

Spatial Information Based Image Segmentation Using a Modified Particle Swarm Optimization Algorithm

Swagatam Das, Ajith Abraham* and Amit Konar

Dept. of Electronics and Telecommunication Engg, Jadavpur University,
Kolkata 700032, India

*IITA Professorship Program, School of Computer Science and Engineering,
Chung Ang University (CAU), Seoul

E-mail: ajith.abraham@ieee.org, swagatamdas19@yahoo.co.in, konaramit@yahoo.co.in

Abstract

This article proposes a particle swarm based segmentation algorithm for automatically grouping the pixels of an image into different homogeneous regions. In contrast to most of the existing evolutionary image segmentation techniques, we have incorporated spatial information into the membership function for clustering. The spatial function is the summation of the membership function in the neighborhood of each pixel under consideration. The two very important advantages of the new method are: 1) It does not require a priori knowledge of the number of partitions in the image and 2) It yields regions, more homogeneous than the existing methods even in presence of noise.

1. Introduction

Image segmentation may be defined as the process of dividing an image into disjoint homogeneous regions. These homogeneous regions usually contain similar objects of interest or part of them. The extent of homogeneity of the segmented regions can be measured using some image property (e. g. pixel intensity [1]). On the other hand, clustering can be defined as the optimal partitioning of a given set of n data points into c subgroups, such that data points belonging to the same group are as similar to each other as possible whereas data points from two different groups share the maximum difference.

Image segmentation can be treated as a clustering problem where the features describing each pixel correspond to a pattern, and each image region (i.e. a segment) corresponds to a cluster [1]. Therefore many clustering algorithms have widely been used to solve

the segmentation problem (e.g., K-means [2], FCM [3], ISODATA [4] and Snob [5]).

The fuzzy c-means (FCM) [6] seems to be the most popular algorithm in the field of fuzzy clustering. Many researchers have attempted modifications of the classical FCM and applications to image segmentation in the past few years [7-12].

Nevertheless, most of the existing clustering algorithms assume a priori knowledge of the number of classes, c , while in many practical situations, the appropriate number of classes is unknown or impossible to determine even approximately.

Finding an optimal number of clusters in a large dataset is usually a challenging task. Several researchers [13, 14] have investigated the problem. However, the outcome is still unsatisfactory [15]. Works on automatic clustering with evolutionary strategies (ES), evolutionary programming (EP) and variable string-length genetic algorithm (VGA) have been reported in [16-18].

Compared to a huge number of clustering algorithms based on different kinds of evolutionary algorithms, not much work has been reported on algorithms like Particle Swarm Optimization (PSO) or Differential Evolution (DE) for clustering image pixels [19, 20]. To the best of our knowledge, PSO has not been applied to the automatic fuzzy clustering till date.

In this paper, we have used a modified version of the basic PSO algorithm for the fuzzy segmentation of images. Since the pixels in an image generally have high correlation, we incorporated the spatial information of each pixel in its membership grades in different clusters. We borrowed the idea of spatial information from [21], which used it only for FCM algorithm. However, the automatic clustering scheme as well as the multi-elitist PSO model are novel. We provide comparison among our method and a very recent automatic fuzzy clustering technique [22] in

terms of final clustering accuracy, speed of obtaining an acceptable solution and algorithm robustness.

2. The Fuzzy Clustering Problem

Let $P = \{P_1, P_2, \dots, P_n\}$ be a set of n patterns or data points, each having d features. These patterns can also be represented by a profile data matrix $X_{n \times d}$ having n d -dimensional row vectors. The i^{th} row vector \vec{X}_i characterises the i^{th} object from the set P and each element X_{ij} in \vec{X}_i corresponds to the j^{th} real value feature ($j = 1, 2, \dots, d$) of the i^{th} pattern ($i = 1, 2, \dots, n$).

Given such an $X_{n \times d}$, a partitional clustering algorithm tries to find a partition $C = \{C_1, C_2, \dots, C_c\}$ such that the similarity of the patterns in the same cluster C_i is maximum and patterns from different clusters differ as far as possible. The partitions should maintain the following properties:

- 1) $C_i \neq \emptyset \quad \forall i \in \{1, 2, \dots, c\}$
- 2) $C_i \cap C_j = \emptyset, \quad \forall i \neq j \text{ and } i, j \in \{1, 2, \dots, c\}$
- 3) $\bigcup_{i=1}^c C_i = P$

In the case of fuzzy clustering, a pattern may belong to all the classes with a certain fuzzy membership grade for each class. So, in this case we need to evolve an appropriate partition matrix $U = [u_{ij}]_{c \times n}$, where $u_{ij} \in [0, 1]$, such that u_{ij} denotes the grade of membership of the j^{th} element to the i^{th} cluster. In fuzzy partitioning of the data, the following conditions hold:

$$\begin{aligned}
 0 < \sum_{j=1}^n u_{ij} < n & \quad \text{for } i = 1, 2, \dots, c. \\
 \sum_{i=1}^c u_{ij} = 1 & \quad \text{for } j = 1, 2, \dots, n. \\
 \text{and} \\
 \sum_{i=1}^c \sum_{j=1}^n u_{ij} = n & \quad (1)
 \end{aligned}$$

2.1 The Fuzzy c-means algorithm

In the classical fuzzy c-means (FCM) algorithm, a *within cluster sum* function J_m is minimized to evolve the proper cluster centers:

$$J_m = \sum_{j=1}^n \sum_{i=1}^c (u_{ij})^m \|\vec{X}_j - \vec{V}_i\|^2 \quad (2)$$

where \vec{V}_i is the i^{th} cluster center, \vec{X}_j is the j^{th} d -dimensional data vector and $\|\cdot\|$ is an inner product-

induced norm in d dimensions. Here m ($m > 1$) is any real number that influences the membership grade. Given c classes, we can determine their cluster centers \vec{V}_i for $i=1$ to c using the following expression:

$$\vec{V}_i = \frac{\sum_{j=1}^n (u_{ij})^m \vec{X}_j}{\sum_{j=1}^n (u_{ij})^m} \quad (3)$$

Now differentiating the performance criterion with respect to \vec{V}_i (treating u_{ij} as constants) and with respect to u_{ij} (treating \vec{V}_i as constants) and setting them to zero the following relation can be obtained:

$$u_{ij} = \left[\frac{\sum_{k=1}^c \left(\frac{\|\vec{X}_j - \vec{V}_k\|^2}{\|\vec{X}_j - \vec{V}_i\|^2} \right)^{1/(m-1)}}{\sum_{k=1}^c \left(\frac{\|\vec{X}_j - \vec{V}_k\|^2}{\|\vec{X}_j - \vec{V}_i\|^2} \right)^{1/(m-1)}} \right]^{-1} \quad (4)$$

2.2 Cluster Validity Indices in the Fuzzy Environment

To judge the quality of a partition provided by some clustering algorithm, it is necessary to have a well-defined function, called a *cluster validity index*, evaluated on the final clustering solutions. Below, we describe the two well-known validity indices used in the experiments reported here.

2.2.1 Xie-Beni Index

This index, due to Xie and Beni [23], is given by:

$$XB_m = \frac{\sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \|\vec{X}_j - \vec{V}_i\|^2}{n \times \min_{i \neq j} \|\vec{V}_i - \vec{V}_j\|^2} \quad (5)$$

Using XB_m the optimal number of clusters can be obtained by minimizing the index value.

2.2.2 Partition Entropy

The partition entropy function [1] is given by,

$$V_{pe} = \frac{- \sum_{j=1}^n \sum_{i=1}^c [u_{ij} \log u_{ij}]}{n} \quad (6)$$

The idea of the validity function is that the partition with less fuzziness means better performance. Consequently, the best clustering is achieved when the value V_{pe} is minimal.

2.3 The Spatial Information

An important characteristic of an image is the high degree of correlation among the neighboring pixels. In other words, these neighboring pixels possess similar feature values, and the probability that they belong to the same cluster is great. This spatial relationship is important in clustering, but it is not utilized in a standard FCM algorithm. To exploit the spatial information, a spatial function is defined as:

$$h_{ij} = \sum_{k \in \delta(\bar{X}_j)} u_{ik} \quad (7)$$

where $\delta(\bar{X}_j)$ represents a square window centered on pixel (i.e. data point) \bar{X}_j in the spatial domain. A 5×5 window was used throughout this work. Just like the membership function, the spatial function h_{ij} represents the probability that pixel \bar{X}_j belongs to i^{th} cluster. The spatial function of a pixel for a cluster is large if the majority of its neighborhood belongs to the same cluster. We incorporate the spatial function into membership function as follows:

$$u'_{ij} = \frac{u_{ij}^r h_{ij}^t}{\sum_{k=1}^c u_{kj}^r h_{kj}^t} \quad (8)$$

Here in all the cases we have used $r = 1$, $t = 1$ after considerable trial and errors.

3. The Multi-elitist PSO Model (MEPSO)

In PSO [23], a population of particles is initialized with random positions:

$$\bar{Z}_i(t) = [Z_{i,1}(t), Z_{i,2}(t), \dots, Z_{i,d}(t)]$$

and velocities:

$$\bar{v}_i(t) = [v_{i,1}(t), v_{i,2}(t), \dots, v_{i,d}(t)]$$

in d -dimensional space. A fitness function, f is evaluated, using the particle's positional coordinates as input values. Positions and velocities are adjusted, and the function is evaluated with the new coordinates at each time-step. The velocity and position update equations for the p^{th} dimension of i^{th} particle in the swarm may be given as follows:

$$\begin{aligned} v_{ip}(t+1) &= \omega \cdot v_{ip}(t) + C_1 \cdot \phi_1 \cdot (P_{1ip} - Z_{ip}(t)) + \\ &\quad C_2 \cdot \phi_2 \cdot (P_{gp} - Z_{ip}(t)) \\ Z_{ip}(t+1) &= Z_{ip}(t) + v_{ip}(t+1) \end{aligned} \quad (9)$$

The variables ϕ_1 and ϕ_2 are random positive numbers, drawn from a uniform distribution, and with an upper limit ϕ_{\max} , which is a parameter of the system. C_1 and C_2 are called acceleration constants, and ω is

the inertia weight. P_{1i} is the best solution found so far by an individual particle, while P_g represents the fittest particle found so far in the entire community. The canonical PSO has been subjected to empirical and theoretical investigations by several researchers [24, 25]. In many occasions, the convergence is premature, especially if the swarm uses a small inertia weight ω or constriction coefficient [25]. As the global best found early in the searching process may be a poor local minima, a multi-elitist strategy is proposed for searching the global best of the PSO. We call the new variant of PSO the MEPSO. The idea draws inspiration from the work reported in [26]. We define a growth rate β for each particle. When the fitness value of a particle of t^{th} iteration is higher than that of a particle of $(t-1)^{\text{th}}$ iteration, β will be increased. After the local best of all particles are decided in each generation, we move the local best, which has higher fitness value than the global best into the candidate area. Then the global best will be replaced by the local best with the highest growth rate β . Therefore, the fitness value of the new global best is always higher than the old global best. Pseudo code for MEPSO is as follows:

```

For  $t=1$  to  $t_{\max}$ 
  For  $j=1$  to  $N$  // swarm size is  $N$ 
    If (the fitness value of particle $_j$  in  $t$ -th time-step >
      that of particle $_j$  in  $(t-1)$ -th time-step)
       $\beta_j = \beta_j + 1$ ;
    End
    Update Local best $_j$ .
    If (the fitness of Local best $_j$  > that of Global best
      now)
      Choose Local best $_j$  put into candidate area.
    End
  End
  Calculate  $\beta$  of every candidate, and record the
  candidate of  $\beta_{\max}$ .
  Update the Global best to become the candidate of
   $\beta_{\max}$ .
  Else
    Update the Global best to become the particle of
    highest fitness value.
  End
End

```

4. The MEPSO Based Automatic Clustering Algorithm

4.1 Particle Representation

In the proposed method, for n data points, each p -dimensional, and for a user-specified maximum number of clusters c_{\max} , a particle is a vector of real numbers of dimension $c_{\max} + c_{\max} \times p$. The first c_{\max}

entries are positive floating-point numbers in (0, 1), each of which controls whether the corresponding cluster is to be activated (i.e. to be really used for classifying the data) or not. The remaining entries are reserved for c_{\max} cluster centers, each p -dimensional. A single particle is illustrated as:

$$\vec{Z}_i(t) =$$

| | | | | | | | |
|-----------|-----------|---------|------------------|---------------------|---------------------|---------|--|
| $T_{i,1}$ | $T_{i,2}$ | \dots | $T_{i,c_{\max}}$ | $\vec{V}_{i,1}$ | $\vec{V}_{i,2}$ | \dots | $\vec{V}_{i,c_{\max}}$ |
| | | | | flag _{i,1} | flag _{i,2} | | flag _{i,k_{ma}} <small>x</small> |

⏟

Activation Threshold

⏟

Cluster Centroids

Every probable cluster center $m_{i,j}$ has p features and a binary $\text{flag}_{i,j}$ associated with it. The cluster center is active (i.e., selected for classification) if $\text{flag}_{i,j} = 1$ and inactive if $\text{flag}_{i,j} = 0$. Each flag is set or reset according to the value of the activation threshold $T_{i,j}$. Note that these flags are latent information associated with the cluster centers and do not take part in the PSO-type mutation of the particle. The rule for selecting the clusters specified by one particle is:

$$\text{If } T_{i,j} > 0.5 \text{ Then } \text{flag}_{i,j} = 1 \text{ Else } \text{flag}_{i,j} = 0 \quad (10)$$

Note that the flags in an offspring are to be changed only through the T_{ij} s (according to the above rule). When a particle jumps to a new position, according to (9), the T values are first obtained which then are used to select (via (10)) the m values. If due to mutation some threshold T in a particle exceeds 1 or becomes negative, it is fixed to 1 or zero, respectively. However, if it is found that no flag could be set to one in a particle (all activation thresholds are smaller than 0.5), we randomly select 2 thresholds and re-initialize them to a random value between 0.5 and 1.0. Thus the minimum number of possible clusters is always 2.

4.2 Fitness Function

The quality of a partition can be judged by an appropriate cluster validity index. In the present work we have based our fitness function on the Xie-Benni index described in (5). The fitness function may be written as

$$f = \frac{1}{XB_i(c) + \text{eps}} \quad (11)$$

where XB_i is the Xie-Benni index of the i -th particle and eps is a very small constant (we used 0.0002). So maximization of this function means minimization of the XB index.

4.3 Avoiding Erroneous particles with Empty Clusters or Unreasonable Fitness Evaluation

There is a possibility that in our scheme, during computation of the XB index, a division by zero may be encountered. This may occur when one of the selected cluster centers is outside the boundary of distributions of the data set. To avoid this problem we first check to see if any cluster has fewer than 2 data points in it. If so, the cluster center positions of this special chromosome are re-initialized by an average computation. We put n/c data points for every individual cluster center, such that a data point goes with a center that is nearest to it.

4.4 Putting it All Together

The clustering method proposed here, is a two-pass process at each iteration or time step. The first pass amounts to calculating the active clusters as well as the membership functions for each particle in the spectral domain. In the second pass, the membership information of each pixel is mapped to the spatial domain, and the spatial function is computed from that. The MEPSO iteration proceeds with the new membership that is incorporated with the spatial function. The algorithm is stopped when the maximum number of time-steps t_{\max} is exceeded. After the convergence, de-fuzzification is applied to assign each pixel to a specific cluster for which the membership is maximal.

5. Experimental Results

Although we tested our algorithm over a large number of images with varying range of complexity, here we show the experimental results for three standard images only, due space limitation. We compared the proposed MEPSO based clustering algorithm with another recently developed fuzzy clustering algorithm known as FVGA (Fuzzy clustering with Variable length Genetic Algorithm) [22]. Parameter set-up for both the algorithms can be found in Table 1.

All the algorithms have been developed from scratch in Visual C++ on a Pentium IV, 1.2 GHz PC, with 512 KB cache and 2 GB of main memory with Windows Server 2003 environment.

For each image data set, a single run continues until the number of function evaluations (FEs) reaches 50,000. Twenty independent runs (with different seeds for the random number generator) have been taken for each algorithm. The results have been stated in terms of the mean best-of-run values and standard deviations

over these 20 runs in each case. Performance comparisons are made on three aspects of the solution: (a) quality of the solution as determined by the two cluster validity indices (described in (5) and (6)), (b) ability to find the optimal number of clusters, and (c) time required to find the solution.

Figures 1 to 3 show the three original images and their segmented counterparts obtained using the FVGA algorithm and the MEPSO based method. In Figures 1-3 the segmented portions of an image have been marked with the grey level intensity of the respective cluster centers.

Table 1. Algorithm parameters

| FVGA | | MEPSO | |
|-----------------------------|-------|--------------|------------------------|
| Parameter | Value | Parameter | Value |
| Pop_size | 20 | Pop_size | 40 |
| Crossover Probability P_c | 0.6 | Inertia | 0.794 |
| Mutation Probability P_m | 0.05 | Weight C_1 | 0.35 \rightarrow 2.4 |
| | | C_2 | 2.4 \rightarrow 0.35 |
| | | V_{max} | 255 |
| c_{max} | 10 | c_{max} | 10 |
| c_{min} | 2 | c_{min} | 2 |

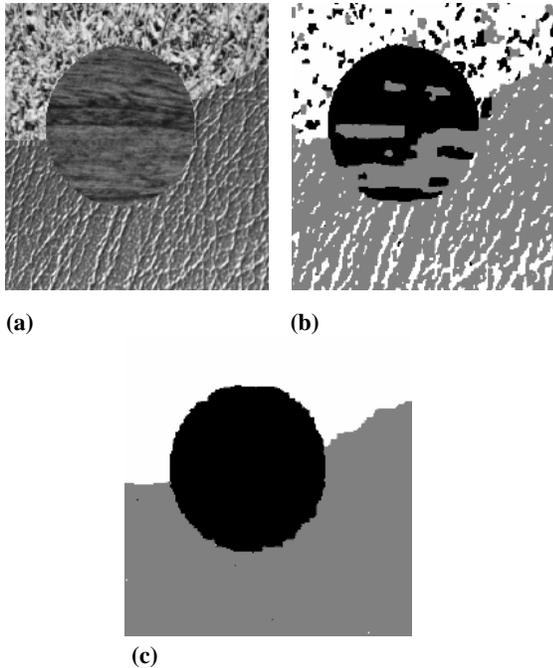


Figure 1. (a) The original Texture image. (b) Segmentation by FVGA ($c=3$) (c) Segmentation by MEPSO based method ($c=3$).

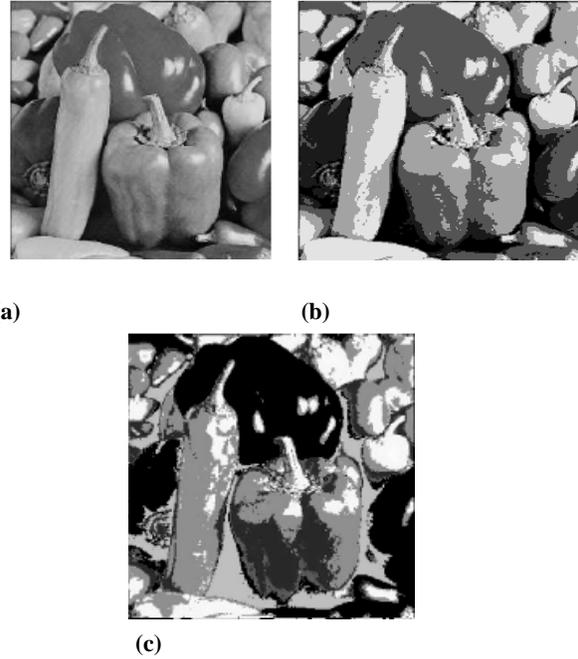


Figure 2. (a) The original Pepper image. (b) Segmentation by FVGA ($c=8$) (c) Segmentation by MEPSO based method ($c=7$).

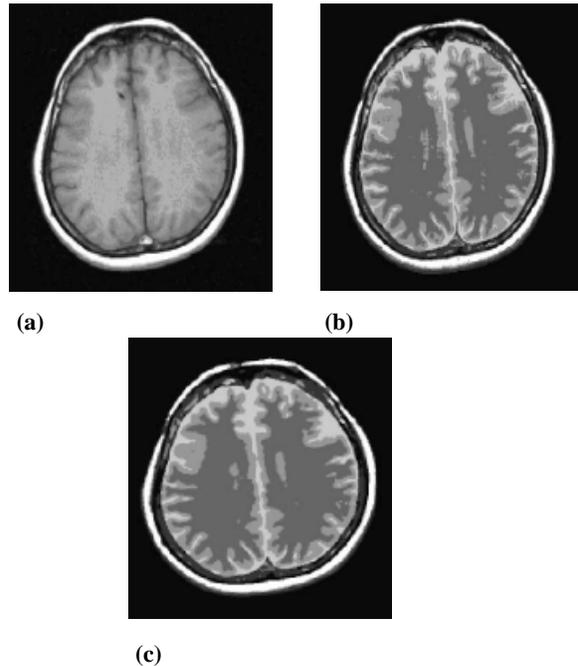


Figure 3. (a) The original MRI image. (b) Segmentation by FVGA ($c=5$) (c) Segmentation by MEPSO ($c=5$).

In Table 2, we report the mean value of two fuzzy validity measures calculated over the ‘best-of-run’ solutions in each case. MEPSO performed well in all

the cases (for both the indices). Table 3 reports the mean time taken by each algorithm to terminate on the image data. Finally, Table 4 contains the mean and standard deviations of the number of classes obtained by the two automatic clustering algorithms.

Table2. Automatic clustering results for three real life grayscale images (over 20 runs; each run continued up

| Image | Validity Index | Mean and Std Dev of the validity indices over the final clustering results of 20 independent runs | | |
|--------------------|-------------------|---|---------------------|---------------------|
| | | MEPSO | FVGA | FCM |
| Texture | Xie-Beni | 0.7283 (0.0001) | 0.7902 (0.0948) | 0.7937 (0.0013) |
| | Partition Entropy | 2.6631 (0.0018) | 2.1193 (0.0826) | 2.1085 (0.0043) |
| MRI Image of Brain | Xie-Beni | 0.2261 (0.0017) | 0.2919 (0.0583) | 0.3002 (0.0452) |
| | Partition Entropy | 0.1837 (0.0062) | 0.1922 (0.0096) | 0.1939 (0.0921) |
| Pepper Image | Xie-Beni | 0.05612 (0.0092) | 0.09673 (0.0043) | 0.09819 (0.0001) |
| | Partition Entropy | 0.8872 (0.0137) | 1.1391 (0.0292) | 1.1398 (0.0884) |

to 50,000 function evaluations)

Table 3. Comparison among the mean execution time taken by the different algorithms

| Image | Mean and Std Dev of the execution time (in seconds) taken by the different algorithms | |
|---------|---|-------------|
| | MESPO | FVGA |
| Texture | 32.05±0.076 | 47.25±0.162 |
| MRI | 24.15±0.016 | 34.65±0.029 |
| Pepper | 49.20±0.201 | 67.85±0.817 |

Table 4. Automatic clustering results for three grayscale images (over 20 runs; each runs continued for 50,000 function evaluations)

| Image | Optimal No. of Clusters | Mean and Std Dev of the number of classes estimated by the different algorithms | |
|---------|-------------------------|---|------------|
| | | FVGA | MEPSO |
| Texture | 3 | 3.75±0.211 | 3.05±0.132 |
| MRI | 5 | 5.05±0.428 | 5.25±0.212 |
| Pepper | 7 | 8.15±0.772 | 6.95±0.982 |

6. Conclusions

This paper has presented a new, PSO-based strategy for fuzzy clustering of images. An important feature of the proposed algorithm is that it is able to find the optimal number of clusters automatically (that is, the number of clusters does not have to be known in advance). Moreover, the proposed algorithm utilizes spatial information of each pixel, apart from the pixel intensity. The PSO algorithm has been modified with a multi-elitist strategy for improving its convergence behavior. Experimental results show that our approach outperforms the state-of-the-art FVGA strategy over a variety of image data sets. We could not include the full set of results obtained over a test suit of 30 images due to space constraints. Yet one may note that, not only does the proposed method find the optimal number of clusters, it also manages to find better clustering of the data points for major cluster validity indices used in the literature. Our algorithm may have a worse run-time than FCM, but clearly this is an unfair comparison as the FCM has to be fed with the correct number of clusters, whereas the proposed method finds it itself.

References

- [1] Jain, A.K., Murty M.N., and Flynn P.J. (1999): Data Clustering: A Review, ACM Computing Surveys, Vol 31, No. 3, 264-323.
- [2] Tou J.T, and Gonzalez R.C. (1974): Pattern Recognition Principles. London, Addison-Wesley.
- [3] Trivedi M. M, Bezdek J. C, Low-level segmentation of aerial images with fuzzy clustering , IEEE Trans.on Systems, Man and Cybernetics, Volume 16 , Issue 4 July, 1986.
- [4] G. Ball and D. Hall, A Clustering Technique for Summarizing Multivariate Data, Behavioral Science 12, 153-155 (1967).
- [5] Wallace, C.S. and Boulton, D.M. (1968), 'An Information Measure for Classification', Computer Journal, Vol. 11, No. 2, 1968, pp. 185-194.
- [6] Bezdek, J.C. 1981. Pattern recognition with fuzzy objective function algorithms. New York: Plenum.
- [7] L.O. Hall, I.B. Özyurt, J.C. Bezdek, Clustering with a genetically optimized approach, IEEE Trans. Evo Comp. 3 (2)(1999) 103–112.
- [8] Gath, I. and Geva. A.: Unsupervised optimal fuzzy clustering. IEEE Transactions on PAMI, 11(1989) 773-781.

- [9] Bensaid, A.M., Hall, L. O., Bezdek, J.C. and Clarke, L. P., Partially supervised clustering for image segmentation. *Pattern Recognition*, 29:859-871, 1996.
- [10] Clark MC, Hall LO, Goldgof DB, Clarke LP, Velthuizen RP, Silbiger MS. MRI segmentation using fuzzy clustering techniques. *IEEE Eng Med Biol (1994)* 13:730-42.
- [11] Ahmed, M. N, Yamany, S. M, Mohamed, N, Farag, A. A, Moriarty, T. A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. *IEEE Trans Med Imaging* 2002; 21:193-9.
- [12] Wang X, Wang Y, Wang L. Improving fuzzy c-means clustering based on feature-weight learning. *Pattern Recognit Lett* 2004; 25: 1123-32.
- [13] Halkidi, M., Batistakis, Y., Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems (JIIS)*, 17(2-3), 107-145.
- [14] Theodoridis S. and Koutroubas K. (1999) *Pattern recognition*, Academic Press.
- [15] Rosenberger, C., Chehdi, K., "Unsupervised clustering method with optimal estimation of the number of clusters : Application to image segmentation", in Proc. IEEE International Conference on Pattern Recognition (ICPR), pp. 1656-1659, vol. 1, Barcelona, September 2000,
- [16] Lee C-Y and Antonsson E K, 2000. Self-adapting vertices for mask layout synthesis *Modeling and Simulation of Microsystems Conference (San Diego, March 27-29)* eds. M Laudon and B Romanowicz pp 83-6
- [17] Sarkar, M., Yegnanarayana, B., Khemani, D.: "A clustering algorithm using an evolutionary programming-based approach". *Pattern Recognition Letters*, 18, (1997) 975-986.
- [18] Bandyopadhyay, S., and Maulik, U., Genetic clustering for automatic evolution of clusters and application to image classification, *Pattern Recognition*, 35, (2002) 1197-1208.
- [19] Xiao, X., Dow, E., Eberhart, R., Ben Miled, Z. and Oppelt, R.J., 2003. Gene Clustering using Self-Organizing Maps and Particle Swarm Optimization. In: *Proceeding of Second IEEE International Workshop on High Performance Computational Biology*, Nice, France.
- [20] Omran, M., Engelbrecht, A, P., Salman, A., *Differential Evolution Methods for Unsupervised Image Classification*, Proc. Seventh Congress on Evolutionary Computation (CEC-2005). IEEE Press.
- [21] Ahmed MN, Yamany SM, Mohamed N, Farag AA, Moriarty T. A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. *IEEE Trans Med Imaging* 2002; 21:193-199.
- [22] Pakhira, M. K., Bandyopadhyay, S., and Maulik, U., A Study of Some Fuzzy Cluster Validity Indices, *Genetic clustering And Application to Pixel Classification, Fuzzy Sets and Systems* 155 (2005) 191-214
- [23] Kennedy, J, Eberhart R. Particle swarm optimization, In *Proceedings of IEEE International Conference on Neural Networks*, (1995) 1942-1948.
- [24] Eberhart, R. C., Shi, Y.: Particle swarm optimization: Developments, applications and resources, In *Proceedings of IEEE International Conference on Evolutionary Computation*, vol. 1 (2001), 81-86.
- [25] Clerc, M., Kennedy, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space, In *IEEE Transactions on Evolutionary Computation* (2002) 6(1): 58-73.
- [26] Deb, K, Pratap, A., Agarwal, S, and Meyarivan, T, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, Vol.6, No.2, 2002.