Edge detection of digital images using a conducted ant colony optimization and intelligent thresholding

Hamid-Reza Reza-Alikhani, Ali-Reza Naghsh, Razieh Jalali-Varnamkhasti

Abstract—An edge detection algorithm based on Ant Colony Optimization (ACO) and Fuzzy Inference System (FIS) and neural network is presented. This algorithm uses a FIS with 4 simple rules to identify the probable edge pixels in 4 main directions, then the ACO is applied for assigning a higher pheromone value for the probable edge pixels rather than other pixels so that the ants movement toward edge pixels get faster. Another factor that needs to be considered in order to conduct the ants’ movement is the influence of the heuristic information in the movement of any ant to be proportional to local change in intensity of each pixel. Finally, by using an intelligent thresholding technique which is provided by training a neural network, the edges from the final pheromone matrix are extracted. Experimental results are provided in order to demonstrate the superior performance of the proposed approach.

Index Terms -Ant Colony Optimization, Edge detection, Intelligent thresholding

I. INTRODUCTION

The main aim of edge detection of a digital image is to find and illustrate the objects in an image. There are many applications in the industries, such as image processing (wildly used in medicine), computer vision and so on. In the last decade many methods have been proposed in the area of edge detection. One of the techniques mostly used in edge detection is in taking advantage of linear-time-invariant (LTI) filters which have little calculation complexity in computation [1]. In the first order filters any abrupt changes in the pixels intensity between two neighboring pixels is called to be an edge and therefore the pixels in the image which have its derivative value (amplitude) of the gray level intensity to be high, is considered to be an edge [1]. In spite of the vast usage of these filters in edge detection, one cannot get good results once we are dealing with the images with non-uniform lighting [2]. Usage of second-order derivative is one of the traditional approaches for edge detection that is rarely used; it is because the second-order derivatives are highly sensitive to noise. The Laplace value produces a doubling edge effect which is not acceptable [1]. However, Laplacian can be a powerful complementary in combination with the other edge detection techniques. In recent years, the temptation and tendency of using intelligent techniques for edge detection among the researchers is increased in order to overcome the drawbacks of traditional methods. The FIS is presented for edge detection by considering the edge continuity. It also works well with noisy images. However, the increase in the noise level will degrade the results, this is because of the fact that as the noise level increases, the noise pixels become connected, and they lose their randomness [3]. Another type of fuzzy edge detector is presented in which all inputs to the FIS are obtained by applying the original image to a linear high-pass filter, a first-order edge detector filter (Sobel horizontal and perpendicular operators), and a low-pass filter. Then the output of the operators is used as the input to the fuzzy inference system [2]. This method according to the simulation of [2], in a non-uniform lighting condition gives an acceptable solution for the images and prevents the double edge effect. An edge detection algorithm employing multi-state adaptive linear neurons has been presented which can reduce the noise effect without increasing the mask size. The inputs are defined using the local mean in a predefined mask of 3×3 and then the one-dimensional edges are defined so that they are linearly separable from the non-edges. This technique needs a vast computational effort [4]. In [5], by using a recurring neural network, a general solution for the edge detection is presented. There are three outputs for each pixel of the input image at network where two of these outputs represent discontinuities in horizontal and vertical directions and the third one separates the edge pixels from non-edge pixels. However, this method has great computational effort and as the image size increases, the number of neurons in neural network increases monotonically. As an example, for an image with 128×128 pixels a neural network with 3×128×128 is needed. ACO is one of the heuristic approaches that has many applications [6]-[11] and is applied to edge detection problem in this paper. The biological origination of ACO refers to depositing of pheromone that ants leave on the ground while moving. Through smelling these chemicals, ants use it for exchanging the information about routs and choose the appropriate path. Each ant that moves on an inoculated path with pheromone strengthens the pheromone on the path with their pheromone. This positive

Hamid-Reza Reza-Alikhani, Electrical Department, Tafresh University (Email: alikhani.hamidi@gmail.com)
Ali-Reza Naghsh, Electrical Department, najaf Abad Azad University (Email: Naghsh.ali@yahoo.com)
Razieh Jalali-Varnamkhasti, Electrical Department, Tafresh University (Email: raziehjalali89@yahoo.com)
feedback mechanism finally leads the ants to follow the best paths after a transient time. Dorigo proposed the first ACO model, Ant System (AS) in 1991 for solving the Travel Salesman Problem (TSP) in a small dimension. AS is divided into three models; Ant Cycle (AC), Ant Quantity (AQ) and Ant Density (AD), according to the update time of the pheromone [12]. Since then, a numerous number of extensions to AS have been developed such as the Ant Colony System (ACS) [13] and Max-Min Ant System (MMAS) [14]. In [15], ACO is used for edge detection problem. This work is performed by employing the special functions for heuristic information and using the thresholding technique named Otsu technique [16]. Experimental results in [15] show that this method does not have the ability of extracting the edges completely. Another usage of the ACO for edge detection is introduced in [17]. This method has employed the composition of AS and ACS models and benefits from performing the Daemon operations. The method that is proposed in [18] exploits the AS model for edge detection in which further adjustment to refine the edge (morphological thinning operation) is done as a post-processing.

In this paper, ACO is used for edge detection problem. For updating pheromone similar to ACS model, it uses both local and global updating, however the state transition rule is introduced by introducing a new innovation at the state transition rule of AS. This new innovation is so that the ants are directed to the edges with higher probability. The rest of our paper is organized as follows: in section II, a brief introduction is done to illustrate the fundamental concepts of ACO and ACS. Then, an image edge detection approach is presented in section III. Simulation results are presented in section IV. Finally, section V covers the conclusion of this paper.

II. ACO AND ACS

One of the improved developments of the AS is the ACS [13]. Updating the pheromone rule in the ACS is the combination of the AD and the AC models which are two models of the AS. To start with, the K numbers of ants are distributed on the graph of problem randomly and each ant updates the pheromone value of visiting the arcs (linking the neighboring nodes) based on the local updating rule. At the end of each iteration, after all ants finished the construction process, some amount of pheromone is updated on special arcs based on global update rule. Fig. 1 shows the structure of the ACS algorithm.

A. The State Transition Rule in ACS

The state transition rule in ACS is such that, the ant in node i chooses an adjacent node j according to the following probability function [13].

\[
    j = \begin{cases} 
    \max_{j \in O_i \{ (\tau_{ij}), (\eta_{ij}) \}^\beta} & \text{if } q \le q_0 \\
    \text{equation in (2)} & \text{otherwise}
    \end{cases}
\]  

(1)

Where \( q \) is a uniform random variable in the interval \([0, 1]\) and \( q_0 \) is a parameter in the same interval.

\[
    p_{ij}^{(n)} = \frac{(\tau_{ij}^{(n-1)})^\alpha (\eta_{ij})^\beta}{\sum_{j \in O_i} (\tau_{ij}^{(n-1)})^\alpha (\eta_{ij})^\beta}
\]  

(2)

In (1) and (2); \( \tau_{ij} \) is the value of pheromone on the connecting line between nodes i and j; \( \eta_{ij} \) is the heuristic information on the connecting line between nodes i and j; \( O_i \) is the neighboring node for an ant which is in node i. The constants \( \alpha \) and \( \beta \) control the influence of the pheromone and heuristic information, respectively. In (2), \( \sum_{j \in O_i} (\tau_{ij}^{(n-1)})^\alpha (\eta_{ij})^\beta \) is the normalization factor.

![Fig.1. the general structure of the ACS](image)

B. Global Update Rule in ACS

At the end of each iteration and after checking the quality of answers that is generated by the movement of ants, global updating of the pheromone is performed according to (3).

\[
    \tau_{ij}^{(n)} = (1 - \rho)\tau_{ij}^{(n-1)} + \rho \Delta \tau_{ij}^{(best)}
\]  

(3)

Where \( \Delta \tau_{ij}^{(best)} \) is equal to:

\[
    \Delta \tau_{ij}^{(best)} = \begin{cases} 
    \frac{1}{L_{kbs}} & \text{if } (i, j) \in \text{best tour of } k^{th} \text{ ant} \\
    0 & \text{otherwise}
    \end{cases}
\]  

(4)

Where in (3) and (4), \( 0 < \rho < 1 \) is the pheromone evaporation rate, \( L_{kbs} \) is the length of the best obtained result. Thus in the ACS, the global pheromone update is performed just on the best solution. The best result can be the best-so-far from the starting point of solving the problem or the best result is the current iteration.

C. Local Update Rule in ACS

According to this rule, all of the ants when solve the problem gradually, update the pheromone value of arcs which they pass through based on (5).

\[
    \tau_{ij}^{(n)} = (1 - \varphi)\tau_{ij}^{(n-1)} + \varphi \tau_{init}
\]  

(5)
Where $\varphi \in (0, 1)$ is the pheromone decay coefficient, $\tau_{init}$ is the initial pheromone value.

### III. ACO-BASED IMAGE EDGE DETECTION

In this paper, we used combinational structure of the FIS, ACO and neural network for the image edge detection.

#### A. Identification of Possible Edge Pixel Using FIS

We use a simple fuzzy system for identification of the edge of the pixels before setting the parameters of the ACO. This is done so that one can distinguish between the edge of the pixels and the rest of pixels in adjusting the parameters in order to have the movement of the ants directed toward the edges in a directional manner. Fuzzy system detects the edges in 4 main directions by utilizing 4 simple rules. For $3 \times 3$ neighborhood shown in Fig. 2, the difference between the intensity of the central pixel $G$ and its neighboring pixels is calculated according to (6) and then, they form the inputs of the fuzzy system.

$x_i = I_i - I_G, \quad i \in \{1, 2, ..., 8\}$  

Where $I_G$ is the intensity value of the central pixel $G$, and $I_i$ is the intensity value of the neighbor pixel $i$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>G</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>

![Fig. 2. illustration of 3x3 neighborhood](image)

Trapezoidal membership functions are used for all of the input variable as shown in Fig. 3. One should notice that the intensity value of the pixels for the input image is in the interval $[0, 255]$, and the input of fuzzy system will be in the interval $[-255, 255]$.

![Fig. 3. Membership function of the variable input $x_i$](image)

Based on input membership functions, four rules are defined for identification of the edges which are listed in table (1).

<table>
<thead>
<tr>
<th>TABLE I FUZZY RULES FOR IDENTIFICATION THE EDGE PIXELS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule(1): $x_1, x_7, x_8 \rightarrow x_3, x_4, x_5 \rightarrow x_2, x_6$, $G$</td>
</tr>
<tr>
<td>Rule(2): $x_1, x_2, x_3 \rightarrow x_5, x_6, x_7 \rightarrow x_4, x_8$, $G$</td>
</tr>
<tr>
<td>Rule(3): $x_1, x_2, x_8 \rightarrow x_4, x_5, x_6 \rightarrow x_3, x_7$, $G$</td>
</tr>
<tr>
<td>Rule(4): $x_6, x_7, x_8 \rightarrow x_2, x_3, x_4 \rightarrow x_1, x_5$, $G$</td>
</tr>
</tbody>
</table>

Rules (1) and (2) are identified on horizontal and vertical edges, respectively and two other rules are to detect the edges in diagonal directions. For example the third law states that if the inputs $x_1, x_2$, and $x_8$ are in low membership function and the $x_4, x_5$, and $x_6$ are in high membership function then the probability that the central pixel and the pixel 3 and 7 in Fig. 2 are along an edge.

#### B. Proposed Method for Image Edge Detection Based on the ACO

Ant’s algorithm can be used for solving the discrete optimization problems in which their space solution is discrete. A two dimensional digital image with the size $M \times N$ can be defined as a discrete space of pixels. This discrete space can be represented by a graph that graph nodes are image pixels and graph connections are done by connecting the adjacent pixels together as in Fig. 4.

![Fig. 4. An $M \times N$ graphical representation of an image.](image)

In order to start the edge detection by ACO, the first stage is setting the initial value of parameters. The important parameters that should be set are the pheromone value assigned to each pixel and the heuristic information of the pixels. In all the previous works, a small number that is close to zero is considered as the initial value of the pheromone for all the pixels but in our paper the initial value of the pheromone that is assigned to the identified edge pixel by fuzzy system is more than of pheromone value of the rest of the pixels. This difference in pheromone value causes to create a pre-understanding for the movement of the ants. This means that in the beginning of task it seems that the ants have already passed through the edge pixel and have increased the amount of pheromone for those pixels. This causes the ants to move toward the edge pixels more. The manner that the initial pheromone assigned to pixels is expressed by (7).

$$
\tau_{ij} = \begin{cases} 
\tau_{\text{max}} & \text{if } (i,j) \text{ is possible edge} \\
\tau_{\text{min}} & \text{if } (i,j) \text{ is nonpossible edge}
\end{cases}
$$

(7)

Another important parameter is heuristic information matrix which is made based on local changes in the pixel’s intensity and its value at pixel $(i, j)$ is determined by (8).

$$
\eta_{ij} = V_c(I_{ij})
$$

(8)

Where, $I_{ij}$ is the intensity value of the pixel at $(i, j)$ in the normalized image, $V_c(I_{ij})$ is a function that operates on the...
neighboring pixels around the pixel \((i, j)\) and is expressed as follow:

\[
V_c(i, j) = f\left(\frac{|l_{i-1,j} - l_{i+1,j} + l_{i,j+1} + l_{i,j-1}|}{|l_{i-1,j} - l_{i+1,j} + l_{i,j+1} + l_{i,j-1}|}\right) \quad (9)
\]

Where \(f(.)\) in (9), is an exponential function and its mathematical form is according to the (10).

\[
f(x) = 0.1e^x \quad (10)
\]

After adjusting the initial value of the parameters, \(K\) numbers of ants become randomly distributed on the graph. On every iteration, each ant moves across the image, from one pixel to another, until it has made \(L\) construction steps (\(L\) is the number of the construction steps). Each ant moves according to the amount of pheromone and heuristic information related to the neighboring pixels of the current position in the format of the state transition law. The state transition rule of our paper is different from the rules of AS that is expressed in (11).

\[
p_{(i_0,j_0)\rightarrow(i,j)}^{(n)} = \frac{\left(\tau_{ij}^{(n-1)}\right)^{\alpha} \left(\eta_{ij}\right)^{\beta}}{\sum_{i,j} \left(\tau_{ij}^{(n-1)}\right)^{\alpha} \left(\eta_{ij}\right)^{\beta}} \quad (11)
\]

Where \((i_0, j_0)\) is the current position of the ant; \((i, j)\) is the next position for the ant; \(\tau_{ij}\) is the pheromone value for pixel \((i, j)\); \(\Omega_{(i_0,j_0)}\) is the neighborhood pixels of pixel \((i_0, j_0)\); \(\eta_{ij}\) is the heuristic information at pixel \((i, j)\); and the constants \(\alpha\) and \(\beta\) control the influence of the pheromone and the heuristic information, respectively.

The difference between the state transition rule of our paper and the AS model is in the manner of controlling the influence of heuristic information in changing the position of the ants. We control the influence of heuristic information at the state transition rule by a matrix in which its elements have been selected according to the heuristic matrix elements. Thus the pixels that their change in local intensity in neighborhood is higher, have a greater share in the state transition rule with respect to the other pixels, so the movement of the ants will be directed towards these pixels. In the presented paper, we use \(\beta_{\text{max}}\) for the pixels that have the higher change in their intensity more than a threshold and we use \(\beta_{\text{min}}\) for the rest of the pixels. This statement is given by (12).

\[
\beta_{ij} = \begin{cases} 
\beta_{\text{max}} & \text{if } \eta_{ij} \geq \text{Threshold} \\
\beta_{\text{min}} & \text{if } \eta_{ij} < \text{Threshold}
\end{cases} \quad (12)
\]

Where, the threshold is selected by the designer.

According to the above comments, the state transition rule in our paper is shown in (13) in which the influence of the heuristic information is controlled by a matrix \(\beta\) instead of constant \(\beta\).

\[
p_{(i_0,j_0)\rightarrow(i,j)}^{(n)} = \frac{\left(\tau_{ij}^{(n-1)}\right)^{\alpha} \left(\eta_{ij}\right)^{\beta}}{\sum_{i,j} \left(\tau_{ij}^{(n-1)}\right)^{\alpha} \left(\eta_{ij}\right)^{\beta}} \quad (13)
\]

The local update rule that is used in the presented paper is similar to the local update rule of ACS algorithm. According to this rule, each ant when moving from one pixel to another, updates the pheromone value of the arcs that they pass through by (14).

\[
\tau_{ij}^{(n)} = (1 - \varphi)\tau_{ij}^{(n-1)} + \varphi \Delta \tau_{ij} \quad (14)
\]

Where \(\varphi \in (0,1]\) is the pheromone decay coefficient; \(\tau_{ij}\) is the pheromone value that is assigned to the each pixel.

Each ant is considered to have a memory in order to avoid local convergence. The address of the pixels that is visited by each ant is saved in its memory to prevent the repetitive movement of ants.

In the each iteration, after all ants finished \(L\) construction steps, the global update is performed on the pheromone. In ACS, the global update is done for the best result whereas in the presented paper the global update is performed for all of the visited pixels. This is due to the fact that some of the details in ACS algorithm are not suitable for the edge detection problem. Originally, the ACO is made to solve the TSP that in the TSP each ant produces a complete solution but in edge detection the aim of the ants are not to produce a complete solution but each of them has a share in edge detection. Therefore, as the nature of the ACO technique applied to the TSP is different from the nature of the ACO edge detection technique described in this paper it is needed that ACO with a little change to be used where one of this change is updating all the visited pixels instead of updating the best results. The global updating process is expressed in (15).

\[
\tau_{ij}^{(n)} = (1 - \rho)\tau_{ij}^{(n-1)} + \rho \Delta \tau_{ij} \quad (15)
\]

Where \(\Delta \tau_{ij}\) is equal to:

\[
\Delta \tau_{ij} = \begin{cases} 
\frac{\sum_{l<j} k l \eta(I(l), j) \text{if}(l, j)\text{visitedbyant}}{K} & \text{if } i = j \\text{otherwise}
\end{cases} \quad (16)
\]

At the end of iterations, we have a pheromone matrix that represents the edge information at each pixel and it can be used in order to extract edges.

C. Intelligent Thresholding

Using thresholding, one can extract the edges from the final pheromone matrix. There are several methods for thresholding such as classical thresholding in which a number is selected as a threshold and then all elements of pheromone matrix are divided into two groups due to the fact that each element can have the value higher or lower than the threshold. A thresholding technique called Otsu technique is introduced [16], and used in [15] and [17]. In thresholding technique Otsu, the initial threshold \(T^{(0)}\) is selected as the mean value of the pheromone matrix. Then the matrix as explained is divided in two groups according to whether or not it is higher or lower.
than the threshold $T^{(0)}$ so that the new threshold is computed as the average of two mean values of each of the above two groups. The above procedure is repeatedly carried out until the successive threshold values $T^{(n)}$ and $T^{(n-1)}$ is almost the same (where the tolerance is defined by the user). In the final pheromone matrix, many of the pixels at background and adjacent pixels of edge pixels have the value of the pheromone near to the value of the pheromone for the edge pixels; therefore by using a classical thresholding those pixels are represented as an edge too. For the above reason in this paper, an intelligent thresholding technique is proposed to utilize the capabilities of neural network. This innovation causes to be able to extract maximum number of the edges. It is necessary to train a neural network to make an intelligent threshold. The neural network training and training patterns are described in the following.

A three layer neural network with 9-20-1 structure is used. The training algorithm that is used is the Back Propagation (BP) type and transfer function used for all neurons is the sigmoid type. The final pheromone matrix that is used as the input for neural network is used to create the target matrix. Using of masks $5 \times 5$ reduces to zero value of the pheromone of the singular or the discontinuous points that have the value of pheromone greater than a threshold defined by the designer. Moreover, human observer performed the thinning operation to create an optimal target matrix. Input range for neural network is in the interval $[0, 255]$ and its outputs have only values of $\{0, 1\}$. Network training operation is performed for the standard image of the woman and its convergence profile is shown in Fig. 5. As seen in Fig. 5, network convergence profile is acceptable, therefore after training, it can be used for intelligent thresholding on the pheromone matrix of different images. Experimental simulating results demonstrate the ability of this thresholding method.

### IV. EXPERIMENTAL SIMULATING RESULTS

We have applied and simulated the procedure on a few standard images in order to show the evaluation of the performance of the techniques by the results on two images of Lena and Peppers in Fig. 6 and Fig. 7, respectively. Various parameters of the proposed method are chosen as follows:

- Iteration number = 7
- Ant number = 512
- Step number = 200
- Threshold = 0.125
- $\rho$ = 0.1
- $\beta_{\min}$ = 0.1
- $\beta_{\max}$ = 2
- $\tau_{\min}$ = 0.00001
- $\tau_{\max}$ = 0.9
- $\alpha$ = 1

These values are chosen in such a way that the best results are obtained.

Methods presented in [15] and [17] and [18] are simulated under similar condition for comparison. Their result with the result of proposed method in our paper is to confirm the performance the method. As for the comparison, the morphological thinning operation of [18] is not taken into account because it is a post-processing to further refine the edge information that is extracted by ACO [18]. As seen in Fig. 6 and Fig. 7, the method in this paper is led to thinner edges compared with the method presented in [18] and also in comparison with the methods presented in [15] and [17], our method has extracted some more edges.
V. CONCLUSION

In this paper a new method for image edge detection is presented by using the intelligent techniques. We used a simple fuzzy system to be able to detect the edges in four directions and this identification is used for directing the movement of ants in the ACO. In ACO, we can conduct the movement of ants toward edges further and we increase the ACO convergence by increasing the influence of edge pixel that is detected by fuzzy system and by increasing the influence of pixels that have greater change in intensity at their local neighborhood. Finally, we could extract the edges of the final pheromone matrix by introducing an intelligent thresholding technique. Experimental simulation results show that the proposed method in our paper has been able to extract more edges with a better quality (subjectively) than the other methods.

VI. REFERENCES