EDGE DETECTION USING BACTERIAL FORAGING AND UNIVERSAL LAW OF GRAVITY

MAJOR PROJECT SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY

in

INFORMATION SYSTEM

SUBMITTED BY

Tanya Kumari

[2K14/ISY/16]

Under the Guidance

Of

Dr. O.P VERMA

(Prof. and Head, Department of CSE)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(2014-16)

CERTIFICATE

This is to certify that report entitled **TANYA KUMARI (2K14/ISY/16)** has carried out the major project titled **"Edge Detection Using Bacterial Foraging And Universal Law Of Gravity"** in partial fulfilment of the requirement for the award of degree of Master of Technology degree in Information System by **Delhi Technological University,** New Delhi.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session 2014-2016. To the best of my knowledge the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any other degree or diploma.

Dr.O.P.Verma

Professor and Head

Department of Computer Science and Engineering

Delhi Technological University

Delhi-110042

ACKNOWLEDGEMENT

It gives me a great sense of pleasure to present the report of the Major Project work, undertaken during M. Tech. I owe special debt of gratitude to Dr. O.P Verma, Professor and Head of Department, Department of Computer Science and Engineering, Delhi Technological University, Delhi, for his constant support and guidance throughout the course of my work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for me. It is only his cognizant efforts that my endeavours have seen light of the day.

I take the opportunity to acknowledge the contribution for his full support and assistance during the development of the project. I also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of my Major Project report.

Tanya Kumari

Roll No: 2K14/ISY/16

M.Tech (Information Systems)

E-mail: tanyahome1991@gmail.com

ABSTRACT

Edge detection is an important step used in various image processing algorithms. Edge gives the boundary of objects and provides information to separate the object from background or other overlapping objects. This work presents a new approach for edge detection inspired by Bacterial Foraging Algorithm (BFA) and Universal Law of Gravity. Among the many evolutionary algorithms proposed BFA is one of them. It consists of 4 stages viz chemotaxis, reproduction, swarming, elimination and dispersal. Chemotaxis stage decides the direction of movement of bacteria. Our aim is to move the bacteria through the edge pixels. The direction of movement of bacteria has been determined using the concept of Universal Law of gravity. The results of the proposed approach are compared with traditional edge detectors using different sample images. Quantitative analysis of the proposed approach with respect to traditional edge detectors has been done using Kappa and Entropy measures.

TABLE OF CONTENTS

FIGURES AND TABLES

Chapter 1

INTRODUCTION

Edges detection is the most important problem in image processing due to the fact that the most important information in images is carried by edges. Edge detection is the technique of marking sharp intensity changes, which is used to further analyse the image content. Edges mostly occur between the boundaries of different regions in an image. The intensity changes in an image may be caused by-

1. Geometric events such as object boundary, surface boundary
2. Non-geometric events such as specularity, shadow, inter-reflection.

Most of the techniques of edge detection described in literature follow three steps i.e smoothing, detection and localization.

1.1 Type of edges

The following type of edges exists according to [1]

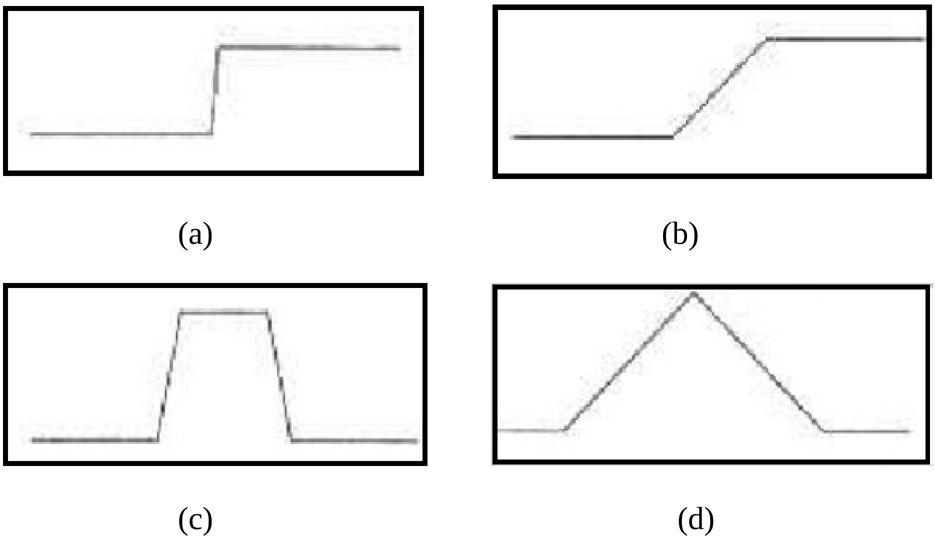


Fig 1.1 Types of edges (a) Step Edge (b) Ramp Edge (c) Ridge edge (d) Roof Edge

1. **Step edge-** The change in grey level value abruptly from one side to another value on the other side of the discontinuity. This kind of edge is considered as an ideal edge.
2. **Ramp Edge-** In this type of edge the change of grey level value is not sharp as in a sharp edge, rather the change is continuous.

3. **Ridge/Line edge-** There is an abrupt change in grey value but after a short distance it returns to the starting grey value
4. **Roof Edge-** It is a kind of ridge edge where the grey level value does not change abruptly rather the change occurs over a finite distance.

There are 2 types of edge detectors depending on the use of edge detector in computer vision system:-

1. Autonomous detectors – These kinds of detectors do not require prior knowledge about the edge and scene to be detected. A pixel is labelled as an edge based on its neighbouring pixels.
2. Contextual detectors- These kinds of detectors require prior knowledge about the structure of the scene and the edge. They perform for only precise context.

There are various approaches available for edge detection such as wavelet approach evolutionary algorithms, gradient based approach fuzzy logic etc.

1.2 Application of edge detection

Following are the applications of edge detection

1. Image segmentation- the process of dividing an image into different segments where each segment represents an object or part of an object. Edge detection can be used to determine the boundary of such objects.
2. Image Compression- When an image is represented s an edge map the amount of data required to be stored for the image is reduced. When the edge pixels of an image are transmitted it results in compression.
3. Image Encryption- The edge data in an image can be altered skilfully to carry hidden information.
4. Security- various biometric methods exist to verify humans such as face recognition, iris recognition, fingerprint recognition etc. These biometric methods can use the edge maps.
5. Medical Imaging- edge detection is used for X-ray, CT-Scan , tumour detection in MRI images of the brain..
6. Geological information extraction- The images from captured by satellites are first converted into digital images. Edge detection when applies on these images can be used to geological faults in area.

Since most of these applications use edge detection as their primary step, therefore edge detection must be efficient and reliable.

1.3 Motivation

The motivation for edge detection is to capture changes in properties of the world and important events. Locating an edge accurately makes sure that that the objects are located properly and some of their properties such as shape, circumference perimeter can be measured. Edge detection serves as an important tool foe separating and objects from its background which can be further used for applications such as image segmentation.

Three types of edge can be produced by edge detectors:-

1. True edges- Edges which are actually present in an image and are detected
2. False positive- Edges which are not present an image but are detected
3. False negative- Edges which are present in an image but are not detected

The edge detectors should –

1. Maximize the true edges and minimize the false positive and false negative edges.
2. The edge detectors should have good localization i.e. the edges which are detected should be near to the correct position of edges in the image.
3. Noisy edges should be filtered out by the edge detector depending on the threshold value being used

1.4 Scope

Scope of the project is as follows:-

- Study of various edge detection algorithms.
- Implementation of Bacterial foraging optimization and universal law of gravity for edge detection
- Study the output of edge detection.
- Compare the results with classical edge detectors

1.5 Thesis Organization

The rest of the thesis is organized as follows-

Chapter 2 provides the literature review which gives brief idea about some basic edge detectors and some edge detectors based on evolutionary algorithms.

Chapter 3 describes the bacterial foraging algorithm, 4 stages of bacterial foraging. Some heuristics are also laid down and guidelines for parameter choices are explained.

Chapter 4 explains the concept of universal law of gravity and how this concept can be applied to images for edge detection.

Chapter 5 explains the proposed methodology. It explains the method of edge detection using bacterial foraging and universal law of gravity.

Chapter 6 shows the experimental results which includes qualitative as well as quantitative comparison with some well-known edge detectors.

Chapter 7 concludes the thesis.

Chapter 2

LITERATURE SURVEY

2.1 General edge detectors

Many edge detectors have been proposed in history which are broadly classified as first order or second order derivative edge detectors. In first order operators edge occurs at the maxima whereas in second order operators edges occur at the zero crossing of second derivative. Sobel [2] is first derivative operator which uses a 3X3 convolution kernel. It uses two types of kernels one running horizontally and other vertically to detect these two kind of edges. Prewitt [3] is also a first derivative operator and is similar to Sobel but the kernel used for Prewitt is different from that of Sobel. In Prewitt greater weight is not given to the pixels that are nearer to the centre, unlike Sobel. Robert [4] is also a first derivative edge detector that uses 2X2 convolution kernels. The kernels are designed to respond to edge running in vertical directions. All these first order operators are sensitive to noise and produce thick edges; therefore second order operators were introduced.

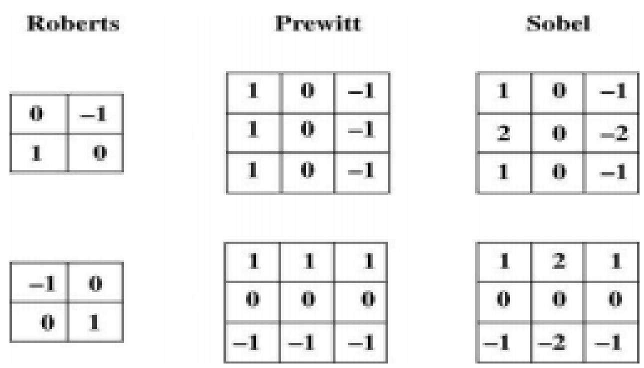


Fig. 2.1 Convolution kernels of Robert, Prewitt and Sobel

Laplacian of Gaussian is a second order edge detector in which the image is first blurred to reduce noise using Gaussian smoothing filter. After this the Laplacian operator is applied. Canny [5] edge detector is the optimal edge detector. Edges are detected by applying smoothing, non-maximum suppression, thresholding and hysteresis i.e. edges which are not connected to good edges are removed. In Canny double threshold can be used to improve the results. Canny has better results than other edge detectors according to [6] Canny has the advantage of low error rate, good localization and one response to a single edge point , therefor it is the most preferred one among the above edge detectors. The disadvantage of second order operators is that they are sensitive to noise.

2.2 Edge detection using evolutionary algorithms

Charles Darwin proposed the biological model of natural selection and evolution. The algorithms inspired by this model are known as evolutionary algorithms. Most of these algorithms are based on the phenomenon which occurs in nature and animals. These algorithms have been successful in solving complex computational problems like edge detection, image segmentation and many more. In this section edge detection using evolutionary algorithms will be discussed.

Ant Colony Optimization (ACO) proposed by Dorigo *et al* [7] is a technique which is derived from the behaviour of the ants to search for food i.e. the foraging strategy of ants. An ant while moving on its path deposits pheromones. Greater is the pheromone deposited on a path greater is the probability that the path will be followed by other ants [8]. To represent the edge information of a pixel pheromone matrix is used. R.Rajeswari *et al* [9] proposed a method in which heuristic information is used by ants based on the edginess of a pixel. Further to determine whether a pixel is edge or not fuzzy clustering has been used. Lu *et al* a [10] proposed that ACO must be used as a post processing step to link the disconnected edges.

OP Verma *et al* [11] proposed a technique for edge detection using adaptive thresholding and ACO which is described in brief as follows-

1. Perform threshholding on the image to get an edge map
2. Place the ants on the end points of the edges obtained in step1 and perform the procedure of ACO
3. Combine the edge map obtained in step 1 and step 2.
4. Perform thinning on the edge map obtained in step 3 to remove double edges.

Ari Samit *et al* [12] proposed that the threshold to be used on the pheromone matrix to determine edge map should be calculated by Fisher-Ratio. A technique for edge detection using ACO and fuzzy derivative was proposed by Verma Om Prakash *et al* [13]. It was proposed to use Sobel operator to reduce the discontinuity present in an image.

Gravitational search algorithm is based on the newton law of gravity. Sun Genyun *et al* [14] introduced the first approach for edge detection using universal law of gravity in 2007. OP Verma *et al* [15] presented an approach for edge detection using Gravitational Search Algorithm. The movement of agents over the search space takes place according to the theory of GSA i.e. acceleration and velocity are calculated to update the position of the agents. Lopez Molina *et al* [16] proposed an approach which uses t-norms and GSA for edge detection. In this approach the product operation in the formula to calculate force has been replaced by T-norm operation. This has been done to create a fuzzy set representing edges. By using T-norm operation the magnitude of force lies in the interval [0,1], so that the force can be used as valid membership value.

OP Verma *et al* [17] proposed a technique for edge detection using bacterial foraging algorithm. In this approach the direction of movement of bacteria has been found out using probabilistic derivative technique. Further in another paper bacterial foraging is used on a binarized image [18]. Binarization results in image where the pixel can have only 2 possible values 0 and 255. Whenever a bacterium find out that the intensity difference with a neighbouring pixel is 255, it will mark that pixel as an edge and move toward it.

Chapter 3

BACTERIAL FORAGING ALGORITHIM

3.1 Description

Bacterial Foraging Optimization Algorithm proposed by Passino [19] is a nature inspired optimization algorithm. It exploits the foraging of E.colli bacteria found in the gut of human beings. The bacteria want to maximize the energy per unit time by eating as much food as they can. BFOA has been applied to solve some problems successfully such as colour image enhancement [20], harmonic estimation [21] etc. BFOA is inspired by the chemotaxis behaviour of the bacteria. The E.colli bacterium consists of flagella on their body which helps them in movement. The rotation of the flagella in clockwise direction causes the bacteria to swim/run, whereas the rotation in anticlockwise direction causes the bacteria to tumble in a new direction [22] as shown in fig 3.1

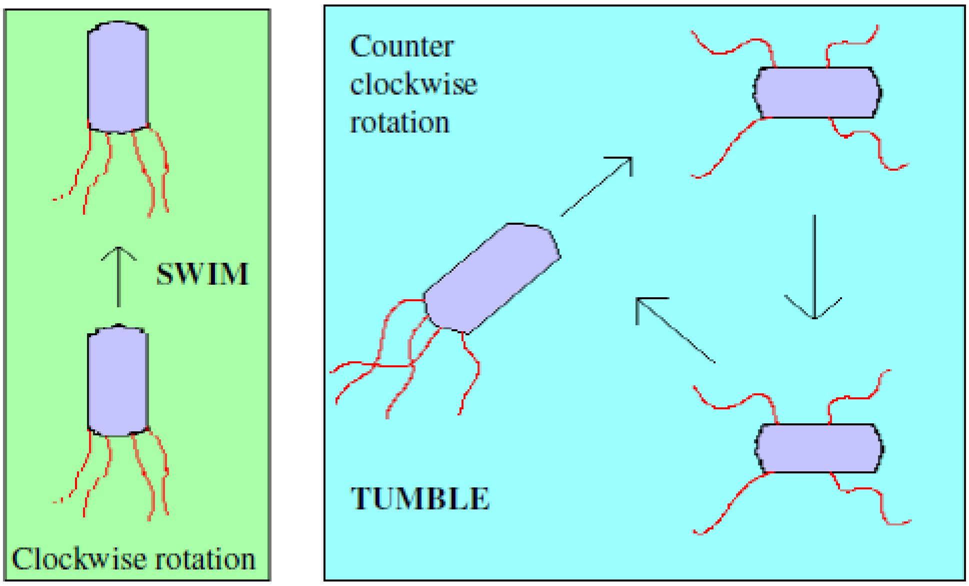
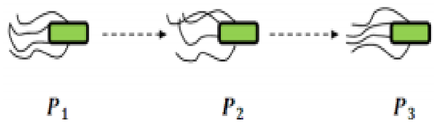
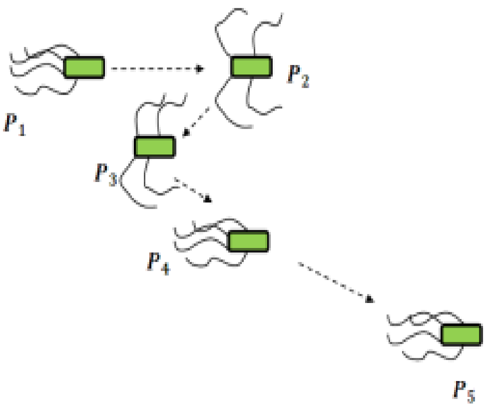


Fig 3.1 Swim and Tumble of bacterium

During chemotaxis the bacteria try to move up the nutrient gradient and avoid noxious regions. When a bacterium approaches a nutrient gradient, it swims in the same direction fir some steps. Tumbling occurs when the bacteria need to search for more food in some other direction. The illustration of when bacteria swims and tumbles is shown in the figure 3.2 below



(a) Moving forward continuously(swim)



(b) Moving forward-tumbling-swim

Fig 3.2 Illustration of chemotaxis pattern of E.colli bacteria[19]

The swimming of bacterium is is illustrated in fig 3.2(a). When the bacterium moves from position P_1 to P_2 it observes that the nutrient concentration at the new position is higher than the previos position therefore it continues to swim in the same direction for some number of steps. Figure 3.2(b) shows both swimming and tumbling of bacteria. When the bacteria moves from position P_1 to P_2 it finds out that the nutrient concentartion at this new position is lower than the previous one,therefore it randomly tumbles in a new direction to position P_3. Again the nutrient concentration at position P_3 is lower than position P_2 so the bacterium tumbles to anew direction P_4. Now the nutrient concentration at P_4 is higher than P_3 therefore the bacterium swims in the same direction for some number of steps.

Suppose there is a need to find out the global minima of $J(\Theta)$, where Θ is the position of the bacteria. Then the condition $J(\Theta) <0$ indicates that the bacteria is in nutrient rich condition, $J(\Theta)>0$ indicates that the bacteria is in noxious region and $J(\Theta) =0$ indicates that the bacteria is in neutral region.

In addition studies have shown that bacterium act as a group and can affect each other. The bacteria communicate their situation to other bacterium through cell to cell communication i.e. swarming.

3.2 Constituting Steps

During its lifetime the bacterium goes through 4 different stages-

1. Chemotaxis
2. Swarming
3. Reproduction
4. Elimination and Dispersal

Each of them is described as below [19]

Chemotaxis

This process stimulates the tumbling and swimming of the E.colli bacterium via the flagella. Let $\theta^i(j,k,l)$ be the position and J(i,j,k,l) be the nutrient concentration of the *i*th bacteria at *j*th chemotactic, *k*th reproductive and *l*th elimination-dispersal step. C(i) is the chemotactic step size during each swim or a tumble. Thus for each chemotactic step the movement of bacteria, the movement of the bacteria may be represented by

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + \frac{C(i).\Delta(i)}{\sqrt{\Delta(i)^T\Delta(i)}} \quad (1)$$

Where, Δ(i) represents a random number that defines the direction of the bacterium movement. If the cost function value J(i,j+1,k,l) at the position $\theta^i(j+1,k,l)$ is lower than J(i,j,k,l) then the bacteria will move one more step in the same direction as the previous one. N_s are the maximum number of swim steps in the same direction. The bacterium tumbles after N_s. The chemotaxis of bacteria is an integration of the following

1. Constant swim
2. Swim after tumble
3. Tumble after tumble
4. Tumble after swim

Swarming

The bacterium depends on each other. When a bacterium finds a nutrient rich area it releases an attractant signal so that other bacteria swarm towards it. It also releases a repellent signal so that safe distance is maintained between the bacteria. This behaviour causes the bacteria to move in a concentric pattern. A cell to cell signalling function is introduced to accommodate the attraction and repletion phenomenon.

$$J_{CC}\left(\theta^i(j,k,l),\theta(j,k,l)\right)=\sum_{t=1}^{s}\left[-d_{att}exp\left(-w_{att}\sum_{m=1}^{P}(\theta_m^i-\theta_m^t)^2\right)\right]+\sum_{t=1}^{s}\left[h_{rep}exp\left(-w_{rep}\sum_{m=1}^{P}(\theta_m^i-\theta_m^t)^2\right)\right] \quad (2)$$

Where

P= dimension of search space

S=number of bacterium in search space

h_{rep}= magnitude of repellent effect

d_{att}= amount of attractant released

w_{att}= diffusion rate of attractant

w_{rep}= width of repellent

θ^i= position of *i*th bacteria

θ_m^i= *m*th component of θ^i

Thus the optimization function used in the nutrient medium is the cell to cell signalling function and cost function i.e.

$$J(i,j,k,l)+J_{CC}\left(\theta^i(j,k,l),\theta(j,k,l)\right) \quad (3)$$

Reproduction

In the reproduction stage the least healthy bacteria die and the healthy bacteria reproduce i.e. split into two. To model reproduction the accumulated cost function of each bacterium is calculated after N_C chemotactic steps as follows

$$J_{health}^i=\sum_{j=1}^{N_C+1}J(i,j,k,l) \quad (4)$$

The bacteria are then arranged in ascending order according to the accumulated cost function value. For a minimizing cost function the bacteria having higher value of accumulated cost function are the least healthy and the ones having higher values are healthier. So now the bacteria are divided into two halves by

$$S_r = \frac{s}{2} \tag{5}$$

Here least healthy S_r bacteria die and the healthiest S_r bacteria reproduce. The new bacteria are placed at the same position as their parent bacteria. This also makes sure that the population size remains constant.

Elimination and Dispersal

To model an event of calamity in the bacterial population elimination and dispersal is incorporated. This has the advantage of making sure that the bacteria do not stagnate at local optimum. Bacterial with probability lower than P_{ed} are eliminated and dispersed to new location whereas bacteria with probability higher than P_{ed} stay at their current position.

Following are some of the heuristics to be followed for bacterial foraging

1. The chemotaxis step size in the BFO is generally a small value in comparison with the search space.
2. The number of iterations for various phases can be adjusted according to the problem in hand. Generally the number of chemotactic steps is greater in comparison to other steps.
3. This algorithm has been designed to solve various optimization problems with low usage of memory.
4. In the swarming step some parameters are given default values i.e
   h_{rep}= 0.1
   d_{att}= 0.1
   w_{att}= 0.2
   w_{rep}= 10
5. The value of probability of elimination and dispersal is kept large.

3.3 Guidelines for algorithm parameter choices

1. Population size ‘S’
   As the population size increases the computational complexity of the algorithm also increases. However a large population size makes sure that at least some of the

bacteria start at the optimum and consequently the other bacteria can also reach that position by reproduction, chemaotaxis and swarming.

2. Chemotactic step size C(i)

   If the value of chemotactic step size is too high then there is a possibility that the bacteria may jump over an optimum solution and miss it. However if the step size is too small than the algorithm will converge slowly.

3. No of chemotactic steps N_C

   If the value of N_C is too small the algorithm will depend more on reproduction and luck and could get trapped at local minima.

4. Reproduction steps N_{re}

   Large value of N_{re} will cause computational complexity, on the flip side lower value may lead to premature convergence

5. Elimination and Dispersal N_{ed}

   A high value of N_{ed} indicates more elimination and dispersal for bacteria so the bacteria can look into more number of regions, however the computational complexity will increase. On the other side a lower value of N_{ed} less number of random dispersals of bacteria to find favourable regions.

Chapter 4

UNIVERSAL LAW OF GRAVITY

4.1 Introduction

According to the universal law of gravity [23] proposed by Newton every object in the universe attracts every other object with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. The direction of force is along the line centre of the two objects.

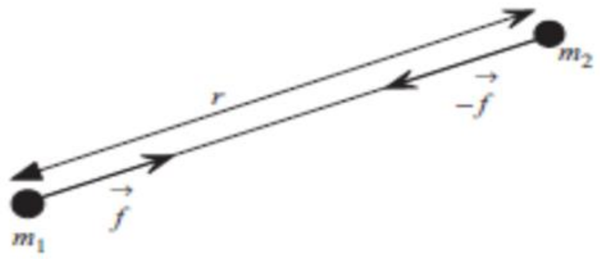


Fig 4.1 Newton's law of Universal Gravitation

According to figure 4.1

$$\overrightarrow{f_{1,2}} = \frac{Gm_1m_2\widehat{r_{2,1}}}{\|\overrightarrow{r_{2,1}}\|^2} = \frac{Gm_1m_2\overrightarrow{r_{2,1}}}{\|\overrightarrow{r_{2,1}}\|^3} \quad (6)$$

Where,

$\overrightarrow{f_{1,2}}$= force acting on object 1 because of object 2

m_1= mass of object 1

m_2= mass of object 2

$\overrightarrow{r_{2,1}}$= distance between the two objects

$\widehat{r_{2,1}}$= unit vector from object 2 to1

G= Gravitational constant

Similarly the force acting on object 2 because of object 1 can be taken as

$$\overrightarrow{f_{2,1}} = -\overrightarrow{f_{1,2}}$$

4.2 Law of gravity for images

To apply the universal law of gravity for images it is assumed that each pixel in an image acts as a celestial body. Each pixel has a relationship with its neighbouring pixels gravitational forces. For pixels beyond a specific range the force is considered to be zero.

According to [12] the following steps need to be followed to find out the force acting on a particular pixel in an image:

1. For a pixel I(i, j) consider an m X n neighbourhood *Ω* with pixels (k, l) ∈ *Ω* and (k,l) ≠(i,j). For each pixel in an image the gravitational force the pixel exerts on its neighbouring pixels is calculated according to equation 6 as given below

$$\overrightarrow{f_{i,j;k,l}} = \frac{G m_{i,j} m_{k,l} \vec{r}}{\|\vec{r}\|^3} \qquad (7)$$

$\overrightarrow{f_{i,j;k,l}}$= the force exerted by pixel (k, l) on pixel (i, j)

$m_{i,j}$= gray value of pixel I(i, j)

$m_{k,l}$= gray value of pixel I(k, l)

$\vec{r}$= Euclidian distance between the two pixels

$$\|\vec{r}\| = \sqrt{(k-i)^2 + (l-j)^2} \qquad (8)$$

2. Since images are 2D, force is calculated in x and y direction

$$f^x_{i,j;k,l} = \frac{G m_{i,j} m_{k,l} (k-i)}{\|\vec{r}\|^3} \qquad (9)$$

$f^x_{i,j;k,l}$= force on pixel (i, j) due to pixel (k, l) in x-direction

$$f^y_{i,j;k,l} = \frac{G m_{i,j} m_{k,l} (l-j)}{\|\vec{r}\|^3} \qquad (10)$$

$f^y_{i,j;k,l}$= force on pixel (i, j) due to pixel (k, l) in y direction

Therefore the value of vector $\overrightarrow{f_{i,j;k,l}}$ is computed as follows:

$$\overrightarrow{f_{i,j;k,l}} = f^x_{i,j;k,l}\hat{x} + f^y_{i,j;k,l}\hat{y} \qquad (11)$$

3. The vector sum of the gravitational forces acting on pixel (i, j) due to its neighbouring pixels given as

$$\overrightarrow{F_{i,j}} = \sum_{(k,l)\in\Omega \,\&\, (k,l)\neq(i,j)} \overrightarrow{f_{i,j;k,l}} = F^x\hat{x} + F^y\hat{y} \qquad (12)$$

Where

$$F^x = \sum f^x_{i,j;k,l} \quad (k,l) \in \Omega \,\&\, (k,l) \neq (i,j) \qquad (13)$$

$$F^y = \sum f^y_{i,j;k,l} \quad (k,l) \in \Omega \,\&\, (k,l) \neq (i,j) \qquad (14)$$

4. The final magnitude of force acting on pixel I(i, j) is formulated as follows

$$\|\vec{F}\| = \sqrt{(F_x)^2 + (F_y)^2} \qquad (15)$$

4.3 Example [12]

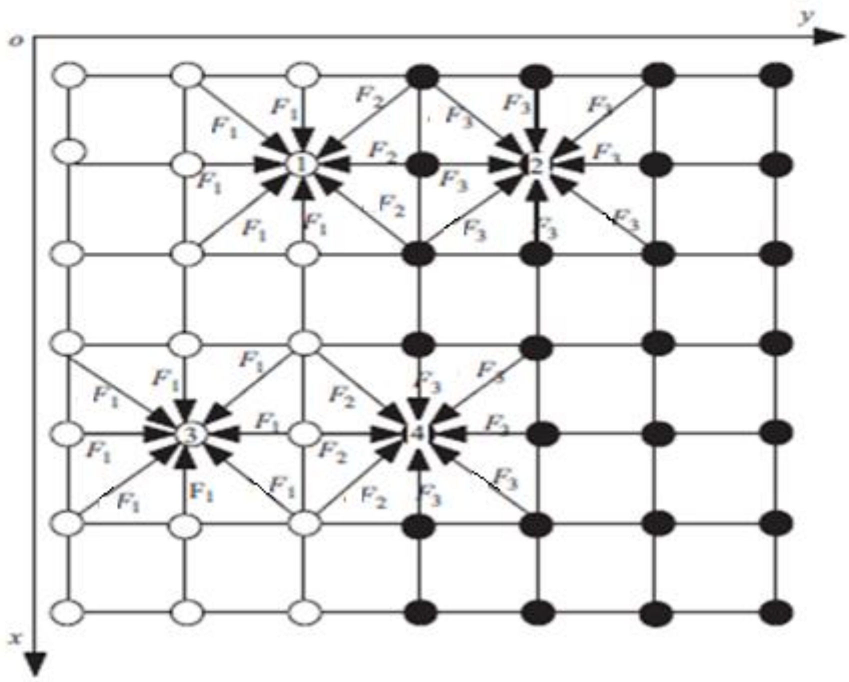


Fig 4.2 Basic edge structures

I_1= gray value of white pixel

I_2= gray value of black pixel

Since the gray value of white pixel is lower than that of black pixel, therefore, $I_2 > I_1$.

F_1= force between two white pixels per unit distance

F_2= force between black and white pixel per unit distance

F_3= Force between two black pixels per unit distance

According to equation 7-: $F_3 > F_2 > F_1$.

Now we consider the 4 pixels shown in Fig 4.2 and calculate the force acting on each of them in their 3X3 neighbourhood. Out of these pixels 2 and 3 are non-edge pixels whereas pixels 1 and 4 are latent edge pixels.

1. The force acting on pixel 4 is calculated as follow

   Using equations 9 and13

$$F_x = \left(\frac{F_3}{2\sqrt{2}} + F_3 + \frac{F_2}{2\sqrt{2}}\right) - \left(\frac{F_3}{2\sqrt{2}} + F_3 + \frac{F_2}{2\sqrt{2}}\right) = 0$$

   Using equation 10 and14

$$F_y = \left(\frac{F_3}{2\sqrt{2}} + F_3 + \frac{F_3}{2\sqrt{2}}\right) - \left(\frac{F_2}{2\sqrt{2}} + F_2 + \frac{F_2}{2\sqrt{2}}\right) = \left(\frac{1}{\sqrt{2}} + 1\right)(F_3 - F_2)$$

   The magnitude of total force acting on pixel 4 using equation 15 is

$$F^4 = \left(\frac{1}{\sqrt{2}} + 1\right)(F_3 - F_2)$$

2. The force acting on pixel 1 is calculated as follows

   Using equation 9 and 13

$$F_x = \left(\frac{F_1}{2\sqrt{2}} + F_1 + \frac{F_2}{2\sqrt{2}}\right) - \left(\frac{F_1}{2\sqrt{2}} + F_1 + \frac{F_2}{2\sqrt{2}}\right) = 0$$

   Using equation 10 and 14

$$F_y = \left(\frac{F_2}{2\sqrt{2}} + F_2 + \frac{F_2}{2\sqrt{2}}\right) - \left(\frac{F_1}{2\sqrt{2}} + F_1 + \frac{F_1}{2\sqrt{2}}\right) = \left(\frac{1}{\sqrt{2}} + 1\right)(F_2 - F_1)$$

   The magnitude of total force acting on pixel 1 using equation 15 is

$$F^1 = \left(\frac{1}{\sqrt{2}} + 1\right)(F_2 - F_1)$$

3. The force acting on pixel 2 is calculated as follows

   Using equation 9 and 13

$$F_x = \left(\frac{F_3}{2\sqrt{2}} + F_3 + \frac{F_3}{2\sqrt{2}}\right) - \left(\frac{F_3}{2\sqrt{2}} + F_3 + \frac{F_3}{2\sqrt{2}}\right) = 0$$

Using equation 10 and 14

$$F_y = \left(\frac{F_3}{2\sqrt{2}} + F_3 + \frac{F_3}{2\sqrt{2}}\right) - \left(\frac{F_3}{2\sqrt{2}} + F_3 + \frac{F_3}{2\sqrt{2}}\right) = 0$$

The magnitude of total force acting on pixel 2 using equation 15 is

$$F^2 = 0$$

4. The force acting on pixel 3 is calculated as follows

   Using equation 9 and 13

$$F_x = \left(\frac{F_1}{2\sqrt{2}} + F_1 + \frac{F_1}{2\sqrt{2}}\right) - \left(\frac{F_1}{2\sqrt{2}} + F_1 + \frac{F_1}{2\sqrt{2}}\right) = 0$$

   Using equation 10 and 14

$$F_y = \left(\frac{F_1}{2\sqrt{2}} + F_1 + \frac{F_1}{2\sqrt{2}}\right) - \left(\frac{F_1}{2\sqrt{2}} + F_1 + \frac{F_1}{2\sqrt{2}}\right) = 0$$

   The magnitude of total force acting on pixel 3 using equation 15 is

$$F^3 = 0$$

As it can be seen from above equations the force acting on non-edge pixels is 0.

Since $F_3>F_2>F_1$, therefore it can be concluded that $F^1 \neq F^4$ and $F^4 > F^1 > 0 = F^3 = F^2$. A pixel will be declared as an edge pixel if the force acting on it is greater than some threshold value i.e. F, theoretically let us assume that

$$F^4 > F > F^1 > 0 = F^3 = F^2$$

Therefore, F^4 will be labelled as an edge pixel.

From the above discussion it can be concluded that the **force acting on edge pixel is greater than that on other pixels.**

Chapter 5

PROPOSED APPROACH

5.1 APPROACH USED

The proposed approach uses universal law of gravity in the chemotaxis stage of the bacterial foraging algorithm to determine the direction of movement of bacteria.

5.1.1 Direction of movement using universal law of gravity

The procedure to find the direction of movement of bacteria to find the edge pixels and the nutrient concentration is now explained. As already explained in chapter 4 the force acting on edge pixels is greater than that on other pixels and should be greater than some threshold value. Therefore the nutrient concentration η on pixel (x, y) will be considered as follows

$$\eta_{xy} = \begin{cases} F_{xy} & F_{xy} > \lambda \\ 0 & otherwise \end{cases} \quad (16)$$

Where

F_{xy} =force on pixel (x, y) calculated using 9, 10, 13, 14, 15 as given in Chapter 4.

η_{xy}= nutrient concentration at pixel (x, y)

λ= threshold value for a pixel to be considered as an edge pixel.

Consider a 3x3 neighbourhood of pixel as shown in figure 5.1

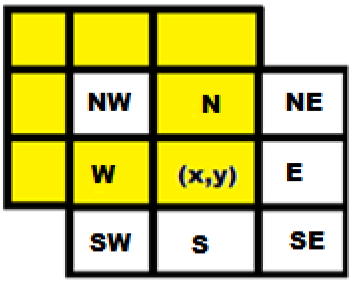


Fig 5.1 Pixels and their 3X3 neighbourhood

Suppose the bacterium is at position (x, y).

Calculate the nutrient concentration on each of the neighbouring pixels of (x, y) using the equation 16

For example the nutrient concentration on pixel NW will be due to the pixels coloured yellow and let it be called η_{NW}.

Similarly the nutrient concentration on other pixels accordingly will be η_N, η_{NE}, η_{SW}, η_{SE}, η_S, η_W, η_E. Now since the force acting on edge pixel is greater than the rest of the pixels. Let

$$\eta = \max(\eta_N, \eta_{NE}, \eta_{SW}, \eta_{SE}, \eta_S, \eta_W, \eta_{NW}, \eta_E) \tag{17}$$

Now if $\eta = 0$ it is sure that no edge pixel exists in the 3X3 neighbourhood of the bacteria. On choosing one of the eight directions in this situation randomly for the movement of the bacteria a non- edge will be marked as an edge pixel. To tackle this situation while choosing the direction of tumble if $\eta = 0$ then skip the tumbling and swimming of bacteria and disperse the bacteria to a random location for the next chemotactic step ,otherwise if $\eta \neq 0$ move the bacteria to position having value η .

Since the bacteria foraging mostly deals with finding the minimum value of the cost function J(Θ), therefore the cost function will be the inverse of the nutrient concentration acting on the pixel at which the bacteria is present

$$J(\Theta) = 1/\eta(\Theta) \tag{18}$$

Where Θ= position of the bacteria

Hence higher the force acting on a pixel, higher will be the nutrient concentration and more will be the movement of bacteria towards it.

5.2 ALGORITHIM

Initialize the parameters:

p: dimension of the search space=2,

S: the number of bacteria in the community iterated by variable i

S_r: bacteria split ratio,

N_c: number of chemotactic steps iterated by variable j

N_{re}: the number of reproduction steps iterated by variable k

N_{ed}: the number of elimination-dispersal events iterated by variable l

N_s:swim length iterated by variable m

P_{ed}: elimination-dispersal probability

C(i): the size of the step taken in the direction specified by the tumble

θ^i (1, 1, 1): initial positions of the bacterium selected randomly.

J(i, 1, 1, 1): Initialized value at the pixel given by θ^i (1, 1, 1) calculated by nutrient concentration on the pixel

[Step 1] Elimination-dispersal loop: l = l + 1

[Step 2] Reproduction loop: k= k + 1

[Step 3] Chemotaxis loop: j=j+1

[a] For *i* =1,2…S take a chemotactic step for bacterium *i* as follows.

[b] Compute fitness function, J(i, j, k, l)

Let, J(i, j, k, l) = J(i, j, k, l) + Jcc(θ^i (j, k, l), θ (j, k, l))

[c] Let J_{last}= J(i, j, k, l) used for updating value of cost function in case of a more appropriate solution.

[d] η =max $(\eta_N, \eta_{NE}, \eta_{SW}, \eta_{SE}, \eta_S, \eta_W, \eta_{NW}, \eta_E)$ using equation 16 and 17

if($\eta \neq 0$)

[e] Tumble: Find the directions of possible movement

[f] Move: Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(i)$$

$\phi(i)$= direction of movement towards pixel having value η.

This results in a step of size C(i) in the direction of the tumble for bacterium.

[g] Compute J(i, j + 1, k, l)

J(i, j+1, k, l) = J(i, j+1, k, l) +J_{cc}((θ^i (j+1, k, l), θ (j+1, k, l))

[h] Swim

(i) Let m = 0

(ii) While m < Ns

- Let m = m + 1.
- If J(i, j + 1, k, l) < Jlast

  update θ^i (j+1, k, l)as in step 3(e).

  Use this θ^i (j + 1,k, l) to compute the new J(i, j + 1, k, l) as in step 3[g]

- Else m = m + 1
- End if

else

Displace the bacteria to a random location for next chemotactic step

θ^i (j+1, k, l)=rand_post()

end if

[i] If i+1≠S move to the next bacteria

[Step 4] If j < Nc, go to Step 3 since the life of bacteria is not yet finished.

[Step 5] Reproduction:

[a] For each i = 1, 2, . . . , S, let

$$J_{health}^{i} = \sum_{j=1}^{N_C+1} J(i,j,k,l)$$

be the health of the bacterium

Sort bacteria and C(i) in the order of increasing cost J_{health} (higher cost means lower health).

[b] The S_r bacteria with the highest J_{health} values die and the remaining S_r bacteria with the best values split . The new bacteria are placed at the same position as their parent bacteria.

[Step 6] If k < Nre, go to Step 2, since the specified number of reproduction steps have not been reached.

[Step 7] Elimination-dispersal: For i = 1, 2, ... , S, eliminate and disperse each bacterium with probability P_{ed}. If a bacterium is eliminated, simply disperse another bacterium to a random location in the search space. If l< N, then go to Step 1; otherwise end.

5.3 FLOWCHART

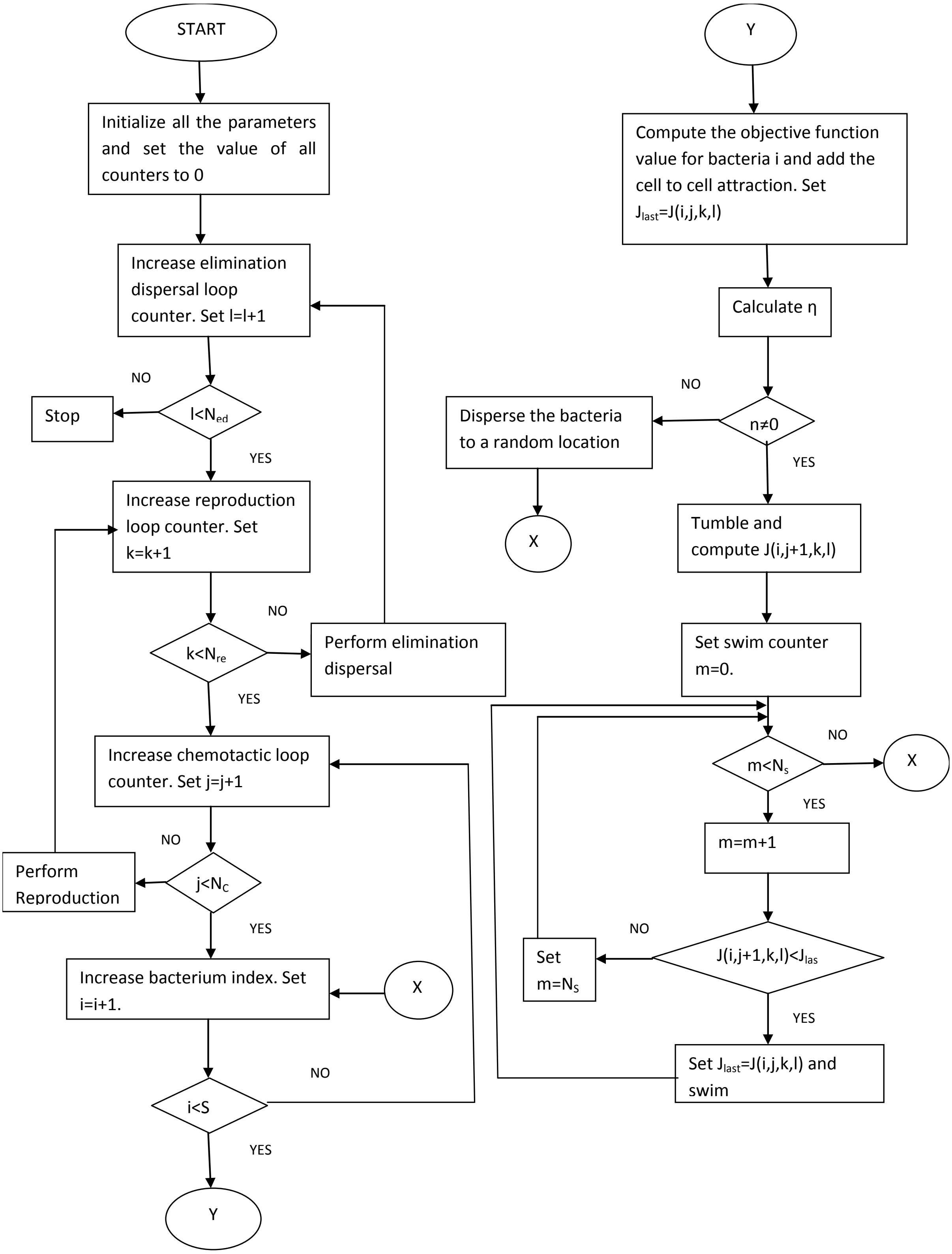


Fig 5.2 Flowchart of proposed approach

Chapter 6

RESULTS AND DISCUSSION

The following system configuration has been used while conducting the experiments:

Processor: Intel Core i5

Clock Speed: 2.3 GHz

Main memory: 4 GB

Hard Disk Capacity: 750 GB

Software Used: MATLAB 7.10 (R2010a)

This section shows the implementation results as well as the quantitative results of the proposed approach. The computer simulations are performed using MATLAB to access the performance of proposed algorithm on five test images viz Lena, Cameraman, Peppers, and House with different features to find out the edges.

6.1 Experimental Results

The results of the proposed edge detection technique are compared to the commonly used edge detectors viz Canny, Roberts, Sobel and Prewitt. These traditional edge detectors have been implemented using MATLAB Toolbox.

The various parameters used in the algorithm has been set as follows

P=2

S=800

S_r= S/2

N_S=10

P_{ed}=0.95

N_{ed}=10

N_{re}=1

N_c=200

C(i)=1

h_{rep}= 0.1

d_{att}= 0.1

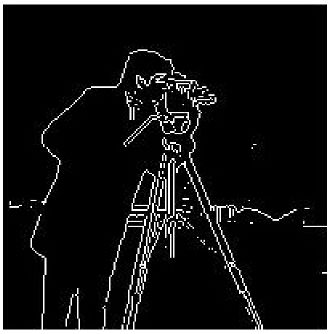
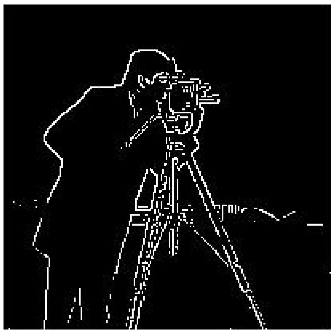
w_{att}= 0.2

w_{rep}= 10

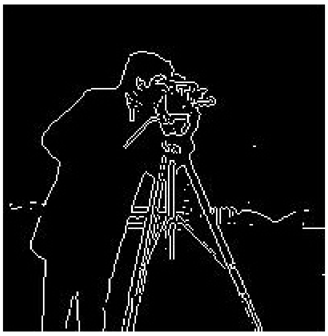
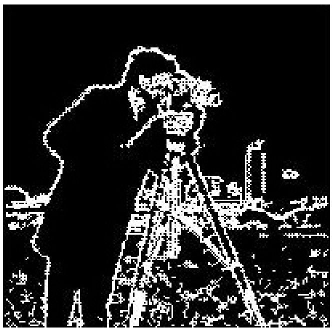
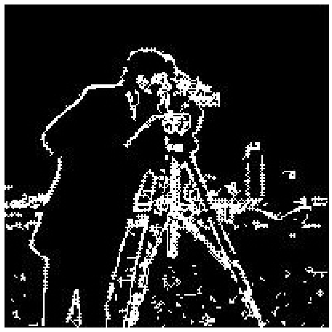
λ= It is found out experimentally depending on the image used

The upcoming figures show the result of the traditionally used edge detectors viz. Canny, Prewitt, Robert, Prewitt and the proposed approach on some commonly used images.

(a)

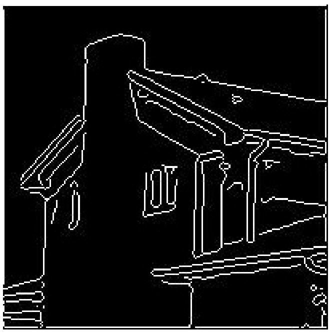
(b) (c) (d)

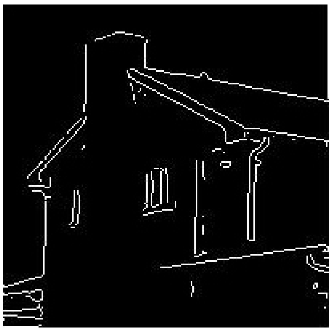
  

(e) (f) (g)

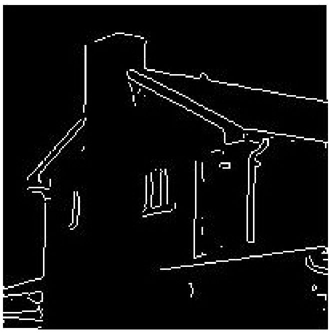
FIG. 6.1 Result 1 (a) Original Cameraman image (b) Canny edge detector (c) Prewitt edge detector (d) Robert edge detector (e) Sobel edge detector (f) Proposed approach (threshold=0.115) and (g) Proposed approach (threshold= 0.15)

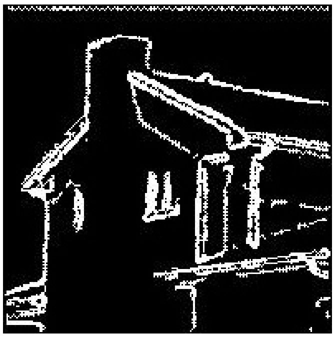
(a)

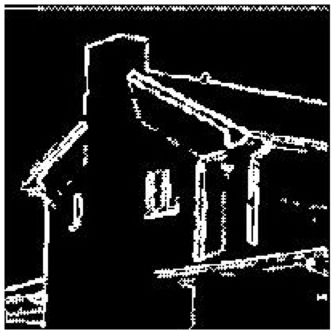




(b) (c) (d)







(e) (f) (g)

FIG. 6.2 Result 2 (a) Original House image (b) Canny edge detector (c) Prewitt edge detector (d) Robert edge detector (e) Sobel edge detector (f) Proposed approach (threshold=0.1) and (g) Proposed approach (threshold= 0.115)

(a)

(b)

(c)

(d)

(e)

(f)

(g)

FIG. 6.3 Result 3 (a) Original Lena image (b) Canny edge detector (c) Prewitt edge detector (d) Robert edge detector (e) Sobel edge detector (f) Proposed approach (threshold=0.115) and (g) Proposed approach (threshold= 0.15)

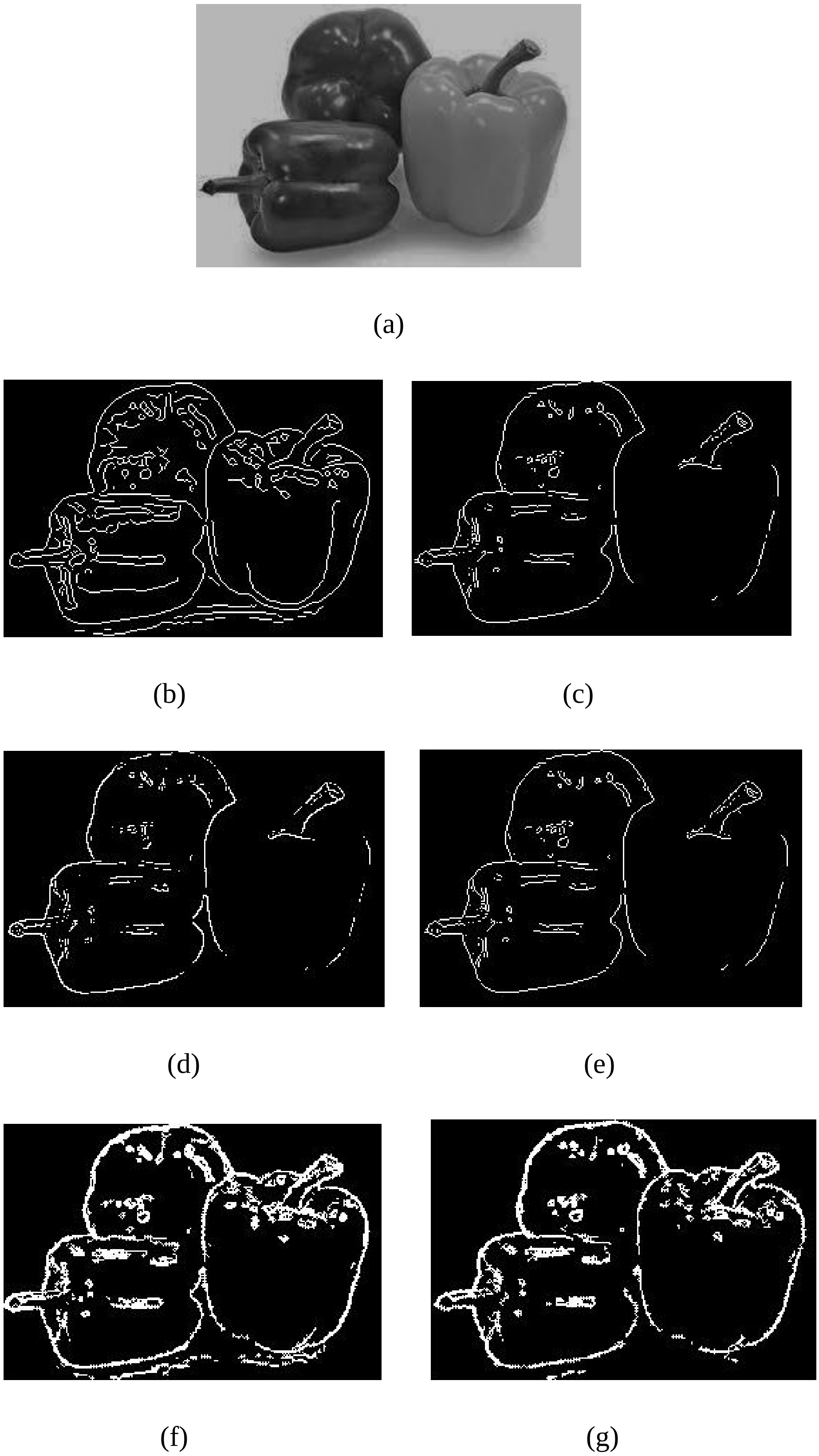


FIG. 6.4 Result 4 (a) Original Pepper image (b) Canny edge detector (c) Prewitt edge detector (d) Robert edge detector (e) Sobel edge detector (f) Proposed approach (threshold=0.115) and (g) Proposed approach (threshold= 0.14)

It can be concluded from figure 6.1 that the background edges are being detected by the proposed method which is not being detected by Prewitt, Sobel, and Robert (see figure 6.1(c), 6.1(d), 6.1(e)). In figure 6.1(f) and 6.1(g) the edges of the buildings are visible. However 6.1(f) is noisier than 6.1(g) due to the lower value of threshold being used. Similarly on comparing figure 6.3(c), 6.3(d), 6.3(e) with figure 6.3(f) we can see that the edges detected by proposed method are better. Edges near Lena hair, bottom right corner and top right corner are being detected by the proposed method. Also by comparing the images of house (figure 6.2) and pepper (figure 6.4) we can see that meaningful edges are being detected by the proposed approach. It can be seen that the proposed approach provides much finer details than Prewitt, Robert and Sobel.

6.2 Cohen's Kappa

Cohen proposed a scalar parameter called Cohen's Kappa [24] which measures accuracy. It was initially introduced for problems in the medical world but since then it has been extended further. When two observers are observing the same event it measures the degree of agreement between the two of them and also compensates for chance agreements.

It is written as

$$K = \frac{P_0 - P_C}{1 - P_C}$$

Where
P_0= probability of total agreement
P_C= probability of chance agreement

The kappa value is a measure of accuracy for pixel to pixel comparison between two images, thus it shows percentage match of the image; we consider the combination of all images to compare the result. The value of Kappa lies between [0, 1]. The value of 1 means perfect agreement and 0 means no agreement. It can be seen the proposed approach has good kappa values.

TABLE 6.1

COHEN'S KAPPA VALUES

Image Used	Value of threshold	Kappa Value				
		Proposed approach	Canny	Prewitt	Robert	Sobel
Lena	0.115	0.6478	0.4222	0.1923	0.2232	0.1942
	0.15	0.5598	0.5263	0.2173	0.2554	0.2197
Cameraman	0.115	0.6726	0.3781	0.2077	0.2283	0.2078
	0.15	0.6068	0.4681	0.2245	0.2520	0.2248
House	0.1	0.7116	0.3177	0.2183	0.2766	0.2180
	0.115	0.6970	0.3403	0.2157	0.2816	0.2157
Pepper	0.115	0.7096	0.3731	0.2135	0.2304	0.2130
	0.14	0.6731	0.4172	0.2146	0.2513	0.2145

6.3 Shannon's Entropy

In the field of scientific theory Shannon's Entropy measures the degree of uncertainty in a system. It calculates the amount of information content in the resultant image by measuring indefiniteness.

Its formula is as follows [17]

$$H(I) = \sum_{i=0}^{L} p_i \log p_i$$

Where

I=image under consideration

p_i= number of pixels with intensity *i*

Higher the value of entropy greater is the randomness present in an edge image so the image is not good. Moreover sometimes low entropy value may be shown when the edges are incomplete which is not good and should be observed in case the entropy is low. Visual analysis of image can help in this state of confusion. The proposed approach has higher entropy values but its result are visually better than Prewitt, Robert and Sobel.These detectors have less entropy values because they give less edge information.

TABLE 6.2

SHANNON'S ENTROPY VALUES

Image Used	Entropy value of							
	Proposed Image at threshold value				Canny	Prewitt	Robert	Sobel
Lena	0.115	0.6117	0.15	0.4769	0.4297	0.2225	0.1982	0.2240
Cameraman	0.115	0.6824	0.15	0.5658	0.4258	0.2292	0.2306	0.2281
House	0.115	0.5563	0.1	0.5892	0.3318	0.2202	0.2153	0.2209
Pepper	0.115	0.5681	0.14	0.5011	0.3159	0.1764	0.1750	0.1761

Chapter 7

CONCLUSION

A novel method is developed for edge detection in images based on Newtonian law of gravity and Bacterial Foraging Algorithm to detect edges in images. This approach takes advantage of the fact that the force acting on edge pixels is always greater than the non-edge pixels. In this method for the chemotaxis step of the bacterial foraging algorithm the direction in which the bacteria has to move is found out making use of the force acting on neighbouring pixels using Newtonian theory of gravity.

We have also performed qualitative and quantitative analysis of the proposed method on different images such as Lena, cameraman, house etc. and compared it with some of the already existing methods in literature. The quantitative analysis has been performed using entropy and kappa measures.

From experimental results we can say that the proposed approach detects most of the edges in an image is better than some edge detectors such as Sobel, Robert, and Prewitt. The combined use of two optimization algorithms makes this approach appropriate for detection of edges in most of the images.

REFERENCES

1. R. C. Gonzalez and R. E. Wood, "Digital Image Processing", 3rd Ed., Prentice Hall, 2009.
2. Sobel I. Camera models and machine perception. Stanford Univ Calif Dept of Computer Science; 1970 May.
3. Prewitt, Judith MS. "Object enhancement and extraction." Picture processing and Psychopictorics 10.1 (1970): 15-19.
4. Rosenfeld, Azriel. "The max Roberts operator is a Hueckel-type edge detector." Pattern Analysis and Machine Intelligence, IEEE Transactions on 1 (1981): 101-103.
5. Canny, John. "A computational approach to edge detection." Pattern Analysis and Machine Intelligence, IEEE Transactions on 6 (1986): 679-698
6. Juneja, Mamta, and Parvinder Singh Sandhu. "Performance evaluation of edge detection techniques for images in spatial domain." international journal of computer theory and Engineering 1.5 (2009): 614.].
7. Dorigo, Marco, Mauro Birattari, and Thomas Stützle. "Ant colony optimization." Computational Intelligence Magazine, IEEE 1.4 (2006): 28-39.
8. Dorigo, Marco, Mauro Birattari, and Thomas Stuetzle. "Ant colony optimization-Artificial ants as a computational intelligence technique." IEEE Computational Intelligence Magazine 1.4 (2006): 28-39.
9. Rajeswari, R., 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. IEEE, 2011.
10. Lu, De-Sian, and Chien-Chang Chen. "Edge detection improvement by ant colony optimization." Pattern Recognition Letters 29.4 (2008): 416-425.
11. Om Prakash, Prerna Singhal, Sakshi Garg, and Deepti Singh Chauhan. "Edge detection using adaptive thresholding and ant colony optimization." In Information and Communication Technologies (WICT), 2011 World Congress on, pp. 313-318. IEEE, 2011.
12. Ari Samit, Dipak Kumar Ghosh, and Prashant Kumar Mohanty. "Edge detection using ACO and F ratio." Signal, Image and Video Processing 8, no. 4 (2014): 625-634.
13. Verma, Om Prakash, M. Hanmandll, Puneet Kumar, and Shivangi Srivastava. "A Novel Approach for Edge Detection using AntColony Otimization and Fuzz

Derivative Technique." In Advance Computing Conference, 2009. IACC 2009. IEEE International, pp. 1206-1212. IEEE, 2009.

14. Sun Genyun, Qinhuo Liu, Qiang Liu, Changyuan Ji, and Xiaowen Li. "A novel approach for edge detection based on the theory of universal gravity." Pattern Recognition 40, no. 10 (2007): 2766-2775.
15. Verma Om Prakash, Rishabh Sharma, Manoj Kumar, and Neetu Agrawal. "An Optimal Edge Detection Using Gravitational Search Algorithm." Lecture Notes on Software Engineering 1, no. 2 (2013): 148.
16. Lopez-Molina, Carlos, H. Bustince, Javier Fernández, Pedro Couto, and Bernard De Baets. "A gravitational approach to edge detection based on triangular norms." Pattern Recognition 43, no. 11 (2010): 3730-3741.
17. Verma Om Prakash, Madasu Hanmandlu, Puneet Kumar, Sidharth Chhabra, and Akhil Jindal. "A novel bacterial foraging technique for edge detection." Pattern recognition letters 32, no. 8 (2011): 1187-1196.
18. Verma Om Prakash, Rishabh Sharma, and Deepak Kumar. "Binarization based image edge detection using bacterial foraging algorithm." Procedia Technology 6 (2012): 315-323.
19. Passino, Kevin M. "Biomimicry of bacterial foraging for distributed optimization and control." Control Systems, IEEE 22.3 (2002): 52-67.
20. Hanmandlu, Madasu, Om Prakash Verma, Nukala Krishna Kumar, and Muralidhar Kulkarni. "A novel optimal fuzzy system for color image enhancement using bacterial foraging." Instrumentation and Measurement, IEEE Transactions on 58, no. 8 (2009): 2867-2879.
21. Mishra S. “A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation.” Evolutionary Computation, IEEE Transactions on. 2005 Feb;9(1):61-73.
22. Adler, Julius. "Chemotaxis in bacteria." Science 153.3737 (1966): 708-716.
23. csep 10.phys .utk .edu /astr161 /lect/ history /newtongrav.html
24. Cohen, J.:"A coefficient of agreement for nominal scales". Educ. Psychol. Meas. 20(1), 37–46 (1960)