AntHeaps: A New Hybrid Image Segmentation Algorithm using Ant Colonies

Asmaa Reda
Faculty of Computers and Information, Fayoum University
Fayoum, Egypt

Hala Abdel-Galil
Faculty of Computers and Information, Helwan University
Cairo, Egypt

Atef Z. Ghalwash
Faculty of Computers and Information, Helwan University
Cairo, Egypt

ABSTRACT
During the last few years, the Segmentation problem has been tackled from different disciplines. Many algorithms have been developed to solve this problem. AntClust algorithm is an ant-based algorithm that uses the self-organizing and autonomous brood sorting behavior observed in real ants for unsupervised partitioning. A population of artificial ants provides an image segmentation of the relevant classes without any previous knowledge about the number of classes needed. This paper proposes a hybrid solution based on AntClust algorithm and data mining (e.g., Kmeans). Experimental results demonstrate that the proposed solution is able to extract the correct number of clusters with better clustering quality and execution time compared to the results obtained from AntClust algorithm.

Keywords
Image Segmentation, Image Clustering, AntClust algorithm.

1. INTRODUCTION
Image segmentation is a fundamental task in a vision system. Its purpose is to subdivide an image into meaningful non-overlapping regions [1]. It may consist of two processes: recognition and delineation. Recognition is the process of determining nearly the whereabouts of an object of interest in the image. Delineation is the process of determining the accurate locative scope and point-by-point composition of the object in the image. In solving the segmentation problems, humans are more qualitative than computerized algorithms, whilst, computerized algorithms are more quantitative but less qualitative. This weakness of computers is the most of current segmentation methods drawbacks as it remained a challenge to amalgamation high-level expert human knowledge into the computer.

There are many methods to segmentation, for example: Thresholding, Compression-based methods, Histogram-based methods, Edge detection, and Clustering method. In clustering method segmentation aims to partition the image into clusters, so it may be viewed as a clustering problem. In each cluster, pixels are as homogenous as possible whereas the clusters are as heterogeneous as possible among each others with respect to a similarity measure. Kmeans algorithm[1] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problems. It is an iterative technique that is used to partition an image into K clusters.

[1] http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

AntClust algorithm is an algorithm based on ant-based systems, it solves image segmentation problems using the principles of stochastic and distributed exploration in a population of artificial ants. Ants move in image pixels picking up/dropping a pixel from/in clusters according to a similarity function, which measures the pixel similarity with other pixels in a cluster. In this way, ants cluster pixels into distinctive independent groups. Quadfel and Batouche [1] demonstrated the ability of AntClust to extract the correct number of clusters and to give better clustering quality compared to those obtained from Kmeans algorithm. To take advantage of Kmeans and AntClust and avoid their drawbacks, this paper propose a new hybrid algorithm that executes the AntClust algorithm with a limited number of iterations, then speeding up convergence with the Kmeans algorithm and using hierarchical clustering on heaps of objects.

The reminder of this paper is organized as follows: Section 2 describes some current ant-based systems; Section 3 submits the proposed algorithm; Section 4 states the experimental analysis and results to show the proposed algorithm effectiveness; Section 5 concludes the work.

2. RELATED WORK
In ant-based approaches, one of the first studies on this ant behavior has been done by Deneubourg [3], using simple local rules and without any central control upon a population of simple ants that cluster objects together. The objects to be collected are randomly placed on a 2D grid representing ant's environment. Ants are modeled by simple agents that are randomly placed on the grid in order to move basic objects so as to classify them according to the similarity of these objects and other objects in the immediate environment-neighborhood.

This algorithm has been further developed by Lumer and Faieta [4] with extending its application to clustering objects that represent records in a numerical data set with different data representation on similarity and the principle of neighborliness. And then by Kuntz and Snyers [5] studied a real clustering problem, in order to efficiently resolve an optimization problem.

Monmarche in [6] introduced a classification algorithm based on Lumer and Faieta with basic modification. This algorithm called "AntClass". Using a 2D toroidal grid, it introduces heuristics for the ant colony and hybridization with the Kmeans algorithm.

These algorithms have become well-known models. From these basic models, some works have been done to improve the clustering quality. One of these works named AntClust algorithm. In AntClust Quadfel and Batouche [1] replaced the grid that represented the environment with an array of N cells. Experiments demonstrated the ability of AntClust to extract the correct number of clusters and give better clustering quality compared to those obtained from Kmeans algorithm.

AntClust was inspired by brood sorting behavior observed in real ants. A population of artificial ants provided image segmentation by moving on pixels through an iterative process picking up and dropping pixels to gather it into clusters. AntClust improved results by increasing the number of iterations; hence, the time complexity was affected which is one of the problems to be recovered by the proposed algorithm.

Based on the previous work, a hybrid algorithm based on AntClust and Kmeans algorithms is proposed in this paper to improve the segmentation in terms of execution time and quality. It executes the AntClust algorithm with a limited number of iterations to initiate the number of clusters, then speeding up convergence with the Kmeans algorithm. Finally, hierarchical clustering on heaps of objects is used rather than on individual objects followed by Kmeans algorithm once more to improve the results and remove the classification error.

3. THE PROPOSED ANTHEAPS ALGORITHM

AntHeaps is a hybrid algorithm of AntClust and Kmeans algorithms. Initially, AntClust algorithm is applied but with a limited number of iterations, which helps in finding an initial solution which is not optimal but a seed for the following steps towards the optimal solution.

Initially, AntClust algorithm places the pixels in an array and defines the number of ants. Ants can pick up or drop pixels in order to cluster them into homogenous clusters. So, in the beginning, each ant picks up a pixel randomly. Then, the clustering phase started through an iterative process. For each ant, if it picked a pixel it would search for a cell and decides, based on a probabilistic rule, whether to drop the pixel or not. If the ant does not pick a pixel "free ant", it will search for a new pixel to pick up. Knowing the list of all pixels that are not carried by ants, the ant randomly chooses one of these free pixels according to a probabilistic rule and decides whether or not to pick up that pixel and this process will be repeated for all ants along all iterations. When all iterations finished, cells of pixels are obtained in which each cell presents a cluster from our image.

The Kmeans needs k centroids, one for each cluster to be defined. Calculating the center for each cell is required to be the input for the Kmeans algorithm. Kmeans assigns each object to the class that has the closest centroid. Then recalculate the position of each centroid. This process is repeated until the centroids no longer change. Kmeans helps remove small classification errors as well as assign free pixels (pixels left alone with no cluster) to classes. At this stage, too many but homogenous clusters are still exist.

Subsequently, the AntClust algorithm is applies once more on heaps of pixels (classes) rather than single pixels. Using the results obtained from the Kmeans algorithm, AntClust treats classes as if they were pixels, where classes are picked up and dropped based on the distance between two heaps (the distance between their centers). It hierarchically builds more important classes. The real number of classes is very well approximated, but there are still some classes which are not assigned. Therefore, calculating the centers again for each cell obtained by the last step is required to pass it to the Kmeans algorithm to get the final partition. Kmeans her removes the classification errors, and also assigns free cells. The final result will be of high quality since the input partition given to the Kmeans is very close to the optimal one.

So, AntHeaps algorithm steps are AntClust algorithm for clustering pixels, and then the Kmeans algorithm with results obtained from AntClust, followed by AntClust algorithm but on heaps previously found, and finally the Kmeans algorithm once more (see Figure 1).

```
Initialization phase
Reading an image and put each pixel Pi value in a cell of one dimensional  array "Pixels"

First step: Calling AntClust
[Cells] = AntClust (Iteration Max Number, Ants Number, Pixels Array)
For each cell ci in Cells
Cell Center = mean value for all ci pixels (gray level)
endfor

Second step: Calling Kmeans
[New Centers] = Kmeans (Pixels Array, Cell Center)

Third step: Calling Antclust for heaps
[Cells] = AntClust (Iteration Max Number, Ants Number, New Centers Array)
For each cell ci in Cells
Cell Center= mean value for all ci cells(cell center)
endfor

Fourth step: Calling Kmeans
[New Centers] = Kmeans (Pixels Array , Cell Center)
```

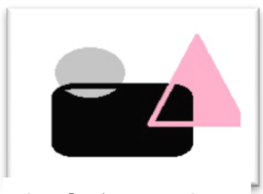
Fig. 1 Proposed AntHeaps algorithm

4. EXPERIMENTAL RESULTS

4.1 Test Images

The AntHeaps algorithm was tested on synthetic, real, and medical images. The real images were selected form segmentation evaluation database [7]. The medical images are parts from a CT images for a liver and it has been cropped to obtain the liver only. The synthetic images are supervised (class is known for each object) in order to assess the quality of partitioning, and evaluate the results obtained by the presented algorithm. Otherwise, in real and medical images, the actual number of classes is unknown for each object, so manual clustering is used. Only two images from each type are shown in Figures 2, 3, and 4. Table 1 lists the number of clusters (K) for each image.

Synthetic Image1

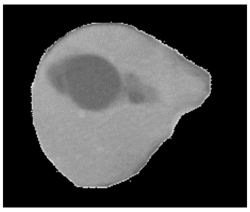
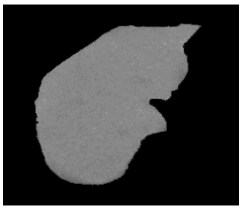


Synthetic Image2

Fig. 2: Synthetic test images


Real Image 1 **Real Image2**

Fig. 3: Real test images

Medical Image1 **Medical Image2**

Fig. 4: Medical test images

Table 1: Characteristics of test images

	K
Synthetic Image1	2
Synthetic Image 2	4
Real Image 1	[5-10]
Real Image 2	[2-4]
Medical Image 1	[5-7]
Medical Image 2	[2-3]

4.2 Measures of Clustering Quality

In order to formally evaluate the quality of the algorithm on the test images that are already labeled in reference segmentation, the results are compared to desired results in the reference segmentation. Number of classes, run time, the rand index, classification error, and the numerical criteria such inertia interclass and intra-class inertia are the commonly used measures for the quality of clustering [8]. They are defined as follows:

4.2.1 Rand Index R

It determines the frequency of pixels (pairwise co-assignments) classified correctly by the total number of pixels. (Eq. (1))

$$R = \frac{a+d}{a+b+c+d} \qquad (1)$$

Where a, b, c, and d are computed for each couple of pixels as following:

$$a = |\{ i , j \setminus c_{ref}(i) = c_{ref}(j) \wedge c_{seg}(i) = c_{seg}(j) \}| \qquad (2)$$

$$b = |\{ i , j \setminus c_{ref}(i) = c_{ref}(j) \wedge c_{seg}(i) \neq c_{seg}(j) \}| \qquad (3)$$

$$c = |\{ i , j \setminus c_{ref}(i) \neq c_{ref}(j) \wedge c_{seg}(i) = c_{seg}(j) \}| \qquad (4)$$

$$d = |\{ i , j \setminus c_{ref}(i) \neq c_{ref}(j) \wedge c_{seg}(i) \neq c_{seg}(j) \}| \qquad (5)$$

Where $c_{ref}(i)$ and $c_{seg}(i)$ are the labels of cluster of pixel (i), in the reference segmentation, and in the results obtained by the clustering algorithm respectively.

4.2.2 Classification Error E:

Determines the frequency of pixels (pairwise co-assignments) classified wrongly by the total number of pixels. (Eq. (6))

$$E = \frac{b+c}{a+b+c+d} \qquad (6)$$

4.2.3 The Time Complexity:

As a measure of algorithm quality, where the execution times of both the proposed algorithm and AntClust algorithm are compared when they are implemented with the same programming language and run on the same machine.

4.2.4 The Inertia Interclass and Intra-Class Inertia M:

Measures the homogeneity of each class in the segmented image and disparity between the classes [1]

$$M = \frac{D(seg)+1-d(seg)}{2} \qquad (7)$$

Where D(seg) represents the intra-region disparity, and

$$D(seg) = \frac{1}{NC}\sum_{K=1}^{NC}\frac{|C_k|}{|seg|} D(C_k) \qquad (8)$$

NC is the number of clusters, $|C_k|$ is the number of pixels in cluster k, $|seg|$ is the number of pixels in the segmented image, and $D(C_k)$ is the disparity of cluster k.

$$D(C_k) = \frac{2}{255}\sqrt{\frac{1}{|C_k|}\sum_{p_i \in C_k} ng(p_i) - \frac{1}{|C_k|^2}\left(\sum_{p_i \in C_k} ng(p_i)\right)^2} \qquad (9)$$

$ng(p_i)$ is the gray level of the pixel p_i

And $d(seg)$ measures the inter-classes disparity:

$$d(seg) = \frac{1}{q^{(k)}}\sum_{j=1}^{q^{(k)}} d(C_k, C_j) \qquad (10)$$

$q^{(k)}$is the number of clustersC_j that are neighbor to the cluster C_K

$$d(C_k, C_j) = \frac{|CC_k - CC_j|}{NG} \qquad (11)$$

CC_j is the center of cluster C_j ,and NG is the total gray level in the image.

Since M in (Eq. (7)) measures homogeneity and disparity between the classes, a smaller value of M indicates a better clustering algorithm.

4.3 Results

The performance of the clustering process performed by the presented algorithm is compared with that performed by AntClust algorithm.

4.3.1 The Execution Time (E):

AntHeaps uses Kmeans in two steps within its four steps that improved the obtained results. In contrary, AntClust improves results by increasing number of iterations. Image size affect the time, largest image needs time more than others. Figure 5 shows the execution time for each test image (in seconds), and demonstrates that AntHeaps improves the time complexity problem on AntClust.

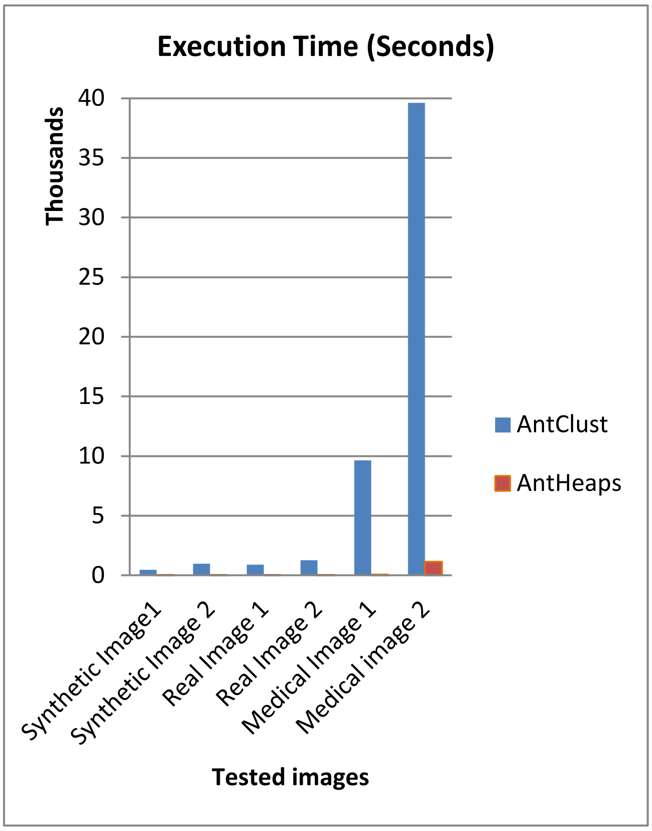


Fig. 5: Variation of execution time

4.3.2 The Number of Classes:

Both number of classes in AntClust and AntHeaps are comparable and analogous to optimal (see Figure 6).

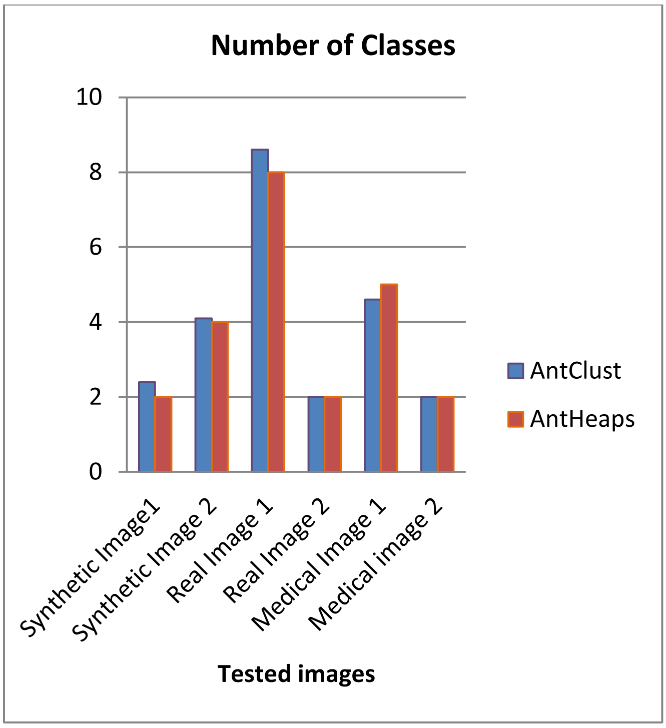


Fig. 6: Variation of number classes

4.3.3 Classification Error E:

The use of Kmeans in AntHeaps for more research improved results and makes AntHeaps more efficient in term of error rate. E is calculated only for the synthetic images as they are supervised (class is known for each object) (see Figure 7).

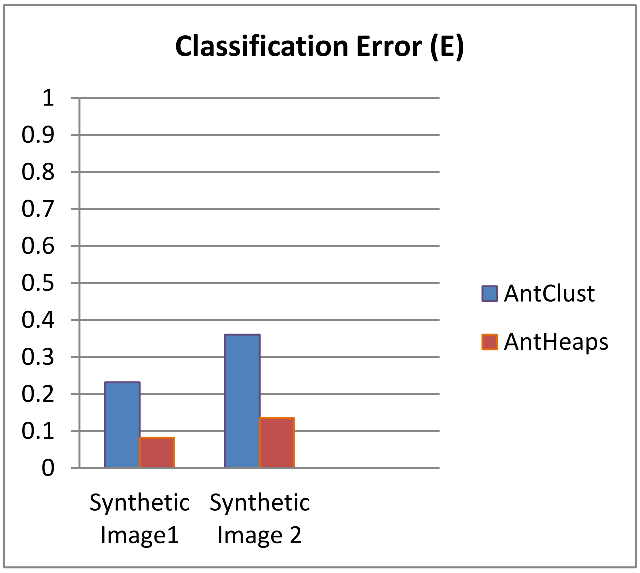


Fig. 7: Variation of classification error (E)

4.3.4 The inertia interclass and intra-class inertia measure M:

The lower value of M, the better is the clustering algorithm. The obtained results show that AntHeaps outperforms AntClust in all tested images (see Figure 8).

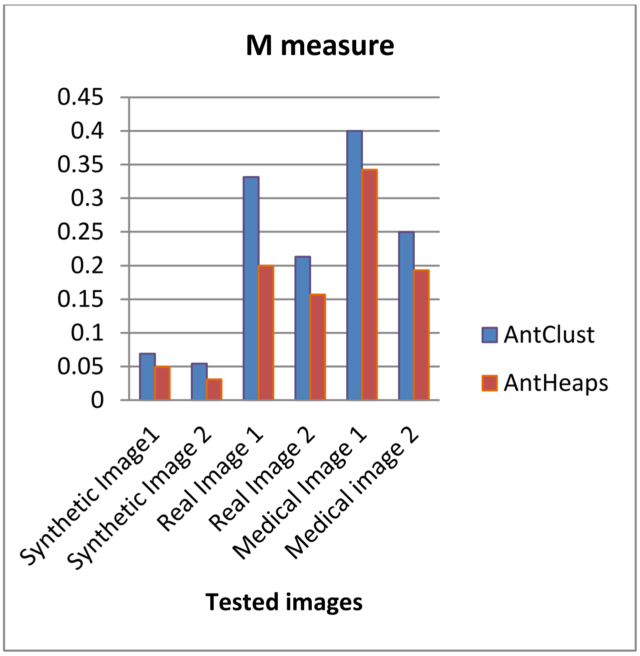


Fig. 8: Variation of M measure

Different images were used to evaluate AntHeaps and the above results show that AntHeaps surpass AntClust in term of quality, speed, and all other quality measures.

5. CONCLUSION

In this paper, a hybrid solution based on AntClust and Kmeans algorithms is proposed. It first used AntClust algorithm to create an initial partition followed by the Kmeans algorithm to improve the quality of classification with respect to time constrain. In the second phase, hierarchical clustering was used where ants acted with on heaps of pixels rather than individual pixels. This helped in approximating the real number of classes, followed again with Kmeans to finalize this partitioning with results close to the optimal one. Experimental results on images demonstrate that the proposed algorithm outperforms the AntClust algorithm in extracting the correct number of clusters with better clustering quality and execution time.

6. REFERENCES

[1] S. Quadfel and M. Batouche, "An Efficient Ant Algorith for Swarm-Based Image Clustering," Journal of Computer Science, vol. 3, no. 3, pp. 162-167, 2007.

[2] M. Dorigo, M. Birattari and T. Stützle, "Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique," IEEE Computational Intelligence Magazine, pp. 28-39, November 2006.

[3] J. Deneubourg, S. Goss and N. Franks, "The Dynamic of Collective Sorting Robot-like Ants and Ant-like Robots," in the first conference on Simulation of Adaptive Behaior, 1990.

[4] E. Lumer and B. Faieta, "Diversity and Adaptation in Populations of Clustering Ants," in The 3rd International Conference on the Simulation of Adaptive Behavior, 1994.

[5] P. Kuntz and D. and Snyers, "Emergent colonization and graph partitioning," in the third Tnternational Conference on simulation of Adabtive Behavior, 1994.

[6] N. Monmarche, "On Data Clustering with Artificial Ants," Data Mining with Evolutionary Algorithms: Research Directions – Papers from the AAAI Workshop, pp. 23-26, 1999.

[7] S. Alpert, M. Galun, R. Basri and A. Brandt, "Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[8] M. Hameurlaine, A. Moussaoui and H. Cherroun, "AntMeans: A New Hybrid Algorithm based on Ant Colonies for Complex Data Mining," International Journal of Computer Applications (0975 – 8887), vol. 60, no. 17, December 2012.

[9] M. Chong and . M. Munusamy, "A Hybrid Ant-Based Clustering Algorithm," The International Federation for Information Processing, vol. 187, pp. 247-256, 2005.

[10] O.A. Mohamed and R. Sivakumar, "Ant-based Clustering Algorithms: A Brief survay," International Journal of Computer Theory and Engineering, vol. 2, no. 5, pp. 787-796, October 2010.

[11] J. Smaldon and A. A. Freitas, "A New Version of the Ant-Miner Algorithm Discovering Unordered Rule Sets," in The 8th annual conference on Genetic and evolutionary computation, New York, 2006.

[12] W. Omar, A. Badr and A. E.-F. Hegazy, "Hybrid Ant-Based Clustering Algorithm with Cluster Analysis Teshniques," Journal of Computer Science, vol. 9, no. 6, pp. 780-793, september 2013.

[13] M. Dorigo and T. Stützle, Ant Colony Optimization, Massachusetts Institute of Technology, 2004.

[14] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," Theoretical Computer Science, vol. 344, p. 243 – 278, 2004.

[15] N. Labroche, C. Guinot and . G. Venturini, "Fast Unsupervised Clustering with Artificial Ants," Lecture Notes in Computer Science, vol. 3242, pp. 1143-1152, 2004.