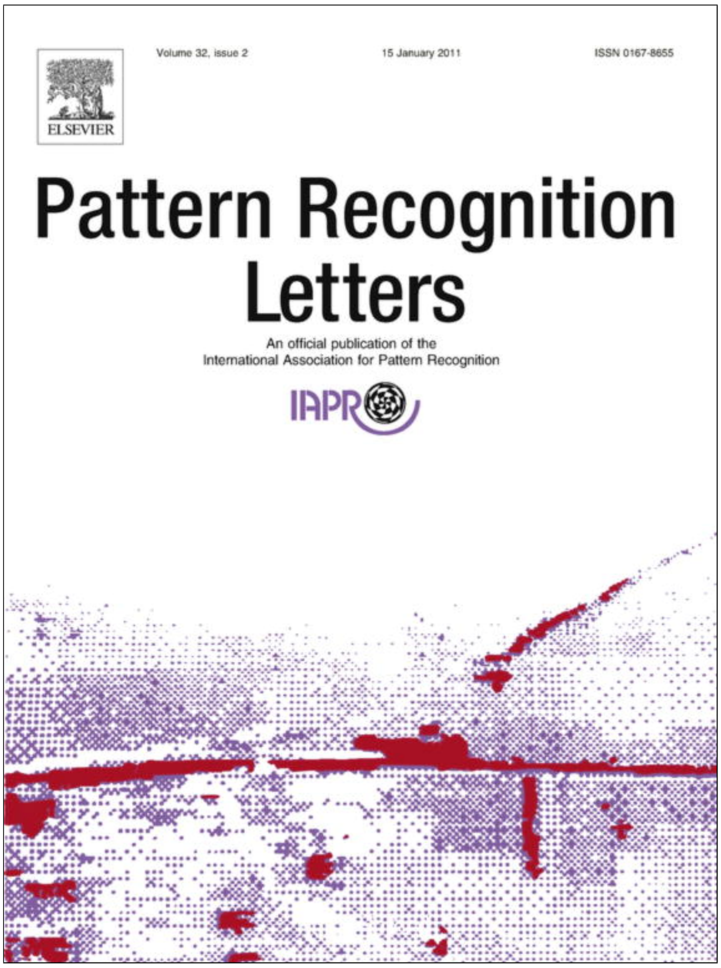

Volume 32, issue 2
15 January 2011
ISSN 0167-8655
ELSEVIER
Pattern Recognition Letters
An official publication of the International Association for Pattern Recognition
IAPR

(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Pattern Recognition Letters 32 (2011) 1187–1196

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A novel bacterial foraging technique for edge detection

Om Prakash Verma [a,*], Madasu Hanmandlu [b], Puneet Kumar [c], Sidharth Chhabra [a], Akhil Jindal [a]

[a] Delhi Technological University (Formerly Delhi College of Engineering), Delhi, India
[b] Department of Electrical Engineering, IIT Delhi, Delhi, India
[c] Advanced Systems Laboratory, Hyderabad, India

ARTICLE INFO

Article history:
Received 11 February 2010
Available online 12 March 2011
Communicated by H.H.S. Ip

Keywords:
Ant Colony System
Bacterial foraging
Derivative
Direction probability matrix
Edge detection

ABSTRACT

A new approach for edge detection using a combination of bacterial foraging algorithm (BFA) and probabilistic derivative technique derived from Ant Colony Systems, is presented in this paper. The foraging behavior of some species of bacteria like *Escherichia coli* can be hypothetically modeled as an optimization process. A group of bacteria search for nutrients in a way that maximizes the energy obtained per unit time spent during the foraging. The proposed approach aims at driving the bacteria through the edge pixels. The direction of movement of the bacteria is found using a direction probability matrix, computed using derivatives along the possible directions. Rules defining the derivatives are devised to ensure that the variation of intensity due to noise is discarded. Quantitative analysis of the feasibility of the proposed approach and its comparison with other standard edge detection operators in terms of kappa and entropy are given. The effect of initial values of parameters of BFA on the edge detection is discussed.

1. Introduction

Detection of edges in an image is a very important step for the image understanding. Indeed, high-level processing tasks such as image segmentation and object recognition, etc. directly depend on the quality of the edges detected. Moreover, the generation of an accurate edge map becomes a very critical issue when the images are corrupted by noise.

Edges in an image are marked with discontinuity or significant variation in intensity or gray levels. The methods of identifying the intensity discontinuity associated with edges in an image are normally based on the calculation of the intensity gradient in the whole image. The underlying idea of most edge detection techniques is the computation of a local first or second derivative operator, followed by some regularization technique to reduce the effects of noise.

Canny' method (Canny, 1986) for the edge detection counters the noise problems, wherein an image is convolved with the first-order derivatives of Gaussian filter for smoothing in the local gradient direction followed by the edge detection using thresholding. Marr and Hildreth (1980) propose an algorithm that finds edges at the zero-crossings of the image Laplacian. Non-linear filtering techniques for edge detection also have witnessed much advancement through the SUSAN method (Smith and Brady, 1997), which works by associating a small area of neighboring pixels with similar brightness to each center pixel. Existing edge detectors like Gradient operator and the Laplacian Operator are based on the assumption that edges in the images are step functions in intensity. Prewitt detector (Raman and Sobel, 2006) uses the local gradient operators which only detect edges having certain orientations and perform poorly on blurred or noisy images. Different algorithms for the fuzzy based edge detection are proposed in Cheung and Chan (1995), Kuo et al. (1997), Russo (1998), El-Khamy and S. (2000). Abdallah and Ayman (2009) introduce a fuzzy logic reasoning strategy for the edge detection in the digital images without determining a threshold.

Most of the existing operators are confronted with a huge search space for the detection of edges. Considering an image of size 1024 by 1024 pixels, the required solution space is of the order of $2^{1024*1024}$. Therefore the task of edge detection is time consuming and memory exhausting without the optimization.

A novel bacterial-derivative based algorithm that exploits the foraging behavior of bacteria to collectively detect edges in an image is developed in Passino (2002), Liu and Passino (2002). Bacterial foraging optimization algorithm (BFAO) has already been applied in the optimal control engineering, harmonic estimation (Mishra, 2005), transmission loss reduction (Tripathy et al., 2006), machine learning (Kim and Cho, 2005), active power filter design (Mishra and Bhende, 2007), color image enhancement (Hanmandlu et al., 2009), etc.

Bacteria foraging is an optimization process where bacteria seek to maximize the energy by eating up as many nutrients as they can. Nutrients in our case correspond to tracing the edge pixels. Bacteria move by either tumbling or swimming. In the classical approach, the direction of movement is decided randomly while

[*] Corresponding author. Tel.: +91 1127294672; fax: +91 1127871023.
E-mail addresses: opverma@dce.ac.in, opverma.dce@gmail.com (O.P. Verma).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.03.008

tumbling and every direction is equally preferred. In our method, a probabilistic approach, inherited from Ant Colony System (Dorigo and Gambardella, 1997; Verma et al., 2009), is used. This causes the bacterium to move along the direction, where the probability of finding a nutrient (edge) is highest. The proposed algorithm also distinguishes between the local variations due to noise and image structures, using a derivative. Another important characteristic of bacteria's life cycle is swarming. In our approach, swarming is not only dependent upon other bacteria's positions but also on the clique of its current position.

The proposed approach is well suited to address the uncertainty introduced while extracting the edge information from the image data. The innovative aspect of this approach lies in the development of a noise-protected operator that combines the rules framed for the noise cancellation and edge detection in the same structure. The application of modified bacterial foraging in combination with a derivative approach leads to a minimal set of input data to be processed thus making the process faster and memory-efficient. As a result, the proposed approach outperforms the existing state-of-the-art techniques.

The paper is organized as follows: In Section 2 a brief introduction to bacterial foraging technique is provided to present the basic idea. A modification of this technique is discussed in Section 3. Section 4 presents the probabilistic derivative approach to find the direction of movement of a bacterium. An algorithm for the edge detection along with the pseudo code is developed in Section 5. Section 6 gives the experimental results followed by the conclusions in Section 7.

2. Bacterial foraging technique

A new evolutionary technique, called bacterial foraging scheme appeared in Passino (2002), Liu and Passino (2002). Foraging can be modeled as an optimization process where bacteria seek to maximize the energy obtained per unit time spent during foraging. In this scheme, an objective function is posed as the effort or a cost incurred by the bacteria in search of food. A set of bacteria tries to reach an optimum cost by following four stages: Chemo taxis, swarming, reproduction, and elimination and dispersal. To start with, there will be as many solutions as the number of bacteria. So, each bacterium produces a solution iteratively for a set of optimal values of parameters. Gradually all the bacteria converge to the global optimum.

In the chemo taxis stage, the bacteria either resort to a tumble followed by a tumble or make a tumble followed by a run or swim. This is the movement stage of bacteria accomplished through swimming and tumbling. On the other hand, in swarming, each *Escherichia* coli bacterium signals another bacterium via attractants to swarm together. This is basically the cell to cell signaling stage. Furthermore, in the reproduction the least healthy bacteria die and of the healthiest, each bacterium splits into two bacteria, which are placed at the same location. While in the elimination and dispersal stage, any bacterium from the total set can be either eliminated or dispersed to a random location during the optimization. This stage prevents the bacteria from attaining the local optimum.

Let θ be the position of a bacterium and $J(\theta)$ be the value of the objective function, then the conditions $J(\theta) < 0$, $J(\theta) = 0$, and $J(\theta) > 0$ indicate whether the bacterium at location θ is in nutrient-rich, neutral, and noxious environments, respectively. Basically, chemo taxis is a foraging behavior that implements a type of optimization where bacteria try to climb up the nutrient concentration (find the lower values of $J(\theta)$), avoid noxious substances, and search for ways out of neutral media (avoid being at positions θ where $J(\theta) \geqslant 0$). This is just like a type of biased random walk.

3. The modified bacterial foraging technique for edge detection

The original bacterial foraging (BF) technique Liu and Passino (2002) is now modified to make it suitable for the edge detection. The nutrient concentration at each position, i.e. the cost function is calculated using a derivative approach. The implications in lieu of modifications of the technique for the edge detection are furnished here.

3.1. Search space

The 2-dimensional search space for bacteria consists of the *x* and *y*-coordinates (i.e. discrete values) of a pixel in an image. Being limited by the image dimensions, i.e. horizontal and vertical pixels of the image, the search space is finite.

3.2. Chemotaxis

This is a very important stage of BF. It decides the direction in which the bacterium should move. Depending upon the rotation of the flagella, each bacterium decides whether it should swim (move in a predefined direction) or tumble (move in an altogether different direction). Our goal is to let the bacterium search for the edge pixels in an image. Another important goal is to keep the bacterium away from the noisy pixels.

As we are dealing with 2D discrete values of coordinates in an image with 8-connectivity of the neighborhood pixels, the probable directions to move for a bacterium from a particular pixel are: E, W, N, S, NE, SE, NW, SW. Out of these eight directions the bacterium of interest has to decide one direction that lead to an edge pixel but not a noisy pixel. A probabilistic derivative approach is used to find out the direction (one out of the eight possible directions) most suitable to hit upon an edge and to cut off the directions leading to noise. This is a major deviation in the chemotaxis step of BFA where the bacteria either tumble in a random direction or swim in the same direction as in the previous step. This is elaborated here.

Let $\theta^i(j,k,l)$ represent the position of the *i*th bacterium at *j*th chemotactic, kth reproductive and *l*th elimination-dispersal step. A pixel position in an image can be represented by the *x* and *y* -co-ordinates of the bacterium. So let $\theta^i(j,k,l)$ be the position of a bacterium in an array $\varphi[m,n,i,j,k,l]$ where *m*, *n* represent the *x*–*y* coordinates of the bacterium.

The initial positions of the bacteria are selected randomly. They move on to the edge pixels during the run of the algorithm. The path is recorded only after a certain number of steps made by each bacterium.

The movement of the bacterium may be represented by

$$\theta^i(j+1,k,l) = \phi[m',n',i,j+1,k,l], \tag{1}$$

where m', n' are the coordinates to which the bacterium should move in order to reach the edge pixel by avoiding the noisy pixels. During the tumble, the direction is determined from

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + \frac{C(i)\cdot\Delta(i)}{\sqrt{\Delta(i)^T\Delta(i)}} \tag{2}$$

where $\Delta(i)$ indicates a random number in $\mathscr{R}^2$ and $C(i)$ is the length of a step size.

The above approach is modified to include a probabilistic derivative as explained in detail in Section 4.

3.3. Swarming

It is assumed that a bacterium relies on other bacteria. This property of bacterium is exploited here. In this step, the bacterium that has searched an optimum path, signals other bacteria so that they can together reach the desired optimum path swiftly. As each bacterium moves, it releases an attractant to signal other bacteria to swarm towards it. Also, each bacterium releases a repellent to warn other bacteria by keeping a safe distance from them. Because of this, bacteria congregate into groups and move in a concentric pattern. This is achieved by using a cell to cell signaling function which combines both the attraction and repelling effects. Thus we have

$$\begin{aligned} J_{cc}(\theta^i(j,k,l),\theta(j,k,l)) &= \sum_{t=1}^{s} j_{cc}^{t}(\theta^i,\theta^t) \\ &= \sum_{t=1}^{s}\left[-d_{att}\exp\left(-w_{att}\sum_{m=1}^{P}\left(\theta_m^i-\theta_m^t\right)^2\right)\right] \\ &\quad+\sum_{t=1}^{s}\left[h_{rep}\exp\left(-w_{rep}\sum_{m=1}^{P}\left(\theta_m^i-\theta_m^t\right)^2\right)\right] \end{aligned} \tag{3}$$

where θ^i is the location of ith bacterium, P is the number of dimensions of the optimization domain (here, $P = 2$), $\theta = \{\theta^i | i = 1,\ldots,S\}$ represents the positions of the ith bacteria, θ_m^i is the mth component of θ^i, d_{att} is the measure of how much attractant is released, w_{att} is the diffusion rate, h_{rep} and w_{rep} are the magnitude and width of the repelling effect. Empirically, $d_{att} = 0.1$, $w_{att} = 0.2$, $h_{rep} = 0.1$, $w_{rep} = 10$ are found to be optimal.

It may be noted from Eq. (3) that this is basically a function of distance of the bacterium under consideration from all other bacteria. This function is now modified to make it dependent on the clique of the position (pixel) of the bacterium and its distance from other bacteria.

$$J_{cc}(\theta^i(j,k,l),\theta(j,k,l)) = f(distance) + f(clique) \tag{4}$$

The clique is defined as the local group of pixels around the pixel of interest. It is represented as

$$f(clique) = \mu\Delta_{\theta^i} \tag{5}$$

where μ is a constant (i.e. 1). This is introduced to make the value of cell to cell signaling function negative as the bacterium reaches an edge pixel, and Δ_{θ^i} is the function that uses the local intensity difference statistic at the pixel location given by $\theta^i \cdot \Delta_{\theta^i}$:

$$\Delta_{\theta^i} = \frac{V_c(I_{\theta^i})}{Z} \tag{6}$$

where, I_{θ^i} represents the pixel of interest in the image. If the pixel location is at (m, n) then

$$\begin{aligned} V_c(I_{\theta^i}) = V_c(I_{m,n})&|I_{m-2,n-1}-I_{m+2,n+1}| + |I_{m-2,n+1}-I_{m+2,n-1}| \\ &+|I_{m-1,n-2}-I_{m+1,n+2}| + |I_{m-1,n-1}-I_{m+1,n+1}| \\ &+|I_{m-1,n}-I_{m+1,n}| + |I_{m-1,n+1}-I_{m-1,n-1}| \\ &+|I_{m-1,n+1}-I_{m-1,n-2}| + |I_{m,n-1}-I_{m,n+1}| \end{aligned} \tag{7}$$

and Z is the normalization factor defined as

$$Z = \sum_{m=1:M}\sum_{n=1:N} V_c(Im,n) \tag{8}$$

The modified form of cell to cell signaling for edge detection may be represented as

$$\begin{aligned} J_{cc}(\theta^i(j,k,l),\theta(j,k,l)) &= \sum_{t=1}^{s} j_{cc}^{t}(\theta^i,\theta^t) \\ &= \sum_{t=1}^{s}\left[-d_{att}\exp\left(-w_{att}\sum_{m=1}^{P}\left(\theta_m^i-\theta_m^t\right)\right)\right] \\ &\quad+\sum_{t=1}^{s}\left[h_{rep}\exp\left(-w_{rep}\sum_{m=1}^{P}\left(\theta_m^i-\theta_m^t\right)^2\right)\right] \\ &\quad-\mu\frac{V_c(I_{\theta^i})}{Z} \end{aligned} \tag{9}$$

$J_{cc}(\theta^i(j, k, l), \theta(j, k, l))$ has its initial value the derivative at the pixel given by $\theta^i(j, k, l)$. The calculation of the derivative at a pixel (x, y) is explained in Section 4.1.

3.4. Reproduction step

After N_c chemotactic steps, a reproduction step begins its loop. Let N_{re} be the number of reproduction steps. For convenience, we assume that the number of bacteria S is a positive even integer. We choose

$$S_r = \frac{S}{2} \tag{10}$$

as the number of population members having sufficient nutrients so that they will reproduce (split up into two) with no mutations. For reproduction, the population is sorted in the ascending order of the accumulated cost (the higher cost indicates that a bacterium has not got as many nutrients during its lifetime of foraging and hence is not "healthy" and thus unlikely to reproduce); then the least healthy bacteria S_r die and each of the other healthiest bacteria S_r split up into two, which are placed at the same location. This method rewards bacteria that have encountered a lot of nutrients and allows us maintain a constant population size.

3.5. E. Elimination step

Let N_{ed} be the number of elimination-dispersal events and each bacterium in the population is subjected to elimination-dispersal with a probability P_{ed}. We assume that the frequency of chemotactic steps is greater than the frequency of reproduction steps, which is in turn greater than the frequency of elimination-dispersal events (e.g. a bacterium will take many chemotactic steps before reproduction, and several generations may take place before an elimination-dispersal event).

4. Finding the direction of movement using probablistic derivative approach

4.1. Computing derivative value for a pixel

The procedure to find the nutrient concentration as well as the suitable direction in which the bacterium should move in order to find the edge pixels is now explained. The common feature between an erroneous pixel and a real edge pixel is that the intensity difference around both the pixels is high. It is very important to differentiate them properly in case the image is noisy; hence a derivative based approach is used. This restrains the bacterium from moving towards the noisy pixels.

Consider the neighborhood of a pixel (x, y) in Fig. 1(a). We consider positive X-axis as vertical downward and positive Y-axis as horizontal right side. A derivative at the central pixel position (x, y) in the direction D (D = {NW, W, SW, S, SE, E, NE, N}) is defined as the difference between the pixel of interest and its neighbor in the corresponding direction.

For example, the derivative at a pixel (x, y) in the north-west direction is given by

$$\partial_{(x,y)}^{NW} = I(x-1, y-1) - I(x, y) \tag{11}$$

where $I(x, y)$ is the intensity at pixel (x, y).

The choice of the derivative is made from the intensities of pixels. Consider an edge passing through the pixel (x, y) in the NE-SW direction as in Fig. 1(b). Since it's an edge it should contain pixels at NE and SW positions as its constituents. Therefore, the derivative

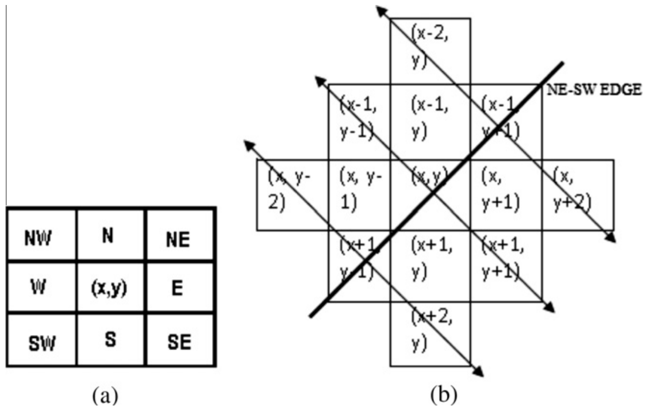


Fig. 1. (a) Pixel (x, y) with its 8-connectivity (b) The pixels to be considered for the edge in NE-SW direction.

values in the direction NW and SE (perpendicular to the edge) for the pixels at positions (x, y), $(x + 1, y - 1)$ and $(x - 1, y + 1)$ should also be high (Verma et al., 2009). This greatly reduces the chances of selecting a noisy pixel. The number of noisy pixels is negligible as compared to the number of edge pixels. Thus, for the pixel (x, y) let ∂_1 be given by

$$\partial_1 = \partial^{NW}_{(x,y)}. \tag{12}$$

Similarly, we have at the neighboring pixels as

$$\partial_2 = \partial^{NW}_{(x+1,y-1)} = I(x+1,y-1) - I(x,y-2) \tag{13}$$

$$\partial_3 = \partial^{NW}_{(x-1,y+1)} = I(x-1,y+1) - I(x-2,y) \tag{14}$$

The average of the above three values can be taken as the net derivative value in NW direction, given by

$$\partial_{avg1} = \frac{(\partial_1 + \partial_2 + \partial_3)}{3} \tag{15}$$

Similarly, we calculate ∂_{avg2} as the average of $\partial^{SE}_{(x,y)}$, $\partial^{SE}_{(x+1,y-1)}$ and $\partial^{SE}_{(x-1,y+1)}$. Having these two net derivatives we are looking for an edge having the intensity difference in one region only. The maximum of these two is the derivative for the edge pixel at (x, y) in the direction NE–SW, denoted by

$$\partial_{NE-SW} = \max(\partial_{avg1}, \partial_{avg2}) \tag{16}$$

Similarly the derivatives along the remaining three directions are: ∂_{NW-SE}, ∂_{N-S} and ∂_{W-E}.

The final derivative at pixel (x, y) which is the maximum of the net derivatives obtained in all the four possible edge directions is:

$$\partial_{x,y} = \max(\partial_{W-E}, \partial_{NW-SE}, \partial_{NE-SW}, \partial_{N-S}) \tag{17}$$

Since our aim is to find the edge pixels, use is made of the observation that larger the value of $\partial_{x,y}$ more is the chance for a bacterium to reach an edge pixel. Therefore, the nutrient concentration at any position, (x, y) should be the function of $\partial_{x,y}$. So we have

$$[\eta_i] = \partial_{(x,y)} \tag{18}$$

Hence, higher the nutrient concentration along an edge more is the movement of bacteria along it.

4.2. Direction probability matrix

The concern here is to locate an edge pixel to which a bacterium should move from a given pixel. To do this we find the values of the derivatives for all the 8 neighborhood pixels around the pixel under consideration in terms of their nutrient concentration. Out of the eight possible directions the next direction is found out using the transition matrix derived from the Ant Colony System (Dorigo and Gambardella, 1997; Verma et al., 2009).

In an artificial Ant Colony System, developed by imitating the real ant's behavior, ant chooses its next step for its movement depending upon the transition probability matrix which is a function of amount of pheromone discharged by ants on the path and the heuristic factor. The transition probability matrix at position i is given by:

$$\rho_{ij} = \left\{ \frac{([\tau_j(t)]^{\alpha})([N_j]^{\beta})}{\sum_{j\in allowed_j}(([\tau_j(t)]^{\alpha})([N_j]^{\beta}))} \right\} \tag{19}$$

where, N_j is the heuristic factor, $\tau_j(t)$ is the pheromone concentration that gives the permissible directions in which an ant movement (in our case ant is replaced by bacterium) has to move; α and β are two parameters that control the relative importance of the two main factors: $\tau_j(t)$ and N_j.

In the proposed approach, pheromone concentration and α and β are taken to be unity. The heuristic factor N_j is taken as the nutrient concentration value η_j. Also, from Eq. (18), nutrient concentration is high along the direction of an edge. Thus the probability of movement along the direction $i \to j$ is calculated from:

$$\rho_{ij} = \left\{ \frac{[\eta_j]}{\sum_{j\in allowed_j}([\eta_j])} \right\} \tag{20}$$

A random direction is selected for the movement, with ρ_{ij} as the probability of selection of direction $i \to j$

$$\Delta(k) = rand(\rho_{ij} \forall j \in allowed_j) \tag{21}$$

where, $\Delta(k)$ is the direction vector for the kth bacterium at position i and $allowed_j$ is the set of possible moves for the bacterium. This random direction gives the direction of movement out of the eight possible directions. Suppose the current location is $[x, y]$, step size is unity and the random direction selected is NW then the next location of bacteria would be $[x - 1, y - 1]$.

5. The algorithm and pseudo code

5.1. Algorithm

[Step 1] Initialize the parameters n, S, N_C, N_{re}, N_{ed}, P_{ed}, $C(i)$ $(i = 1, 2, \ldots, S)$, $F(i)$, $\theta^i(1, 1, 1)$, $V_c(I)$, $J(i, 1, 1, 1)$

where

n: dimension of the search space(2),
S: the number of bacteria in the population,
S_r: bacteria split ratio,
N_C: chemotactic steps,
N_{re}: the number of reproduction steps,
N_{ed}: the number of elimination-dispersal events,
N_S: swim length,
P_{ed}: elimination-dispersal with probability,
$C(i)$: the size of the step taken in the direction specified by the tumble(unit),
$F(i)$: flag bit for each pixel indicating whether it has already been traversed or not,
$V_c(I)$: is the clique matrix for the image I.
$\theta^i(1, 1, 1)$: initial positions of the bacterium selected randomly.
$J(i, 1, 1, 1)$: Initialized derivative value at the pixel given by $\theta^i(1, 1, 1)$.

[Step 2] Elimination-dispersal loop: $l = l + 1$
[Step 3] Chemotaxis loop: $j = j + 1$

[a] For $i = 1, 2, \ldots, S$, take a chemotactic step for bacterium i as follows.

[b] Compute the fitness function, $J\ (i, j, k, l)$.

Let, $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), \theta(j, k, l))$ (i.e. add on the cell-to cell attractant–repellant profile to simulate the swarming behavior)

[c] Tumble: Find the directions of possible movement from the derivative value and compute the direction probability matrix using:

$$\rho_{ij} = \left\{ \frac{[\eta_j]}{\sum_{j \in allowed_j}([\eta_j])} \right\}$$

Now select a random direction using Eq. (21) and find the next direction of movement.

[d] Reproduction loop:

For each possible direction, $k = k + 1$.

[e] Move: Let

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

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i. In our case Eq. (1) is used where (m', n') is found out using the step size movement along the direction of tumble.

[f] Compute $J(i, j+1, k, l)$ and let $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), \theta(j, k, l))$

[g] Swim

(i) Let $m = 0$ (counter for swim length).

(ii) While $m < Ns$ (if have not climbed down too long).

- Let $m = m + 1$.
- If $J(i, j+1, k, l) > J_{last}$

then update $\theta^i(j + 1k, l)$ as done in step 3(e). Use this $\theta^i(j + 1k, l)$ to compute the new J$(i, j+1, k, l)$ as in step 3[f]

- Else, let $m = m + 1$

[h] Go to next bacterium $(i + 1)$ if $i \neq$ S (i.e., go to [b] to process the next bacterium).

[Step 4] If $j < N_c$, go to Step 3[e]. In this case, continue chemotaxis, since the life of bacteria is not over.

[Step 5] Reproduction:

[a] For each $i = 1, 2, \ldots, S$, let

$$J^i_{health} = \sum_{j=1}^{N_{c+1}} J(i, j, k, l)$$

be the health of the bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding the noxious substances).

Sort bacteria and chemotactic parameters $C(i)$ in the ascending order of the 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 (this process is performed by the copies that are placed at the same location as that of their parents).

[Step 6] If $k < N_{re}$, go to Step 3[e]. This means that the number of specified reproduction steps is not reached, so the next generation of the chemotactic loop is started.

[Step 7] Elimination-dispersal: For $i = 1, 2, \ldots, S$, eliminate and disperse each bacterium with probability P_{ed}, This results in the number of bacteria a constant. To do this, if a bacterium is eliminated, simply disperse one to a random location in the optimization domain. If $l < N$, then go to Step 2; otherwise end.

5.2. Pseudo code

Bacteria_Edge (Image I)
FOR each pixel in I

$$\begin{aligned} V_c(I_{m,n}) = {} & |I_{m-2,n-1} - I_{m+2,n+1}| + |I_{m-2,n+1} - I_{m+2,n-1}| \\ & + |I_{m-1,n-2} - I_{m+1,n+2}| + |I_{m-1,n-1} - I_{m+1,n+1}| \\ & + |I_{m-1,n} - I_{m+1,n}| + |I_{m-1,n+1} - I_{m-1,n-1}| \\ & + |I_{m-1,n+1} - I_{m-1,n-2}| + |I_{m,n-1} - I_{m,n+1}| \end{aligned}$$

```
FOR (each bacterium $i = 1:S$)
  $\theta^i(1, 1, 1)$ = rand_post( )
  $J(i, 1, 1, 1)$ = derivative_value($\theta^i(1, 1, 1)$)
  END FOR
FOR (elimination-dispersal loop $l = 1:N_{ed}$)
    FOR (reproduction-loop $k = 1:N_{re}$)
    FOR (chemotactic-loop $j = 1:N_c$)
    FOR (each bacterium $i = 1:S$)
      Calculate
      $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), \theta(j, k, l))$
      Set $J_{last} = J(i, j, k, l)$
        Tumble:
      Find the direction of possible movement from the
  direction probability matrix.
        Move:
        $\theta^i(j + 1, k, l) = \varphi[m', n', i, j + 1, k, l]$
        Compute $J(i, j + 1, k, l)$
        $m = 0$
        WHILE ($m < N_s$)
          $m = m + 1$
          IF ($J(i, j + 1, k, l) < J_{last}$)
            $J_{last} = J(i, j + 1, k, l)$
            Update $\theta(i, j + 1, k, l)$
            Recalculate $J(i, j + 1, k, l)$
          ELSE
            $m = N_s$
          ENDIF
        END WHILE
      END FOR (Bacterium)
    END FOR (Chemotaxis)
Reproduction:
    For given $k$ and $l$, and each bacterium $i = 1, 2, \ldots, S$
    Sum:
      $J^i_{health} = \sum_{j=1}^{N_{c+1}} J(i, j, k, l)$
    Sort:
      Sort bacteria and chemotactic parameters $C(i)$ in order
  of ascending cost $J_{health}$.
    Split and Eliminate:
      The $S_r$ bacteria with the highest $J_{health}$ values die and the
  remaining $S_r$ bacteria with the best values split.
    END FOR (Reproduction)
    Disperse:
      For $i = 1, 2, \ldots, S$, with probability $P_{ed}$, randomize a
  bacterium's position
    END FOR (Elimination and Dispersal)
END
```

The pixels visited by the bacterium are considered to be the desired edge pixels. Thus, the path traced out by bacteria gives the edges.

6. Results

An edge detector can be evaluated based on two parameters. First, its accuracy in determining the edge pixels, and second, it should provide useful information in the form of meaningful edges.

The accuracy is ascertained using Relative Grading technique (Bryant and Bouldin, 1979). In this, a majority image is found using the results of other five edge detectors: Canny, Edison, Rothwell, Sobel, and SUSAN. Then a pixel-by-pixel comparison of the output of the proposed method is made with the true image.

A pixel in the majority image is an edge pixel, if the majority of the methods claim to have an edge pixel in its neighborhood, with at least one centered on it. For example, Fig. 5(h), shows the majority image obtained from Fig. 5(b)–(f).

A majority image is obtained from methods $1, 2, \ldots, n$ as $M(method1, method2, \ldots, methodn)$.

The kappa (a measure of accuracy) (Cohen, 1960) for the pixel-to-pixel comparison between two images I_1 and I_2 is denoted by $k(I_1, I_2)$.

The information content of the output image is measured by using Shannon's entropy function (Shannon, 1948). It gives the indefiniteness in an image and is calculated from

$$H(I) = \sum_{-i=o}^{L} p_i log p_i \tag{22}$$

where, I stand for the Image. p_i is the frequency of pixels with intensity i. As we have binary levels a window of 3 × 3 centered at the pixel of concentration is considered as the intensity value.

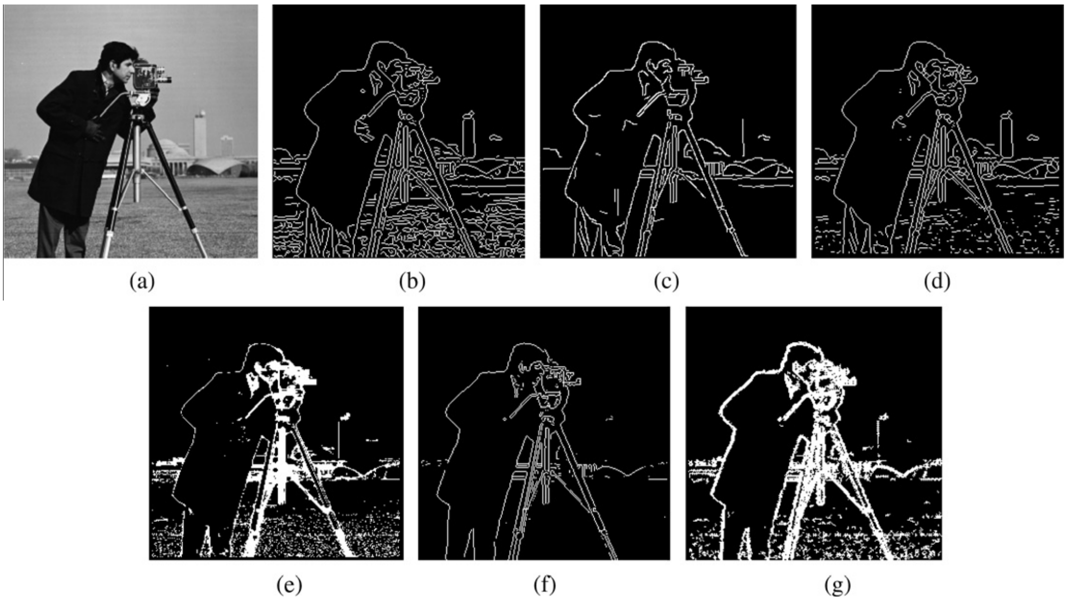


Fig. 2. (a) Original Cameraman image (b) Canny Edge Detector (c) Edison Edge Detector (d) Rothwell Edge Detector (e) SUSAN Edge Detector (f) Sobel Edge Detector and (g) The proposed approach.

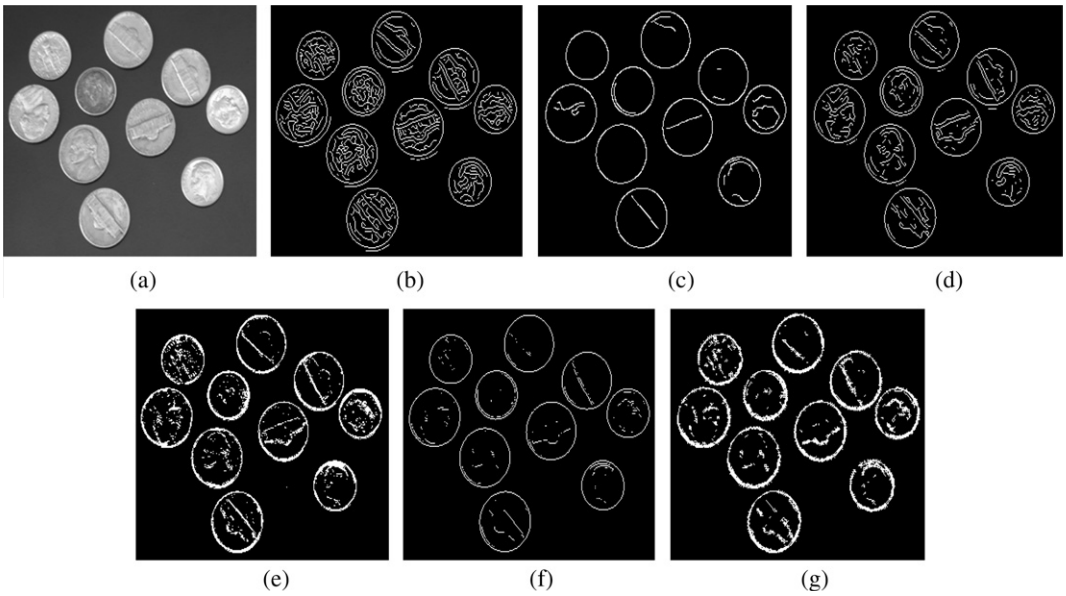


Fig. 3. (a) Original Coins image (b) Canny Edge Detector (c) Edison Edge Detector (d) Rothwell Edge Detector (e) SUSAN Edge Detector (f) Sobel Edge Detector and (g) The proposed approach.

6.1. Comparison with other techniques

The performance of the proposed technique is compared against that of the traditional edge detectors such as Canny, Edison, Rothwell, Sobel and SUSAN. The, traditional edge detectors are implemented using the MATLAB toolbox. The proposed method is also coded in the MATLAB. The parameters are taken as: $S = 100$, $S_r = 0.05S$, $N_s = 10$, $P_{ed} = 0.95$, $N_{ed} = 15$, $N_{re} = 1$, $N_c = 100$. The path traversed by a bacterium represents the edge pixels and is colored white on the black background.

For images in Figs. 2–6, the captions are as follows: (a) the original image, (b) the result of Canny Edge Detector, (c)the result of Edison Edge Detector, (d) the result of Rothwell Edge Detector, (e) the result of SUSAN Edge Detector, (f) the result of Sobel Edge Detector, and (g) the result of the proposed approach.

It is observed that the edges are accurately detected. But our method lacks in presenting the complete edges. This is because of restrictions imposed on the maximum swim length for a bacterium. Also, thick edges can be seen due to bacteria moving parallel to an edge.

Table 1 shows the kappa values in a comparison of several edge detectors. The column 2 of Table 1 shows $k(M(C, E, R, So, Su), P)$, where, C stands for Canny, E for Edison, R for Rothwell, So for Sobel, Su for SUSAN, and P for the proposed method. It may be noted that the values of kappa around 0.5 indicate the poor performance.

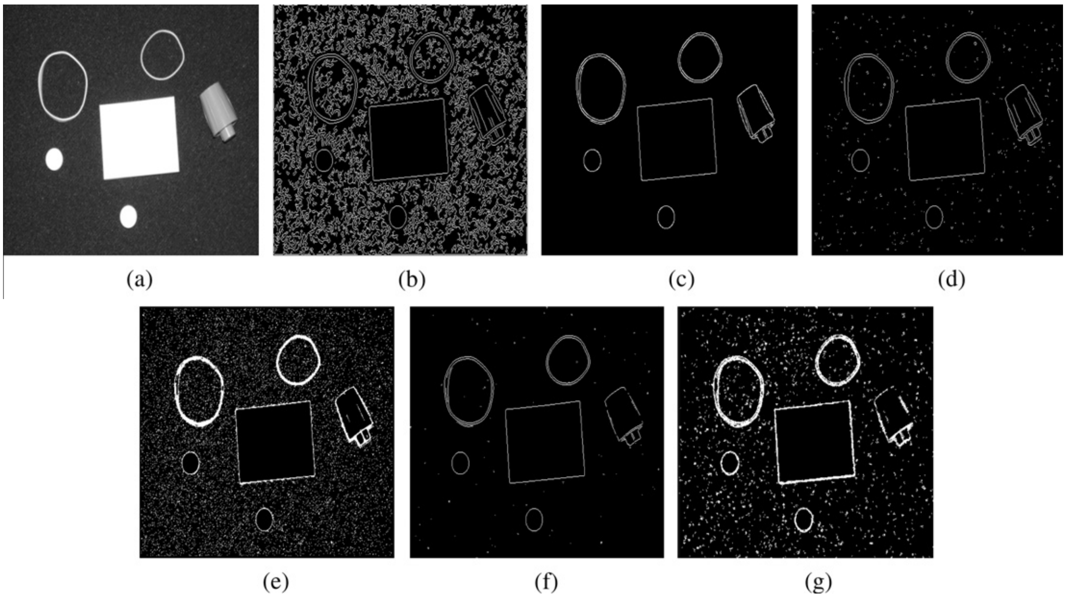


Fig. 4. (a) Original Pillsetc image (b) Canny Edge Detector (c) Edison Edge Detector (d) Rothwell Edge Detector (e) SUSAN Edge Detector (f) Sobel Edge Detector and (g) The proposed approach.

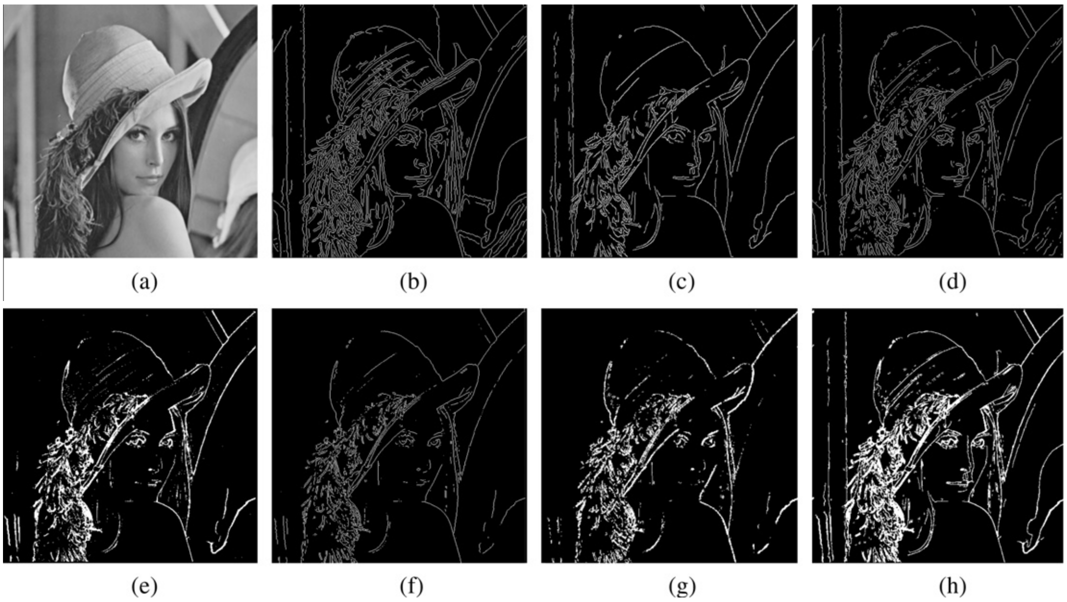


Fig. 5. (a) Original Lena image (b) Canny Edge Detector (c) Edison Edge Detector (d) Rothwell Edge Detector (e) SUSAN Edge Detector (f) Sobel Edge Detector (g) The proposed approach and (h) Majority image obtained using (b) to (f).

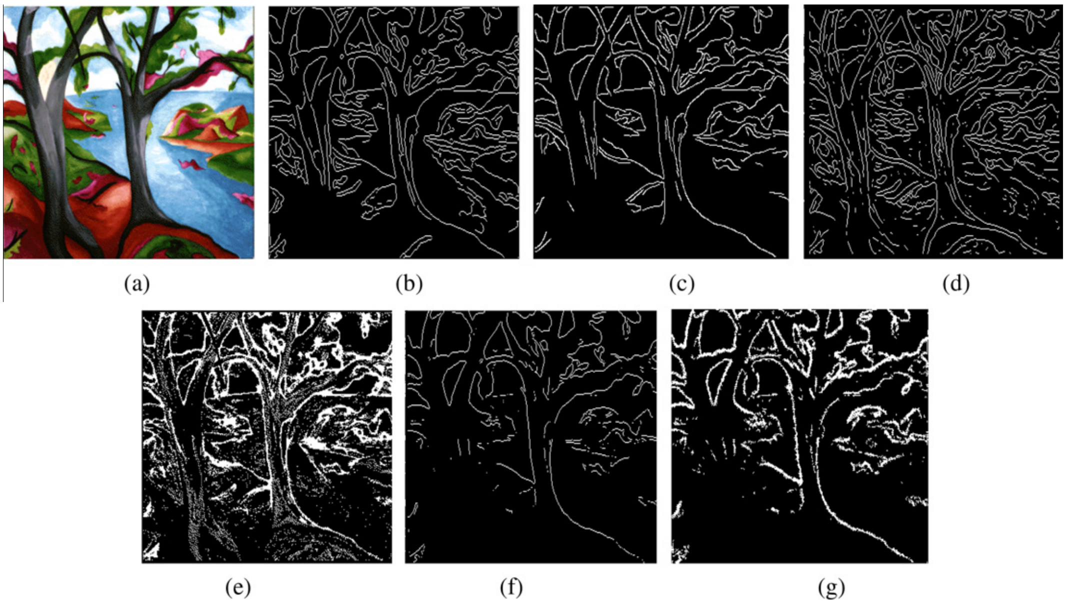


Fig. 6. (a) Original Trees image (b) Canny Edge Detector (c) Edison Edge Detector (d) Rothwell Edge Detector (e) SUSAN Edge Detector (f) Sobel Edge Detector and (g) The proposed approach.

Table 1
Kappa values for comparison with majority image. Column 2 Kappa for proposed edge detector's output comparison with *M*(*C*, *E*, *R*, *So*, *Su*), Columns 3–7: A comparison of Kappa for the ratio, standard edge detector/the proposed edge detector with majority image from other detectors.

Image	Majority	Sobel/Prop.	Canny/Prop.	Edison/Prop.	Rothwell/Prop.	SUSAN/Prop.
Trees	0.4381	0.2802/0.4357	0.3751/0.5103	0.4243/0.4546	0.3349/0.5234	0.3771/0.4915
Lena	0.4594	0.3994/0.4567	0.3831/0.5163	0.4658/0.4996	0.4491/0.5172	0.4622/0.3884
Pillsetc	0.4792	0.3749/0.4744	0.0785/0.4967	0.4133/0.4755	0.4727/0.5016	0.2541/0.3505
Coins	0.5433	0.3878/0.5406	0.3577/0.6098	0.43/0.549	0.4504/0.6108	0.4835/0.4806
Cameraman	0.5953	0.3581/0.5895	0.3474/0.6272	0.3855/0.61	0.433/0.6309	0.4252/0.4198

The column 3 of Table 1 shows the ratio between M of *So* and M of *P*.

$$k(M(C,E,R,So,Su),So)/k(M(C,E,R,So,Su),P)$$

Similarly, the ratios due to other standard edge detectors are given in other columns.

It may be observed from Table 1 that the proposed method outperforms 4 out of 5 methods in all images. The results of the proposed method are poor with respect to Susan edge detector in three images viz. Lena, Coin and Camera man.

Table 2 shows the entropy values for the outputs of different methods on various images. A high entropy value signifies more randomness and less information. Sobel edge detector has the least value of entropy for all images and can be seen to have most appropriate edges. Edison edge detector also performs well. The edge detectors of Rothwell and Canny have a comparable performance over the proposed method but SUSAN edge detector performs poorer than all. It is evident from the results that our method finds meaningful edges in most images but is partially successful in curbing noise as shown in Fig. 4(g).

Table 2
Entropy values of different edge detector for multiple images.

Image	Edison	SUSAN	Rothwell	Canny	Sobel	Proposed
Trees	0.8682	1.7299	1.0936	0.9109	0.5791	0.8682
Lena	0.6777	0.7928	0.7438	0.8848	0.5303	0.7146
Pillsetc	0.2369	1.1692	0.3332	1.421	0.2265	0.7903
Coins	0.4992	0.8759	0.7201	0.9201	0.4821	0.86
Cameraman	0.6852	1.1499	0.8015	0.9931	0.5633	1.2212

6.2. Effects of parameter variation

We now discuss the effects of varying different parameters of the proposed method namely: *S*, *Sr*, N_s, P_{ed}, N_{ed}, N_{re} and N_c on the resultant image of cameraman.

The variation of parameters is judged by two measures: entropy and kappa. To calculate kappa, the output image is compared with the majority image obtained from other 5 methods as explained above. Moreover, it is well known that an optimum value of parameter is the one with less entropy and high kappa.

The results are of edge detection as shown in Fig. 7(a)–(g). We find that N_{re} and N_c have no considerable effect on the results. Both kappa and entropy remain constant. N_s, the swim length causes both entropy and kappa to decrease as it is increased. Decrease in kappa is gradual than the decrease in entropy. Change in bacteria split ratio (S_r) has no effect on kappa but entropy is significantly varied as can be seen in Figs. 7(b) and 8. In Fig. 7(b) we have taken different values of S_r from 0.1*S* to 0.5*S*, where *S* is the number of becteria. Change in N_{ed} causes entropy to drop after $N_{ed} = 10$. Though, kappa also drops but not significantly whereas in Fig. 7(a), we find that there is no change in entropy with change in number of bacteria (*S*) but kappa increases (Fig. 9(a) and (b)). It may also be noted that though increase in *S* is favorable here but it adds more burden on computing resources. Thus, an optimal

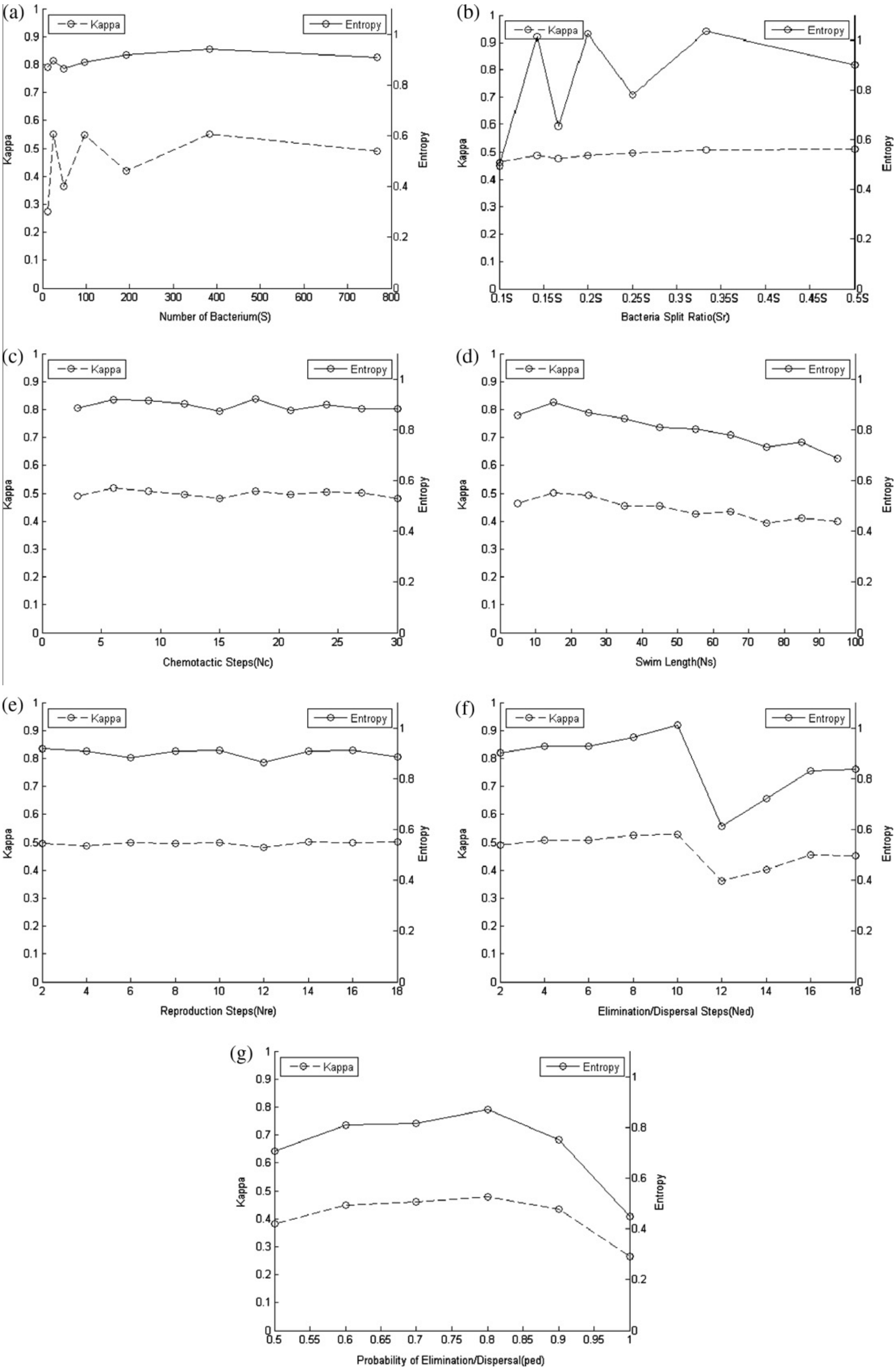
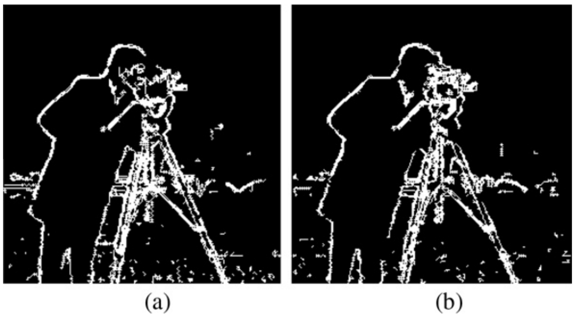


Fig. 7. Plot between Kappa and Entropy v/s initial values parameters (a) S (b) S_r (c) N_c (d) N_s (e) N_{re} (f) N_{ed} (g) P_{ed}.

trade-off has to be found between performance and resources. In Fig. 7(g), we observe that the shape of plots of both kappa and entropy is a parabola facing downwards and centered on 0.8. It also validates that a value of P_{ed} around 0.9 would be optimum.

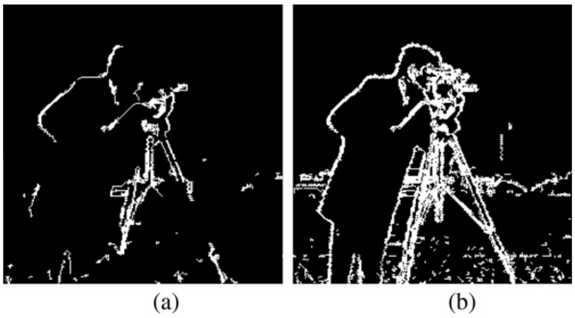
7. Conclusion and future work

Edge detection is essential in many tasks of image processing. This study proposes a novel BF based approach for edge detection.



(a) (b)

Fig. 8. Output images for the split ratio (a) $S_r = 0.25S$ (b) $S_r = 0.33S$.



(a) (b)

Fig. 9. Output images for the split ratio (a) $S = 12$ and (b) $S = 768$.

The proposed method finds robust edges even in the complex and noisy images. This work opens a new domain of research in the field of edge detection using bio-inspired algorithms.

The results of the proposed method are compared with those of other standard methods using kappa and entropy. Our method performs better than many other standard methods. The variation of several initial parameters on the output of the proposed edge detector is discussed. Their values are derived empirically. These values may also be found specifically for each image to gain maximum benefit. So a way to calculate the optimum values of all the parameters in less time needs to be investigated.

It is noted that our results show some disconnected edges. Since BFOA has been devised with the aim of finding global extremes, this error is expected. If a form of preferential treatment such that pixels connected to edge pixels get an advantage is introduced then this problem can be mitigated. Also, some form of repellant need to be added to the path already traced by bacteria so that parallel/double edges are not formed.

References

Abdallah, A.A., Ayman, A.A., 2009. Edge detection in digital images using fuuzy logic techniques. World Academy of Sci. Eng. Technol. 51, 178–186.

Bryant, D.J., Bouldin, D.W., 1979. Evaluation of edge operators using Relative and absolute grading. In: Proc. IEEE Comput. Soc. Conf. Pattern Recognition and Image Processing, pp. 138–145.

Canny, J.F., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Machine Intell. 8 (6), 679–698.

Cheung, K., Chan, W., 1995. Fuzzy one –Mean algorithm for edge detection. IEEE Internat. Conf. Fuzzy Systems, 2039–2044.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Measure. 20 (1), 37–46.

Dorigo, M., Gambardella, L.M., 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput., 73–81.

El-Khamy, S., El-Yamany, N., Lotfy, M., 2000. A modified fuzzy sobel edge detector. In: Seventeenth National Radio Science Conf. (NRSC'2000), February 22–24, Minufia, Egypt, 2000.

Hanmandlu, M., Verma, O.P., Kumar, N.K., Kulkarni, M., 2009. A novel optimal fuzzy system for color image enhancement using bacterial foraging. IEEE Trans. Instrum. Measure. 58 (8), 2867–2879.

Kim, D.H., Cho, C.H., 2005. Bacterial foraging based neural network fuzzy learning. IICAI, 2030–2036.

Kuo, Y., Lee, C., Liu, C., 1997. A new fuzzy edge detection method for image enhancement. IEEE Internat. Conf. Fuzzy Systems, 1069–1074.

Liu, Y., Passino, K.M., 2002. Biomimicry of social foraging bacteria for distributed optimization models, principles and emergent behaviors. J. Optim. Theory Appl. 115 (3), 603–628.

Marr, D., Hildreth, E.C., 1980. Theory of edge detection. Proc. Roy. Soc. London b207, 187–217.

Mishra, S., 2005. A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. IEEE Trans. Evol. Comput. 9 (1), 61–73.

Mishra, S., Bhende, C.N., 2007. Bacterial foraging technique-based optimized active power filter for load compensation. IEEE Trans. Power Delivery 22 (1), 457–465.

Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. Control Systems Magazine, IEEE 22 (3), 52–67.

Raman, Maini, Sobel, J.S., 2006. Performance evaluation of prewitt edge detector for noisy images. GVIP J. 6 (3).

Russo, F., 1998. Edge detection in noisy images using fuzzy reasoning. IEEE Trans. Instrum. Measure. 47 (5), 1102–1105.

Shannon, C.E., 1948. A mathematical theory of communication. Bell Syst. Tech. J. 27, 379–423. 623-656.

Smith, S.M., Brady, J.M., 1997. SUSAN – A new approach to low level image processing. Internat. J. Comput. Vision 23 (1), 45–78.

Tripathy, M., Mishra, S., Lai, L.L., Zhang, Q.P., 2006. Transmission loss reduction based on FACTS and bacteria foraging algorithm. PPSN, 222–231.

Verma, O.P., Hanmandlu, M., Kumar, P., Srivastava, S., 2009. A novel approach for edge detection using ant colony optimization and fuzzy derivative technique. Proc. IEEE, IACC, 1206–1212.