

# Hybrid Exponential Particle Swarm Optimization K-Means Algorithm for Efficient Image Segmentation

<sup>1</sup>K.M. Murugesan and <sup>2</sup>S. Palaniswami

<sup>1</sup>Department of ECE,

Government Polytechnic College, Kaniyalampatti-641044, India

<sup>2</sup>Department of Electrical and Electronics Engineering,

Government College of Technology, Coimbatore-641013, India

Received 2012-05-14, Revised 2012-10-16; Accepted 2012-10-25

## ABSTRACT

The introduction of unsupervised learning techniques like K-means inside the domain of Image Processing plays a vital role in Image Segmentation. The hybridization of this Algorithm by using Swarm Intelligent techniques further more improves the efficiency. Various works on hybridization of Particle Swarm Optimization (PSO) with K-means have been proposed and are found to be efficient in Image Segmentation. However, the PSO has a problem of getting stagnated with the local optima. This results in the degradation of the Image Segmentation process in most cases. The main reason behind this problem is the constancy of the inertia weight. The inertia weight when varied dynamically and exponentially could afford a better performance in the process of finding better global optima. We use the Exponential Particle Swarm Optimization to enhance the K-means Algorithm. This shows a drastic improvement in the process of Image Segmentation. The experimental results obtained also add to this process enhancement. The EPSO K-means Algorithm is much efficient compared to its previous types.

**Keywords:** Image Segmentation, Exponential Particle Swarm Optimization, Fuzzy Logic, PSO, EPSO, Global Best, Local Best

## 1. INTRODUCTION

Image segmentation is one of the utmost vital constituents of Digital Image Processing whose aim is to interpret the various kinds of images. Image segmentation techniques can be widely categorized as edge-based segmentation, region-oriented segmentation, histogram thresholding and clustering algorithms (Gonzalez and Woods, 1992). The main objective of using a clustering algorithm is to combine data into small groups such that the data in each group possess similar characteristics though the data clusters are distinct from each other. The most widely used methods to classify the structure of data is the K-means algorithm (Tou and Gonzalez, 1974). It is an unsupervised clustering technique which has a strong inclination towards the local minima while finding an optimal solution. Therefore, the distribution of initial cluster centers

drastically decides the process of clustering. Therefore, the determination of the good initial parameters is a challenging problem and hence the clustering algorithms necessitate more number of experimentation to decide the input parameters for the optimal or suboptimal clustering results.

Competitive learning model (Rumelhart and Zipser, 1987) is a remarkable and potential learning algorithm which can be applied in unsupervised training for image classification (Hung, 1993). Simple Competitive Learning (SCL) shows steadiness over various run trials but this stable outcome is not always the global optima. In some cases the SCL sticks to the local optima over all run trials and the learning rate needs to be adjusted in the duration of experimentation so that the global optimization can be achieved efficiently. There are numerous techniques, developed for optimization, based on the behaviors of natural systems

**Corresponding Author:** K.M. Murugesan, Department of ECE, Government Polytechnic College, Kaniyalampatti-641044, India

(Pham and Karaboga, 2000). Swarm Intelligence (SI) which includes Ant Colony Optimization (ACO) (Dorigo *et al.*, 1996) and Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) has been introduced in the literature as optimization techniques. There are several SI methods for clustering of data in the literature which use classical clustering techniques such as K-means algorithm. In most of these approaches ACO or PSO are used to find the initial cluster centers for the K-means algorithm. We propose a hybrid algorithm which combines Exponential Particle Swarm Optimization (EPSO) with K-means.

Our main objective is to make segmentation results of K-means less dependent on the initial cluster centers. Hence, the results obtained are more accurate and stable by applying EPSO optimization technique. This improvement is mainly due to the large search space provided by this technique. Our method considers both spatial and spectral features of the image and thereby produces results with improved accuracy. In this study we will study the hybridization of EPSO with K-means. A detailed comparison study on EPSO K-means, PSO K-means and K-means will also be provided.

### 1.1. K-Means Clustering Algorithm

The K-means algorithm (McQueen, 1967) is an unsupervised clustering algorithm which partitions a set of data, usually termed dataset into a certain number of clusters. Minimization of a performance index is the primary basis of K-means Algorithm, which is defined as the sum of the squared distances from all points in a cluster domain in the cluster center (Tou and Gonzalez, 1974). Initially K random cluster centers are chosen. Then, each sample in the sample space is assigned to a cluster based on the minimum distance to the center of the cluster. Finally the cluster centers are updated to the average of the values in each cluster. This is repeated until cluster centers no longer change. Steps in the K-Means algorithm are:

- Step 1: Initialize K initial cluster centers randomly.
- Step 2: For each pixel, calculate the distance to the cluster centers and assign the pixel to a cluster which has the minimum distance to its center.
- Step 3: Calculate the average of the pixel values in each cluster and take them as new cluster centers.
- Step 4: Repeat steps 2 and 3 until new cluster centers converge to the previous ones.

The prime motive of the K-means Algorithm is to find the local minima rather than the global minima. Therefore, the algorithm severely sticks to the initial

choice of cluster centers and distribution of data. The results become acceptable most of the time when the initial cluster centers are selected relatively far away. It is because; the main clusters in a given data are commonly differentiated in such a manner. If the main clusters in a given data are too close to one another in the sample space, the K-means algorithm fails to identify these clusters. For its improvement the K-means algorithm needs to be enhanced with some optimization technique in order to be less reliant on a given dataset and initial cluster centers. The output for the K-means with respect to Image Segmentation is shown in **Fig. 1**.

### 1.2. Particle Swarm Optimization

The PSO algorithm is inspired by the group behavior of schools of fish, flocks of birds and swarms of insects. As an example, birds are likely to find food in flocks, without knowing its location in advance. The PSO algorithm consists of a swarm of particles flying through the search space (Kaewkamnerdpong and Bentley, 2005). Each particle's position is a potential solution to the problem. Each particle's velocity is modified based on its distance from its personal best position and the global best position. In other words the particles move according to their experience and that of their neighbors which yields to the best fitness value. Each particle  $i$  maintains the following information (Merwe and Engelbrecht, 2003):

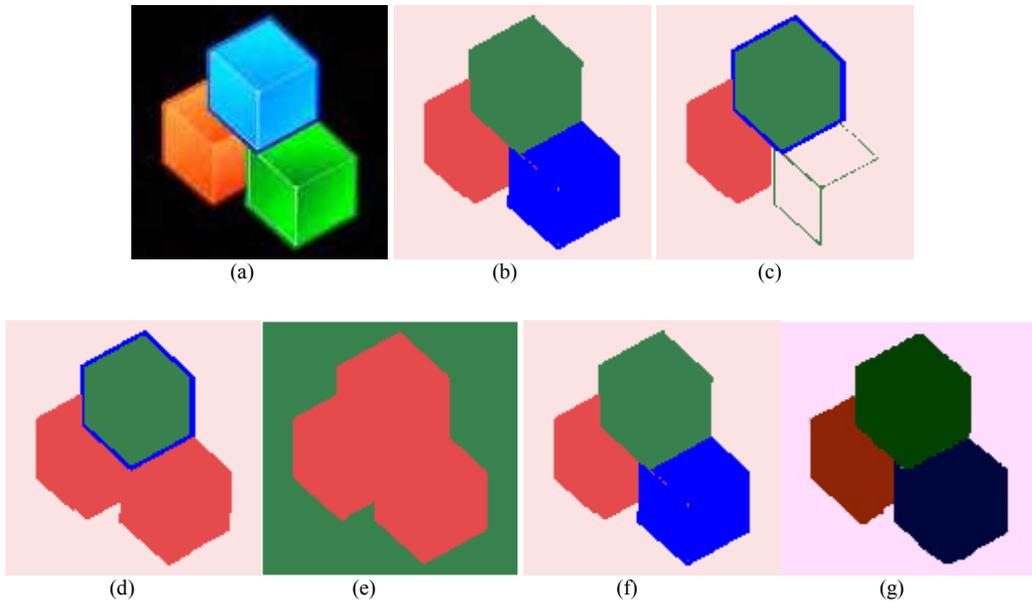
- $x_i$  = The current position of the particle
- $v_i$  = The current velocity of the particle
- $y_i$  = The personal best position of the particle (pbest); the best position visited so far by the particle and
- $\hat{y}$  = The global best position of the swarm (gbest); the best position visited so far by the entire swarm.

The objective function evaluates the positions of the particles. Personal best position (pbest) is then obtained through Eq. 1 which is as follows:

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } (x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } (x_i(t+1)) < f(y_i(t)) \end{cases} \quad (1)$$

where,  $f$  is the objective function. The global best position (gbest) is obtained by Eq. 2 which is as follows (Merwe and Engelbrecht, 2003):

$$\hat{y}(t) \in \{y_0, y_1, y_2, \dots, y_s\} = \min \{f(y_0(t)), f(y_1(t)), \dots, f(y_s(t))\} \quad (2)$$



**Fig. 1.** The most occurred results out of 30 trials (a) Original image, (b-e) K-means, (f) Hybrid PSO K-means, (g) Hybrid EPSO K-means

For each iteration of a PSO algorithm,  $v_i$  and  $x_i$  are updated by Eq. 3 and 4 which are as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(t)(y_i(t) - x_i(t)) + c_2 r_2(t)(\hat{y}(t) - x_i(t)) \tag{3}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{4}$$

where,  $\omega$  is the inertia weight which serves as a memory of previous velocities. The inertia weight controls the impact of the previous velocity. The cognitive component  $y_i(t) - x_i(t)$ , represents the particle's own experience as to where the best solution is. The social component,  $\hat{y}(t) - x_i(t)$  represents the belief of the entire swarm as to where the best solution is.  $c_1$  and  $c_2$  are acceleration constants and  $r_1(t), r_2(t) \sim U(0,1)$  is a random number between 0 and 1.

The PSO algorithm is repeated until a termination criterion is reached or the changes in velocity get near to zero. A fitness function is used to evaluate the optimality of the solution. The following algorithm outlines a PSO based image classification (Omran *et al.*, 2002). In this algorithm, a single particle  $x_i$  represents  $N$  cluster means such that  $x_i = (m_{i1}, \dots, m_{ij}, \dots, m_{iN})$ , where  $m_{ij}$  represents the  $j$ -th cluster centroid vector of the  $i$ -th particle. Therefore, a swarm represents a number of

candidate cluster centers. The fitness of each set of cluster is measured using Eq. 5:

$$f(x_i, Z_i) = \omega_1 \bar{d}_{\max}(Z_i, x_i) + \omega_2 (z_{\max} - d_{\min}(x_i)) \tag{5}$$

where,  $z_{\max} = 2s-1$  for an  $s$ -bit image;  $Z$  is a matrix representing the assignment of pixels to clusters of particle  $i$ . Each element  $z_{ijp}$  indicates if pixel  $z_p$  belongs to cluster  $C_{ij}$  of particle  $i$ . The constants  $\omega_1$  and  $\omega_2$  are user defined constants. Also:

$$\bar{d}_{\max}(Z_i, x_i) = \max_{j=1, \dots, N_c} \left\{ \sum_{z_p \in C_{ij}} d(z_p, m_{ij}) / |C_{ij}| \right\} \tag{6}$$

Equation 6 shows the maximum average Euclidean distance of particles to their associated clusters and:

$$d_{\min}(x_i) = \min_{\forall j_1, j_2, j_1 \neq j_2} \{d(m_{ij_1}, m_{ij_2})\} \tag{7}$$

Equation 7 shows the minimum Euclidean distance between any pair of clusters. The algorithm is as follows:

- Step 1: Initialize cluster centers for each particle randomly.
- Step 2: For each particle, assign each pixel to a cluster that has the minimum distance to its cluster center.

- Step 3: Calculate the fitness function for each particle and find the global best solution.
- Step 4: Update the cluster centers using Eq. 3 and 4.
- Step 5: Repeat the procedure until the stopping criterion is reached.

### 1.3. Exponential Particle Swarm Optimization (EPSO):

In linear PSO, the particles tend to fly towards the global best position (gbest) of the particles for which it is found. This would help to discover the optimal and best solution rapidly. However, at some instance, this collaboration sticks with the local optima and fails to converge with the global optimum. Once a global best position is found, it is communicated over the neighboring particles immediately. As a result, all particles are attracted to this position in the subsequent iterations until another better solution is found. Thus the stagnation is caused due to the overall speed diffusion of the newly found global best position. An improvement is made in the PSO by not keeping  $\omega$  constant throughout the execution. Starting from the maximal value, it is linearly decremented as the number of iterations increases down to a minimal value. Initially set to  $\omega_{max}$ , decreasing to  $\omega_{min}$  over the initial  $n$  iterations. If the iterations are above  $n$  and remaining  $\omega_{min}$  over the remainder of the run according to Eq. 8:

$$\omega = (\omega - \omega_{min}) \left( \frac{n_{max} - n}{n_{max}} \right) + \omega_{min} \tag{8}$$

$n_{max}$  represents the maximum number of iterations.

EPSO greatly influences the global and local exploration and helps greatly in bringing out the search behavior quickly and intelligently as it does not encourage the particles to be stagnant with the local optima. Instead, it varies the inertia weight exponentially as given in Eq. 9:

$$\omega = (\omega - \omega_{min}) e^{\frac{n_{max} - n}{n_{max}} - 1} + \omega_{min} \tag{9}$$

### 1.4. Hybrid PSO K-Means Algorithm (Saatchi and Hung, 2007)

The PSO based clustering (Omran *et al.*, 2002) initializes the cluster centers assigned to particles in a random fashion. Each pixel is distributed within a cluster with minimal Euclidean distance. Particle Swarm Optimization is used to refine the cluster centers using

fitness functions. The K-means algorithm is used to feed one particle of the initial swarm [Engelbrecht, 03] and the rest of the swarm is initialized randomly. In the hybridization of PSO-K-means algorithm, the K-means is applied to all particles and solutions are obtained in a way almost like that of the evaluation used in other hybridized K-means algorithm.

The PSO K-means algorithm is as presented below:

- Step 1: Initialize the number of clusters to  $K$  and number of particles to  $m$ .
- Step 2: Initialize  $m$  sets of  $K$  random cluster centers to be used by  $m$  particles.
- Step 3: For each particle, let each pixel  $x$  belong to a cluster in which it has the smallest Euclidean distance to the centroid.
- Step 4: Calculate new cluster centers; if the new cluster centers converge to the old ones, go to the next step. Otherwise, go to Step 3.
- Step 5: Save the best solution found so far performed by each particle. Call it *pbest* or personal best solution.
- Step 6: Save the best solution among the  $m$  personal best solutions found. Call it *gbest* or global best solution.
- Step 7: Update cluster centers of each particle according to the cluster center values of the *pbest* and *gbest* solution, using Eq. 3 and 4.
- Step 8: If the termination criterion is satisfied go to next step. Otherwise, go to Step 3.
- Step 9: Output the optimal solution.

### 1.5. Hybrid EPSO K-Means Algorithm

The usage of Exponential Particle Swarm Optimization with K-means algorithm is similar to that of PSO K-means Algorithm. This provides a further more refined cluster centers using fitness functions making the system to be free from the inclination towards local optima. The role of EPSO is to assign each pixel to a cluster. The steps involved in EPSO K-means Algorithm are similar to the PSO K-means Algorithm. The variation can be observed only in the internal functionality of the usage of  $\omega$ , since it varies dynamically to obtain the best form of clustering.

The PSO K-means algorithm is as presented below:

- Step 1: Initialize the number of clusters to  $K$  and number of particles to  $m$ .
- Step 2: Initialize  $m$  sets of  $K$  random cluster centers to be used by  $m$  particles.

- Step 3: For each particle, let each pixel  $x$  belong to a cluster in which it has the smallest Euclidean distance to the centroid.
- Step 4: Calculate new cluster centers; if the new cluster centers converge to the old ones, go to the next step. Otherwise, go to Step 3.
- Step 5: Save the best solution found so far performed by each particle. Call it  $pbest$  or personal best solution.
- Step 6: Save the best solution among the  $m$  personal best solutions found. Call it  $gbest$  or global best solution.
- Step 7: Update cluster centers of each particle according to the cluster center values of the  $pbest$  and  $gbest$  solution, using Eq. 3 and 4 where  $\omega$  is given by (9).
- Step 8: If the termination criterion is satisfied go to next step. Otherwise, go to Step 3.
- Step 9: Output the optimal solution.

The outcome of the usage of EPSO K-means Algorithm is shown in **Fig. 1d**.

## 2. MATERIALS AND METHODS

The experimentation is done using Matlab R2009 using a PC running in Windows Vista Operating System, with Dual Core Processor, 4 GB RAM and 3.16 GHz processor. The sample image is taken and is processed for K-Means, Hybrid PSO K-means and Hybrid EPSO K-means.

## 3. RESULTS AND DISCUSSION

### 3.1. Simulation Results

Experimental results from our proposed EPSO K-means Algorithm are compared with the traditional K-means and PSO K-means Algorithm. The PSO K-means include the parameters  $c_1$ ,  $c_2 = 1.49$  and  $\omega = 0.72$ . In case of EPSO K-means algorithm, the parameters  $c_1$  and  $c_2$  are same as that of PSO K-means Algorithm. The parameter  $\omega$  varies between 0.99 and 0.72. That is,  $\omega_{max} = 0.99$  and  $\omega_{min} = 0.79$ . The algorithms are examined for cubes in which 30 different trials are made to find the most occurred result.

It could be visualized in **Fig. 1** from the above images that the output obtained as a result of Hybrid EPSO K-means Algorithm is much more accurate than those obtained from the other schemes. The PSO K-means Algorithm shows a little deflection in the cube at

the right bottom in terms of segmentation. This is caused due to the stagnation of the swarm towards the local optima. The same is overcome in case of the Hybrid EPSO K-means Algorithm. Thus the stability of EPSO K-means is higher compared to that of PSO K-means Algorithm and ordinary K-means. Thus the EPSO K-means proves to be globally optimal than the PSO K-means algorithm.

## 4. CONCLUSION

From the experimental results, it is clearly visible that the usage of Swarm Intelligence techniques improves the performance of K-means Algorithm which greatly enhances the unsupervised learning process. The introduction of PSO K-means itself provides a larger search space compared to the K-means. The EPSO K-means enhances this feature to obtain a much better result. The EPSO K-means algorithm is less dependent on the randomly chosen initial points and is more efficient in finding the global optima.

## 5. REFERENCES

- Dorigo, M., V. Maniezzo and A. Colomi, 1996. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybernetics*, 26: 29-41. DOI: 10.1109/3477.484436
- Gonzalez, R.C. and R.E. Woods, 1992. *Digital Image Processing*. 3rd Edn., Addison-Wesley, Reading, ISBN-10: 0201508036, pp: 528.
- Hung, C.C., 1993. Competitive learning networks for unsupervised training. *Int. J. Remote Sens.*, 14: 2411-2415. DOI: 10.1080/01431169308954045
- Kaewkamnerdpong, B. and P.J. Bentley, 2005. Perceptive particle swarm optimisation: An investigation. *Proceedings of the IEEE Swarm Intelligence Symposium*, Jun. 8-10, IEEE Xplore Press, pp: 169-176. DOI: 10.1109/SIS.2005.1501618
- Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, Nov. 27-Dec. 01, IEEE Xplore Press, Perth, WA., pp: 1942-1948. DOI: 10.1109/ICNN.1995.488968
- McQueen, J.B., 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, (MSP' 67), University of California Press, Berkeley, pp: 281-297.

- Merwe, D.W.V.D. and A.P. Engelbrecht, 2003. Data clustering using particle swarm optimization. Proceedings of the 2003 Congress on Evolutionary Computation, Dec. 8-12, IEEE Xplore Press, pp: 215-220. DOI: 10.1109/CEC.2003.1299577
- Omran, M., A. Salman and A.P. Engelbrecht, 2002. Image classification using particle swarm optimization. Comput. Inform. Sci., 3: 397-406.
- Pham, D.T. and D. Karaboga, 2000. Intelligent Optimisation Techniques. 1st Edn., Springer, New York, ISBN-10: 1852330287, pp: 302.
- Rumelhart, D.E. and D. Zipser, 1987. Feature Discovery by Competitive Learning. In: Parallel Distributed Processing, Vol. 2: Psychological and Biological Models, McClelland, J.L. and D.E. Rumelhart (Eds.), MIT Press, Cambridge, ISBN-10: 0262631105, pp: 151-193.
- Saatchi, S. and C.C. Hung, 2007. Swarm intelligence and image segmentation swarm intelligence. ARS J.
- Tou, J.T. and R.C. Gonzalez, 1974. Pattern Recognition Principles. 1st Edn., Addison-Wesley, London, ISBN-10: 0201075865, pp: 400.