On Edge Detection of Images Using Ant Colony Optimization and Fisher Ratio

A Thesis submitted to the department of

Electronics & Communication Engineering

of

National Institute of Technology Rourkela

in partial fulfilment of the requirements

for the degree of

Master of Technology

by

Prashant Kumar Mohanty

(Roll- 210EC3268)

Department of Electronics & Communication Engineering

National Institute of Technology Rourkela

Rourkela- 769008, India

2012

On Edge Detection of Images Using Ant Colony Optimization and Fisher Ratio

A Thesis submitted to the department of

Electronics & Communication Engineering

of

National Institute of Technology Rourkela

in partial fulfilment of the requirements

for the degree of

Master of Technology

in

Electronics & Instrumentation

by

Prashant Kumar Mohanty

(Roll- 210EC3268)

under the supervision of

Prof. Samit Ari

&

Prof. Sarat Kumar Patra

Department of Electronics & Communication Engineering

National Institute of Technology Rourkela

Rourkela- 769008, India

2012

Dedicated
to
My Parents

Department of Electronics & Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

Certificate

This is to certify that the work in the thesis entitled ***"On Edge Detection of Images Using Ant Colony Optimization and Fisher Ratio"*** by ***Prashant Kumar Mohanty*** is a record of an original research work carried out by him under our supervision and guidance in partial fulfilment of the requirements for the award of the degree of ***Master of Technology*** in ***Department of Electronics & Communication Engineering*** with specialization in ***Electronics & Instrumentation Engineering*** during session ***2011-2012*** at ***National Institute of Technology, Rourkela***. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Guide
Dr. Samit Ari
Assistant Professor
Department of ECE
NIT Rourkela-769008

Co-Guide
Dr. Sarat Kumar Patra
Professor
Department of ECE
NIT Rourkela-769008

Date :
Place : Rourkela

Acknowledgment

The Summit and success of this work would not have been possible without the help of several individuals in one way or another, who have given guidance throughout my research work.

I am expressing my sincere respect to my supervisor ***Prof. Samit Ari*** for his guidance, and constant belief in me that thrives me to overcome any difficulties that I had faced during my research work. Also I want to thank him for being my advisor at National Institute of Technology, Rourkela and helping me to know my potential as a researcher.

Next I want to express my regards to ***Prof. Sarat Kumar Patra*** for being my co-guide here at N.I.T Rourkela and helped me to overcome several problems with his precious advice.

I would like to thank *Prof. S. Meher*, *Prof. K. K. Mahapatra*, *Prof. Poonam Singh*, *Prof. S. K. Behera*, *Prof. D. P. Acharya*, *Prof. A. K. Sahoo*, *Prof. S.K. Das* for enabling me to learn my topic fundamentals as easily as possible. They are always there when we need them and I thank them from the bottom of my heart.

I would like to give deference to all the faculty and staff members of N.I.T Rourkela for their generosity towards us and helpful to us in every possible way as they could for the completion of my thesis.

I also want to give my compliance towards *Manab*, *Dipak*, *Prashant*, *Bijoy* and all other Phd. students for helping me by sharing their valuable experience and helped me in the completion of the thesis.

I also want to thank my friends *Tusar*, *Himanshu*, *Manas*, *Sankata*, *Anil* and all other fellow classmates for their valuable advice and keeping their constant faith in me to make the impossible possible.

I am especially grateful to *my parents*, *elder brother* and my room-mate *Radharaman Dalai* for their constant support,love and keeping their faith in me, helped me to grow as a personality and making me believe that I can do achieve anything in life with hard work and strong determination.

Prashant Kumar Mohanty

Abstract

Edge detection is one of the important parts of image processing. It is essentially involved in the pre-processing stage of image analysis and computer vision. It generally detects the contour of an image and thus provides important details about an image. So, it reduces the content to process for the high-level processing tasks like object recognition and image segmentation. The most important step in the edge detection, on which the success of generation of true edge map depends, lies on the determination of threshold. In this work, purpose of edge detection, inspired from Ant Colonies, is fulfilled by Ant Colony Optimisation (ACO). For the determination of threshold calculation, a novel technique of Fisher ratio (F-ratio) is used. The success of the work done is tested visually with the help of test images and empirically tested on the basis of several statistical parameter of comparison.

De-noising is the process of extracting the important features present in an image, keeping the unnecessary or unimportant information present in the form of noise out as much as possible. There are many Denoising methods that have been developed in these field, but the most trustworthy and used among them is the wavelet thresholding denoising method with hard thresholding. The proposed novel method presented in this thesis is tested on the denoised images. The Edge detected images obtained on the denoised images are showing better results than the other conventional edge detectors.

***Keywords*:** Ant Colony Optimization (ACO), Edge Detection, Fisher ratio (F-ratio), Denoising, Thresholding, Statistical evaluation

Contents

List of Figures

List of Tables

List of Abbreviations, symbol notations

ACO	Ant Colony Optimization
F-ratio	Fisher Ratio
τ	Pheromone Concentration
$\mathbb{NP}$	Non-Deterministic Polynomial-Time
η	Heuristic information
TSP	Travelling Salesman Problem
φ	Pheromone Decay Co-efficient
σ	Standard Deviation
α	Pheromone influencer
β	Heuristic influencer
ρ	Evaporation Rate
FOM	Figure-of-Merit
BDM	Baddeley's Delta Metric

Chapter 1

Introduction

Chapter 1

Introduction

1.1 Introduction

Ant colonies, besides the fact that they are simple and small by nature, are distributed system that is able to perform a highly structured social organisation. This happens as they are capable to perform many complex tasks which is far exceeding the individual capabilities of single ant.The ant algorithms take basic feature from real world ants and are helpful in the design of novel algorithms for the development of optimization and distributed systems [1]. The self-organising principles of the real world ants which are the basis of the highly coordinated behaviour can be researched further to develop some algorithms related to computational problems. Some of these features are foraging, division of labour, brood sorting and cooperative transport. The underlying nature behind all these activities is a form of indirect communication [2] known as *stigmergy*, which happens because of modification of the environment. What is happening here is that the foraging ants deposit some type of chemical on the ground and other ants because of this increasing probability follow the same path. Researchers have tried to implement this stigmergy [3] in the artificial ants to coordinate the societies of artificial ants [1].

1.2 Foraging Behaviour of Ants

Through some biologist's point of view, it is quite known that the visual sensory organs of the real world ants are rudimentary by nature and in some cases they are completely blind. A in depth research in the ants behaviour shows that the large part of communication, individual or between individuals by ants is quite done by the use of chemicals produced by the ants, known as pheromone.Foraging behaviour of ant species is also based on the indirect communication possibly done by pheromones. While having a walking from the food sources to the nest or vice-versa, the ants are depositing pheromone on the ground, forming in this way, a pheromone trail [3]. By sensing the path for any possible pheromone concentration, they choose paths probabilistically in the favour of any strong pheromone's concentration [1].

1.3 Meta-Heuristic

Meta-heuristic [4] can be related as: meta menas "beyond" and Heuristic means "to find" [3]. Also it is known as modern heuristics [5]. A disadvantage of single run algorithm like constructive method and iterative improvement is that the constructive method generates a very limited number of solutions and in case of iterative method the systems stops for every less local optimal values. Meta-heuristic [3] is a set of algorithmic concepts that define heuristic methods applicable to a wide set of different problems. It is a way of achieving high quality solutions in a local search space which is guided towards a problem specific heuristic but in truly is a general purpose heuristic method. The use of meta-heuristic has significantly the ability of finding the high quality solutions to hard, practically relevant combinatorial optimization(CO) problem in a reasonable time. Some of the CO problems are Travelling Salesman Problem and Protein Folding [3]. Metaheuristics is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide sort of problems. The use of these metaheuristic made it possible to solve quality solutions to hard $\mathbb{NP}$ problems in a reasonable time. ACO metaheuristic has been developed from the very behaviour of real life ants. ACO metaheuristic has been proposed as a common framework for the existing applications and algorithmic approaches of a variety of ant colony algorithms [1]. Several types of meta-heuristic related algorithms are Evolutionary computation, tabu search [3].

1.3.1 Meta-heuristic of Ant Colony Optimization (ACO)

Metaheuristic of ACO started by Dorigo and his colleagues in 1999 [6] Ant Colony optimization is a type of meta-heuristic in which a colony of artificial ants cooperate in finding good solutions to difficult optimization (discrete) problems. The way to handle the problems is that the computational resources are distributed between the relatively simple agents i.e., artificial ants which communicate indirectly by stigmergy with the help of modification of environment. The good solutions are the emergent property of the algorithm. There are some basic differences between a real life ants and artificial ants like [3]

- Asynchronous movement is seen in real ants and artificial ants are synchronous.
- The foraging behaviour of real ants are based on some form of implicit solution.

ACO algorithms may be used to solve both static and dynamic combinatorial Optimization problems. The static problems are those problems where the characteristics of the problem when defined once, doesn't change during the solving process of the problem e.g., TSP. The dynamic problem is the problem where the problem characteristics are based on the underlying principles which are dynamic by nature i.e., they change during the run time and so need to be adapted on-line to the changing environment., the data packets in the network routing problems. Artificial ants in ACO are adding opportunely defined solution components to an already made partial solution under construction in the way of stochastic constructive procedure. Therefore, the ACO can be applied to any combinatorial optimization for which a constructive heuristic can be defined.Although it means that any combinatorial optimization problem can be solved by the ACO metaheuristic but the real problem is that the mapping of a particular problem to the format that the artificial ants can use to build solutions [1].

1.3.2 NP Problems

There is a class of problems known as combinatorial optimization [7] which are which are often easy to state but difficult to solve. Many of the on-going problems are $\mathbb{NP}$. hard i.e., there is strong belief that these type of algorithms cannot be solved to optimality within a polynomial bounded computation time. So, for solving the practical problems, one often need some type of approximation which led to return near-optimal solutions in a relatively short time. Algorithms of this type are called heuristics [1].

1.3.3 Combinatorial optimization Description

Combinatorial optimization problems involve finding values for discrete variables such that the optimal solution with respect to a given objective function is found. Many problems of practical and theoretical importance are of combinatorial nature e.g., shortest path problems. And some real world problems like delivering goods to customers optimal time, optimal employees allotted to a particular task, etc. A combinatorial optimization can be either maximization or minimization problem which is associated with a lot of problem instances. The problem is defined in generic way i.e., general question with several parameters with unspecified values but the term instance refers to a problem with parameters given a particular specified values. The Travelling Salesman problem is the general problem of finding a minimum cost Hamiltonian circuit in a weighed graph but the TSP instance has specified number of nodes and specified arc weighs. The instance of a combinatorial optimization problem is specified in concise mathematical problem [1].

1.3.4 Complexity of ACO met-heuristic

A straight forward approach to this type of problem is to the exhaustive search and that is searching for all the possible solutions and chooses the best among those. But in some cases it becomes infeasible as the number of possible solutions for these problems increase exponentially with the instance size n, where n being the size of the digits needed for encoding of the instance. But in case of some problems, looking deeply into the problem finds some alternative way to find the optimal solution much quicker than the exhaustive search does. But sometimes exhaustive search proved to be better. Worst case complexity is the way to find how difficult it is to find the optimal solution for attacking any particular combinatorial optimization problem [1].

1.3.5 Study of Worst Case Complexity

It can be explained as follows: a combinatorial optimization can be said to have worst case complexity if the best algorithm for that particular algorithm solves the all possible instances of having size n in a computation time bounded from above by *const. (n).* We can also say that the problem is defined in a polynomial time if the maximum amount of computing time required for solving any possible instances of size n is bounded by the above by a polynomial of size n. Although some of the combinatorial problems seem to find optimal solution in a polynomial time, but for majority of the problems no polynomial bound solution is possible for the worst case solution. For these problems the run time increases exponentially with the instances size 'n',so also the time required to find the optimal solution. An important concept that can be taken for classifying these problems is that to see the $\mathbb{NP}$-completeness. The theory helpful to classify the combinatorial problems into two categories: those that is tractable and intractable. Tractable problems are those that are solved in polynomial time and those that are not. The problems that are discussed above are related to search problems. The theory of *NP*-completeness distinguishes between two classes of problems: P and *NP*. Problem is $\mathbb{NP}$ hard if every other problem in $\mathbb{NP}$ be transformed to it by a polynomial time reduction. Therefore $\mathbb{NP}$ hard problem is as hard as any other problems in $\mathbb{NP}$. However hard problems dont necessarily belong to $\mathbb{NP}$ .A $\mathbb{NP}$ hard problem that is in $\mathbb{NP}$ is said to be $\mathbb{NP}$ complete. Therefore, the $\mathbb{NP}$-complete problems are the hardest problem in $\mathbb{NP}$ [1].

1.4 Behaviour of Ants

ACO algorithms are the construction procedures related to optimization building solutions through the movement of artificial ants with the help of construction graph $G_C = (C, L)$ as set L connected with the components C. The problem are dealt with ants heuristic. The solution is made with the mixture of possible and non-possible solutions, together they are helpful for the formation of the complete algorithm. $c_i \in C$, $l_{ij} \in L$, are having their respective trail of pheromone τ and heuristic's value η. The trail of pheromone helps the remaining ants for the search process information and about the heuristic status.

1.5 Edge Detection

Edge detection is fundamentally important for image analysis like segmentation, registration, and identification of scene's objects [8]. It is the most used form for detecting the useful discontinuities in gray level image [9].

1.5.1 Background

An edge can be defined as a group of connected pixels lying between boundaries of two regions. Edge can also be defined as in binary images as the black pixels with one nearest white neighbour [8]. An Edge is a local concept but the boundary is a global concept. An ideal edge is a group of pixels located at an orthogonal step transition in gray level. Blurry edges are also acquired by the factors like problems or imperfections happened at the time during of optics , sampling and image acquisition systems. So, edges can be closely seen as having a profile as that of ramp-like profile. The ramp's slope is related to the degree of blurriness inverse proportionally. The thickness of the edge is the length of the ramp. Blurred edges are thick and sharp edges are thin. It is well observed that the first derivative it is positive along the ramp, zero where the intensity level is constant and it is constant along the ramp. Also it can be observed in the 2^{nd} derivative that it is positive along the dark side of the edge and negative along the light side of the edge. Also it is zero along the ramp and outside the ramp. So, From the derivative aspect, of the edge, it is concluded that the first derivative is giving the indication or presence of the edge at a point or not and 2^{nd} derivative is providing the details about that whether the edge is present on the lighter side or dark side [9].

Two additional properties can be deduced from the 2^{nd} derivative:

- First is the fact that the it provides more than one value for an edge .
- Also it is showing a *zero-crossing property* which is nothing but an imaginary line connecting between the extreme positive and extreme negative values along the ramp .i.e., it shows the zero along the middle of the edge. Thus it is helpful in the detection or knowing location of the centres of the thick edges [9].

It is decided that to be classified as a meaningful edge point, the gray level transition has to be stronger than the background at that point. Being dealing with the local domain, the choice of decision for taking this meaningfulness is to select the threshold. So, we are taking a point in an image as an edge point if the 2D 1^{st} order derivative is greater than the specified given threshold. A set of these types of points satisfying these predefined criteria when connected together forms an edge. Edge segment can be used in place of edge if the length of the edge is short in terms of image. Edges location can be defined as the zero crossing of its second derivative. First order derivative can be defined as the gradient of an image. Second order derivative can be taken as the laplacian of the image [9].

The gradient operators are also called as masks in digital images which calculate finite differential approximations of either horizontal or vertical directions. The Sobel, Prewitt etc. are computing the local sums of horizontal and vertical directions. Also these operators reduces noise [8].

1.5.2 Gradient operators

First-order derivatives [9] of an image are approximated as the gradient of an image. The gradient at location (x, y) of an image $f(x, y)$ is defined as:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{1.5.1}$$

With the help of vector analysis it can be observed that the gradient vector is directing in the direction of maximum rate of change off at (x, y) coordinates.
Magnitude of this gradient is :

$$\nabla f = mag(\nabla f) = \left[G_x^2 + G_y^2\right]^{1/2} \tag{1.5.2}$$

The quantity ∇f is known as the gradient of a vector.
The direction of the gradient is given as:

$$\alpha\,(x, y) = \tan^{-1}\left(\frac{G_x}{G_y}\right) \tag{1.5.3}$$

Here the angle is measured with reference to x-axis. The direction of the edge at any point is perpendicular to the direction of the gradient at that point. The computation of the gradient of an image is calculated by the finding the partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$
In a 2D image the gradient are approximated as [10]:
$G_x = f\,(i+1, j) - f(i, j)$ and $G_x = f\,(i, j+1) - f(i, j)$
Two masks are required to find the gradient: one along the X direction and one along the Y direction. The vector sum of these two gradients are assumed to be taken as the magnitude of the gradient and the angle represents the gradient angle.

Another way to find the gradient is to convolving the image with a set of eight gradient templates in which each template represents the gradient in a particular direction.

1.5.3 Detection of Ideal Edge Detection

The expectation from an ideal edge detector [10] is that any true edge point present in the image should not be messed and also the erroneously detection of any other edge point as edge should be reduced as much as possible. These two requirements are often having a trade off each other. The selection of a proper optimum threshold point is a minimum requirement of any edge detector. The threshold value should not be low as it can lead to the detection of noise as edges and also the threshold value as high causes some true edge points undetected. The SNR is improved when true edges are detected and false edges are avoided. The removal of false responses reduces the corrupted edges happened due to noise.

The performance measure of edge detection [8] operations are as follow

- The results can be compared visually as the eyes are behaving and acting like some sort of edge detection.
- Also the edge detection rate can be evaluated.
- The Figure of merit can also be evaluated.

The canny edge detector is a combination of many steps. The steps are as follows [10]

At first the convolution of the image with a smoothing filter (Gaussian) having standard deviation σ .Also this step is followed by a gradient computation on the resultant smoothed image.

Non-Maxima-Suppression: The operation of the suppression suppresses or thins the thick ridges that are wider than a pixel.

Double Thresholding: The gradient images obtained after the application of non-maximal suppression may contain some false edge points. For the removal of these false points, the application of thresholding is applied over it. So, all depends upon the detection of proper threshold value. So, removal the confusion of low and high thresholding causes the selection of the two threshold points where T2 $\approx 1.5T1$. This process provides the complete contour formed by the true edges of the image.

1.5.4 Limitation of Edge based Segmentation

The limitations of Edge Based Segmentation are as follows [10]

1. Presence of spurious edges and gaps, causing limitation on the applicability.

2. Ignoring the model based information embedded in an image.

3. Ignoring of the higher order meaningfully organization present in an image.

4. The edge linking process causes discontinuities and gaps in the image.

5. Dependency on some arbitrary interpolation for completing boundary gaps.

6. Difficulty in classification of spurious edges.

1.6 Summary

This chapter provides the details about the fundamentals of Ant Colony Optimization(ACO) and Edge detection. It also mentioned about the underlying concept behind the ACO, its development, application and implementations in various problems of computational world. It is helpful,especially in the areas of the $\mathbb{NP}$ related problems. These details will be quite useful for the understanding of the later chapters.

Chapter 2

Edge Detection of Clean Images

Chapter 2

Edge Detection of Clean Images

2.1 Background Details

Edge detection refers to the process of extracting edges from the image where there are sudden changes or discontinuities. These extracted edge points from an image provides an insight into the important details in the field of image analysis and machine vision [11]. It acts as a preprocessing step for feature extraction and object recognition [12]. Various techniques are reported in the literature like Sobel [9], Prewitt [13], Roberts [14], Log [15] and Canny [16] detection techniques. However, most of the existing detection techniques use a huge search space for the image edge detection [17]. Therefore, without optimization the edge detection task is memory and time consuming. What is happening in this

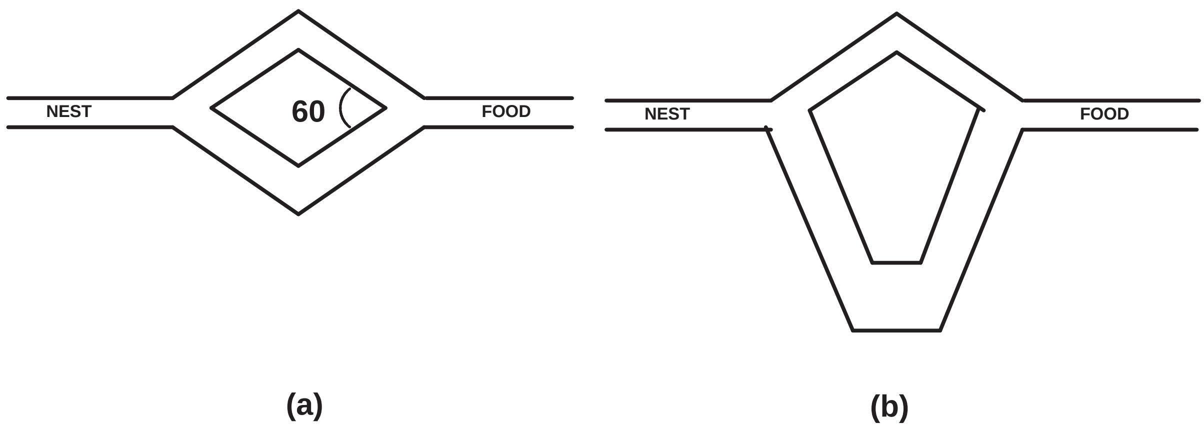

Figure 2.1: DOUBLE BRIDGE EXPERIMENT *[1]*

experiment [2] is that the several cases of ant colony optimization have been tested here:

- First case is with the cases where ants are made to walk over two branches of equal length. Here it is observed that in the initial phase, both paths are having same number of ants and after that one path gets more number of ants than the other. The reason behind this phenomenon is that ants at the very outset select both paths equally. But after some time due to random nature one path gets more preference than the other. As ants are leaving pheromone trails behind, so the path selected by the more number of ants gets more amount of pheromone which further reinforces the selection of that path. This nature of natural phenomenon is in another terms can be described as auto-catalytic or feedback process [1]. This is also explaining the stigmergy i.e., the indirect mode of communication happened due to the modifications in the environment.

- Second experiment leads to the possibility that the one path is double the longer than the other path [18]. At the very start, selection is equal for both paths but after some time what is happening is that ants are selecting the path shorter among the two. The main reason behind is that ants choosing the shorter path are coming back from to their nest from food sources quickly. What is affecting their decision is that the shorter path is containing more amount of pheromone, so lead to the selection of that path more due to the auto-catalytic process, as described earlier. The effect of randomness is greatly reduced here and stigmergy, auto-catalysis, and differential path length are coming into action. Despite the fact that the shorter path is present, still some ants choose the longer path due to the path exploration [1].

- A third case is also studied where a shorter path is added after a long time , and what happened here is that the ants are still attached with longer path due to the auto-catalytic nature and slow evaporation of pheromone trails.

2.1.1 Nature of Artificial Ants

With the help of the double bridge experiment, it is quite known that ants have a built-in optimization capability: i.e., they are able to select the shortest path between two points in their environment depending upon the probabilistically rules based on the local information. This concept is researched further and tried to be implemented on the case of the artificial ants. These artificial ants are further used for the purpose of getting shorter paths on the connected graph. But the problem that arises in this case is that the ants while building the solution may generate loops. Removal of the forward pheromone update is the solution to this problem, but this will lead to the disturbance of the system. Even the double bridge experiment will cease to work.

So, the properties of the artificial ants are prepared in such a way that they are able to solve minimum cost path problem on generic graphs. Those are that they are given some amount of limited memory in which are able to remember the paths traversed and also the cost of the links associated with it. With the help of memory, they are able to build solutions for the minimum cost path problems. These behaviours are [1] (1) Without forward pheromone updating, probabilistic construction of solutions which is biased by pheromone trails; (2) backward path of deterministic nature with loop elimination and with pheromone updating; and (3) evaluation of the generated solutions quality and use that for the determination of the quantity of pheromone to deposit [1]. Also it is observed

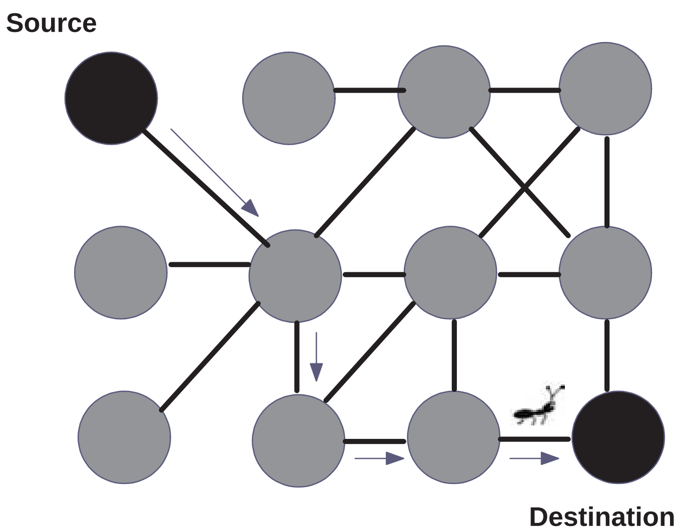

Figure 2.2: THE MOVEMENT OF ANT ON THE IMAGE FROM SOURCE TO DESTINATION *[1]*

that by taking into account the pheromone evaporation, the performance of the ants are greatly improved. The first algorithm we are going to discuss is Simple ACO(S-ACO). It acts a instructive way to explain the basic steps hidden behind the concept of ACO algorithms [1].

2.1.2 Solution Construction through probabilistic movement

S-ACO is completely operating in two modes: backward and forward. Forward mode becomes functional when travelling from nest to the food source and in reverse mode while travelling from food source to the nest. In S-ACO, forward ants while travelling from node i to node j undergo decision probabilistically. The node j is in the neighbourhood of the node i. Given a graph $G = (N, A)$, two nodes i,j belong to N are neighbourhood. The probabilistic dependency is based on the pheromone concentration trails left over earlier by the other ants. Due to the reason that the forward ants are not leave any pheromone on the path while moving, it do happen in the cause of avoiding of the formation of loops [1].

2.1.3 Updation of pheromone through deterministic backward movement

These ants while travelling upto the destination node tries to keep the information in the memory. Also the improvement in the system is seen with the elimination of loop. While moving backward, they try to leave pheromone on the path they traverse [1].

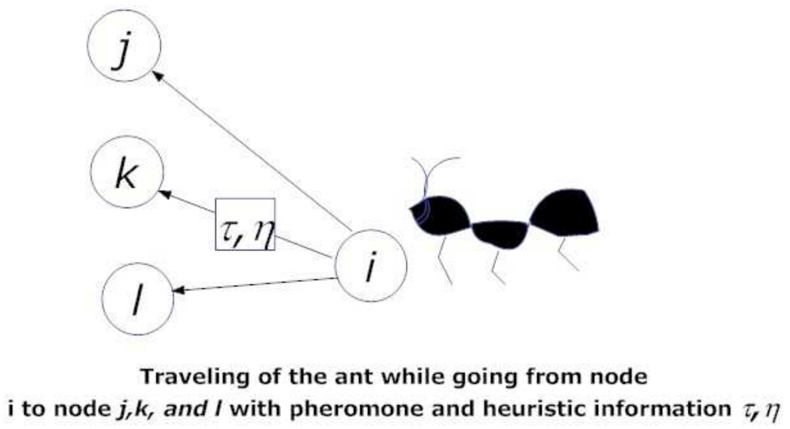

Figure 2.3: The probabistic decision movement of ants *[1]*

2.1.4 Solution quality determination by pheromone updates

In S-ACO, ants try to memorize the path they travel and also try to calculate the cost of the various paths traversed. Ants try to evaluate the cost and depending upon it, it leaves the pheromone on the path in the backward mode. The pheromone update being related to the generated solution quality helps the future ants to move towards a better solution. Increasing pheromone concentration on the shorter path made the decision of

the algorithms to bias towards the best solutions [1].
Real ants leave pheromone trails on the path searched and this evaporates at a constant rate. But in the case of artificial ants, this pheromone rate can be managed to reduce at a prefixed rate by defining pheromone evaporation rule. This pheromone decay rate reduces the effect of the chances of the build-up of the poor quality solutions in the early stages of the search [1].

2.2 Simple-ACO

The understanding of the simple ACO [1] is required to move further for the proceeding of our real topic of interest. There are some things that need to be understood before proceeding further:

- Ants, with the help of decision policy, try to build solution from the starting node. At each node, ant collects all the local information on that node, including the information on the outgoing arc is sensed and decision is taken in a stochastic way. At the very start, all the nodes are initialized to a constant amount of pheromone [1].

- When present at a node i, ant k decide to move on to the node j with a probability, according to the pheromone trails τ_{ij}. An ant repeatedly changes nodes using this decision policy, until it actually reaches the destination node. The time required by the ants to reach the destination node varies from ant to ant depending upon the decision or solution.

- While reaching the destination node, ants travelled back by switching to backward mode from forward mode and in this process retraces back the same path travelled in forward mode but keeping in the fact of loop elimination. The problem of loop elimination is development of self-reinforcing loops.

- Pheromone trail evaporation is helpful for the exploration mechanism which prevents the convergence of the algorithm towards a sub-optimal solution. The pheromone evaporation doesn't affect the functioning of real ants but a importance in the artificial ants as it favours exploration for new paths during the whole search process. The importance of pheromone evaporation in the lives of artificial ants might be due to fact that optimization problems solved by the artificial ants are far more complex than solved by the real life ants.

Forgetting the errors done in the past or of poor choices allows continuous improvement in the learned problem structure. Also the pheromone evaporation also does the function of bounding for maximum achievable solutions. We call a combination of pheromone evaporation, pheromone deposit and ant's movement as a complete cycle [1].

2.3 General Steps of ACO Meta-Heuristic

ACO algorithm is in general interplay of three procedures: [3] Construction of Ant Solutions, Updation of Pheromones, and Daemonic Actions.

Construction of Ant Solutions : This step is primarily involved with the concurrent and asynchronously movement of ants over the complete problem space in adjacent states and visiting the neighbour nodes of the construction graph G_C. The movement is quite influenced by the effect of pheromone trails and heuristic information and the incrementally build-up of the solutions is being done.

Updation of Pheromones : It is the way the pheromone matrix is updating. Either the concentration of pheromone on the matrix will increase due to ants deposition of pheromone or the concentration will decrease due to the pheromone evaporation. Practically, the usage is that the more concentration of pheromone in the matrix is due to the components or connections being favoured by more ants or cause may be because that the solution produced by single ant is building a good solution for future ants. Also the pheromone evaporation is helpful with the forgetting because of this the rapid convergence towards a sub-optimal solution is avoided and also it helps in the favouring of the exploration of new paths in the defined search space.

Daemonic Actions : These are the actions taken by the centralized team of the collection of ants which is not performed by single ants. These daemonic actions start the local optimization procedure, collect and analyse the global information which is used to decide the future decision of deposition of pheromone on the connections to bias the search space from a non-local perspective. Practically, what is happening is that the daemon observed all the path covered by all the ants in the colony and allow the ants that have built the best solutions to deposit additional pheromone on the components/connections [1].

General steps of ACO [19] is summarized as follows:

1. Setting parameter and initialization of ACO
2. **While** termination condition not met **do**
3. Construction of Ant Solutions
4. Applying local search (optional)
5. Updation of pheromones
6. **EndWhile**

The ACO metaheuristic is available in pseudo code. The main ACO metaheuristic constructs three steps for the ScheduleActivities. Those are (1) the constructions of ant solutions ;(2) the updation of pheromones;(3) the daemonic actions. These three steps are independent and can operate in parallel or not, can act in a synchronized way or not. So, the designer can use his ideas to implement the three steps in any of his desired way keeping in view of the considered problem [1].

2.3.1 Applications of ACO

The use of ACO can be applied to the combinatorial optimization problem. The various applications of ACO are as follows :

1. Travelling Salesman Problem (TSP) [1]
2. Evolutionary Computation (EC) [1]

2.3.2 Ant Colony optimizations algorithms for TSP

Travelling Salesman Problem (TSP) is the first instance of the problem definition where the first testing of the all ACO algorithms was being done. There are various reasons behind this: It is an important NP-hard optimization problem that is taken for various problems. ACO algorithms can be easily applied to the TSP. A good performance improvement on the application on the TSP is taken as a proof for the performance improvement in all ACO algorithms.

TSP implemented in ACO : The first ACO algorithm, Ant system [3], was introduced using the TSP as an example application. The importance of the AS lies in the fact that it acts as an inspiration for the development of several other ACO algorithms like Elitist AS [20], Rank based AS [21] and Max-Min AS [22]. The important difference between the AS and its other successors is that in the way the pheromone updates are performed and also some details in the management of the pheromone trails. The most important one we are using is the Ant Colony System (ACS) [23]. The ACS is different from AS in three main points. First the ACS uses the search experience in far better use than the AS does with the more aggressive action rule. Second, pheromone deposition and pheromone evaporation is taking place on the arcs that come out in the best-so-far tour [1]. Third, while each time an ant chooses an arc (i, j) while moving from point i to point j is remove some pheromone from that arc so to enhances the exploration on the new path on the search space [1].

Construction of Tour : In ACS, when located on the node i and moving towards node j by ant k is chosen according to a pseudo-random proportional rule [3].

$$j = \begin{cases} \arg\max_{l \in N_i^k} \left\{ \tau_{il} \; [\eta_{il}]^{\beta} \right\}, if \; q \leq q_0 \\ J \qquad\qquad\qquad otherwise \end{cases} \tag{2.3.1}$$

Where q is a variable random in nature and distributed uniformly in $[0, 1]$, q_0 $0 \leq q_0 \leq 1$ acts as a parameter and J a variable randomly selected depending on probability distribution with $\alpha = 1$. In other words, it is presented in a particular way that the ants are biased towards the arc influenced with the pheromone trail and heuristic information and it searches for the new path or tries exploration with the probability 1-q_0 [1].

Update of the Global Pheromone Trail : In ACS only the ant (best so far) is allowed to add pheromone after each iteration. The ACS update is done as follows: $\tau_{ij} = (1 - \rho)\, \tau_{ij} + \rho \Delta\tau_{ij}^{bs}, \; \forall (i, j) \in T^{bs}$ where $\Delta\tau_{ij}^{bs} = {}^{1}\!/_{C^{bs}}$ Important thing to note is that the update of pheromone trail which is applying only to the arcs of T^{bs} not to all the arcs in AS. This reduces the iteration from $O(n^2)$ to $O(n)$, "n" being the size of the instance involved and ρ is the pheromone evaporation. It is behaving different from AS in the view that the deposited pheromone is discounted by the factor ρ, so new pheromone trail being is the weighted average of the old pheromone value and the amount of pheromone added [1].

Local update of the pheromone trail : In addition to the global pheromone update, the ACS also uses local pheromone update immediately after crossed an arc (i, j) during the tour construction. $\tau_{ij} \leftarrow (1 - \zeta)\,\tau_{ij} + \zeta\,\tau_0$ where $\zeta, 0 < \zeta < 1$ and τ_0 are two parameters. The value of τ_0 is assumed as the initial value for the entire pheromone trail matrix. The effect of the local pheromone update is that the effect of the nodes already visited by ants gets reduced and this also encourages the exploration for the new paths and avoids the stagnation behaviour. Also it is important to consider here that in AS variants the construct operation by ants being done parallel or sequentially doesn't count but in case of ACS it counts due to the presence of local pheromone update rule. There exists a relationship between MMAX and ACS. Both of them are using pheromone trail limits but in case of former these are explicit and in the latter it is implicit. The pheromone trail value is managed in such a way it never gets reduced below τ_0 [1].

2.4 General Behaviour of ACO Algorithms

Artificial ants iterates tour construction loop which is biased with the artificial pheromone trails and the heuristic information. The main mechanism at work in ACO which is reason behind the discovery of good tours is the positive feedback done through the pheromone update by the ants. The shorter the ants tour, the more amount of pheromone is deposited by ants. This in turn caused the ants to select the same arcs in the subsequent iterations of the algorithm. The emergence of arcs with high pheromone values are further reinforced by the mechanism of pheromone evaporation that avoids an unlimited amount of pheromone and decrease the pheromone content from the arcs that rarely receive additional pheromone [1].

Ant colony optimization (ACO) [1] is an evolutionary computation based algorithm inspired from natural phenomenon of foraging. In this optimization method, ants follows an intelligent way of information transfer about the food sources to their colony with the help of some natural chemical secretion, also known as pheromone [1]. The Ants left behind these pheromone trails on their marching path. Also these pheromone trails evaporate over time. Ants marched over these paths over and over. The more time it takes by the ants to pass through these paths, the more time the pheromone trails evaporate. Therefore, shortest path receive greater pheromone concentration and more number of ants travel over it. This creates a positive feedback mechanism which causes ants to follow the shorter path [24]. The first ACO approach, known as *ant system*, was developed by Dorigo *et. al* [25]. Since then, several ACO approaches have been reported [19], like as *Max-Min Ant system* [22] and *Ant Colony System* [23].

The conventional edge detectors shows a lot of loopholes [24] :(1) provides poor result when object and background are not clear; (2) incapable to detect the edge in rigid environment; and (3) calculated threshold is sensitive to noise. ACO based technique overcomes all these demerits of traditional methods using inherent capability of parallelization.

This work propose a new algorithm in the field of edge detection using a combination of ACO and F-ratio through extracting edge information from the pheromone matrix [1]. The proposed approach drives the artificial ants on the image pixels, influenced by the local intensity variation. The direction of ants movement is determined using a probabilistic technique [19] of direction selection. The ants move in such a way that maximizes the intensity of pheromone on the path depending upon image intensity variation. In this work, F-ratio technique [26], which adaptively calculate the threshold value from updated pheromone matrix, is proposed to determine the optimum threshold value. This optimum threshold value is utilized to extract binary edge map from pheromone matrix. The proposed technique provides a state-of-the-art advantage of both ACO and F-ratio. The proposed algorithm is tested on the images of *Cameraman, Lena, Coins, Peppers and House*. The experimental results show that the proposed method performs better in terms of statistical evaluation and visual comparison as compared to earlier reported technique.

2.5 Ant Colony Optimization

The existence of the idea of Ant Colony Optimization (ACO) is based on a biological inspiration. It is related to the concept of stigmergy [1], proposed by Grassé. Stigmergy is the natural adaptation that differentiates ACO from other systems. It is an indirect mode of communication in which ants being distant from each other tries to contact with each other through producing and reacting with the stimuli. In this way they deposit a chemical like substance called pheromone on the ground while foraging for food. Other ants of the same colony when cross through this path reacts in a particular way that made it easier for the whole colony in the searching process of food and saves time. In ACO, the artificial ants following the artificial intelligence concept, simulates the natural environment behaviour and applied it in the combinatorial optimization [1] problem like Travelling Salesman Problem [23, 25, 27].

Different ACO optimization algorithms had been proposed earlier. First ACO, as mentioned earlier was known as *Ant System* [25]. Several ACO algorithms have been developed after that such as *Max-Min Ant System*, and *Ant Colony System*. The proposed ACO is presented in earlier chapter.

The ACO algorithm while being implemented on the image undergo some changes. The solution space for ants now is the 2D image and the artificial ants [1] are now made to move over the image. Therefore, the artificial ants, simulating the real ants, leave pheromone on the nodes or image pixels. The edges of the image becomes the food for the ants. Therefore, in this way the ants develop a pheromone matrix. The decision of the path taken is also influenced by the local intensity values [24]. The parameters taken here are total ants K and τ_{init}. τ_{init} is the starting initial value of pheromone matrix. Construction of ant solution gets possible through the local search on the solution space i.e., the image matrix. Ants decide to move from node i to another j through the probabilistic action rule [19] which is as follows

$$p_{i,j}^{(n)} = \frac{\left(\tau_{i,j}^{(n-1)}\right)^{\alpha} (\eta_{i,j})^{\beta}}{\sum_{j\in\Omega_i} \left(\tau_{i,j}^{(n-1)}\right)^{\alpha} (\eta_{i,j})^{\beta}}, if\ j \in \Omega_i \tag{2.5.1}$$

where $\left(\tau_{i,j}^{(n-1)}\right)$ is pheromone information in the previous loop while moving from node i to node j; Ω_i is the neighborhood nodes for the recent ant given that it is in the node i; the constants α and β influences the pheromone information and heuristic information, respectively. ACO is an iterative algorithm and two update operation are included in it [24]. The updates are performed over the pheromone matrix. First update is done by all ants after each construction step (local pheromone update) to the last edge known to be traversed [19] and second one after all ants have completed their one cycle of iteration by only one ant (offline pheromone update) [19].

The local pheromone update [19] is performed as followed by the equation:

$$\tau_{ij} = (1 - \varphi).\tau_{ij} + \varphi.\tau_0 \tag{2.5.2}$$

where $\varphi \in (0, 1]$ is the pheromone decay coefficient and τ_0 is the pheromone initial values. The local update is performed to enable the search process more easy for next iterating ants. The offline pheromone update [19] is performed by the equation as follows:

$$\tau_{i,j}^{(n-1)} = \begin{cases} (1-\ \rho)\,.\ \tau_{i,j}^{(n-1)}\ +\ \rho.\,\Delta_{i,j}^{(k)}, & if\,(i,j)\,belongs\,to \\ & the\,best\,tour; \\ \tau_{i,j}^{(n-1)} & else \end{cases} \tag{2.5.3}$$

Pheromone evaporation causes the ants to search for some new paths and in this way provides opportunity to discover a new shorter path in the unexplored area during the whole search process. This is called "path exploration". Also it avoids the system a quick convergence towards a suboptimal path [1]. Exploitation is the process of attaining the maximum probability path [28]. The exploitation of the learned experience is applied during solution construction with the help of pseudo-random proportion rule of ACS [1]. ACO exploits the advantage [28] of exploitation and exploration to get the solution through iteration of the optimal search path. This reduces the premature convergence of the system [29]. Also ACO is versatile in nature as algorithm can be used for various versions of the same problem. Also, it is robust due to its applicability to the other problems in the same field with minimum changes [28].

ACO has been widely applicable to NP-hard (non-deterministic polynomial-time) problems i.e., those problems for which the best of the known algorithms seem to find optimal solution with exponential time worst case complexity [19].

2.6 Proposed Methodology

The proposed image edge detection based on ACO combined with F-ratio is applied on a 2D image to generate a pheromone matrix. Each entry of that pheromone matrix represents the intensity change in the original image influenced by the edge location. A heuristic matrix [30] is also giving guidance to the algorithm to attain the optimum point easily and in less computation time.

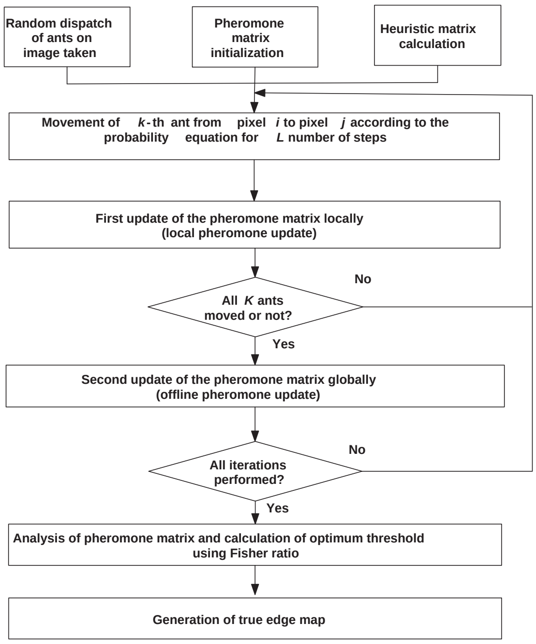

Figure 2.4: *Flow chart of Proposed algorithm using ACO technique and Fisher Ratio*

The following algorithm applies K number of ants on the image I randomly and then it undergoes N number of iterations and update operation are being done on the pheromone matrix and at last decision is taken on that pheromone matrix to determine the threshold to calculate the edge map. This proposed algorithm is basically divided into four phases: Initialisation, Construction, Update, and Decision phase.

2.6.1 Initialization phase

Input image I of dimension $M \times N$ is taken as input which works as a solution space for the artificial ants. The K number of ants are moved over the whole image such that the every pixel of the image is covered by an ant. A pheromone matrix of dimension, same as that of the image, is taken and initialized to a very small value τ_{init} . A Heuristic matrix $\eta_{i,j}$ is evaluated based on the local statistics of the image which depends on *clique*. The local statistics at the pixel location (i, j) is calculated as follows [30]

$$\eta_{i,j} = \frac{V_c(I_{i,j})}{Z} \tag{2.6.1}$$

where

$$\begin{aligned} V_c(I_{i,j}) = f(&|I_{i-2,j-1} - I_{i+2,j+1}| + |I_{i-2,j+1} - I_{i+2,j-1}| \\ &+ |I_{i-1,j-2} - I_{i+1,j+2}| + |I_{i-1,j-1} - I_{i+1,j+1}| \\ &+ |I_{i-1,j} - I_{i+1,j}| + |I_{i-,j} - I_{i+1,j}| \\ &+ |I_{i-1,j+2} - I_{i-1,j-2}| + |I_{i,j-1} - I_{i,j+1}|) \end{aligned} \tag{2.6.2}$$

and $Z = \sum_{i=1:M_1} \sum_{j=1:M_2} V_c\left(I_{i,j}\right)$ is normalization factor [30] and $f(x) = \lambda x\,,\ x \geq 0$ [30].

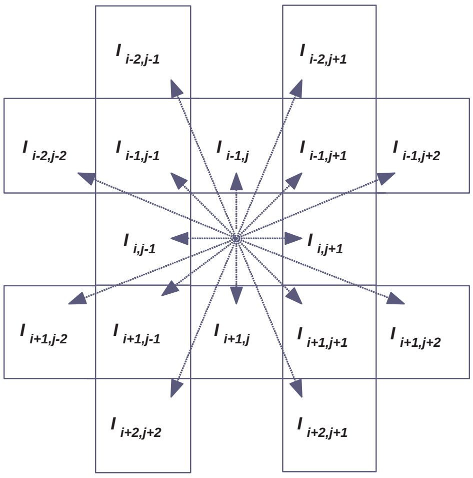

Figure 2.5: *Clique matrix [30]*

2.6.2 Construction phase

At each and every n^{th} construction step, one ant being randomly selected from K ants and it move over the image for L movement steps. Ant move from the pixel (l, m) to pixel (i, j) with a probability transition rule based on [30].

$$p_{(l,m),(i,j)}^{(n)} = \frac{\left(\tau_{i,j}^{(n-1)}\right)^{\alpha} \left(\eta_{i,j}\right)^{\beta}}{\sum_{j\in\Omega_i} \left(\tau_{i,j}^{(n-1)}\right)^{\alpha} \left(\eta_{i,j}\right)^{\beta}}, if\ j \in \Omega_i \tag{2.6.3}$$

$\tau_{i,j}^{(n-1)}$ represents the pheromone value at pixel (i, j) , Ω_i is the neighbourhood of pixel (l, m), $\eta_{i,j}$ represents value from heuristic matrix at pixel (i, j). The α and β influences the pheromone matrix and heuristic matrix respectively. Ant at pixel location (i, j) in the 8-neighbourhood in Fig 2.6 can wander in any of the 8 directions (NW, N, NE, W, E, SW, S, SE). Ants memory length is a parameter that needs some highlight. The location in ant's memory are non-admissible. Therefore, its choice is a crucial one. Small length may cause the algorithm idle whereas large length might miss the details. It is empirically chosen [32] in the interval [0.85 A, 1.15 A] where A is 40 for image of size 128×128 .

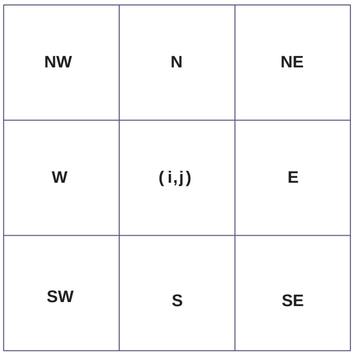

Figure 2.6: *8-connectivity neighbourhood of pixel* (i, j) *[31]*

2.6.3 Update phase

The updation is done on the pheromone matrix, first time after each ant is within n^{th} construction loop which is as follows

$$\tau_{i,j}^{(n-1)} = \begin{cases} (1-\ \rho)\,.\ \tau_{i,j}^{(n-1)} \ +\ \rho.\,\Delta_{i,j}^{(k)},\ \ if\,(i,j)\,belongs\,to \\ \qquad\qquad\qquad the\,best\,tour; \\ \tau_{i,j}^{(n-1)} \qquad\qquad else \end{cases} \tag{2.6.4}$$

and the second update process occurs after each and every ant have completed the n^{th} construction loop where

$$\tau_{ij} = (1-\varphi).\tau_{ij} + \varphi.\tau_0 \tag{2.6.5}$$

The local update broadens the search for the subsequent ants by reducing the pheromone level on the traversed edges. This way it provides an opportunity for the subsequent ants to produce necessary solutions. Therefore, the chance of repetition becomes less likely in the same iteration [24].

2.6.4 Decision phase

In this phase, the output pheromone matrix so obtained undergoes a threshold calculation procedure. This work proposes a F-ratio [26] based technique for determination of optimum threshold value which is further used for converting the resultant pheromone matrix to binary edge map. F-ratio [26] is a statistical measure in the analysis of variance for multi-cluster data. It is defined as

$$F-ratio = \frac{Variance\ of\ means\,between\ the\ clusters}{Average\ Variance\ within\ the\ clusters} \tag{2.6.6}$$

If k number of clusters are available, n_j is total number of data points for j^{th} cluster, μ_j represents mean of j^{th} cluster, $\overline{\mu}$ is the total mean, x_{ij} is i^{th} data of j^{th} cluster then above equation is represented as

$$F - ratio = \frac{\frac{1}{k}\sum_{j=1}^{k}\left(\mu_j - \overline{\mu}\right)^2}{\frac{1}{k}\sum_{j=1}^{k}\frac{1}{n_j}\sum_{i=1}^{n_j}\left(x_{ij} - \mu_j\right)^2} \tag{2.6.7}$$

If all data points is segmented into two cluster based on their value using a threshold T then F-ratio can be represented as

$$F - ratio_T = \frac{\left(\mu_{1T} - \mu_{2T}\right)^2}{2\left(v_{1T} + v_{2T}\right)} \tag{2.6.8}$$

where μ_{1T}, μ_{2T}, v_{1T}, v_{2T} are the mean and variance of the cluster 1 and cluster 2 respectively.

The proposed F-ratio based optimum threshold calculation algorithm is as follows

1. Initially, using a threshold T based on intensity level, total elements of pheromone matrix is segmented into two classes C_1 and C_2, where C_1 and C_2 consists of all data points having intensity level below and above T respectively.

2. Compute $F - ratio_T$ for all possible threshold T as in (2.6.8).

3. The optimum threshold T^* is determined as the value of T for which $F - ratio_T$ maximizes.

The calculated optimum threshold T^* is finally utilized to get the binary edge map from updated pheromone matrix.

2.7 Edge Detector

There are various operator [10] based on the single derivative like Robert operator, Sobel operator, Prewitt operator, canny operator etc.

2.7.0.1 Robert operator based edge detector

It is a simple gradient operator based on a 2×2 mask gradient operator.

$G\left[f(i,j)\right] = \left[f(i,j) - f(i+1,j+1)\right] + \left[f(i+1,j) - f(i,j+1)\right]$

The convolution kernel for Roberts's operator [10] is as follows:

1	1
-1	-1

and as its kernel is a smaller one, so it is quite sensitive to noise.

2.7.0.2 Sobel operator based edge detector

Its [10] kernel is a 3×3 mask operator [10]. The convolution masks for the Sobel based operator is as:

1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

The separate applications of these two masks is applied on the image and two gradient components G_x and G_y in the directions, horizontal and vertical, is calculated respectively.

$$\begin{aligned} G_x = \; & [f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1)] - \\ & [f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1)] \end{aligned} \tag{2.7.1}$$

and

$$\begin{aligned} G_x = \; & [f(i-1, j-1) + 2f(i, j-1) + f(i+1, j-1)] - \\ & [f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1)] \end{aligned} \tag{2.7.2}$$

The gradient magnitude is calculated as:

$$G\left[f\left(x,y\right)\right] = \sqrt{G_x^2 + G_y^2} \tag{2.7.3}$$

2.7.0.3 Prewitt operator based Edge detector

It [10] is defined by a group of eight masks, four of them are:

1	1	1
0	0	0
-1	-1	-1

0	1	1
-1	0	1
-1	-1	0

-1	0	1
-1	0	1
-1	0	1

-1	-1	0
-1	0	1
0	1	1

And other four are generated by a rotation of the above four masks by 90°.

2.7.0.4 Cannys Edge Detector

This edge detector provides good noise immunity [10] and also detects true edge points with minimum error. The optimization of canny edge detector is done by

- Maximisation of the SNR value of the gradient value of an image
- The calculation of an edge localization factor, which ensures the localization of the detected edge as accurately as possible
- Minimization of multiple values of a single edge.

2.7.1 Selection of Edge Threshold

The detection of edges is depending on [10] the comparison of the edge gradient with a threshold. The threshold value can be low when the image is free from noise in the image, so all true edges can be detected. But in case of noisy images, the low threshold can create a problem.

2.7.2 Second derivative operators

The double derivative rule states that the edge exists at the points of local maxima in the gradient values. This is resulted by the maximum in the first derivative and a zero crossing for the second derivative at the edge points. Laplacian operator is a most commonly second derivative based operator.

Laplacian operator: The gradient operator [8, 10] as presented earlier is anisotropic, i.e., they are rotation invariant. Application of isotropic operator before and after the resultant image is having no effect on the image. Laplacian operator is one such isotropic rotation invariant operator. Laplacian operator can be expressed as:

$$\begin{aligned}\tfrac{d^2f}{dx^2} = \tfrac{dG_x}{dx} = \tfrac{d[f(i,j)-f(i,j-1)]}{dx} &= \tfrac{df(i,j)}{dx} - \tfrac{df(i,j-1)}{dx} \\ &= [f(i,j+1) - f(i,j)] - [f(i,j) - f(i,j-1)] \\ &= f(i,j+1) - 2f(i,j) + f(i,j-1)\end{aligned} \tag{2.7.4}$$

0	0	0
1	-2	1
0	0	0

0	1	0
0	-2	0
0	1	0

Laplacian, being the second derivative operator has zero response to linear ramp. It has positive and negative response on the either side of it. The detection of edge points possessing local maxima in the gradient values is the principle behind the double derivative edge points.

Laplacian of Gaussian (LOG) edge detector: Laplacian operator [8,10] being susceptible to noise, Laplacian of Gaussian operator is used. It performs Gaussian smoothing, followed by the Laplacian operator. The smoothing operation reduces the noise susceptibility and as a result reduces the probability of detection of false edges. The LOG Function as defined for convolution is defined as:

$$LOG(x,y) = \frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.7.5}$$

The three dimensional plot given by the LOG function [10] looks like a Mexican hat. Increase in the sigma widens the convolution mask. The first zero crossing of the LOG function is happening around at $\sqrt{2\sigma}$

2.7.3 Image thresholding techniques

Threshold calculation from [10] the gray level image is a computationally inexpensive method. Thresholding involves the splitting up of the image into several meaningful regions. One of the earliest thresholding operations was given by Otsu [33]. The logic for threshold selection is that the gray-level for which the inter-class variance is maximum is selected as the threshold. The criterion proposed by Otsu maximises the inter-class variance of the V (k) as maximum. This method is providing more computational complexity due to calculation of the between class variance. There are various methods for threshold calculation. Some of them are [10]:

1. Bimodal thresholding
2. Multi-level thresholding
3. Entropy based thresholding

4. Wavelet Thresholding

2.8 Results and Discussions

The proposed approach has been tested on the test images *Cameraman*, *Lena*, *Coins*, *Peppers* and *House*. All the images taken are of size 128 × 128. The experimental parameter and their values are given in Table 2.1. All statistical method of comparison of edge detectors need some type of reference image as a basis for parameter comparison. However due to various limitations, it is unable to get or create a ideal reference image. Some takes help of the ground truth image [34] whose creation is a very tedious task. Therefore, we have a taken a majority image concept. The accuracy of edge detectors is determined from Relative Grading technique [35]. This majority image is created with help of other five edge detectors: Sobel, Prewitt, Robert, Canny and LoG. A pixel in a majority image claims to be an edge pixel if majority of the detectors detects a edge pixel in its neighbourhood with at least one on the center [17]. The validation of improvement of edge detectors is possible by two methods: First, it can be done by visual comparison and second, it can be done by the statistical parameter comparison. The images as shown from Fig. 2.7 - Fig 2.11 in sequence as follows: a) original image b) Majority image c) Tian *et al.*'s. method d) Proposed method. On the visual comparison front, the results show that the proposed approach is better than Tian *et al.*'s. method [30] for the images taken for experiment. The validation of the proposed algorithm is further evaluated on the basis of statistical parameter like kappa [36], Figure-of-merit (FOM) [37], Baddeley's delta metric (BDM) [38] and Hausdorff's distance [39]. The statistical parameters mentioned above are discussed in brief as follows.

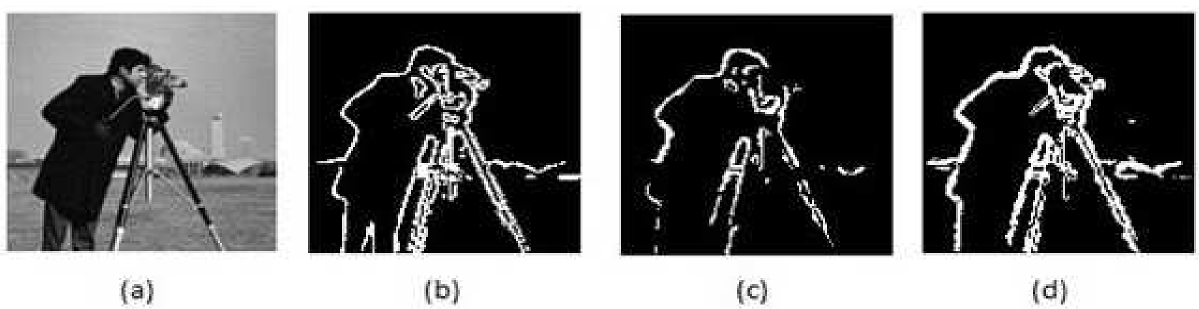

Figure 2.7: *(a) Original Cameraman image (b) Majority Image (c) Tian et al.'s method [30] (d) Proposed Method*

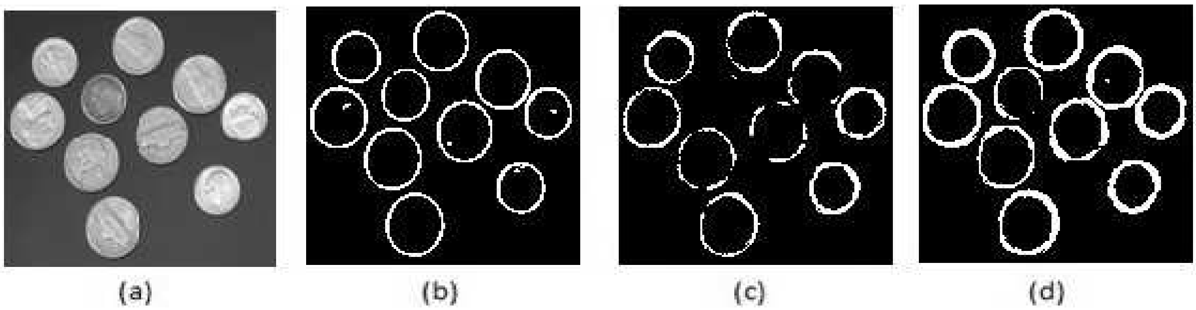

Figure 2.8: *(a) Original coins image (b) Majority Image (c) Tian et al.'s method [30] (d) Proposed Method*

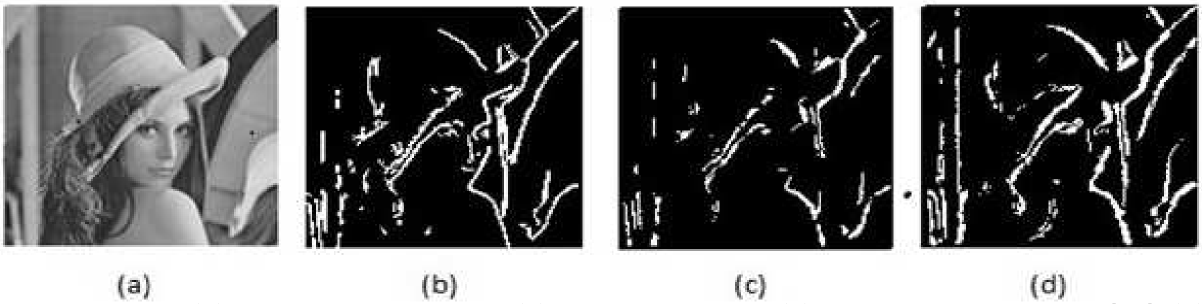

Figure 2.9: *(a) Original lena image (b) Majority Image (c) Tian et al.'s method [30] (d) Proposed Method*

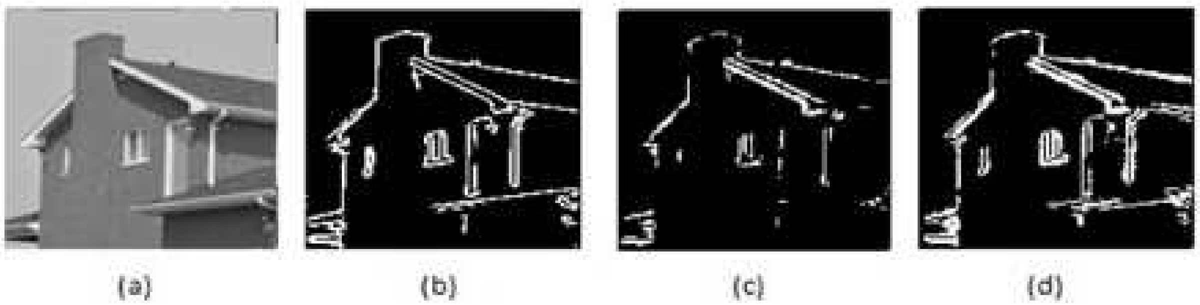

Figure 2.10: *(a) Original house image (b) Majority Image (c) Tian et al.'s method [30] (d) Proposed Method*

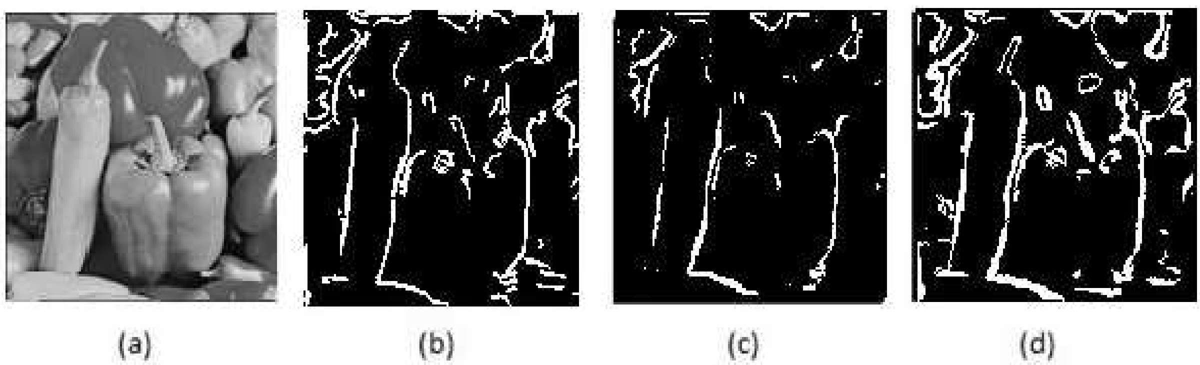

Figure 2.11: *(a) Original peppers image (b) Majority Image (c) Tian et al.'s method [30] (d) Proposed Method*

Table 2.1: EXPERIMENTAL PARAMETER AND THEIR VALUE

Parameter	Value
K (total number of ants)	$\sqrt{M1 \times M2}$
ρ (evaporation rate)	0.05
λ (constant)	1
τ_{init} (initial value of each element of pheromone matrix)	0.0001
α(weighing factor of pheromone information)	4
β(weighing factor of heuristic information)	0.2
L(No. of ants movement steps within each construction step)	250
φ(pheromone decay coefficient	0.05
N(total no. of construction steps)	2

2.8.1 Kappa

It is a scalar parameter and used to measure accuracy. Introduced as a measure of agreement of medical world problems and measuring the degree of agreement between two observers observing similar phenomenon and also compensates the agreements that occurred by chance. Its use is now extended for classifier's accuracy. It can be represented as [40]

$$K = \frac{P_o - P_e}{1 - P_c} \tag{2.8.1}$$

where P_o represents the probability of total agreement and P_c represents the probability of agreement due to chance. It varies from -1 to +1 [40]. It is robust and conservative measure. Improves the stability of the selection method and kappa punishes randomness [41]. The kappa [36] is a parameter for the measurement of accuracy by a pixel to pixel comparison basis of two images I_1 and I_2 and represented by $k(I_1, I_2)$.

2.8.2 Figure-of-Merit (FOM)

Figure of merit is a measure for accuracy assessment of extracted edges. It represents the deviation from ideal image to the known image. It can be represented as [37]

$$FOM = \frac{1}{\max(I_I, I_A)} \sum_{i=1}^{I_{\mathbf{A}}} \frac{1}{1 + \alpha \times \ d_i^2} \tag{2.8.2}$$

where I_I and I_A are the number of ideal and actual edge points, $d(i)$ is the pixel miss distance of the i^{th} edge detected, and α is a scaling constant.

Table 2.2: Kappa and Figure of Merit Comparison

	Kappa		FOM	
	Tian *et al.*s [30]	Proposed	Tian *et al.*s [30]	Proposed
Cameraman	0.5048	0.6281	0.4905	0.9461
Lena	0.4866	0.5622	0.5245	0.8619
Coins	0.6783	0.6774	0.8493	0.9514
House	0.4906	0.6387	0.4803	0.9281
Peppers	0.4889	0.5781	0.4848	0.9008

Table 2.3: BDM and Hausdorff's distance comparison

	BDM		Hausdorffs Distance	
	Tian *et al.*s [30]	Proposed	Tian *et al.*s [30]	Proposed
Cameraman	2.8496	2.8207	6	5.381
Lena	4.1582	0.5905	4.5826	4.5826
Coins	1.561	1.0443	4.899	4.4721
House	3.4112	1.6079	4.3589	4.899
Peppers	4.2494	2.3386	6.7823	6.245

Table 2.4: Kappa values for comparison with majority image. column 2 Kappa for proposed ACO with majority image ,column 3-7:A comparison of Kappa,ratio of conventional edge detector to proposed ACO with respect to majority image obtained from other edge detectors

	Canny/Proposed	Sobel/Proposed	Prewitt/Proposed	Log/Proposed	Roberts/Proposed
CAMERAMAN	0.4363/0.6234	0.5928/0.6275	0.5917/0.6240	0.3526/0.6060	0.4411/0.6053
LENA	0.4138/0.5480	0.5799/0.5607	0.5803/0.5584	0.3145/0.5209	0.3580/0.5546
COINS	0.5638/0.6752	0.6725/0.6741	0.6727/0.6754	0.53120.6393	0.6215/0.6070
HOUSE	0.52890.6292	0.6676/0.6394	0.6677/0.6397	0.4913/0.6297	0.5453/0.6184
PEPPEERS	0.3802/0.5480	0.5638/0.5711	0.5621/0.5689	0.2903/0.5255	0.3701/0.5639

2.8.3 Baddeley's Delta Metric (BDM)

It is a parameter used to calculate the dissimilarity measure between two binary images. i.e., more the value, more the detected image varies from ideal image [42] [38].

2.8.4 Hausdorffś distance

Hausdorff distance is a metric used to find distance between two data sets. Since we want it for the digital images, it is limited to two dimensions. It is calculated as [39] let A=$\{a_1, a_2, a_3......, a_p\}$ and B=$\{b_1, b_2, b_3......., b_q\}$ be two data point sets. Then Hausdorff distance is defined as $H(A, B) = \max(h(A, B), h(B, A))$ where $h(A, B) = \max_{a\in A} \min_{b\in B} \|a - b\|$.

Table 2.2 and Table 2.3 is showing the comparison results between our proposed method and the Tian *et al.*'s method with respect to all the statistical parameters discussed above. Table 2.4 is showing relative comparison of our proposed edge detector and conventional edge detectors like Canny, Sobel, Prewitt, Log, and Roberts. F-ratio is providing a better one class separability measure because it is maximizing with the interclass difference being maximized and the intraclass difference being minimized. It was initially involved in the field of feature selection in linear separable problems [43]. As F-ratio can be assumed as a Signal-to-Noise ratio measurement [44], this methodology is helpful for the rejection of noisy components present in the feature domain . The implementation of conventional edge detectors is made possible with the help of MATLAB toolbox.

The kappa and FOM are the parameters representing the matching of two data sets. Therefore, the maximum of these values, the better the edge detectors performance. It is observed that the kappa values of comparison is showing the improvement of proposed method in most of the images. Also the proposed method is showing improvement in case of FOM in all images. BDM is showing poor performance in case of *Lena* image. The Hausdorff's distance is showing poorer performance in case of $House$ image. The BDM and Hausdorff's distance are the parameters representing the mismatch between two data sets. Therefore, the less these values of distance metric, the better the performance of edge detectors.

2.9 Summary

This chapter discusses in detail about the work presented in this thesis. It presents the development of the ACO in the area of the edge detection using F-ratio. Also it discusses about the parameters used for the comparison of the images developed through this work with the earlier developed method in this area. The visual comparison and statistical comparison is presented here for presenting the authentication of the developed method. The later chapter will discuss about the application of this novel work or technique in case of denoised images.

Chapter 3

Application Relative to De-noised Images

Wavelets

De-noising

Wavelet Thresholding De-noising

Gaussian Noise

Haar Transform

Results and Discussions

Summary

Chapter 3

Edge Detection on Denoised Images

3.1 Wavelet

A function represented by $\psi \in L^2(\Re)$ with [45] zero average $\int\limits_{-\infty}^{+\infty} \psi(t)\, dt = 0$.
It is normalized with $\|\psi\| = 1$ and neighbourhood is cantered on t=0.
A dictionary of time-frequency atoms is obtained by scaling by s and translation by u:

$$D = \left\{ \psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right) \right\}_{u \in, s \in^+} \tag{3.1.1}$$

Atoms are normalized with $\|\psi_{u,s}\| = 1$. The wavelet transform is represented as $f \in L^2(\Re)$ at time u and scale s as

$$W f(u,s) = \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt \tag{3.1.2}$$

3.2 De-noising

Signals get corrupted when it is generated, transmitted or processed [46,47]. So, denoising is the process of extracting important information from raw data and keeping out the noise or unnecessary details as much as possible. The new time frequency analysis, wavelet denosing can accurately detect the local features of signals, acting as a good method for noise removal.

3.3 Wavelet Thresholding De-noising

This technique [46, 47] is based on the fact that the energy of the signal information to be extracted is concentrated on some wavelet coefficients, while noise energy is spreaded over the all wavelet coefficients. Similarities between the fundamental basic wavelet and the signal defined are playing very important role which causes the possible for the signal to be spread over some of the few co-efficient. The impulses components are made as prominent as possible for getting improvement in the function of isolation of impulses. Wavelet thresholding de-noising [48–50] is efficiently removing the Gaussian noise which is identically and freely distributed . Let $x(t) = \{x_1(t), x_2(t), x_3(t), x_4(t), ...x_n(t),\}$ is the signal to be acquired by means of a sensor. The signal series, as generally known, if full of impulses and noises. So, x (t) can be represented as: p (t) + n(t)
where $p(t) = \{p_1(t), p_2(t), p_3(t), p_4(t), ...p_n(t),\}$
and $n(t) = \{n_1(t), n_2(t), n_3(t), n_4(t), ...n_n(t),\}$
Represents the identically distributed and statistically distributed Gaussian noise with mean zero and standard deviation σ. The wavelet de-noising method [46, 51] consists of three steps:

- Transformation of the original signal to the time-scale plane with the help of wavelet transforms. The results of the transform can be obtained in various different scales.
- Accessing a proper value of threshold and according to the rules of the shrinkage shrinks the coefficients.
- The shrunken coefficients are further undergoes through the inverse wavelet transform.

As from the second step, it is very crucial for the process and so determined by universal threshold rule $\lambda = \sigma\sqrt{2 \ln N}$ where σ refers to the standard deviation of the noise and if not known, it is estimated with the finest scale co-efficient with the development of a median absolute deviation where $\sigma = MAD/0.6745$ where N refers to the number of data samples of the measured signal.
Thresholding function in this case is [46, 47, 52] of two types:

- Hard thresholding
- Soft thresholding

Thresholding function now determines the wavelet shrinkage function which decides hoe the threshold will operate on the wavelet co-efficient.

Hard thresholding: The hard thresholding [46, 47, 52] function as prescribed by Donoho is as follows:

$$w_{j,k} = \begin{cases} w_{j,k} & |w_{j,k}| \geq \lambda \\ 0 & |w_{j,k}| < \lambda \end{cases} \tag{3.3.1}$$

It is a following a very hard and fast rule that the co-efficient having value less than the threshold will all be replaced by zero. Also the values greater than the threshold are kept as it is .

Soft thresholding: The soft thresholding [46, 47, 52] is somehow provides the real function of shrinkage like as follows:

$$\begin{aligned} \overline{w_{j,k}} &= \{\mathrm{sgn}(\ w_{j,k})\,(|w_{j,k}| - \lambda) \quad if\ |w_{j,k}|\ > \lambda \\ &= 0 \qquad\qquad\qquad\qquad\quad if\ |w_{j,k}|\ < \lambda \end{aligned} \tag{3.3.2}$$

The above one is equivalent to the shrinkage function as it keeps the value above the threshold shrinks by the amount of and otherwise keeps the value to zero. Sgn is the signum function whose function is defined as:

$$\mathrm{sgn}(n) = \begin{cases} 1 & n > 0 \\ -1 & n < 0 \end{cases} \tag{3.3.3}$$

3.4 Gaussian Noise

Because of the mathematical tractability of the Gaussian noise in spatial and frequency domain [9], Gaussian models are used in situations in the case where they are marginally at best.

The Pdf of a guassian random variable, z, is given by:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\left.(z-\mu)^2\right/2\sigma^2} \tag{3.4.1}$$

Where z is representing the gray level, is the mean of average value of z, and σ the standard deviation. Variance of z is the standard deviation squared.

3.5 Haar Transform

The Haar Transform [9,53] is having relation with multi-resolution analysis in the imaging-related operation. The basis functions are the oldest and simplest known orthonormal wavelets. The Haar transform is both separable and symmetric and expressed as:

$$\mathbf{T=HFH} \tag{3.5.1}$$

Where F is an $N \times N$ image matrix, H is $N \times N$ transformation matrix, and T is the resulting $N \times N$ transform. The haar basis functions are:

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, z \in [0, 1] \tag{3.5.2}$$

The 4×4 transformation matrix, H4, is represented as

$$H_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \tag{3.5.3}$$

The 2×2 transformation matrix H_2 is

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.5.4}$$

Haar Transform calculates the differences [8] of the values of the samples for the calculation.

3.5.1 Haar Transform Properties

The General properties [8] of the Haar Trasform are as follows:

1. Haar transform possesses orthogonality and real properties i.e., $Hr = Hr^*$

2. It is a very fast in transform operations.

3. Sequential ordered basis vectors containing Haar matrix.

4. Containing poor energy crunch for images.

3.6 Results and Discussions

The denoised images are obtained by the application of the Haar wavelet hard thresholding on the noisy images corrupted by Gaussian Noise. The proposed technique mentioned in previous chapter is applied over the denoised images. For the Denoising [47] purpose hard thresholding is used and set of images for comparison are applied with the hard thresholding [47].

The test images, images after the mixture of gaussian noise and also the denoised images are shown in sequence for *Cameraman, Lena, Coins, House and Peppers* from Fig 3.1,3.2, 3.3, 3.4,3.5. The images comparison is shown in Fig. 3.6, 3.7, 3.10, 3.8, 3.9. The comparison shows that the proposed approach is quite producing a good true edge map as compared with its conventional edge detectors edge maps. It is also seen with the observation that the proposed approach of Edge detection based on the combination of ACO and F-ratio.

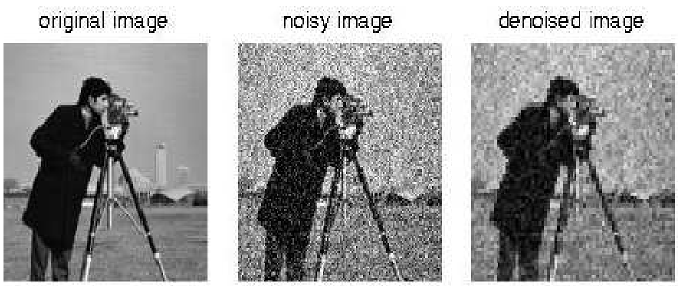

Figure 3.1: *Left-Cameraman test Image, Middle-Noisy Image, and Right-Denoised Image*

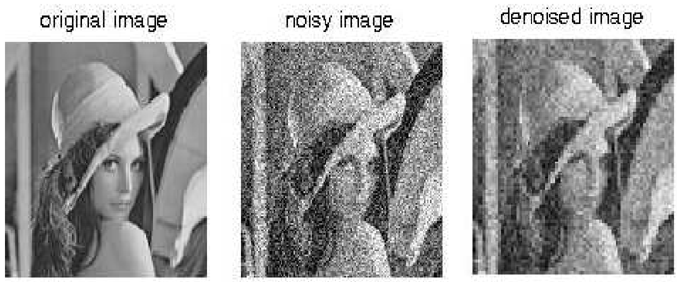

Figure 3.2: *Left-Lena test Image, Middle-Noisy Image, and Right-Denoised Image*

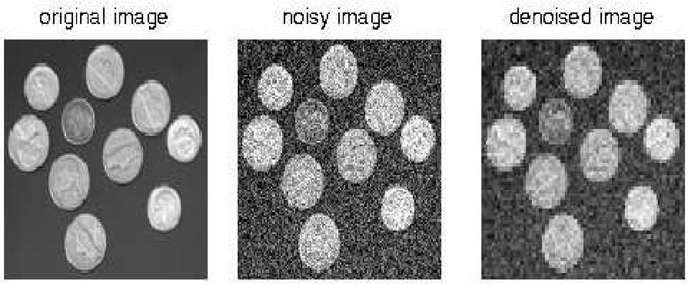

Figure 3.3: *Left-Coins test Image, Middle-Noisy Image, and Right-Denoised Image*

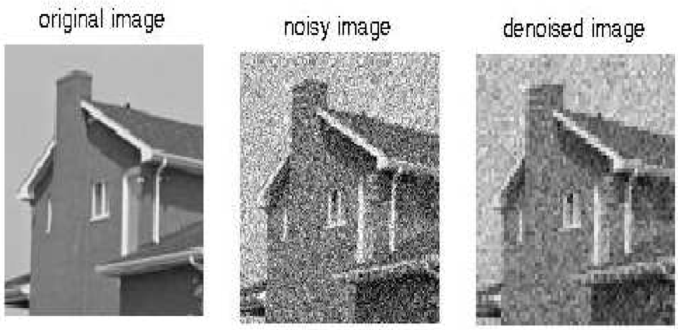

Figure 3.4: *Left-House test Image, Middle-Noisy Image, and Right-Denoised Image*

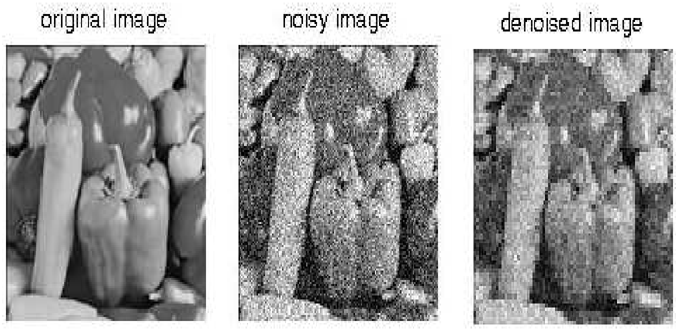

Figure 3.5: *Left-Peppers test Image, Middle-Noisy Image, and Right-Denoised Image*

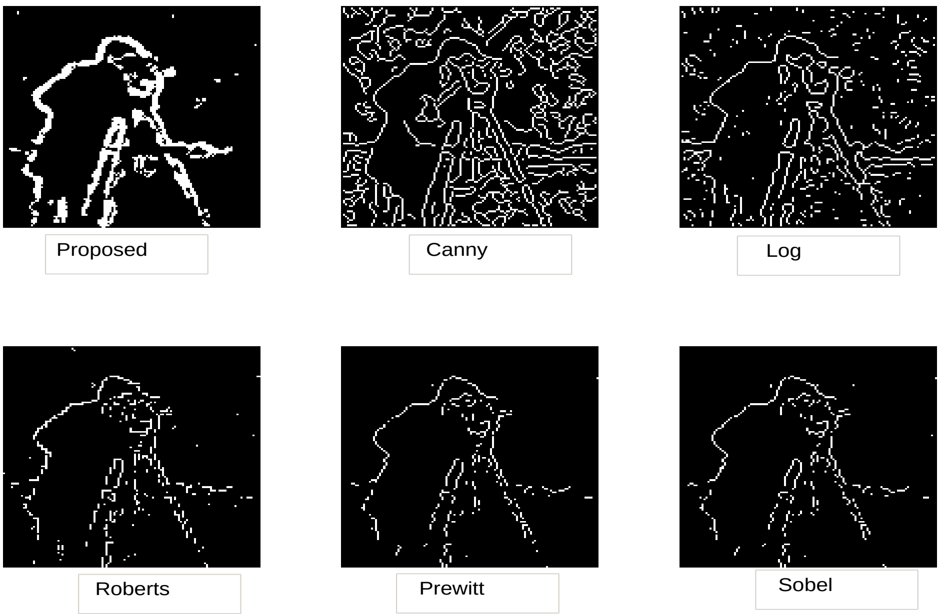

Figure 3.6: *Top left-Cameraman Proposed Method, Top Middle-Canny's, Top Right-Log, Bottom left-Roberts, Bottom-middle-Prewitt ,and Bottom Right-Sobel*

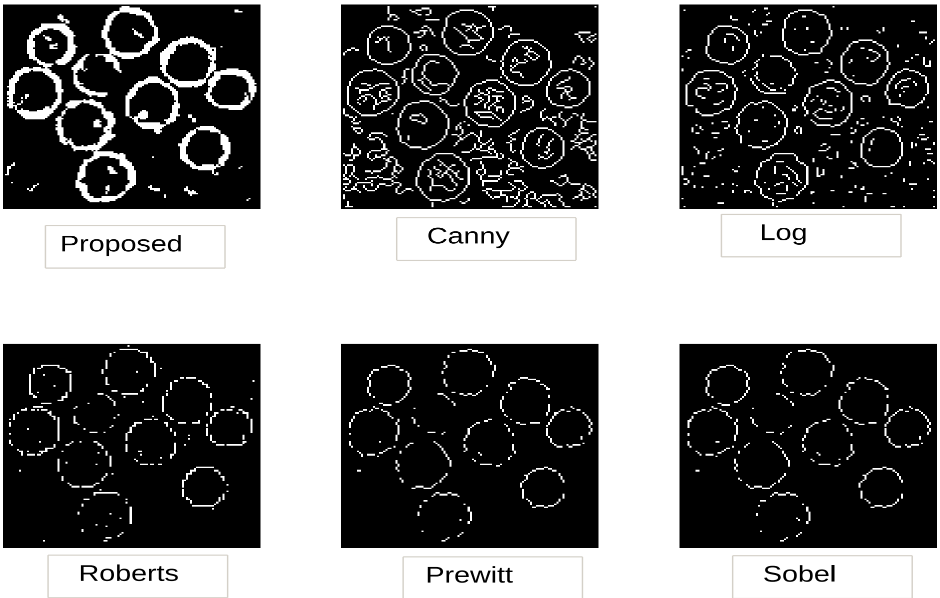

Figure 3.7: *Top left-Coins Proposed Method, Top Middle-Canny's, Top Right-Log, Bottom left-Roberts, Bottom-middle-Prewitt ,and Bottom Right-Sobel*

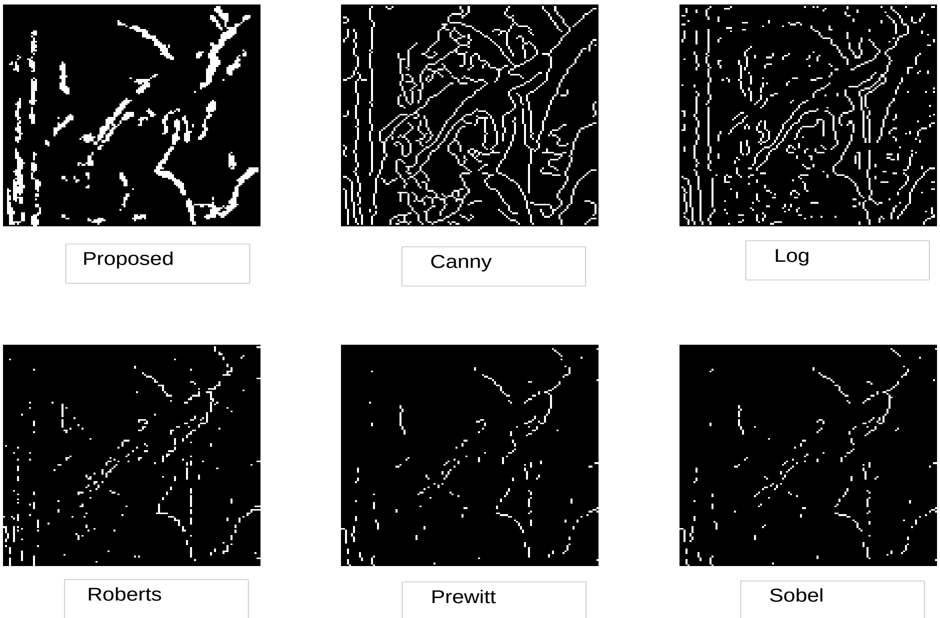

Figure 3.8: *Top left-Lena Proposed Method, Top Middle-Canny's, Top Right-Log, Bottom left-Roberts, Bottom-middle-Prewitt ,and Bottom Right-Sobel*

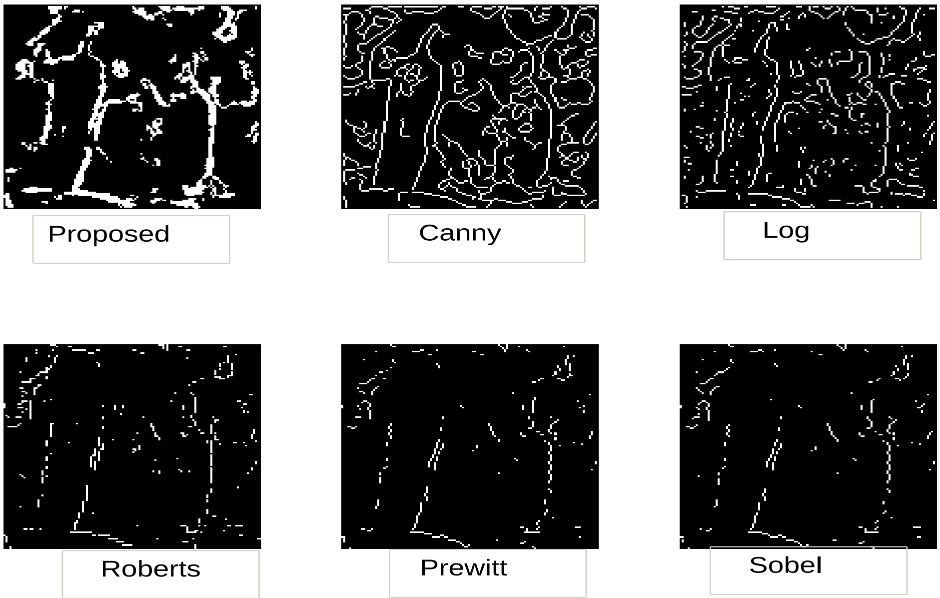

Figure 3.9: *Top left-Peppers Proposed Method, Top Middle-Canny's, Top Right-Log, Bottom left-Roberts, Bottom-middle-Prewitt ,and Bottom Right-Sobel*

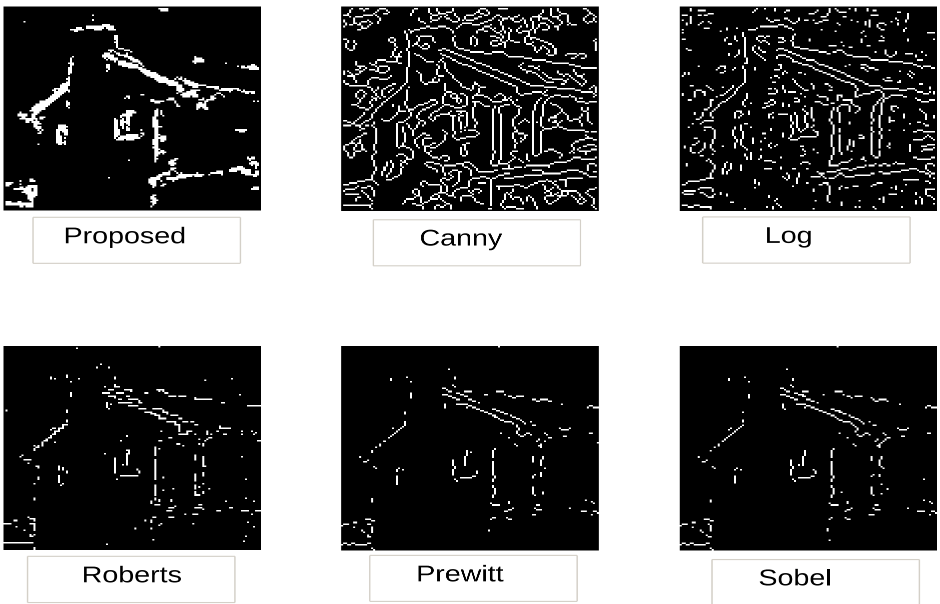

Figure 3.10: *Top left-House Proposed Method, Top Middle-Canny's, Top Right-Log, Bottom left-Roberts, Bottom-middle-Prewitt ,and Bottom Right-Sobel*

3.7 Summary

Denoising is the process of extracting out the important details from an image, keeping the noise out of it as much as possible. The wavelet thresholding is used here to get the denoising process done. The proposed technique of edge detection is implemented on the denoised images. The images so obtained is compared with other conventional edge detectors images and it is seen that the proposed approach is quite immune to noise and at the same time provides good edge maps.

Chapter 4

Conclusions and Future Work

Chapter 4

Conclusion and Future Work

4.1 Conclusion

Edge detection process is an important part of image processing. It is beneficial for many research areas of computer vision and image segmentation. Edge detection provides important details for the high-level processing tasks like feature detection etc. The success of edge detection depends on the optimal calculation of threshold. This thesis discusses the achievement obtained by the implementation of a novel technique of image edge detection based on Ant Colony Optimization (ACO) and F-Ratio. ACO is a nature inspired algorithm. It takes into account the various advantage of ant colony like stigmergy, distributed computation, pheromone evaporation, decision-making based on pseudo-random proportional rule. These features are quite helpful for the determination of pheromone matrix, which contains information related to the edge. Edges are the areas with sharp intensity change. The pheromone matrix so obtained is processed with the help of class separability measure,F-ratio. The output of the F-ratio provides the index that leads to the determination of optimum threshold. This threshold value is used further for the development of edge-map.

The testing of the success of the edge-map developed by the proposed method presented in this thesis is also evaluated with the help of statistical parameters like kappa, Figure-of-Merit, Hausdorff's distance and Baddeleys's Delta Metric. The proposed method is also applied on the denoised images developed with the help of Wavelet thresholding. The results obtained are compared with the traditional edge detectors.

4.2 Future Work

This proposed work in this thesis is having a lot of potential for further research in the area of edge detection using different paradigm making the work more versatile and flexible. The research can be extended in the area of noisy images directly as input in the methodology presented in this work. Also the proposed work can be further studied observing the different parameter variations and inclusion of some dynamic problem sensing feature which can adjust the parameter values to the values optimal for the specific situation.

Bibliography

[1] M. Dorigo and T. Stützle, *Ant Colony Optimization.* Cambridge, Massachusetts: The MIT Press, 2004.

[2] Deneubourg, S. Aron, S. Goss, and J. M. Pasteels, "The self-organizing exploratory pattern of the argentine ant," *Journal of Insect Behavior*, vol. 3, pp. 159–168, Mar. 1990.

[3] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life Reviews*, vol. 2, pp. 353–373, Dec. 2005.

[4] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, pp. 533–549, May 1986.

[5] C. R. Reeves, ed., *Modern heuristic techniques for combinatorial problems.* New York, NY, USA: John Wiley & Sons, Inc., 1993.

[6] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, pp. 137–172, 1999.

[7] S. K. Papadimitriou CH, "Combinatorial optimizationalgorithms and complexity," vol. New York: Dover,, 1982.

[8] A. K. Jain, *Fundamentals of digital image processing.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Reading, MA: Addison-Wesley, 1992.

[10] T. Acharya and A. K. Ray, *IMAGE PROCESSING Principles and Applications.* Wiley Interscience, 2005.

[11] R. Rajeswari and R. Rajesh, "A modified ant colony optimization based approach for image edge detection," in *Image Information Processing (ICIIP), 2011 International Conference on*, pp. 1–6, 2011.

[12] O. Verma and R. Sharma, "An optimal edge detection using universal law of gravity and ant colony algorithm," in *Information and Communication Technologies (WICT), 2011 World Congress on*, pp. 507 –511, Dec. 2011.

[13] R. Maini and J.S.Sohal, "Performance evaluation of prewitt edge detector for noisy images," *ICGST International Journal on Graphics, Vision and Image Processing*, vol. 6(3), pp. 39–46, 2006.

[14] L. G. Roberts, *Machine Perception of 3-D Solids*, pp. 159–197. MIT Press, 1965.

[15] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London Series B*, vol. 207, pp. 187–217, 1980.

[16] J. Canny, "A computational approach to edge detection.," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[17] O. P. Verma, M. Hanmandlu, P. Kumar, S. Chhabra, and A. Jindal, "A novel bacterial foraging technique for edge detection," *Pattern Recogn. Lett.*, vol. 32, pp. 1187–1196, June 2011.

[18] S. Goss, S. Aron, J. Deneubourg, and J. Pasteels, "Self-organized shortcuts in the Argentine ant," *Naturwissenschaften*, vol. 76, pp. 579–581, Dec. 1989.

[19] M. Dorigo, M. Birattari, and T. Sttzle, "Ant Colony Optimization: Artificial ants as a computational intelligence technique," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[20] M. Dorigo, *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.

[21] B. Bullnheimer, R. F. Hartl, and Ch, "A New Rank Based Version of the Ant System," *Central european journal of operations research*, vol. 7, no. 1, pp. 25–38, 1999.

[22] T. Stützle and H. H. Hoos, "Max-min ant system," *Future Generation Comp. Syst.*, vol. 16, no. 8, pp. 889–914, Jun. 2000.

[23] M. Dorigo and L. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Trans. on Evolutionary Computation*, vol. 1, pp. 53 –66, Apr 1997.

[24] P. Xiao, J. Li, and J.-P. Li, "An improved ant colony optimization algorithm for image extracting," in *Apperceiving Computing and Intelligence Analysis (ICACIA), 2010 International Conference on*, pp. 248 –252, Dec. 2010.

[25] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of cooperating agents," *IEEE Trans. on Systems, Man, and Cybernetics - Part B*, vol. 26, no. 1, pp. 29–41, 1996.

[26] G. Saha, S. Senapati, and S. Chakroborty, "An F-ratio based optimization on noisy data for speaker recognition application," in *IEEE Indicon Conference*, pp. 352 – 355, ,Chennai,India, Dec.,2005.

[27] A. Paul and S.Mukhopadhyay, "An adaptive pheromone updation of the ant system using LMS technique," pp. 498–503, Modelling, Optimization and Computing (ICMOC-2010) AIP, vol. 1298, 2010.

[28] D.-S. Lu and C.-C. Chen, "Edge detection improvement by ant colony optimization," *Pattern Recogn. Lett.*, vol. 29, pp. 416–425, Mar. 2008.

[29] J. Yang and Y. Zhuang, "An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem," *Appl. Soft Comput.*, vol. 10, pp. 653–660, Mar. 2010.

[30] J. Tian, W. Yu, and S. Xie, "An ant colony optimization algorithm for image edge detection," in *IEEE World Congress on Evolutionary Computation*, pp. 751 –756, Jun. 2008.

[31] O. P. Verma, M. Hanmandlu, A. K. Sultania, and Dhruv, "A novel fuzzy ant system for edge detection," in *Proceedings of the 2010 IEEE/ACIS 9th International Conference on Computer and Information Science*, ICIS '10, (Washington, DC, USA), pp. 228–233, IEEE Computer Society, 2010.

[32] H. Nezamabadi-pour, S. Saryazdi, and E. Rashedi, "Edge detection using ant algorithms," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 10, pp. 623–628, May 2006.

[33] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[34] Y. Yitzhaky and E. Peli, "A method for objective edge detection evaluation and detector parameter selection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1027 – 1033, Aug. 2003.

[35] D. J. Bryant and D. W. Bouldin, "Evaluation of edge operators using Relative and absolute grading," in *IEEE Comput. Soc. Conf. on Pattern Recognition and Image Processing*, pp. 138–145, 1979.

[36] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[37] I. Abdou and W. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proceedings of the IEEE*, vol. 67, pp. 753 – 763, May 1979.

[38] A. J. Baddeley, "An error metric for binary images," *Robust Computer Vision Quality of Vision Algorithms*, no. march 1992, pp. 59–78, 1992.

[39] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 850 –863, Sep 1993.

[40] A. Ben-David, "About the relationship between roc curves and cohen's kappa," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 6, pp. 874 – 882, 2008.

[41] S. Vieira, U. Kaymak, and J. Sousa, "Cohen's kappa coefficient as a performance measure for feature selection," in *2010 IEEE International Conference on Fuzzy Systems (FUZZ),*, (Piscataway, NJ, USA), pp. 1–8, 2010.

[42] A. J. Baddeley, "Errors in binary images and an Lp version of the Hausdorff metric," *Nieuw Arch. Wiskunde*, vol. 10, pp. 157–183, 1992.

[43] K. Mao, "RBF neural network center selection based on fisher ratio class separability measure," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1211 – 17, Sept. 2002.

[44] T. H. Dat and C. Guan, "Feature selection based on fisher ratio and mutual information analyses for robust brain computer interface," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, pp. I–337 –I–340, April 2007.

[45] S. Mallat, *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Academic Press, 2 ed., Sept. 1999.

[46] H. Zang, Z. Wang, and Y. Zheng, "Analysis of signal de-noising method based on an improved wavelet thresholding," in *Electronic Measurement Instruments, 2009. ICEMI '09. 9th International Conference on*, pp. 1–987 –1–990, aug. 2009.

[47] S. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *Image Processing, IEEE Transactions on*, vol. 9, pp. 1532 –1546, Sep 2000.

[48] D. Donoho, "De-noising by soft-thresholding," *Information Theory, IEEE Transactions on*, vol. 41, pp. 613 –627, may 1995.

[49] D. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, pp. 1200–1224, 1995.

[50] R. R. Coifman and D. L. Donoho, "Translation-Invariant De-Noising," tech. rep., Department of Statistics, 1995.

[51] Q. Tianshu, W. Shuxun, C. Haihua, and D. Yisong, "Adaptive denoising based on wavelet thresholding method," in *Signal Processing, 2002 6th International Conference on*, vol. 1, pp. 120 – 123 vol.1, aug. 2002.

[52] L. Su and G. Zhao, "De-noising of ecg signal using translation- invariant wavelet de-noising method with improved thresholding," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pp. 5946 –5949, jan. 2005.

[53] T. Davis, "Fast decomposition of digital curves into polygons using the haar transform," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, pp. 786 –790, aug 1999.

Publications

1. Prashant K Mohanty, Samit Ari, Dipak Kumar Ghosh, Sarat K Patra, "On an Edge Detection algorithm based on Ant Colony Optimization and Fisher Ratio," *International Journal of Electronics and Communications*, (Communicated).