

Use of aggregation pheromone density for image segmentation

Susmita Ghosh^a, Megha Kothari^a, Anindya Halder^b, Ashish Ghosh^{c,*}

^a Department of Computer Science and Engineering, Jadavpur University, Kolkata, West Bengal 700 108, India

^b Center for Soft Computing Research, Indian Statistical Institute, Kolkata, West Bengal 700 108, India

^c Machine Intelligence Unit and Center for Soft Computing Research, Indian Statistical Institute, Kolkata, West Bengal 700 108, India

ARTICLE INFO

Article history:

Received 12 March 2008

Received in revised form 24 December 2008

Available online 20 March 2009

Communicated by L. Heutte

Keywords:

Aggregation pheromone

Ant colony optimization

Clustering

Image segmentation

ABSTRACT

Ants, bees and other social insects deposit pheromone (a type of chemical) in order to communicate between the members of their community. Pheromone that causes clumping or clustering behavior in a species and brings individuals into a closer proximity, is called aggregation pheromone. This paper presents a novel method for image segmentation considering the aggregation behavior of ants. Image segmentation is viewed as a clustering problem which aims to partition a given set of pixels into a number of homogenous clusters/segments. At each location of a data point, representing a pixel, an ant is placed; and the ants are allowed to move in the search space to find out positions with higher pheromone density. The movement of an ant is governed by the amount of pheromone deposited at different positions of the search space. More the deposited pheromone, more is the aggregation of ants. This leads to the formation of homogenous groups of data. The proposed algorithm is evaluated on a number of different types of images using various cluster validity measures. Results are compared with those obtained using *k-means* and *mean shift* clustering algorithms and are found to be superior.

1. Introduction

Image segmentation plays a vital role in image analysis and computer vision (Gonzalez and Woods, 2003). The objective of image segmentation is to extract meaningful non-overlapping regions from an image. A number of image segmentation techniques exists in the literature (Zhang, 2006), which can be grouped into several categories such as edge based (Gonzalez and Woods, 2003; Pal and Pal, 1993), region based (Haralick and Shapiro, 1985; Freixenet et al., 2002), thresholding based (Pal and Pal, 1993; Sezgin and Sankur, 2004) and clustering based (Gonzalez and Woods, 2003; Pal and Pal, 1993; Kettaf et al., 1996).

Image segmentation when viewed as a clustering problem aims to partition the image into clusters (segments) such that the pixels within a cluster are as homogeneous as possible whereas the pixels from different clusters are as heterogeneous as possible with respect to some similarity measure. Features that are commonly used for image segmentation by clustering not only include the gray values, but also the textural features defined on local neighborhood (Kettaf et al., 1996). Several clustering techniques exist (Gonzalez and Woods, 2003; Pal and Pal, 1993; Haralick and Shapiro, 1985; Kettaf et al., 1996) to group pixels into segments.

Swarm intelligence (Englebrect, 2002), which takes inspiration from the social behavior of insects and other animals, is a relatively new computational approach to solve problems. In particular, ants have inspired a number of techniques collectively known as ant colony optimization (Dorigo and Stützle, 2005). Ant colony based algorithms are bio-mimetic evolutionary algorithms. These algorithms have parallel positive feedback mechanisms having advantages of parallelism, robustness and easy combination with other methods. The discreteness and parallelism of the ant colony based algorithms make them potential candidates for digital image analysis and hence can be applied to image processing.

In this article image segmentation is considered as a clustering problem and is performed using aggregation pheromone density based clustering (APC) algorithm which is inspired by the aggregation behavior found in ants and other social insects. Experimental results on a wide variety of images justify the potentiality of the proposed APC algorithm both in terms of segmentation (clustering) quality as well as execution time compared to other algorithms.

The paper is organized as follows: in Section 2 we describe the motivation for the proposed and related works. Section 3 gives the detailed description of the proposed method together with theoretical details of the validity indices used. In Section 4 we evaluate the performance of the proposed method using a wide variety of images and compared the results with two other

* Corresponding author. Tel.: +91 33 2575 3110; fax: +91 33 2578 3357.
E-mail address: ash@isical.ac.in (A. Ghosh).

clustering based segmentation techniques. Finally, conclusions are drawn in Section 5.

2. Motivation and related work

The social insects' behavior such as finding the best food source, building of optimal nest structure, brooding, protecting the larva, guarding, etc., show intelligent behavior on the swarm level (Englebrecht, 2002). A swarm behavior is not determined just by the behavior of individuals, but the interactions among individuals play a vital role in shaping the swarm behavior (Englebrecht, 2002). Computational modeling of swarms' behavior is found to be useful in various application domains like function optimization (Tsutsui et al., 2004; Tsutsui and Ghosh, 2004), finding optimal routes (Dorigo and Gambardella, 1997), scheduling (Dorigo and Stützle, 2005), image and data analysis (Wang et al., 2005). Different applications originated from the study of different swarms. Among them the popular ones are ant colonies and bird flocks (Englebrecht, 2002). Ant colony optimization (ACO) (Dorigo et al., 1996) and aggregation pheromone systems (APS) (Tsutsui et al., 2004; Tsutsui and Ghosh, 2004) are computational algorithms modeled on the behavior of the ant colonies. ACO (Dorigo et al., 1996) algorithms are designed to emulate ants' behavior of laying pheromone on the ground while moving from one position to another for solving optimization problems. Pheromone is a type of chemical emitted by an organism to communicate between members of the same species. Pheromone which is responsible for clumping or clustering behavior in a species and brings individuals into closer proximity is known as aggregation pheromone. Thus, aggregation pheromone causes individuals to aggregate around good positions which in turn produces more pheromone to attract individuals of the same species. In APS (Tsutsui et al., 2004; Tsutsui and Ghosh, 2004), a variant of ACO, this behavior of ants is used to solve real parameter optimization problems (Tsutsui et al., 2004, 2005). A model for continuous parameter optimization problem was also proposed in (Socha and Dorigo, 2008, de Franca et al., 2008) as an extension of ant colony optimization (ACO).

Numerous abilities of ants have inspired researches for designing various clustering techniques (Deneubourg et al., 1991; Lumer and Faieta, 1994). Several species of ants cluster their corpses into cemeteries in an effort to clean up their nests. Experimental works illustrate that ants group corpses, which are initially randomly distributed in space, into clusters within a few hours. It seems that some feedback mechanism (using local density or similarity of data items) determines the probability that an ant will pick up or drop a corpse. Such behavior is used as a model to design algorithms for clustering data (Deneubourg et al., 1991; Lumer and Faieta, 1994; Monmarché et al., 1999; Handl et al., 2003; Liu et al., 2004; Vizine et al., 2005).

Lumer and Faieta (1994) generalized this method and proposed an algorithm known as *Ant Colony Clustering*, which was applied for exploratory data analysis. In this work, data movements were implemented through the ants' movements requiring additional storage and computational burden as the ants make idle movement while carrying no data object. Moreover, in this model, ants carrying isolated corpses (data items) make everlasting move since they never find a proper location to drop them. This consumes a large amount of computing time.

To speed up convergence and to reduce parameter settings, Monmarché et al. (1999) proposed an interesting hybridization of this algorithm with *k*-means (MacQueen, 1967) and named the hybrid technique *AntClass*. They compared it with the traditional *k*-means (MacQueen, 1967) and *ISODATA* (Ball and Hall, 1965) clustering algorithms on various data sets using classification error as evaluation criterion. Although *AntClass* algorithm

gives satisfactory results, computational time is high. In *AntClass* algorithm, objects (data points) are scattered randomly on the grid board. As a result, the objects within a high density region may be dispersed in different cells and it may need longer time for the ants to collect similar objects into one cell.

To improve the performance, stability and convergence of the *Ant Colony Clustering* algorithm of Lumer and Faieta (1994), Vizine et al. (2005) proposed *An Adaptive Ant Clustering Algorithm* with (i) a progressive vision field that allows ants to see over a wider area, (ii) pheromone heuristics to promote reinforcement for dropping of objects at more dense regions of the grid, and (iii) cooling schedule of the parameters that controls the probability of ants picking up objects from the grid. They evaluated their algorithm on a number of well known benchmark data sets as well as on a real world bioinformatics data set. The modified model is found to have significant improvement over the original algorithm.

In another attempt to speed up the process, Liu et al. (2004) proposed *DBAntCluster* algorithm, by incorporating information from the input data distribution. In this method first the high density clusters are detected by *DBSCAN* (Ester et al., 1996) and the clusters so formed are scattered on the grid board (treating the clusters as special objects with the single objects). Afterwards, *Ant Colony Clustering* algorithm is used to cluster the data objects on the grid board.

To enable an unbiased interpretation of the solutions obtained using ant based clustering algorithms, Handl et al. (2003) and Handl et al. (2006) proposed a method to determine suitable parameter settings across different test sets. They also suggested a technique to convert the spatial embedding generated by the ant algorithms, which implicitly contains clusters, to an explicit partitioning of the data set. To evaluate the results obtained by *k*-means (MacQueen, 1967), agglomerative average linkage clustering (Vorhees, 1985) and one dimensional self organizing map (Kohonen, 1997) on synthetic and real data sets, they used different analytical measures and showed that the ant based algorithms perform well.

Ramos and Merelo (2002) and Ramos et al. (2002) developed an ant clustering system called *ACLUSTER*, for textual document clustering and retrieval of digital images. Unlike the *Ant Colony Clustering* algorithm as developed by Lumer and Faieta (1994), here ants do not move randomly, rather they move according to some transition probabilities depending on the spatial distribution of the pheromone across the environment. This eliminates the need of short term memory (required earlier in Lumer and Faieta model (Lumer and Faieta (1994))) for storing past movements of ants. They used the combination of following two independent response threshold functions, associated with different environmental conditions: (i) number of objects in an area and (ii) their similarity.

A comprehensive review on ant and swarm based clustering is done by Handl and Meyer (2007). They categorized ant based clustering into the following two groups: (i) methods that directly mimic the clustering behavior observed in real ant colonies; and (ii) methods in which the clustering task is reformulated as an optimization task and general purpose ant based optimization heuristics are utilized to find good or near optimal clustering.

In a recent work Ouadfel and Batouche have proposed *AntClust* algorithm (Ouadfel and Batouche, 2007), that uses the self-organizing and autonomous brood sorting behavior of real ants for image segmentation. In this method the image pixels are scattered within the cells of the array and can be moved from one cell to another by the movement of the ant to form clusters. By this process, ants dynamically cluster pixels into distinct groups. Another object segmentation method using ant colony optimization (ACO) and fuzzy entropy was proposed by Tao et al. (2007). Lai et al. proposed an ant colony system (ACS)

based image texture segmentation (Chu et al., 2006). They used ACS to find the trade-off between texture segments and fragments. A hybrid ant colony system and Markov random field (ACS-MRF) was proposed by Ouadfel and Batouche (2003) where a colony of artificial ants searched for a globally optimum solution defined as a correct labeling of image pixels with respect to the contextual constrains.

Most of the ant based clustering algorithms, developed till now, are inspired by the ants' property of piling up the corpses to clean the nest. Besides nest cleaning, many functions of aggregation behavior have been observed in ants and ant like agents (Bell, 1984; Ono et al., 1995; Sukama and Fukami, 1993). These include foraging-site marking and mating, finding shelter and defense. For example, after finding safe shelter, cockroaches produce a specific pheromone with their excrement, which attracts other members of their species (Sukama and Fukami, 1993). Based on the similar property i.e., ants need to find comfortable and secure environment to sleep, Chen et al. proposed *Ant Sleeping Model* (Chen et al., 2004) which makes ants to group with those that have similar physiques. They defined a fitness function to measure the ants' similarity with their neighbors. They stated that when an ant's fitness is low, it has a higher probability to wake up and stay in active state. Thus an ant will leave its original position to search for a more secure and comfortable position to sleep. Since each individual ant uses local information to decide whether to be in active state or sleeping state, the whole ant group dynamically self organizes into distinctive, independent subgroups. Using similar concept Tsutsui et al. (2004) and Tsutsui and Ghosh (2004) used *aggregation pheromone systems* for continuous function optimization where aggregation pheromone density is defined by a density function in the search space.

As mentioned above, many functions of aggregation behavior have been observed in ants and ant like agents. Inspired by this behavior found in ants and other similar agents, in earlier works preliminary attempts are made for solving classification (Halder et al., 2008), clustering (Kothari et al., 2006; Ghosh et al., 2008), image segmentation (Ghosh et al., 2006) and change detection (Kothari et al., 2007) problems with encouraging results.

3. Aggregation pheromone based image segmentation

Clustering is a popular technique for image segmentation (Zhang, 2006). As mentioned in the introduction, aggregation pheromone brings individuals into closer proximity. This group formation nature of aggregation pheromone is being used as the basic idea of the proposed algorithm. Here each ant represents a pixel of the input image. The ants move virtually with an aim to create homogenous groups of data. The amount of virtual movement of an ant towards a point is governed by the intensity of aggregation pheromone deposited by all other ants at that point. This gradual movement of ants in due course of time will result in formation of groups or clusters of homogeneous pixels (segments). The proposed technique has two parts. In the first part, from the pixels of the input image clusters of homogeneous pixels (segments) are formed based on ants' property of depositing aggregation pheromone. The number of segments (clusters) thus formed might be more than the desired number. So, to obtain the desired number of clusters, in the second part, agglomerative *average linkage* clustering algorithm is applied on these already formed clusters. Clusters so formed represent different homogeneous segments of an image. Finally the clustering results obtained are evaluated by different validity indices and the best results corresponding to each validity index are chosen as the final result.

3.1. Formation of clusters/segments

While performing image segmentation for a given image we group similar pixels together to form a set of coherent image regions. Similarity of pixels can be measured based on different features like intensity, color, texture, local entropy, etc. Individual features or combination of them can be used to represent an image pixel. Thus for each image pixel we associate a feature vector \mathbf{x} . Clustering is then performed on this set of feature vectors so as to group them. Finally, clustering result is mapped back to the original spatial domain to obtain segmented image.

3.2. Aggregation pheromone density based clustering/segmentation

Consider a data set of n patterns $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n$ and a population of n -ants $a_1, a_2, a_3, \dots, a_n$ where an ant a_i represents the data pattern \mathbf{x}_i . In this article we assume each pixel of the input image as a data point, and hence as an individual ant. Each individual ant emits pheromone around its neighborhood. The intensity of pheromone emitted by an individual ant a_i (located at \mathbf{x}_i) decreases with its distance from \mathbf{x}_i . Thus the pheromone intensity at a point closer to \mathbf{x}_i is more than those at other points that are farther from it. To achieve this, the pheromone intensity emitted by ant a_i is modeled by a Gaussian distribution. The pheromone intensity deposited at \mathbf{x} by an ant a_i (located at \mathbf{x}_i) is thus computed as

$$\Delta\tau(a_i, \mathbf{x}) = \exp \frac{d(\mathbf{x}_i, \mathbf{x})^2}{2\delta^2} \quad (1)$$

where δ denotes the spread of Gaussian function and $d(\mathbf{x}_i, \mathbf{x})$ is the Euclidean distance between \mathbf{x}_i and \mathbf{x} . The total aggregation pheromone density at \mathbf{x} deposited by the entire population of n ants is computed by the following equation

$$\Delta\tau(\mathbf{x}) = \sum_{i=1}^n \exp \frac{d(\mathbf{x}_i, \mathbf{x})^2}{2\delta^2} \quad (2)$$

Now, an ant a_i which was initially at location \mathbf{x}_i moves to the new location \mathbf{x}'_i (computed using Eq. (3)) if the total aggregation pheromone density at \mathbf{x}'_i is greater than that at \mathbf{x}_i . The movement of an ant is governed by the amount of pheromone deposited at different points in the search space; and is defined as

$$\mathbf{x}'_i = \mathbf{x}_i + \eta \cdot \frac{Next(a_i)}{n}, \quad (3)$$

where

$$Next(a_i) = \sum_{j=1}^n (\mathbf{x}_j - \mathbf{x}_i) \cdot \exp \frac{d(\mathbf{x}_j, \mathbf{x}_i)^2}{2\delta^2} \quad (4)$$

with η (a proportionality constant) as the step size. This process of finding a new location continues until an ant finds a location where the total aggregation pheromone density is more than its neighboring points. Once the ant a_i finds out such a point \mathbf{x}'_i , then that point is assumed to be a new potential cluster center, say \mathbf{Z}_j ($j = 1, 2, \dots, C$ C being number of clusters); and the data point with which the ant was associated earlier (i.e., \mathbf{x}_i) is assigned to the cluster so formed with center \mathbf{Z}_j . Also the data points that are within a distance of $\delta/2$ from \mathbf{Z}_j are assigned to the newly formed cluster. On the other hand, if the distance between \mathbf{x}'_i and the existing cluster center \mathbf{Z}_j is less than 2δ and the ratio of their densities is greater than *threshold_density* (a predefined parameter), then the data point \mathbf{x}_i is allocated to the cluster having cluster center \mathbf{Z}_j . Higher value of density ratio indicates that the two points are of nearly similar density and hence should belong to the same cluster. The proposed aggregation pheromone based clustering (APC) algorithm is given below.

Algorithm

```

Initialize  $\delta$ , threshold_density,  $\eta$ 
 $C = 0$ 
for  $i = 1 : n$  do
  if (the data pattern  $\mathbf{x}_i$  is not already assigned any cluster)
    Compute  $\Delta\tau(\mathbf{x}_i)$  using Eq. (2).
    label 1:
      Compute new location  $\mathbf{x}'_i$  using Eq. 3.
      Compute  $\Delta\tau(\mathbf{x}'_i)$ .
      // End of label
    if ( $\Delta\tau(\mathbf{x}'_i) > \Delta\tau(\mathbf{x}_i)$ )
      Update the location of ant  $a_i$  (at  $\mathbf{x}_i$ ) to  $\mathbf{x}'_i$ 
      and goto label 1.
    end of if
    if ( $C == 0$ ) //If no cluster exists
      Consider  $\mathbf{x}'_i$  as cluster center  $\mathbf{Z}_1$  and increase C by one.
      Assign all the data points within a distance of  $\delta/2$ 
      from  $\mathbf{x}'_i$  to the newly formed cluster with center  $\mathbf{Z}_1$ .
    else
      for  $j = 1 : C$ 
        if ( $\min(\Delta\tau(\mathbf{x}'_i), \Delta\tau(\mathbf{Z}_j)) / \max(\Delta\tau(\mathbf{x}'_i), \Delta\tau(\mathbf{Z}_j)) >$ 
        threshold_density and  $d(\mathbf{x}'_i, \mathbf{Z}_j) < 2\delta$ )
          Assign  $\mathbf{x}'_i$  to  $\mathbf{Z}_j$ . //  $\mathbf{Z}_j$  already exists.
        else
          Assign  $\mathbf{x}'_i$  as a new cluster center say,  $\mathbf{Z}_{C+1}$  and
          increase C by one.
          Assign all the data points that are within a dis-
          tance of  $\delta/2$ 
          from  $\mathbf{x}'_i$  to the newly formed cluster with center
           $\mathbf{Z}_{C+1}$ .
        end of if
      end of for
    end of if
  end of for
end of if (if the data pattern  $\mathbf{x}_i \dots$ )
end of for

```

3.3. Merging of clusters/segments

In the proposed method (described above), we have applied the APC algorithm on the whole data set in only one pass (iteration). The number of clusters produced (depending on the parameter values) may be more than the desired number of clusters. To obtain the desired number of clusters, we applied the average linkage agglomerative hierarchical clustering algorithm (average linkage, in short) (Vorhees, 1985) for merging them. In this sense both the steps are applied in combination (one after another in only one iteration). In other words, the algorithm stops after one iteration only, and we get the desired number of segments of an image.

3.4. Objective evaluation of segmentation results

As discussed earlier, in the present article image segmentation is viewed as a clustering problem and hence segmentation (clustering) results are quantified using the following two popular cluster validity measures.

- **Davies Bouldin Index:** This index is a function of the ratio of the sum of within-cluster scatters to between-cluster separation (Theodoridis and Koutroumbas, 2003). The average scatter of order q within the i th cluster (denoted by $S_{i,q}$) and the distance between i th and j th clusters (denoted by $d_{ij,q}$) are computed as

$$S_{i,q} = \left(\frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{Z}_i\|^q \right)^{\frac{1}{q}}, \quad (5)$$

$$d_{ij,q} = \|\mathbf{Z}_i - \mathbf{Z}_j\|^q, \quad (6)$$

where \mathbf{x} is the data point belonging to cluster C_i , \mathbf{Z}_i is the centroid of cluster C_i , $q \geq 1$ and $d_{ij,q}$ is the Minkowski distance of order q . Subsequently, we compute the index for the i th cluster (denoted by $R_{i,q}$) as

$$R_{i,q} = \max_{j,i \neq j} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,q}} \right\}. \quad (7)$$

The *Davies–Bouldin (DB)* index for C clusters is defined as

$$DB = \frac{1}{C} \sum_{i=1}^C R_{i,q}. \quad (8)$$

In this article we have chosen the value of $q = 2$. *The smaller the DB value, better is the clustering.*

- **S_Dbw:** *S_Dbw* index with C number of clusters is based on the cluster compactness in terms of intra-cluster variance and inter-cluster density (Halkidi and Vazirgiannis, 2001). It is defined as

$$S_Dbw(C) = Scat(C) + Den(C), \quad (9)$$

where $Scat(C)$ represents the intra-cluster variance and is defined as

$$Scat(C) = \frac{1}{C} \sum_{i=1}^C \|\sigma(\mathbf{Z}_i)\| / \|\sigma(X)\|; \quad (10)$$

the term $\sigma(X)$ is the variance of the data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ and $\sigma(\mathbf{Z}_i)$ is the variance of cluster C_i . Inter-cluster density, $Den(C)$, is defined as

$$Den(C) = \frac{1}{C(C-1)} \sum_{i=1}^C \left(\sum_{i=1, i \neq j}^C \frac{den(\mathbf{u}_{ij})}{\max\{den(\mathbf{Z}_i), den(\mathbf{Z}_j)\}} \right) \quad (11)$$

where \mathbf{Z}_i and \mathbf{Z}_j are centers of clusters C_i and C_j , respectively, and \mathbf{u}_{ij} is the mid-point of the line segment joining \mathbf{Z}_i and \mathbf{Z}_j . The term $den(\mathbf{u})$ is defined as

$$den(\mathbf{u}) = \sum_{\mathbf{x} \in C_i \cup C_j} f(\mathbf{x}, \mathbf{u}). \quad (12)$$

The function $f(\mathbf{x}, \mathbf{u})$ is defined as

$$f(\mathbf{x}, \mathbf{u}) = \begin{cases} 0, & \text{if } d(\mathbf{x}, \mathbf{u}) > stdev; \\ 1, & \text{otherwise;} \end{cases} \quad (13)$$

where $stdev$ is the average standard deviation of C clusters and is defined as

$$stdev = \frac{1}{C} \sqrt{\sum_{i=1}^C \|\sigma(\mathbf{Z}_i)\|} \quad (14)$$

and $d(\mathbf{x}, \mathbf{u})$ is the Euclidean distance between \mathbf{x} and \mathbf{u} .

Lower the value of S_Dbw, better is the clustering.

4. Experimental evaluation

The experimental studies presented here provide an evidence of the effectiveness of the proposed APC algorithm for image segmentation. In the subsequent sections we report on the details of the description of the experimental setup and then analyze the results.

4.1. Description of experiments

Experiments were carried out on 150 different kinds of images. Validity measures for a few of them are summarized

Table 1
Performance (*DB, S.Dbw*) measures and execution time of the proposed APC, *k*-means and mean shift methods.

Image	Method		δ	<i>DB</i> value	<i>S.Dbw</i> value	Time (s)
Tank (512 × 512 <i>C</i> = 2)	APC	Selection criterion	δ			
		Best <i>DB</i> and <i>S.Dbw</i>	0.17	0.258957	0.102615	1.062
	<i>k</i> -means			0.467821	0.4012517	0.727
	Mean shift	(<i>bandwidth</i> = 0.3; <i>stop_threshold</i> = 3.5)		0.378253	0.301745	5.746
Stream-Bridge (512×512 <i>C</i> = 4)	APC	Selection criterion	δ			
		Best <i>DB</i> and <i>S.Dbw</i>	0.09	0.484889	0.261569	1.156
	<i>k</i> -means			0.59375	0.385273	0.859
	Mean shift	(<i>bandwidth</i> = 0.3; <i>stop_threshold</i> = 3.5)		0.79088	0.582161	5.702
House (256 × 256 <i>C</i> = 6)	APC	Selection criterion	δ			
		Best <i>DB</i> and <i>S.Dbw</i>	0.13	0.44245	0.154929	0.282
	<i>k</i> -means			0.74475	0.602215	0.227
	Mean shift	(<i>bandwidth</i> = 0.2; <i>stop_threshold</i> = 2.5)		0.47049	0.21527	1.162
Einstien (375 × 500 <i>C</i> = 4)	APC	Selection criterion	δ			
		Best <i>DB</i> and <i>S.Dbw</i>	0.48	0.404219	0.163483	0.844
	<i>k</i> -means			0.726847	0.371559	0.706
	Mean shift	(<i>bandwidth</i> = 0.3; <i>stop_threshold</i> = 2.0)		0.483725	0.152308	2.0475
Arial (256 × 256 <i>C</i> = 4)	APC	Selection criterion	δ			
		Best <i>DB</i> and <i>S.Dbw</i>	0.14	0.441098	0.185137	0.281
	<i>k</i> -means			0.717741	0.447064	0.016
	Mean shift	(<i>bandwidth</i> = 0.25; <i>stop_threshold</i> = 2.5)		0.475027	0.205142	1.703
Marble (256×256 <i>C</i> = 4)	APC	Selection criterion	δ			
		Best <i>DB</i>	0.06	0.410238	0.225205	0.328
		Best <i>S.Dbw</i>	0.1	0.476579	0.196084	0.297
	<i>k</i> -means			0.523499	0.285837	0.185
	Mean shift	(<i>bandwidth</i> = 0.18; <i>stop_threshold</i> = 2.0)		0.493574	0.190382	1.504
Lena (512 × 512 <i>C</i> = 3)	APC	Selection criterion	δ			
		Best <i>DB</i> and <i>S.Dbw</i>	0.26	0.416895	0.339411	1.094
	<i>k</i> -means			0.467821	0.4012517	0.727
	Mean shift	(<i>bandwidth</i> = 0.3; <i>stop_threshold</i> = 2.5)		0.378253	0.301745	5.746
Tulips (256 × 256 <i>C</i> = 3)	APC	Selection criterion	δ			
		Best <i>DB</i>	0.07	0.481992	0.275278	0.312
		Best <i>S.Dbw</i>	0.12	0.497296	0.260385	0.297
	<i>k</i> -means			0.460269	0.210957	0.141
	Mean shift	(<i>bandwidth</i> = 0.3; <i>stop_threshold</i> = 3.2)		0.624495	0.531792	1.026
Iga (256 × 256 <i>C</i> = 3)	APC	Selection criterion	δ			
		Best <i>DB</i>	0.22	0.475058	0.551056	0.281
		Best <i>S.Dbw</i>	0.2	0.508503	0.483094	0.297
	<i>k</i> -means			0.446207	0.419203	0.224
	Mean shift	(<i>bandwidth</i> = 0.25; <i>stop_threshold</i> = 2.0)		0.683772	0.605733	1.104
Pepper (256 × 256 <i>C</i> = 4)	APC	Selection criterion	δ			
		Best <i>DB</i>	0.26	0.47659	0.408604	0.281
		Best <i>S.Dbw</i>	0.15	0.507058	0.321668	0.297
	<i>k</i> -means			0.627044	0.440179	0.224
	Mean shift	(<i>bandwidth</i> = 0.43; <i>stop_threshold</i> = 4.0)		0.450253	0.388291	1.471
Baboon (256 × 256 <i>C</i> = 4)	APC	Selection criterion	δ			
		Best <i>DB</i>	0.09	0.454274	0.310389	0.282
		Best <i>S.Dbw</i>	0.3	0.473884	0.293841	0.266
	<i>k</i> -means			0.32879	0.240582	0.146
	Mean shift	(<i>bandwidth</i> = 0.25; <i>stop_threshold</i> = 2.0)		0.41826	0.273841	1.502

in Table 1. The corresponding images are displayed in Figs. 4–14 for typical illustration. Number of segments for each image is pre-assumed. For example, in case of Einstien image the preassumed number of segments is 4 (indicated as *C* = 4). Also the size of the Einstien image is represented by 375 × 500, which indicates that the images has 375 rows and 500 columns.

4.1.1. Feature generation

The investigation was done by considering the gray value of a pixel as one feature and local entropy (Zhang, 2006) over an window of size 9 × 9 around that pixel as another feature (to take into account the neighborhood effects of the pixel under consideration). Please note that in general we can take any number of features.

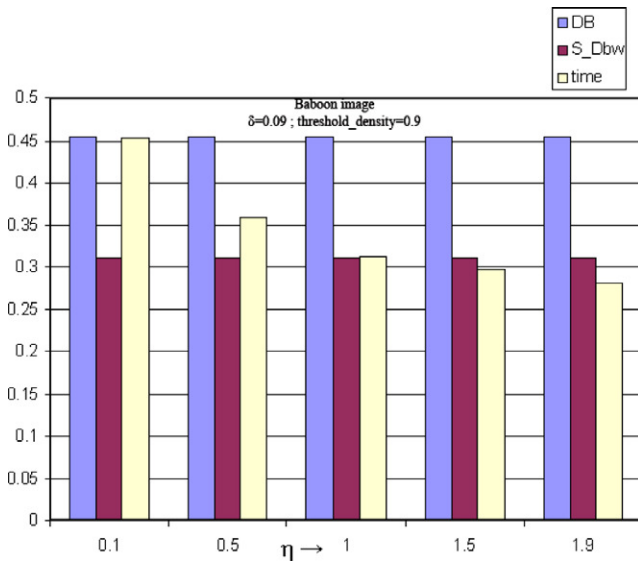


Fig. 1. Variation of DB and S.DbW and execution time with respect to η for Baboon image.

4.1.2. Parameters selection

It is evident from the algorithm described in Section 3.2 that the proposed method has three parameters namely η , *threshold_density* and δ .

Here η is the step size. The smaller the step size, more will be the time taken to explore the search space. The performance of the algorithm in terms of validity measures is found to remain almost constant for a wide range [0.1–1.9] of η . We have reported results of the experiments with step size $\eta = 1$, as the performance is found to be constant over a wide range around it. For typical illustration, the variations of the execution time and the DB & S.DbW measures with respect to η (keeping $\delta = 0.09$ and *threshold_density* = 0.9) is shown in Fig. 1 for Baboon image.

If the ratio of pheromone density of a data point and an already formed cluster center (within distance 2δ) is higher than the *threshold_density* then the said point is assigned to this cluster. This assumes that two closer points having nearly similar pheromone density should belong to the same cluster. High *threshold_density* value indicates that pheromone densities of two points (within

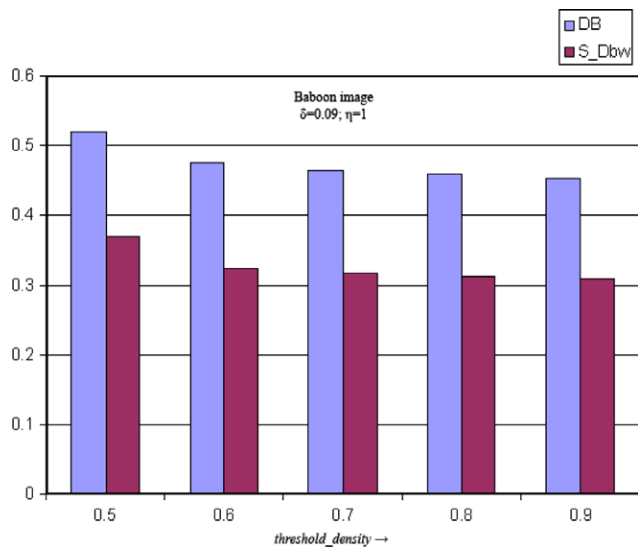


Fig. 2. Variation of DB and S.DbW with respect to *threshold_density* for Baboon image.

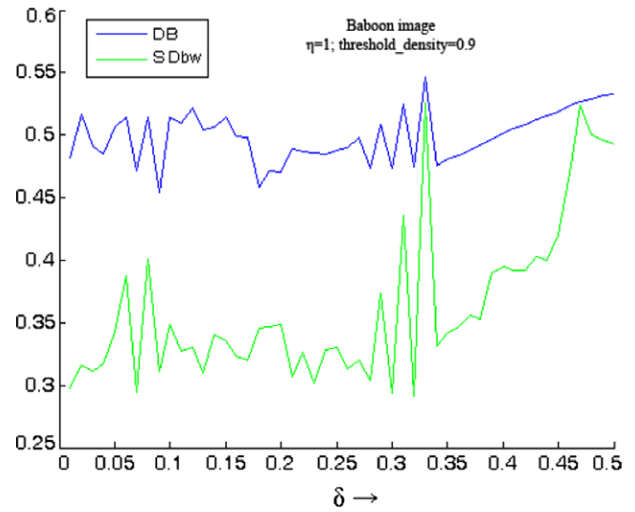


Fig. 3. Variation of DB and S.DbW with respect to δ for Baboon image.

2δ) should be very close to come into the same cluster. Less *threshold_density* value indicates that the two closer points may reside in the same cluster even if their pheromone densities are not very similar. If the *threshold_density* value is high, it is likely to form large number of clusters in the initial phase (before merging the clusters); and if it is less, the number of clusters formed (in the initial phase) may be small. We have executed the algorithm taking different values of *threshold_density* over the range [0.5–0.9] and on an average 0.9 was found to be a suitable value. For typical illustration, the variations of DB and S.DbW measures with respect to *threshold_density* (keeping $\eta = 1$ and $\delta = 0.09$) is shown in Fig. 2 for Baboon image.

The algorithm is executed for different δ (spread of the Gaussian) values in the range (0–0.5); and used the experimentally determined δ to produce the best results in terms of the validity measures (S.DbW, DB). For typical illustration the variations of S.DbW and DB indices with respect to δ (keeping $\eta = 1$ and *threshold_density* = 0.9) are shown in Fig. 3 for the Baboon image. From the figure it is seen that minimum DB value occurred for $\delta = 0.09$ and minimum S.DbW occurred for $\delta = 0.3$. Hence these δ values are selected.

As the image segmentation problem is treated as a clustering problem, results obtained are evaluated using two cluster validity measures described in Section 3.4. Results obtained by the proposed APC algorithm are compared with those of *k*-means (KM) and mean shift (MS) algorithms (described in next section). Table 1 gives the comparative results. In the table, two results are reported for the APC method. One result corresponds to the best S.DbW value (in figures denoted as APC-best S.DbW) and the other result for the best DB value (in figures denoted as APC-best DB). For some images these two best values (APC-best DB and APC-best S.DbW) occurred for the same δ value. In those cases one image (corresponding to that δ value) is shown and indicated as APC-best DB and APC-best S.DbW in segmented image by APC method. Evaluation measures corresponding to that δ value are also shown in the table.

The CPU time, in seconds (for a Sun Fire V 890 with 2x ULTRA Sparc IV @ 1.20 GHz Processors with 16 MB Cache and 8 GB DDR1 main memory) for all the methods is also given in Table 1 for comparison.

4.2. Methods compared with

As in the present work image segmentation is viewed as a clustering problem, we have compared the proposed method with two

popular clustering based image segmentation techniques namely k -means (KM) method and mean shift (MS) method.

k-means method: Starting with k randomly-chosen patterns or k randomly defined points inside the hypervolume containing the data set, the KM algorithm (Theodoridis and Koutroumbas, 2003) repeatedly (i) (re)assign each pattern (data item) to the closest cluster center and (ii) (re)computes the current cluster centers (i.e., the average vector of each cluster in data-space). It terminates when no more reassignments of the data points take place. In this way, the intra-cluster variance, that is, the sum of squares of the differences between data items and their associated cluster centers, is locally minimized. We have used the batch version of the KM algorithm, that is, cluster centers are recomputed only after reassignment of all data items. KM is run repeatedly (20 times) using random initialization of the cluster centers and the average result is listed in Table 1. Also one typical segmented image are shown for every individual image.

Mean shift based method: This is a kernel based iterative mean shift (MS) procedure introduced by Fukunaga and Hostetler (1975). It provides an efficient way to locally estimate the density gradient. Given a set of points $\{\mathbf{x}_i\}_{i=1,2,\dots,n} \in \mathfrak{R}^d$, the multivariate density estimate with kernel $K(\mathbf{x}_i)$ and a window or hypersphere of radius h (called *bandwidth*) is computed at the point \mathbf{x}_i (Silverman, 1986). The kernel is a scalar function which must satisfy some properties defined in Fukunaga and Hostetler (1975). The mean shift vector $m_h(\mathbf{x}_i)$ of a point \mathbf{x}_i is the difference between the average of data points over a window of radius h (centered

around \mathbf{x}_i) and \mathbf{x}_i itself. The mean shift based clustering algorithm is as follows:

Initialize: Set *cluster_number* = 0, all the data points as *not_visited*, *bandwidth* and *stop_threshold* properly.

- (i) Choose any point \mathbf{x}_i from the data set (which is not yet visited).
- (ii) Compute the mean shift vector $m_h(\mathbf{x}_i)$.
- (iii) Translate the window by $m_h(\mathbf{x}_i)$ if $|m_h(\mathbf{x}_i)|$ is larger than a threshold value (called *stop_threshold*); shift the previous mean \mathbf{x}_i to $m_h(\mathbf{x}_i) + \mathbf{x}_i$ and goto step (ii).
- (iv) Store the point \mathbf{x}_i as converging point. Increase the cluster number by one.
- (v) Set the data points traversed so far as *visited*; and they will form a cluster.
- (vi) Repeat Steps (i)–(v) until all points are visited.
- (vii) Clusters having centers within a distant of $h/2$ (i.e., *bandwidth/2*) are merged.

For detailed description of the mean shift method refer to Cheng (1995) and Meer and Comaniciu (1999). In this article we have used Epanechnikov kernel (Silverman, 1986).

Note that the procedure automatically detects the number of segments depending upon the value of the *bandwidth* and *stop_threshold* and it also depends on the starting point. Even with the same *bandwidth* and *stop_threshold*, depending upon the starting point the automatically detected segments may vary. In this

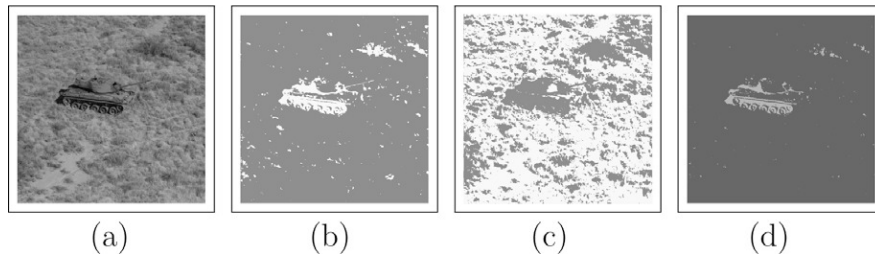


Fig. 4. (a) Original image of Tank. Segmented image by: (b) APC-best *S.Dbw* and APC-best *DB*, (c) *k*-means, and (d) mean shift.

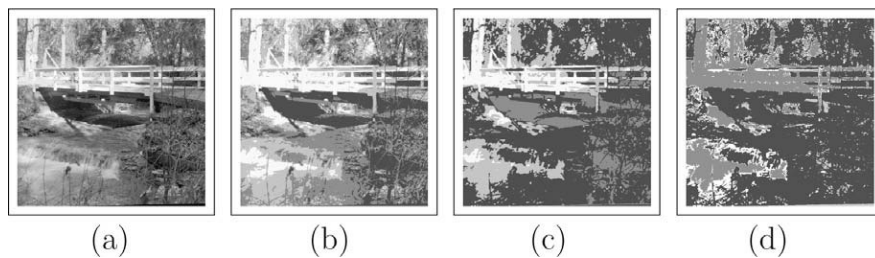


Fig. 5. (a) Original image of Stream-Bridge. Segmented image by: (b) APC-best *DB* and APC-best *S.Dbw*, (c) *k*-means, and (d) mean shift.

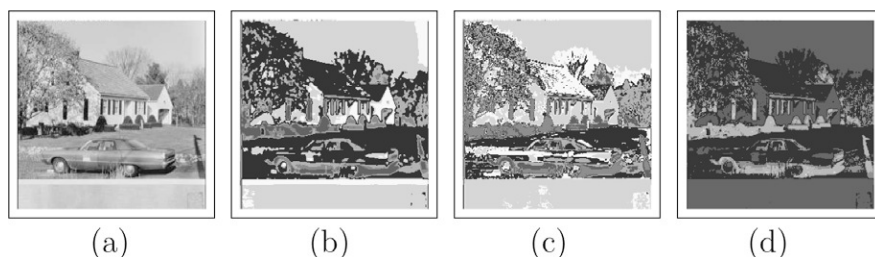


Fig. 6. (a) Original image of House. Segmented image by: (b) APC-best *DB* and APC-best *S.Dbw*, (c) *k*-means, and (d) mean shift.

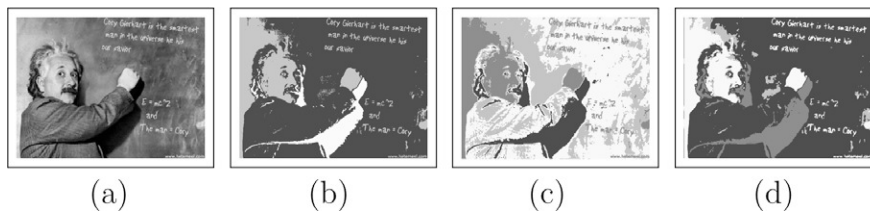


Fig. 7. (a) Original image of Einstein. Segmented image by: (b) APC-best *DB* and APC-best *S.Dbw*, (c) *k*-means, and (d) mean shift.

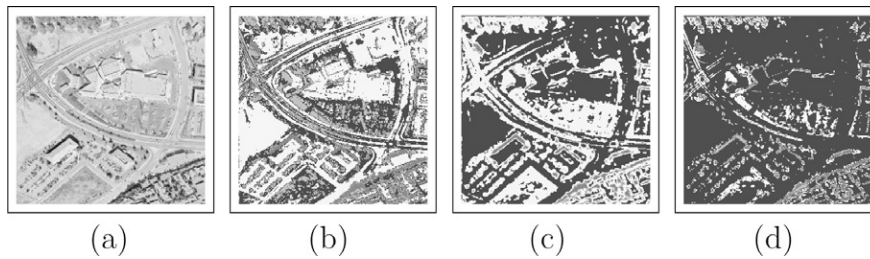


Fig. 8. (a) Original image of Aerial. Segmented image by: (b) APC-best *DB* and APC-best *S.Dbw*, (c) *k*-means, and (d) mean shift.

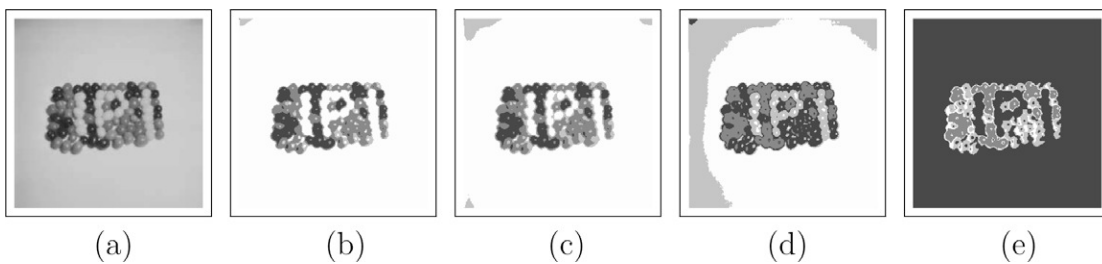


Fig. 9. (a) Original image of Marble. Segmented image by: (b) APC-best *DB*, (c) APC-best *S.Dbw*, (d) *k*-means, and (e) mean shift.



Fig. 10. (a) Original image of Lena. Segmented image by: (b) APC-best *S.Dbw* and APC-best *DB*, (c) *k*-means, and (d) mean shift.

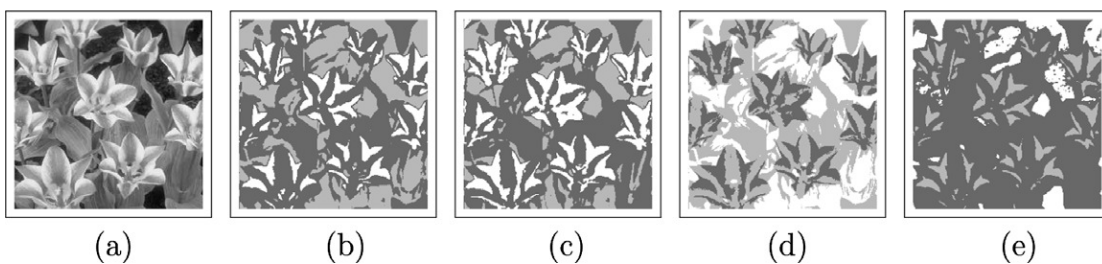


Fig. 11. (a) Original image of Tulips. Segmented image by: (b) APC-best *DB*, (c) APC-best *S.Dbw*, (d) *k*-means, and (e) mean shift.

article we have set the *bandwidth* and *stop_threshold* for each image such that the number of detected segments is the same as that of other methods (i.e., user given value for a particular image). We

have reported the average (quantitative) results of 20 runs and one typical image with the same parameter setting for every image. The average execution time of the 20 runs are also shown in

the table. The parameter values chosen are further added in the table.

4.3. Analysis of results

Fig. 4 shows the segmentation results obtained on the Tank image by all the three algorithms. As may be seen, MS algorithm fails to detect the barrel of the tank (Fig. 4d), KM algorithm segmented out the tank with lots of false classification (Fig. 4c); but the proposed APC algorithm successfully segmented the tank with very less false classification (Fig. 4b). From Table 1 the performance of the proposed APC algorithm is also found to be better than that of the other two algorithms.

Fig. 5 shows the original Stream-Bridge image and the segmented results obtained by all the three algorithms. As seen from the figures, the bridge is clearly segmented out using APC (Fig. 5b) and KM algorithm (Fig. 5c); on the other hand MS (Fig. 5d) fails to do so. But, the trees and fine bushes in the original figure are properly detected by APC, and not by other two methods. In terms of DB and S_Dbw measures also the APC method outperformed the other two.

Next, we analyze the results for the House image (Fig. 6a). Segmented versions are shown in Fig. 6b–d. It is seen from the segmented images that KM failed to detect the roof, some portion of the trees and roads. For APC method, there are some misclassification in the right side of the sky, but the houses and the fine structures of the trees are moderately identified. Though there is no misclassification in the sky & road, and also the house and trees

are segmented well, yet horizontal stirp-like structures in the road side is missing in MS (Fig. 6d) as well as in KM algorithm; but it is properly detected by the APC method. In terms of the DB and S_Dbw measures, the APC algorithm performed better than other methods.

For Einstein image (Fig. 7) segmentation results are shown in Fig. 7b–d. From the segmented results it is seen that by APC method the text are better visible than those by other methods. Also there is less noise or misclassification in the blackboard portion. In this case, KM failed to detect the blackboard as one segment. Considering the evaluation measures, APC method outperformed the others.

In the case of Arial image, from the segmentation results (Fig. 8b–d) it is seen that curvy road is better identified by APC method than MS and KM methods. Other regions of the image is also better segmented by APC compared to other two methods. Superiority of the APC method is also obvious from the evaluation measures DB and S_Dbw (Table 1).

Few other images and their segmented versions using different algorithms are shown in Figs. 9–14 and corresponding evaluation measures are shown in Table 1.

Note that among the reported results, for Lena, Tank, Stream Bridge, House, Einstein and Arial images best DB and S_Dbw values were found for only one δ value; and segmented images and validation indices corresponding to that δ value are reported in Table 1.

So from the experimental outcome it is seen that for most of the images the performance of the proposed APC method (in terms of the segmented image quality by minute visual observation and

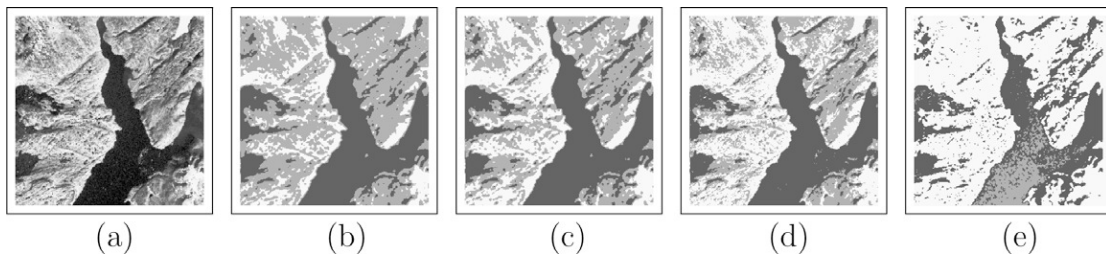


Fig. 12. (a) Original image of Iga. Segmented image by: (b) APC-best DB , (c) APC-best S_Dbw , (d) k -means, and (e) mean shift.

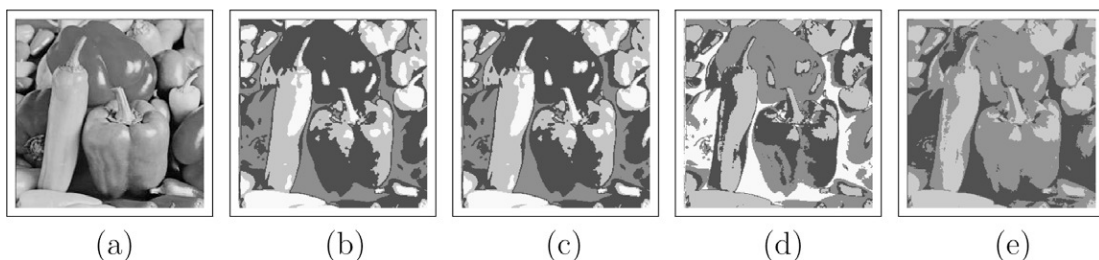


Fig. 13. (a) Original image of Pepper. Segmented image by: (b) APC-best DB , (c) APC-best S_Dbw , (d) k -means, (e) mean shift.

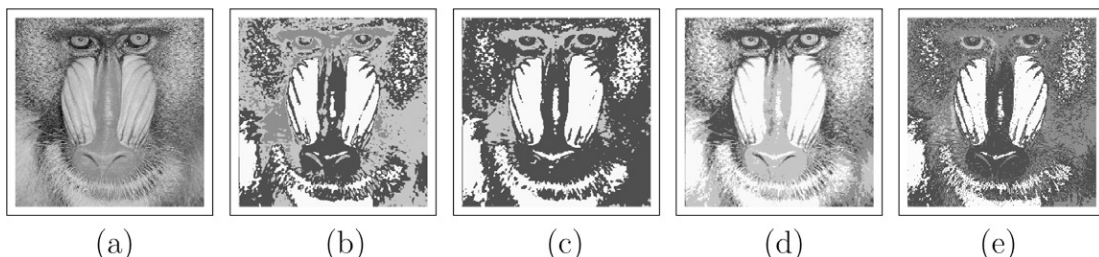


Fig. 14. (a) Original image of Baboon. Segmented image by: (b) APC-best DB , (c) APC-best S_Dbw , (d) k -means, and (e) mean shift.

also from the evaluation measures) is better (for a set of images) or comparable (for another set of images) to other existing similar methods.

The CPU time (in second) consumed by the algorithms is also put in Table 1. It is evident from the table that computational time required by the APC method is less compared to that of MS method for all the cases and KM takes the least execution time.

In brief, experimental results justify the potentiality of the proposed APC algorithm both in terms of segmentation (clustering) quality as well as execution time.

5. Conclusions

In this paper we have proposed a new algorithm for image segmentation based on the concept of aggregation pheromone density, which is inspired by the ants' property to accumulate around points with higher pheromone density. Experiments were carried out with different kinds of images to evaluate the performance of the proposed algorithm both qualitatively as well as quantitatively. Segmentation is viewed as a clustering problem and hence comparative study is made with two popular clustering algorithms namely, *k-means* and mean shift. Experimental results on a large number of different kinds of images show that the proposed method performs fairly well both in terms of the segmentation quality and execution time. In the proposed algorithm we need to specify the number of segments (clusters). Future scope of research for improvement of the algorithm will include automatic detection of the appropriate number of segments.

Acknowledgments

Authors acknowledge the Department of Science and Technology, Government of India and University of Trento, Italy, the sponsors of the project titled "Advanced Techniques for Remote Sensing Image Processing" jointly being carried out by Jadavpur University and Indian Statistical Institute, Kolkata. Also support of the Department of Science and Technology, Government of India to the Center for Soft Computing Research through its IRHPA scheme is thankfully acknowledged by Mr. Anindya Halder, Research Scholar, Center for Soft Computing Research, Indian Statistical Institute, Kolkata. The authors are also thankful to the anonymous reviewers for their stimulating comments, which helped to improve the quality of the manuscript

References

- Ball, G.H., Hall, D.J., 1965. ISODATA: A Novel Method of Data Analysis and Pattern Classification. Technical report, Stanford Research Institute, Menlo Park, CA.
- Bell, W.J., 1984. Chemo-orientation in walking insects. *Chem. Ecol. Insects*, 93–109.
- Chen, L., Xu, X.H., Chen, Y.X., 2004. An adaptive ant colony clustering algorithm. In: Proc. 3rd Internat. Conf. on Machine Learning and Cybernetics, Shanghai, pp. 1387–1392.
- Cheng, Y., 1995. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Machine Intell.* 17 (8), 233–260.
- Chu, Shu-Chuan, Lai, Mei-Shin, Huang, Hsiang-Cheh, 2006. Image texture segmentation with ant colony systems. In: Proc. 1st Internat. Conf. on Innovative Computing, Information and Control, vol. 1, ICIC'06, pp. 652–656.
- de Franca, F.O., Coelho, G.P., Zuben, F.J.V., Attux, R.R. de F., 2008. Multivariate ant colony optimization in continuous search spaces. In: GECCO'08: Proc. 10th Annual Conference on Genetic and Evolutionary Computation, New York, NY, USA, ACM, pp. 9–16.
- Deneubourg, J.L., Goss, S., Franks, N., Franks, A.S., Detrain, C., Chretien, L., 1991. The dynamics of collective sorting: Robot-like ants and ant-like robots. In: Meyer, J.A., Wilson, S.W. (Eds.), Proc. 1st Conf. on Simulation of Adaptive Behavior: From Animals to Animats, vol. 1. MIT press/Bradford Books, pp. 356–365.
- Dorigo, M., Gambardella, L.M., 1997. Ant colony system: A cooperative learning approach to the travelling salesman problem. *IEEE Trans. Evol. Comput.* 1 (1), 53–66.
- Dorigo, M., Stützle, T., 2005. *Ant Colony Optimization*. Prentice Hall of India Private Limited, New Delhi.
- Dorigo, M., Maniezzo, V., Colomi, A., 1996. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. SMC B* 26 (1), 29–41.
- Englebrecht, A.P., 2002. *Computational Intelligence: An Introduction*. John Wiley and Sons, New York.
- Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. 2nd Internat. Conf. on Knowledge Discovery and Data Mining, Menlo Park, CA. AAAI Press, pp. 226–231.
- Freixenet, J., Muñoz, X., Raba, D., Martí, J., Cufí, X., 2002. Yet another survey on image segmentation: Region and boundary information integration. In: ECCV'02: Proc. 7th Eur. Conf. on Computer Vision – Part III. Springer-Verlag, pp. 408–422.
- Fukunaga, K., Hostetler, L.D., 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inform. Theory* 21 (1), 32–40.
- Ghosh, S., Kothari, M., Ghosh, A., 2006. Aggregation pheromone density based image segmentation. In: Kalra, P., Peleg, S. (Eds.), Proc. 5th Indian Conf. on Computer Vision, Graphics and Image Processing. (ICVGIP'06), LNCS 4338. Springer-Verlag, Berlin, Heidelberg, pp. 118–127.
- Ghosh, A., Halder, A., Kothari, M., Ghosh, S., 2008. Aggregation pheromone density based data clustering. *Inform. Sci.* 178 (13), 2816–2831.
- Gonzalez, R.C., Woods, R.E., 2003. *Digital Image Processing*, second ed. Pearson Education.
- Halder, A., Ghosh, S., Ghosh, A., 2008. Aggregation pheromone density based classification. In: Proc. 10th Internat. Conf. on Information Technology (ICIT'08). IEEE Computer Society Press, pp. 100–105.
- Halkidi, M., Vazirgiannis, M., 2001. Clustering validity assessment: Finding the optimal partitioning of a data set. In: Proc. ICDM, California, USA.
- Handl, J., Meyer, B., 2007. Ant-based and swarm-based clustering. *Swarm Intel.* 1, 95–113 (Published online on 13th November 2007).
- Handl, J., Knowles, J., Dorigo, M., 2003. On the performance of ant-based clustering. In: Proc. 3rd Internat. Conf. on Hybrid Intelligent Systems, Design and Application of Hybrid Intelligent Systems. IOS press, pp. 204–213.
- Handl, J., Knowles, J., Dorigo, M., 2006. Ant-based clustering and topographic mapping. *Artificial Life* 12 (1), 35–62.
- Haralick, R.M., Shapiro, L.G., 1985. Image segmentation techniques. *Comput. Vision Graphics Image Process.* 29 (1), 100–132.
- Kettaf, F.Z., Bi, D., de Beauville, J.P.A., 1996. A comparison study of image segmentation by clustering techniques. In: Proc. ICSP'96, pp. 1280–1283.
- Kohonen, T., 1997. *Self-organizing Maps*. Springer, Berlin.
- Kothari, M., Ghosh, S., Ghosh, A., 2006. Aggregation pheromone density based clustering. In: S.P. Mohanty, A. Sahoo (Eds.), Proc. 9th Internat. Conf. on Information Technology (ICIT06). IEEE Computer Society Press, Los Alamitos, California, pp. 259–264.
- Kothari, M., Ghosh, S., Ghosh, A., 2007. Aggregation pheromone density based change detection in remotely sensed images. In: Pal, P. (Ed.), Proc. 6th International Conference on Advances in Pattern Recognition (ICAPR'07). World Scientific Publishing Co. Pvt. Ltd., Singapore, pp. 193–197.
- Liu, S., Dou, Z.T., Li, F., Huang, Y.L., 2004. A new ant colony clustering algorithm based on DBSCAN. In: Proc. 3rd Internat. Conf. on Machine Learning and Cybernetics, Shanghai, pp. 1491–1496.
- Lumer, E.D., Faieta, B., 1994. Diversity and adaptation in populations of clustering ants. In: SAB94: Proc. 3rd Internat. Conf. on Simulation of Adaptive Behavior: From Animals to Animats, vol. 3. MIT Press, Cambridge, MA, USA, pp. 501–508.
- MacQueen, L., 1967. Some methods for classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1. University of California Press, Berkeley, CA, pp. 281–297.
- Meer, P., Comaniciu, D., 1999. Distribution free decomposition of multivariate data. *Pattern Anal. Applications* 2, 22–30.
- Monmarché, N., 1999. On data clustering with artificial ants. In: Freitas, A.A., (Ed.), AAAI-99 and GECCO-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions, Orlando, Florida, pp. 23–26.
- Monmarché, N., Slimane, M., Venturini, G., 1999. On improving clustering in numerical databases with artificial ants. In: Floreano, D., Nicoud, J.D., Mondala, F. (Eds.), 5th Eur. Conf. on Artificial Life (ECAL'99), Lecture Notes in Artificial Intelligence, Swiss Federal Institute of Technology, Lausanne, Switzerland, vol. 1674. Springer-Verlag, pp. 626–635.
- Ono, M., Igarashi, T., Ohno, E., Sasaki, M., 1995. Unusual thermal defence by a honeybee against mass attack by hornets. *Nature* 377 (6547), 334–336.
- Ouadfel, S., Batouche, M., 2003. MRF-based image segmentation using ant colony system. *Electron. Lett. Comput. Vision Image Anal.* 2, 12–24.
- Ouadfel, S., Batouche, M., 2007. An efficient ant algorithm for swarm-based image clustering. *J. Comput. Sci.* 3 (7), 162–167.
- Pal, N.R., Pal, S.K., 1993. A review on image segmentation techniques. *Pattern Recognition* 26 (9), 1277–1294.
- Ramos, V., Merelo, J.J., 2002. Self-organized stigmergic document maps: Environment as a mechanism for context learning. In: Alba, E., Herrera, F., Merelo, J.J. (Eds.), Proc. 1st Internat. Conf. on Metaheuristic, Evolutionary and Bio-Inspired Algorithms, Spain, pp. 284–293.
- Ramos, V., Muge, F., Pina, P., 2002. Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies. In: del Solar Ruiz, J., Abraham, A., Koppen, M. (Eds.), *Frontiers in Artificial Intelligence and Applications, Soft Computing Systems – Design, Management and Applications*, Amsterdam, vol. 87, 2002. IOS Press, pp. 500–509.
- Sezgin, M., Sankur, B., 2004. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging* 13 (1), 146–168.
- Silverman, B.W., 1986. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York.

- Socha, K., Dorigo, M., 2008. Ant colony optimization for continuous domains. *Eur. J. Operat. Res.* 185 (3), 1155–1173.
- Sukama, M., Fukami, H., 1993. Aggregation arrestant pheromone of the German cockroach, *Blattella germanica* (L.) (Dictyoptera: Blattellidae): Isolation and structure elucidation of blastellastanoside-A and B. *J. Chem. Ecol.* 19, 2521–2541.
- Tao, W., Jin, H., Liu, L., 2007. Object segmentation using ant colony optimization algorithm and fuzzy entropy. *Pattern Recognition Lett.* 28, 788–796.
- Theodoridis, S., Koutroumbas, K., 2003. *Pattern Recognition*, second ed. Elsevier Academic Press, Amsterdam.
- Tsutsui, S., Ghosh, A., 2004. An extension of ant colony optimization for function optimization. In: *Proc. 5th Asia Pacific Conference on Simulated Evolution and Learning (SEAL04)*, Pusan, Korea.
- Tsutsui, S., 2004. Ant colony optimization for continuous domains with aggregation pheromones metaphor. In: *Proc. 5th Internat. Conf. on Recent Advances in Soft Computing (RASC'04)*, United Kingdom, pp. 207–212.
- Tsutsui, S., Pelikan, M., Ghosh, A., 2005. Performance of aggregation pheromone system on unimodal and multimodal problems. In: *Proc. Congress on Evolutionary Computation*, Menlo Park, CA, IEEE, pp. 880–887.
- Vizine, A.L., de Castro, L.N., Hruschka, E.R., Gudwin, R.R., 2005. Towards improving clustering ants: An adaptive ant clustering algorithm. *Informatica* 29, 143–154.
- Vorhees, E., 1985. The Effectiveness and Efficiency of Agglomerative Hierarchical Clustering in Document Retrieval. Ph.D. Thesis, Department of Computer Science, Cornell University, Ithaca, NY.
- Wang, X.N., Feng, Y.J., Feng, Z.R., 2005. Ant colony optimization for image segmentation. In: *Proc. 4th Internat. Conf. on Machine Learning and Cybernetics*, pp. 5355–5360.
- Zhang, Y.J. (Ed.), 2006. *Advances in Image and Video Segmentation*, first ed. IRM Press, USA.