An improved ant colony algorithm for fuzzy clustering in image segmentation

Yanfang Han*, Pengfei Shi
Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University, Shanghai 200030, PR China
Available online 15 October 2006

Abstract

Ant colony algorithm (ACA), inspired by the food-searching behavior of ants, is an evolutionary algorithm and performs well in discrete optimization. In this paper, it is used for fuzzy clustering in image segmentation. Three features such as gray value, gradient and neighborhood of the pixels, are extracted for the searching and clustering process. Unexpectedly, tests show that it is time consuming when dealing with the vast image data. In view of this drawback, improvements have been made by initializing the clustering centers and enhancing the heuristic function to accelerate the searching process. Experiments and comparisons are done to show that the improved ACA-based image segmentation is an efficient and effective approach.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Ant colony algorithm; Fuzzy clustering; Image segmentation; Feature extraction

1. Introduction

Image segmentation, which plays an important part in image processing, has always been one of the most difficult tasks due to the complexity and diversity of images. Influencing factors range from illumination, image contrast and image noises to even image size. Therefore, great interest has been shown in this area and numerous approaches have been proposed. Generally, most of these approaches are based on two strategies, e.g. generating regions or recognizing contours considering homogeneity [16] or discrepancy of image features. More specifically, traditional techniques such as thresholding [1], region growing [4], template matching [19] and characteristic feature clustering [6] are all grounded on obtaining regions adopting homogeneity of image features. While various edge detection operators, utilizing the discrepancy of image features, segment the image by searching the contours of different objects. In many applications, these approaches are proved to be successful. However, none of them are generally applicable to all images and different algorithms are usually not equally suitable for a particular application. In other words, these approaches, from time to time, show their advantages and disadvantages, respectively when dealing with different problems. For example, thresholding technique, which segments images by comparing the image features with one threshold and is claimed to be one of the simplest approaches in image segmentation, is preferred for its less calculation and high efficiency when the image histogram has two obviously separated peaks, usually need much more efforts in threshold selecting so as to ensure segmenting quality and rapidity otherwise. In addition, the sensitivity to noises is another challenge to this approach. As far as template matching is concerned, it is simple in principle, since it is in essence a convolution between the template and image matrices, but becomes time consuming when the image becomes more complex or the image size is larger [13]. Furthermore, the designing of the template always needs much mathematical effort. Morphological methods [9] are well known since they can reduce the influence of noises to great extent. But the repetitive use of open, close, eroding and dilating operation will lead to excessive smoothing, which results in the loss of details and deformity of the images. Clustering methods, viewing an image as a set of multi-dimensional data and
classifying the image into different parts according to certain homogeneity criterion such as Euclidean distance, can get much better segmenting results with help of fuzzy theory [12], neural networks [17] and evolutionary techniques [7]. But over-segmentation is the problem must be settled. Besides, feature extraction is an important task before clustering. Region growing [3], which begins with seeds and attempts to merge the neighboring pixels into the growing regions, often exploits splitting and merging techniques during the growing process to get better results. Likewise, there exist the over-segmentation and time-consuming problems again. When it comes to the edge detection approaches, which searching contours with respect to discrepancy of image features, one and two order difference operators [10] as well as the gradient operators [11] such as Sobel [18], Prewitte, Log and Canny are widely used. However, based on these operators, noises are usually mistaken as edges. So how to detect edges without or with fewer influence of noises is a major task.

In view of the problems mentioned above, plenty of approaches and their corresponding improvements have been proposed to ensure the accuracy and rapidity of image segmentation. But there is still much work to be done to overcome their drawbacks. Considering the abundant approaches, it is difficult to present a totally new theory; however, attempts at utilizing knowledge on other domains, especially artificial intelligence, should be highly appreciated.

Actually, intelligent approaches such as neural networks [14] and genetic algorithms have already been used widely and successfully in image segmentation. The working mechanism for neural network is that by training the neurons using training samples, parameters such as weights connecting neurons between neighboring layers can be gradually stabilized, which is called a learning process. Then, new samples can be classified with the trained network. Ref. [5] gives a detailed survey of its application in image segmentation. In terms of genetic algorithm, which mainly uses two operators such as crossover and mutation for ‘offspring’ generation and adopts one fitness function to judge whether the population is good, it is often used to learn processing parameters of other segmenting systems such as neural networks. In Ref. [20], genetic algorithm is used to learn parameters in the morphology process.

This paper investigates an evolutionary method of ant colony algorithm (ACA) [15] and its application in image segmentation. Unlike traditional clustering methods, ACA is an intelligent approach, which is successfully used in discrete optimization. Its convergence ability is highly preferred in virtue of its parallel and positive feedback ability. Here, it is used to cluster the pixels for image segmentation. Three features, including intensity, gradient, neighborhood of the pixels, are extracted synthetically for better application. Then, these features are used for ACA clustering. As a parallel, robust and evolutionary method, ACA should have improved the clustering efficiency. However, unfortunately, it turns out to be time consuming due to the vast amount of multi-dimensional image data. According to this drawback, two measures, including the improvement of the heuristic function and the initialization of the clustering centers, are taken to accelerate the clustering process. Consequently, the searching time of ACA is reduced greatly. Experiments and comparisons show that improved ACA is an efficient approach in image segmentation.

The rest of this paper is organized as follows: Section 2 describes ACA and its application in image segmentation. More specifically, the feature extraction process is introduced in detail and the testing results of ACA are illustrated and analyzed. Then, according to the drawbacks of ACA in image segmentation, Section 3 elaborates two measures to improve ACA. One is to enhance the heuristic function and the other is to initialize the clustering centers, both aim at reducing the computational quantity to improve the efficiency of ACA. After that, Section 4 gives the experimental results of improved ACA, which are compared with those of Sobel, Canny and Watershed algorithm. Finally, a conclusion is drawn in Section 5.

2. ACA in image segmentation

2.1. Description of ACA

As social insects, ants exhibit great organization and construction abilities by the colony behaviors. One of the most important and fascinating one is their food searching behavior, in particular, how they can find the shortest paths between the food sources and their nest. While walking from their nest to the food sources and vice versa, ants deposit on the way a kind of substance called pheromone whose concentration becomes weaker with the elapse of time due to evaporation, forming in this way a pheromone trail. Ants can smell the pheromone and when choosing their way, they tend to choose, in probability, the paths with the strongest pheromone concentration. And the more ants choose the path, the stronger the pheromone concentration. Thus the pheromone trail can help the ants find quickly the shortest way (the most preferred way) to the food sources.

Inspired by the food searching behaviors of ants, Marco Dorigo et al. [15] proposed ACA. Mathematically, the algorithm is described as follows.

When going from location $i$ to location $j$, each ant $X_i$ ($i = 1, ..., n$) lays pheromone $p_{ij}$ on path $(i, j)$.

Each ant chooses the path with a probability $p_{ij}$, which is a function of the path length, $d_{ij}$ and the pheromone concentration, $p_{ij}$ on the path. The computing formulae are as follows:

$$d_{ij} = \sqrt{\sum_{k=1}^{m} p_k (x_{ik} - x_{jk})^2},$$

(1)
\[ ph_{ij} = \begin{cases} 1 & d_{ij} \leq r \\ 0 & d_{ij} > r \end{cases}, \]

\[ p_{ij} = \frac{\sum_{a=1}^{m} p_{a}^2(0_{a}) \cdot \eta_{ij}(t)}{\sum_{a=1}^{m} p_{a}^2(0_{a})}, \quad j \in S, \]

\[ = 0, \quad \text{otherwise} \]

where \( m \) is the dimension of vector \( X \), \( p \) the weight decided by the importance of each component and \( r \) the clustering radius. \( ph_{ij}(t) \) represents the pheromone concentration on path \((i, j)\) at time \( t \). \( \eta_{ij}(t) = 1/d_{ij} \) denotes the heuristic value when moving from \( i \) to \( j \). \( S = \{X_{S}/d_{S} \leq r, S = 1, 2, \ldots, N\} \) is the set of all available paths. \( x \) and \( \beta \) are two parameters that control the relative weight of the pheromone concentration and the heuristic value. When \( x = 0 \), the most closest path is likely to be selected, which corresponds to a classical stochastic greedy algorithm (with multiple starting points since ants are initially randomly distributed). If on the contrary \( \beta = 0 \), then only pheromone amplification is at work, this will lead to the rapid emergence of a stagnation situation with the corresponding generation of tours which, in general, are strongly suboptimal. A trade-off between heuristic value and pheromone concentration therefore appears to be necessary.

While walking, the pheromone concentration on the paths will change. After one cycle, the pheromone concentration is adjusted as follows:

\[ ph_{ij}(t) = \rho ph_{ij}(t) + \Delta ph_{ij}, \]

where \( \rho \) represents the evaporating degree of pheromone concentration with the elapse of time. \( \Delta ph_{ij} \) is the increase of pheromone concentration on path \((i, j)\) after this cycle:

\[ \Delta ph_{ij} = \sum_{k=1}^{N} \Delta ph_{ij}^k, \]

where \( \Delta ph_{ij}^k \) is the pheromone concentration left on path \((i, j)\) by the \( k \)-th ant. In general, ACA is an evolutionary algorithm after genetic algorithm [8]. As a combination of distributed computation, positive feedback and constructive greedy heuristic, it is rapid in discovering good solutions for optimization and can avoid premature convergence at the same time. These properties help a lot and make ACA an efficient tool in fuzzy clustering.

2.2. Feature extraction

Roughly speaking, background, object, edges and noises are what constitute an image. Image segmentation by fuzzy clustering is in essence to classify the different contents into different classes. Where, features are primary elements, which must be representative and comprehensive. Feature extraction, which influences not only the representation of image information but also the accuracy and efficiency of the subsequent algorithm, is significant in image segmentation. By analyzing the image, a feature extraction method is proposed in this paper.

Gray value, which is an important and most-often used feature, is adopted as the first feature in this paper. Due to the fact that edges and noise pixels always have abrupt changes in gray values while background and object ones always account for flat regions, the gray value gradient, which well represents the gray value changes, is employed as another important feature when differentiating edge and noise pixels from the background and object ones. As for the high-gradient pixels such as the noise pixels and edge ones, the sizes of their neighboring regions in the image are different. Noise pixels always account for much smaller regions while the edge ones the larger. Based on this fact, the 3 x 3 neighboring regions of each pixel \((i, j)\) is considered and the feature extracting process is described as follows:

For the convenience of expression, we set a variant \( Num = 0 \).

For each pixel in the neighboring region, compute the difference of gray value with pixel \((i, j)\). Then, compare this difference with a given threshold \( T \), if the difference is less than \( T \), then \( Num = Num + 1 \). Repeat this process over the neighboring regions of each pixel and ensure each pixel in the image get a \( Num \).

Thus, \( Num \) represents the number of similar pixels in the neighboring region. Generally, for the object and background pixels, \( Num = 8 \). And it is larger for the edge pixels than for the noise pixels. Where, threshold \( T \) is different for different images. For more smoothing or flat images, \( T \) is smaller. While for images with more details, \( T \) is larger. Experimentally, \( T \) usually varies from 50 to 90.

From the above, we get the three-dimensional (3D) feature vector whose components are the gray value, gradient and \( Num \). Since \( Num \) reflects the neighboring attributes, it is called neighboring feature in this paper.

2.3. Testing results of ACA for image segmentation

From the above analysis, image segmentation is viewed as a fuzzy clustering one, which can be facilitated by ACA. Based on this idea, tests are done.

Given 256 gray levels, \( 224 \times 323 \) image as Fig. 1, we employ ACA into image segmentation. The algorithm is described as follows:

Step 1. Feature extraction: By extracting features following the approaches mentioned above, a 3D data set is obtained with each element being seen as an ant. Step 2. For each ant \( k \), calculate the distance to all the other ants by formula (1).

Step 3. According to formulae (2) and (3), calculate the probability.

Step 4. If the probability is larger than the given \( \lambda \), then merge \( X_i \) with the ants to form a new class. Refresh the pheromone concentration following formulae (4) and (5). Otherwise, leave it into a new set waiting for next cycle.
Step 5. Calculate the clustering centers by the following formula:

$$\bar{C}_j = \frac{1}{J} \sum_{K=1}^{J} X_k.$$  

(6)

Step 6. Calculate the distance between the clustering centers. If the distance is less than the given threshold $\varepsilon$, merge the classes. Refresh the clustering centers by (6).

Step 7. If there is any unclassified ant, take the clustering centers as new ants, then goto step 2. Otherwise, end.

When $\alpha = 1$, $\beta = 1$, $r = 50$, $\lambda = 0.9$ and $T = 80$, the result image is shown in Fig. 2.

The test is done under the environment of PIII800, Win2000, Matlab6.5. Successfully, the boundaries, especially the darker parts of Fig. 1 are detected. However, it takes more than 10 min to gain the result, which greatly reduces its value of application. Time consuming is relative to the computational complexity and quantity. From the above steps, we can find that the computational quantity is vast. For a $m \times n$ image, each ant will have one distance and probability computation with other ants. Thus, the quantity is at least $2[(m \times n-1) + (m \times n-2) + \cdots + 1] = (m \times n) \times (m \times n-1)$. For Fig. 1, the number is larger than $5 \times 10^7$. Since each ant is 3D, the complexity and quantity will triple.

3. Improved ACA

As a parallel, robust and evolutionary method, ACA performs well in discrete optimization and should have improved the clustering efficiency. However, from the above test in image segmentation, it turns out to be time-consuming in view of the vast amount of multi-dimensional image data. Due to this drawback, we will consider reducing the computational quantity to improve the efficiency of ACA. To do that, two measures are taken. One is to set initial clustering centers by using the information of original image. The other is to improve the heuristic function to accelerate the searching process. And these measures are elaborated as follows.

3.1. Initialization of the clustering centers

As it is mentioned in Section 2, with ACA, the computing quantity is large. To cut down the computing quantity, we may select some representing pixels with limited number, namely, clustering centers, for comparison with other pixels. The number of clustering centers and their features are determined by the following procedures:

**Step 1.** Determine the number and the gray value of the clustering centers.

As we know, since pixels belonging to the same object always have similar gray values, histogram represents the extent that the pixels ‘gather’ at different gray values. In other words, it shows, at certain level, the clustering results based on gray value. Fig. 3 gives the
histogram of Fig. 1. In the histogram, it can be noted that there are several peaks, which are comparatively higher than the average ones. So, from the histogram of Fig. 1, we can roughly select the first \( n \) gray values with the most pixels as the \( n \) clustering centers with the gray values as their gray features. Where, when \( n \) is large, the clustering process will be quick but the clusters’ merging process will take some time. On the contrary, the merging process will be shortened but the time for comparing computation will increase. Here, in this test, \( n \) is 8.

**Step 2.** Determine the gradient feature of the clustering centers.

From step 1, \( n \) clustering centers are roughly selected based on the image histogram, namely, the clustering results with respect to gray values. And according to the process, it can be deduced that the gray values with the most pixels always correspond with backgrounds or objects, while others with fewer pixels are relative to edges and noises. At the same time, it can be noted that the gradient of the background and object pixels, which account for large part of the image, is zero. While the gradient of the edge and noise pixels, which account for much less part of the image, is much larger. Thus, for those clustering centers with far more pixels than others in the histogram, we can roughly set their gradient feature to zero. While for other clustering centers, the gradient feature is roughly set according to formula (7):

\[
gf = \frac{1}{m} \left( \text{Max}_{j=1,...,n} \left( \sum_{i=1}^{m} \text{grad}(i,j) \right) \right),
\]

where \( gf \) is the gradient feature to be got. \( \text{grad}(i,j) \) is the gradient of pixel \((i,j)\). And \( m \times n \) is the size of the image.

**Step 3.** Determine the neighboring feature of the clustering centers.

According to the above, for the clustering centers whose gradient feature is zero, set the neighboring feature to 8. For those with larger gradient feature, if the gray feature is much larger, the neighboring features are 6. Otherwise, if the gray feature is small, the neighboring features are 3.

### 3.2. Improvement of the heuristic function

From formula (3) and the program of ACA in Section 2, clustering probability computation is vital to decide whether an ant belongs to one cluster. Accordingly, \( \eta_{ij} \), the heuristic function and a key component of clustering probability, plays an important role. In ACA, the heuristic function is an inverse proportion to \( d_{ij} \). With the increase of \( \eta_{ij} \), the clustering probability will be increased, i.e., the pixel being judged is more likely to be classified into one class. To accelerate the searching process, we improve the heuristic function by formula (8):

\[
\eta_{ij} = \frac{r}{d_{ij}^s} = \frac{r}{\sqrt{\sum_{k=1}^{m} p_k (x_{ik} - c_{jk})^2}},
\]

where \( r \) is the clustering radius, which is often no less than 1. Then, it can be seen that the larger the \( r \), the larger the \( \eta_{ij} \), and accordingly, the larger will be the clustering probability. When computing \( d_{ij} \), instead of using each pixel in the image like ACA, we adopt \( c_{jk} \), the clustering center, which is initialized according to Section 3.1. Since the number of clustering centers is much less than that of the image pixels, the computing time will be greatly reduced. And according to formula (8), the larger the distance to the clustering center, the smaller the \( \eta_{ij} \); consequently, the smaller will be the clustering probability.

### 3.3. Algorithm of improved ACA

The program of improved ACA is as follows:

**Step 1.** Initialize the parameters of \( \alpha, \beta, r \) and \( \lambda \).

**Step 2.** Initialize clustering centers according to 3.1.

**Step 3.** For each ant, calculate the distance to the clustering centers \( d_{ij} \). If \( d_{ij} = 0 \), the probability is 1. Otherwise, if \( d_{ij} < r \), calculate \( \eta_{ij} \) according to formula (8).

**Step 4.** According to formulae (2) and (3), calculate the clustering probability. If the probability is larger than \( \lambda \), put the ant to that class. Refresh the pheromone concentration following formulae (4) and (5). Refresh the clustering centers by (6). Otherwise, leave the ant into a new set waiting for next cycle.

**Step 5.** Calculate the distance between the clustering centers. If the distance is less than the given threshold \( \epsilon \), merge the classes and refresh the clustering centers.

**Step 6.** If there is any unclassified ant, take the clustering centers as new ants, goto step 3. Otherwise, end.

### 3.4. Discussion of improved ACA

From the above approaches, it can be noted that the computation among all ants of ACA is transformed into the computation between the ants and the clustering centers. In addition, the number of clustering centers can be greatly reduced employing the histogram of the image, which will result in much less computing times for the ants. While the clustering radius, \( r \) in the heuristic function will increase the heuristic value of \( \eta_{ij} \) which will in turn increase the clustering probability so that the clustering process can be accelerated.

### 4. Experimental results

To show the effectiveness of improved ACA, experiments are done. At first, two well-known edge detection
operators such as Sobel and Canny are used. Then, we adopt one more recent but widely used approach, watershed [2], to segment the image. At last, improved ACA is tested and the segmenting result and running time are compared with the former approaches and original ACA, respectively.

Fig. 4 shows the segmentation results of Sobel edge detection. It can be noticed from this figure that the darker parts, or the left halves of the two tigers are lost. And the contours of the two tigers are not consistent. While in Fig. 5, the detecting result by Canny edge detection, the effect is much better compared with that of Sobel, and the darker parts are detected; however, there seems to be too many details. Fig. 6 illustrates the segmenting results of watershed. From this figure, it can be seen that the contours of the two figures are clear and the darker parts are not lost, but the oversegmentation problem can be clearly noted.

Testing result by improved ACA is given in Fig. 7 when \( \alpha = 1, \beta = 1, r = 50, \lambda = 0.9 \) and \( T = 80 \). Comparing ACA with Sobel approach, that is comparing Figs. 2 and 7 with Fig. 4, it can be noted that the darker parts, which cannot be detected by Sobel, is successfully detected by ACA. While when compared with Canny and watershed approaches, which also successfully detect the darker parts, the problem of too many details and oversegmentation are avoided. Therefore, the effectiveness of ACA in image segmentation can be proved.

Then, in terms of efficiency, it can be deduced from Section 3 that by setting the clustering centers and enhance the heuristic functions, the computing quantity is greatly deduced and the convergence process are accelerated. According to the comparisons made between ACA and the improved approach based on Fig. 1 under the same testing environment, the running time of the improved approach is about 26.22 s, which is much less than the 10 min of ACA.

Extension experiments are roughly done to compare the time used by ACA and the improved approach with the increase of image size. And the results are illustrated in Fig. 8. Where, the red curve is the computing time of ACA with the increase of image size. Since computing time of
improved ACA is far less than that of ACA, to make it clearer, we multiply the actual times with $10^5$ and show it in green curve. From this result, it can be seen that either in increasing trend or in increasing quantity, the improved ACA is superior to the original ACA.

5. Conclusions

This paper presents a new approach to image segmentation. Combined with fuzzy clustering, ACA is successfully applied in image segmentation. Good segmentation results are obtained. And the superiority of ACA over Sobel, Canny and Watershed algorithm is proved when dealing with complex images. Particularly, after the improvement of ACA, the computation quantity is greatly reduced. And the running time drops from more than 10 min to 26.22 s. Tests show that improved ACA-based fuzzy clustering is an efficient approach in image segmentation.

References


Yanfang Han is a Ph.D. at Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University. Her research interests include image processing, pattern recognition and machine vision.

Pengfei Shi is a professor at Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University since 1993. His research interests include image processing, pattern recognition, artificial intelligence, biometrical identification, OCR, machine vision and so on.