ISSN 2047-3338

Image Edge Detection based on ACO Technique

Mahendra Pratap Panigrahy[1] and Shakil Ahmed[2]
[1]ECE, Institute of Technology Roorkee (ITR), India
[2]CSE, Institute of Technology Roorkee (ITR), India
[1]contact_mahen@rediffmail.com, [2]shakil_36@yahoo.co.in

***Abstract*– Edges are the characteristic boundaries of any digital image and are therefore plays a fundamental importance in image processing task. Digital noisy Image Edge detection significantly reduces the amount of data and filters out the useless information, while preserving the important structural properties in an image. It is an important tool used in many applications: such as image processing, computer vision and pattern recognition. There exist several ant colony optimization algorithms, whereas the original algorithm, known as Ant System (AS), originates from the early nineties. Since then, a number of other algorithms, among the more successful variants MAX-MIN Ant System (MMAS) and Ant Colony System (ACS), were introduced. Several ACO-based approaches have been proposed to the edge detection problems. In this paper, a new ACO-based approach is applied to image edge detection. The approach makes use of improvements introduced in ACS, with the addition of a new direction control feature. The reason for introducing direction control is to make the ants better suited for detecting edges belonging to long and narrow openings, hence, edges belonging to cracks.**

***Index Terms*– Image Processing, Noisy, Edge Detection, Filters, AS, MMAS and ACS**

1. INTRODUCTION

EDGES are the characteristic boundaries of any digital image and are therefore plays a fundamental importance in image processing task. Image edge detection deals with extracting edges in an image by identifying pixels where the intensity variation is high. There are many well-known edge detection algorithms. The most privileged edge detection techniques used are first-order derivative operators such as the Sobel edge operator, the Prewitt edge operator and the Robert cross edge operator. The Laplacian operator is a second-order derivative operator for functions of two-dimension operators and is used to detect edges at the locations of the zero crossing. Hilbert Transform for Edge Detection is another method using convolution. Ants communicate with each other using pheromones. They leave pheromone trails on the ground in order to mark a path between their colony and a food source for other members in the colony to follow. The more ants following the same path, the higher its pheromone concentration becomes. Over time pheromone trails evaporate. The longer it takes for an ant to walk back and forth, the more time the pheromone has to evaporate. Hence, pheromone density remains higher at shorter and more favorable paths where pheromone is deposited at a much higher rate. This behavior helps ants successfully establish, and follow, the better paths. ACO is inspired by this foraging behavior. There exist several ant colony optimization algorithms, whereas the original algorithm, known as Ant System (AS). Since then, a number of other algorithms such as MAX-MIN Ant System (MMAS) and Ant Colony System (ACS) were introduced. Several ACO-based approaches have been proposed to the edge detection problem. In this project, a new ACO-based approach is applied to image edge detection.

A. Fist-Order Derivative Edge Detection

The Sobel Operator, Robert's cross operator, Prewitt's operator are methods of the Fist-Order Derivative Edge Detection. For this Fist-Order Derivative Edge Detection:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}. \qquad (1.1.1)$$

An important quantity in edge detection is the magnitude of this vector, denoted Δf, where

$$\nabla f = \left|\nabla \mathbf{f}\right| = \sqrt{G_x^2 + G_y^2}\,. \qquad (1.1.2)$$

Another important quantity is the direction of the gradient vector. That is:

$$\text{angle of } \nabla \mathbf{f} = \tan^{-1}\left(\frac{G_y}{G_x}\right) \qquad (1.1.3)$$

Computation of the gradient of an image is based on obtaining the partial derivatives of $\partial f/\partial x$ and $\partial f/\partial y$ at every pixel location.

B. Second-Order Derivative Edge Detection

The Laplacian of a 2-D function f (x, y) is a second-order derivative defined as:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (1.2.1)$$

The Laplacian is usually combined with smoothing as a precursor to finding edges via zero-crossings. The 2-D Gaussian functions:

$$h(x,y) = -e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.2.2)$$

Where σ is the standard deviation, blurs the image with the degree of blurring being determined by the value of σ. The Laplacian of h is

$$\nabla^2 h(x,y) = -\left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}} \quad (1.2.3)$$

C. Hilbert Transform for Edge Detection

There is another method for edge detection that uses the Hilbert transform (HLT). The HLT is:

$$g_H(\tau) = h(x) * g(x), \quad \text{where } h(x) = \frac{1}{\pi x} \quad (1.3.1)$$

And * means convolution. Alternatively,

$$G_H(f) = H(f)G(f) \quad (1.3.2)$$

Where G(f) = FT [g(x)] (FT means the Fourier transform), GH(f) = FT [gH (x)], and

$$H(f) = -j\,\mathrm{sgn}(f), \quad (1.3.3)$$

Where the sign function is defined as

$$\begin{aligned} &\mathrm{sgn}(f) = 1 \text{ when } f > 0, \\ &\mathrm{sgn}(f) = -1 \text{ when } f < 0, \\ &\mathrm{sgn}(0) = 0 \end{aligned} \quad (1.3.4)$$

D. Some Disadvantages of Classical (1st Order & 2nd Order Derivative) Edge Detectors

Simple edge detectors are affected by noise – filters can be used to reduce noise. Edge is several pixels wide for Sobel operator– edge is not localized properly. Roberts's operator is very sensitive to noise. These do not differ (except Laplace) between upward (from "dark" to "bright") and downward (from "bright" to "dark") brightness jumps. Such is resulted in "bold" ("thick") edges where the same brightness jump can be detected twice. All of them (except Prewitt) give a preference to the pixel of interest (the symmetry center). This may help to suppress noise, to avoid its detection, but simultaneously some edges can be emphasized, while some other can be suppressed.

II. AN ANT COLONY SYSTEM

The ant colony system has been applied in optimization, which is the Ant Colony Optimization (ACO) algorithm. The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm is a member of ant colony algorithms family, in swarm intelligence methods, and it constitutes some meta-heuristic optimizations. In ACO, the solution to a problem corresponds to a state transfer sequence, i.e. a path, from the starting state to the goal state in the discrete state space.

The optimal solution corresponds to the shortest path. The ants move randomly between neighboring states from the starting state until the goal state is reached. The state-transfer probability is calculated according to the trial intensity. On the other hand, each ant also increases the trial intensity on the way it has passed according to the quality of the solution found. This is a kind of positive feedback mechanism, which leads to fast solution searching by ACO. The probability defined to move from state s_i to s_j given by:

$$p_{i,j}^{(n)} = \frac{\left(\tau_{i,j}^{(n-1)}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}}{\sum_{j\in\Omega_i}\left(\tau_{i,j}^{(n-1)}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}}, \quad \text{if } j \in \Omega_i \quad (2.1)$$

where τ_{ij} (t) is the trial intensity between s_i and s_j at time t. Further, α and β are two parameters having positive values η_{ij} is the reciprocal of the distance between s_i and s_j, which is the heuristic information. A is the set of neighboring states that have not been experienced by the current ant.

There is a crucial issue in the construction process. The first issue is the determination of the heuristic information $\eta_{(i,j)}$. In this paper, it is proposed to be determined by the local statistics at the pixel position (i, j) as

$$\eta_{i,j} = {}_{1/Z}(V_c(I_{i,j})) \quad (2.2)$$

$I_{(i,j)}$ is the intensity value of the pixel at the position (i, j) of the image I, the function $V_c(I(I_{i,j}))$ is a function of a local group of pixels c (called the clique. More specifically, for the pixel $I_{i,j}$ under consideration, the function $Vc(I_{i,j})$ is:

$$V_c(I_{i,j}) = f(|I_{i-2,j-1} - I_{i+2,j+1}| + |I_{i-2,j+1} - I_{i+2,j-1}| + |I_{i-1,j-2} - I_{i+1,j+2}| + |I_{i-1,j-1} - I_{i+1,j+1}| + |I_{i-1,j} - I_{i+1,j}| + |I_{i-1,j+1} - I_{i-1,j-1}| + |I_{i-1,j+2} - I_{i-1,j-2}| + |I_{i,j-1} - I_{i,j+1}|) \quad (2.3)$$

To determine the function f (*) in (3), the following four functions are considered in this paper:

$$\left.\begin{array}{ll} f(x) = \lambda x, & \text{for } x \geq 0 \\ f(x) = \lambda x^2 & \text{for } x \geq 0 \\ f(x) = \begin{cases} \sin(\pi x / 2\lambda) & \text{for } 0 \leq x \leq \lambda \\ 0 & \text{else} \end{cases} & \end{array}\right\}$$

$$f(x) = \left.\begin{cases} \pi x \sin((\pi x/\lambda)/\lambda) & \text{for } 0 \leq x \leq \lambda \\ 0 & \text{else} \end{cases}\right\} \qquad (2.4)$$

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

A. Pheromone update

Once all the ants have terminated their local searches, a pheromone update phase is started in which pheromone trails are modified. The proposed approach performs two updates operations for updating the pheromone matrix.

• The first update is performed after the movement of each ant within each construction-step. Each component of the pheromone matrix is updated according to:

$$\tau_{i,j}^{(n-1)} = \begin{cases} (1-\rho)\ .\ \tau_{i,j}^{(n-1)} + \rho.\,\Delta_{i,j}^{(k)} \text{ if } (i,j) \\ \quad \text{is visited by the current k-th ant;} \\ \tau_{i,j}^{(n-1)} \quad \text{otherwise} \end{cases} \qquad (2.1.1)$$

where ρ is the evaporation rate. $\Delta_{i,j(k)}$ is determined by the heuristic matrix; that is, $\Delta_{i,j(k)} = \eta_{i,j}$.

• The second update is carried out after the movement of all ants within each construction-step according to:

$$\tau^{(n)} = (1-\psi).\ \tau^{(n-1)} + \psi.\tau^{(0)} \qquad (2.2.2)$$

III. ACO IN IMAGE EDGE DETECTION

Image edge detection deals with extracting edges in an image by identifying pixels where the intensity variation is high. There are many well-known edge detection algorithms. Prewitt, Sobel and Canny, to mention a few. Although originating from the early days of computer vision, some are still considered state-of-the-art edge detectors. ACO is a probabilistic technique for finding optimal paths in fully connected graphs through a guided search, by making use of the pheromone information. This technique can be used to solve any computational problem that can be reduced to finding good paths on a weighted graph. In an ACO algorithm, ants move through a search space, the graph, which consists of nodes and edges. The movement of the ants is probabilistically dictated by the transition probabilities. The transition probability reflects the likelihood that an ant will move from a given node to another. This value is influenced by the heuristic information and the pheromone information. The heuristic information is solely dependent on the instance of the problem. Pheromone values are used and updated during the search. Fig. 1 shows a pseudocode of the general procedure in an ACO metaheuristic.

```
Initialize
  SCHEDULE_ACTIVITIES
  ConstructAntSolutions
  DoDaemonActions optional_
```

Fig. 1: ACO metaheuristic

A. Proposed ACO-based Algorithm for image edge detection

The proposed ACO-based approach to image edge detection can be thought of as a three-phased process as described in the Algorithm. The first phase is an initialization phase. The second, which is the construction phase, covers all ant movement. During this phase the pheromone information is continuously updated as the ants «walk on» the image. In the last phase, the decision phase, the solution is made from the values of the elements in the final pheromone matrix.

Algorithm:

1. Initialization Phase

Initialize a gray scale intensity value matrix, a pheromone matrix, a normalized intensity variation matrix (the heuristics) and a set of eight polar angles.

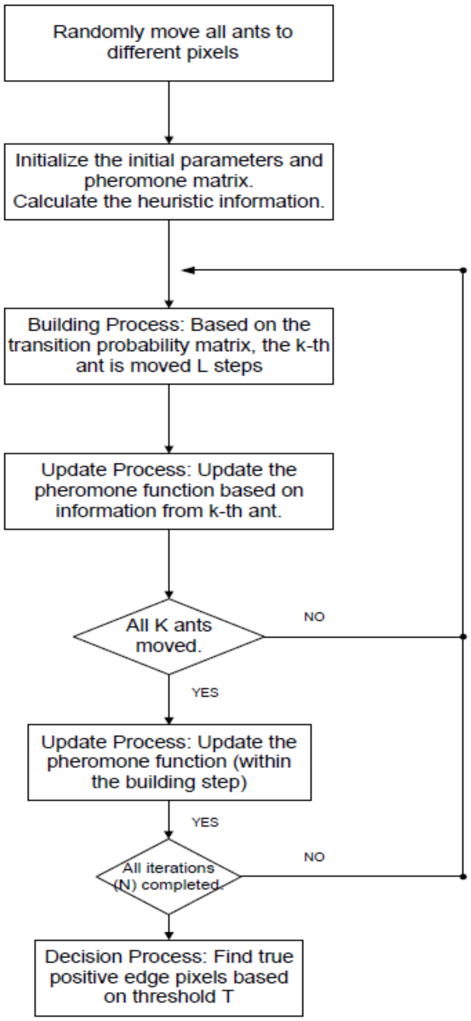
2. Construction Phase

```
for construction step (round) n = 1: N
    Randomly position all ants.
    for movement step l = 1: L
        for ant k = 1: K
          Select, and move ant to, next pixel.
        Immediate local update of the --
        pixel's pheromone (pheromone--
        decay).
        end
     end
     Offline update of all visited pixels
     pheromone (pheromone evaporation).
end
```

3. Decision Phase

The solution is made based on the values in the final pheromone matrix.

B. Flow Chart of the Proposed Algorithm



IV. SIMULATION RESULTS AND DISCUSSION

Experiments were conducted using several test images. The proposed ACO-based edge detection method was implemented using MATLAB. The critical issues in the above are to fix standards value for τ_{in}, α, η, β, ρ etc. For unvaried parameters, the values used were α = 1.0 (ACS constraint), β=1.0, ρ=0.1, η = 0.05, q_0=0.75, τ_{in}= 0.0000001.

Fig. 2: Original Image

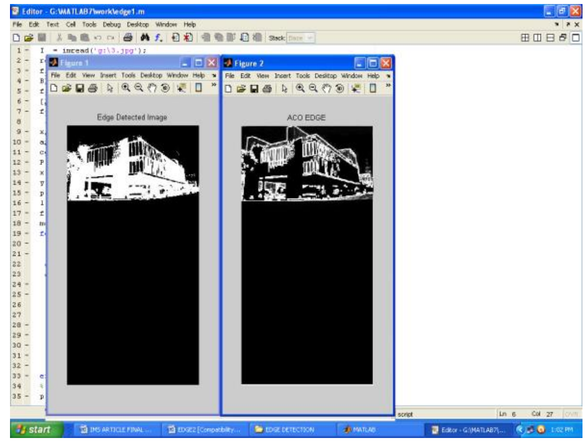


Fig. 3: ACO Edge Simulation

V. COMPARISON WITH THE EXISTING IMAGE DETECTION TECHNIQUES

The comparison with the existing system of image edge detection techniques based on 1st order, 2nd order with this ACO based system, gives better performance result which is shown in Fig. 4.

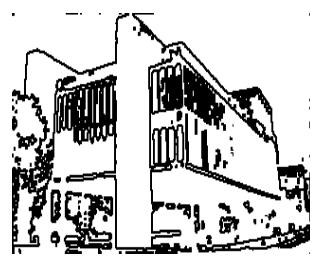


Fig. 4 (a): Edge by using Sobel Edge Detector

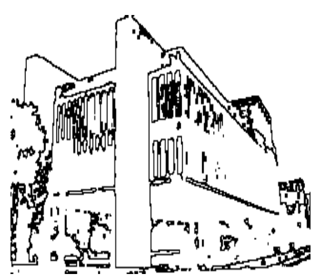


Fig. 4 (b): Edge by using Prewitt Edge Detector

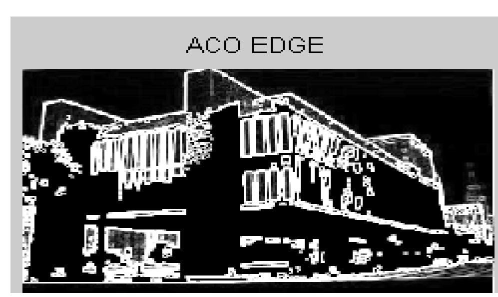


Fig. 4 (c): Edge by using ACO Edge Detector

VI. CONCLUSION

An ACO-based image edge detection algorithm that takes advantage of the improvements introduced in ACS has been successfully developed and tested. Experimental results show that there is more feasibility in identifying edges in an image. With suitable parameter values, the algorithm was able to successfully identify edges. It is noted that the appropriate parameter values depend on the nature of the image, and thus, may vary per application. In recent studies, techniques that could enhance the performance of ACS have been explored. In ants are assigned different pheromone sensitivity levels, which make some ants more sensitive to pheromone than the others. In multiple ant colonies with new communication strategies were employed. The proposed ACS method for edge detection could be extended and possibly be improved by making use of such techniques.

REFERENCES

[1] N. Senthilkumaran and R. Rajesh, “Edge Detection Techniques for Image Segmentation - A Survey”, Proceedings of the International Conference on Managing Next Generation Software Applications (MNGSA-08), 2008, pp.749-760.

[2] M. Dorigo, Ant colony optimization web page, http://iridia.ulb.ac.be/mdorigo/ ACO/ACO.html

[3] M. Dorigo and G. Di Caro, “The Ant Colony Optimization meta-heuristic”, in New Ideas in Optimization, D. Corne et al., Eds., McGraw Hill, London, UK, pp. 11-32, 1999.

[4] Javad Rahebi and Hamid Reza Tajik “Biomedical Image Edge Detection using an Ant Colony Optimization Based on Artificial Neural Networks” in IJEST Vol. 3 No.12 December 2011.

[5] Mantas Paulinas and Andrius Usinskas, “A Survey of Genetic Algorithms Applications for Image Enhancement and Segmentation”, Information Technology and Control, Vol.36, No.3, 2007, pp.278-284.

[6] Lihong Zheng and Xiangjian He, “Edge Detection Based on Modified BP Algorithm of ANN”, published in web server of Taiyuan University of Technology, China.

[7] Song Wang, Feng Ge, Tiecheng Liu, "Evaluating Edge Detection Through Boundary Detection", Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, USA, 2006.

[8] Nor Ashidi Mat Isa, " Automated Edge Detection Technique for Pap Smear Images Using Moving K-Means Clustering and Modified Seed Based Region Growing Algorithm", International Journal of The Computer, the Internet and Management Vol. 13.No.3 (September-December, 2005) pp 45-59.

[9] Green B. Canny Edge Detection Tutorial. Drexel University: PRISM, http://www.pages.drexel.edu/inmeg22/can_tuthtml, 2003.

[10] Gonzalez R. and Woods R. Digital Image Processing (Second Edition). Prentice-Hall Inc. 2002, pp. 567 – 612.

[11] Rubino M. T. Edge Detection Algorithms. http://homopages.inf.ed.ac.uk/rbf/HIPR2/featops.htm, 2005.

[12] Shahana B., Prasad B., E.D.Nagendra Prasad T. (2011). A New Approach for Edge Detection Using First Order Techniques. IJCST, Volume 2, Number 1, pp. 78-82.

[13] Piriya Kumar D, Levi P. “A new approach to exploiting parallelism in ant colony algorithm” In: Stone P, ed. Proc. of the Int'l Symp. On Micromechatronics and Human Science. 2002. 237-243.

[14] Middendorf M, Reischle F, Schmeck F. Information exchange in multi colony ant algorithms. In: Rolim JDP, ed. Proc. of the Int'l Parallel and Distributed Processing Symp. Springer-Verlag, 2000, pp-645-652.

[15] I. Fourmousis, U. Burgin, M. Tonetti, and N.P. Lang, "Digital image processing I Evaluation of gray level correction methods in vitro", Clin. Oral Impl. Res., pp.1-11, Sept. 1993.

[16] J. Matthews. “An introduction to edge detection: The sobel edge detector,” Available at http:// www.generation5.org/ content/ 2002/ im01.asp, 2002.