Chapter 1

INTRODUCTION

1.1 Image Definition

A digital image is a representation of a two-dimensional image as a finite set of digital values. In image processing, the digitization process includes sampling and quantization of continuous data. The sampling process samples the intensity of the continuous-tone image, such as a monochrome, color or multi-spectrum image, at specific locations on a discrete grid. The grid defines the sampling resolution. The quantization process converts the continuous or analog values of intensity brightness into discrete data, which corresponds to the digital brightness value of each sample, ranging from black, through the grays, to white. A digitized sample is referred to as a picture element, or pixel. The digital image contains a fixed number of rows and columns of pixels. Pixels are like little tiles holding quantized values that represent the brightness at the points of the image. Pixels are parameterized by position, intensity and time. Typically, the pixels are stored in computer memory as a raster image or raster map, a two-dimensional array of small integers. Image is stored in numerical form which can be manipulated by a computer. A numerical image is divided into a matrix of pixels (picture elements).

Digital image processing allows one to enhance image features of interest while attenuating detail irrelevant to a given application, and then extract useful information about the scene from the enhanced image. Images are produced by a variety of physical devices, including still and video cameras, x-ray devices, electron microscopes, radar, and ultrasound, and used for a variety of purposes, including entertainment, medical, business (e.g. documents), industrial,

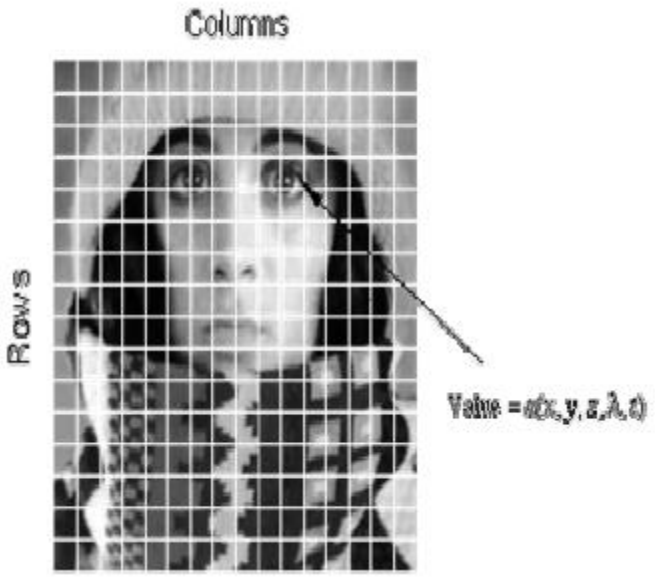


Figure 1.1: Digitization of a continuous image.

military, civil (e.g. traffic), security, and scientific. The goal in each case is for an observer, human or machine, to extract useful information about the scene being imaged.

1.2 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.

1.3 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.

i) 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.

ii) 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.

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

iv) 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.

v) 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.

1.4 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.

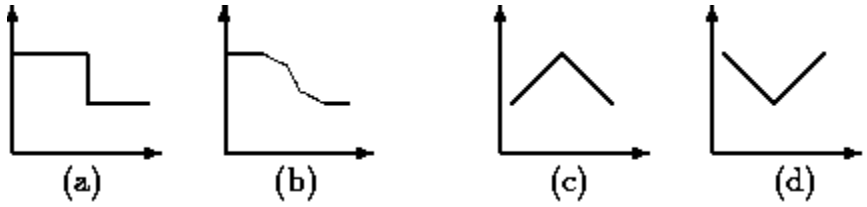


Figure 1.2: Types of Edges (a) Sharp step (b) Gradual step (c) Roof (d) Trough

A Sharp Step, as shown in Figure 1.2(a), 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 1.2(b), 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 1.2(c), 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 1.2(d), is the inverse of a Roof.

1.5 Literature Survey of Edge Detection

In this section, work done in the area of edge detection is reviewed and focus has been made on detecting the edges of the digital images using fuzzy logic approach. Edge detection is a problem of fundamental importance in image analysis. In typical images, edges characterize object boundaries and are therefore useful for segmentation, registration, and identification of objects in a scene. Edge detection of an image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image [1].

In 1997 Ng Geok See, Chan Khue Hiang, proposed a technique for edge detection based on neural network. Neural network has many processing elements joined together and usually organized into groups called layers. Training is provided to the neural network in supervised or unsupervised learning mode, to force the network to yield particular result to a specific input [2].

In 1998 Zhengquan He, M.Y.Siyal, proposed a new technique based on neural network. Most of the existing techniques like Sobel [3] are effective in certain senses and require more computation time. [4].The proposed edge detection technique a three layer BP neural network is employed to classify the edge elements in binary images into one of the predefined categories. To detect edges first binarize the image by choosing threshold by some optimal criteria and classify the edge patterns of binary images in different categories. Train the neural network on

these patterns and on their noisy patterns. After the network is trained, it can recognize the input pattern as a most like pattern in our edge pattern bank. This technique is more flexible to the edge structures in the image. It can not only extract straight lines but also can extract corners and arcs edges.

In 2005 Zhang, Zhao and Li Su, proposed a technique based on the integer logarithm ratio of gray levels. In order to remove the ability of noise rejection they proposed a ratio of gray levels between the two successive image points rather than the difference of gray levels to denote the variation in the gray levels [5]. In this, division operation becomes the subtraction operation of the logarithmic ratio of gray levels. This is more convenient for calculations.

In 2005 Stamatia Giannarou, Tania Stathaki, proposed a technique that allows combining the methods of different edge detection operators in order to yield improved results for edge detection in an image. This is called Receiver Operating Characteristics (ROC) analysis [6]. This technique uses the statistical approach to automatically form a optimum edge map, by combining edge images from different detectors. The characteristics of this method are to produce accurate and noise free results. One possible concern regarding these techniques is the selection of the edge detectors to be combined.

In 2006 M.Hanmandlu, Rohan Raj Kalra, Vamsi Krishna Madasu, proposed a technique based on Univalue Segment Assimilating Nucleus (USAN) area i.e. fuzzy technique. The USAN characterizes the structure of the edge present in the neighborhood of a pixel and can thus be considered as a unique feature of the pixel and is fuzzified [7]. This technique is best in yielding the large number of longest edge segments. This is used for the applications like face recognition

and fingerprint identification, as it does not distort the shape of the image and is able to retain all the important edges.

Later on Fast fuzzy edge detection technique was proposed. Heuristic membership functions, simple fuzzy rules, and fuzzy complements were used to develop new edge detectors [8]. Then Fuzzy edge detector using entropy optimization was proposed. The proposed fuzzy edge detector involves two phases: – global contrast intensification and local fuzzy edge detection. In the first phase, a modified Gaussian membership function is chosen to represent each pixel in the fuzzy plane [9].

To realize the fast and accurate detection of the edges from the blurry images, the Fast Multilevel Fuzzy Edge Detection (FMFED) algorithm was proposed [10]. The FMFED algorithm first enhances the image contrast by means using a simple transformation function based on two image thresholds. Second, the edges are extracted from the enhanced image by the two-stage edge detection operator that identifies the edge candidates based on the local characteristics of the image and then determines the true edge pixels using the edge detection operator based on the extreme of the gradient values.

After going through all the techniques in area of image it was realized that fuzzy is the most accurate and best technique. So the major stress will be on development of algorithms for improving the quality of detecting edges by Edge detection technique using fuzzy logic approach.

Mehul et al. [11] present Fuzzy logic based automatic edge thresholding technique for edge detection that overcomes the drawback of Dong Liu, algorithm [12]. Another technique for edge detection proposed in [13] 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.* [14]. For instance Verma *et al.* [15] 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 [16-19]. A recent approach using the ant colony optimization is proposed by Jian Zhang et al.[20] 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.* [21] 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.* [22] uses both multiscale wavelet transform and fuzzy c-means clustering algorithm to obtain the edge map of the image adaptively.

1.6 Brief review of Fuzzy Logic

Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing [23]. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions. As shown in Figure 1.3 fuzzy logic and probability theory are the most powerful tools to overcome the imperfection.

Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge

and experience. Fuzzy logic allows expressing the knowledge with subjective concepts such as very hot, bright red and very small height, which are mapped into exact numeric ranges.

The notion central to fuzzy systems is that truth values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value on the range [0.0, 1.0], with 0.0 representing absolute Falseness and 1.0 representing absolute Truth. For example, let us take the statement:

"Jane is old."

If Jane's age was 75, we might assign the statement the truth value of 0.80. The statement could be translated into set terminology as follows:

"Jane is a member of the set of old people."

This statement would be rendered symbolically with fuzzy sets as:

mOLD(Jane) = 0.80

Where m is the membership function, operating in this case on the fuzzy set of old people, which returns a value between 0.0 and 1.0.

Fuzzy logic and Fuzzy set theory provide a rich and meaningful addition to standard logic. The mathematics generated by these theories is consistent, and fuzzy logic may be a generalization of classic logic. Fuzzy Logic has been gaining increasing acceptance during the past few years. There are over two thousand commercially available products using Fuzzy Logic, ranging from washing machines to high speed trains. Nearly every application can potentially realize some of the benefits of Fuzzy Logic, such as performance, simplicity, lower cost, and productivity [23].

1.6.1 Structure of Fuzzy Image Processing

Fuzzy image processing is not a unique theory. Fuzzy image processing is the collection of all approaches that understand, represent and process the images, their segments and features as fuzzy sets. The representation and processing depend on the selected fuzzy technique and on the problem to be solved. Fuzzy image processing has three main stages: image fuzzification, modification of membership values, and, if necessary, image defuzzification. (Figure 1.4)

Figure 1.3: The general structure of fuzzy image processing

The fuzzification and defuzzification steps are due to non availability fuzzy hardware. Therefore, the coding of image data (fuzzification) and decoding of the results (defuzzification) are steps that make possible to process images with fuzzy techniques. The main power of fuzzy image processing is in the middle step i.e. modification of membership values, Figure1.4. After the image data are transformed from gray-level plane to the membership plane (fuzzification), appropriate fuzzy techniques modify the membership values.

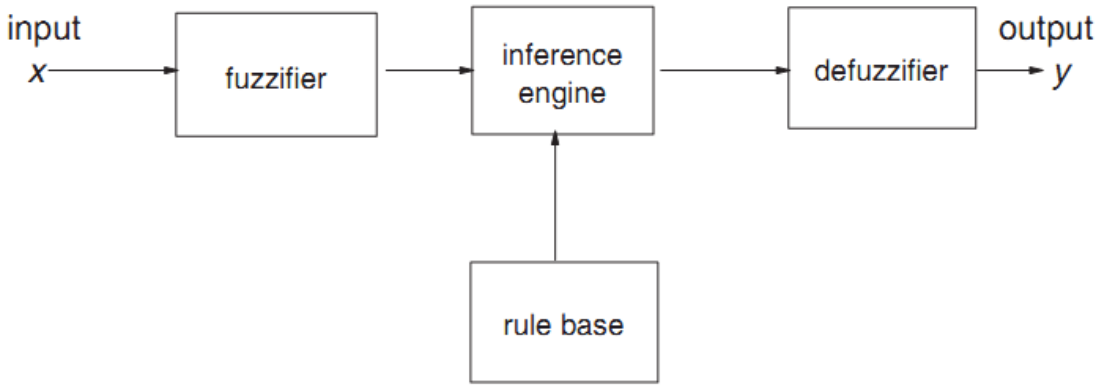


Figure 1.4: Steps of Fuzzy Image Processing

1.6.2 Motivation behind Fuzzy Logic

In many image processing applications, we have to use expert knowledge to overcome the difficulties (e.g. object recognition, scene analysis). Fuzzy set theory and fuzzy logic offer us powerful tools to represent and process human knowledge in form of fuzzy if-then rules. On the other side, many difficulties in image processing arise because the data/tasks/results are uncertain. This uncertainty, however, is not always due to the randomness but to the ambiguity and vagueness. Beside randomness which can be managed by probability theory we can distinguish between three other kinds of imperfection in the image processing :-

a) Grayness ambiguity

b) Geometrical fuzziness

c) Vague (complex/ill-defined) knowledge

The question whether a pixel should become darker or brighter than it already is, the question where is the boundary between two image segments, and the question what is a tree in a scene

analysis problem, all of these and other similar questions are examples for situations that a fuzzy approach can be the more suitable way to manage the imperfection.

General observations about fuzzy logic are:

a) Fuzzy logic is conceptually easy to understand. The mathematical concepts behind fuzzy reasoning are very simple.

b) Fuzzy logic is flexible. With any given system, it's easy to manage it or layer more functionality on top of it without starting again from scratch.

c) Fuzzy logic is tolerant of imprecise data. Everything is imprecise if we look closely enough, but more than that, most things are imprecise even on careful inspection. Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end.

d) Fuzzy logic can model nonlinear functions of arbitrary complexity. We can create a fuzzy system to match any set of input-output data. This process is made particularly easy by adaptive techniques like ANFIS (Adaptive Neuro- Fuzzy Inference Systems), which are available in the Fuzzy Logic Toolbox.

e) Fuzzy logic can be blended with conventional control techniques. Fuzzy systems don't necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation.

Chapter 2

BLURRING

2.1 Introduction of Blurring

Blur is an artifact that occurs in images due to un-regularities in image acquisition. The blurring process can be described as some kind of weighted averaging of pixel values in a certain neighborhood. Figure 2.1 illustrates this. Each pixel of the blurred picture on the right is a weighted average of pixels from the original image on the left. The weights of this averaging process are visualized in the blur image in the middle, where the neighborhood of a pixel is shown. White pixels give a large contribution, whereas dark pixels give a small contribution.

Original image blur blurred image

Figure 2.1: A blurring process. Left: Original, unblurred image. Middle: Visualization of the blur. Right: Blurred image.

Motion blur of this kind is normally caused by camera or object motion in x-direction during the camera exposure time, whereas the disc-like defocus blur comes from photographing objects that are outside the focus of the camera. Atmospheric perturbations or lens errors are normally the

cause of Gaussian blurs and the “shake" blur arises e.g. when one takes a picture with a small hand-held camera and his hands are shaking.

2.2 Spatial Variance

If the blur is the same for every pixel in the blurred image, then we talk about spatially invariant blur. Figure 2.2 is an example for such a process. There is also another type of blur, namely spatially variant blur. In this case the blur may vary over the image. An example of an image that results from this type of blur is figure 2.2. Clearly, the television in the background is much more blurred than the pets in the foreground.

Figure 2.2: Spatially variant blur.

Although two different causes can give the same blur, in most cases the type of a blur is significant for a cause. So one can conclude from an image, how it got blurred. Figure 2.3 shows some common types of blur.

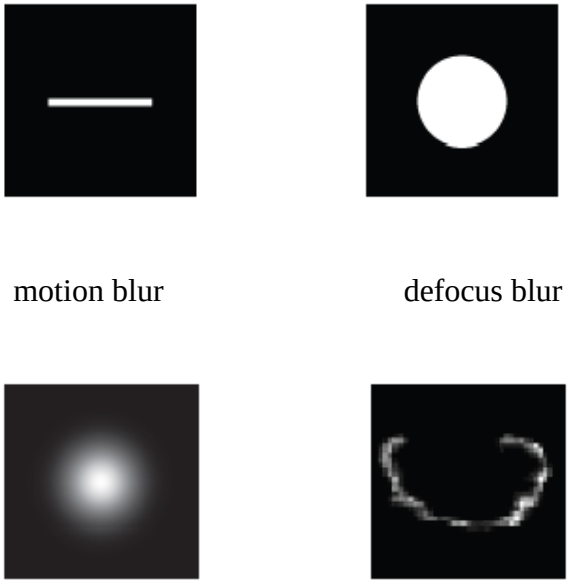
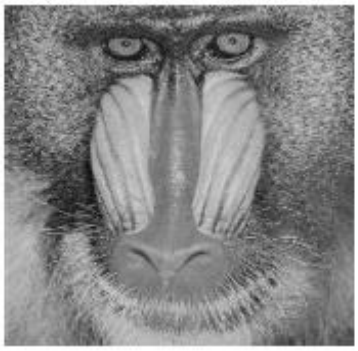


Figure 2.3: Different types of blur.

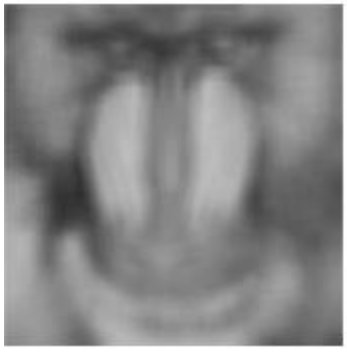
Although two different causes can give the same blur, in most cases the type of a blur is significant for a cause. So one can conclude from an image, how it got blurred. Figure 2.3 shows some common types of blur.

One now may ask oneself how pictures deviated by such blurs look like. This is illustrated by figure 2.4. In this example, the images are blurred in a spatially invariant way. It is easy to observe that the blurred images look differently.

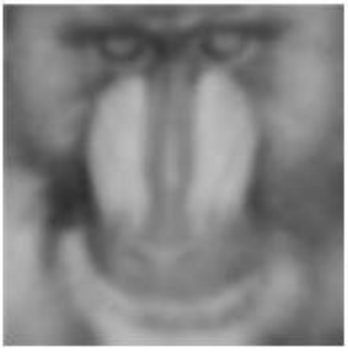


sharp image

motion blurred



defocused image



Gaussian blurred image

Figure 2.4: Blurred images.

Chapter 3

DEBLURRING

3.1 Introduction of Deblurring

Image deblurring is an inverse problem whose aim is to recover an image from a version of that image which has suffered a linear degradation, with or without noise. This blurring degradation can be shift-variant or shift-invariant. Although there have been some proposed methods for recovering shift variant linear degradations [24]–[31], the majority of existing deblurring methods was developed for invariant degradations, and the blind recovery from shift-invariant degradations is still considered a rather challenging problem. This paper focuses on shift-invariant blurs, and, in the context of this paper, "blur" will refer to a linear, shift-invariant degradation, i.e., a convolution, with or without noise, unless stated otherwise.

Automatic image deblurring is an objective of great practical interest for the enhancement of images in photo and video cameras [32]–[34], in astronomy [35], in remote sensing [36], in to mography [37], [38], in other biomedical imaging techniques [39]–[41], etc.

Image deblurring methods can be divided into two classes: non-blind, in which we assume the blurring operator to be known, and blind, in which we assume that the blurring operator is unknown. The method that we describe here belongs to the latter class. The application range of non-blind methods is much narrower than the one of blind methods: in most situations of practical interest the blurring filter's impulse response, also called point spread function (PSF), is not known with good accuracy. Since non-blind deblurring methods are very sensitive to mismatches between the PSF used by the method and the true blurring PSF, a poor knowledge of the blurring PSF normally leads to poor deblurring results.

Despite its narrower applicability, non-blind deblurring already is a difficult problem. The main difficulty faced by non-blind deblurring methods has to do with the presence of noise in the blurred image. Since the blurring operator typically is very ill-conditioned, this noise, even if very weak, can strongly contaminate the deblurred image. The problem is serious in situations in which the blurring PSF is exactly known, and gets worse if there is even a slight mismatch between the PSF used for deblurring and the one that caused the blur. Most non-blind deblurring methods [42]–[46] overcome this difficulty through the use of prior information about the image to be recovered, often doing this within a Bayesian or maximum a posteriori framework.

In blind image deblurring (BID), not only the degradation operator is ill-conditioned, but the problem also is, inherently, severely ill-posed: there is an infinite number of solutions (original image + blurring filter) that are compatible with the degraded image. For an overview of BID methods, see [47] and [48].

Most previously published blind deblurring methods are very limited, since they do not allow the use of a generic PSF. Most of them are based, instead, on PSF models with a small number of parameters [49]–[53]. For example, to model an out-of-focus blur, they normally use a circle with uniform intensity, having as single parameter the circle's radius [49]. Similarly, to model a motion blur, they normally use a straight-line segment with uniform intensity, the only parameters being length and slope [49]–[51]. These approaches are very limited, because such models rarely fit actual blurring PSFs well. For example, the out-of-focus blurring PSF generally is more complex than a simple uniform circle, and the camera motion that causes a motion blur generally is much more complex than a uniform, straight-line motion. And, as was emphasized above, even a slight mismatch between the deblurring PSF and the blurring PSF strongly degrades the quality of the deblurred image.

A recent work [51] manages to estimate the blur under the variational Bayesian approach. However, this method models the blur by means of a Gaussian filter, which is completely defined by a single parameter (the Gaussian's variance), and is a very weak model for real-life blurs.

In an attempt to encompass less restrictive blurs, a fuzzy technique that uses several prespecified PSF models has been considered in [53]. Another blind deconvolution method, which is fast and has a proof of convergence, is described in [54]. However, this method assumes that the PSF is zero-phase and, furthermore, depends on the existence of a good initial estimate of the PSF.

An interesting method for blind deblurring of color images was proposed in [55]. This method appears not to pose any strong restrictions on the blurring filter. In the cited paper, several experimental results on synthetic blurs are shown, but little information is provided about them. From the information that is given, it appears that the blurring filters that were used in the experiments were either circularly symmetric (including simulated out-of-focus blurs), or corresponded to straight-line motion blurs. There seems to be no reason for the method not to be able to successfully deal with other kinds of blurs, however.

The difficulties of PSF estimation and image deconvolution give rise to recent research efforts. These efforts attempt to solve the deblurring problem with additional conditions [56], [57]. The characteristics of multi-exposed images have been exploited to obtain an image adopting the advantages of each of the images [56]. Otherwise, the PSF is estimated from a short-time exposed image, and deconvolution is applied based on the estimated PSF [57]. Even though recent techniques produce satisfactory results, the computational complexities of these methods can be a problem when implemented into real applications. Also, since a short-time exposed

image tends to have a low signal-to-noise ratio (SNR), the information from this image is not always reliable.

In this letter, we propose a novel image deblurring algorithm using multiple blurred images to reconstruct an original scene. The mutually different pieces of information from the multiple blurred images of a scene are merged in the frequency domain by using the projection onto convex sets (POCS). When merging the information, the concept of the fuzzy convex set [58] is exploited to express the reliability of each spectrum component. Since only reliable frequency component is extracted from each blurred image, the resultant image contains the maximum possible information of the scene. Thus, image blur is efficiently reduced.

3.2 PROPOSED ALGORITHM FOR DEBLURRING

In general, image degradation is modeled by:

$$g = h * f + n \qquad (1)$$

Where f, g, h and n denote the original image, the blurred image, the PSF, and the observation noise, respectively. The degradation model for multiple images is expressed as:

$$g_i = h_i * f + n_i \quad i = 1,2,3 \ldots \ldots N \qquad (2)$$

In (2) g_i, h_i and n_i denote the Nth blurred image, PSF, and observation noise, respectively. The objective of deblurring with multiple images is to estimate f using N blurred images. Since N images are assumed to be blurred by the different PSFs, each blurred image tends to have different information on the original image.

Wavelet

The fundamental idea behind wavelets is to analyze according to scale. The wavelet analysis procedure is to adopt a wavelet prototype function called an analyzing wavelet or mother wavelet. Any signal can then be represented by translated and scaled versions of the mother wavelet.

Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques such as Fourier analysis miss aspects like trends, breakdown points, discontinuities in higher derivatives, and self-similarity. Furthermore, because it affords a different view of data than those presented by traditional techniques, it can compress or de-noise a signal without appreciable degradation [7].

A wavelet prototype function at a scale *s* and a spatial displacement *u* is defined as:

$$\psi_{s,u}(x) = \sqrt{s}\ \psi\left[\frac{(x-u)}{s}\right] \tag{3}$$

Replacing the complex exponential in Equation 23 with this function yields the continuous wavelet transform (CWT):

$$C(x,u) = \int_{-\infty}^{+\infty} f(t)\sqrt{s}\ \psi\left[\frac{(x-u)}{s}\right] \tag{4}$$

which is the sum over all time of the signal multiplied by scaled and shifted versions of the wavelet function ψ. The results of the CWT are many wavelet coefficients C, which are a function of scale and position. Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal.

We analyze differently blurred multiple images on the same scene in the frequency domain. The original image spectrum is distorted by the blur, which has a low-pass characteristic in the frequency domain. Then, even though all image spectrums are damaged by different blurs, each spectrum tends to have mutually different partial information of the original image spectrum Therefore, it is possible to gather this information for compensating the lost frequency components.

In the proposed method, we exploit the concept of fuzzy convex sets [5], [7] to merge the mutually different pieces of information. Let A_f denote the fuzzy set on the universal set E. Then, an α-cut of A_f ,A_f^{α} is defined as

$$A_f^{\alpha} = \begin{cases} \{x|\mu_A(x) \geq \alpha, x \epsilon E\} & \text{if } \alpha \neq 0 \\ E & \text{otherwise} \end{cases} \quad (5)$$

Where $0 \leq \alpha \leq 1$ and $\mu_A(x)$ is the membership function that maps in E to a real value [5]. For an ordinary (nonfuzzy) set A , $\mu_A(x)$ is defined as

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \epsilon A \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

However, for a fuzzy set, the membership function returns a real value whose range is [0, 1]. In this case, the membership function value, μA(x), represents the strength of being a member of A. Then, the fuzzy set is called A_f convex if and only if the following condition is satisfied:

$$\mu_A[\lambda x1 + (1-\lambda)x2\,] \geq \min\{\mu_A(x1), \mu_A(x2)\} \quad (7)$$

Where $0 \leq \lambda \leq 1$

Given the definition of the membership function, we can expand the convex sets by morphological dilation [5], [6]. We use the fuzzy convex set theory for the multiple image deblurring problem. Let G_i denote the ith spectrum of the blurred image. For given blurred images, we can obtain spectrums using the Wavelet Transform. The strength of each frequency coefficient being a member of the original spectrum can be represented by using the membership function. We define the membership function as follows:

$$\mu_A\big(G_i(u,v)\big) = 1 - e^{-|G_i(u,v)|} \tag{8}$$

Where u and v are spectrum coordinates, γ is a positive scalar, | | and denotes the magnitude of the corresponding coefficient. Note that the larger the magnitude is, the larger the membership function value is obtained. This is because the coefficient of large magnitude is less affected by the blur. Therefore, among N spectrums, the coefficient whose magnitude is larger than the others is more likely to be a member of the resultant spectrum. Also, since the membership function in (6) is a monotonically increasing function, it satisfies (5). Therefore, A_f is a convex set. Then, the projection onto the convex α-cut, A_f^{α} , is defined as

$$P^{\alpha}\big(G_i(u,v)\big) = \begin{cases} 0 & \text{if} \mu_A(G_i(u,v)) < \alpha \\ G_i(u,v) & \text{otherwise.} \end{cases} \tag{9}$$

By projecting all coefficients using P^{α} , every coefficient whose membership function value is less than α is mapped to 0. Then, the expansion of the convex set is performed by decreasing which is called dilation of the convex set [5].

Based on the above definitions, we propose a deblurring algorithm whose procedure is given as follows.

Step 1: First of all we decompose all the N blurred images using 2-dimensional Stationary Wavelet [Reference] at single level decomposition. Now we get N Approximate, N Horizontal ,N Vertical and N Diagonal spectrums.

Step 2: Now we perform following steps for N Approximate spectrums.

Step i: initialize αmax ,αmin and αstep .

Step ii: set α = αmax ,i=1, and F=G1

Step iii: for every u and v project F onto A_f^{α} by (7) as follows:

$$\hat{F}(u,v) = P^{\alpha}(\hat{F}(u,v) \tag{10}$$

Step IV: For every u and v

$$\hat{F}(u,v) = \begin{cases} \hat{F}(u,v) & \text{if} |G_{i+1}(u,v)| \leq |\hat{F}(u,v)| \\ G_{i+1}(u,v) & \text{otherwise.} \end{cases} \tag{11}$$

Step v: if i<N, i=i+1 and go to step iii. Otherwise go to step vi

Step vi: if α = αmin , go to step viii. Otherwise go to step vii.

Step vii: set i=1 and α= α- αstep. Go to step iii

Step 3: Similarly step 2 will perform for all N Horizontal, N Vertical and N diagonal spectrums.

Step 4: obtain the final result $\hat{f}$ by reconstructing the spectrums.

When we merge the spectrums, we only consider the magnitude of the coefficient. Therefore, the resultant spectrum tends to have a phase distortion. We found that this distortion produces artifacts in the outcome image. These distortions are especially visible in the smooth region of the image. Based on the fact that the smooth region is not severely damaged by the blur, we

replace the smooth region by the average of the N sample pixel values in the same position [see (8) at the bottom of the page], where g_k denotes kth blurred image, i and j are image coordinates, var(.) and avg(.) return the variance and average value, respectively. The threshold is empirically chosen to classify the smooth region. From this post-processing, the artifacts caused by the phase distortion can be effectively reduced.

$$\hat{f}(i,j) = \begin{cases} \hat{f}(i,j) & \text{if } \mathrm{var}\left(g_1(i,j), \ldots \ldots g_n(i,j)\right) > \sigma_{th}^2 \\ \mathrm{avg}\left(g_1(i,j), \ldots \ldots g_n(i,j)\right) & \text{otherwise} \end{cases} \qquad (12)$$

Chapter 4

EDGE DETECTION USING BACTERIA FORAGING

4.1 Bacteria Foraging

A new evolutionary technique, called Bacterial Foraging scheme, was introduced by K.M.Passino [59].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

1 depicts how clockwise and counter clockwise movement of a bacterium take place in a nutrient solution.

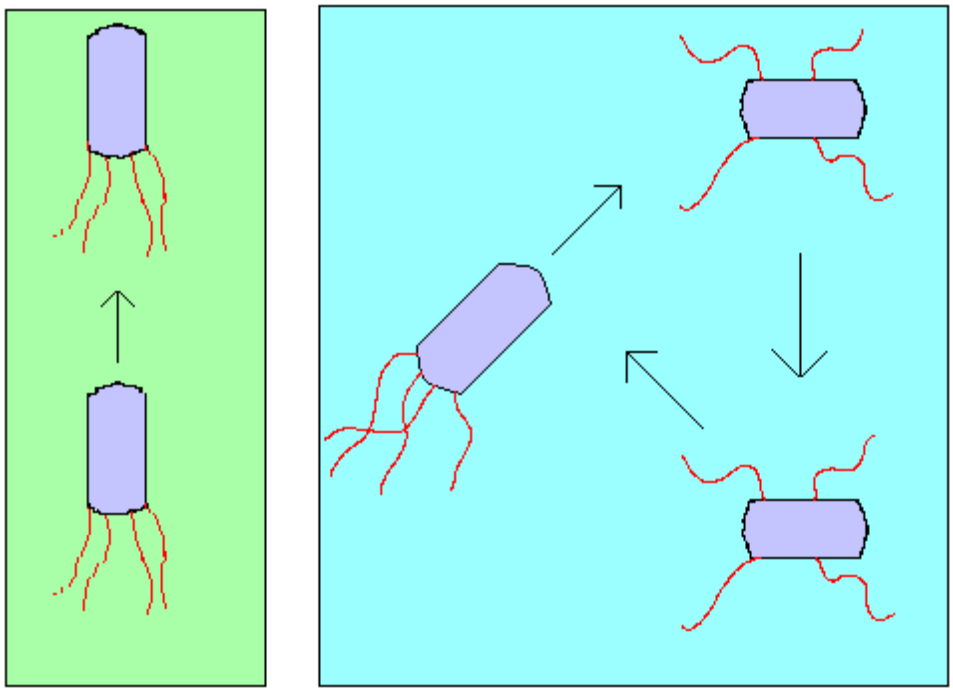


Fig.4.1. Swim and tumble of a bacterium

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 $J(\theta)$ where p $\theta \in \Re$ (i.e. θ is a p-dimensional vector of real numbers), and we do not have measurements or an analytical description of the gradient $\nabla J(\theta)$. 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.

N_{re} : The number of reproduction steps,

N_{ed} : The number of elimination-dispersal events,

P_{ed} : Elimination-dispersal probability,

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

Let $P(j, k, l) \{ (j, k, l) \mid i$ 1,2,..., $S\}$ 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 $\theta_p^i(j, k, l) \in \Re$ (sometimes we drop the indices and refer to the i-th bacterium position as θ^i). 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 $\theta^i(j, k, l)$ represents i-th bacterium at jth 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

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \qquad (13)$$

Where Δ 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.

$$J_{cc}\big(\theta, P(j,k,l)\big) = \sum_{i=1}^{S} J_{cc}\left(\theta, \theta^{i}(j,k,l)\right)$$

$$= \sum_{i=1}^{S}\left[-d_{attractant}\exp\left(-w_{attractant}\sum_{m=1}^{P}\left(\theta_m - \theta_m^i\right)^2\right)\right]$$

$$+ \sum_{i=1}^{S}\left[-h_{repellant}\exp\left(-w_{repellant}\sum_{m=1}^{P}\left(\theta_m - \theta_m^i\right)^2\right)\right] \qquad (14)$$

where $J\ cc\ (\theta,\ P(\ j,\ k,\ l))$ 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 $\theta = [\theta_1, \theta_2, \ldots, \theta_P]^T$ is a point in the p-dimensional search domain. $d_{attractant}, w_{attractant}, h_{repellant}, w_{repellant}$ 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:

4.2 The BFOA Algorithm

Parameters:

[Step 1] Initialize parameters $p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C(i)(i=1,2\ldots S), \theta^i$.

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

Where, J_{cc} is defined in (2).

[c] Let $J_{last}=J(i, j, k, l)$ to save this value since we may find a better cost via a run.

[d] Tumble: generate a random vector $\Delta(i) \in \Re_p$ with each element i=1,2,..., p .

Δ is a random number on [-1, 1].

[e] Move: Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (15)$$

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

$$J(i,j+1,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j+1,k,l), P(j+1,k,l))$$

[g] Swim

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

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

• Let $m=m+1$.

• If $J(i, j+1, k, l) < J_{last}$ (if doing better), let $J_{last} = J(i, j+1, k, l)$ and let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (16)$$

And use this $\theta^i(j+1,k,l)$ to compute the new $J(i, j+1, k, l)$ as we did in [f]

• Else, let $m= N_s$. This is the end of the while statement.

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

[Step 5] If $j < N_c$, 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

$$J_{health}^{i} = \sum_{j=1}^{N_c+1} J(i,j,k,l) \qquad (17)$$

Let 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 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 made are placed at the same location as their parent).

[**Step 7**] If $k < N_{re}$, 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 P_{ed} , 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 $l < N_{ed}$, 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.

4.3 Proposed method

I. Defining the Nutrient Function

Nutrient Function is defined, based on the intensity difference value of bacterium with respect to its neighboring pixels in 8-neighborhood. Each bacterium tries to find out the edge pixels in its 8-neighborhood with the help of intensity difference .For any pixel to become an edge pixel its intensity difference from neighborhood pixel must be greater than a threshold value. Intensity Difference Value (IDV) of pixel p9 is as follows:

$$IDV(p9) = |p9-p1| + |p9-p2| + |p9-p3| + |p9-p4| + |p9-p5| + |p9-p6| + |p9-p6| + |p9-p7| + |p9-p8| \quad (18)$$

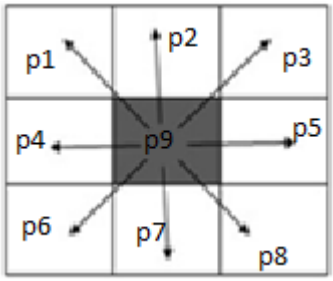


Figure 4.2 Neighboring pixels in Color Image

II. 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 and for that we keep tumbling the bacteria in the whole image. In order to move the current bacterium to its next position, it must satisfy the following condition:

"The next position is the position which has maximum intensity difference value in its 8-neighborhood."

Suppose P is the current position of bacterium. Beginning from the maximum intensity difference pixel P4, bacteria will take its next move. If no such pixel is found, then we eliminate that bacteria and dispersed that to some other location.

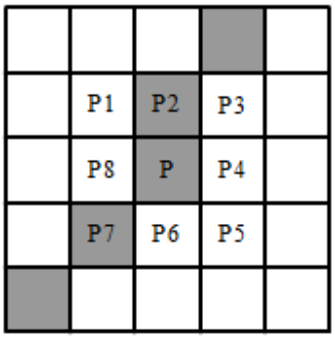


Fig. 4.3 Chemotaxis step: Darker pixels represent the edge and P1, P2,…,P8 are the 8-neighborhood of P.

III. Elimination and Dispersal

If we do not find any edge pixel around bacteria then we have to eliminate that bacterium from that position and dispersed it to some other location.

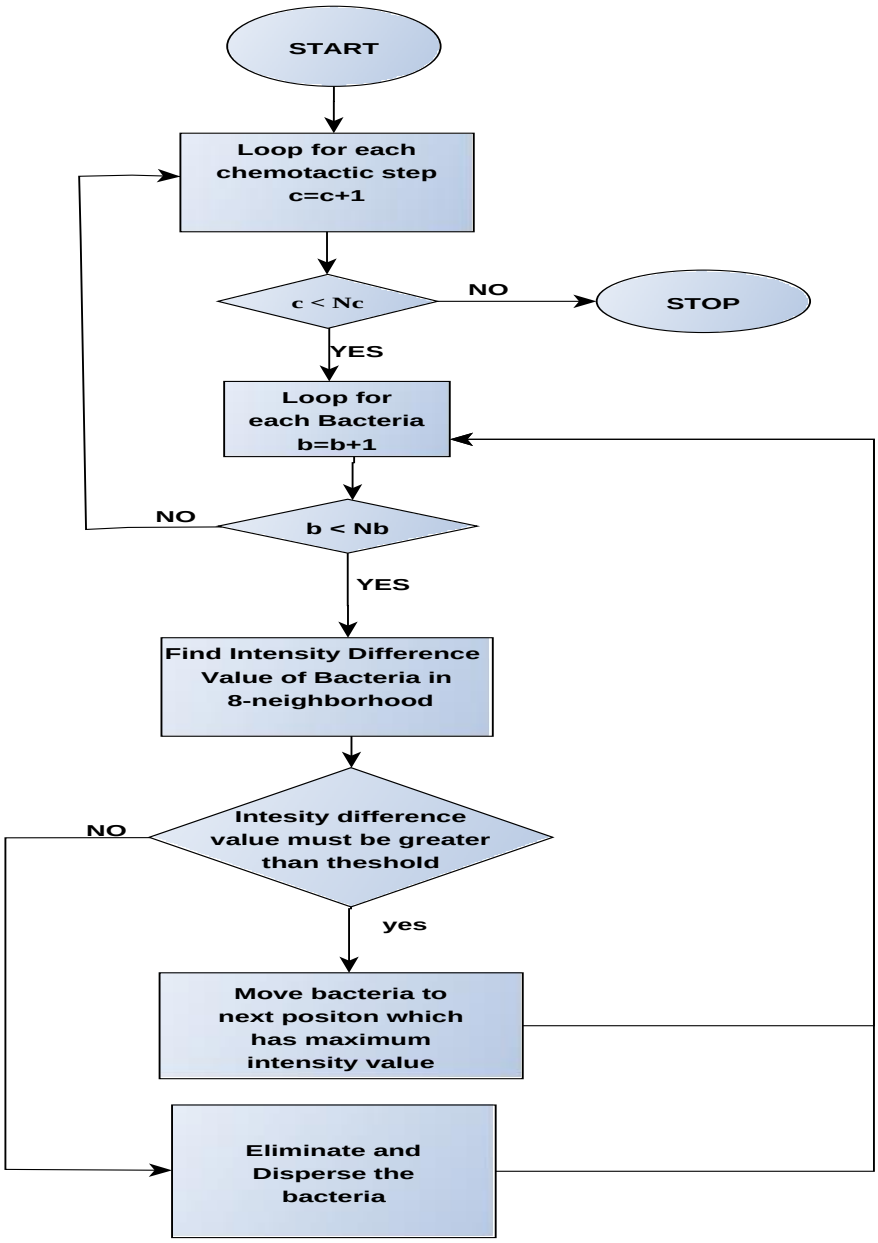


Fig. 4.4 Flow Diagram of proposed method

Chapter 5

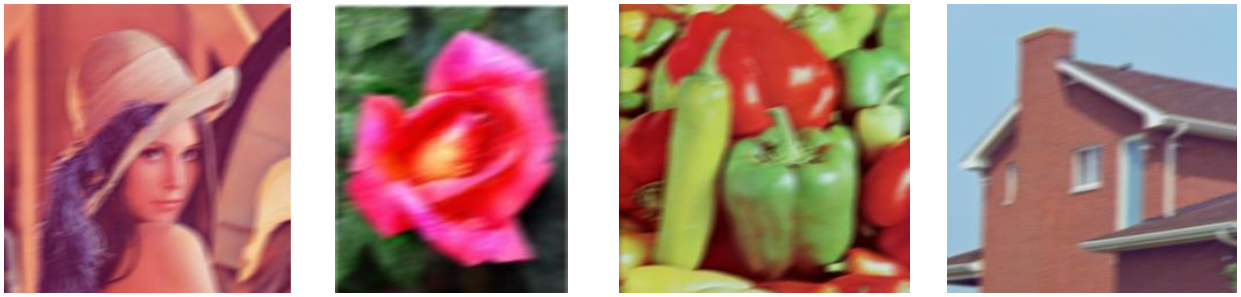
RESULTS AND DISCUSSION

5.1 Result of Deblurring

We tested the proposed method both on synthetic blurred images. Figure 5.1 shows original image of Lena, rose, peppers and house. Figure 5.2 shows blurred images of images shown in Figure 5.1 at L=8 and θ=11. Given the blurred image, the proposed algorithm is performed with empirically chosen parameters: $\gamma=5x10^{-5}$, $\sigma_{th}^2 = 2$, αmax=.99, αmin=0, and αstep=.01. Therefore, 100 iterations it takes to deblurr. Figure 5.3 shows the resultant deburred images corresponding to the blurred images shown in figure 5.2. From the results it clearly shown that even though not all the frequency components can be recovered but the lost frequency components are efficiently compensated.

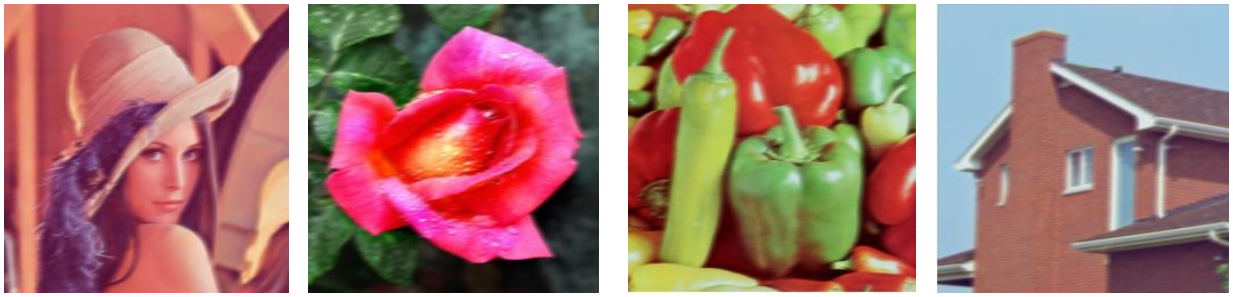
(a) lena (b) rose (c) peppers (d) house

Fig 5.1. Original images



(a) lena (b) rose (c) peppers (d) house

Fig 5.2. Blurred images.



(a) lena (b) rose (c) peppers (d) house

Fig 5.3. Deblurred images

5.2 Result of Edge Detection

The result of the proposed method is being tested against the majority image formed by the result of three other edge detection methods: Canny, Prewit and Sobel. We perform a pixel-by-pixel comparison of the output of the proposed method with the majority image. The methods used for quantitative analysis is:

Shannon's Entropy Function

The information content of the output image is measured by using Shannon's entropy function [64]. It gives the indefiniteness in an image and is calculated as:

$$H(I) = -\sum_{i=0}^{n} p_i log p_i \quad (19)$$

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

There are two parameters that require to be initialized to perform the proposed approach efficiently and effectively. These are initial number of bacteria (Nb), minimum and number of chemotactic steps (Nc), i.e. lifetime of a bacterium.

To analyze the effect of these parameters, Shannon's Entropy quantitative analytic methods are used. Figure 5.4 shows the variation of entropy with respect to the number of initial bacteria. Entropy comes very low for twenty five number of bacteria, since the resultant image (Fig 5.5.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

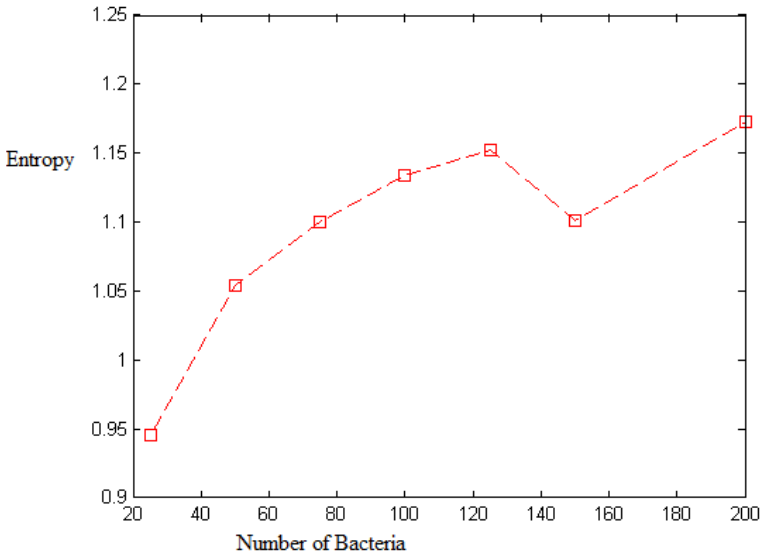


Fig. 5.4 Entropy versus initial number of bacteria

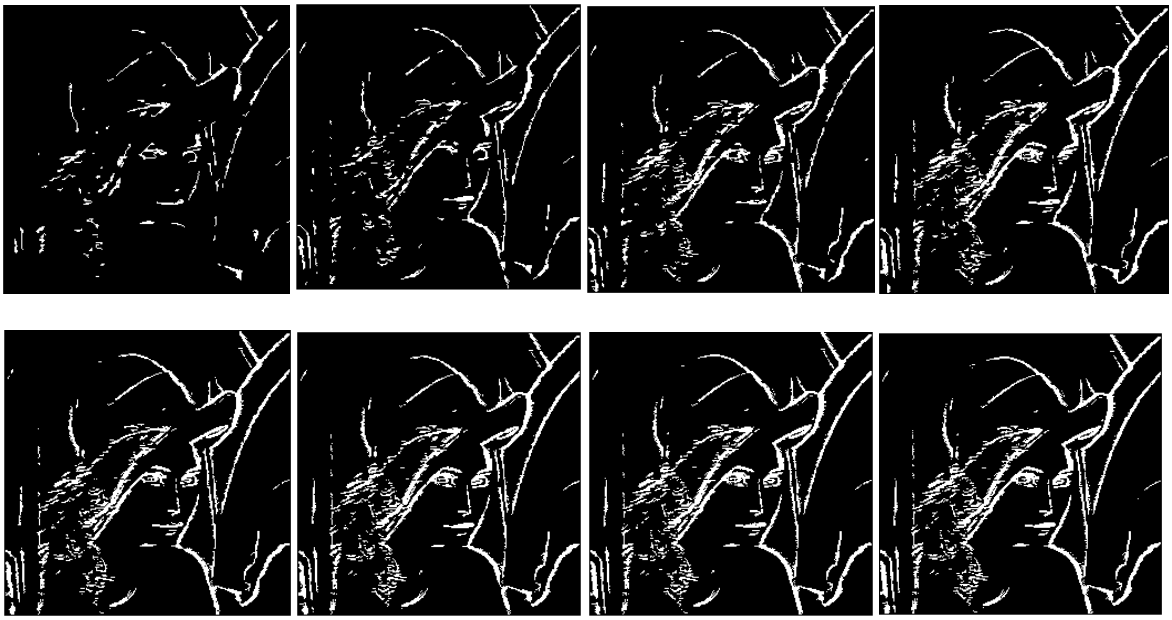


Fig.5.5 Result of proposed approach for different number of initial bacteria (Nb) (a) Nb =25, (b) Nb = 50, (c) Nb = 75, (d) Nb =100, (e) Nb = 125, (f) Nb = 150, (g) Nb = 200, (h) Nb = 250.

form of unconnected weak edges. To have enough amount of information while having the noisy pixels as small as possible, we have taken the initial number of bacteria as 80.

Fig.5.6 shows the variation of number of chemotactic steps (N_c) 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 90. Fig. 5.7 shows result of proposed approach for different number of chemotactic steps.

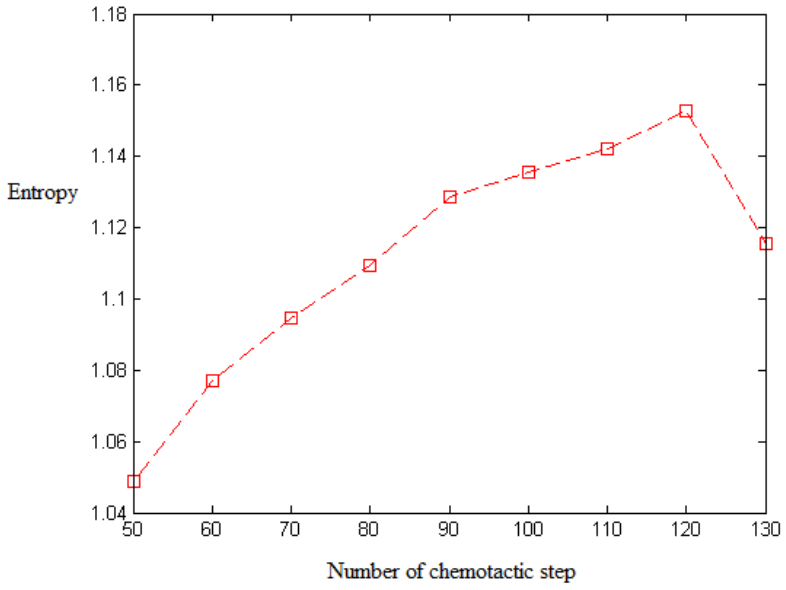


Fig. 5.6 Entropy versus number of chemotactic steps (N_c)

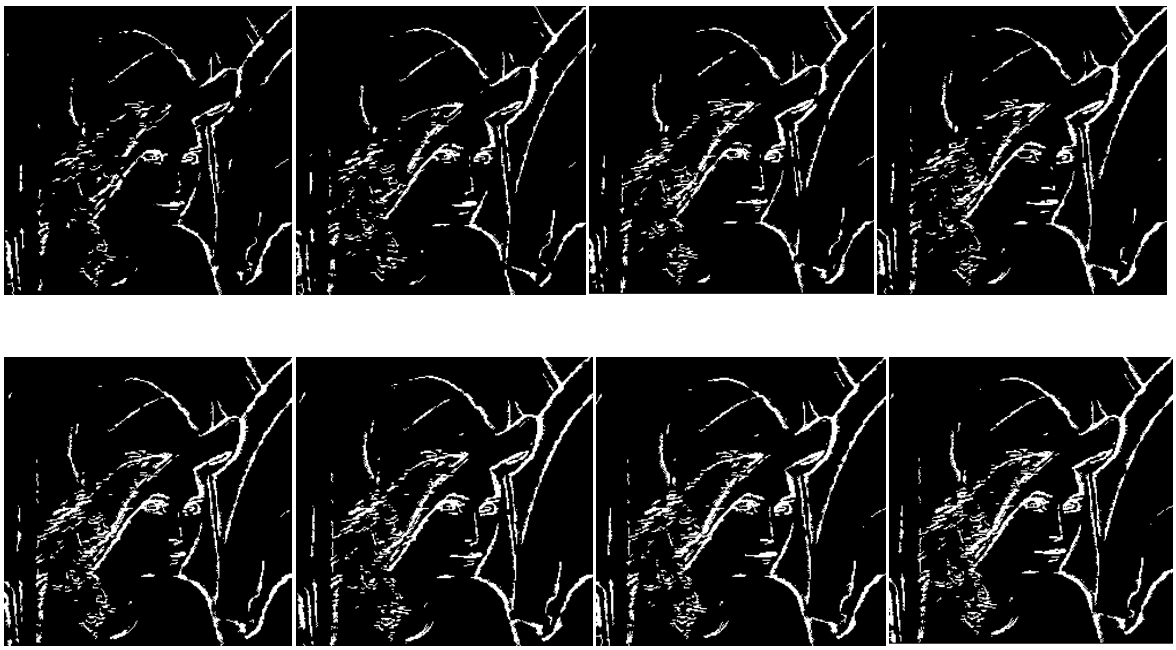


Fig.5.7 Results of proposed approach for different number of chemotactic steps (N_c), (a) N_c = 50, (b) N_c = 60, (c) N_c = 70, (d) N_c = 80, (e) N_c = 90, (f) N_c = 100, (g) Nc = 110, (h) N_c =120

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 entropy function for quantitative analysis. Entropy represents the amount of information present in the image.

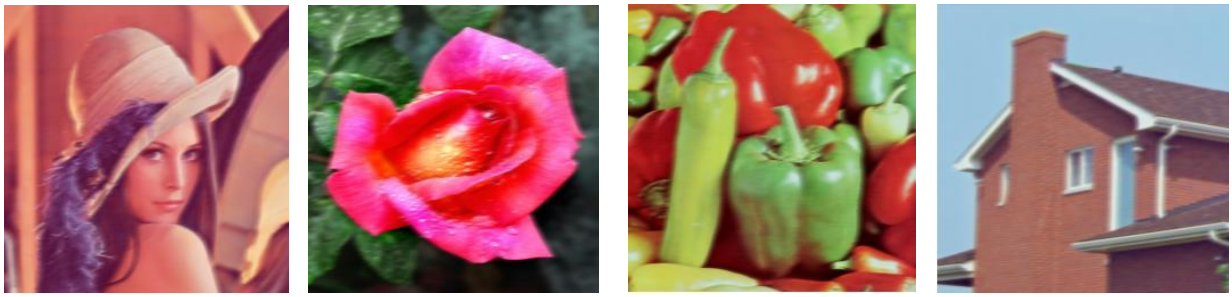
Table 1 represents the values of entropy measure for Majority image and for the results of various approaches: Sobel, Prewitt, Canny and proposed approach. The entropy for results of Sobel and Prewitt is comes out to be smaller than the proposed approach for all the four test images, because they provide less edge information. 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.

Table 1: Entropy

	Sobel	prewitt	canny	Ours
Rose	0.8833	0.8923	1.1188	1.2007
Lena	0.8899	0.8894	1.0648	1.2657
Peppers	0.8715	0.8732	1.0950	1.1671
House	0.8804	0.8791	1.0406	1.111

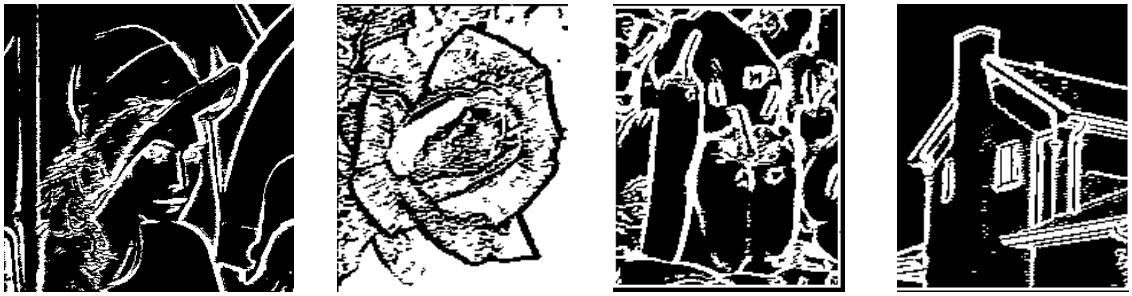
(a) lena (b) rose (c) peppers (d) house

Fig 5.8 Original images.



(a) lena (b) rose (c) peppers (d) house

Fig 5.9 Deblurred images.



(a) lena (b) rose (c) peppers (d) house

Fig 5.10 Result of proposed approach on proposed deblurred images .

(a) lena (b) rose (c) peppers (d) house

Fig 5.11 Result of Canny edge detector on proposed deblurred images.

(a) lena (b) rose (c) peppers (d) house

Fig 5.12 Result of prewitt edge detector on proposed deblurred images .

(a) lena (b) rose (c) peppers (d) house

Fig 5.13 Result of Sobel edge detector on proposed deblurred images .

Chapter 6

Conclusion and Future Work

In this letter in first step we have proposed a novel technique for deblurring the color images using wavelet. Here blind deblurring is performed without calculating the PSF in the frequency domain. Experimental tests showed good results on a variety of color images. The proposed algorithm is applicable to digital cameras that capture multiple images by splitting the exposure time. As a result, the motion blur caused by camera shake can be reduced by merging the successive blurred images.

The method can be extended in other directions: For example:

- Modification of fuzzy membership function can produce better result.
- Different kind of approaches can be used to transfer the image in frequency domain which is more efficient than this.
- Here we have proposed our method for single level decomposition. Research can be directed for multilevel decomposition.

In second step we have proposed a novel technique for edge detection using Bacteria Foraging on color images. Here Intensity Difference value of bacteria around its 8-nighborhood has been calculated for determining edge pixels. In Bacteria Foraging, we used the steps, Defining the Nutrient Function, Chemotaxis and Elimination-Dispersion. Here we have performed edge detection on the deblurred color images and the results allow us to conclude that, the implemented technique presents greater robusteness comparative to standard available techniques.

Here we have taken the Intensity Difference value as a nutrient function but research can also directed with some other nutrient function.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] Kretschmer U., Orozco MS. and Ruiz O.E. "Edge and corner identification for tracking the line of sight",Medical information and learning Environment Vol, 1, No 2, PP 5-23, Sept 2005,.

[2] Sau K.,Pal M. and Karmaka P. "Unique Techniques for edge detection methods In digital Image Processing". IEMCON 2011 organised in collaboration with IEEE on 5th & 6th of Jan,2011.

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

[4] Zhengquan He., Siyal M.Y., "An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis". Pattern Recognition Letters, Elsevier Science Inc. 12, Volume 16 Issue, Dec. 1995 New York, NY, USA

[5] Zhang R., Zhao J., and Li Su., "New edge detection method in image processing", Communications and Information Technology, (ISCIT). PP 445-448, 12 Oct. 2005

[6] Stamatia G., Tania Stathaki. " Edge Detection Using Quantitative Combination of Multiple Operators". Communications and Signal Processing Group, Imperial College London Exhibition Road, SW7 2AZ London, UK.

[7] Hanmandlu M., Kalra R.R., and Madasu V.K., Shantaram Vasikarla. "Area based Novel Approach for Fuzzy Edge Detection". , TENCON, IEEE, Department of Electrical Engineering, Indian Institute PP 1-4, 4-Nov., 2006

[8] Hamid R. Ezhoosh, “Fast Fuzzy Edge Detection”.Pattern Recognition and Machine Intelligence Lab Systems Design Engineering, University of Waterloo Waterloo, Ontario, N2L 3G1, Canada.

[9] Hanmandlu, M.; See, J. and Vasikarla, S.;. “Fast fuzzy edge detector using entropy optimization”. Information Technology Coding and Computing, Proceedings, IEEE. PP-665, April 2004.

[10] Jinbo Wu., Zhouping Yin, and Youlun Xiong, “The Fast Multilevel Fuzzy Edge Detection of Blurry Images” Signal Processing Letter, IEEE, Vol. 14, PP 344-347 2007.

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

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

[13] 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.

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

[15] Verma, O.P.; Hanmandlu, M.; Sultania, A.K.; Dhruv, D.;"A Novel Fuzzy Ant System For Edge Detection**"**, Proceedings of 9th International Conference on Computer and Information Science, Yamagata, , 2010, pp.228 - 233 Japan.

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

[17] Jing Tian, Weiyu Yu, and Shengli Xie，"An ant colony optimization algorithm for image edge detection," *2008 Congress on Evolutionary Computation*, IEEE, pp.751-756, Jun. 2008.

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

[19] A. Jevtic, J. Quintanilla-Dominguea, M. G. Cortina-Januchs and D. Andina, "Edge detection using ant colony search algorithm and multiscale constrast enhancement," *2009 IEEE conf. on Systems, Mans, and Cybernetics*, pp. 2193-2198, Oct. 2009, USA,.

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

[21] Barkhoda, W.; Tab, F.A.; and Shahryari, O.K.," Fuzzy edge detection based on pixel's gradient and standard deviation values", Proceedings of International Multi-conference on Computer Science and Information Technology, IEEE., Mragowo, pp. 7 – 10 2009 .

[22] Yishu Zhai; Xiaoming Liu, "Multiscale Edge Detection Based on Fuzzy C-Means Clustering**",** Proceedings of 1st International Symposium on Systems and Control in Aerospace and Astronautics, Harbin, IEEE , pp. 1201-1204, 2006,.

[23] Hamid R. Tizhoosh, "Fuzzy Image Processing", ww.pami.uwaterloo.ca/tizhoosh/edge.html

[24] Y. P. Guo, H. P. Lee, and C. L. Teo, “Blind restoration of images degraded by space-variant blurs using iterative algorithms for both blur identification and image restoration,” Image Vis. Comput., vol. 15, no. 5, pp. 399–410, 1997.

[25] Y.-L. You and M. Kaveh, “Blind image restoration by anisotropic regularization,” IEEE Trans. Image Process., vol. 8, no. 3, pp. 396–407, Mar. 1999.

[26] M. Blume, D. Zikic, W. Wein, and N. Navab, “A new and general method for blind shift-variant deconvolution of biomedical images,” in Proc. MICCAI (1), 2007, pp. 743–750.

[27] M. Sorel and J. Flusser, “ Space-variant restoration of images degraded by camera motion blur,” IEEE Trans. Image Process., vol. 17, no. 1, pp. 105–116, Jan. 2008.

[28] M. Welk, D. Theis, and J. Weickert, “Variational deblurring of images with uncertain and spatially variant blurs,” in Proc. DAGM Symp., 2005, pp. 485–492.

[29] A. Kubota and K. Aizawa, “ Reconstructing arbitrarily focused images from two differently focused images using linear filters,” IEEE Trans. Image Process., vol. 14, no. 11, pp. 1848–1859, Nov. 2005.

[30] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro, “A variational framework for simultaneous motion estimation and restoration of motion-blurred video,” in Proc. IEEE 11th Int. Conf. Computer Vision, 2007, pp. 1–8.

[31] M. S. C. Almeida and L. B. Almeida, “Blind deblurring of foreground- background mages,” presented at the Int. Conf. Image Processing, Cairo, Egypt, Nov. 2009.

[32] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," ACM Trans. Graph, (TOG), vol. 25, pp. 787–794, 7 2006.

[33] M. Bertero and P. Boccacci, Introduction to Inverse Problems in Imaging. Bristol, U.K.: IOP, 1998.

[34] M. S. C. Almeida and L. B. Almeida, "Blind deblurring of natural images," in Proc. ICASSP, Las Vegas, NV, Mar. 2008, pp.1261–1264.

[35] M. Bertero and P. Boccacci, " Image restoration methods for the large binocular telescope," Astron. Astrophys. Suppl., vol. 147, pp. 323–333, 2000.

[36] J. P. Muller, Digital Image Processing in Remote Sensing. New York: Taylor & Francis, 1988.

[37] S. Jefferies, K. Schulze, C. Matson, K. Stoltenberg, and E. K. Hege, "Blind deconvolution in optical diffusion tomography," *Opt. Exp.*, vol. 10, pp. 46–53, 2002.

[38] A. M. Bronstein, M. M Bronstein, M. Zibulevsky, and Y. Y. Zeevi, "Quasi-maximum likelihood blind deconvolution of images acquired through scattering media," in *Proc. ISBI*, Apr. 2004, pp. 352–355,.

[39] J. Markham and J. Conchello, "Parametric blind deconvolution: A robust method for the simultaneous estimation of image and blur," J. Opt. Soc. Amer. A, pp. 2377–2391,1999.

[40] D. Adam and O. Michailovich, "Blind deconvolution of ultrasound sequences using non-parametric local polynomial estimates of the pulse," IEEE Trans. Biomed. Eng., vol. 42, no. 2, pp. 118–131, Feb. 2002.

[41] P. Pankajakshan, B. Zhang, L. Blanc-Feraud, Z. Kam, J.-C. Olivo- Marin, and J. Zerubia, "Blind deconvolution for diffraction-limited fluorescence microscopy," presented at the ISBI, 2008.

[42] G. K. Chantas, N. P. Galatsanos, and A. C. Likas, "Bayesian restoration using a new nonstationary edge-preserving image prior," IEEE Trans. Image Process., vol. 15, no. 10, pp. 2987–2997, Oct. 2006.

[43] J. A. Guerrero-Colon and J. Portilla, "Deblurring-by-denoising using spatially adaptive gaussian scale mixtures in overcomplete pyramids," in Proc. IEEE Int. Conf. Image Processing, , Oct. 2006, pp. 625–628, Atlanta, GA.

[44] L. Bar, N. Kiryati, and N. Sochen, "Image deblurring in the presence of impulsive noise," Int. J. Comput. Vis., vol. 70, no. 3, pp. 279–298, Mar. 2006.

[45] V. Katkovnik, K. Egiazariana, and J. Astola, "A spatially adaptive nonparametric regression image deblurring," IEEE Trans. Image Process., vol. 14, no. 10, pp. 1469–1478, Oct. 2005.

[46] G. Chantas, N. Galatsanos, A. Likas, and M. Saunders, "Variational bayesian image restoration based on a product of t-distributions image prior," IEEE Trans. Image Process., vol. 17, no. 10, pp. 1795–805, Oct. 2008.

[47] D. Kundur and D. Hatzinakos, "Blind image deconvolution," IEEE Sig. Process. Mag., pp. 43–64, May 1996.

[48] P. Campisi and K. Egiazarian, Blind Image Deconvolution: Theory and Applications. Boca Raton, FL: CRC, 2007.

[49] H. Yin and I. Hussain, “Blind source separation and genetic algorithm for image restoration,” presented at the ICAST, , Sep. 2006, Islamabad, Pakistan.

[50] F. Krahmer, Y. Lin, B. McAdoo, K. Ott, J. Wang, D. Widemannk, and B. Wohlberg, Blind Image Deconvolution: Motion Blur Estimation, Tech Rep., Univ. Minnesota, 2006.

[51] J. Oliveira, M. Figueiredo, and J. Bioucas-Dias, “Blind estimation of motion blur parameters for image deconvolution,” presented at the Iberian Conf. Pattern Recognition and Image Analysis, Girona, Spain, Jun. 2007.

[52] S. Chang, S. W. Yun, and P. Park, “PSF search algorithm for dual exposure type blurred,” Int. J. Appl. Sci., Eng., Technol., vol. 4, no. 1, pp. 1307–4318, 2007.

[53] S. D. Babacan, R. Molina, and A. K. Katsaggelos, “Variational bayesian blind deconvolution using a total variation prior,” IEEE Trans. Image Process., vol. 18, no. 1, pp. 12–26, Jan. 2009.

[53] K.-H. Yap, L. Guan, and W. Liu,“A recursive soft-decision approach to blind image deconvolution,” IEEE Trans. Signal Process., vol. 51, no. 2, Feb. 2003.

[54] L. Justen and R. Ramlau, “A non-iterative regularization approach to blind deconvolution ” Inst. Phys. Pub. Inv. Probl., vol. 22, pp. 771–800, IOP Science, 2006.

[55] R. Kaftory and N. Sochen, “Variational blind deconvolution of multichannel images,” Int. J. Imag. Syst. Technol ., vol. 15, no. 1, pp. 55–63, 2005.

[56] J. Jia, J. Sun,C.-K. Tang and H.-Y. Shum, “Bayesian correction of image intensity with spatial consideration,” ECCV 2004, Lecture Notes in Computer Science, vol. 3023, pp. 342–354, May 2004.

[57] M. Ben-Ezra and S. K. Nayar, “Motion-based motion deblurring,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 6, pp. 689–698,Jun. 2004.

[58] S. Oh and R. J. Marks, “Alternating projection onto fuzzy convex sets,” in Proc. IEEE Int. Conf. Fuzzy Systems, 1993, vol. 1, pp. 1148–1155.

[59] Passino, K.M., Biomimicry of bacterial foraging for distributed optimization and control, Control Systems Magazine, IEEE, Jun 2002, Volume: 22, Issue: 3 pp. 52-67

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

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

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

[63] Coifman R. R., Meyer Y., and Wickerhauser M. V., “Size properties of waveletpackets. In Wavelets and their applications”, pages 453-470. Jones and Bartlett, Boston, MA, 1992.

[64] Chambolle A., DeVore R. A., Lee N. Y. and Lucier B. J., “Nonlinear Wavelet Image Processing: Variational Problems, Compression, and Noise Removal through Wavelet Shrinkage”. IEEE Trans. Image Processing 7 (1998), 319-335.

[64] Yishu Zhai; Xiaoming Liu, “Multiscale Edge Detection Based on Fuzzy C-Means Clustering”, Proceedings of 1st International Symposium on Systems and Control in Aerospace and Astronautics, Harbin, 2006, pp. 1201-1204.