AOS Mini-Project- work

EDGE DETECTION USING ANT ALGORITHM

Somnath Paul, 19MCMI09

INSTRUCTOR

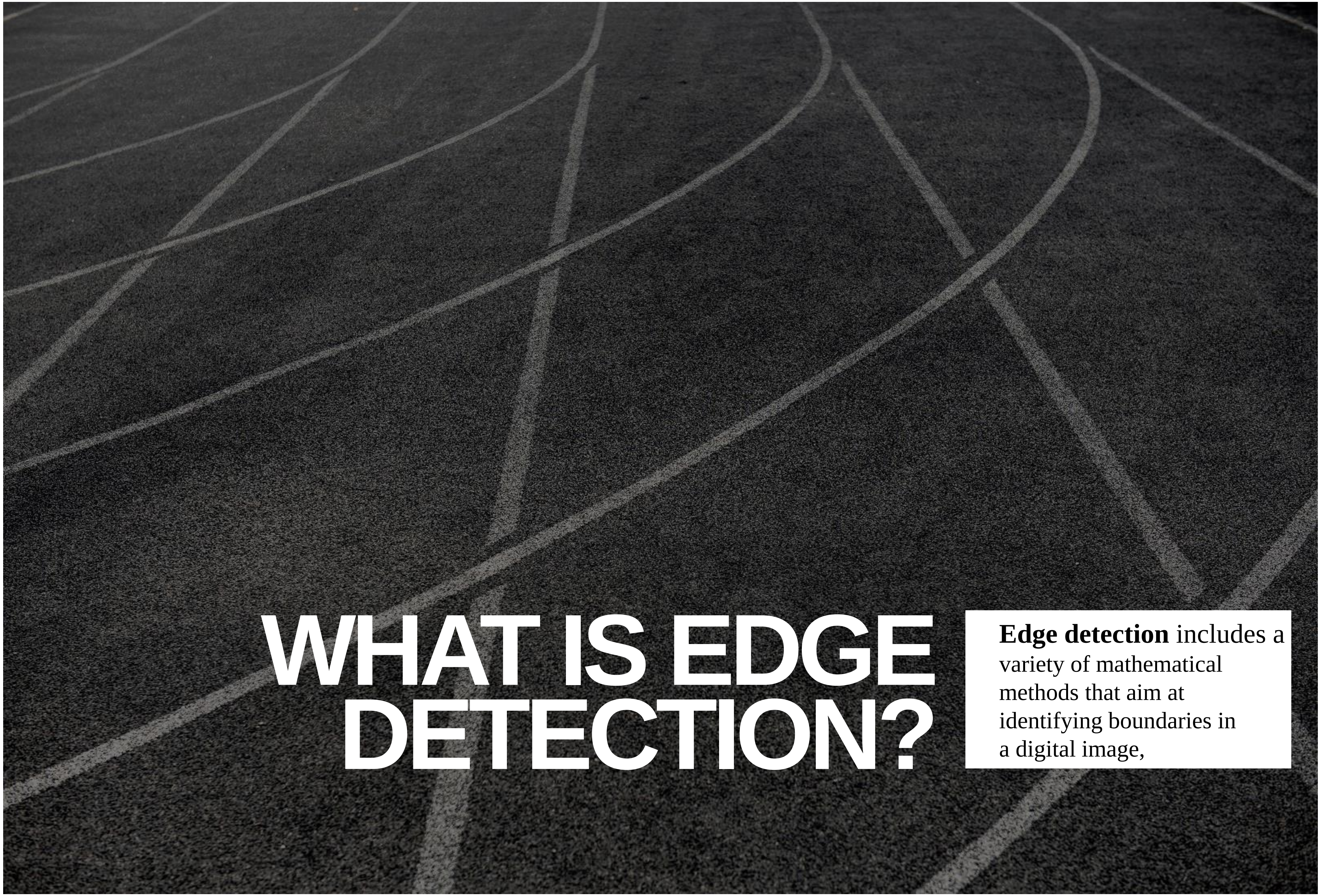
Dr. M Nagamani

Assistant Professor, School Of Computer and Information Science

University of Hyderabad, India

ABSTRACT

In this project a new algorithm for edge detection using ant colony search is proposed. The problem is represented by a directed graph in which nodes are the pixels of an image. To adapt the problem, some modifications on original ant colony search algorithm (ACSA) are applied.



WHAT IS EDGE DETECTION?

Edge detection includes a variety of mathematical methods that aim at identifying boundaries in a digital image,

EDGE DETECTION

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital 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. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

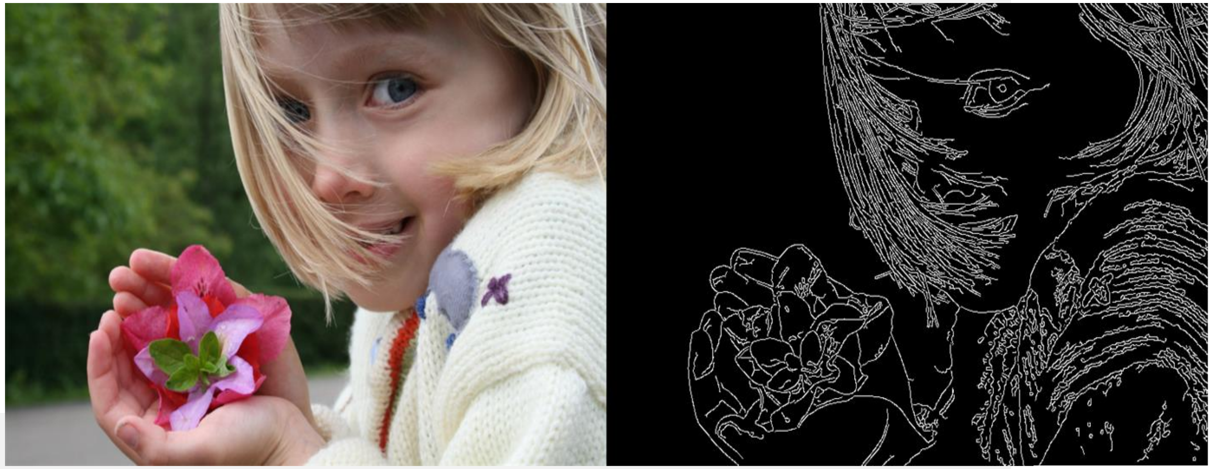
The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to:

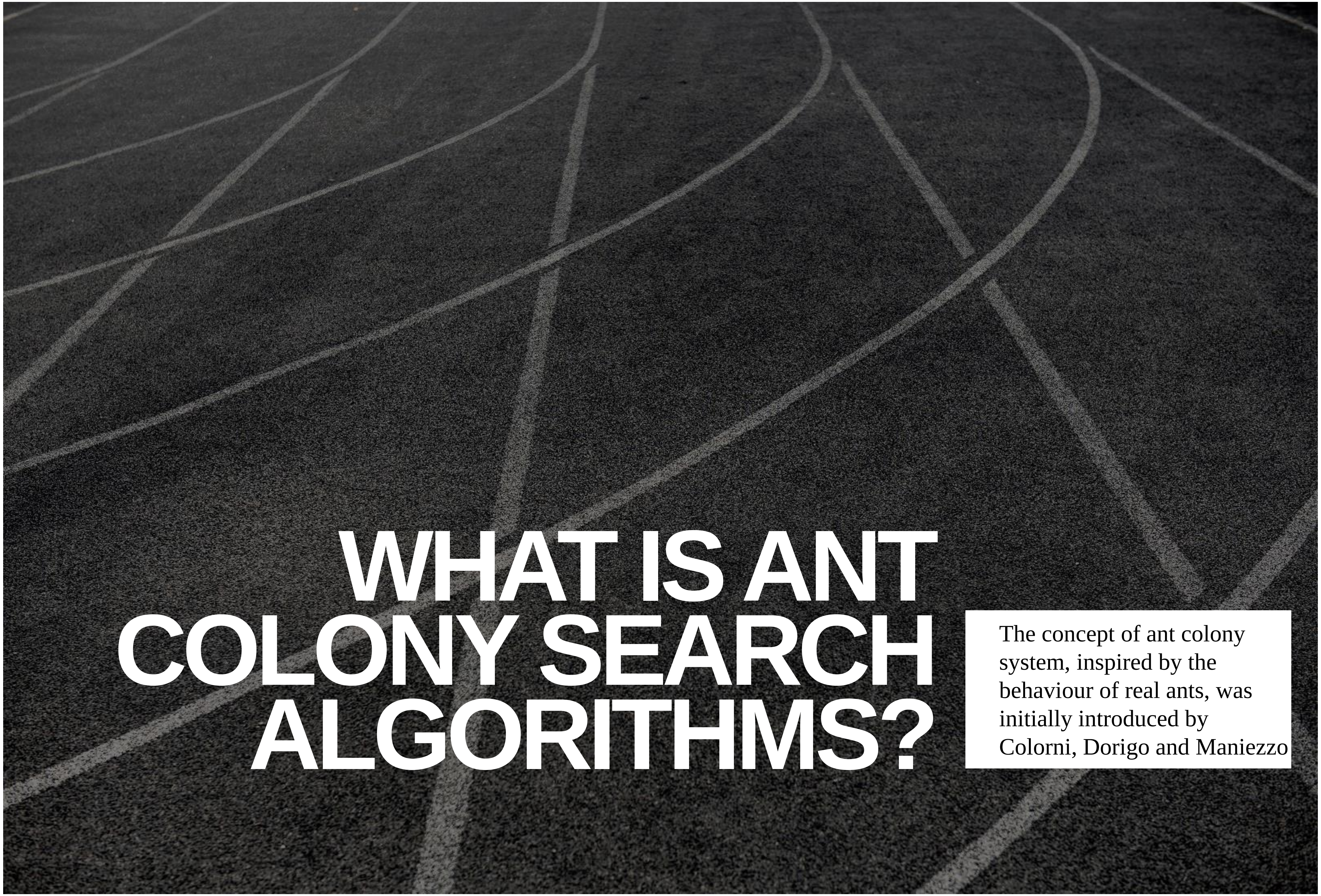
discontinuities in depth,

discontinuities in surface orientation,

changes in material properties and

variations in scene illumination.





WHAT IS ANT COLONY SEARCH ALGORITHMS?

The concept of ant colony system, inspired by the behaviour of real ants, was initially introduced by Colorni, Dorigo and Maniezzo

ANT COLONY SEARCH ALGORITHMS

Real ants follow their own agenda of tasks independent from each other, however, when act as a community, they are capable to solve their daily complex problems, which require sophisticated planning, such as selecting and picking up materials, or, finding and storing foods. However, there is no kind of supervising or controlling. Finding the shortest route between the colony and a food source, is done by an exchange of information about the path should be followed. Ants communicate with each other by means of pheromone trail. They mark the path by leaving a certain amount of pheromone. Ants probabilistically prefer to follow a direction, proportional to the amount of pheromone on it. The more ants follow a trail, the more attractive this trail becomes to be followed by other ants. This process could be considered as a positive feedback loop.

An artificial ant colony system is an agent-based algorithm, which simulates the behaviour of real ants. Artificial

ants are like real ants with some major differences:

– Artificial ants have memory;

– They are not completely blind;

– They live in a discrete time environment.

ANT COLONY SEARCH ALGORITHMS

However they have some adopted characteristics from real ants:

– Artificial ants probabilistically prefer paths with a larger amounts of pheromone.

– Shorter paths have larger rate of growth in the amount of pheromone.

– The ants communicate to each other by means of the amount of pheromone laid on each path.

ACS is an iterative algorithm. At each iteration, it performs a loop containing two basic operations:

(1) construction or modification of solutions of the problem,

(2) updating the pheromone trails.

ANT COLONY SEARCH ALGORITHMS

Different steps of an ACS algorithm are the followings:

1. Problem graph representation: problems which could be solved by ACSA are often discrete, so they could be represented by a graph with N nodes and E edges, G = N, E.
2. Initializing ants distribution: a number of ants are placed on the randomly chosen nodes.
3. Node transition rule: the node transition rule specifies how ants must move from node to node. The node transition is probabilistic. The probability of displacing kth ant from node i to node j is given by:

$$p_{ij}^{k} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{h \notin \mathrm{tabu}_k} (\tau_{ih})^{\alpha}(\eta_{ih})^{\beta}} & \text{if } j \notin \mathrm{tabu}_k, \\ 0 & \text{otherwise,} \end{cases}$$

where τij and ηij are the intensity of pheromone and the visibility of edge (i, j), respectively, and, α and β are control parameters. By tabu_k we present the set of inaccessible nodes.

ANT COLONY SEARCH ALGORITHMS

4. Pheromone updating rule: a cycle of ACS algorithm is completed when every ant has constructed a solution. At the end of each cycle, the intensity of pheromone is update by a pheromone trail updating rule:

$$\tau_{ij}(\text{new}) = (1 - \rho)\tau_{ij}(\text{old}) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}, \qquad (2)$$

where ρ represents the pheromone evaporation, and $\Delta\tau_{ij}^{k}$ is the amount of pheromone laid on edge (i, j) by the kth ant and could be given by:

$$\Delta\tau_{i,j}^{(k)} = \begin{cases} \frac{1}{L_k}, & \text{if ant k used edge (i, j)} \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

where $\mathbf{L_k}$ is the tour value of the solution found by $\mathbf{k^{th}}$ ant and 1 is a constant.

5. Stopping criterion: The end of the algorithm could be achieved by a predefined number of cycles, or the maximal number of cycles between two improvements of the global best solution.

ANT COLONY SEARCH ALGORITHMS

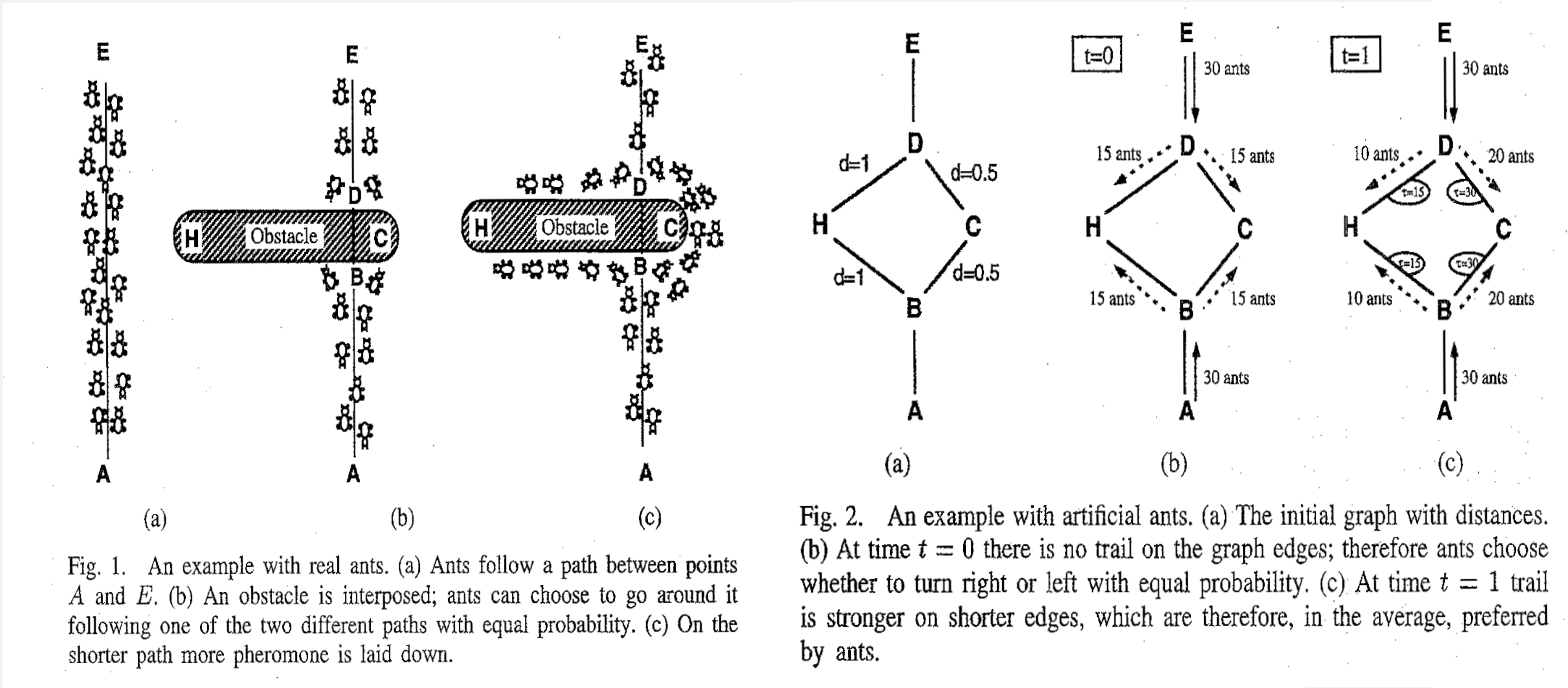
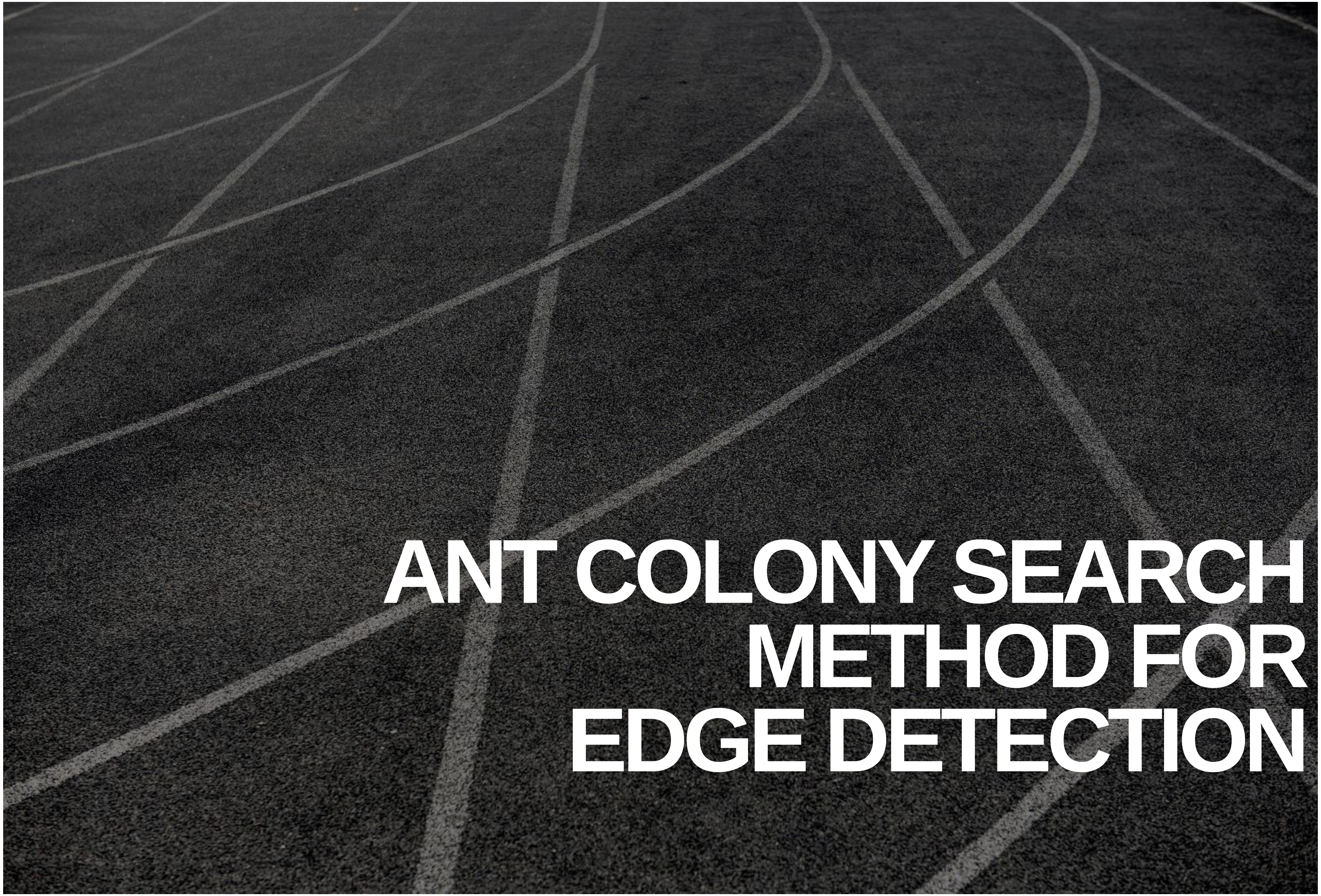


Fig. 1. An example with real ants. (a) Ants follow a path between points A and E. (b) An obstacle is interposed; ants can choose to go around it following one of the two different paths with equal probability. (c) On the shorter path more pheromone is laid down.

Fig. 2. An example with artificial ants. (a) The initial graph with distances. (b) At time $t = 0$ there is no trail on the graph edges; therefore ants choose whether to turn right or left with equal probability. (c) At time $t = 1$ trail is stronger on shorter edges, which are therefore, in the average, preferred by ants.



ANT COLONY SEARCH METHOD FOR EDGE DETECTION

ACO BASED EDGE DETECTION

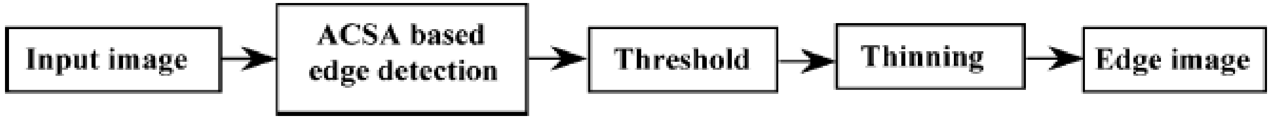
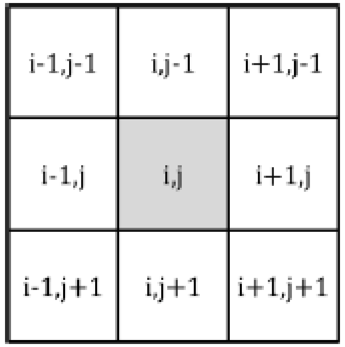
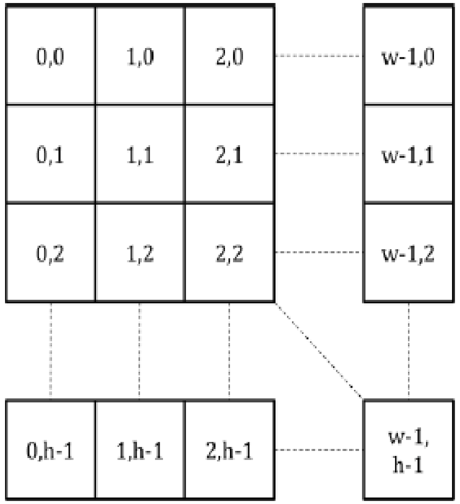

Input image → ACSA based edge detection → Threshold → Thinning → Edge image

Image edge detection can be thought of as a problem of identifying the pixels in an image that correspond to edges. A **w × h two-dimensional digital image** can be represented as a two-dimensional matrix with the image pixels as its elements.

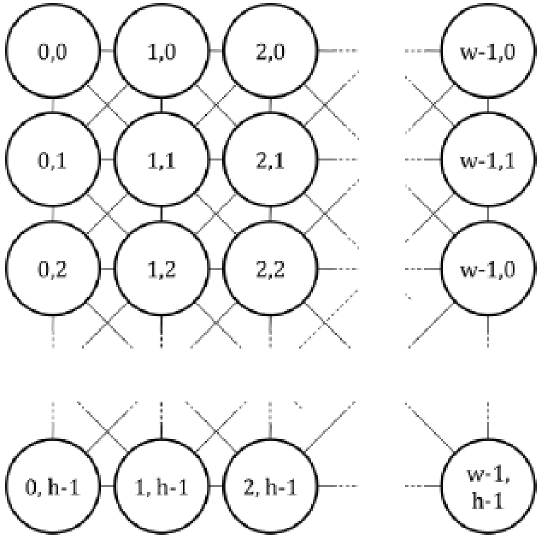
The graph is defined as follows. The components of the graph are the pixels of the image. The connections of the graph connect adjacent components or pixels together. An **8-connectivity pixel configuration is used**: a pixel is connected to every pixel that touches one of its edges or corners. Ants traverse the graph by moving from one pixel to another, through their connections. An ant cannot move to a pixel if it is not connected to the pixel where the ant is currently located. This means that an ant can move only to an adjacent pixel.



i-1,j-1	i,j-1	i+1,j-1
i-1,j	i,j	i+1,j
i-1,j+1	i,j+1	i+1,j+1



0,0	1,0	2,0	w-1,0
0,1	1,1	2,1	w-1,1
0,2	1,2	2,2	w-1,2
0,h-1	1,h-1	2,h-1	w-1, h-1



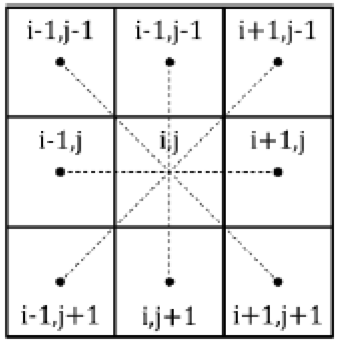
ACO BASED EDGE DETECTION ALGORITHM

1. Initialization Process

In the initialization process, each of the K ants is assigned a random position in the M1 x M2 image. The initial value of each element in the pheromone matrix is set to a constant τ_{init}, which is small but non-zero. Also, the heuristic information matrix is constructed based on the local variation of the intensity values. The heuristic information is determined during initialization since it is dependent only on the pixel values of the image, thus, constant.

$$\eta_{i,j} = \frac{1}{I_{Max}} \times \text{Max}\begin{bmatrix} |I(i-1, j-1) - I(i+1, j+1)|, \\ |I(i-1, j+1) - I(i-1, j+1)|, \\ |I(i, j-1) - I(i, j+1)|, \\ |I(i-1, j) - I(i+1, j)| \end{bmatrix}. \qquad (4)$$

In this approach, the value of visibility is determined using the maximum variation of gray level of the image intensity. Therefore, edge pixels are expected to have a greater value of visibility. In addition, all pixels currently registered in ants memory are considered as non-admissible pixels.

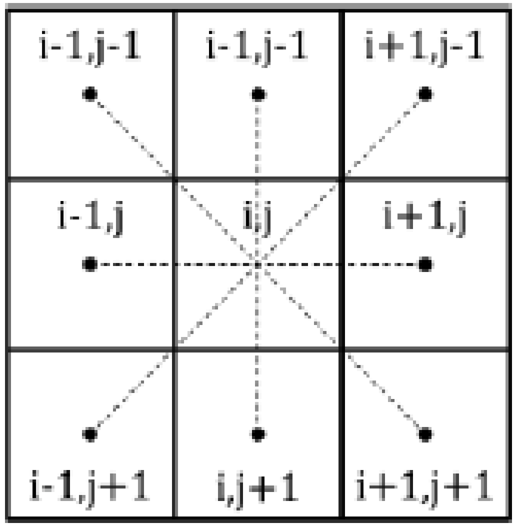


ACO BASED EDGE DETECTION ALGORITHM

2. Iterative Construction and Update Process

On every iteration, each ant moves across the image, from one pixel to the next, until it has made L construction steps (a construction step consists of a single movement from one pixel to another). An ant moves from the pixel (i_0, i_1) to an adjacent pixel (i, j) according to the pseudorandom proportional rule. The transition probability for the biased exploration is given by

$$p^{(n)}_{(i_0,j_0),(i,j)} = \frac{\left(\tau^{(n-1)}_{i,j}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}}{\sum_{(i,j)\in\Omega_{(i_0,j_0)}}\left(\tau^{(n-1)}_{i,j}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}} \qquad (9)$$



After every step, the pheromone level is updated according to Eq.

$$\tau_{(i,j)}(\text{new}) = (1-\rho)\tau_{(i,j)}(\text{old}) + \Delta\tau_{(i,j)} \qquad (5)$$

where

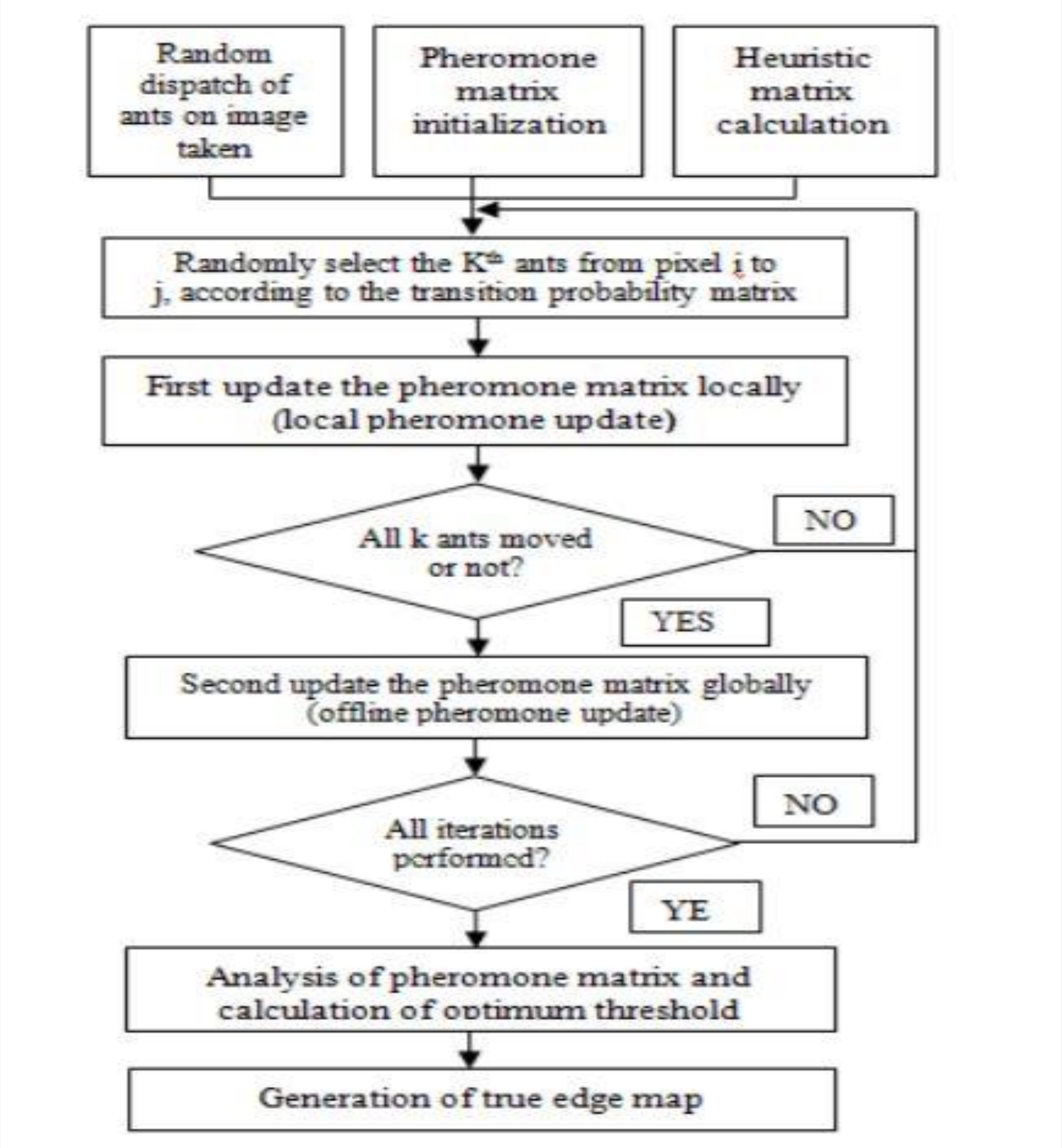
$$\Delta\tau_{(i,j)} = \sum_{k=1}^{m} \Delta\tau^{k}_{(i,j)}, \qquad \text{and,}$$

$$\Delta\tau^{k}_{(i,j)} = \begin{cases} \eta_{(i,j)} & \text{if } \eta_{(i,j)} \geq b \& k\text{th ant displaces to pixel } (i,\ j) \\ 0 & \text{otherwise} \end{cases}$$

2. Stopping criterion:

The end of algorithm achieves by a pre-defined number of cycles, for which each cycle contains a fix number of steps. The number of cycles is chosen to be 3. This value works well enough for all the experiments.

ACO BASED EDGE DETECTION ALGORITHM



Random dispatch of ants on image taken

Pheromone matrix initialization

Heuristic matrix calculation

Randomly select the K^{th} ants from pixel i to j, according to the transition probability matrix

First update the pheromone matrix locally (local pheromone update)

All k ants moved or not?

NO

YES

Second update the pheromone matrix globally (offline pheromone update)

All iterations performed?

NO

YE

Analysis of pheromone matrix and calculation of optimum threshold

Generation of true edge map

HOW EDGE DETECTION WORKS ON OS

HOW EDGE DETECTION WORKS ON OS

- A Digital Image Processing system is the combination of the computer hardware and the image processing software.
- Edge detection is part of image processing. We need a Operating System to run a edge Detection.
- We need an Input device to take a input as a image file. And an output Device to show output, CPU and GPU to compute different functionality, Memory to store data, We need compiler to compile our programs.

REFERENCES

- Image Edge Detection Using Ant Colony : Anna Veronica Baterina and Carlos Oppus : January 15, 2010.
- Ant System: Optimization by a Colony of Cooperating : Marco Dorigo, Member, ZEEE, Vittorio Maniezzo, and Albert0 Colorni : December 28, 1994
- Edge detection using ant algorithms: Hossein Nezamabadi-pour · Saeid Saryazdi Esmat Rashedi : Published online: 1 August 2005 © Springer-Verlag 2005
- Edge Detection of an Image based on Ant Colony Optimization Technique , Charu Gupta, Sunanda Gupta : 6 June, 2013

THANK YOU