International Journal of Research in Computer and Communication Technology
ISSN (O) 2278-5841
ISSN (P) 2320-5156

Image Edge Detection Using Improved Ant Colony Optimization Algorithm

P.Thukaram1, S.J.Saritha2

1JNTUACEP, INDIA
, thukaram@ymail.com, sarithajntucep@gmail.com

ABSTRACT

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. Image edge detection is the one of the method in the image processing. Edges are significant local changes of intensity in an image. Edges typically occur on the boundary between two different regions in an image. The aim of the edge detection is to extract the important features from edges of images. In this paper, Ant Colony Optimization Algorithm is introduced to tackle the Image edge detection problem. The proposed ACO-based edge detection approach is able to establish a pheromone matrix that represents the edge information presented at each pixel of the image, according to the movements of a number of ants which are dispatched to move on the image. Furthermore, the movements of these ants are riven by the local variation of the images intensity values. Experimental results are provided to demonstrate the superior performance of the proposed approach.

Key words:Ant colony optimization, Edge detection, image processing

1. INTRODUCTION

Image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or, a set of characteristics or parameters related to the image. Most image processing techniques involve treating the image as a two-dimensional signal and applying standard signal processing techniques to it.

Edge is an important feature in an image and carries important information about the objects present in the image. Extraction of edges is known as edge detection. Edge detection aims to localize the boundaries of objects in an image and significantly reduces the amount of data to be processed.

Ant colony optimization (ACO) is a nature-inspired optimization algorithm [1], [2], which is motivated by the natural phenomenon of ants. The ants deposit pheromone on the ground to denote the shortest path that is to be followed by other members in the colony. The ACO algorithm is first referred as Ant System, proposed by [3]. Many algorithms have been developed on ACO [4] like Max-Min ant system [5] and the Ant Colony System [6]. In this paper, ACO is used for image edge detection. The aim of ACO is to extractthe edge information of the image, as it plays a crucial role to comprehend the image's content. The proposed approach exploits the movement of the number of ants on the image which is based on the local variation in the intensity value of the image. This information is used to establish a pheromone matrix, whichgives the edge information of the image.

2. ANT COLONY OPTIMIZATION

ACO is inspired by food foraging behaviour exhibited by ant societies. Ants as individuals are unsophisticated living beings. Thus, in nature, an individual ant is unable to communicate or effectively hunt for food, but as a group, they are intelligent enough to successfully find and collect food for their colony. This collective intelligent behaviour is an inspiration for one of the popular evolutionary techniques (ACO algorithms). The adoption of the strategies of ants adds another dimension to the computational domain. The ants communicate using a chemical substance called pheromone. As an ant travels, it deposits a constant amount of pheromone that other ants can follow. When looking for food, ants tend to follow trails of pheromones whose concentration is higher [9]. There are two main operators in ACO algorithms. These are:

Route construction: Initially, the moving ants construct a route randomly on their way to food. However, the subsequent ants follow a probability-based route construction scheme.

Pheromone update: This step involves two important stages. Firstly, a special chemical „pheromone" is deposited on the path traversed by the individual ants. Secondly, this deposited pheromone is subject to evaporation. The quantity of pheromone updated on an individual path is a cumulative effect of these two stages.

3. Proposed ACO-Based Image Edge Detection Approach.

3.1. Initialization Process

The parameters α and β are initialized. The heuristic information is set. The number of ants is K:$\sqrt{M1.M2}$where M1 is the length, M2 is the width of the image I. All the K ants are propagated on the 2-D image I such that at most one ant is on each pixel. Every pixel in the image is a node and the initial value of the pheromone matrix τ^0 is set to a constant value.

3.2. Construction Process

At each building step, an ant, which is chosen from the K ants, moves L steps on the image I. The ant say A_k moves from node (*l, m*) to its neighbouring node (*i, j*) according to the probabilistic transition matrix defined as,

$$P^n_{(l,m)(i,j)}=\frac{(\tau^{n-1}_{i,j})^{\alpha}.(\eta_{i,j})^{\beta}}{\Sigma_{(i,j)\in\Omega_{(l,m)}}(\tau^{n-1}_{i,j})^{\alpha}.(\eta_{i,j})^{\beta}} \qquad (1)$$

Where,

$\tau^{n-1}_{i,j}$Is the pheromone value of node (*i, j*) and Ω (l,m) is the neighbouring nodes of(*l,m*). In order words, Ω (l,m) is all the pixels that can be in the 8-neighborhood of the pixel (*l,m*). The heuristic information of the node (*i, j*) is$\eta_{i,j}$. In order, to determine the heuristic information [7], the local configuration at each pixel (*i, j*) is defined as,

$$\eta_{i.j} = \frac{1}{Z}V_c(I_{i,j}) \qquad (2)$$

Where, Z is a normalization factor used to isolate error and defined as

$$Z=\Sigma_{i=1:M1}\Sigma_{j=M2}V_c(I_{i,j}) \qquad (3)$$

Where, Ii,j represents the intensity value of the pixel (*i, j*) of image I. The variation of the image's intensity values depends on *c* which a group of pixels which are similar in some form. This group of pixels forms the function$V_c(I_{i,j})$.

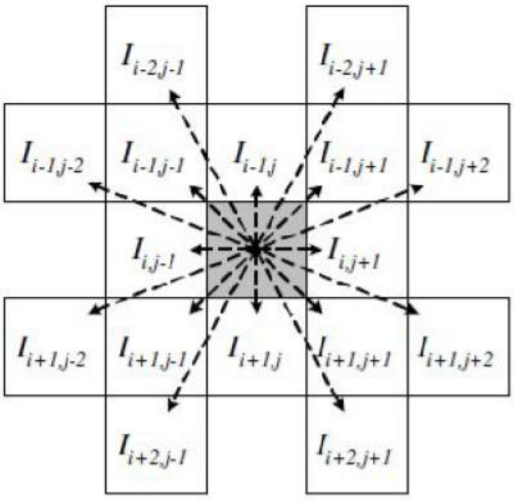


Figure 1: Neighbours of pixel (i,j)

As shown in Figure 1, the function $V_c(I_{i,j})$.depends on its neighbouring group of pixels c which is defined as,

$$V_c(I_{i,j})=f(|I_{i-2,j-1}-I_{i+2,j+1}|+|I_{i-2,j+1}-I_{i+2,j-1}|+|I_{i-1,j-2}-I_{i+1,j+2}|+|I_{i-1,j-1}-I_{i+1,j+1}|+|I_{i-1,j}-I_{i-1,j}|+|I_{i-1,j+1}-I_{i-1,j-1}|+|I_{i-1,j+2}-I_{i-1,j-2}|+|I_{i,j-1}-I_{i,j+1}|)(4)$$

The function ensures that sharp turns in the image are less likely than small angle turns. Thus, each ant in the colony has the tendency to move in the forward direction. In order to change the respective shapes the function Eq. (4) is modified mathematically by the following,

f(x)=λx for x≥0,

(5)

f(x)=$\lambda x^2 for\ x \geq 0$

(6)

$$f(x)=\begin{cases}\sin\left(\frac{\pi x}{2\lambda}\right) & 0 \leq x \leq \lambda \\ 0 & else\end{cases} \qquad (7)$$

$$f(x)=\begin{cases}\frac{\pi x\sin\left(\frac{\pi x}{\lambda}\right)}{\lambda} & 0 \leq x \leq \lambda \\ 0 & else\end{cases}$$

(8)

The parameter λ in each of above functions adjusts the function's respective shapes.

3.3. Update Process

The proposed paper performs both the update steps, one after each ant has moved and the other after all ants on each building step have moved, on opposed to only one update step in many other papers on the similar topic. An attempt to alter the algorithm to one update process yielded a binary image with missing information. The update process, which updates the pheromone matrix after each ant is moved, is

$$\tau^{n-1}_{i,j} = \begin{cases}(1-\rho).\ \tau^{n-1}_{i,j}+\rho.\Delta^k_{i,j}, \\ if\ (i,j) is\ visited\ by\ kth\ ant & (9) \\ \tau^{n-1}_{i,j}, & Otherwise.\end{cases}$$

ρ is the evaporation rate,$\Delta^k_{i,j}$ is determined by the heuristic matrix; i.e $\Delta^k_{i,j} = \eta_{i,j}$

The heuristic information is added into the ant's memory and used for further steps. The second update is made at the end of each building step i.e. all the ants K within the step have moved. Since all the ants have moved at the end of the building step, the equation is

$\tau^n = (1-\Psi).\tau^{n-1} + \Psi\tau^n$(10)

Where Ψ is pheromone decay coefficient

The pheromone matrix is updated at this stage with the consideration of the decay coefficient and the pheromone matrix built until now.

3.4. Decision Process

Being a very important process as it incorporates the results from the previous steps to determine if at each pixel it is an edge or not. In order to find out about the edge information, a threshold value T is used on the pheromone matrixτ^N. The iterative method proposed in [17] is used to compute T. To convert the intensity image to binary image, a normalized intensity value in the range of [0, 1] is considered. Using the starting threshold value, the histogram is segmented into two parts. The mean of the gray values associated with the foreground pixels and the sample mean of the gray values associated with the background of the pixels are computed. This new threshold value is considered as the average of the two samples. This process is recurred based on the new threshold value until here is no change in the value noted. The initial threshold value T^0 is considered as the mean value for the pheromone matrix. Each index value of the pheromone matrix is segregated as below the initial threshold value or above the threshold value. Based on these two categories the average of the mean values is computed which is the new threshold value. As, mentioned earlier this process is repeated till the threshold value becomes constant. The process is as follows

Step 1: Initialize $T^{(0)}$ as

$$T^{(0)} = \frac{\sum_{i=1:M_1}\sum_{j=1:M_2}\tau_{i,j}^{(N)}}{M_1 M_2}$$

(11)

Step 2: Separate the pheromone matrix $\tau^{(N)}$ into two separate class using $T^{(l)}$, where the first class consists entries of $\boldsymbol{\tau}$ that have smaller values than $T^{(l)}$, while the second class consists the rest entries of $\boldsymbol{\tau}$. Next, calculate the mean of each of the above two categories via

$$m_L^{(l)} = \frac{\sum_{i=1:M1}\sum_{j=1:M2} g_{T^{(l)}}^{L}(\tau_{i,j}^{(N)})}{\sum_{i=1:M1}\sum_{j=1:M2} h_{T^{(l)}}^{L}(\tau_{i,j}^{(N)})} \quad (12)$$

$$m_U^{(l)} = \frac{\sum_{i=1:M1}\sum_{j=1:M2} g_{T^{(l)}}^{U}(\tau_{i,j}^{(N)})}{\sum_{i=1:M1}\sum_{j=1:M2} h_{T^{(l)}}^{U}(\tau_{i,j}^{(N)})} \quad (13)$$

Where

$$g_{T^{(l)}}^{L}(x) = \begin{cases} x, & if\ x \le T^{(l)} \\ 0 & otherwise \end{cases}$$

(14)

$$h_{T^{(l)}}^{L}(x) = \begin{cases} x, & if\ x \le T^{(l)} \\ 0 & otherwise \end{cases}$$

(15)

$$g_{T^{(l)}}^{U}(x) = \begin{cases} x, & if\ x \le T^{(l)} \\ 0 & otherwise \end{cases} \quad (16)$$

$$h_{T^{(l)}}^{U}(x) = \begin{cases} x, & if\ x \le T^{(l)} \\ 0 & otherwise \end{cases} \quad (17)$$

Step 3: Set the iteration index l==l+1, and updates the threshold as

$$T^{(l)} = \frac{m_L^{(l)} + m_U^{(l)}}{2}$$

(18)

Step 4: Check if the new threshold $T^{(l)}$value is equal to $T^{(n-1)}$. If false, then go to step 2, otherwise the process is terminated and the threshold value is recorded. To determine if an edge Ei,j is found at the pixel (*i, j*),

$$E_{i,j} = \begin{cases} 1\ if\ \tau_{ij}^{(N)} \ge T^{(l)} \\ 0 \quad otherwise \end{cases}$$

(19)

Where, Ei,j is an edge at pixel (*i,j*)

.

4. EXPERIMENTAL RESULTS:

Various parameters used in this paper are

- K=$\sqrt{M1M2}$: The total number of ants is the size of the image.
- τ_{init}=0.0001: the initial value of each component of the pheromone matrix.
- α=1:the weighting factor of the pheromone information
- β=0.1: the weighting factor of the heuristic information.
- *Ω*=8- connectivity neighbourhood
- *λ*=1 the adjusting factor
- ***ρ***=0.1: the evaporation rate
- L=40: ant's movement steps
- **Ψ**=0.05: the pheromone decay coefficient
- N=4:total number of construction steps

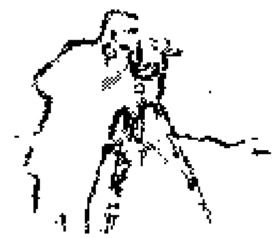
The results obtained for this parameter values are

(A)

(C)

(D)

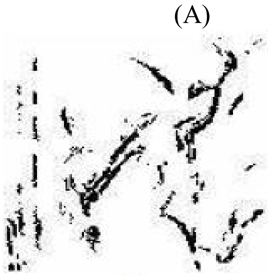
(D)



(E)

Figure 2: Various extracted edge information of the test image camera: (A) the input image with 512x512 pixels resolution. (B)The proposed ACO based image edge detection algorithm with the function defined in Eq: 5 (C) The proposed ACO based image edge detection algorithm with the function defined in Eq: 6(D) The proposed ACO based image edge detection algorithm with the function defined in Eq: 7 (E) The proposed ACO based image edge detection algorithm with the function defined in Eq:8

(A)



(B) (C)

(D) (E)

Figure 3: Various extracted edge information of the test image Leena: (A) The input image with 512x512 pixels resolution. (B)The proposed ACO based image edge detection algorithm with the function defined in Eq: 5 (C) The proposed ACO based image edge detection algorithm with the function defined in Eq: 6(D) The proposed ACO based image edge detection algorithm with the function defined in Eq: 7 (E) The proposed ACO based image edge detection algorithm with the function defined in Eq: 8

5. CONCLUSION

This paper introduces and claims that the results are true for solving optimization problems as well as a renowned problem in the field of image processing being image edge detection. The implementation of paper presents image edge detection using ant colony optimization algorithm. Through this paper an ACO based image edge detection approach has been successfully deployed. The proposed approach yields superior subjective performance to that of the existing edge detection algorithm [8].

6. REFRENCES

1. D. Martens, M. D. Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Aesens, Classification with ant colony optimization, *IEEE Trans. on Evolutionary Computation*, Oct. 2007, vol. 11, pp. 651–665.
2. R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, Data mining with an ant colony optimization algorithm, *IEEE Trans. on Evolutionary Computation*, Aug. 2002, vol. 6, pp. 321–332.
3. S. Ouadfel and M. Batouche, Ant colony system with local search for Markov random field image segmentation, *Proc. IEEE Int. Conf. on Image Processing, Barcelona, Spain*, Sep. 2003, pp. 133–136.
4. S. L. Hegarat-Mascle, A. Kallel, and X. Descombes, Ant colony optimization for image regularization based on a nonstationary Markov modeling, *IEEE Trans. on Image Processing*, Mar. 2007, vol. 16, pp. 865–878.
5. Y. T. Kim, Contrast Enhancement Using Brightness Preserving Bi-Histogram Equalization, *IEEE Trans., Consumer Electronics*, 1997, vol. 43, no. 1, pp. 1-8.
6. H. Ibrahim, and N. S. P. Kong, Brightness Preserving Dynamic Histogram Equalization for Image Contrast Enhancement, *IEEE Trans., Consumer Electronics*, Nov. 2007, vol. 53, no. 4, pp. 1752–1758.
7. M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. on Systems, Man and Cybernetics*, 26:29-41, 1996.
8. H. Nezambadi Pour, S. Saryazdi and E. Rashedi, “Edge detection using ant algorithms,” Soft computing, vol 10, pp.623-628, May 2006.
9. J. Lewis and J. Lawson. Starcat: An architecture for autonomous adaptive behaviour. *Proceedings of the 2004 Hawaii International Conference on Computer Science*, Honolulu, HI, 2004. IEEE.
10. H. Nezamabadi-Pour, S. Saryazdi, and E. Rashedi. Edge detection using ant algorithms. *Soft Computing*, 10:623-628, 2006.

11. A. Colorni, M. Dorigo, V. Maniezzo. Distributed optimization by ant colonies. *Proceedings of the First European Conference on Artificial Life*, Paris, France, 1991. Elsevier Publishing.
12. J. Tian, W. Yu, and S. Xie. An ant colony optimization algorithm for image edge detection. *IEEE Congress on Evolutionary Computation*, 2008:751-756, 2008.
13. Azi Sharif. Antcat. Master's thesis, San Diego State University, San Diego, CA, 2004.
14. F. Glover and M. Laguna. *Tabu search*. Kluwer academic publishers, Boston, MA, 1998.
15. M. Dorigo. Optimization, learning and natural algorithms. Doctoral dissertation, Dip. Elettronica e Informazione, Politecnico di Milano, Italy, 1992.
16. M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo and F. Glover, editors, *new ideas in optimization*, pages 11-32. McGraw- Hill, London, UK, 1999.
17. T. W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Trans. System, Man and Cybernetics*, 8:630-632, 1978.
18. H. Holland. What is a Learning Classifier System, 2000.code.ulb.ac.be/dbfiles/HolBooCol-etal2000lcs.pdf, accessed Mar. 2011.
19. M. V. Butz. Learning classifier systems. *Proceedings of the GECCO conference Companion on Genetic and Evolutionary Computation*, London, United Kingdom, 2007. Springer.