

Ant Colony Optimization based Edge Detection in Digital Images

Atul Srivastava

*Amity School of Engineering and Technology
Amity University
Lucknow, India
asrivastava5@lko.amity.edu*

Garima Verma

*Department of computer science and Engineering
Pranveer Singh Institute of Technology
Kanpur, India
garimaverma311016@gmail.com*

Rajeev Singh

*Department of computer science and Engineering
ABES Engineering College
Gaziabad, India
rajeev.singh@abes.ac.in*

Shilpa Juneja

*Department of Computer Science and Engineering
Nirankari Baba Gurbachan Singh Memorial College
Sohna, India
shilpjuneja11@gmail.com*

Abstract— Edge detection is identifying the discontinuities in the intensity of the pixel and grouping the contour of edges. Edge detection in digital images is a significant phenomenon in computer vision. In the initial stages of development of edge detection techniques, typical methods like Sobel Canny came into existence. These methods are fast but sensitive towards noise. Soft computing techniques like ant colony optimization (ACO), Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and Fuzzy Logic System (FLS) have extensive application in image edge detection due to their adaptiveness. In this paper, prominent ACO based techniques for edge detection are discussed and compared. The quality of edges identified by ACO based edge detection technique depends on the choice of constants, pheromone evaporation rate, number of iterations etc.

Keywords— Edge Detection, ACO, Soft Computing, Computer Vision.

I. INTRODUCTION

“Edge in a gray level image is the boundary between two regions of different gray levels [1]”. Edge detection is identification of boundaries in image by using texture and intensity [2]. The structure of the image depends on several parameters such as hardware specifications of sensing device, lighting conditions and noise [3]. Edge detection is generally done in three phases – (1) Smoothening and reduction of noise, (2) Image differentiation (magnitude and orientation computation of edge) and (3) Labelling of pixels.

Smoothening techniques are used to diminish the noise on the edge and prepare the image for numerical computations. Smoothening of the image has positive as well as negative effect on the image as the repercussion of noise reduction can be loss of information. Poggio and Torre [4, 5] proposed significant techniques for image smoothening.

Image differentiation is finding first or second derivative of the image [6]. Image is basically interpreted by an analogous function $G(x, y): \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ which denotes the intensity of the image at position (x, y) . The first order derivative is calculated as equation (1).

$$\begin{aligned} \frac{\partial G}{\partial c} &= \frac{\partial G}{\partial x} \frac{\partial x}{\partial c} + \frac{\partial G}{\partial y} \frac{\partial y}{\partial c} \\ &= G_x \cos(\gamma) + G_y \sin(\gamma) \end{aligned} \quad (1)$$

Where, c is the parameter in the direction of derivative \vec{c} and γ is the angle between \vec{c} and x-axis. Gradient of $G (\nabla G)$ is a vector having same direction for which,

$$\frac{\partial}{\partial \gamma} \left(\frac{\partial G}{\partial c} \right) = 0 \quad (2)$$

Magnitude and direction of the gradient is calculated equation 3 and 4 [Intro-].

$$\gamma_{\nabla} = \arctan \left(\frac{G_y}{G_x} \right) \quad (3)$$

$$|\nabla G| = \sqrt{G_x^2 + G_y^2} \quad (4)$$

Edge pixels are supposedly present where the modulus of gradient is highest along with in orthogonal direction to contour of these pixels. The algorithms which use second order derivatives use Laplacian of $G (|\nabla^2 G|)$ [6]. Laplacian operator has certain advantages over second derivative such as less computation, linearity and non-directional.

In digital images due to discrete quantification of pixels, discrete approximation of differentiation operators is required and amplification of noise due to application of operator is inevitable. Digital images are obtained from analogous images by using quantization. The digital image G is defined as $P \times Q \rightarrow X$, where $P=\{0,1,2,\dots,c\}$, $Q=\{0,1,2,\dots,r\}$ and $X=\{0,1,2,\dots,p\}$. c , r and p represent number of columns of image, number of rows of image and highest intensity of any pixel respectively. First order derivatives are represented as equation 5.

$$\begin{aligned} G_p(i, j) &\cong I_p(i, j) = I(i, j) - I(i+1, j) \\ G_q(i, j) &\cong I_q(i, j) = I(i, j) - I(i, j+1) \end{aligned} \quad (5)$$

These operators are commonly represented as masks (equation 6).

$$\begin{aligned} I_p &= M_x \begin{bmatrix} I(i, j) \\ I(i+1, j) \end{bmatrix}, \\ I_q &= [I(i, j) \quad I(i, j+1)] M_y \\ M_p &= [1 \quad -1], \quad M_q = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \end{aligned} \quad (6)$$

The above mentioned masks are not symmetric therefore odd number of elements are used in masks as equation 7.

$$M_p = [1 \ 0 \ -1], \quad M_q = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad (7)$$

Other significant approximations used are Roberts, Prewitt, Sobel and Frei-cheng [8], [9][1] as shown in equation 8, 9, 10 and 11 respectively.

Roberts:

$$M_1 = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} \quad M_2 = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad (8)$$

Prewitts:

$$M_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad M_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (9)$$

Sobel:

$$M_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad M_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (10)$$

Frei-Chen:

$$M_x = \begin{bmatrix} -1 & 0 & +1 \\ -\sqrt{2} & 0 & +\sqrt{2} \\ -1 & 0 & +1 \end{bmatrix} \quad M_y = \begin{bmatrix} +1 & +\sqrt{2} & +1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad (11)$$

Edge labelling is localization of edges. Localization using thresholding of gradient magnitude results in thick images. Another technique NMS (Non-Maximum Suppression) proposed by Canny [10] is commonly used. It finds local maximum in the direction of gradient.

The first order derivative based edge detection techniques use above mentioned derivatives – Roberts, Prewitt, Sobel etc. [7]. These algorithms are vastly sensitive to noise despite being simple and fast [11]. Identified edges are thick as well.

II. ANT COLONY SYSTEM (ACS)

Dorigo and Caro [12] presented an optimization technique constructed on the basis of behaviour of ants to find most suitable path. The technique is called Ant Colony System [ACS]. ACS is most suitable for optimization environments hence the name Ant Colony Optimization (ACO). ACO perceives the problem environment as collection of states in which one state is the initial state or starting state and one state is goal state. Rest of the states are intermediate states which may or may not fall on one or more of several possible paths. The path from initial state to target state is the solution. As there can be multiple such paths, the shortest path is the best solution. ACO tries to find out the sequence of state transitions from initial state to the goal state in a discrete state space. The ants start moving on adjacent neighbouring randomly until they find the goal state. When an ant transfers from one state to another state the probability of state transfer is computed as per the trial intensity (pheromone). The ants keep increasing the trial intensity as per the quality of solution they encounter on their path. The probability of moving from state s_i to s_j is calculated using equation 12.

$$P_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{s_j \in \text{Allowed}} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta} & \text{if } s_j \in A \\ 0 & \text{Otherwise} \end{cases} \quad (12)$$

Where, $\tau_{ij}(t)$ is trial intensity, at time t, between s_i and s_j . α and β are non-negative constants. η_{ij} is used as heuristics and contains inverse of distance between s_i and s_j . A contains adjacent states yet to be visited by the current ant. Tabu list records the state transition order. When all the ants finish traversing the states, the trial intensity of each state is updated using equation 13.

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (13)$$

Where, $\tau_{ij}(t+1)$ and $\tau_{ij}(t)$ are new and old trial intensities of (s_i, s_j) . ρ is a constant s.t. $0 < \rho < 1$. $\Delta\tau_{ij}(t, t+1)$ is calculated as follows (equation 14).

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^n \Delta\tau_{ij}^k(t, t+1) \quad (14)$$

Where, n is number of ants. k-th ant from s_i to s_j updates the value of trial intensity as $\Delta\tau_{ij}^k(t, t+1)$ at t-th iteration. It is calculated using equation 15.

$$\Delta\tau_{ij}^k(t, t+1) = \begin{cases} \frac{1}{L_k} & \text{if the } k-\text{th ant goes from } s_i \text{ to } s_j \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

Where, L_k is the path length travelled by k-th ant. The ACO system can be mapped onto any optimization problem with discrete state space.

III. EDGE DETECTION USING ANT COLONY OPTIMIZATION

Zhuang [13] used ACO to find edges in an image. Adjacent points in an image depict a relationship among themselves which is represented using perceptual graph. A machine vision model is constructed using layered model of machine vision based on ant colony optimization. ACO helps in identifying the edge features in digital images. Perceptual graph contains weights on edges and represents connection between neighbouring points. ACS builds an evolving pheromone field corresponding to perceptual graph. Each point in the image which is not on the border has 4 neighbours, hence four connections to adjacent points. Figure 1 depicts the perceptual graph example.

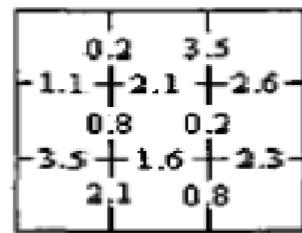


Fig. 1. Perceptual graph Example

Edge Feature Extraction is carried out as – one ant is assigned corresponding to each pixel. Ant starts moving for predefined number of steps. The cost of the path is defined as equation 16.

$$C^k(t) = \sum_{i=1}^n \frac{1}{\max(|\Delta_i^k(t)|, 0.05)} \quad (16)$$

Where, cost of k-th ant is represented as $C^k(t)$ in t-th iteration. n is predefined number of steps for an ant. $\Delta_i^k(t)$ is difference between points s_{i-1} and s_i covered by ant in i-th step. Trial intensity is updated as equation 17.

$$\Delta\tau_{ij}^k(t, t+1) = \begin{cases} \frac{1}{C^k(t)} & \text{if the } k\text{-th ant goes from } s_i \text{ to } s_j \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

In the experiments three characteristics of perceptual graph are examined (1) Length of edge point's weight vector, (2) maximum of each edge point's weight vector and (3) Variance of each component of weight vector.

Hosseini et al. [14] too considered input image as graph represented as two-dimensional adjacency matrix. Each pixel is a node. Each pixel has eight neighbours as shown in figure 2.

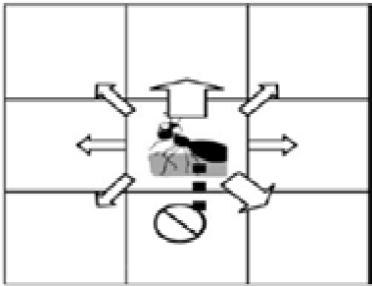


Fig. 2. Node transition strategy

Each pixel is assigned at most one ant which traverses the image pixel by pixel. m ants are placed and initial intensity of pixels is set to 0.0001. Ants on background pixels surrounded by background pixels are relocated to other pixels randomly. Otherwise, ant moves to one of eight neighbour pixels as per the following equation 18:

$$P_{(p,q)(r,s)}^k = \begin{cases} \frac{[\tau_{(r,s)}]^\alpha \cdot [\eta_{(r,s)}]^\beta}{\sum_u \sum_v [\tau_{(u,v)}]^\alpha \cdot [\eta_{(u,v)}]^\beta} & \text{if } (r,s) \text{ and } (u,v) \in \text{admissible nodes} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Where, ant moves from (p,q) to (r,s). τ_{rs} is pheromone intensity and η_{rs} is pixel visibility (r,s) calculated using equation 19. α and β are control parameters.

$$\eta_{(r,s)} = \frac{1}{I_{Max}} \times \text{Max} \left[\begin{array}{l} |I(r-1, s-1) - I(r+1, s+1)|, \\ |I(r-1, s+1) - I(r-1, s+1)|, \\ |I(r, s-1) - I(r, s+1)|, \\ |I(r-1, s) - I(r+1, s)| \end{array} \right] \quad (19)$$

After every step pheromone is updated as shown in equation 20.

$$\tau_{rs}(\text{new}) = \rho \cdot \tau_{rs}(\text{old}) + \Delta\tau_{rs} \quad (20)$$

Lu et al [15] proposed an approach based on ACS to compensate broken edges. Ants discover broken edges and compensate them. Ants are assigned to each endpoints of the edge in the image. The ants may end up searching redundant region which incurs in computation time. To tackle this, the ants are partitioned into different groups and the ant stops moving if two ants of different groups meet at pixel or any ant intersects with already discovered path by some other ant or the nearby track has larger density as compared to the track discovered by this ant. Maximal neighbouring difference and maximal connective similarity avoids blur edges.

The visibility of path between pixel i and j is expressed as equation 21.

$$\eta_{ij} = \frac{V(p_j)}{\max\{1, |p_j - p_i|\}} \quad (21)$$

Where, p_i and p_j denote pixel intensity of pixel i and j respectively. $V(p_j)$ is neighbouring difference.

Pheromone trail is updated locally and globally using equation 22, 23 and 24.

$$\tau_{ij} = (1 - \xi) \cdot \tau_{ij} + \xi \cdot \tau_0 \quad (22)$$

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij} \quad (23)$$

$$\Delta\tau_{ij} = \sum_{m=1}^k \frac{\text{avg}(D_m)}{\tau_{max}} \quad (24)$$

Where, ξ and ρ are local and global pheromone evaporation rates respectively. $\text{avg}(D_m)$ represents average step length of ant m.

Tian et al. [16] proposed optimization algorithm for edge determination using ACO. The image of size $M1 \times M2$ is assigned K ants randomly. At n-th step, an ant, selected randomly, will move from pixel (r, s) to pixel (i, j) according to the probability defined in equation 25.

$$P_{(r,s),(i,j)}^{(n)} = \frac{[\tau_{i,j}^{(n-1)}]^\alpha \cdot [\eta_{i,j}]^\beta}{\sum_{(i,j) \in \Omega(r,s)} [\tau_{i,j}^{(n-1)}]^\alpha \cdot [\eta_{i,j}]^\beta} \quad (25)$$

Where $\tau_{i,j}^{(n-1)}$ represents quantity of pheromone the node (i, j) , $\Omega(r,s)$ signifies the adjacent nodes of the node (r,s) , $\eta_{i,j}$ denotes heuristic value at the node (i, j) . The stimulus of pheromone matrix and matrix of heuristics are denoted by the constants α and β , respectively.

The edge pixel is determined using threshold value T. The mean values of pheromone are categorised into two categories, values lower than initial value and values greater than initial values. Average of mean values of these two categories is used to calculate the threshold. The process converges when two consecutive means are same.

A similar approach is proposed by Jevtic et al. [17] which uses ant colony system for improvements of edges. This method provides a novel fitness function defined over connecting edge length and pixel's grayscale visibility. The pheromone intensity is initialized with the grayscale visibility matrix. The pixels which are chosen by the ants on their early routes have high probability of being the edge pixels. This reduces the computational overhead. Pixel transition rule is same as [B]. The fitness function of a pixel is defined as equation 26.

$$f_k = \frac{\bar{\xi}}{\sigma_{\xi} \cdot N_p} \quad (26)$$

Where, $\bar{\xi}$ is mean value of pixel's grayscale visibility and standard deviation of pixel's grayscale visibility is denoted by σ_{ξ} . N_p denotes number of pixels on the path.

The procedure stops when all the ants stop their iterations. The ants may end up either finding the endpoints or getting stuck without making any further advancement.

Jevtic et al. in [18] propose an ant colony system based edge detection algorithm which enhances image contrast using Multiscale Adaptive Gain (MAG). This method first extracts multiple grayscale images from the original grayscale image using MAG. ACS is then applied on each extracted image for edge determination. The pheromone trails of all images are accumulated to identify the final edges of the original image.

XIAO et al. [19] proposed an optimization algorithm for edge detection based on ACS. This approach eliminates blindness of the ants by putting up a heuristic function.

Baterina and Oppus [20] also presented edge detection technique based on ACS, where information of the edge at every pixel based on the route developed by the ants is represented using a pheromone matrix.

Benhamza et al. [21] proposed a technique, adaptive in nature, for determination of edges using the concepts of CO. The ants are injected with additional behaviour which helps them react towards local stimuli such as the ant can reproduce their progenitors and guide them in appropriate direction for enhancement of efficiency in search area or the ant can die too to eliminate ineffective search or if it has exceeded a specific age. A population regulation method is used for elimination and creation of ants. Each ant is initialized with an initial energy $e(0)$. The energy is updated using following equation 27.

$$e(t) = e(t - 1) - \alpha + \alpha \left(\frac{\Delta_{gl}}{\max \Delta_{gl}} \right) \quad (27)$$

Where, Δ_{gl} is median of value of grayscale levels of previous cell and current cell, $\max \Delta_{gl}$ is maximum of Δ_{gl} obtained in previous iterations. α is set 0.025.

Probability of death of an ant of age 'a' is determined as follows (equation 28).

$$P = 1 - e(a) \quad (28)$$

Similarly, the probability of reproduction depends on two factors - number of cells, around the target cell, which are engaged by other ants and difference in gray levels of two cells. The formula of probability of reproduction is defined in equation 29.

$$P = W(n) \cdot (\mu + (1 - \mu) \cdot \left(\frac{\Delta_{gl}}{\max \Delta_{gl}} \right)) \quad (29)$$

Where, μ is constant and $W(n)$ is weight affecting neighbors.

Ari et al. [22] developed an effective ACO based edge detection technique which uses Fisher ratio (F ratio). Most suitable threshold value from pheromone matrix is determined by using F ratio. Binary edge map is also extracted from pheromone matrix using F ratio. F ratio is defined as equation 30.

$$\begin{aligned} F \text{ ratio} &= \frac{\text{Variance of means between the clusters}}{\text{Average Variance within the clusters}} \\ &= \frac{\frac{1}{k} \sum_{j=1}^k (\mu_j - \bar{\mu})^2}{\frac{1}{k} \sum_{j=1}^k \frac{1}{n_j} \sum_{i=1}^{n_j} (x_{ij} - \mu_j)^2} \end{aligned} \quad (30)$$

Where, number of clusters is k, j-th cluster contains n_j data points, mean of j-th cluster is μ_j and total mean is $\bar{\mu}$, i-th data of j-th cluster is denoted by x_{ij} . Based on the initial threshold T, the pheromone matrix is partitioned into two clusters C1 and C2. F ratio of each possible threshold is calculated as equation 31.

$$F \text{ ratio}_T = \frac{(\mu_1 T - \mu_2 T)^2}{2(v_1 T - v_2 T)} \quad (31)$$

Where, $\mu_1 T$ and $\mu_2 T$ are means of C1 and C2, $v_1 T$ and $v_2 T$ are variance of C1 and C2 respectively.

Liu and Fang [23] proposed a robust edge detection using ACO with a novel heuristics deployed along with threshold defined by the user to update pheromone using new parameters. The new heuristic function is defined as equation 32.

$$\eta_{pq} = \frac{1}{I_{max}} \cdot \max[|I_{(p-u,q-v)}, I_{(p+u,q+v)}|] \quad (32)$$

Where, I_{max} is maximum intensity value of gray-level image. I_{pq} is intensity of pixel (p, q) and $\max[\dots]$ is maximum difference of intensities of two pixel having same color. This approach gives upper hand as compared to traditional approaches but its computational overheads a bit higher as compared to other traditional approaches which make it slow.

Kumar and Raheja [24] proposed an adaptive edge detection method based on ACO which takes weighted average of threshold as compared to other approaches which

take simple average of threshold during pheromone update step. The weighted average of threshold is defined as equation 33.

$$Th^q = \frac{w_1 m_L^{(q)} + w_2 m_U^{(q)}}{2} \quad (33)$$

Where, $m_L^{(q)}$ and $m_U^{(q)}$ are means of intensities of pixels having intensity lower than threshold and greater than threshold respectively, $w_1 + w_2 = 1$. The proposed algorithm outperforms recently proposed methods in F-score. The overall efficiency is 87%.

Kumar and Raheja [25] in further research proposed a method for edge detection using guided image filtering to enhance the edges then ACO is used to find the edges. The

computational complexity does not increase in this approach and remains same as normal ACO based edge detection method.

IV. COMPARISON OF PARAMETER VALUES USED IN ACO BASED TECHNIQUES

Table I displays parameter values taken by various techniques discussed in previous section. However, new researchers did not mention clearly the values of some or all parameters used during experiments. Such values are denoted by ‘-’ in the table I.

Table II and table III show the comparative performance of various ACO based edge detection techniques discussed in section 3.

TABLE I. VALUES OF PARAMETER USED BY ACO BASED EDGE DETECTION TECHNIQUES

Reference	K	τ_0	α	β	ρ	ψ	T	N	L
[13]	-	-	-	-	-	-	-	-	-
[14]	$\sqrt{M \times N}$	0.0001	2.5	2	0.04		0.08		40 / 30 / 20
[15]	-	0.000001	1	2	0.3 (Global) 0.2 (Local)	-	0.05	-	100 to 400
[16]	$\sqrt{M \times N}$	0.0001	1	0.1	0.1	0.05	Variable.	4	40
[17]	Equal to end points of found edges.	Equal to grayscale visibility value	10	1	0.05	-	-	-	100
[18]	$\sqrt{M \times N}$	0.01	2.5	2	0.04	-	-	-	40/80 for 128 × 128 / 256 × 256
[19]	$2\sqrt{M \times N}$	0.1	1	0.1	0.1	0.05	Otsu [26]	4	40
[20]	$\sqrt{M \times N}$	0.1	1	1	0.1	0.05	-	10	40
[21]	-	-	-	-	-	-	-	-	-
[22]	$\sqrt{M \times N}$	0.0001	4	0.2	0.05	0.05	-	2	250
[23]	$\sqrt{M \times N}$	0.0001	2	2	0.02	-	-	3	$\sqrt[3]{M \times N}$
[24]	vary	0.0001	1	0.1	0.1	0.05	Adaptive	8	40
[25]	$\sqrt{M \times N}$	0.0001	1	0.1	0.1	0.05	Adaptive	-	40

TABLE II. PERFORMANCE COMPARISON OF ACO BASED EDGE DETECTION TECHNIQUES.

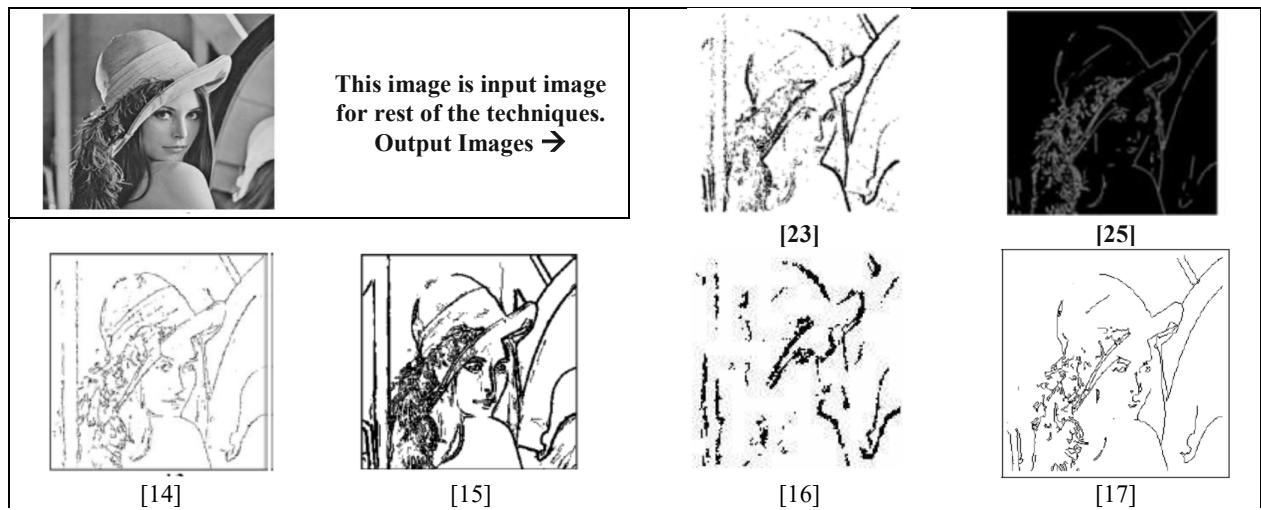




TABLE III. PERFORMANCE COMPARISON OF ACO BASED EDGE DETECTION TECHNIQUES

Reference	Original Image	Output
[13]		
[21]		
[24]		

V. CONCLUSION

Edge detection is identifying image boundaries caused by discontinuity in intensity. Edge detection covers a major portion of computer vision. Traditionally, first and second order derivatives of the image are prominently used for edge detection. Roberts, Prewitts, Sobel and Frei-Chen are few significant approximations utilized for quantification of pixels. These are generally used in the techniques based on first order derivatives which are highly sensitive to the noise. Second order derivative based techniques like LoG are also sensitive to noise.

Due to adaptive behaviour ACO has substantial application in edge detection in digital images. ACO technique originally mimics the behaviour of ants to find optimal path. Researchers have successfully applied ACO for edge determination in digital images by interpreting the digital image as perceptual graph. Number of ants initially put on the image, control parameters, pheromone evaporation rate, number of construction steps and ant movement steps majorly affect the quality of detected edges.

The results are presented comparatively which establishes the fact that relevant application of ACO in edge detection and computer vision is no less than other more popular approaches like machine learning and deep learning.

REFERENCES

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2007.
- [2] D. H. Lim, "Robust edge detection in noisy images," *Computational Statistics and Data Analysis*, vol. 50, no. 3, pp. 803–812, 2006.
- [3] H. Chidiac and D. Ziou, "Classification of image edges," in *Proceedings of the Conference on Vision Interface, Canada*, 1999, pp. 17–24.
- [4] T. Poggio and V. Torre. Ill-Posed Problems and Regularization Analysis in Early Vision. Artificial Intelligence Lab. Memo, No. 773, Massachusetts Institut of Technology, 1984.
- [5] T. Poggio and V. Torre. A Regularized Solution to Edge Detection. Artificial Intelligence Lab. Memo, No. 833, Massachusetts Institut of Technology, 1985.
- [6] A. J. Pinho and L. B. Almeida, "A review on edge detection based on filtering and differentiation," *REVISTA DO DETUA*, vol. 2, no. 1, pp. 113–126, 1997.
- [7] M. Roushdy, "Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter," *ICGST International Journal on Graphics, Vision and Image Processing*, vol. 6, pp. 17–23, 2007.
- [8] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "Comparison of edge detectors: a methodology and initial study," *Computer Vision and Image Understanding*, vol. 69, no. 1, pp. 38–54, 1998.
- [9] E. Clavier, S. Clavier, and J. Labiche, "Image sorting - image classification: A global approach," in *ICDAR '99: Proceedings of the Fifth International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 1999, pp. 123–129.
- [10] J. Cannby, "Finding edges and lines in images," Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1983.
- [11] M. Sharifi, M. Fathy, and M. T. Mahmoudi, "A classified and comparative study of edge detection algorithms," in *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2002, pp. 117–120.
- [12] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406), 1999, pp. 1470–1477 Vol. 2, doi: 10.1109/CEC.1999.782657.
- [13] X. Zhuang, "Edge feature extraction in digital images with the ant colony system," *2004 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2004. CIMA., 2004, pp. 133–136, doi: 10.1109/CIMA.2004.1397248.
- [14] Nezamabadi-pour, H., Saryazdi, S. & Rashedi, E. Edge detection using ant algorithms. *Soft Comput* 10, 623–628 (2006). <https://doi.org/10.1007/s00500-005-0511-y>.
- [15] Lu, De-Sian, and Chien-Chang Chen. "Edge detection improvement by ant colony optimization." *Pattern Recognition Letters* 29.4 (2008): 416–425.
- [16] Jing Tian, Weiyu Yu and Shengli Xie, "An ant colony optimization algorithm for image edge detection," 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 751–756, doi: 10.1109/CEC.2008.4630880.
- [17] Jevtić, Aleksandar & Melgar, Ignacio & Andina, Diego. (2009). Ant based edge linking algorithm. *IECON Proceedings (Industrial Electronics Conference)*. 3353 - 3358. 10.1109/IECON.2009.5415195.
- [18] Jevtić, J., Quintanilla-Dominguez, M. G., Cortina-Januchs and D. Andina, "Edge detection using ant colony search algorithm and multiscale contrast enhancement," 2009 IEEE International Conference on Systems, Man and Cybernetics, 2009, pp. 2193–2198, doi: 10.1109/ICSMC.2009.5345922.

- [19] Xiao, Peng, Jun Li, and Jian-Ping Li. "An improved ant colony optimization algorithm for image extracting." The 2010 International Conference on Apperceiving Computing and Intelligence Analysis Proceeding. IEEE, 2010.
- [20] Peng Xiao, Jun Li and Jian-Ping Li, "An improved ant colony optimization algorithm for image extracting," The 2010 International Conference on Apperceiving Computing and Intelligence Analysis Proceeding, 2010, pp. 248-252, doi: 10.1109/ICACIA.2010.5709894.
- [21] O. P. Verma, P. Singhal, S. Garg and D. S. Chauhan, "Edge detection using adaptive thresholding and Ant Colony Optimization," 2011 World Congress on Information and Communication Technologies, 2011, pp. 313-318, doi: 10.1109/WICT.2011.6141264.
- [22] Ari, Samit & Ghosh, Dipak & Mohanty, Prashant, "Edge detection using ACO and F ratio. Signal, Image and Video Processing", 8. 10.1007/s11760-013-0569-4, 2014.
- [23] Liu, Xiaochen, and Suping Fang. "A convenient and robust edge detection method based on ant colony optimization." Optics Communications 353, 147-157, 2014.
- [24] Raheja, S., & Kumar, A., "Edge Detection using Ant Colony Optimization under Novel Intensity Mapping Function and Weighted Adaptive Threshold", International Journal of Integrated Engineering, 12(1), 13-26, 2020.
- [25] Kumar, Akshi & Raheja, Sahil., "Edge Detection using Guided Image Filtering and Enhanced Ant Colony Optimization". Procedia Computer Science. 173. 8-17. 10.1016/j.procs.2020.06.003, 2020.
- [26] N.Otsu, "A threshold selection method from gray level histograms, "IEEE Trans.Syst.,Man,Cybern., vol.9, pp.62–66, Jan.1979.