##### **Chapter 1**

##### **Introduction**

Edge Detection in image is an important step for a complete image understanding system. Its importance arises from the fact that edges carry most important information in the image. Accuracy of many high-level image processing tasks such as image segmentation and object recognition directly depend on the quality of the edge detection procedure.

**1.1 Brief review of Edge Detection**

In the images, edges are marked with discontinuity or significant variation in intensity or gray levels. An edge is not a physical entity, just like a shadow. It is where the picture ends and the wall starts, where the vertical and the horizontal surfaces of an object meet. If there were sensor with infinitely small footprints and zero-width point spread functions, an edge would be recorded between pixels within in an image. In reality, what appears to be an edge from the distance may even contain other edges when looked close-up. Edges are scale-dependent and an edge may contain other edges, but at a certain scale, an edge still has no width. If the edges in an image are identified accurately, all the objects are located and their basic properties such as area, perimeter and shape can be measured. Therefore edges are used for boundary estimation and segmentation in the scene. Since computer vision involves the identification and classification of objects in an image, edge detection is an essential tool.

**1.1.1 Types of Edges**

All edges are locally directional. Therefore, the goal in edge detection is to find out what occurred perpendicular to an edge. The following is a list of commonly found edges.



Figure 1. Types of Edges (a) Sharp step (b) Gradual step (c) Roof (d) Trough

A Sharp Step, as shown in Figure 1, is an idealization of an edge. Since an image is always band limited, this type of graph cannot ever occur. A Gradual Step, as shown in Figure 1 is very similar to a Sharp Step, but it has been smoothed out. The change in intensity is not as quick or sharp. A Roof, as show in Figure 1, is different than the first two edges. The derivative of this edge is discontinuous. A Roof can have a variety of sharpness, widths, and spatial extents. The Trough, also shown in Figure 1, is the inverse of a Roof.

Edge detection is very useful in a number of contexts. Edges characterize object boundaries and are, therefore, useful for segmentation, registration, and identification of objects in scenes [1].

The goal of the edge detection process in a digital image is to determine the frontiers of all represented objects, based on automatic processing of the color or gray level information in each present pixel. Edge detection has many applications in image processing and computer vision, and is an indispensable technique in both biological and robot vision [3]. The main objective of edge detection in image processing is to reduce data storage while at same time retaining its topological properties, to reduce transmission time and to facilitate the extraction of morphological outlines from the digitized image.

**1.1.2 Criteria for Edge Detection**

There are large numbers of edge detection operators available, each designed to be sensitive to certain types of edges. The Quality of edge detection can be measured from several criteria objectively. Some criteria are proposed in terms of mathematical measurement, some of them are based on application and implementation requirements. In all five cases a quantitative evaluation of performance requires use of images where the true edges are known.

a) *Good detection*: There should be a minimum number of false edges. Usually, edges are detected after a threshold operation. The high threshold will lead to less false edges, but it also reduces the number of true edges detected.

b) *Noise sensitivity***:** The robust algorithm can detect edges in certain acceptable noise (Gaussian, Uniform and impulsive noise) environments. Actually, an edge detector detects and also amplifies the noise simultaneously. Strategic filtering, consistency checking and post processing (such as non-maximum suppression) can be used to reduce noise sensitivity.

c) *Good localization***:** The edge location must be reported as close as possible to the correct position, i.e. edge localization accuracy (ELA).

d) *Orientation sensitivity***:** The operator not only detects edge magnitude, but it also detects edge orientation correctly. Orientation can be used in post processing to connect edge segments, reject noise and suppress non-maximum edge magnitude.

e) *Speed and efficiency***:** The algorithm should be fast enough to be usable in an image processing system. An algorithm that allows recursive implementation or separately processing can greatly improve efficiency.

Criteria of edge detection will help to evaluate the performance of edge detectors. Correspondingly, different techniques have been developed to find edges based upon the above criteria, which can be classified into linear and non-linear techniques.

**1.1.3 Motivation behind Edge Detection**

The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. For an image formation model, discontinuities in image brightness are likely to correspond to:-

a) Discontinuities in depth

b) Discontinuities in surface orientation

c) Changes in material properties

d) Variations in scene illumination

In the ideal case, the result of applying an edge detector to an image may lead to a set of connected curves that indicates the boundaries of objects, the boundaries of surface marking as well curves that correspond to discontinuities in surface orientation. If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified. Unfortunately, however, it is not always possible to obtain such ideal edges from real life images of moderate complexity. Edges extracted from non-trivial images are often hampered by fragmentation i.e. the edge curves are not connected, missing edge segments, false edges etc., which complicate the subsequent task of interpreting the image data.

**1.2 Literature surveys for Edge Detection Techniques**

Many edge detection methods are proposed in history which is basically categorized as classical edge detectors and Laplacian edge detector. One of them is Prewitt Edge Detector [1, 2] which was proposed to detect the edges by computing first order derivative. Here, Prewitt calculate two derivatives one for horizontal changes and other for vertical. These are found by using two 3×3 kernels which are convolved with the original image. Similarly Sobel Edge Detector[3] calculates the gradientof the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. Canny edge detector [4] is also one of the edge detectors with four filters to detect horizontal, vertical and diagonal edges of the image. Edge detection in presence of noise is a very difficult task. All the classical edge detectors fail to detect edges in the presence of noise. The nature of the image data is indeterminate and the edges of an object in an image are not very clear and occasionally scene pixel to object ones occurs moderately, so fuzzy reasoning is able to extract useful attributes from the approximate and incomplete data and improve the task of edge detection.

As stated, usually edge detection is performed by smoothing, differentiating and thresholding. Although, the gradient-based edge detection method has been widely applied in practice and a reasonable edge map has been obtained for most images, they suffer from some practical limitations. Firstly, they need a smoothing operation to alleviate the effect of high spatial frequency in estimating the gradient. Usually, this smoothing is applied to all pixels in the image including the edge regions, and so the edge is distorted and missed in some cases in particular at junctions or corners. Secondly, the gradient magnitude alone is insufficient to determine meaningful edges because of the ambiguity caused by underlying pixel pattern, especially in complex natural scenes. Thirdly, the gradient-based edge detection method increases the computational complexity because calculations, such as square root and arctangent, to produce the gradient vector are required. The detailed comparison and evaluation of edge detectors has been performed by Heath et al. [5]. They employed people to evaluate performance of several edge detectors with a number of images and looked for correlations in judgments of participants.

The nature of the image data is indeterminate and the edges of an object in an image are not very clear and occasionally scene pixel to object ones occur moderately, so fuzzy reasoning is able to extract useful attributes from the approximate and incomplete data and improve the task of edge detection. Different algorithms for fuzzy based edge detection have been proposed [6-8]. In most of these methods, adjacent points of pixels are assumed in some classes and then fuzzy system inference are implemented using appropriate membership function, defined for each class. Fuzzy logic by the local approach has been used in Bloch *et al* [9] for morphological edge extraction method. Ho *et al.* [10] used both global and local image information for fuzzy categorization and classification based on edges. Abdallah *et al* [11] propose a fuzzy logic reasoning strategy for edge detection in digital images without determining the threshold value.

The pixels in the image can be categorized into two clusters: Edge and Non-edge. These clusters follows the similarity property i.e. pixels on the edges are similar and pixels on the edge are dissimilar from the pixels on Non-edge cluster. The major alternative to the similarity based categorization of natural concepts is the rule-based categorization where it is argued that the membership in natural categories is not primarily dependent on similarity, and the way in which category membership is determined is different from the way in which typicality is derived. One example of where typicality and category membership apparently have very different determinations is the case of concepts of kinship. Whether someone is a grandmother depends only on whether or not she is female and is the mother of a parent (or some logical equivalent definition rule). Whether someone is typical grandmother however depends on whether the stereotypical grandmother characteristics-white hair, rocking chair, bakes cookies- apply. In this case, similarity, without rules, to the prototype (or more properly stereotype) does not provide any more than probabilistic information about true membership of the category [12-15].

Recently edge detection method has been proposed by [16], in which k-mean Soble algorithm primarily combined with clustering to detect edges. An Improved Canny Edge Detection Algorithm Based on Predisposal Method for Image Corrupted by Gaussian Noise is given by [17] but this method work well for the images that were corrupted by Gaussian noise. Recent approaches include border detection using threshold fusion [18]; Binary Partition Tree [19]; multi-structuring elements edge detection based on mathematical morphology [20], unsupervised edge detection using wavelet is based on managing the multi-scale data in wavelet domain [21], a scheme, which allows the threshold to be adapted in according to the correlation between neighboring regions [22], edge detection and image restoration with anisotropic topological gradient [23]. Yu Jing et. al. [24], proposed an edge detection approach of oil slick IR aerial images by defining an energy function model combining a region-scalable-fitting concept and a global minimization active contour (GMAC) model. Pablo Arbela´ ezet. al.[25], presents an effective approach for contour detection by combining the multiple local cues into a globalization framework based on spectral clustering. Fuzzy systems and the optimization processes like particle swarm, ant colony and bacteria foraging are also gaining the popularity

In the recent years, fuzzy techniques have found favor for the edge detection [26], for instance Adaptive Fuzzy Classifier Approach (AFCA) [27] for local edge detection in severely degraded images and edge detection using Fuzzy de-blocking algorithm based on ICM filter [28]. Mehul et al. [29] present Fuzzy logic based automatic edge thresholding technique foe edge detection that overcome the drawback of Dong Liu, algorithm [30]. Another technique for edge detection proposed in [31] it uses the fuzzy heuristic edge detection which incorporates particle swarm optimization. Ant Colony Optimization (ACO) is another swarm intelligence technique given by Dorigo *et al.* [32]. For instance Verma *et al.* [33] detect edges in digital images by placing artificial ants on the image and considering intensity differences between image pixels as heuristic information for the ant colony system. Many ACO-based edge detection algorithms have been proposed. In most papers, the ants’ movement is decided by the values of pixels’ gray gradient which is sensitive to noise [34-37]. A recent approach using the ant colony optimization is proposed by Jian Zhang et al. [38] in combination with statistical estimation. In this study ants’ movement is decided by the relative difference of means of pixel circle neighborhood. In another interesting work, Wafa *et al.* [39] compute the gradient and standard deviation for each pixel to obtain two edge sets that serve as inputs to the fuzzy system to decide whether a pixel belongs to the edge pixel or not. Yishu *et al.* [40] uses both multi-scale wavelet transform and fuzzy c-means clustering algorithm to obtain the edge map of the image adaptively. Fujian wang et al. [41] proposes another approach based on non-linear interpolation. Nonlinear interpolation algorithms can reduce the artifacts of linear methods. Edge orientations histograms always represented a middle term between reliability and description dimension. They are often used in practical applications because they provide very efficient and simple solutions. Ant´onio M. G. Pinheiro proposed a new technique the Angular Orientation Partition Edge Descriptor (AOP) suitable for image semantic annotation, and resilient to image rotation and translation, is described. It is based on the Angular Radial Partition Descriptor (ARP) [42].

##### **Chapter 2**

##### **Bacteria Foraging Optimization**

A new evolutionary technique, called Bacterial Foraging scheme, was introduced by K.M.Passino[43].The foraging can be modeled as an optimization process where bacteria seek to maximize the energy obtained per unit time spent during foraging. In this process, the nutrient function is defined and is being maximized by each bacterium in search of food. Each bacterium tries to maximize the amount of nutrient while minimizing time and energy cost by following four stages: 1) Chemo taxis, 2) Swarming, 3) Reproduction, and 4) Elimination & Dispersal. In the beginning, there will be as many solutions as the number of bacteria. So, each bacterium produces a solution for set of optimal values of parameters iteratively, and gradually all the bacteria converge on the global optimum.

During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. Flagella help an *E.coli* bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging. When they rotate the flagella in the clockwise direction, each flagellum pulls on the cell. That results in the moving of flagella independently and finally the bacterium tumbles with lesser number of tumbling whereas in a harmful place it tumbles frequently to find a nutrient gradient. Moving the flagella in the counterclockwise direction helps the bacterium to swim at a very fast rate. In the above-mentioned algorithm the bacteria undergoes chemotaxis, where they like to move towards a nutrient gradient and avoid noxious environment. Generally the bacteria move for a longer distance in a friendly environment. Figure 2 depicts how clockwise and counter clockwise movement of a bacterium take place in a nutrient solution.



Fig.2. Swim and tumble of a bacterium

When they get food in sufficient, they are increased in length and in presence of suitable temperature they break in the middle to from an exact replica of itself. This phenomenon inspired Passino to introduce an event of reproduction in BFOA. Due to the occurrence of sudden environmental changes or attack, the chemotactic progress may be destroyed and a group of bacteria may move to some other places or some other may be introduced in the swarm of concern. This constitutes the event of elimination-dispersal in the real bacterial population, where all the bacteria in a region are killed or a group is dispersed into a new part of the environment.

Now suppose that we want to find the minimum of *J(**)*where  *p*(i.e. is a *p*-dimensional vector of real numbers)*,* and we do not have measurements or an analytical description of the gradient *J(* *)* . BFOA mimics the four principal mechanisms observed in a real bacterial system: chemotaxis, swarming, reproduction, and elimination-dispersal to solve this non-gradient optimization problem.

Let us define a chemotactic step to be a tumble followed by a tumble or a tumble followed by a run. Let *j* be the index for the chemotactic step. Let *k* be the index for the reproduction step. Let *l* be the index of the elimination-dispersal event. Also let

*p*: Dimension of the search space,

*S*: Total number of bacteria in the population,

*Nc*: The number of chemotactic steps,

*Ns*: The swimming length.

*Nre*: The number of reproduction steps,

*Ned*: The number of elimination-dispersal events,

*Ped*: Elimination-dispersal probability,

*C (i)*: The size of the step taken in the random direction specified by the tumble.

Let *P*(*j*, *k*, *l*) { ( *j*, *k*, *l*) | *i* 1,2,..., *S*} *i* represent the position of each member in the population of the *S* bacteria at the *j-*th chemotactic step, *k-*th reproduction step, and *l-*th elimination-dispersal event. Here, let *J* (*i*, *j*, *k*, *l*) denote the cost at the location of the *i-*th bacterium ( *j*, *k*, *l*)(sometimes we drop the indices and refer to the *i-*th bacterium position as ). Note that we will interchangeably refer to *J* as being a “cost” (using terminology from optimization theory) and as being a nutrient surface (in reference to the biological connections). For actual bacterial populations, *S* can be very large (e.g., *S* =109), but *p* = 3. In our computer simulations, we will use much smaller population sizes and will keep the population size fixed. BFOA, however, allows *p* > 3 so that we can apply the method to higher dimensional optimization problems. Below we briefly describe the four prime steps in BFOA.

i) **Chemotaxis**: This process simulates the movement of an *E.coli* cell through swimming and tumbling via flagella. Biologically an *E.coli* bacterium can move in two different ways. It can swim for a period of time in the same direction or it may tumble, and alternate between these two modes of operation for the entire lifetime. Suppose (*i,j*, *k*, *l*) represents *i*-th bacterium at *j*th chemotactic, *k*-th reproductive and *l*-th elimination-dispersal step. *C*(*i*) is the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemotaxis the movement of the bacterium may be represented by

…(1)

Where indicates a vector in the random direction whose elements lie in [-1, 1].

ii) **Swarming**: An interesting group behavior has been observed for several motile species of bacteria including *E.coli* and *S. typhimurium*, where intricate and stable spatio-temporal patterns (swarms) are formed in semisolid nutrient medium. A group of *E.coli* cells arrange themselves in a traveling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo-effecter. The cells when stimulated by a high level of *succinate*, release an attractant *aspertate*, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density. The cell-to-cell signaling in *E. coli* swarm may be represented by the following function.

…(2)

where *J cc* (, *P*(*j*, *k*, *l*)) is the objective function value to be added to the actual objective function (to be minimized) to present a time varying objective function, *S* is the total number of bacteria, *p* is the number of variables to be optimized, which are present in each bacterium andis a point in the *p*-dimensional search domain. are different coefficients that should be chosen properly.

iii) **Reproduction:** The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.

iv) **Elimination and Dispersal**: Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons e.g. a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. Events can take place in such a fashion that all the bacteria in a region are killed or a group is dispersed into a new location. To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space.

The pseudo-code of the complete algorithm is presented below:

**The BFOA Algorithm**

**Parameters**:

**[Step 1]** Initialize parameters *p, S, Nc, Ns, Nre, Ned, Ped, C(i)(i=1,2…S)*, .

**Algorithm**:

**[Step 2]** Elimination-dispersal loop: *l*=*l*+1

**[Step 3]** Reproduction loop: *k*=*k*+1

**[Step 4]** 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, (i.e. add on the cell-to cell attractant–repellant profile to simulate the swarming behavior)

where, *Jcc*is defined in (2).

[c] Let *Jlast=J (i, j, k, l)* to save this value since we may find a better cost via a run.

[d] Tumble: generate a random vector (*i*)*p* with each element (*i*),*m* 1,2,..., *p*, *m* a random number on [-1, 1].

[e] Move: Let

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

[f] Compute *J* (*i*, *j* 1, *k*, *l*) and let

[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*) Jlast( if doing better), let Jlast= J (*i*, *j* 1, *k*, *l*) and let

And use this to compute the new J (*i*, *j* 1, *k*, *l*) as we did in [f]

• Else, let *m*= *Ns*. This is the end of the while statement.

[h] Go to next bacterium (*i*+1) if *i* *S* (i.e., go to [b] to process the next bacterium).

**[Step 5]** If  *j**Nc*, go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.

[**Step 6**] Reproduction:

[a] For the given *k* and *l*, and for each *i* 1,2,..., *S* , let

…(3)

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 noxious substances). Sort bacteria and chemotactic parameters *C*(*i*) in order of ascending cost *Jhealth* (higher cost means lower health).

[b] The *Sr* bacteria with the highest *Jhealth* values die and the remaining *Sr* bacteria with the best values split (this process is performed by the copies that are made are placed at the same location as their parent).

[**Step 7**] If *k* *Nre*, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start the next generation of the chemotactic loop.

[**Step 8**] Elimination-dispersal: For *i* 1,2..., *S* with probability *Ped*, eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If *l* *Ned*, then go to step 2; otherwise end.

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 through swimming and tumbling. On the other hand, in swarming, each *E. coli* bacterium signals another via attractants to swarm together. This is basically the cell to cell signaling stage. Furthermore, in reproduction bacterium with the least energy dies and the other bacteria with high energy survive. 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 process. This stage helps the bacteria avoid the local optimum.

##### **Chapter 3**

##### **The Proposed Approach**

##### **3.1** **Similarity Concept**

The detection of edges in an image could be basically defined as clustering of the pixels into two categories: edge and non-edges. Thus, the pixels in the same category must have similar features. Similarity, a relationship between two perceptual or conceptual objects is one of the central problems of psychology. The concept of similarity is very important as it provides the stepping stones for organizing the world into categories. In daily life, we often come across situations where we have to distinguish similar groups or we have to classify some similar objects. Similarity measure thus becomes an important tool to decide the similarity degree between two groups or between two objects.

Psychologists have developed two main categorization models: Similarity-based categorization and Rule-based categorization. The concept behind these categorizations is discussed by Demirci e*t al* [44]. Prototype Concept by Rocsh *et al* [45], Exemplar model (Generalized Context Model) by Nosofsky *et al* [46] and Feature Contrast Model by Tversky *et al* [47] are the most popular similarity models for classification.

According to Hampton, classifying on the basis of similarity must involve rules-there must be a rule for determining a similarity value for any pair of concepts, and there must be a rule for deriving degree of category membership (either as a binary outcome via a threshold criterion, or as a fuzzy judgment on a response scale) on basis of this similarity [12]. Consequently, the notion of similarity involves an elaborate cognitive process rather than simply a mathematical model. Whenever the assessment of similarity should reproduce the judgment of a human observer based on qualitative features, it is appropriate to model it as a cognitive process that simulates human similarity perception. Fuzzy set theory has been very attractive tool for modeling and mimicking cognitive process, especially those concerning recognition aspects. Also fuzzy set theory is able to handle qualitative non-numerical descriptions, class memberships and human reasoning. The first assessment of similarity and relations in terms of fuzzy logic was studied by Zadeh [48]. Following them, variations of fuzzy similarity measures such as Fuzzy Feature Contrast Model and Generalized Fuzzy Indices have been proposed by researchers [49-51].

# 3.2 Fuzzy Similarity Measure in Color Image [44]

An image is collection pixels which have feature vectors. The features of a pixel could be gray level for gray scale images, or red, green, blue levels for color images. The artificial features: texture, noise etc., could also be added into feature vector. In image processing field, the similarity measure of two pixels has been generally assessed so far by means of Euclidian distance in color space. On the other hand, Wuerger et al [53] showed in their research into proximity judgments in color space that perceptual color proximity is not Euclidean in nature. That means that distance information in Euclidean color space is not adequate for similarity judgment. Recently, for image processing applications, Color Category Map based on Fuzzy Feature Contrast Model was constructed by Seaborn et al [54].

On the other hand, the similarity must involve rules and there must be rules for determining a similarity value for any pair of concepts as Hampton [12] and researchers working on the rule based categorization. Rule-based color similarity was also applied by Demirci *et al* [55] for color image segmentation.

In the proposed approach, the similarity percent of neighboring pixels, including three components i.e. red, green and blue are calculated by means of fuzzy reasoning rules. An image consists of pixels, which are neighbor to each other as shown in Figure 3. Differences of each color component between pixel P1 and P2 could be stated as follows:

***ΔR* =| *L*R,1-*L*R,2|**

***ΔG* =| *L*G,1-*L*G,2|**

***ΔB* =| *L*B,1-*L*B,2| ...** (4)

**P1  P2**

**L R, 1**

**L G, 1**

**L B, 1**

**L R, 2**

**L G, 2**

**L B, 2**

Figure 3 Pixel with color components [55]

In the proposed algorithm, membership functions for the gray level differences of red, green and blue components can be defined. Membership functions being applied can be linear, exponential, gaussian or any other depending upon the ease of usage and the quality of output required. As an example, we have defined the membership functions as the combination of triangular and trapezoidal functions in Fig. 4 to represent the gray level differences.

**ZE MD LR**

**∆R**

**µR**

0 5 128 248 255

Figure 4 Membership function for three linguistic variables for RED component [44]

# 3.2.1 Rule Based Fuzzy Similarity Measure

Gray level differences for each color component have been partitioned into linguistic variables in their respective membership functions. The defined linguistic variables assist in the development of the fuzzy rules for measurement of similarity. The number of fuzzy rules that are required for calculation of similarity would directly depend upon the number of linguistic variables used. Hence, for a set of linguistic variables {Ψ0, Ψ1, Ψ2......ΨN-1}, the number of fuzzy rules would be equal to NN.

Each input linguistic variables were indexed with 0 for Ψ0, 1 for Ψ1, 2 for Ψ2, till (n-1) for ΨN-1. Based on these, we assign (n\*(n-1) +1) linguistic values for color similarity. It is on the basis of these linguistic values that we will infer the extent of presence of an edge. The approach will become clearer from the mentioned example.

In our example, we have considered three linguistic variables, which are *Zero: ZE***,** *Medium:* MD and *Large*: LR as shown in Fig. 4 for red component. A similar membership function is applicable for other two color components. In this case the number of corresponding fuzzy rules would be 27 (ie 33). Seven linguistic values for color similarity have been assigned, which are *Not Similar: NS***,** *Very Little Similar*: *VLS***,** *Little Similar*: *LS***,** *Medium Similar*: *MS, Quite Similar*: *QS*, *Rather Similar*: *RS* and *Exactly Similar*: *ES.*

Generally a fuzzy system is a mapping between its inputs and outputs. For a fuzzy system the mapping of the inputs to the outputs is characterized by a set of *condition-action* rules, or in *modus- pones* form, If *premise* then *consequent.* Generally, the inputs of the fuzzy system are associated with the premise, and the outputs are associated with the consequences. The color edge detection in image processing could be defined as a system in which there are three inputs and single output. Consequently, linguistic rules for color edge detection have been devised as follows: [44]

***Rule1: If Δ****R* is *Zero* ***and Δ****G* is *Zero* **and *Δ****B* is *Zero* ***Then*** P1 and P2 are *Exactly Similar*,

***Rule2: If Δ****R* is *Zero* ***and Δ****G* is *Zero* **and *Δ****B* is *Medium* ***Then*** P1 and P2 are *Rather Similar*,

***Rule3: If Δ****R* is *Zero* ***and Δ****G* is *Large* **and *Δ****B* is *Large* ***Then*** P1 and P2 are *Little Similar*,

***Rule4: If Δ****R* is *Large* ***and Δ****G* is *Large* **and *Δ****B* is *Large* ***Then*** P1 and P2 are *Not Similar*,

***Rule5: If Δ****R* is *Large* ***and Δ****G* is *Zero* **and *Δ****B* is *Medium* ***Then*** P1 and P2 are *Medium Similar*,

***Rule6: If Δ****R* is *Large* ***and Δ****G* is *Zero* **and *Δ****B* is *Zero* ***Then*** P1 and P2 are *Quite Similar,*

***Rule7: If Δ****R* is Medium ***and Δ****G* is *Large* **and *Δ****B* is *Large* ***Then*** P1 and P2 are *Very Little Similar*,

***Rule8: If Δ****R* is Medium ***and Δ****G* is *Zero* **and *Δ****B* is *Medium* ***Then*** P1 and P2 are *Quite Similar,*

***Rule9: If Δ****R* is Medium ***and Δ****G* is *Zero* **and *Δ****B* is *Zero* ***Then*** P1 and P2 are *Rather Similar,*

*,* so on. ...(5)

# 3.2.2 Fuzzy Rules to Determining the Similarity Relation

As mentioned earlier, each input linguistic variables were indexed with 0 for Ψ0, 1 for Ψ1, 2 for Ψ2, till (n-1) for ΨN-1 whereas the output linguistic variables were indexed from 0 to (n\*(n-1) +1). For each rule, index of similarity function was found as (Using Table 1):

...(6)

Where, θ0, θ1, θ2... θN-1 are index number of linguistic variable of each pixel.

In the discussed example, input linguistic variables were indexed with 0 for *ZE*, 1 for *MD* and 2 for *LR* whereas the output linguistic variables were indexed from 0 to 6. For each rule, index of similarity function is given as:

...(7)

Where *k, l* and *m* are index number of linguistic variable of each pixel as shown in Table 2.

Index ‘i’ of the similarity function is used to generate the weighted participation of each rule in the calculation of net similarity of pixels P1 and P2. A variable Sj was defined to represent the weight corresponding to each fuzzy rule depending upon the derived value of index for that particular rule. It is given by:

...(8)

The similarity percent of P1 and P2 could be explicitly represented as [55]:

...(9)

Where, Z is the total number of fuzzy rules (27 in our case). Sj represents the weighted participation of each fuzzy rule as discussed above. µjprem(θ)is the certainty of the premise of the jth rule given by:

µjprem(θ)= µRj(θΔR).µBj(θΔB).µGj(θΔG) ...(10)

µjprem(θ)therefore defines the certainty or level of participation of the Sj values that correspond to every rule with reference to the pixel under consideration.

In the test case we present, the value of µjprem(θ)can be calculated using Table 2 and values µR, µB and µG from fig 4 as follows:

µjprem(1)=ZEb\*ZEg\*ZEr

µjprem(2)=ZEb\*ZEg\*MDr

µjprem(3)=ZEb\*ZEg\*LRr ...(11)

And so on.

Table 1.Fuzzy Rules for similarity

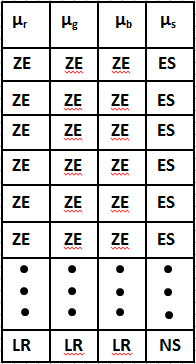
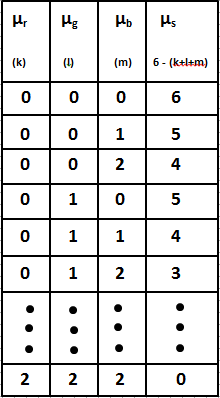


Table 2. Index table for similarity



# 3.2.3 Directional Similarity Image

The similarity of any neighboring two pixels was estimated by means of fuzzy rules as explained previous sections. A pixel in an image has eight neighboring pixels as shown in Figure 5. Therefore the similarity calculations for all the possible combinations are performed as shown in Figure 5.

As we can see in Figure 4, there is no need to consider the similarity of central pixel itself. Similarity in eight proposed directions surrounding the central pixel is calculated. The eight directions proposed are shown in Fig 6.

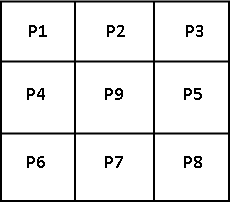


Figure 5 Neighboring pixels in Color Image

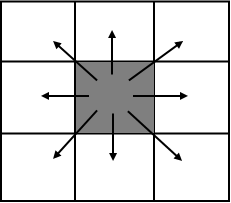


Figure 6 Directions of Similarity Calculation for edge occurrence

The pixels with maximum similarity percent value (=1) show maximum similarity in color image whereas the pixels with zero similarity percent value show dissimilarity in color image.

##### **3.3 The Bacterial Foraging methodolgy For Edge Detection**

The application of Bacterial Foraging methodology has a pre-requisite of defining certain parameters before the algorithm proceeds. We need to define a nutrient function such that the value of this function defines the health and movement of the bacterium. We assume that each individual pixel in the image under consideration is the representational analogue of a bacterium. The proposed algorithm now follows the following set of steps:

**A. Defining the Nutrient Function**

Nutrient Function is defined, based on the similarity measure of bacterium with respect to its neighboring pixels in 8-neighborhood. Each bacterium tries to minimize its similarity value. The similarity percent of every pixel is calculated with its surrounding eight pixels.

**B. Chemotactic Step:**

This is a very important stage of Bacteria Foraging methodology. 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. A new Fuzzy similarity measure proposed in [44] is used for finding the direction of edge pixels. In order to move the current bacterium to its next position, it must satisfy the following two conditions:

* There has to be at least one dissimilar pixel in its 8-neighborhood
* the next position is the most similar pixel in its 8-neighborhood which is also in the neighbor of dissimilar pixel

Suppose P in Fig 7 is the current position of bacterium. Beginning from the most dissimilar pixel P4, the pixel in the neighbor of P4 which is similar to P is searched. If no similar pixel is found, move to the next less dissimilar pixel P3 and search for the similar pixel as described previously. P2 is found to be similar to P and therefore, bacteria will move to P2.

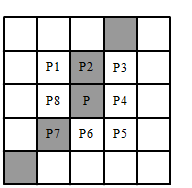


Fig. 7 Chemotaxis step: Darker pixels represent the edge and P1, P2,…,P8 are the 8-neighborhood of P.

**C. Reproduction Step:**

In the Reproduction step, the healthy bacteria are reproduced in a sufficient number such that the generated bacteria are stable. For example, a bacterium is healthy if it is present on the edge and is ready for reproduction if it is present at joint of more than one edge, then it will reproduced equal to the number of edges as shown in the figure 8.

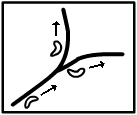


Fig. 8 Reproduction Step

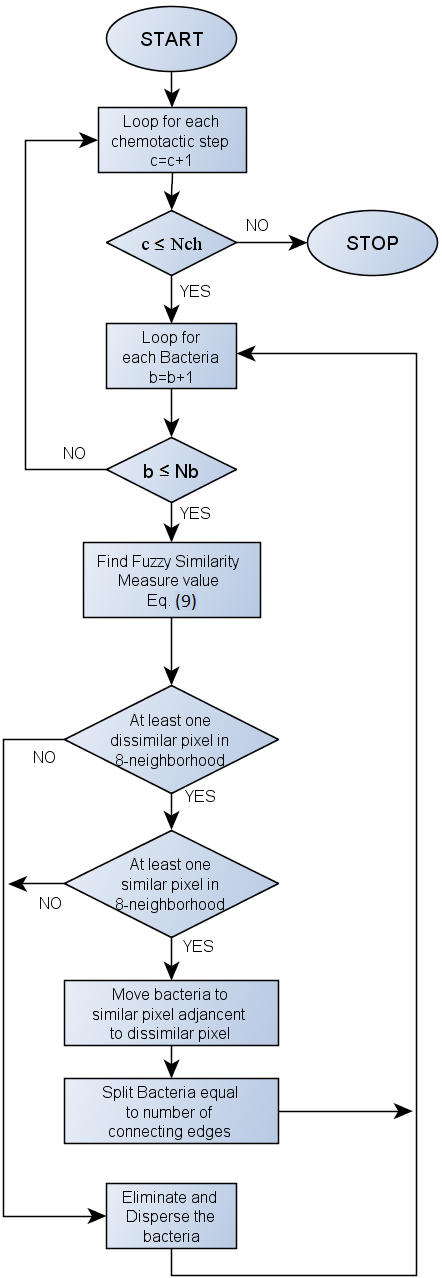


Fig. 9 Flowchart of the proposed approach

##### **Chapter 4**

##### **Experiment, Results and Discussion**

The result of the proposed method is being tested against the majority image formed by the result of five other edge detection methods: Canny, Edison, Prewit, Sobel and Susan. 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, Figure 4h, shows the majority image obtained from Figure 4b-4e. We perform a pixel-by-pixel comparison of the output of the proposed method with the majority image. The methods used for quantitative analysis are:

1. Cohen’s Kappa Measure

The pixel-to-pixel comparison between two images I1 and I2 is done using Cohen’s kappa measure [56] as:

…(12)

Where, Pr(*a*) is the relative observed agreement among images, and Pr(*e*) is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly saying each category. If the images are in complete agreement then κ =1.If there is no agreement among the images (other than what would be expected by chance) then κ ≤ 0.

1. Shannon’s Entropy Function

The information content of the output image is measured by using Shannon’s entropy function [57]. It gives the indefiniteness in an image and is calculated as:

…(13)

Where, stands for Image whose entropy is to be measured. is the frequency of pixels with intensity . Here we have binary levels therefore we consider a window of 3 X 3 centered at the pixel of concentration as the intensity value

# 4.1 Results and Discussion

There are four parameters that require to be initialized to perform the proposed approach efficiently and effectively. These are initial number of bacteria (NB), minimum value of dissimilarity required for the presence of edge (dsm) and number of chemotactic steps (NCh), i.e., lifetime of a bacterium.

To analyze the effect of these parameters, two quantitative analytic methods are used, Shannon’s Entropy and Cohen’s Kappa Measure. Figure 10 shows the variation of entropy with respect to the number of initial bacteria. Entropy comes very low for five number of bacteria, since the resultant image(Fig 12.a) have very less information and as we increase the number of bacteria the amount of information in the image i.e., entropy is also increase but it also introduces the noise in the edge-map in the form of unconnected weak edges. When number of bacteria are less than 45 then entropy curve is showing a steady growth and after that it shows a behavior of saturation i.e., further increase in the initial number of bacteria will not create the much difference in the entropy value. To have enough amount of information while having the noisy pixels as small as possible, we have taken the initial number of bacteria as 10. This estimation is also being justified by the Fig. 11, which shows the effect of increase in initial number of bacteria against the Kappa value. Increasing the initial number of bacteria from 10 to 15 does not give much difference in the value of Kappa measure, i.e., between this interval the resultant images are equally similar to the Majority Image. While further increase in the number of bacteria will shows the increase in the kappa value, because the Majority Image have thick edges and result of the proposed approach also starts giving thick edges as it also introduces some of weak unconnected edges.

Fig. 10 Entropy versus initial number of bacteria

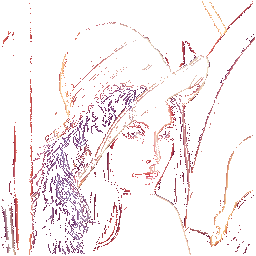
Fig. 11 Kappa versus initial number of bacteria



(a) (b) (c) (d)



(e) (f) (g) (h)



1. (j)

Fig. 12 Result of proposed approach for different number of initial bacteria (NB), (a) NB = 5, (b) NB = 10, (c) NB = 15, (d) NB = 20, (e) NB = 25, (f) NB = 30, (g) NB = 35, (h) NB = 40, (i) NB = 45, (j) NB = 50.

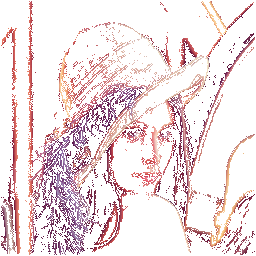
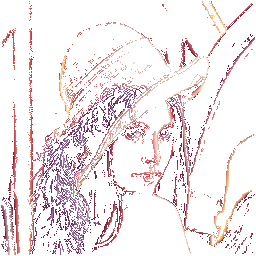
Fig.13 shows the variation of dissimilarity value with respect to Entropy measure. By analyzing this, we have found that with decrease in the dissimilarity value the Entropy is increases steadily. Since the larger dissimilarity results into the less amount of information (i.e., reduced number of edges), and less dissimilarity results into the more amount of information (i.e., weak unconnected edges in the form of noisy pixels are included) as shown in the Fig. 15.a-g. To have enough amount of information while having the noisy pixels as small as possible, we have taken the dissimilarity value as 0.04. We can also verify this by analyzing the Fig. 14, which shows the effect of variation of dissimilarity value with respect to Kappa measure. Kappa value has a steady increase around the dissimilarity value of 0.04, because of thickening of edges and increasing of number of edges. But after dissimilarity value of 0.04, noisy pixel starts appearing more and more as shown in Fig. 15.

Fig. 13 Entropy versus dissimilarity value

Fig. 14 Kappa versus dissimilarity value



(a) (b) (c) (d)



(e) (f) (g)

Fig. 15 Results of proposed approach for different dissimilarity values (DV), (a) DV = 0.045, (b) DV = 0.04, (c) DV = 0.035, (d) DV = 0.03, (e) DV = 0.025, (f) DV = 0.02, (g) DV = 0.015.

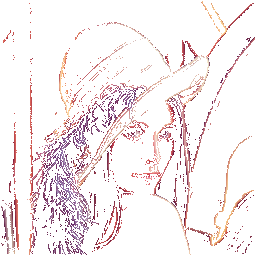
Fig.16 shows the variation of number of chemotactic steps (NCh) with respect to Entropy measure. Increasing the number of chemotactic step means increasing the lifetime of the bacteria. If bacteria are allowed to live longer then it will also try to explore more number of edges and thus requires longer time. To make bacteria to complete its life in a sufficient time with providing good number of edges and less noisy pixels, the number of chemotactic steps taken are 70. Fig. 18 shows result of proposed approach for different number of chemotactic steps.

Fig. 16 Entropy versus number of chemotactic steps (NCh)

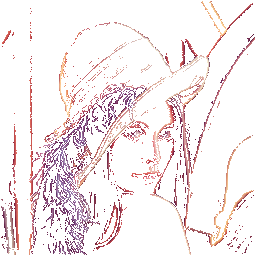
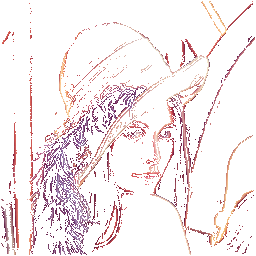
Fig. 17 Kappa versus number of Chemotactic steps (NCh)



(a) (b) (c) (d)



(e) (f) (g) (h)



(i) (j)

Fig. 18 Results of proposed approach for different number of chemotactic steps (NCh), (a) NCh = 50, (b) NCh = 60, (c) NCh = 70, (d) NCh = 80, (e) NCh = 90, (f) NCh = 100, (g) NCh = 110, (h) NCh = 120, (i) NCh = 130, (j) NCh = 140.

# 4.2 Comparison with Other Techniques

The performance of most of the edge detectors proposed in the literature is visually analyzed. Sometimes the visual analysis is insufficient to prove that the proposed method gives more connected edges. To overcome this problem we use the Kappa value and entropy function for quantitative analysis. Entropy represents the amount of information present in the image and Kappa represents the amount of similarity between the two images.

Table 3 represents the values of entropy measure for Majority image and for the results of various approaches: Sobel, Prewitt, Canny, Edison, Susan, and proposed approach. The entropy for results of Sobel and Prewitt is comes out to be smaller than the proposed approach for all the four test images, because they provide less edge information. SUSAN method produces a larger noise content i.e., the results have very thick edges and if there are close edges in image then Susan method connects them in its result and thus makes it difficult to identify the edges differently. Because of thick edges Susan method has the higher entropy value than proposed approach. The canny method gives very thin edges and it does not work on the color images thus there will be information loss in the result, therefore the entropy value obtained using this methods is less than the proposed method. The Edison edge detector produces double edges and thus make it difficult for the image to recognize, therefore the entropy value obtained using this methods is more than the proposed method.

Table 4 represents the values of Kappa measure of Majority image with the results of various approaches: Sobel, Prewitt, Canny, Edison, Susan, and proposed approach. The Majority Images have very thick edges thus representing more influence of Susan method, therefore the value of Kappa measure comes out to be higher for Susan method. Whereas except peppers and house in case of edison, the proposed method have higher value of Kappa measure than other edges detectors. Hence table 2 shows that proposed method work well with acceptable accuracy.

Table 3. Entropy of Results of various Edge detectors

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | edison | susan | prewitt | canny | sobel | proposed | majority\_image |
| Fruit | 1.0613 | 1.1457 | 0.4866 | 0.7991 | 0.4898 | 0.8859 | 1.1061 |
| Lena | 0.9344 | 1.4284 | 0.6147 | 0.9724 | 0.6267 | 1.1018 | 1.3233 |
| Peppers | 0.9306 | 1.345 | 0.587 | 0.9027 | 0.5952 | 0.9656 | 1.291 |
| House | 0.6245 | 1.0981 | 0.4897 | 0.8406 | 0.4919 | 0.7980 | 1.0454 |

Table 4. Kappa Value with respect to Majority Image

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | sobel | prewitt | canny | edison | susan | ours |
| Fruit | 0.2781 | 0.2781 | 0.3267 | 0.3527 | 0.8867 | 0.4323 |
| Lena | 0.2619 | 0.2577 | 0.321 | 0.3708 | 0.8624 | 0.4059 |
| Peppers | 0.259 | 0.2555 | 0.3381 | 0.3842 | 0.8663 | 0.3680 |
| House | 0.3137 | 0.3409 | 0.3409 | 0.4282 | 0.8825 | 0.3763 |

   D:\ajay\New21_6_2011\Results_3_6_2011\Original\house256.tif

(a) (b) (c) (d)

Fig 19. Original images. (a) lena (b) fruit (c) peppers (d) house

(a) (b)

(c) (d)

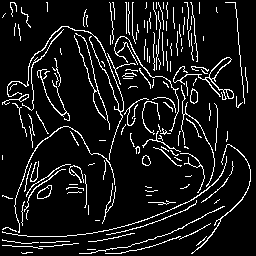
 

(e) (f)

Fig 20. Result of various edge detector for Lena image. (a) Susan method (b) Canny method (c) Edison method (d) Sobel method (e) Prewitt method and (f) proposed method

(a) (b)

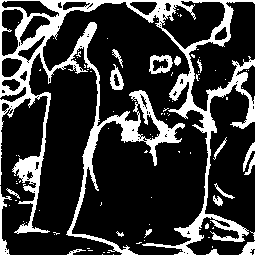
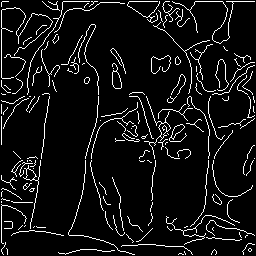
 

(c) (d)

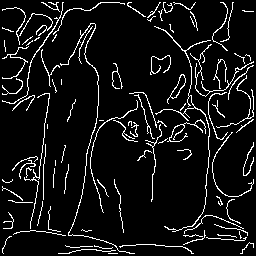
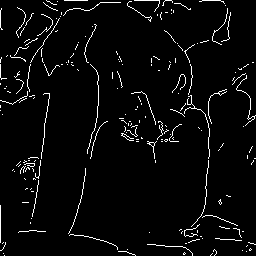
 

(e) (f)

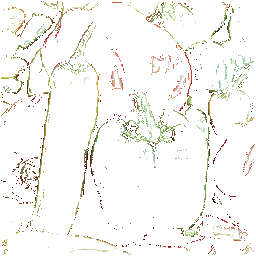
Fig 21. Result of various edge detector for Fruit image. (a) Susan method (b) Canny method (c) Edison method (d) Sobel method (e) Prewitt method and (f) proposed method

(a) (b)

(c) (d)

(e) (f)

Fig 22. Result of various edge detector for Peppers image. (a) Susan method (b) Canny method (c) Edison method (d) Sobel method (e) Prewitt method and (f) proposed method

D:\ajay\New21_6_2011\Results_3_6_2011\susan_results\house256_susan.tif D:\ajay\New21_6_2011\Results_3_6_2011\canny\house256_canny.tif

(a) (b)

D:\ajay\New21_6_2011\Results_3_6_2011\edison\house256_edison.tif D:\ajay\New21_6_2011\Results_3_6_2011\sobel\house256_sobel.tif

(c) (d)D:\ajay\New21_6_2011\Results_3_6_2011\prewitt\house256_prewitt.tif 

(e) (f)

Fig 23. Result of various edge detector for House image. (a) Susan method (b) Canny method (c) Edison method (d) Sobel method (e) Prewitt method and (f) proposed method

##### **Chapter 5**

##### **Conclusion & Future Work**

In the field of edge detection, many optimization techniques are being explored nowadays that makes the process of edge detection more efficient, fast and also robust. In our method, we also used the bacteria foraging with fuzzy similarity measures. In Bacteria Foraging, we used the steps, Defining the Nutrient Function, Chemotexis, Reproduction and Elimination-Dispersion. Fuzzy Similarity Measure used to provide the direction of movement of the bacteria by finding probability of edge occurrence in the neighborhood of the bacteria. Thus the algorithm proceeds by while exploring the benefits of both bacteria and fuzzy similarity measure.

In future, modification of fuzzy rules can produce better result. Further tuning of the weights associated to the fuzzy inference rules is still necessary to reduce even more inclusion in the output image of pixels not belonging to edges.

Our proposed technique is not considering the swim movement and the cell to cell attraction for the bacteria, this can be included to significantly increase the speed and efficiency of the technique.

**References**

[1] Gonzalez, R.C., and Wintz, P.: 'Digital image processing' (Addison-Wesley,Reading, 1992

[2] Gonzalez RC,Woods RE. Digital Image Processing. Reading,MA: Addison-Wesley; 1993

[3] Nalawa,V.S.:'A guided tour of computer vision'(Addison-Wesley,Reading, MA, USA, 1993

[4] Canny JF. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 1986;8(6):679–98.

[5] Heath M, Sarkar S, Sanocki T, Bowyer K.W., “Edge detector comparison: initial study and methodology”, Computer Vision Image Understanding 1998, pp. 38–54.

[6] K. Cheung and W. Chan, "Fuzzy One –Mean Algorithm for Edge Detection, "IEEE International Conference On Fuzzy Systems, pp. 2039- 2044, 1995.

[7] Y. Kuo, C. Lee, and C. Liu, "A New Fuzzy Edge Detection Method for Image Enhancement", IEEE International Conference on Fuzzy Systems, pp. 1069-1074, 1997.

[8] S. El-Khamy, N. El-Yamany, and M. Lotfy, "A Modified Fuzzy Sobel Edge Detector," Seventeenth National Radio Science Conference (NRSC'2000), February 22-24, Minufia, Egypt, 2000.

[9] Bloch I., “Fuzzy sets in image processing”, ACM Symposium on Allied Computing, 1994.

[10] Ho, K.H.L., and Ohnishi, N., “FEDGE – Fuzzy edge detection by fuzzy categorization and classification of edges”, Fuzzy Logic in Artificial Intelligence, Springer (IJCAI’95) Workshop, Montreal, Canada, pp. 182-196, 1995

[11] Abdallah A.A., Ayman A.A, “Edge detection in digital images using fuzzy logic techniques, World Academy of Sc. Engg. Technolgy 51, pp. 178-186, 2009.

[12] J.A. Hampton, “Similarity-based categorization and fuzziness of natural categories” Cognition 65, pp. 137–165, 1998.

[13] E. E. Smith, S.A. Sloman, “Similarity- versus rule-based categorization”, Memory & Cognition 22, pp 377–386, 1994.

[14] F. G. Ashby and R. E. Gott, “Decision Rules in the Perception and Categorization of Multidimensional Stimuli, Journal of Experimental Psychology: Learning”, Memory and Cognition 1, Vol.14, pp 33-53, 1988.

[15] M. A. Erickson and J. K. Kruschke, “Rule-based extrapolation in perceptual categorization”, Psychonomic Bulletin & Review 9, pp. 160-168, 2002.

[16] Miao-le Hou, “K-means Sobel Algorithm in Edge Extracting of Mural Diseases” 2nd International Conference on Digital Object Identifier, pp. 1 – 4, 2010.

[17] Wang Xiao, “An Improved Canny Edge Detection Algorithm Based on Predisposal Method for Image Corrupted by Gaussian Noise” World Automation Congress (WAC), pp. 113 – 116, 2010.

### [18] Celebi, M.E, “Robust Robust border detection in dermoscopy images using threshold fusion border detection in dermoscopy images using threshold fusion”, 17th IEEE International Conference on Image Processing (ICIP), pp. 2541 – 2544, 2010.

[19] Pont-Tuset, J, “CONTOUR DETECTION USING BINARY PARTITION TREES”, 17th IEEE International Conference on Image Processing (ICIP), pp. 1609 – 1612, 2010.

[20] Ge Xing-wei, “Edge Detection and Target Recognition from Complex Background”, 2nd International Conference on Advanced Computer Control (ICACC), Vol. 2, pp. 441 – 444, 2010.

[21] Tello Alonso, M., “Edge Enhancement Algorithm Based on the Wavelet Transform for Automatic Edge Detection in SAR Images”, IEEE Transactions on Geoscience and Remote Sensing, Vol. 49, Issue: 1, Part: 1, pp. 222 – 235, 2011.

[22] I-Hsien Lee, “Information Re-use and Edge Detection In Intra Mode Prediction”, 2nd International Conference on Signal Processing Systems (ICSPS), Vol. 1, pp. V1-591 - V1-594, 2010.

[23] Larnier, S., “EDGE DETECTION AND IMAGE RESTORATION WITH ANISOTROPIC TOPOLOGICAL GRADIENT”, IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1362 – 1365, 2010.

[24] Yu Jing et. al., “A Novel Edge Detection Algorithm Based on Global Minimization Active Contour Model for Oil Slick Infrared Aerial Image”, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, 2011.

[25] Pablo Arbela´ ezet. al., “Contour Detection and Hierarchical Image Segmentation”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 33, 2011.

[26] Chen Xu, Liu Wei, “Study on Shot Boundary Detection Based on Fuzzy Subset-hood Theory”, Intelligent System Design and Engineering Application, pp. 476 – 480, 2011.

[27] Juan F Ramirez Rochac, Lily Liang, Byunggu Yu, Zhao Lu, “An Adaptive Fuzzy Classifier Approach to Edge Detection in Latent Fingerprint Images”, 22nd International Conference on Tools with Artificial Intelligence, pp. 178 – 185, 2010

[28] Xiao Ping, Zhou Zhiheng, “Fuzzy de-blocking algorithm based on ICM filter”, Seventh International Conference on Fuzzy Systems and Knowledge Discovery, pp. 551 – 554, 2010.

[29] Mehul Thakkar, Hitesh Shah, “Automatic Thresholding in Edge Detection Using Fuzzy Approach”, International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1 – 4, 2010.

[30] Dong Liu, Zhaohui Jiang ,Huanqing Feng, “A novel fuzzy classification entropy approach to image thresholding,” Pattern Recognition Letters, Vol. 27, pp. 1968–1975, 2006.

[31] Khalid, N.E.A.  Manaf, M. Aziz, M.E., “Fusion of Fuzzy Heuristic and Particle Swarm Optimization as an edge detector”, International Conference on Information Retrieval & Knowledge Management, (CAMP), pp. 250 – 254, 2010.

[32] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents”, Part B: Cybernetics, IEEE Transactions on Systems, Man, Cybernetics, Vol. 26, Issue 1, pp. 29–41, 1996.

[33] Verma, O.P.; Hanmandlu, M.; Sultania, A.K.; Dhruv, D.,”**A Novel Fuzzy Ant System For Edge Detection”**, Proceedings of [9th International Conference on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5589121)  Computer and Information Science, Yamagata, Japan, pp.228 - 233, 2010.

[34] H. Nezamabadi-pour, S. Saryazdi and E. Rashedi, “Edge detection using ant algorithms,” Soft Computing, Vol. 10, pp. 623–628, May 2006.

[35] Jing Tian, Weiyu Yu, and ShengliXie，“An ant colony optimization algorithm for image edge detection”, *2008 Congress on Evolutionary Computation*, pp.751-756, Jun. 2008.

[36] De-Sian Lu and Chien-Chang Chen, “Edge detection improvement by ant colony optimization,” Pattern Recognition Letters, Vol. 29, pp. 416-425, Mar. 2008.

[37] A. Jevtic, J. Quintanilla-Dominguea, M. G. Cortina-Januchs and D. Andina, “Edge detection using ant colony search algorithm and multi-scale contrast enhancement,” 2009 IEEE conf. on Systems, Mans, and Cybernetics, USA, pp. 2193-2198, Oct. 2009.

[38] Jian Zhang, Kun He, Jiliu Zhou, “An Ant Colony Optimization Algorithm for Image Edge Detection”, International Conference on Artificial Intelligence and Computational Intelligence, 2010.

[39] barkhoda, W.; Tab, F.A., Shahryari, O.K., “ [Fuzzy edge detection based on pixel's gradient and standard deviation values](http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5352742&queryText%3Dfuzzy+edge+detection%26openedRefinements%3D*%26searchField%3DSearch+All)”, Proceedings of [International Multi-conference on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5341930)  Computer Science and Information Technology, Mragowo, pp. 7 – 10, 2009.

[40] YishuZhai; Xiaoming Liu, “**Multiscale Edge Detection Based on Fuzzy C-Means Clustering”,** Proceedings of [1st International Symposium on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=10835) Systems and Control in Aerospace and Astronautics, Harbin, pp. 1201-1204, 2006.

[41] Fujian Wang, YirenXu, Yuming Zhao, Fuqiao Hu, "A new nonlinear interpolation algorithm for edge preserving", International Conference on Multimedia Technology, pp. 1-4, 2010.

[42] Pinheiro, A.M.G., “THE ANGULAR ORIENTATION PARTITION EDGE DESCRIPTOR”, International Conference on Acoustics Speech and Signal Processing, pp. 1250, 2010.

[43]Passino K.M., “Biomimicry of bacterial foraging for distributed optimization and control”, **Control Systems Magazine, IEEE**, Vol. 22, [Issue: 3](http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=21675&isYear=2002) pp. 52-67, Jun 2002.

[44] Verma, O.P., “Fuzzy Edge Detection Based on Similarity Measure in Colour Image”, Annual IEEE India Conference, pp. 1 – 6, 2011.

[45] Recep Demirci, “Similarity relation matrix based color edge detection”, AEU - International Journal of Electronics and Communications, [Vol. 61, Issue 7](http://www.sciencedirect.com/science?_ob=PublicationURL&_tockey=%23TOC%2320469%232007%23999389992%23659681%23FLA%23&_cdi=20469&_pubType=J&view=c&_auth=y&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=bab7dae514367cba6188ec3ef20d72a4), pp. 469-477, 2 July 2007.

[46] E. Rosch, “Cognitive representation of semantic categories”. Journal of Experimental Psychology: General, vol. 104, pp. 192–233, 1975.

[47] R. M. Nosofsky, “Attention, similarity and the identification- categorization relationship”, Journal of Experimental Psychology: General, vol. 115, pp. 39-57, 1986.

[48] A. Tversky, “Feature of similarity”, Psychological review, vol. 84, 327-352, 1977.

[49] L. A. Zadeh, “Similarity relations and Fuzzy ordering”, Information Science, Vol. 3, pp- 177-200, 1971.

[50] W. J. Wang, “New similarity measures on fuzzy sets and on elements”, Fuzzy Sets and Systems, Vol. 85, 3, pp. 305-309, 1997.

[51] S. Santini, and R. Jain, “Similarity Measures”, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 21, No. 9, pp. 871-883, 1999.

[52] Y. A. Tolias, S. M. Panas and L. H. Tsoukalas, “Generalized fuzzy indices for similarity matching”, Fuzzy Sets and Systems, Vol. 120, 2, pp 255-270, 2001.

[53] S. M. Wuerger, L. T. Maloney, J. Krauskopf, “Proximity judgments in color space: tests of a Euclidean color geometry”, Vision Res. 35 (6), pp. 827-835, 1995.

[54] M. Seaborn, L. Hepplewhite and J. Stonham , “Fuzzy colour category map for the measurement of colour similarity and dissimilarity”, Pattern Recognition Letters, Vol. 38, 2, pp. 165-177, 2005.

[55] Recep Demirci, “Rule-based automatic segmentation of color images ”, AEU - International Journal of Electronics and Communications, Vol. 60, Issue 6, pp 435- 442, 2006.

[56] Cohen, Jacob, "A coefficient of agreement for nominal scales", Educational and Psychological Measurement, pp. 37–46, 1960.

[57] C.E. Shannon, “A Mathematical Theory of Communication”, Bell System Technical Journal, vol. 27, pp. 379-423, 623-656, July - October, 1948