iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** II **Month of publication:** February 2021

DOI: https://doi.org/10.22214/ijraset.2021.32999

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

Overview of Swarm Optimization Techniques (Ant-Colony Optimization and Bee Colony Optimization) for Image Edge Detection

Sai Dheeraj Chindam[1], Avyay Yennamaneni[2]
[1, 2]Sreenidhi institute of science and technology

***Abstract:** Image edge detection mainly involves in identifying points in an image at which the image brightness changes sharply or more formally has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. Edges give important structural information about the images which would play a vital role in various fields of image processing and computer vision. In order to optimize the process of image edge detection and produce better edges for images, we have implemented edge detection using two swarm intelligence techniques. In this chapter we will study in detail how ant colony optimisation and particle swarm optimization techniques were utilised to identify edges of an image.*
***Keywords:** Edge, Detection, Swarm, Images*

I. INTRODUCTION

The concept of edge detection mainly involves in identifying local variation in brightness levels in images. Image edge detection is mainly employed for image processing and computer vision. Computer vision helps in extracting useful information from the detected objects such as images, videos.
In last two decades numerous techniques and variations have been developed. Despite them being carried out, main skeleton of these techniques remains similar. With varying application domains, most of these techniques have been modified to fit into various applications. The changes undergo mostly in perspective of encoding scheme, parameter tuning and search strategy.
The edge detection task is to find the boundaries of image regions based on properties such as intensity and texture [3]. It is a critical low-level procedure of image processing because edges carry a lot of information. The edge detection process generally includes five steps:

1) *Filtering:* Filtering out noise from the image and improving the performance of the edge detector.
2) *Enhancement:* Emphasising pixels which have important change in local intensity.
3) *Detection:* Identifying the edges and thresholding.
4) *Link:* Linking the broken edges (such as hysteresis thresholding techniques).
5) *Localisation:* Locating the edge accurately and estimating the edge orientation (edge and orientation map).

In the modern age computer vision concepts are useful for detecting checks through images and in automated driving cars. Edge detection helps in identifying the important structural object of the image, which thereby is used for identifying important information from image such as number, size, and relative location of objects in an image. If we compare black and white grayscale images the variation in colour for our eyes is very sudden but at pixel level, it is not very rapid. The edge detected is a constant change of pixel values when we travel from one bright level to another bright level in the space of pixels of an image. Usually, the technique for identifying variations in the pixel values to identify edge variation is calculated for each pixel value in image space which is not an optimistic solution for identifying an image. So, to optimize the solution for this scenario in this paper we will further discuss about implementing swarm intelligence for detecting edges. Swarm intelligence mainly involves the study of collective and individual behaviour of biological systems which help in improving efficiency for solving complex problems such as identifying shortest path for food source or identifying their enemies in nature. Even though all the insects are unsophisticated and unorganised they make wonders by adapting to their problems in environment and solving their daily life scenarios.

This paper mainly focuses mainly on implementing edge detection using below mentioned swarm computing techniques.

a) Ant Colony optimization.
b) Particle Swarm Optimization

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

II. ANT-COLONY OPTIMIZATION

A. Introduction

This optimization technique is inspired from the ants in our nature [1],[2]. Even though ants are unsophisticated, Ants process for finding food source can be implemented in real time technologies to optimize and obtain greater results while solving complex problems[1],[4].In general all ants have to find a food source to survive ,so if any of the ant finds a way to the food source, On its path ants deposit a chemical substance known as pheromone(a fluid produced by ants) which will act as a route map for the other ants. In this scenario if any other ant reaches to the same food source and returns home, all the other ants compare the routes with the density of the deposited pheromone value. One of the characteristics of this liquid is, it evaporates over time which results in the decrease of density of the fluid. So, if the density of the liquid is high in this case distance to the food source would be shorter and similarly if density is less vice versa. Ants follow this technique to direct heavily populated ants to choose the shortest path to food source instead of choosing the longest path.

B. Ant Colony Implementation

Initially all the ants are distributed over the image and the edges are identified based on the movement of ants, deposited pheromone values on each pixel and their evaporation factor with defined constraints as per the iteration termination conditions. So here it mainly involves in three steps.

1) Initialisation Process
2) Constructive and Iterative Process
3) Decision Process

The major agenda for these iterations and updating is to obtain the data in the final pheromone matrix.

C. Initialisation Process

Initially when an image is considered of size(A1*A2) random ants are placed at random positions. Here the heuristic information of a pixel is calculated using neighbouring pixel values associated to it. This heuristic information is useful in next step while calculating transition probability which helps ants to move from one pixel to other.

$$\eta_{i,j} = \frac{V_c(I_{i,j})}{V_{max}} \tag{1}$$

The Heuristic Information at pixel (i,j) is given by the local coordinates for that position.
Below equation is the local variation of neighbouring pixel values where Vmax gives maximum intensity Variation (2).

$$V_c(I_{i,j}) = \left|I_{i-1,j-1} - I_{i+1,j+1}\right| + \left|I_{i-1,j} - I_{i+1,j}\right| + \left|I_{i-1,j+1} - I_{i+1,j-1}\right| + \left|I_{i,j-1} - I_{i,j+1}\right| \tag{2}$$

Ants move from one pixel to another pixel only to its neighbouring pixels as shown in below image 1(A)

1) *Constructive and Iterative Process:* This step is crucial for updating the pheromone values, movement of ants and updating the data in pheromone matrix. In this method movement of ants is determined by the transition probability for every iteration performed. The ant makes its move from one pixel(i,j) to (i*,j*) based on pseudo normal transition probability(3).

$$p^{(n)}_{(i_0,j_0),(i,j)} = \frac{\left(\tau^{(n-1)}_{i,j}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}}{\sum_{(i,j)\in\Omega_{(i_0,j_0)}}\left(\tau^{(n-1)}_{i,j}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}} \tag{3}$$

Whenever any ant reaches the pixel it updates the pheromone locally by the Pheromone equation (4). If an ant visits the pixel (i*, j*) the pheromone value on the nth iteration would be based on the equation (4)

$$\tau^{(n)}_{i,j} = (1-\varphi)\cdot\tau^{(n)}_{i,j} + \varphi\cdot\tau_{init} \tag{4}$$

The ants move from one location to another with 8 neighbouring algorithm and it avoids them to visit the frequently visited pixels. Once ants finish their constructive steps in iteration the global pheromone update is made for the pixel value which has at least one ant visited during the iteration process by global pheromone update given by (5)

$$\tau^{(n)}_{i,j} = (1-\varphi)\cdot\tau^{(n)}_{i,j} + \varphi\cdot\tau_{init} \tag{5}$$

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

2) *Decision Process:* Once the iteration terminates with a set of constructive steps and pheromone updates globally and locally throughout the process the final updated pheromone matrix values are analysed to identify edges with varying pixel values in a connected graph. The final decision of this process is obtained by applying a threshold to the matrix and the values are based on a method given in [5] called otsu thresholding technique.

a) *Pseudo Code of ACO Technique*

Do initialization procedures
for each iteration n = 1:N do
for each construction step l = 1:L do
for each ant k = 1:K do
Select and go to next pixel
Update pixel's pheromone (local)
End
end
Update visited pixels pheromones (global) end

Parameters of the transition probability are:
N: number of iterations
L: number of construction steps
K: number of ants
q0: parameter for controlling the degree of exploration of the ants
α: parameter for controlling influence of pheromone trail (fixed to 1 for ACS)
β: parameter for controlling influence of heuristic information
ψ: pheromone decay coefficient
ρ: pheromone evaporation coefficient

Default values used in above equation are:
Initial Pheromone Value=0.1
N=10
L=40
K=512(256*256)
q0(variable)
$\alpha = 1$
$\beta = 1$
$\psi = 0.05$
$\rho = 0.1$

b) *Test Results (As per Constant q0)*

Fig. 1 Input image:

Fig. 2 (q0) = 0.0 (Output image when q0=0.0)

Fig. 3 (q0) = 0.1(Output image when q0=0.1)

III. PARTICLE SWARM OPTIMIZATION

A. Introduction

Particle swarm optimization (PSO) is a population based evolutionary algorithm for solving optimization problems based on social-psychological principles. Introduced by Kennedy and Eberhart in 1995, this technique falls under swarm intelligence which is collective behaviour of organised patterns in eco system. Particle swarm optimization has roots in two main component methodologies. Perhaps more obvious are its ties to artificial life in general, and to bird flocking, fish schooling, and swarming theory. It is also related to evolutionary computations such as genetic algorithms and evolutionary programming. It was originally an optimization technique for continuous nonlinear functions; however, some discrete versions of PSO have also been proposed to date. In PSO, there is a finite population of individual solutions (called particles), each having a memory of previous states.

Particle swarm optimization is a simple, yet robust technique and it can be implemented in only few lines of code. It uses only primitive mathematical operators and is very economical in terms of both memory and speed.

Some advantages of PSO technique in comparison to other population-based evolutionary algorithms, such as genetic algorithm (GA), are ease of its implementation and high rate of convergence.

PSO has been successfully applied to training neural networks, optimizing power systems, fuzzy control systems, robotics, radio and antenna design, and computer games.

B. PSO Core Concept

Particle swarm optimization technique is used in various complex computational scenarios to get an optimal solution. This technique was inspired from biological populations and their social behaviour.

Let us consider a scenario between Bob and John who are on 2 different sides of river let us consider a 2d view. So, the problem statement here is to find out the depth of the river in some set of days. So according to PSO John and Bob move simultaneously with respect to time while learning from each other's depth measurements. Initially when John moves to a certain distance, he finds the depth at his position similarly, Bob does the same. After certain amount of time by end of the day both Bob and John share the information of depths in the river. Let us assume Bob measured highest measured value for depth, for the next day Bob remains at same place but John moves towards Bob's direction. In this way by learning from each other's measured values they reach deepest point in the river optimal way without wasting lot of days. The advantage of PSO is particle tries to learn from other particles for finding optimal solution in the entire search space. For each individual particle local best value is calculated and for all the particles global best is calculated (As per above 2d depth measurement scenario there will be one global best value for all the particles similar to current deepest point in the lake, in case a new deepest point is calculated the global best value gets updated). With every iteration the particle location gets updated by learning from other particles

In our Scenario edge detection is implemented by the following steps:

- Applying Morphological Operator
- PSO Implementation
- Thresholding

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

1) *Morphological Operator:* Morphological operators are used to refine the image and this operation mainly involves in calculating the structural element (3*3) matrix for the pixels which are hovered over pixel values of the matrix and they are updated based on the neighbouring pixel values and structural (3*3) matrix pixel values.

Below mentioned Morphological Operations are applied on image.

a) Dilation
b) Erosion
c) Closing
d) Opening

Equations for morphological operators are given below.

$$E1 = \sum \{(M.\ Bi) \ominus Bi - (M.\ Bi)\} \quad (6)$$
$$E2 = \sum \{(M.\ Bi) - (M.\ Bi) \oplus Bi\} \quad (7)$$
$$E3 = \sum \{(M.\ Bi) \ominus Bi - (M.\ Bi) \oplus Bi\} \quad (8)$$

Where Bi is the square structuring element of 3×3. F is the input image then M is given by equation (9).

$$M = (F.\ Bi) \circ Bi \quad (9)$$

Final Output of Morphological Operator is:

$$E4 = (\sum Ei\ 3\ i{=}1\ ) /3 \quad (10)$$

2) *PSO Implementation:* In General, PSO Contains a population of candidate solutions called swarm. Every particle(pixel) is a candidate solution to optimization problem [6]. Every particle has a position in search space and has set of all possible values.

In this scenario, there are total number of m particles that travel through an n-dimensional search space (Image). The position of ith particle is given by equation (10).

$$Xi(t)=(Xi1(t), Xi2(t), Xim(t)) \quad (11)$$

This equation changes according to its history and neighbours, let us assume Xi(t) be the position of particle at time t. The Xi is changed at each iteration by adding velocity Vi is given by (11)

$$Xi(t + 1) = Vi(t) + Xi(t) \quad (12)$$

The Velocity is mainly dependant on current motion, particle memory influence, and swarm influence.

$$Vi(t + 1) = \omega Vi(t + 1) + C1r1\ (Xpbesti - Xi(t)) + C2r2\ (Xgbesti - Xi(t)) \quad (13)$$

In (12) The value of random variables r1 and r2 is between 0 and 1, ω is the inertia weight which controls the impact of the previous velocity C1 (self-confidence) and C (population confidence) are the learning factors. Xpbesti is the best position of ith so far and Xgbesti denotes the best position of population so far [7].

3) *Thresholding:* This final step is performed after the detection of edges by PSO, here the aim of thresholding is to analyse whether the edge is present at an image point. Once thresholding is completed the edge roadmap is generated. This transition to edges is given by (14).

$$f(x, y) =255,\ g(x, y) \geq T \quad (14)$$
(or)
$$f(x, y) =0, g(x, y) < T$$

The above described methodology is an optimal solution for detecting edges by implementing particle swarm optimization as core concept involved in detecting the best edge road map for a given input image.

4) *Test Results*

Fig. 4 Sample Input Image

Fig. 5 quality edges obtained using PSO proposed method (optimal solution)

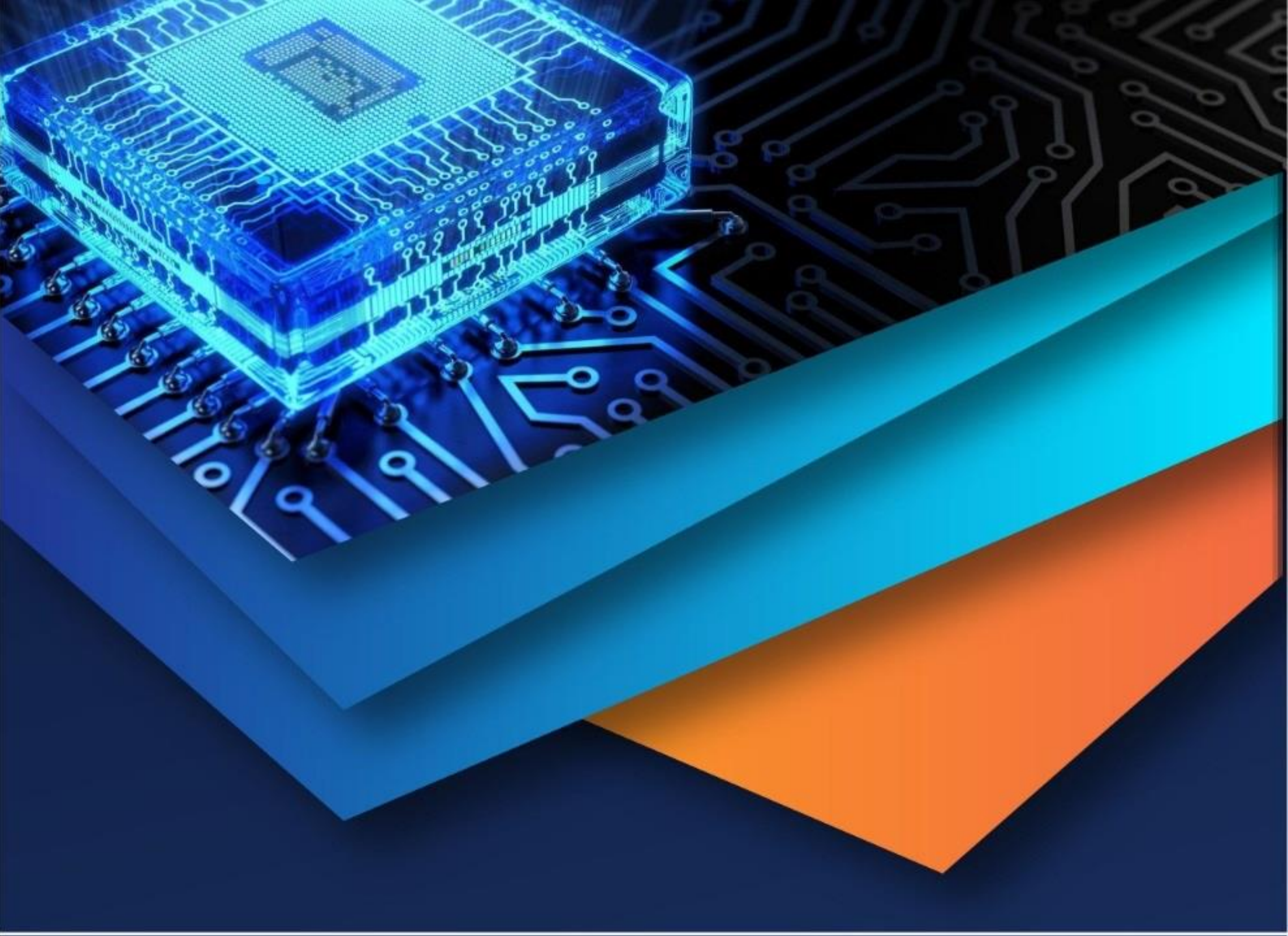
Fig. 6 edges obtained using Sobel operator
(not optimal)

IV. CONCLUSIONS

Image edge detection plays a vital role in understanding the information available in the image. Edge detection tries to localize the boundaries of an image, so it plays an important role in machine vision and image analysis. Usually we see implementation of edge detection in initial stages of many computer vision applications. Conventional approaches for edge detection are more expensive in terms of computation because each combination of operations is applied on each pixel and the computation time is directly proportional to the size of input image. Swarm Computing techniques have the capability to overcome this issue of conventional methodology. As mentioned in this paper the operation of edge detection is not carried on every pixel, there by this reduces the computation time while improving the quality of the image.

REFERENCES

[1] M. Dorigo and T. Stützle, Ant Colony Optimization, Cambridge: MIT Press, 2004.
[2] M. Dorigo. (2007) "Ant Colony Optimization," Scholarpedia, 2(3):1461. [Online]. Available: http://www.scholarpedia.org/article/ Ant_colony_optimization.
[3] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a Colony of Cooperating Agents," IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 26, pp. 29-41, 1996.
[4] M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," IEEE Transactions on Evolutionary Computation, vol. 1, pp. 53-66, 1997.
[5] N. Otsu, "A Threshold Selection Method from Gray-level Histograms Transactions on Systems, Man and Cybernetics, vol. 9, no. 1, pp. 62-66, 1979.
[6] J. Kennedy, "Particle swarm optimization," in Encyclopedia of Mach. Learning., Springer, 2011, pp. 760-766
[7] M. Setayesh, M. Zhang, and M. Johnston, "Detection of continuous, smooth and thin edges in noisy images using constrained particle swarm optimisation," in Proc. of the 13th Ann. Conf. on Genetic and Evol. Computation., ACM, 2011, pp. 45-52.," IEEE



doi crossref

10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR: 7.129

SJIFactor Scientific Journal Impact Factor TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

ISI