectors that are not so close to the centroid. This is not a very good solution, but giving the same weight to the cluster with one data vector as the cluster with many data vectors can make it seem to be a good solution [11]. To solve this problem, [11] proposed another equation which gave a higher weighting to the cluster with many data vectors in the calculation of the fitness function. In this paper, the modified quantization error proposed by [11] is called the weighted quantization error J_{e2}.

$$J_{e2} = \{\sum_{j=1}^{N_c}[(\sum_{\forall z_p \in C_{ij}} d(z_p, m_{ij})/|C_{ij}|).(|C_{ij}|/N_o)]\}, \quad (7)$$

where N_o is the total number of data vectors to be clustered.

IV. Our Approach in PSO-based Clustering

Two new fitness functions are proposed in this paper. For the first new fitness function, J_{e2} will be used together with $\bar{d}_{\max}$ and d_{min}. In [11], the weighted quantization error J_{e2} was used alone to cluster three benchmark data sets from the UCI repository of Machine Learning Databases. It reported that the use of J_{e2} in clustering improved the performance when compared to data clustering using J_e alone. In this paper, J_{e2} is first used alone in clustering of natural images and its performance is compared to K-means by using the three evaluation criteria: intra-cluster distance ($\bar{d}_{max}$), quantization error (J_e) and inter-cluster separation (d_{min}) . When J_{e2} is used alone in the fitness function, it gives better result in quantization error when compared to K-means. However, K-means method provides better result in inter-cluster separation. To solve this problem and enhance the performance in inter-cluster separation of PSO-based clustering, this paper proposes that J_{e2} should not be used alone in the fitness function for PSO-based image clustering. This paper proposes a new fitness function similar to Equation (6) used by Omran *et al.* in [10] but replaces J_e by J_{e2}, as given by the equation below.

$$f_2(x_i, Z) = w_1\bar{d}_{max}(Z, x_i) + w_2(z_{max} - d_{min}(x_i)) + w_3J_{e2}, \quad (8)$$

where J_{e2} is given by Equation (7).

The proposed new fitness function in Equation (8) will improve the fitness function used by Esmin *et al.* in [11] (which uses weighted quantization error only) as shown by the experimental result in Section V. Equation (8) will also improve the fitness function used by Omran *et al.* in [10] (which uses Equation (6)) as [11] has shown that J_{e2} solves the problem of J_e in clustering. This paper shows that J_{e2} should be used together with $\bar{d}_{max}$ and d_{min} to obtain compact clusters and large inter-cluster separation.

The second new fitness function proposed in this paper uses the mean square-error (*MSE*) defined by

$$MSE = \frac{1}{n}\sum_{j=1}^{K}\sum_{z_p \in C_j}(z_p - m_j)^2, \quad (9)$$

where n is the total number of pixels in the image, z_p is the p-th pixel, K is the number of clusters, m_j is the centroid of the j-th cluster C_j . *MSE* is a measure of the compactness of the clusters [14] and represents the mean squared distance of the pixels from its associated cluster centroid.

It should be noted that *MSE* defined by Equation (9) does not have the problem of J_e as described in Section III above. A cluster with one data vector will not influence the result as much as another cluster with many data vectors. For example, if one of the particle's clusters has one pixel that is very close to the centroid, and another cluster has many pixels that are not so close to the centroid, the *MSE* in Equation (9) will correctly give a large error value.

Using *MSE* alone in PSO clustering will generally give good performance in $\bar{d}_{max}$ and J_e but slightly worse performance in d_{min} when compared to K-means, as shown by experimental results in Section V. To improve the performance in inter-cluster separation, MSE is used together with $\bar{d}_{max}$ and d_{min} in the fitness function f_3 below.

$$f_3(x_i, Z) = w_1 \bar{d}_{max}(Z, x_i) + w_2\big(z_{max} - d_{min}(x_i)\big) + w_3.MSE \quad (10)$$

V. Results and Discussion

The two new fitness functions f_2 and f_3 in Equations (8) and (10) are used in the PSO-based image clustering algorithm described in Section III. The PSO based image clustering algorithm has been applied to three grayscale images: Lena, Pepper and Airplane. The performance is measured by the following three criteria: intra-cluster distance ($\bar{d}_{max}$), quantization error (J_e) and inter-cluster distance (d_{min}). These three criteria have been used in [8] and [10]. The performance of PSO-based clustering is then compared to the K-means algorithm.

For all the experiments, the following parameters are used for PSO-based clustering:

- Number of particles = 20
- Number of iterations for termination = 150
- Number of clusters = 5
- Acceleration constants c_1 and c_2 = 2

The number of particles used is problem-dependent. The common choice of number of particles varies from 20 to 50 [7, 15]. In all experiments of this paper, 20 particles are used for PSO clustering as smaller number of particles can reduce computation time and 20 particles can provide good clustering performance in this paper when compared with K-means.

The number of clusters is chosen to be 5 for both K-means and PSO clustering to allow a fair comparison of their performance.

For the inertia weight w, the initial weight value is 0.9 and w decreases linearly with the number of iterations. The final value is 0.4 when the termination condition (150 iterations) is reached. By linearly decreasing the inertia weight from a relatively large value to a small value through the course of the PSO run, the PSO tends to have more global search ability at the beginning of run while having more local search ability near the end of the run [5,6]. The acceleration constants c_1 and c_2 are both set to 2. The settings of acceleration constants and the inertia weight are based on the recommendation by [6].

For K-means algorithm, the number of iterations is 3000. This is chosen to equal the number of fitness function evaluations in PSO based clustering (20 particles and 150 iterations will give 3000 fitness function evaluations).

For fitness functions f_2 and f_3 in PSO clustering, each fitness function consists of three sub-objectives. The weighting of each sub-objective (w_1, w_2 and w_3) that provides best performance is determined empirically in this paper. To eliminate the tuning of these weight values, multi-objective optimization approach can be used [10,16,17].

For all data shown in all tables (Tables I to V), they are the averages of 25 program runs. The numbers after the ± symbols in the tables represent the standard deviation.

TABLE I. PSO-based Clustering Using Weighted Quantization Error Only

Images	*Intra-cluster distance* $\bar{d}_{max}$	*Quantization error* J_e	*Inter-cluster distance* d_{min}
Pepper	10.9006 ±1.0487	9.6045±0.1473	30.2707±0.9067
Lena	10.2972±0.1532	8.4037±0.0047	27.0092±0.0307
Airplane	15.9973±0.0183	9.1213±0.0110	13.2569±0.3834

TABLE II. PSO-based Clustering Using Intra-Cluster Distance, Inter-Cluster Distance & Weighted Quantization Error with W_1=0.1, W_2=0.1, W_3=0.8

Images	*Intra-cluster distance* $\bar{d}_{max}$	*Quantization error* J_e	*Inter-cluster distance* d_{min}
Pepper	10.6303 ±0.1005	9.7574±0.0455	41.8520±0.3721
Lena	9.0262±0.1339	8.6304±0.0381	35.3901±0.4734
Airplane	11.2392±0.3263	9.9321±0.2124	41.6484±1.3212

TABLE III. PSO-based Clustering Using MSE Only

Images	*Intra-cluster distance* $\bar{d}_{max}$	*Quantization error* J_e	*Inter-cluster distance* d_{min}
Pepper	10.2567 ±0.2182	9.7898±0.0521	31.7022±0.5417
Lena	9.6387±0.1311	8.4393±0.0036	29.1630±0.1793
Airplane	15.8544±0.0388	9.7643±0.0114	18.7229±0.0941

TABLE IV. PSO-based Clustering Using Intra-Cluster Distance, Inter-Cluster Distance & MSE

Images	*Intra-cluster distance* $\bar{d}_{max}$	*Quantization error* J_e	*Inter-cluster distance* d_{min}
Pepper	10.2110 ±0.0009	9.7692±0.0007	33.8947±0.0226
Lena	9.3937±0.0555	8.4455±0.0009	30.8199±0.0210
Airplane	10.8355±0.0893	9.8273±0.0373	39.5477±0.4057

TABLE V. K-means Clustering

Images	*Intra-cluster distance* $\bar{d}_{max}$	*Quantization error* J_e	*Inter-cluster distance* d_{min}
Pepper	13.0798 ±1.9485	10.0068±0.2643	32.4692±0.1867
Lena	9.7053±0.4377	8.4432±0.0042	29.3819±0.1248
Airplane	15.6241±0.2511	9.8564±0.0986	20.0633±1.5214

A. *PSO-based Clustering Using* f_2 (*Using* $\bar{d}_{max}$, d_{min} *and* J_{e2})

Table I shows the result of PSO-based clustering on the three images: Pepper, Lena and Airplane. The fitness function only uses the weighted quantization error (J_{e2}) given in Equation (7). Table II shows the result of PSO-based clustering using the fitness function f_2 in Equation (8). The fitness function uses all the three evaluation criteria: intra-cluster distance ($\bar{d}_{max}$), inter-cluster distance (d_{min}) and the weighted quantization error (J_{e2}). The following weighting factors are used for all the three images: w_1 = 0.1, w_2 = 0.1, w_3 = 0.8. Table V shows the result of K-means clustering.

By comparing Tables I and V, when J_{e2} is used alone in fitness function of PSO clustering, PSO clustering has smaller quantization error (J_e) when compared to K-means for all the three images. However, its performance is worse in intra-cluster distance ($\bar{d}_{max}$) & inter-cluster separation (d_{min}) than K-means for Lena & Airplane. For Pepper, PSO is better in intra-cluster distance but worse in inter-cluster separation. To improve the performance on intra-cluster distance and inter-cluster distance, it is proposed in this paper that the weighted quantization error J_{e2} should not be used alone in PSO-based image clustering but J_{e2} should be used together with intra-cluster distance $\bar{d}_{max}$ and inter-cluster separation d_{min}, as in Equation (8). Table II shows the result by using $\bar{d}_{max}$, $d_{min,}$ and J_{e2} together with the relative weighting factor w_1=0.1, w_2=0.1 and w_3=0.8

By comparing Tables I and II, PSO clustering using $\bar{d}_{max}$, $d_{min,}$ and J_{e2} together can give more compact clusters (smaller $\bar{d}_{max}$) and larger inter-cluster separation for all the three images while the performance with respect to J_e is comparable.

By comparing Tables II and V, PSO clustering using $\bar{d}_{max}$, d_{min} and J_{e2} together can give better performance than K-means for image Pepper for all the three evaluation criteria. For images Lena and Airplane, PSO clustering has better performance than K-means with respect to $\bar{d}_{max}$ and d_{min} while PSO has slightly higher quantization error J_e than K-means. However, their performance with respect to J_e is still comparable. The big improvement of PSO clustering in $\bar{d}_{max}$ and d_{min} shows that by using $\bar{d}_{max}$, d_{min} and J_{e2} together, more compact clusters and larger inter-cluster separation can be achieved.

B. *PSO-based Clustering Using f_3 (Using $\bar{d}_{max}$, d_{min} & MSE)*

Table III shows the result of PSO clustering using the *MSE* alone in the fitness function. Table IV shows the result of PSO clustering using *MSE*, $\bar{d}_{max}$ and d_{min} together in fitness function f_3 (Equation (10)). For fitness function f_3, the following weighting factors are used:

For Pepper and Lena: $w_1 = 0.1$, $w_2 = 0.2$, $w_3 = 0.7$
For Airplane: $w_1 = 0.5$, $w_2 = 0.3$, $w_3 = 0.2$

By comparing Table III and V, PSO clustering using *MSE* alone has smaller quantization error and intra-cluster distance for both Pepper and Lena. For Airplane image, PSO clustering using *MSE* alone has smaller J_e but slightly higher $\bar{d}_{max}$ when compared to K-means. For the three images under test, PSO clustering using *MSE* alone generally can provide more compact clusters as the values for $\bar{d}_{max}$ and J_e are generally smaller. However, PSO clustering using *MSE* alone have smaller inter-cluster separation when compared to K-means. To improve the performance in inter-cluster separation, *MSE* will be used together with $\bar{d}_{max}$ and d_{min} in the fitness function f_3 (Equation (10)).

By comparing Tables III and IV, PSO clustering using $\bar{d}_{max}$, d_{min} and *MSE* together will give better performance in intra-cluster distance and inter-cluster separation when compared to PSO clustering using *MSE* alone. The performance in J_e is comparable for both methods.

By comparing Tables IV and V, PSO clustering using $\bar{d}_{max}$, d_{min} and *MSE* together have better performance than K-means for all the three images in all evaluation criteria except for Lena image, where the quantization error for the PSO clustering is only slightly worse than J_e for K-means (8.4455 compared to 8.4432). Hence, it can concluded that PSO clustering using $\bar{d}_{max}$, d_{min} and *MSE* together perform better than K-means by giving more compact clusters and larger inter-cluster separation.

C. *Comparison of Fitness Functions f_2 and f_3*

By comparing Tables II and IV, PSO clustering using f_2 has larger inter-cluster separation while Table IV (using f_3) shows better or comparable performance in quantization error when compared to Table II. It should be noted that the performance in each evaluation criterion can be changed by using different weighting factors w_1, w_2 and w_3. In this paper, the objective of setting these weighting factors is to give better performance over K-means in each of the evaluation criteria.

Based on the three images under test, PSO clustering using f_3 is more robust than PSO clustering using f_2. PSO clustering using f_3 can produce nearly same results over repeated runs when compared to clustering using f_2. In Table IV (using f_3), all data have very small standard deviation when compared to results in Table II (using f_2).

The key point in the experiments is that both PSO clustering methods, using either f_2 or f_3, can give more compact clusters and larger inter-cluster separation when compared to K-means.

Fig. 1 to Fig. 3 show the original images of Lena, Pepper and Airplane and their output images after clustering using PSO (using f_2 and f_3 in Equations (8) and (10)) and K-means. The number of clusters used in all experiments is 5. All original images are grayscale images with resolution 512 × 512. For all output images after clustering, the grey level values of the five cluster centroids are used to represent the pixels of the associated clusters.

VI. CONCLUSIONS

This paper has proposed two fitness functions that can improve PSO-based image clustering. A recent paper [11] has proposed to include the effect of number of data vectors inside a cluster when the quantization error is used in a fitness function for PSO clustering. This paper shows that when the modified quantization error proposed in [11], called weighted quantization error (J_{e2}) in this paper, is used alone in image clustering, its performance in inter-cluster distance is worse than K-means though it can give smaller quantization error. To solve this problem, the first proposed fitness function uses J_{e2} together with $\bar{d}_{max}$ and d_{min} to improve clustering quality. In the second proposed fitness function, the mean square-error is used together with $\bar{d}_{max}$ and d_{min}. Experimental results show that PSO-based image clustering, using the two proposed new fitness functions, can have more compact clusters and larger inter-cluster separation when compared to K-means clustering.

For future research, a PSO based automatic clustering algorithm will be developed that can determine the optimum number of clusters of the image, find the cluster centers and perform image clustering in the same program run.

(a) (b)

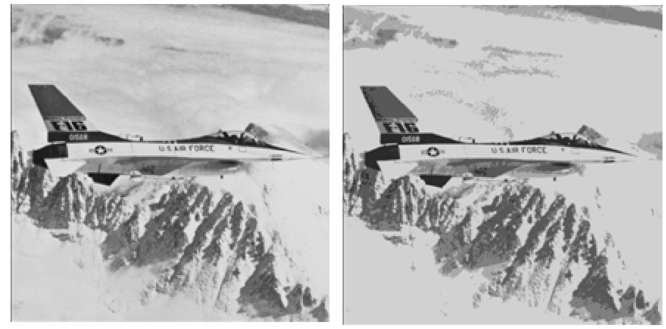
(c) (d)

Fig. 1. (a) Original Lena image; (b) Lena after PSO-based clustering using weighted quantization error, intra-cluster distance & inter cluster distance (5 clusters); (c) Lena after PSO-based clustering using MSE, intra-cluster distance & inter-cluster distance (5 clusters); (d) Lena after K-means clustering (5 clusters).

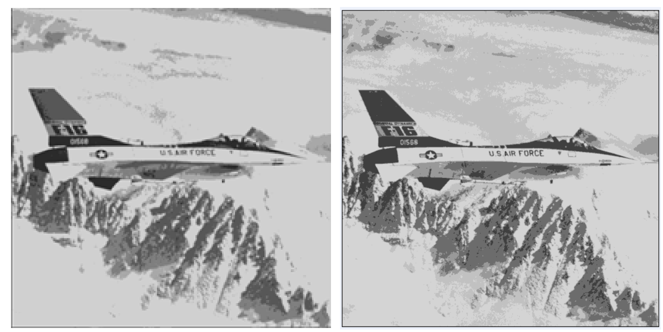
(a) (b)

(c) (d)

Fig. 2. (a) Original Pepper image; (b) Pepper after PSO-based clustering using weighted quantization error, intra-cluster distance & inter cluster distance (5 clusters); (c) Pepper after PSO-based clustering using MSE, intra-cluster distance & inter-cluster distance (5 clusters); (d) Pepper after K-means clustering (5 clusters).



(a) (b)



(c) (d)

Fig. 3. (a) Original Airplane image; (b) Airplane after PSO-based clustering using weighted quantization error, intra-cluster distance & inter cluster distance (5 clusters); (c) Airplane after PSO-based clustering using MSE, intra-cluster distance & inter-cluster distance (5 clusters); (d) Airplane after K-means clustering (5 clusters).

REFERENCES

[1] E. Gose, R. Johnsonbaugh, S. Jost, "Pattern recognition and image analysis," Prentice Hall, 1996.

[2] P. Tan, M. Steinbach, V. Kumar, "Introduction to data mining," Pearson Education, 2006.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of the IEEE International Joint Conference on Neural Networks, Perth, Australia, vol. 4, pp. 1942-1948, 1995.

[4] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," 6th International Symposium on Micro Machine and Human Science, 1995.

[5] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," Proceedings of IEEE International Conference on Evolutionary Computation, World Congress on Computational Intelligence, Anchorage, Alaska, 1998.

[6] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," Proceedings of the 1999 Congress on Evolutionary Computation (CEC 1999), Piscataway, NJ: IEEE Service Center, pp. 1945-1950, 1999.

[7] R. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001), IEEE Press, pp 81-86, 2001.

[8] M. Omran, A. Salman, A. Engelbrecht, "Image classification using particle swarm optimization," Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, Singapore, 2002.

[9] D. Van der Merwe and A. Engelbrecht, "Data clustering and particle swarm optimization," Proceedings of IEEE Congress on Evolutionary Computation (CEC 2003), Caribella, vol. 4, pp. 215-220, 2003.

[10] M. Omran, A. Engelbrecht, A. Salman, "Particle swarm optimization method for image clustering," International journal of Pattern Recognition and Artificial Intelligence, vol. 19, no. 3, pp. 297-322, 2005.

[11] A. A. A. Esmin, D. L. Pereira, F. P. A. de Araújo, "Study of different approach to clustering data by using particle swarm optimization

algorithm," Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2008), Hong Kong, China, 2008.

[12] A. K. Jain, M. N Murty, P. J. Flynn, "Data clustering: a review," ACM Computer Surv., vol. 31, no. 3, pp.264-323, Sep. 1999.

[13] S. Das, A. Abraham, A. Konar, "Automatic clustering using an improved differential evolution algorithm," IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 38, No. 1, Jan 2008.

[14] S. Ouadfel, M. Batouche, A. taleb-Ahmed, "A modified particle swarm optimization algorithm for automatic image clustering," Proceedings of the Int'l Symposium on Modelling and Implementation of Complex Systems, MISC 2010, pp. 49-57, May 2010, Algeria.

[15] R. Poli, J. Kennedy, T. Blackwell, "Particle swarm optimization: an overview," Swarm Intelligence, Vol. 1, No. 1, pp. 33-57, 2007.

[16] C. A. Coello-Coello, "An empirical study of evolutionary techniques for multiobjective optimization in engineering design," PhD thesis, Tulane University, 1996.

[17] C. A. Coello-Coello, M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," IEEE Congress on Evolutionary Computation (CEC 2002), Vol. 2, pp.1051-1056, 2002.