A
Dissertation
On

AN IMPULSE NOISE FILTER

Submitted in Partial fulfillment of the requirement
For the award of Degree of

MASTER OF TECHNOLOGY
Computer Technology and Application (CTA)

Submitted By:

SHWETA
23/CTA/2010

Under the Guidance of:

Dr. O.P. VERMA

Head of Department
Department of Information Technology
Delhi Technological University

DEPARTMENT OF COMPUTER ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(*Formerly Delhi College of Engineering)*
Bawana Road, Delhi-110042

2010-2012

CERTIFICATE

This is to certify that the work contained in this dissertation entitled "**An Impulse Noise Filter**" submitted in the partial fulfillment, for the award for the degree of M.Tech in Computer Technology and Applications at **Delhi Technological University, Delhi** by **SHWETA, Roll No. 23/CTA/2010** is carried out by her under my supervision. This matter embodied in this project work has not been submitted earlier for the award of any degree or diploma in any university/institution to the best of our knowledge and belief.

Dr. O.P.Verma
Head of Department
Department of Information Technology
Delhi Technological University
Delhi

ACKNOWLEDGMENT

I take the opportunity to express my sincere gratitude to my project mentor Dr.O.P.Verma, Head of Department, Department of Information Technology, Delhi Technological University, Delhi for providing valuable guidance and constant encouragement throughout the project.

I also want to express my sincere thanks to our HOD, Dr. Daya Gupta, Department of Computer Engineering, Delhi Technological University, Delhi for providing well equipped infrastructure support.

SHWETA
23/CTA/2010
M.Tech. (Computer Technology and Applications)
Department of Computer Engineering
Delhi Technological University

ABSTRACT

Present day applications require various kinds of images and pictures as sources of information for interpretation and analysis. Whenever an image is converted from one form to another such as, digitizing, scanning, transmitting, storing, etc., some of the degradation occurs at the output. Hence, the output image has to undergo a process called image enhancement which consists of a collection of techniques that seek to improve the visual appearance of an image. Image enhancement is basically improving the interpretability or perception of information in images for human viewers and providing 'better' input for other automated image processing techniques. One such degradation of image is the addition of ***impulse noise*** in an image. One of the widely known forms of impulse noise is ***"salt and pepper"*** noise. Various filtering techniques have been proposed for removing impulse noise in the past, and it is well-known that linear filters could produce serious image blurring. As a result, nonlinear filters have been widely exploited due to their much improved filtering performance, in terms of impulse noise attenuation and edge/details preservation. This dissertation presents a novel and efficient approach to impulse noise detection and filtering. The proposed method applies the adaptive fuzzy rule-based technique to noise detection after selecting the decision boundaries discriminatively and assigning the pixels their appropriate class. Thus the decision map formed is given to filtering process. This new technique can remove the impulse noise (represented using four noise models) from corrupted images efficiently and requires no previous training. Quantitative and qualitative analysis, performed on standard color and gray scale images, shows improved performance of proposed technique over existing state-of-art algorithms. The image is corrupted up to 80% noise density under each noise model and PSNR, MSE, correlation coefficient and delta E color difference are used to compare the results of proposed approach with existing algorithms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

(i, j):	Spatial coordinates
p:	Probability of noise in an image
x:	Original image
y:	Corrupted version of original image
log:	Logarithm operator
r:	Coefficient of Correlation
p_1, p_2:	Probability of salt and pepper or high and low intensity noise ($p=p_1+p_2$)
m:	Range to be specified in noise model for random noise
b_1, b_2:	Decision Boundaries
v_o:	Sorted array of pixels in a window
v_D:	Difference vector
med:	Median of the window
$\mu_{L:}$	Membership function representing fuzzy set "Low"
$\mu_{M:}$	Membership function representing fuzzy set "Medium"
$\mu_{H:}$	Membership function representing fuzzy set "High"
L:	Fuzzy set Low
M:	Fuzzy set Medium
H:	Fuzzy set High
N:	Degree of noisyness
z:	Total number of fuzzy rules (27)
$\mu_{prem}(i)$:	Certainty of premise for i^{th} rule
W:	Window Size
$F(i,j,z)$:	Filtered Image
d_{rg} , d_{rb} , d_{gb} :	Represents differences between red-green, red-blue and green-blue components of a pixel
μ_l:	Membership function "large"
β_1, β_2 :	Parameters for membership function large

n_d:	Degree of noise
$\Delta(i,j,z)$:	Correction term for color images
$O(i,j,z)$:	Output color image
RVIN:	Random Valued Impulsive Noise
SPN:	Fixed Valued Impulsive Noise or Salt & Pepper Noise
PSNR:	Peak Signal to Noise Ratio
MSE:	Mean Sqaure Error
ΔE:	Delta E, represents color difference in CIELAB color space (average value of Delta E is considered in the thesis)
BDND:	Boundary Discriminative Noise Detection [18]
SMF:	Switching Median Filter [24]
ASWM:	Adaptive Switching Median Filter [23]
DBAIN:	Decision Based Algorithm for Impulse Noise [19]
PA:	Proposed Approach

Chapter 1
INTRODUCTION

1.1 Introduction

A major portion of information received by a human from the environment is visual. Hence, processing visual information by computer has been drawing a very significant attention of the researchers over the last few decades. The process of receiving and analyzing visual information by the human species is referred to as sight, perception or understanding. Similarly, the process of receiving and analyzing visual information by digital computer is called *digital image processing*.

An image may be described as a two-dimensional function *I* [2].

$$I = f(i, j) \tag{1.1}$$

where *i* and *j* are spatial coordinates. Amplitude of *f* at any pair of coordinates (*i,j*) is called intensity *I* or gray value of the image. When spatial coordinates and amplitude values are all finite, discrete quantities, the image is called digital image.

Digital image processing may be classified into various sub branches based on methods whose:

- input and output are images and
- inputs may be images where as outputs are attributes extracted from those images.

Following is the list of different image processing functions based on the above two classes.

- Image Acquisition
- Image Enhancement
- Image Restoration
- Color Image Processing
- Multi-resolution Processing

- Compression
- Morphological Processing
- Segmentation
- Representation and Description
- Object Recognition

For the first seven functions the inputs and outputs are images where as for the rest three the outputs are attributes from the input images. With the exception of image acquisition and display most image processing functions are implemented in software. Image processing is characterized by specific solutions; hence the technique that works well in one area can be inadequate in another. The actual solution of a specific problem still requires a significant research and development. Out of the ten sub-branches of digital image processing, cited above, the thesis deals with image restoration. To be precise, the projects devotes on a part of the image restoration i.e. impulse noise removal from images (for both color and gray scale), stated in the Problem Definition.

Image restoration emphasizes on getting back the original image as far as possible from the degraded one. Thus the goal of image enhancement is very different from that of restoration. Digital image restoration is a field of engineering that studies methods used to recover an original scene from degraded observations. Developing techniques to perform the image restoration task requires the use of models not only for the degradations, but also for the images themselves. Image restoration problem is a subset of Inverse Problem. In general, in inverse problems, the values of a certain set of functions are estimated from the known properties of other functions.

Consider the following relationship

$$L(\{f_i\}, \{g_j\}) = 0 \tag{1.2}$$

where L is an operator, the function, $\{f_i\}$, are sought, and the values of the functions, $\{g_j\}$, are known. When the problem is well poised, the existence of solution is assured. Also there exists a unique solution for a given problem. However, in the presence of noise, the uniqueness of solution is not assured. The image degradation and subsequent restoration may be depicted as in Figure 1.1(a). In

this project, however, only noise part of entire degradation is dealt with, which is shown in Figure 1.1(b).

This is chapter is organized as follows. The problem definition is described in Section 1.2. Various performance metrics used to examine the algorithms in the thesis is present in Section 1.3. Motivation behind carrying out the work is stated in Section 1.4. Literature survey with broad classification of filters is discussed in Section 1.5. Organization of the thesis is outlined in Section 1.6.

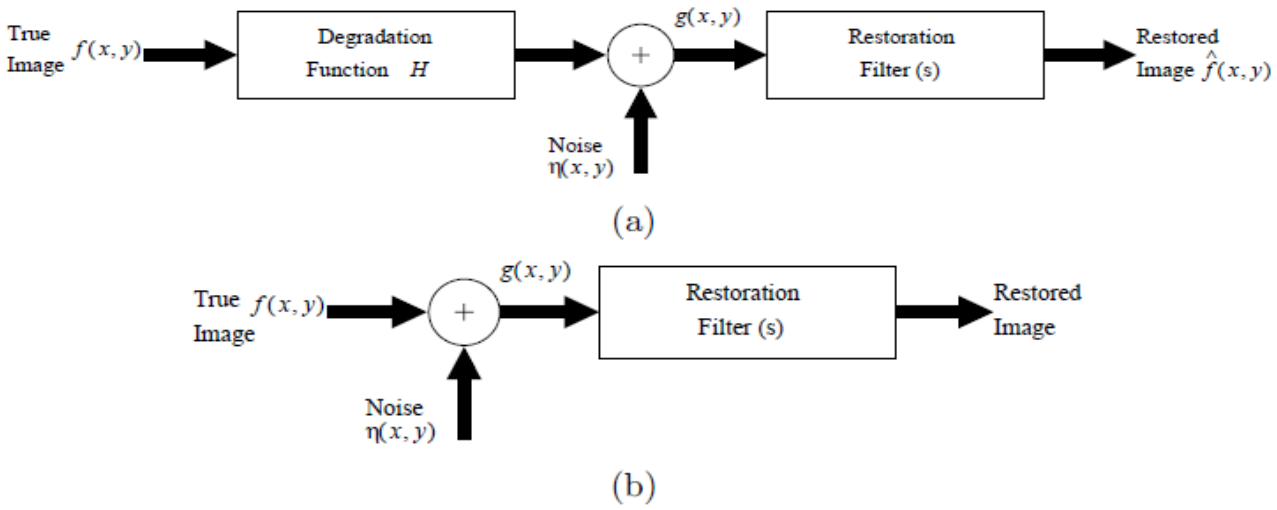


Figure 1.1: (a) Model of the image degradation/restoration process, (b) Model of the Noise Removal Process.

1.2 Problem Definition

Different types of noise frequently contaminate images. Impulsive noise is one such noise, which may affect images at the time of acquisition due to noisy sensors or at the time of transmission due to channel errors or in storage media due to faulty hardware. Two types of impulsive noise models are described below. Let $x_{i,j}$ be the gray level of an original image x at pixel location (i, j) and $[n_{min}, n_{max}]$ be the dynamic range of x. Let $y_{i,j}$ be the gray level of the noisy image y at pixel (i, j) location. *Impulsive Noise* may then be defined as:

$$y_{i,j} = \begin{cases} x & with\ 1-p \\ r_{i,j} & with\ p \end{cases} \tag{1.3}$$

where, $r_{i,j}$ is the substitute for the original gray scale value at the pixel location (i, j). When $r_{i,j} \in [n_{min}, n_{max}]$, the image is said to be corrupted with *Random Valued Impulsive Noise* (RVIN) and when $r_{i,j} \in \{ n_{min}, n_{max} \}$, it known as *Fixed Valued Impulsive Noise* or *Salt & Pepper Noise* (SPN). Pixels replaced with RVIN and their surroundings exhibit very similar behavior. These pixels differ less in intensity, making identification of noise in RVIN case far more difficult than in SPN. The difference between SPN and RVIN may be best described by Figure 1.2. In the case of SPN the pixel substitute in the form of noise may be either $n_{min}(0)$ or $n_{max}(255)$. Where in RVIN situation, it may range from n_{min} to n_{max}.

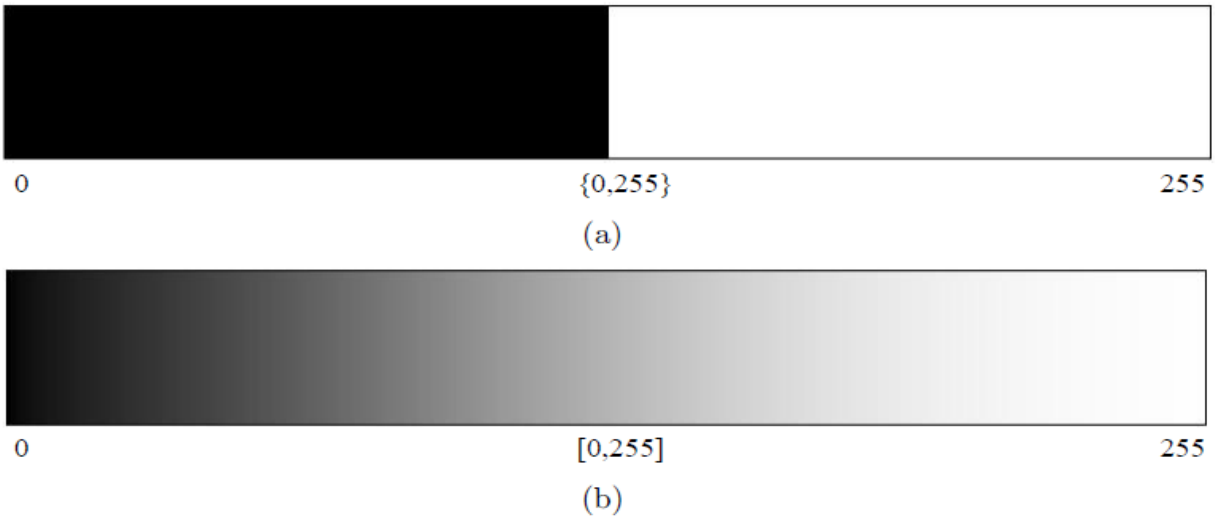


Figure 1.2: Representation of (a) *Salt & Pepper Noise* with $Ri,j \in \{n_{min}, n_{max}\}$, (b) *Random Valued Impulsive Noise* with $Ri,j \in [n_{min}, n_{max}]$

1.3 Performance Measures

The metrics used for performance comparison of different filters (exists and proposed) are defined below:

a.) **Peak Signal to Noise Ratio (*PSNR*)**: *PSNR* analysis uses a standard mathematical model to measure an objective difference between two images. It estimates the quality of a reconstructed image with respect to an original image. The basic idea is to compute a single number that reflects the quality of the reconstructed image. Reconstructed images with higher PSNR are judged better. The *PSNR*(*dB*) is defined as:

$$\text{PSNR} = 10 \log_{10}\left(\frac{255^2}{\text{MSE}}\right) \text{dB} \tag{1.4}$$

b.) **Mean Square Error (MSE)**: Mean square error is given by

$$\text{MSE} = \frac{1}{\text{MN}} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(x_{i,j} - y_{i,j}\right)^2 \tag{1.5}$$

where *x* is an original image and *y* is a corrupted version of it.

b.) **Coefficient of Correlation (r)**: The quantity *r*, called the *linear correlation coefficient*, measures the strength and the direction of a linear relationship between two variables.

$$r = \frac{\sum\sum(x_{i,j} - \bar{x})(y_{i,j} - \bar{y})}{\sqrt{(\sum\sum(x_{i,j} - \bar{x})^2)(\sum\sum(y_{i,j} - \bar{y})^2)}} \tag{1.6}$$

with respect to noise free original image *x*.

d.) **Delta E (ΔE):**

PSNR is just a numerical value more applicable to the monochrome images. To have human perception of color into consideration we need a metric to measure color difference. In response to the same, the perceptually uniform color space CIELAB, standardized by the Commission Internationale de l'Eclairage (CIE) [51], is more accurate for defining quantitative measurements of perceptual error between the two color vectors. In CIELAB color space the color difference is calculated in terms of ΔE (Delta E). It is given by [52]:

$$\Delta\text{E} = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2} \tag{1.7}$$

where (L_1^*, a_1^*, b_1^*) and (L_2^*, a_2^*, b_2^*) are lab transform of the RGB image. Here we will consider the average value of ΔE for complete image to compare color images.

1.4 Literature Survey

Image restoration employs different filtering techniques. Filtering may be done either in *spatial domain* or in *frequency domain*. In this thesis different spatial domain restoration techniques are

studied and proposed. Broadly, filters may be classified into two categories: *Linear* and *Nonlinear*. The filtering methodologies are described below.

1.4.1 Linear Filters

In the early development of image processing, linear filters were the primary tools. Their mathematical simplicity with satisfactory performance in many applications made them easy to design and implement. However, in the presence of noise the performance of linear filters is poor. They tend to blur edges, do not remove impulsive noise effectively, and do not perform well in the presence of signal dependent noise [1].

Mathematically, a filter may be defined as an operator $L(\cdot)$, which maps a signal s into a signal z:

$$z = L(\mathrm{s}) \tag{1.8}$$

When the operator $L(\cdot)$ satisfies both the superposition and proportionality principles, the filter is said to be linear. Two-dimensional and m-dimensional linear filtering is concerned with the extension of one-dimensional filtering techniques to two and more dimensions. If impulse response of a filter has only finite number of non-zero values, the filter is called a *finite impulse response* (FIR) filter. Otherwise, it is an *infinite impulse response* (IIR) filter. If the filter evaluates the output image only with the input image, the filter is called *non-recursive*. On the other hand, if the evaluation process requires input image samples together with output image samples, it is called *recursive* filter [1, 2]. Following are the few main types of filters:

- *Low-pass filter*: Smooth the image, reducing high spatial frequency noise components.
- *High-pass filter*: Enhances very low contrast features, when superimposed on a very dark or very light background.
- *Band-pass filter*: Tends to sharpen the edges and enhance the small details of the image.

1.4.2 Nonlinear Filters

Nonlinear filters also follow the same mathematical formulation as linear filters. However, the operator $L(\cdot)$ is not linear in this case. Convolution of the input with its impulse response does not

generate the output of a nonlinear filter. This is because of the non-satisfaction of the superposition or proportionality principles or both.

Gray scale transformations are the simplest possible nonlinear transformations. This corresponds to a memory-less nonlinearity that maps the signal x to y. The transformation $b = t(a)$ may be used to transform one gray scale a to another b [3]. *Histogram modification* is another form of intensity mapping where the relative frequency of gray level occurrence in the image is depicted. An image may be given a specified histogram by transforming the gray level of the image into another. *Histogram equalization* is one such method that is used for this purpose. The need for it arises when comparing two images taken under different lighting conditions. The two images must be referred to the same base, if meaningful comparisons are to be made. The base that is used as standard has a uniformly distributed histogram. Of course, a uniform histogram signifies maximum information content of the image. Order statistic filters for noise removal are the most popular class nonlinear filters. A number of filters belong to this class of filters, e.g., the median filter, the stack filter, the median hybrid filter etc. These filters have found numerous applications in digital image processing.

One of the most popular and robust nonlinear filters is the *standard median* (SM) filter [4], which exploits the rank-order information of pixel intensities within a filtering window and replaces the center pixel with the median value. Due to its effectiveness in noise suppression and simplicity in implementation, various modifications of the SM filter have been introduced. However, the median filter tends to blur image details and remove thin lines even at low noise densities. To avoid the inherent drawbacks of the standard median filter, the weighted median filter [5] and the center-weighted median filter [6], which are modified median filters, have been introduced. These filters demonstrate better performance in preserving image. However, applying these filters unconditionally across the entire image without considering whether it is uncorrupted or corrupted as practiced in the conventional schemes would inevitably remove the uncorrupted detail pixels, destroy the image quality, and cause additional blur. Efforts are made [7, 8] to reduce blurring at the edges due to linear filtering. An intuitive solution to overcome this problem is to implement an impulse-noise detection mechanism prior to filtering; hence, only those pixels identified as “corrupted” would undergo the filtering process, while those identified as “uncorrupted” would remain intact. By incorporating such noise detection mechanism or “intelligence” into the median filtering framework, the so-called

switching median filters [9]–[11] had shown significant performance improvement. Median based filters are modified for detail-preservation images in [12]. Wenbin [13] presented a novel idea of alpha trimmed mean and the similarity of pixels for the detection of impulse noise. Early-developed switching median filters are commonly found being non-adaptive to a given, but unknown, noise density and prone to yielding pixel misclassifications especially at higher noise density interference. A signal adaptive median filtering algorithm is proposed [14] for the removal of impulse noise in which the notion of homogeneity level is defined for pixel values based on their global and local statistical properties. The co-occurrence matrices are used to represent the correlations between a pixel and its neighbors, and to derive the upper and lower bounds of the homogeneity level. The use of non linear filters for both noise correction and image preservation is also suggested by Russo [15, 16]. Noisy pixels have been categorized into edge and non edge pixels and different filtering schemes are applied. The *noise adaptive soft-switching median* (NASM) filter [11] was proposed, which consists of a three-level hierarchical soft-switching noise detection process. The NASM achieves a fairly robust performance in removing impulse noise, while preserving signal details across a wide range of noise densities, ranging from 10% to 50%. However, for those corrupted images with noise density greater than 50%, the quality of the recovered images become significantly degraded, due to the sharply increased number of misclassified pixels. Mansoor et al. [17] introduce an iterative edge preserving filtering technique using the blur metric. The Boundary Discriminative Noise Detection (BDND) [18] is an algorithm is proposed for the detection of impulse noise based on a large difference between the noisy pixel and the noise free pixel. The paper claims to achieve better results than NASM by passing the pixel from two level of window to confirm whether it is noisy or not. If noisy, adaptive filtering, achieved by some modifications in NASM, is applied. Srinivasn et al. [19] presents a new decision based algorithm (DBAIN) for restoration of images that are highly corrupted by impulse noise. It removes only corrupted pixel by the median value or by its neighboring pixel value. A more improved decision based algorithm for impulse noise removal algorithm is proposed by Nair et al.[20, 21]. This algorithm utilizes previously processed neighboring pixel values to get better image quality than the one utilizing only the just previously processed pixel value. To further improve the performance of high-density salt and pepper noise denoising, a new detail preserving median filter algorithm is proposed [22] by Wei Li et al. The method replaces only corrupted pixels by either the trimmed median or the average of previously processed neighborhood pixels, while uncorrupted pixels remain unchanged. Smail Akkoul et al. presented an adaptive switching median filter (ASWM)

[23] which requires no prior threshold as required by classical switching median filter. Threshold is computed locally from image pixels intensity values in a sliding window. Duan and Zhang presented a two iteration algorithm [24] for impulse noise detection for switching median filter. Very recently, Tripathi et al. in [25] presents a switching median filter which is an advanced boundary discriminative noise detection algorithm.

Image filters works in three domains namely spatial, frequency and fuzzy domains. Fuzzy filters offer several advantages over classical filters even as they preserve the image structure. They provide more advantage as they are realized very easily by means of simple if-then rules called fuzzy rules that characterize a particular noise. Several non-linear filters based on classical and fuzzy techniques have emerged in the past few years. Recent progress in fuzzy logic allows different possibilities for developing new image noise reduction methods. The fuzzy median filter [26, 27] is a modification to the classical median filter. The Fuzzy Inference Rules by Else action (FIRE) filters [28, 29, 30] are a family of non-linear operators that adopt fuzzy rules to remove impulse noise from images**.** Russo introduced a multi-pass fuzzy filter consisting of three cascaded blocks [30]. Each block is hooked to a fuzzy operator that attempts to cancel the noise while preserving the image structure. The fuzzy multilevel median filter introduced by Jiu [31] is manifestation of the multilevel median filter in the fuzzy domain. It includes fuzzy rules for the elimination of impulse noise. The histogram adaptive filter by Wang and Chu [32] belongs to a class of filters which employs the histogram for reducing noise. Androutsos et al. [33] designed a new class of filters called Fuzzy vector rank filters based on a combination of different distance measures, fuzzy membership values and α -trimmed functions. Khriji and Gabbouj [34] developed a multi channel filter by combining fuzzy rational and median functions. This filter preserves the edges and chromaticity of the image. Stefan *et al*. [35] presented a fuzzy two-step color filter for the reduction of impulse noise. This filter utilizes the fuzzy gradient values and fuzzy reasoning for the detection of noisy pixels. Li Song et al. [36] present a new framework of removing impulse noise. The most important point is that the types of images are estimated by using the FINDRM and the efficient detail preserving approach (EDPA). When it is estimated that an image has many white and black pixels, the detected noise pixels from the FINDRM are re-checked by using alpha-trimmed means. Oppositely, when it is estimated that an image has a few white and black pixels, the detection results from the FINDRM are used directly. The paper [37] proposed a mixed impulse fuzzy filter based on medina (MAD), Rank-ordered absolute differences

(ROAD) and Genetic Algorithms. It consists of three components, including fuzzy noise detection system, fuzzy switching scheme filtering, and fuzzy parameters optimization using genetic algorithms (GA) to perform efficient and effective noise removal. MAD and ROAD are used as measures of noise probability of a pixel. Fuzzy reference system is used to justify the degree of which a pixel can be categorized as noisy and further fuzzy switching scheme that adopts median filter as the main estimator is applied to the filtering. Another rule based optimization of fuzzy inference system is presented [38], where Soyturk et al. used only the parameters processed in the current epoch, rather than all parameters of the fuzzy inference system are replaced with the new candidate solutions when examining the neighboring solutions in the search space. Toh at al. [39] presented a cluster based adaptive fuzzy switching median filter for universal impulse noise reduction i.e. random valued or fixed valued impulse noise. Very recently also many research were carried out in the field of noise reduction using fuzzy technique. Meher [40] presents a model that extracts a set of informative features, uses a fuzzy detector based on product aggregation reasoning rule for noisy pixels detection and noise removal operator for filtration. Hussain et al. presents a noise filter using fuzzy logic and alpha-trimmed mean [41] to avoid the outlier effect. Alpha-trimmed mean and median values play an important role to formulate the fuzzy membership function. Madhu et al. [42] proposes a new efficient fuzzy-based decision algorithm (FBDA) for the restoration of images that are corrupted with high density of impulse noises. FBDA is a fuzzy-based switching median filter in which the filtering is applied only to corrupted pixels in the image while the uncorrupted pixels are left unchanged. The proposed algorithm computes the difference measure for each pixel based on the central pixel (corrupted pixel) in a selected window and then calculates the membership value for each pixel based on the highest difference. The algorithm then eliminates those pixels from the window with very high and very low membership values, which might represent the impulse noises. Median filter is then applied to the remaining pixels in the window to get the restored value for the current pixel position. Further Muthukumar in [43] proposed an Improved Fuzzy-Based Decision Algorithm (IFBDA) for restoration of images. Another two phase process of fuzzy logic based impulse noise filtering technique is presented by Aborisade in [44]. Verma et al [45] presents a two stage fuzzy filter for reduction of both impulse and Gaussian noise. It also considers the interactions between the color components. [46] and [47] are further areas where fuzzy rule based system is applied in digital image processing where fuzzy rule based approach is used for edge detection and segmentation respectively. These papers are useful to understand the underlying technique presented in this thesis.

1.5 Motivation

The last section reveals many things about the existing restoration schemes. Most of the reported schemes work well under SPN but fails under RVIN, which is more realistic when it comes to real world applications. Even though some of the reported methods claim to be adaptive, they are not truly adaptive for the simple reason of not considering the image and noise characteristics. These schemes generally use a threshold value for the identification of noise. A predefined parameter is compared with this threshold value. If it exceeds, the pixel is marked as contaminated otherwise not. Usually the threshold value used is either a constant or a set of four/five values. A threshold, which is optimal in one environment, may not be good at all in a different environment. By environment we mean, the type of image, characteristic and density of noise. To incorporate the "real" nature of image, fuzzy techniques can proved to be beneficial in detection process as they consider the overlapping boundaries which are generally neglected in non-fuzzy techniques.

One of the important techniques pointed out in literature survey is Boundary Discriminative Noise Detection [18]. This filter claims to shown superior performance in terms of subjective quality in the filtered image as well as objective quality in the *peak signal-to-noise ratio* (PSNR) measurement. This report as a part of major thesis will be discussing about this filter in detail and proposed our new approach of detection which is providing improved performance as compared to BDND. The proposed algorithm gives an impulse filter for both gray scale and color images.

1.6 Report Organization

Our report is divided into following sections:

Chapter 2 introduces the boundary discriminative noise detection.

Chapter 3 discusses Fuzzy Image Processing.

Chapter 4 presents a proposed algorithm "Fuzzy Based Boundary Discriminative Noise Detection".

Chapter 5 shows simulation results and comparison.

Chapter 6 concludes the project report and discusses the future work.

References are given to include all the sources that had been referred to.

Chapter 2

NOISE DETECTION BY BOUNDARY DISCRIMINATION

2.1 Noise Models

The boundary discriminative noise detection algorithm [18] mentioned and discusses four types of impulse noise models for their simulation. These noise models are implemented, for extensively examining the performance of our proposed filter with boundary discriminative noise detection algorithm. Each model is described in detail as follows:

2.1.1 Noise Model 1

Noise is modeled as salt-and-pepper impulse noise as practiced. Pixels are randomly corrupted by two fixed extreme values, 0 and 255 (for 8-bit monochrome image), generated with the same probability. That is, for each image pixel at location *(i, j)* with intensity value *x(i, j)*, the corresponding pixel of the noisy image will be *y(i, j)*, in which the probability density function of is:

$$f(y) = \begin{cases} \frac{p}{2} & for\ y = 0 \\ 1 - p & for\ y = x(i,j) \\ \frac{p}{2} & for\ y = 255 \end{cases} \quad (2.1)$$

where *p* is the noise density.

2.1.2 Noise Model 2

For the Model 2, it is similar to Model 1, except that each pixel might be corrupted by either “pepper” noise (i.e., 0) or “salt” noise with *unequal* probabilities. That is:

$$f(y) = \begin{cases} \frac{p1}{2} & for\ y = 0 \\ 1 - p & for\ y = x(i,j) \\ \frac{p2}{2} & for\ y = 255 \end{cases} \quad (2.2)$$

where $p=p1+p2$ is the noise density and $p1 \neq p2$.

2.1.3 Noise Model 3

Instead of two fixed *values*, impulse noise could be more realistically modeled by two fixed *ranges* that appear at both ends with a length of each, respectively. For example, if is 10, noise will equal likely be any values in the range of either [0,9] or [246,255]. That is

$$f(y) = \begin{cases} \frac{p}{2m} & for\ 0 \leq y < m \\ 1 - p & for\ y = x(i,j) \\ \frac{p}{2m} & for\ 255 - m < y \leq 255 \end{cases} \quad (2.3)$$

where p is the noise density and m is the range.

2.1.4 Noise Model 4

Model 4 is similar to Model 3, except that the densities of low-intensity impulse noise and high-intensity impulse noise are *unequal*. That is

$$f(y) = \begin{cases} \frac{p1}{2m} & for\ 0 \leq y < m \\ 1 - p & for\ y = x(i,j) \\ \frac{p2}{2m} & for\ 255 - m < y \leq 255 \end{cases} \quad (2.4)$$

where $p=p1+p2$ is the noise density and $p1 \neq p2$.

2.2 Noise Detection Algorithm

The BDND algorithm [18] is applied to each pixel of the noisy image in order to identify whether it is "uncorrupted" or "corrupted." After such an application to the entire image, a two-dimensional binary decision map is formed at the end of the noise detection stage, with "0s" indicating the positions of "uncorrupted" pixels, and "1s" for those "corrupted" ones. To accomplish this objective, all the pixels

within a pre-defined window that center around the considered pixel will be grouped into three clusters; hence, two boundaries b_1 and b_2 are required to be determined. For each pixel $y(i,j)$ being considered, if $0 \leq y(i,j) \leq b_1$ the pixel will be assigned to the lower-intensity cluster; otherwise, to the medium-intensity cluster for $b_1 < y(i,j) \leq b_2$ or to the high-intensity cluster for $b_2 < y(i,j) \leq 255$.

If the center pixel being falls unto middle cluster, it will be treated as "uncorrupted," since its intensity value is neither relatively low nor relatively high. Otherwise, it is very likely that the pixel has been corrupted by impulse noise. Clearly, the accuracy of clustering results (hence, the accuracy of noise detection) ultimately depends on how accurate the identified boundaries b_1 and b_2 are.

First, the intuition that leads to the development of the proposed BDND algorithm simply based on the histogram distribution of a 21 x 21 sub-image extracted from the simulated noisy image "Lena" corrupted by 80% impulse noise density based on the above-mentioned Noise Model 3. For illustrating a "typical" histogram distribution, the sub-image chosen bears a "neutral" image content, meaning that the content is neither too "flat" (containing low frequency) nor too "busy" (containing high frequency). It could be observed that the distribution presented at the two ends of the distribution is most likely contributed by impulse noise. Furthermore, the locations of two distinct gaps (or valleys) mark the most possible positions of the two boundaries, respectively, that clearly separate the impulse noise regions (at the two ends) from the uncorrupted pixel region (a much wider region in between); thus, dividing all the pixels within the window into three groups—the lower intensity impulse noise, the uncorrupted pixels (in the middle) and the higher intensity impulse noise.

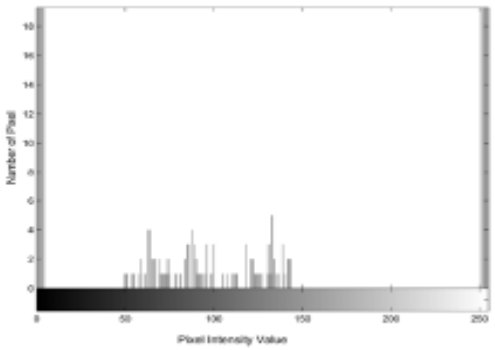


Figure 2.1: Histogram distribution of 21x21 subimage of "Lena" corrupted by 80% impulse noise.

The boundary discriminative process consists of two iterations, in which the second iteration will only be invoked conditionally. In the first iteration, an enlarged local window with a size of 21 x 21 (empirically determined) is used to examine whether the considered pixel is an uncorrupted one. If the pixel fails to meet the condition to be classified as "uncorrupted" (i.e., not falling onto the middle cluster), the second iteration will be invoked to further examine the pixel based on a more confined local statistics by using a 3 x 3 window. In summary, the steps of the proposed BDND are:

Step 1) Impose a 21 x 21 window, which is centered around the current pixel.

Step 2) Sort the pixels in the window according to the ascending order and find the median, *med*, of the sorted vector $\mathbf{V}_O$.

Step 3) Compute the intensity difference between each pair of adjacent pixels across the sorted vector $\mathbf{V}_O$ and obtain the difference vector $\mathbf{V}_D$.

Step 4) For the pixel intensities between 0 and *med* in the $\mathbf{V}_O$, find the maximum intensity difference in the $\mathbf{V}_D$ of the same range and mark its corresponding pixel in the $\mathbf{V}_O$ as the boundary b_1.

Step 5) Likewise, the boundary b_2 is identified for pixel intensities between *med* and 255; three clusters are, thus, formed.

Step 6) If the pixel belongs to the middle cluster, it is classified as "uncorrupted" pixel, and the classification process stops; else, the second iteration will be invoked in the following.

Step 7) Impose a 3 x 3 window, being centered around the concerned pixel and repeat Steps 2) –5).

Step 8) If the pixel under consideration belongs to the middle cluster, it is classified as "uncorrupted" pixel; otherwise, "corrupted."

2.2.1 A Scenario

For the understanding of the algorithmic steps mentioned above, a 5 x 5 (instead of 21 x 21) windowed sub-image with the center pixel "202" (being boxed) is used as an example for illustrating the proposed BDND process as follows:

$$\mathbf{W}= \begin{pmatrix} 255 & 255 & 47 & 255 & 39 \\ 50 & 255 & 255 & 0 & 0 \\ 0 & 0 & \underline{202} & 224 & 205 \\ 62 & 255 & 0 & 0 & 255 \\ 255 & 72 & 81 & 0 & 179 \end{pmatrix}$$

- Pixel intensities are sorted in the ascending order and represented as a vector, where the median *med* is <u>81</u>: $\mathbf{V_O}$ = [0 0 0 0 0 0 0 39 47 50 62 72 <u>81</u> 179 202 205 224 255 255 255 255 255 255 255 255].
- The vecot of intensity differences between each pai of two adjacent pixles in $\mathbf{V_O}$ is computed as $\mathbf{V_D}$ = [0 0 0 0 0 0 39 8 3 12 10 9 98 23 3 22 31 0 0 0 0 0 0 0].
- For the pixels with intensities between 0 and *med* in the $\mathbf{V_O}$, the corresponding maximum difference in the $\mathbf{V_D}$ is 39, which is the difference between the pixel intensities 0 and 39.
- For the pixels with intensities between *med* and 255 in the $\mathbf{V_O}$, the maximum difference in the $\mathbf{V_D}$ is 39 which is the difference between the pixel intensities 0 and 39.
- For the pixels with intensities between *med* and 255 in the $\mathbf{V_O}$, the maximum difference in the $\mathbf{V_D}$ is 98, which is the difference between the pixel intenisites 81 and 179.
- Hence, b_1 =0 and b_2 =81. Thus, the lower intensity cluster is {0,0,0,0,0,0,0}, the medium-intensity cluster is {39,47,50,62,72,81}, and the higher intensity cluster is {179,202,205,224,255,255,255,255,255,255,255,255}.
- Since the center pixel "202" belongs to the higher intensity clister, hence, the second iteration needs to be invoked, and a 3 x 3 window is imposed and centered around it:

$$\mathbf{W_{3X3}}= \begin{pmatrix} 255 & 255 & 0 \\ 0 & \underline{\mathbf{202}} & 224 \\ 255 & 0 & 0 \end{pmatrix}$$

- Now, the pixel intensities are sorted and represented in the vector form $\mathbf{V_O}$ = [0 0 0 0 <u>202</u> 224 255 255 255].
- As before, the vector of intensity differences is computed: $\mathbf{V_D}$ =[0 0 0 202 22 31 0 0].

- The first maximum difference is 202, which is the difference beteen the pixel intensities 0 and 202. The second maximum difference is 31, which is the difference between the pixel intensities 224 and 255.
- Hence, b_1 =0 and b_2 = 224. Thus, the lower intensity cluster is {0,0,0,0}, the medium-intensity cluster is {202,224}, and the higher intensity cluster is {255, 255, 255}.
- At the end of the discrimination process, the center pixel "202" is classified as an "uncorrupted" pixel, since it belongs to the middle cluster.

2.3 Issues

There are mainly two issues in boundary discrimination noise detection algorithm which are exploited in the present project and are reduced by proposing a new fuzzy based algorithm in chapter 4. These issues are:

1. In step 4)-6), pixels are classified into three clusters namely: low intensity clusters, high intensity cluster and middle intensity cluster. The pixels in the middle cluster are marked as "uncorrupted" whereas other two are suspected to be "corrupted" pixels which are further confirmed with second iteration. Here the classification performed by selecting strict thresholds b_1 and b_2. Selection of these kinds of thresholds makes an algorithm rigid and thus cannot adapt to the increasing noise density and also for random type of impulse noise.
2. The version of the algorithm for color images has no mention about the color components. The algorithm is not able to remove impulse noise present in the color components. Our technique presented later in the chapter considers this interaction to reduce further noise in color images.

Chapter 3
FUZZY IMAGE PROCESSING

Before discussing the proposed fuzzy based impulse filter, let's have a brief review of the fuzzy sets and fuzzy logics. The chapter will conclude by discussing the need and importance of fuzzy image processing.

3.1 Fuzzy Sets

Fuzzy set theory [48] is the extension of conventional (crisp) set theory. It handles the concept of partial truth (truth values between 1 (completely true) and 0 (completely false)). It was introduced by Prof. Lotfi A. Zadeh of UC/Berkeley in 1965 [49] as a mean to model the vagueness and ambiguity in complex systems.

The idea of fuzzy sets is simple and natural. For instance, we want to define a set of gray levels that share the property dark. In classical set theory, we have to determine a threshold, say the gray level 100. All gray levels between 0 and 100 are element of this set; the others do not belong to the set. But the darkness is a matter of degree. So, a fuzzy set can model this property much better. To define this set, we also need two thresholds, say gray levels 50 and 150. All gray levels that are less than 50 are the full member of the set, all gray levels that are greater than 150 are not the member of the set. The gray levels between 50 and 150, however, have a partial membership in the set (right image in Fig.3.1).

3.2 Principles of Fuzzy Logic

The term fuzzy logic has been used in two different senses. In a narrow sense, fuzzy logic refers to a logical system that generalizes the two-valued logic for reasoning under uncertainty. In a broad sense, fuzzy logic refers to all of the theories and technologies that employ fuzzy sets. Even though this

broad sense, we can explain the basics of fuzzy logic by using the following three basic concepts: (1) the above commented fuzzy sets, (2) linguistic variables and (3) fuzzy if-then rules.

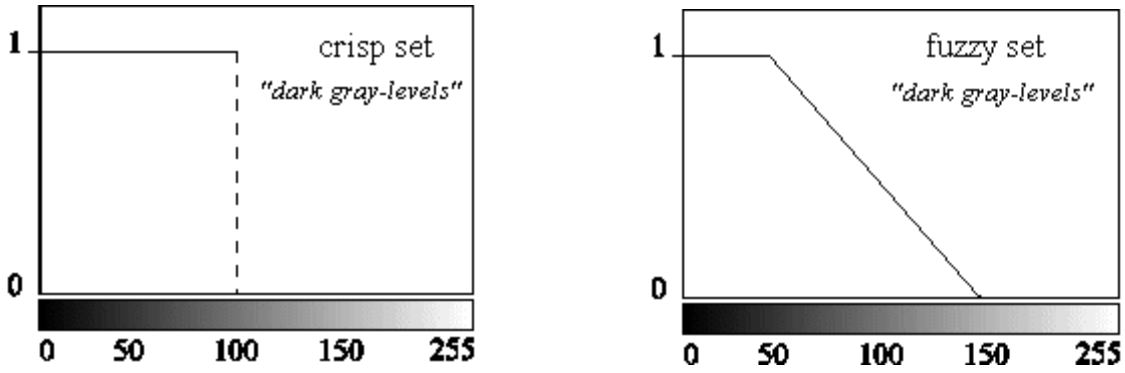


Figure 3.1: Representation of "dark gray-levels" with a crisp and a fuzzy set

3.2.1 Linguistic variables

Having introduced the fundamental concept of fuzzy set, it is natural to see how it can be used. Like a conventional set, a fuzzy set can be used to describe the value of a variable. For example, the sentence "The amount of trading is heavy" uses a fuzzy set "Heavy" to describe the quantity of the stock market trading in one day. More formally it is expressed as: *TradingQuantity is Heavy*. The variable *TradingQuantity* in this example demonstrates an important concept in fuzzy logic: the linguistic variable. A linguistic variable enables its value to be described both qualitatively by a linguistic term (i.e., a symbol serving as the name of a fuzzy set) and quantitatively by a corresponding membership function (which expresses the meaning of the fuzzy set). The linguistic term is used to express concepts and knowledge in human communication, whereas membership function is useful for processing numeric input data.

A linguistic variable is like a composition of a symbolic variable (a variable whose value is a symbol) and a numeric variable (a variable whose value is a number). In our example about stock market trading activities, there are certainly many other linguistic descriptions about the trading quantity such as "light", "moderate", "heavy", and so on. All these linguistic, descriptions, those are indeed imprecise and vague, can be managed using fuzzy sets. In this way, the numerical value of the variable *TradingQuantity* is expressed in terms of its membership degrees to the fuzzy sets used in the

representation. Figure 2.1 shows an example of representation of the linguistic descriptions of the variable *TradingQuantinty* using fuzzy sets.

3.2.2 Fuzzy if-then rules

Among all the techniques developed used fuzzy sets, the fuzzy *if-then* rule (or, in short, fuzzy rule) is by far the most visible one due to its wide range of successful applications. Fuzzy rules have been applied to many disciplines such as control systems, decision making, pattern recognition and system modeling [50]. Fuzzy rules also play a critical role in industrial applications ranging from consumer products, robotics, manufacturing, process control, medical imaging, to financial trading.

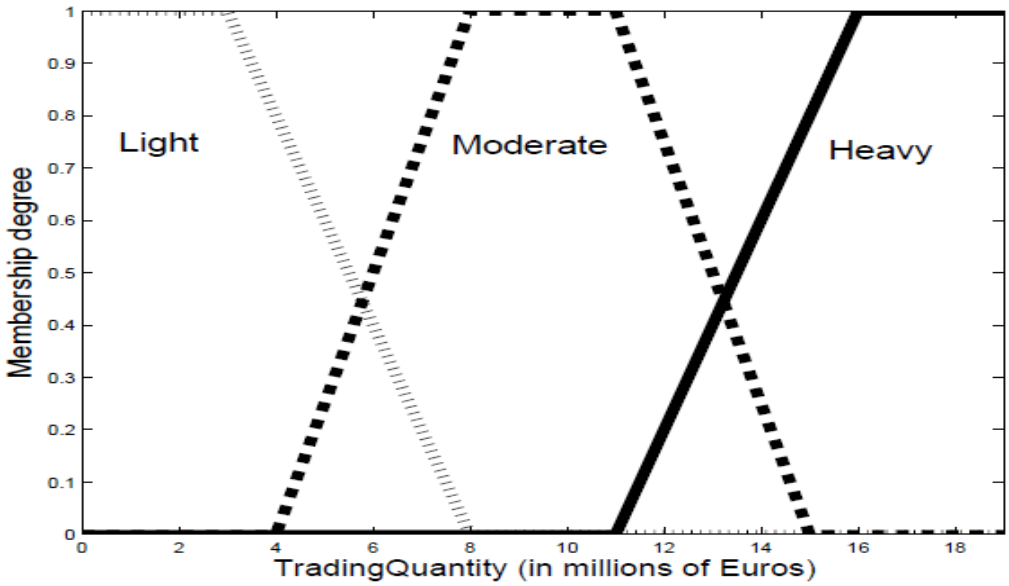


Figure 3.2: Example of representation of the linguistic descriptions of the variable *TradingQuality* using fuzzy sets

Fuzzy rule-based inference can be understood from several viewpoints. Conceptually, it can be understood using the metaphor of drawing a conclusion using a panel of experts. Mathematically, it can be viewed as an interpolation scheme. Formally, it is a generalization of a logic inference called *modus pones.*

In a classical logic, if we know a rule is true and we also know the antecedent of the rule is true, then it can be inferred, by *modus pones*, that the consequent of the rule is true. Analogously to the classical

logic case, the structure of a fuzzy rule has two components: an if-part (also referred to as the antecedent) and a then-part (also referred to as the consequent).

3.3 Fuzzy Image Processing

Fuzzy image processing is not a unique theory. It is a collection of different fuzzy approaches to image processing. Nevertheless, the following definition can be regarded as an attempt to determine the boundaries:

"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.

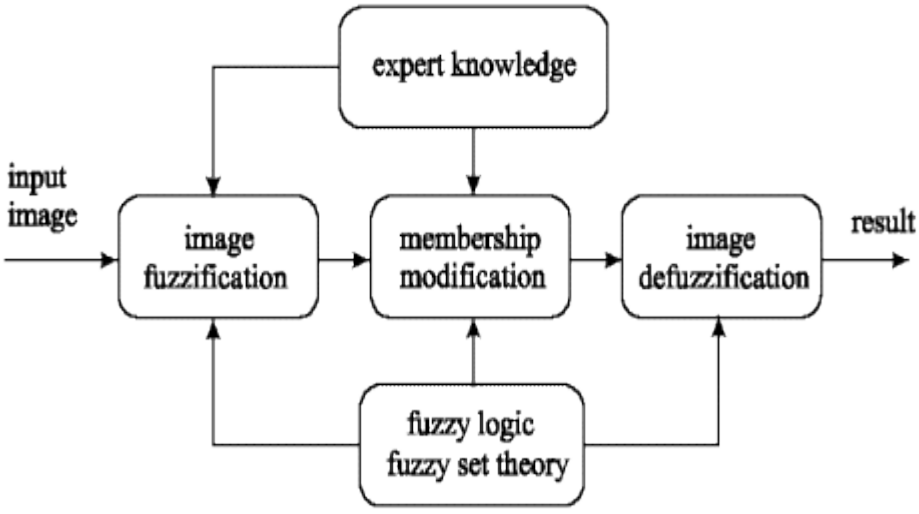


Figure 3.3: The general structure of fuzzy image processing

The fuzzification and defuzzification steps are due to the fact that we do not possess 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 (modification of membership values, see Fig.3.3). After the image

data are transformed from gray-level plane to the membership plane (fuzzification), appropriate fuzzy techniques modify the membership values. This can be a fuzzy clustering, a fuzzy rule-based approach, a fuzzy integration approach and so on.

3.3.1 Why Fuzzy Image Processing?

There are many reasons [48] to support the use of fuzzy techniques in image processing. The most important of them are as follows:

1. Fuzzy techniques are powerful tools for knowledge representation and processing
2. Fuzzy techniques can manage the vagueness and ambiguity efficiently

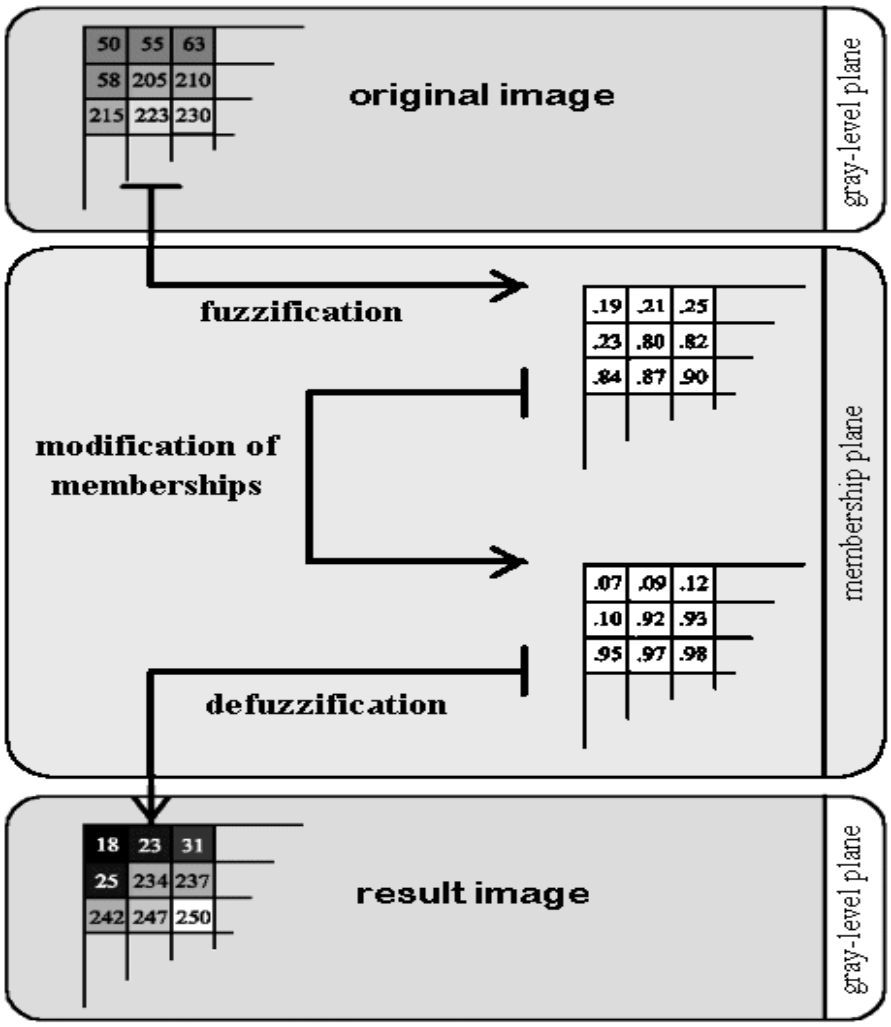


Figure 3.4: Steps of Fuzzy Image Processing

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.

- Grayness ambiguity
- Geometrical fuzziness
- Vague (complex/ill-defined) knowledge

These problems are fuzzy in the nature. 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.

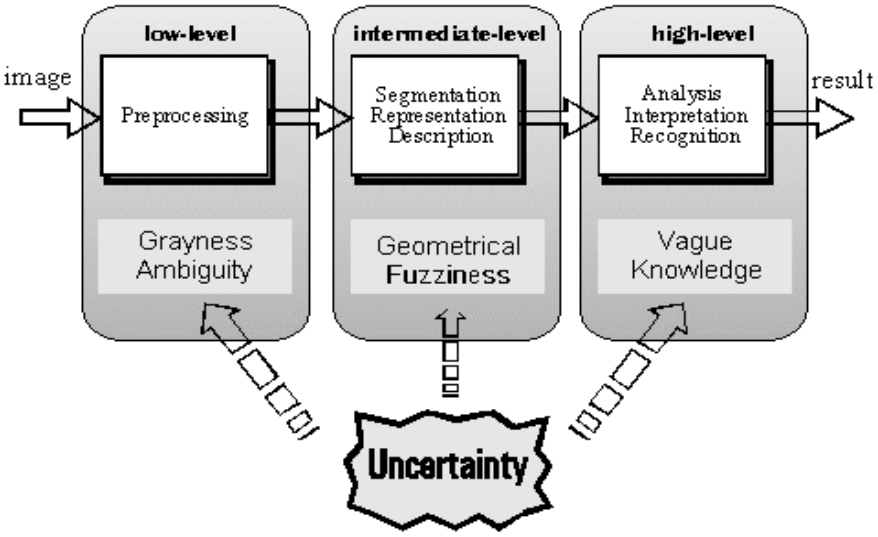


Figure 3.5: Uncertainty/imperfect knowledge in image processing

As an example, we can regard the variable color as a fuzzy set. It can be described with the subsets yellow, orange, red, violet and blue as ***color*** = *{yellow, orange, red, violet, blue}*. The non-crisp boundaries between the colors can be represented in much better way. A soft computing becomes possible.

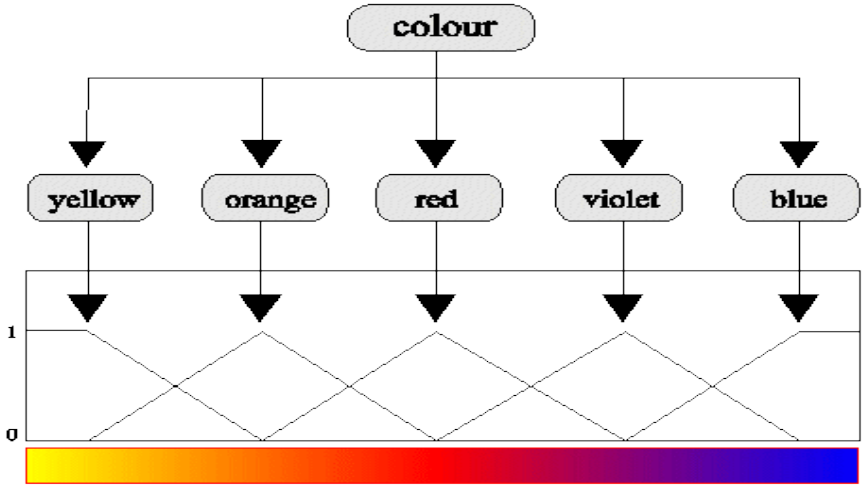


Figure 3.6: Representation of colors as fuzzy subsets

Chapter 4

PROPOSED APPROACH

4.1 Fuzzy Impulse Noise Filter

This chapter presents a fuzzy based approach to boundary discriminative noise detection. The proposed algorithm present an impulse filter which is applicable to both gray scale and color images with slight difference in their implementation. The algorithm is a two step process in which first step is designed for detection and filtering of images, whereas second step is specifically designed for color images. The second step reduces the residue noise presents in the color components of an image by exploiting the interactions between the color components. The filtered image obtained after first stage is given as input to the second stage for further reduction in noise present in color components by adding a correction term. The output image thus obtained is an output of the complete algorithm. Since there are no color components present in gray scale images, the filtered image after stage one is considered as output of gray scale images.

The basic strategy of BDND [18] is to examine each pixel in its neighborhood from coarse to fine. The pixel under consideration is examined in two stages to mark it as "corrupted" or "uncorrupted". The most critical part of this is the determination of the decision boundaries. This criterion can be summarized as follows:

$$y\,(i,j) = \begin{cases} Low\ intensity\ noise, & y(i,j) \le b_1 \\ Uncorrputed, & b_1 < y(i,j) \le b_2 \\ High\ intensity\ noise, & y(i,j) > b_2 \end{cases} \qquad (4.1)$$

where *y(i,j)* is the intensity of the pixel being considered, and b_1 and b_2 are two decision boundaries. The algorithm for the same is discussed in chapter 2 with an example. There are two major issues with the above algorithm as pointed out. Firstly, the boundaries selected in step 4) of the algorithm are strict thresholds and secondly, color components are not handled. Our proposed algorithm for

detection is basically on the same lines of BDND but it applies a fuzzy based approach in order to consider the “real” nature of images.

As discussed in chapter 2, during a detection procedure, a pixel is subjected to 21x21 window. This is because the histogram of 21x21 window as shown in fig. 2.1 is neither too flat nor too busy. Thus BDND uses thresholds to classify the pixels in three clusters. This is where the limitation came into picture. The thresholds selected are strict which tend to select the noisy pixels as uncorrupted as no overlapping of boundaries is considered. Because of this reason a fuzzy rule based system is embedded into the detection process to consider smooth boundaries. The decision boundaries are fuzzified by devising three membership functions, each representing a single cluster. The difference between the clusters and membership functions is that they are overlapping in nature i.e. the boundaries of the cluster overlap in membership functions. Different fuzzy rules are designed to consider all the possibilities of the existence of a pixel in any membership function. Each rule is given a weight, according to which a degree of noisyness is calculated which will decide which pixel is corrupted and which will remain unchanged. The complete procedure is explained in details in later sections of this chapter. The first part of detection process ends by generating a decision map, with “0” representing corrupted and “1” represents “uncorrupted” pixel. The second part of detection process starts by confirming the pixel’s class by imposing a 5x5 window rather than 3x3 window in BDND. The selection of 5x5 window is by extensive experimentation. The decision map formed after this step is given to filtering stage to filter out the corrupted pixels by replacing them with median. The design of adaptive filtering process is discussed later. The output thus obtained is a filtered image and can be considered an output image of fuzzy based filter in case of gray scale images. As mentioned, for color images second step of the filter is applied in order to remove left over impulse noise in color components by checking the difference between the color components and devising a membership function to calculate a correction term to be added to a pixel. The block diagram of the complete system is presented in figure 4.1.

4.2 Noise Detection for gray scale images

The main aim of detection step is to classify the pixels as corrupted (high or low intensity) or uncorrupted. Therefore, it is carried out by first finding the decision boundaries. These boundaries are

itself calculated in two stage i.e. first the pixel are checked globally by passing through a 21x21 window [18] and then locally by 5x5 window just to confirm the classification. The boundaries selected are fuzzified in order to avoid the rigidness of the strict thresholds. The output of detection step is a decision map, where "0" represents corrupted and "1" represents an uncorrupted pixel.

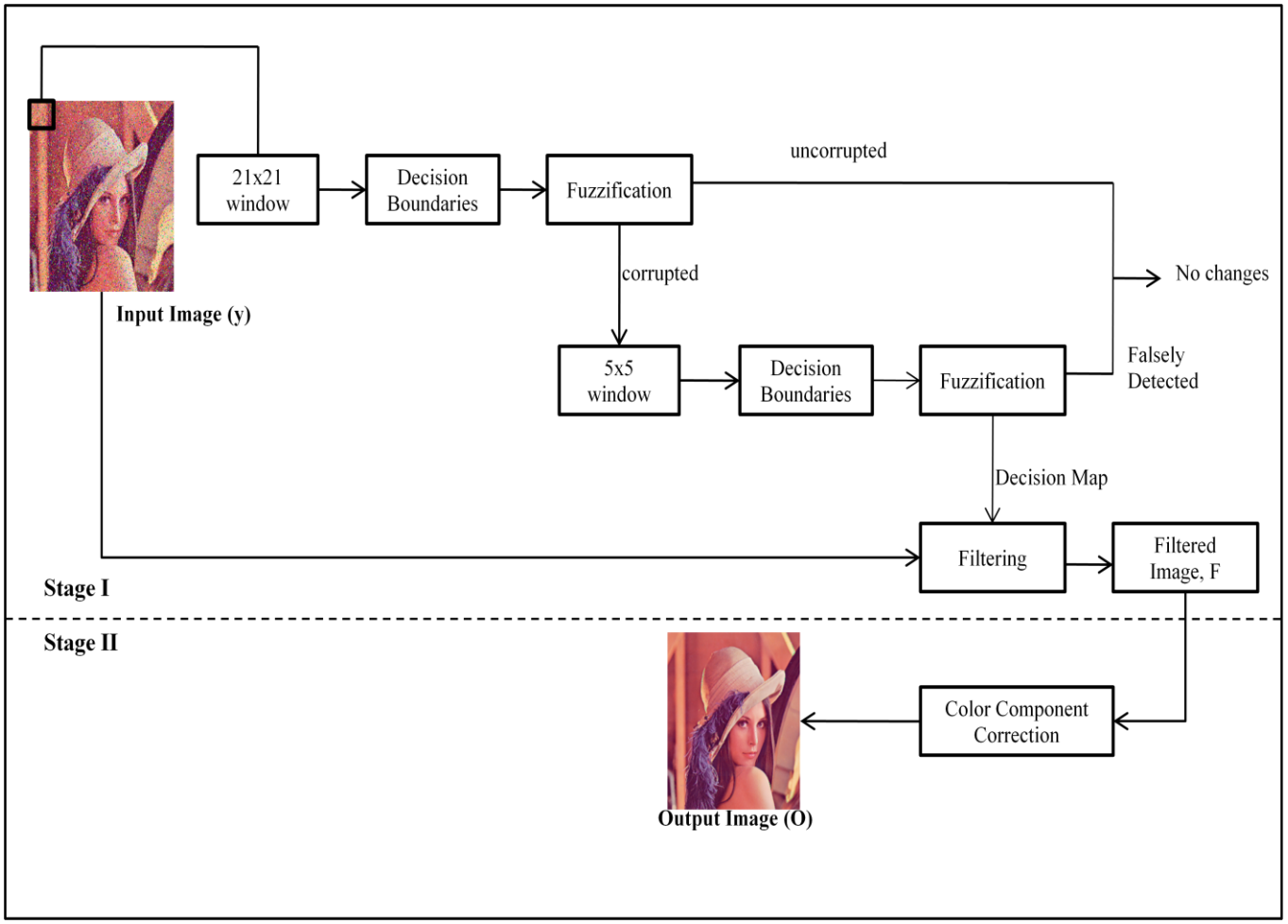


Figure 4.1: Block diagram for complete system

Steps for detection are as follows:

STEP 1) Impose 21x21 window centered on *y(i,j)*

STEP 2) Sort the pixels of the window to an ordered vector v_o and find the median *med.*

STEP 3) Compute the differences between each pair of adjacent pixels in vector v_o denoted as v_D.

STEP 4) Find the pixels which correspond to maximum differences in the intervals of [0, med] and (med, 255]. And set these two pixels' intensities as the decision boundaries b_1 and b_2 respectively.

STEP 5) With b_1, b_2 and *med* values, three membership functions are formed. This step is known as ***"adaptive fuzzification of decision boundaries".***

Three membership functions μ_L, μ_M and μ_H are devised corresponding to a pixel (for each color component) as shown in the figure 1. The membership function, μ_L represents a fuzzy set "low" (L), indicating the pixels belonging to low intensity corrupted class. Similarly μ_H represents fuzzy set "high" (H) for high intensity corrupted class. And μ_M is for fuzzy set "medium" (M) for pixels which are uncorrupted. The closer the value of a pixel to the boundaries b_1 and b_2, the higher is the possibility of a pixel to lie in one of the corrupted classes (low or high).

$$\mu_L = \begin{cases} 1, & y(i,j) \le b_1 \\ \frac{med}{med-b_1} - \frac{y(i,j)}{med-b_1}, & b_1 < y(i,j) \le med \\ 0, & otherwise \end{cases} \tag{4.2}$$

$$\mu_M = \begin{cases} 1, & b_1 < y(i,j) \le b_1 \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

$$\mu_H = \begin{cases} 1, & y(i,j) \ge b_1 \\ \frac{y(i,j)}{b_2-med} - \frac{med}{b_2-med}, & med < y(i,j) \le b_2 \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

The degree of noise present in the pixel is ascertained by forming 27 rules for rule-based fuzzy inference system. These are used to validate the existence of pixel in the particular class of output. The example of rule is as follows:

Rule 1: *If the pixel P lies exclusively in L then P is corrupted (low intensity).*

Rule 2: *If the pixel P lies exclusively in M then P is uncorrupted.*

Rule 3: *If the pixel lies exclusively in H completely P is corrupted (high intensity).*

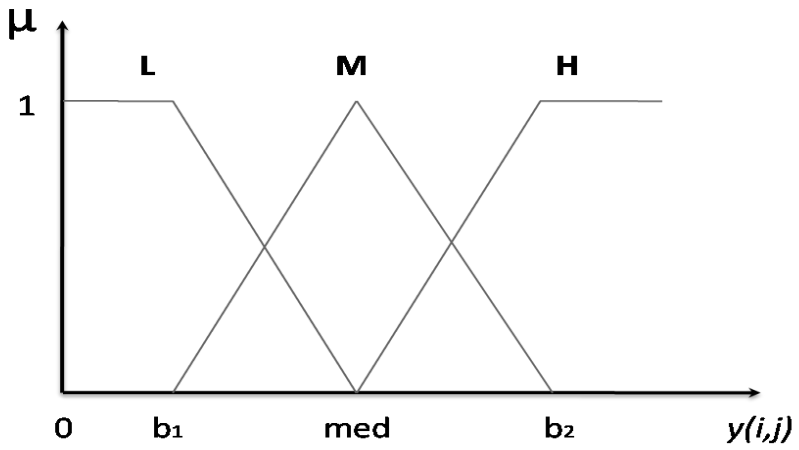


Figure 4.2: The membership function

The degree of exclusiveness is obtained by the value of membership functions computed in the previous step. Similarly more rules are followed with the combinations of values. For example if pixel *P* lies inside the boundaries of both L and M, then its class is obtained by mathematically calculating the percentage of noisyness. The degree of noisyness of the pixel is given as:

$$N = \frac{\sum_z N_i \mu_{prem}(i)}{\sum_z \mu_{prem}(i)} \tag{4.5}$$

where *z* is the number of rules, N_i is the weight of each premise and $\mu_{prem}(i)$ is a certainty of premise for i^{th} rule. Here z is 27, thus *i* ranges from 1 to 27. The pixel is classified in the particular output class by following the equation (4.5).

$$Class = \begin{cases} uncorrputed, & N \leq Threshold \\ corrupted, & N > Threshold \end{cases} \tag{4.6}$$

The value of threshold is experimentally computed.

STEP 6) The validation of noisy candidates is confirmed by imposing a 5x5 window and repeating steps 2)-5).

Figure 4.3 shows the complete flow chart to depict the above steps.

4.3 Filtering

The decision map obtained in previous section is inputted to this stage. Only the uncorrupted pixels are selected for filtering i.e. the pixels represented as "0" in decision map. The important change in the filtering technique is the approach of incremental window size [11, 18]. The window size is increased from 3x3 to 7x7 depending on the criteria that number of uncorrupted pixels should be more than or equal to half of the total number of pixels in that particular window i.e.

$$\textit{No. of uncorrupted pixels in a window} \geq \frac{1}{2}(W \text{ x } W) \tag{4.7}$$

where W is a window size.

If the above condition holds the median value is assigned otherwise W is incremented by 1. The window size is limited to 7x7 because for larger windows severe blurring takes place in high density noise. The median of the particular window is assigned to the pixel. The filtered image ***F*** is obtained in passed onto to the second stage of the impulse filter for color images otherwise for gray scale images, ***F*** is an output noise filter..

4.4 Noise Correction for Color Images

The noise detection in color images is performed on the same line as that of gray scale images. The membership functions are formed for each color components. The only difference for color images is a second stage of **"color correction".** The filtered image ***F*** obtained from the filtering step is subjected to this correction step. This filter invokes the interaction between the color components [20] to remove further left-over impulse noise present in color components. As most widely used color space model, RGB color space will be used in our work. Here the pixel is represented as ***F(i,j,z)*** where z ranges from 1 to 3 representing 1 for red (*r*), 2 for green (*g*) and 3 for blue (*b*) component.

The correction term to be added to the components is decided by examining the differences in each color pair and forming a membership function. The steps are as follows:

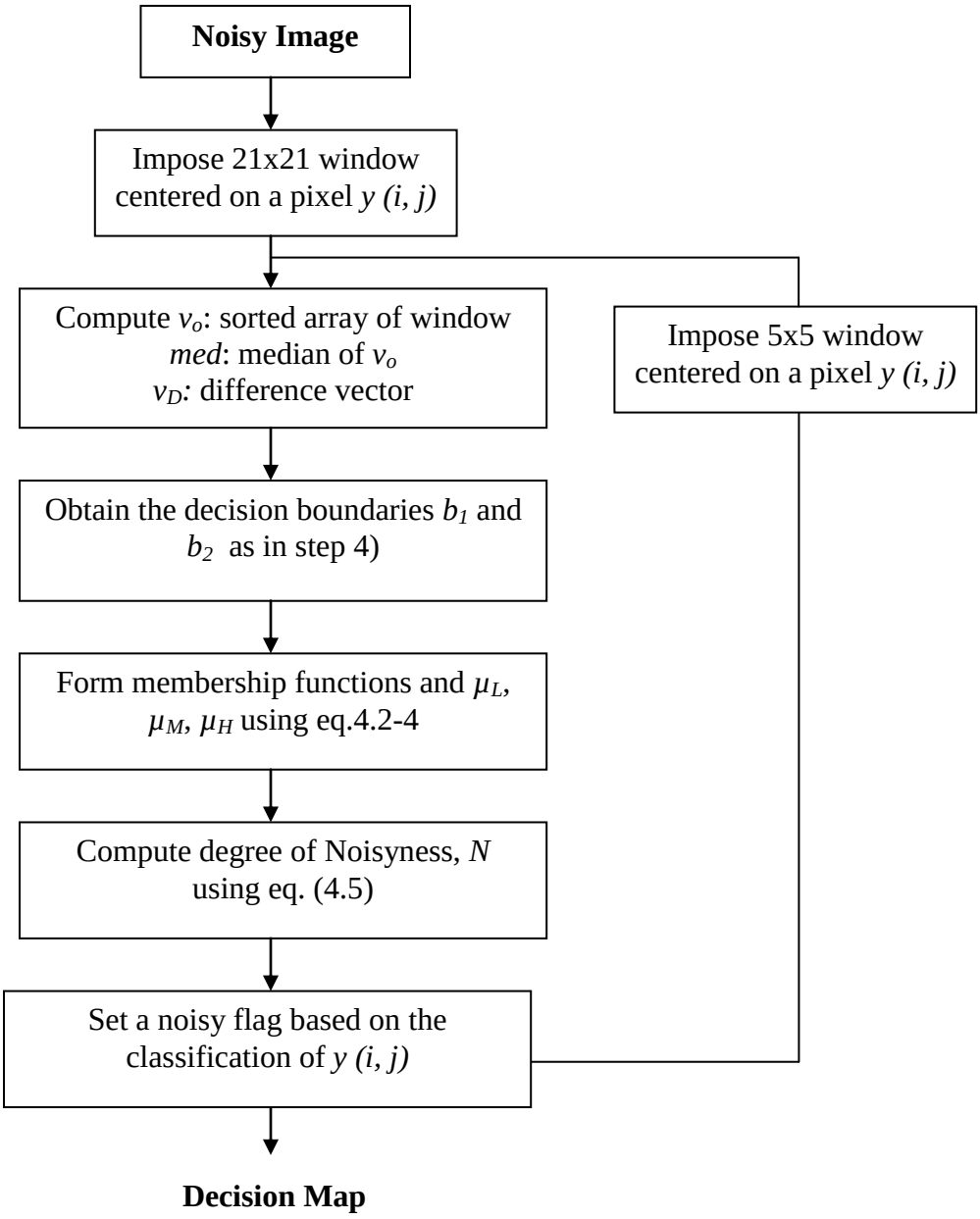


Figure 4.3: Flow chart for detection stage

The steps are as follows:

STEP 1) Difference between the color pairs is calculated to check for any residue impulse noise in individual color components:

$$\begin{aligned} d_{rg}(i,j) &= |F(i,j,1)-F(i,j,2)| \\ d_{rb}(i,j) &= |F(i,j,1)-F(i,j,3)| \\ d_{gb}(i,j) &= |F(i,j,2)-F(i,j,3)| \end{aligned} \tag{4.8}$$

where $d_{rg}(i,j)$, $d_{rb}(i,j)$, $d_{gb}(i,j)$ represents differences between red-green, red-blue and green-blue components for the same pixel of the filtered image **F**.

STEP 2) A fuzzy rule system is framed to compute the degree noise present in the color component of the concerned pixel. For each component the rule of the following form is formed:

Rule: *If $d_{rg}(i,j)$ is **Large** and $d_{gb}(i,j)$ is **Large** then the green component is **noisy**.*

The significance of this rule holds only when there is noise left in the color components. The application of this stage nullifies when there is region of same color as the differences will be large for that area. Let say there is a green area present in an image; this step will not consider this area as noisy as the median of the region will also be green. Thus it makes the filter more efficient with respect to noises present in the color components. Similar fuzzy rules are coined for other color components. The definition of "Large" is expressed by the membership function μ_l with the parameters β_1 and β_2 as shown in fig. 4.4. For every difference computed above (generalized as d), μ_l is given by:

$$\mu_l = \begin{cases} 0, & d \geq \beta_2 \\ \frac{d}{\beta_2-\beta_1} - \frac{\beta_1}{\beta_2-\beta_1}, & \beta_1 \leq \mathrm{d} < \beta_1 \\ 0, & d < \beta_1 \end{cases} \tag{4.9}$$

STEP 3) The degree of noise, n_d in the color component is obtained by the minimum value of the membership functions corresponding to the differences. The correction term is given by:

$$\Delta(i,j,z) = n_d(i,j,z) \times (med(i,j,z)-F(i,j,z)) \tag{4.10}$$

STEP 4) The final output of an impulse filter is given by:

$$\boldsymbol{O(i, j, z)} = F(i, j, z) + \Delta\,(i, j, z) \tag{4.11}$$

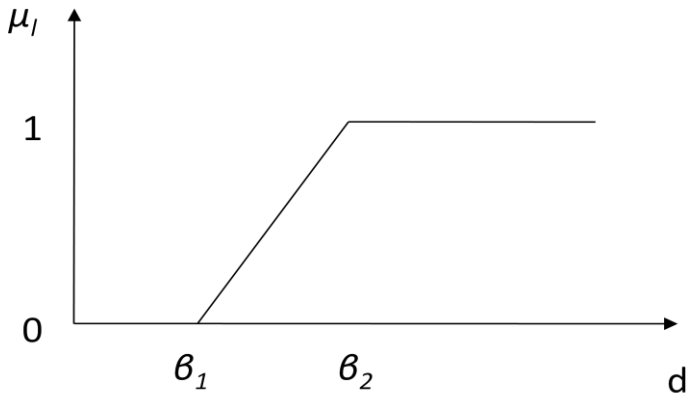


Figure 4.4: The membership function "large"

Figure 4.5 shows the flow chart to represent the second stage of the proposed algorithm.

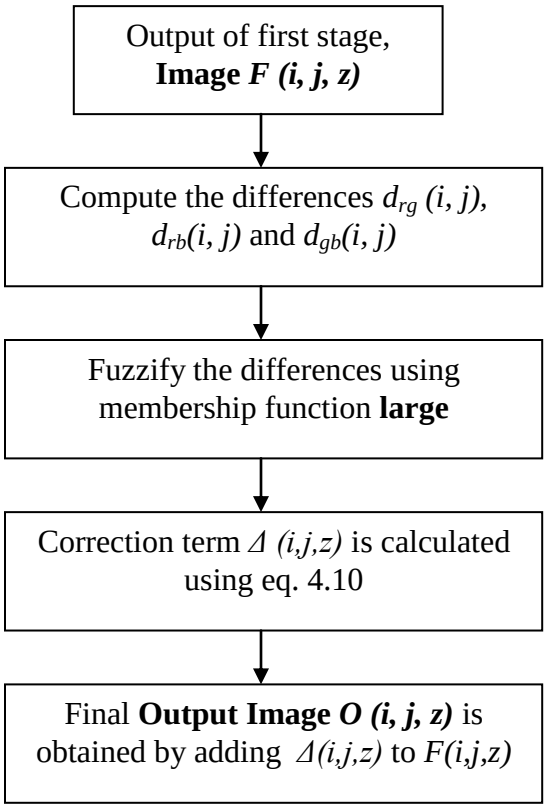


Figure 4.5: Flow chart of Noise Correction

Chapter 5

SIMULATION RESULTS AND COMPARATIVE ANALYSIS

This chapter will show the simulation results and comparisons. The four existing algorithms i.e. ASWM [23], BDND [18], SMF [24] and DBAIN [19] are compared with our proposed approach (PA). The image metrics like Peak Signal to Noise ratio (PSNR), Mean Square Error (MSE), Correlation Coefficient (Corr) and Delta E (ΔE) value for color difference are used in this chapter to compare between the existing and proposed algorithm. The first three metrics are used to compare the gray scale images whereas the fourth metrics is specifically used to find the color differences between the images for better comparison.

5.1 Specifications

Software: Matlab 2010b , Microsoft WIN XP Version 2002, Service Pack 3

Hardware: Intel(R) Core(TM)2 Duo CPU E7200 @ 2.53 GHz, 1.99 GB of RAM.

5.2 Input Images

The algorithms are run on two input mages “Lena.tif” and “baboon.tif”. Both are of size 256x256.

The input image is corrupted with impulse noise corresponding to four noise models as discussed in the above chapters. The simulation results are divided into two sections i.e. one for gray scale monochrome images and second for color images. The image metrics used to compare the results for grayscale and color images are different as discussed in following sections.

Figure 5.1: Gray Scale and Color version of "Lena.tif"; size=256x256

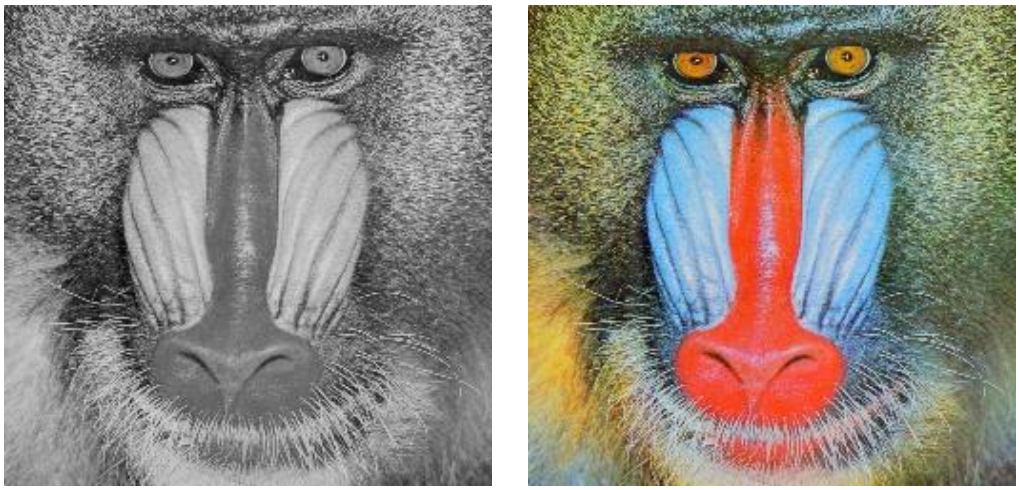


Figure 5.2: Gray Scale and Color version of "Baboon.tif"; size=256x256

5.3 Gray Scale

The present section show the quantitative, qualitative and graphical comparisons of algorithms run on gray scale monochrome images. For faster comparison only a portion of complete image is compared. The coordinates selected are 100:200,100:200.

5.3.1 Quantitative Performance

The performance of the algorithms is compared quantitatively in terms of PSNR, MSE and Correlation Coefficient (r) values. The values are calculated for each noise model extensively.

5.3.1.1 Noise Model 1

Noise	ASWM	BDND	SMF	DBAIN	PA
10%	30.8319	31.7864	32.7434	34.2538	35.9998
20%	29.5805	30.3849	30.9412	32.1411	32.9403
30%	28.5340	28.9833	29.1740	29.5012	29.7801
40%	26.2762	27.2462	26.5411	27.2949	27.2268
50%	20.5751	24.5979	24.5313	24.6926	25.4708
60%	14.2724	22.3144	23.3032	23.0110	23.8326
70%	10.8064	19.8931	21.0751	21.3468	22.2821
80%	8.0358	16.9758	17.5593	18.7533	19.7797

(a)

Noise	ASWM	BDND	SMF	DBAIN	PA
10%	54.1111	43.4351	34.8450	24.6095	16.4626
20%	72.1816	59.9777	52.7664	40.0283	33.3007
30%	92.0612	82.8230	79.2650	73.5128	68.9395
40%	154.4773	123.5537	145.3340	122.1759	124.1083
50%	574.0746	227.3461	230.8581	222.4461	185.9503
60%	2450.4	384.6252	306.3114	327.6246	271.1583
70%	5443.0	671.658	511.6522	480.6135	387.5011
80%	10302.0	1314.9	1149.6	873.2698	689.4579

(b)

Noise	ASWM	BDND	SMF	DBAIN	PA
10%	0.9896	0.9916	0.9933	0.9953	0.9968
20%	0.9861	0.9884	0.9898	0.9923	0.9936
30%	0.9824	0.9839	0.9847	0.9858	0.9866
40%	0.9708	0.9760	0.9716	0.9740	0.9759
50%	0.9010	0.9552	0.9548	0.9600	0.9636
60%	0.6837	0.9232	0.9391	0.9365	0.9467
70%	0.4980	0.8657	0.8967	0.9078	0.9226
80%	0.3491	0.7309	0.7702	0.8322	0.8592

(c)

Table 5.1: (a) PSNR (b) MSE and (c) r values obtained after applying different algorithms on “Lena.tif” corrupted under various noise densities for noise model 1

5.3.1.2 Noise Model 2

Noise		ASWM	BDND	SMF	DBAIN	PA
Salt	**Pepper**					
10	**40**	20.2651	24.9045	25.0823	26.9299	29.0740
20	**30**	16.0569	24.2282	25.7020	25.3987	26.5447
30	**20**	14.3008	19.6833	26.0240	24.7242	26.9852
40	**10**	19.2506	23.4673	27.6948	27.4347	28.7233

(a)

Noise		ASWM	BDND	SMF	DBAIN	PA
Salt	**Pepper**					
10	**40**	616.5540	211.8507	203.3534	132.8903	81.1107
20	**30**	1624.8	247.5490	176.3111	189.0646	145.2152
30	**20**	2434.5	704.9350	163.7128	220.8326	131.2074
40	**10**	778.7827	294.9491	111.4295	118.3064	87.9333

(b)

Noise		ASWM	BDND	SMF	DBAIN	PA
Salt	**Pepper**					
10	**40**	0.8979	0.9599	0.9603	0.9757	0.9843
20	**30**	0.7963	0.9521	0.9657	0.9656	0.9717
30	**20**	0.6883	0.8710	0.9680	0.9593	0.9745
40	**10**	0.8693	0.9436	0.9784	0.9779	0.9830

(c)

Table 5.2: (a) PSNR (b) MSE and (c) r values obtained after applying different algorithms on "Lena.tif" under noise model 2 with 50% noise density

5.3.1.3 Noise Model 3

Noise Low	Noise High	ASWM	BDND	SMF	DBAIN	PA
[0,9]	**[246,255]**	21.4259	24.3495	25.2047	9.9003	25.7278
[0,19]	**[236,255]**	21.1922	24.3180	23.9992	9.5371	25.3177
[0,29]	**[226,255]**	21.0157	22.8547	21.0540	9.3744	23.9141
[0,39]	**[216,255]**	21.0617	18.8471	18.8674	9.0369	22.3208
[0,49]	**[206,255]**	20.2243	16.0213	15.2665	8.8645	21.8548

(a)

Noise Low	Noise High	ASWM	BDND	SMF	DBAIN	PA
[0,9]	**[246,255]**	471.9461	240.7292	197.6998	8512.1	175.2667
[0,19]	**[236,255]**	456.9892	242.4843	260.9523	8180.7	192.6244
[0,29]	**[226,255]**	412.0100	339.6344	514.1399	7569.1	266.1179
[0,39]	**[216,255]**	373.2249	854.6115	850.6306	7290.7	384.0587
[0,49]	**[206,255]**	494.3705	1637.4	1949.1	6705.7	427.5611

(b)

Noise Low	Noise High	ASWM	BDND	SMF	DBAIN	PA
[0,9]	**[246,255]**	0.9174	0.9525	0.9612	0.2946	0.9659
[0,19]	**[236,255]**	0.8922	0.9523	0.9483	0.2814	0.9596
[0,29]	**[226,255]**	0.9258	0.9324	0.8974	0.2925	0.9477
[0,39]	**[216,255]**	0.9334	0.8327	0.8382	0.2986	0.9307
[0,49]	**[206,255]**	0.9096	0.7199	0.6778	0.3137	0.9257

(c)

Table 5.3: (a) PSNR (b) MSE and (c) r values obtained after applying different algorithms on "Lena.tif" under noise model 3 with 50% noise density

5.3.1.4 Noise Model 4

Noise Low	Noise High	ASWM	BDND	SMF	DBAIN	PA
10	**40**	20.0172	25.9520	26.9715	10.8119	28.5691
20	**30**	16.5356	24.1797	26.0515	9.5582	26.7308
30	**20**	15.4430	25.5944	26.0673	8.8341	26.3753
40	**10**	18.8018	25.8867	27.6972	9.9021	28.1307

(a)

Noise Low	Noise High	ASWM	BDND	SMF	DBAIN	PA
10	**40**	641.7655	166.4477	131.6228	5436.1	91.1112
20	**30**	1455.2	250.3275	162.6798	7255.4	139.1225
30	**20**	1871.5	180.7351	162.0864	8571.7	150.9910
40	**10**	863.5668	177.2172	111.3676	6703.0	100.7880

(b)

Noise Low	Noise High	ASWM	BDND	SMF	DBAIN	PA
10	**40**	0.9012	0.9675	0.9744	0.4376	0.9823
20	**30**	0.8053	0.9511	0.9683	0.3276	0.9729
30	**20**	0.7469	0.9647	0.9684	0.3248	0.9705
40	**10**	0.8554	0.9554	0.9784	0.4193	0.9804

(c)

Table 5.4: (a) PSNR (b) MSE and (c) r values obtained after applying different algorithms on "Lena.tif" under noise model 4 with 50% noise density and m=10

Noise Low	Noise High	ASWM	BDND	SMF	DBAIN	PA
10	**40**	20.7064	24.0282	22.3273	11.4686	25.6694
20	**30**	18.8468	23.5360	21.9220	10.2904	24.1737
30	**20**	16.6223	20.0139	20.0031	9.4017	22.0105
40	**10**	20.5809	22.9291	22.8065	10.2371	23.1965

(a)

Noise Low	Noise High	ASWM	BDND	SMF	DBAIN	PA
10	**40**	551.4325	259.2171	383.4855	4673.2	177.6382
20	**30**	854.6818	290.3223	421.0001	6129.7	250.6731
30	**20**	1426.4	653.2621	654.8893	7521.6	412.5067
40	**10**	573.3113	333.8663	343.4262	6205.4	313.9246

(b)

Noise Low	Noise High	ASWM	BDND	SMF	DBAIN	PA
10	**40**	0.9408	0.9499	0.9261	0.4758	0.9656
20	**30**	0.8705	0.9430	0.9187	0.3746	0.9508
30	**20**	0.7835	0.8690	0.8671	0.3458	0.9174
40	**10**	0.9008	0.9344	0.9328	0.4326	0.9375

(c)

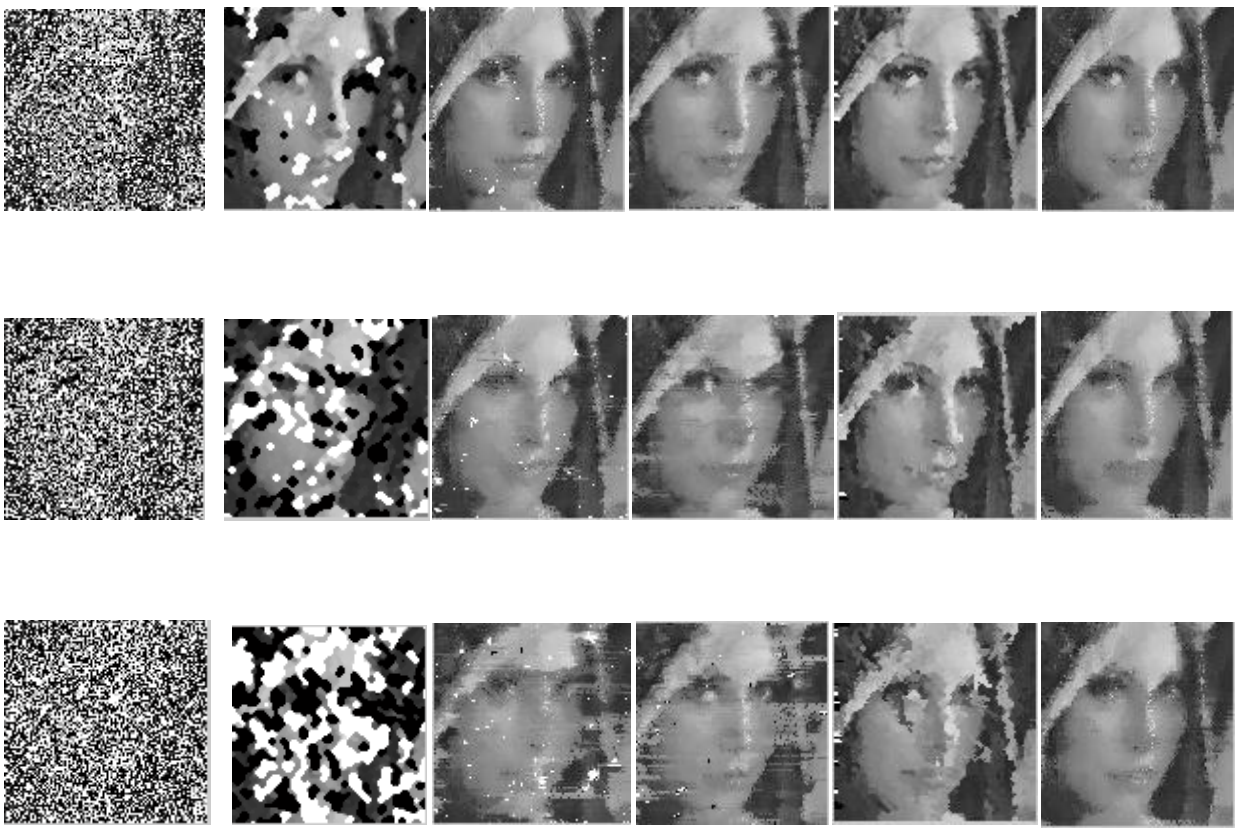
Table 5.5: (a) PSNR (b) MSE and (c) r values obtained after applying different algorithms on "Lena.tif" under noise model 4 with 50% noise density and m=30

5.3.2 Qualitative Performance

The image outputs for each run are as shown below corresponding to the original image present in fig 5.3.

Figure 5.3: Sub-image of the original image used for each run

(a) **(b)** **(c)** **(d)** **(e)** **(f)**



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 5.4: (a): Corrupted images with impulse noise with noise density starting from 10% to 80% (b)-(e): corresponding filtered images of ASWM, BDND, SMF and PA for **NOISE MODEL 1**

Figure 5.5: (a): Corrupted images with impulse noise with noise density 50 % with the salt and pepper percentage changes as {10,40}, {20,30}, {30,20}, {40,10} (horizontally)

(b)-(e): corresponding filtered images of ASWM, BDND, SMF and PA for **NOISE MODEL 2**

(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 5.6

(a): Corrupted images with impulse noise with noise density 50 % with the value of ***m*** range from 10 to 50

(b)-(e): corresponding filtered images of ASWM, BDND, SMF and PA for **NOISE MODEL 3**

(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 5.7

(a): Corrupted images with impulse noise with noise density 50 % with m=10 and salt and pepper percentages differs as {10,40}, {20,30}, {30,20}, {40,10} (horizontally)

(b)-(e): corresponding filtered images of ASWM, BDND, SMF and PA for **NOISE MODEL 4**

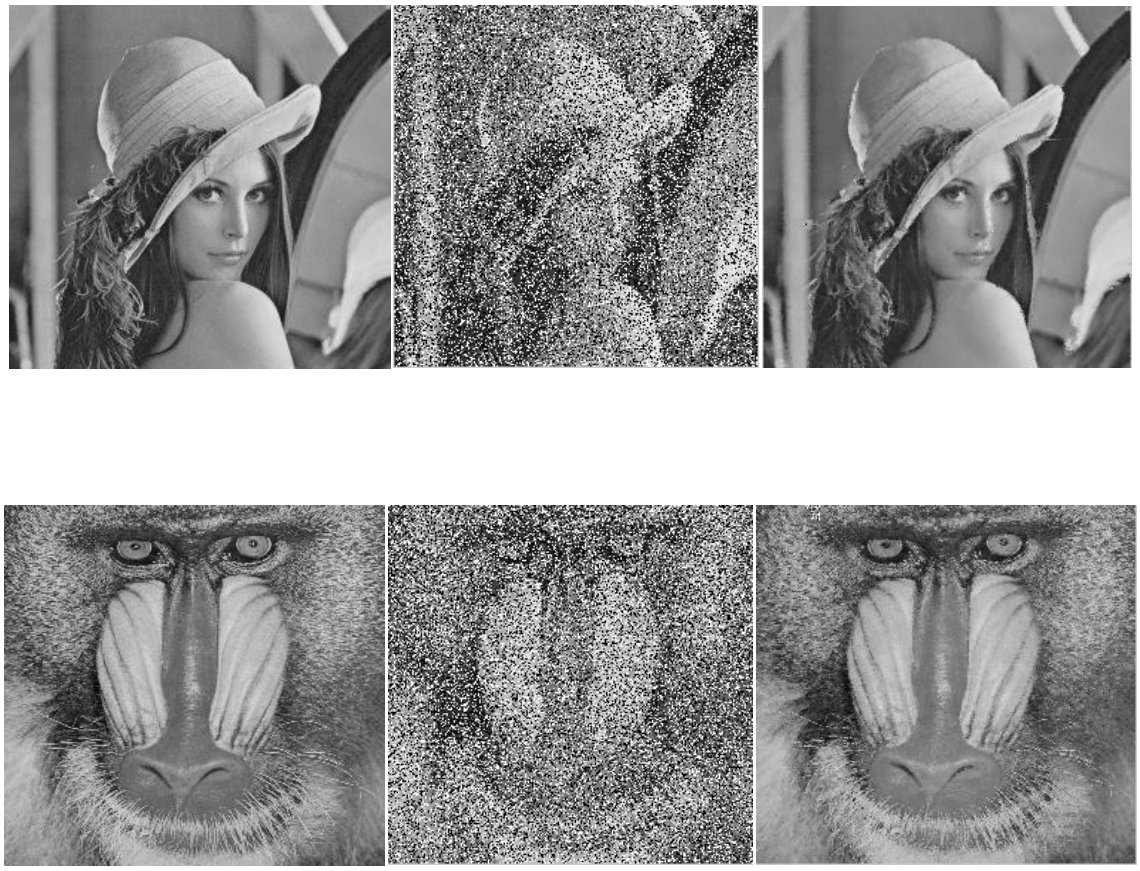
(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 5.8

(a): Corrupted images with impulse noise with noise density 50 % with m=30 and salt and pepper percentages differs as {10,40}, {20,30}, {30,20}, {40,10} (horizontally)

(b)-(e): corresponding filtered images of ASWM, BDND, SMF and PA for **NOISE MODEL 4**

The qualitative performances of the various algorithms of the literature is compared with the proposed approach extensively for “Lena” image. To depict that algorithm works well for other types of images also. Figure 5.7 represent the application of the algorithm to “baboon.tif”.



(a) **(b)** **(c)**

Figure 5.9 (a): Original images of Lena and baboon
(b): The corrupted images with noise density= 40%
(c) Filtered images using the proposed approach

5.3.3 Graphical Comparison

The algorithms can be graphical depicted to represent that proposed approach outperforms other existing algorithms for removing the impulse noises in terms of all the image metrics considered during simulation.

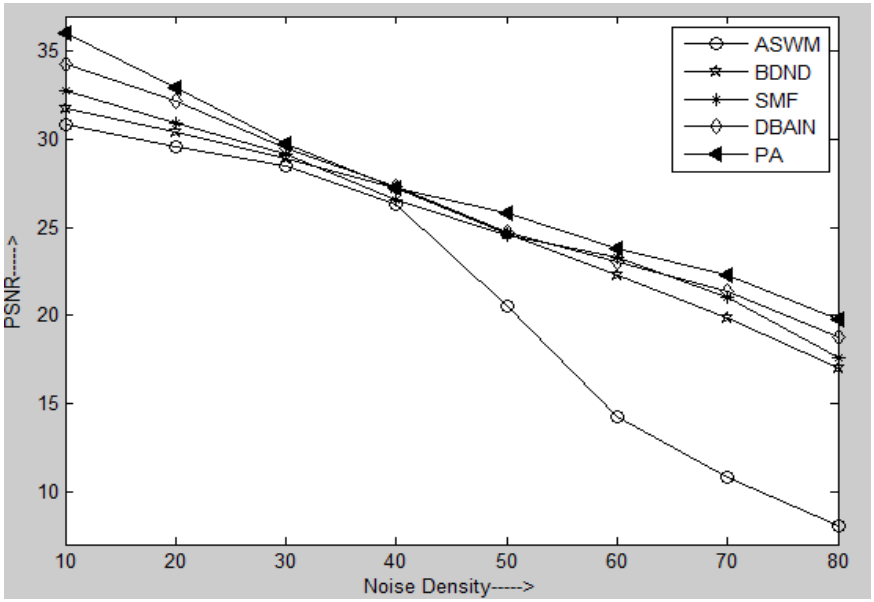


Figure 5.10 Noise Density versus PSNR value for **NOISE MODEL1** for "Lena"

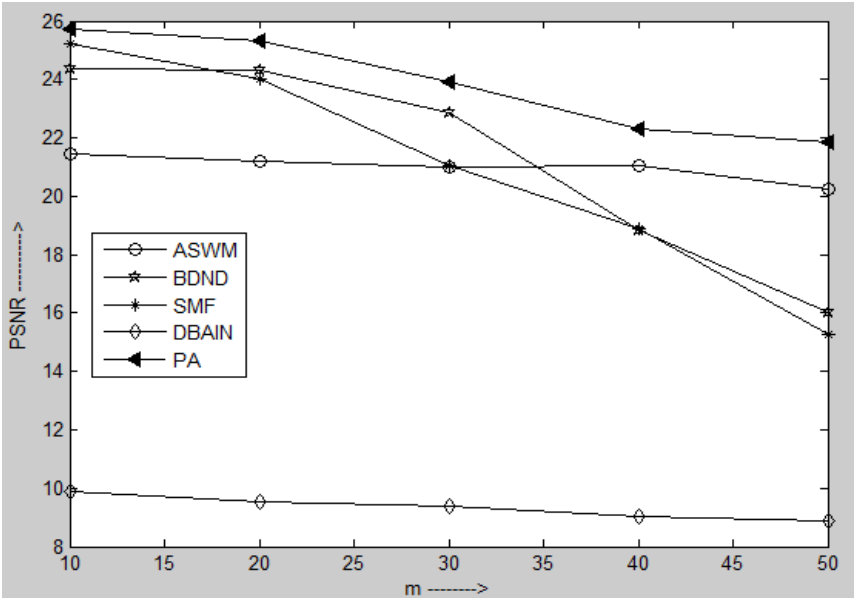


Figure 5.11 m versus PSNR value for **NOISE MODEL3** for "Lena"

5.4 Color Images

This section on color images show the quantitative, qualitative and graphical comparisons of proposed approach with its basis algorithm i.e. BDND. The results are obtained on color input images.

5.4.1 Quantitative Performance

It is well known that the human perception of the *color* differences mismatches the *numerical* differences yielded in the RGB color space. In other words, for color images PSNR value doesn't provide appropriate measure to compare the images, therefore, we need some metric to compute the color difference. In response to the same, the perceptually uniform color space CIELAB, standardized by the Commission Internationale de l'Eclairage (CIE), is more accurate for defining quantitative measurements of perceptual error between the two color vectors.

In CIELAB color space the color difference is calculated in terms of ΔE (Delta E) [formula is mentioned in chapter 2]. Here we will consider the mean value of ΔE ie. ΔE_{mean} to compare the images. Lower the value, more closer the image to the original one.

The following subsections give the tables for each noise models. The values are calculated extensively for the color image Lena.

5.4.1.1 Noise Model 1

Noise	BDND	PA
10%	0.9488	0.9786
20%	1.9685	1.8871
30%	3.3788	2.8696
40%	5.8785	4.0648
50%	9.2032	5.3586
60%	12.8314	6.5674
70%	16.2105	7.8972
80%	20.9704	9.3771

Table 5.6: ΔE obtained after applying BDND and PA on "Lena.tif" corrupted under various noise densities for noise model 1

5.4.1.2 Noise Model 2

Noise		BDND	PA
Salt	**Pepper**		
10	**30**	2.9214	2.9455
20	**20**	14.5764	4.3923
30	**10**	10.1354	3.1248

Table 5.7: ΔE obtained after applying BDND and PA on "Lena.tif" corrupted under noise model 2 for varying percentage of salt and pepper in impulse noise with Noise=40%

5.4.1.3 Noise Model 3

Noise		BDND	PA
Low	**High**		
[0,9]	**[246,255]**	6.3250	4.4945
[0,19]	**[236,255]**	6.6571	4.7845
[0,29]	**[226,255]**	6.8299	5.0938
[0,39]	**[216,255]**	7.1501	5.4772
[0,49]	**[206,255]**	7.7749	6.5822

Table 5.8: ΔE obtained after applying BDND and PA on "Lena.tif" corrupted under noise model 3 for varying m

5.4.1.4 Noise Model 4

Noise		BDND	PA
Salt	**Pepper**		
10	**30**	5.8425	3.7311
20	**20**	10.1033	5.0972
30	**10**	6.1509	3.7518

(a)

Noise		BDND	PA
Salt	**Pepper**		
10	**30**	6.5864	4.8603
20	**20**	11.8013	6.1863
30	**10**	7.2030	4.4028

(b)

Table 5.9: ΔE obtained after applying BDND and PA on "Lena.tif" corrupted under noise model 4 for varying percentage of salt and pepper in impulse noise with Noise=40% with (a) m=10; and (b) m=30

5.4.2 Qualitative Performance

The image outputs for each run are as shown below corresponding to the original image shown in fig 5.1 and 5.2.

Figure 5.12: The images are display as noisy image, filtered images of BDND and PA

(a) **Noise Model 1** : Noise=40% (b) **Noise Model 2**: Noise=40%, {30,10}

(c) **Noise Model 3**: Noise=40%, m=30 (d) **Noise Model 4**: Noise=40%, {30,10}, m=30

5.4.3 Graphical Comparison

The algorithms can be graphical depicted to represent that proposed approach outperforms other existing algorithms for removing the impulse noises in terms of all the image metrics considered during simulation.

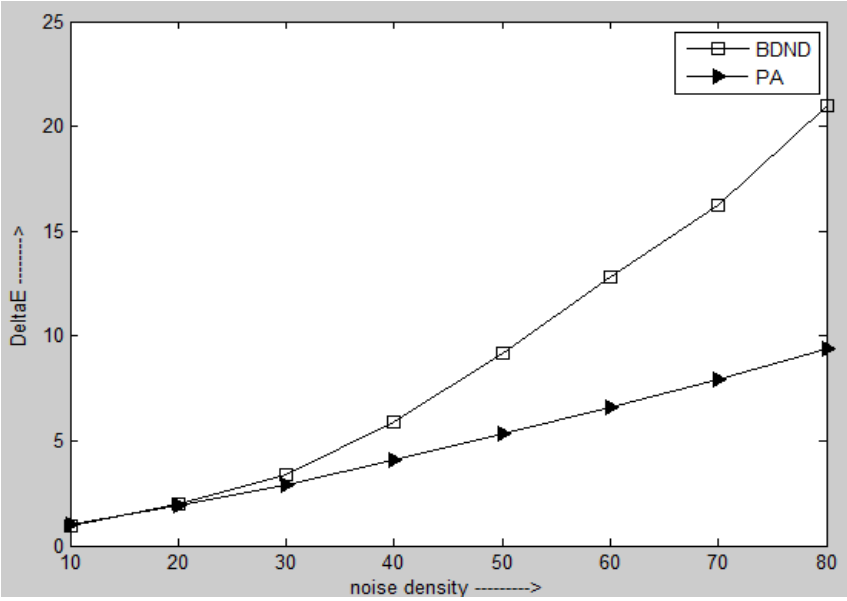


Figure 5.13 Noise Density versus ΔE for **NOISE MODEL1** for "Lena"

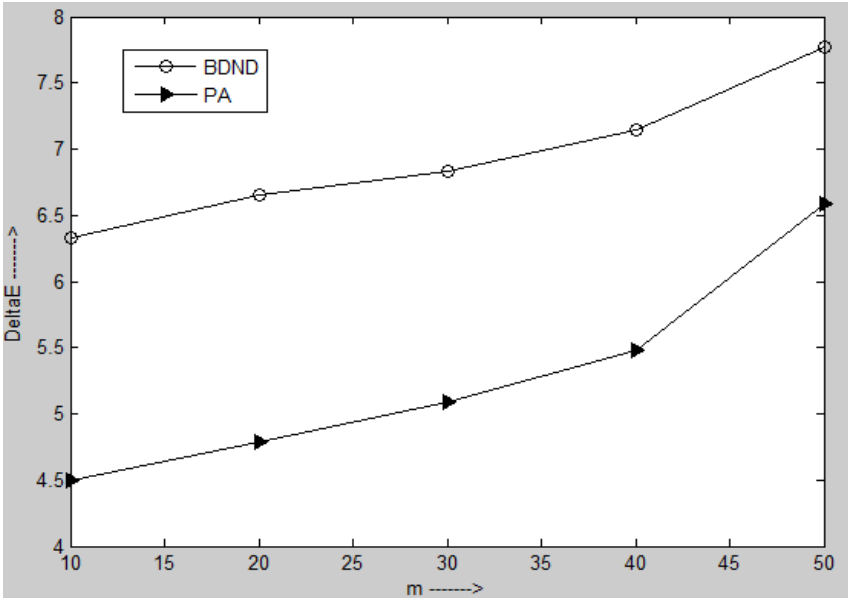


Figure 5.14 m versus ΔE for **NOISE MODEL3** for "Lena"

5.5 Discussion of Results

The filtered image ***F*** of each algorithm is compared with that of proposed approach. For experiments, gray scale version of "Lena" of size 256x256 is selected. All four types of noise models defined in [24] are used to compare the results to show the effectiveness of our algorithm in presence of any kind of impulse noise. To study the quantitative performance of algorithms, an input image is corrupted with impulse noise of different density under four noise models. The selection of window sizes for detection i.e. 21x21 for global checking and 5x5 for local checking is done by intensive experimentation. The approach is being compared with the different algorithms using different window sizes for their detection stage. Here the results are shown to compare the technique with ASWM (initial alpha=20, iterations=7), BDND (two stages of 21 x 21 and 3x3 with adaptive filtering), SMF (21x21 window with four directional kernels using the thresholds values as 5/255 and 1/255) and DBAIN (3x3 window). Table 5.1-5.5 shows the outputs for all the five algorithms implemented for four types of noise models. Fig 5.5-5.9 represents the qualitative performances of each and Fig. 5.10, 5.11 shows the performances graphically for noise model 1 and 3. As mentioned, for gray scale images, only stage I is applicable. The membership functions are formed adaptively according to the value of pixel in hand. That pixel is classified, after fuzzification, into a corrupted or uncorrupted class using a threshold value (eq.4.6). Experimentally the value of threshold for which the technique gives best results is 0.5.

Table 5.6-5.9 shows the comparison of BDND with PA in terms of color difference on color version of "Lena" (265x265). According to the definition of ΔE, the value of 1 means there is almost no perceptible differences or variations between two colors. The value ranging from 2-5 represents minute and 6-10 represents noticeable color differences or variations in high quality imaging systems. According to the values obtained for different noise models, our approach shows acceptable range as compared to BDND. Fig 5.13, 5.14 shows the comparison graphically. The incorporation of the second stage i.e. to add correction term to color components by exploiting the interactions between them proves to be beneficial. The parameters β_1 and β_2 used are found out to be 0.5 and 0.6 respectively experimentally.

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In this thesis, we have presented a fuzzy based approach to boundary discriminative noise detection along with an additional step to consider the interaction between the color components to remove any residue impulse noise present in the color components after the filtering stage. The proposed algorithm is compared with other existing algorithms like ASWM, SMF, BDND and DBAIN. We have separated the results of gray scale and color images. The quantitative, qualitative and graphical comparison in terms of PSNR, MSE and correlation coefficient and average value of ΔE proves that the proposed algorithm outperforms for all kinds of noise models. The best results obtained by the proposed techniques are for noise model 3 and 4 where random noise is used to corrupt the images. This is because the fuzzy rule based system avoids the general method of selecting strict thresholds to classify the pixels into clusters, rather it consider smooth boundaries between the classes. To mention the applicability of this approach, as it is able to filter out the random noise as shown by the figures present in the previous chapter, it can be easily and efficiently applied to real time images.

6.2 Future Scope

The thesis presented an algorithm which uses the fuzzy rule based approach to boundary discriminative noise detection and presents an impulse noise filter for color and gray scale images. The main focus of the thesis is to handle both SPN and RVIN only. The work can be extended to handle other type of noises like Gaussian noise or mixed types of noises.

REFERENCES:

[1] I.Pitas and A.N.Venetsanopoulos. *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academic Publishers, 1990.

[2] I Gonzalez, Rafael C., "Digital Image Processing", Addison-Wesley Pub (Sd); 3rd edition.

[3] J.Astola and P.Kuosmanen. *Fundamentals of Nonlinear Filtering*. CRC Press, 1997.

[4] I. Pitas and A. N. Venetsanopoulos, *"Order statistics in digital image processing,"* Proc. *IEEE*, vol. 80, no. 12, pp. 1893–1921, Dec. 1992.

[5] D. R. K. Brownrigg, *"The weighted median filter,"* Commun. *ACM*, vol. 27, no. 8, pp. 807–818, Aug. 1984.

[6] S.-J. Ko and Y. H. Lee, "*Center weighted median filters and their applications to image enhancement,"* IEEE Trans. Circuits Syst., vol. 38, no. 9, pp. 984–993, Sep. 1991.

[7] K. J. Overton and T. E. Weymouth, *"A noise reducing preprocessing algorithm*", Proceedings of the IEEE Computer Science Conference on Pattern Recognition and Image Processing, pp. 498-507, Chicago, IL, 1979.

[8] P. Perona and J. Malik, "*Scale-space and edge detection using anisotropic diffusion"*, IEEE Trans. Pattern Anal. Machine Intell., Vol. 12, pp. 629639, 1990.

[9] Z. Wang and D. Zhang, *"Progressive switching median filter for the removal of impulse noise from highly corrupted images,"* IEEE Trans. Circuits Syst. II, vol. 46, no. 1, pp. 78–80, Jan. 1999.

[10] S. Zhang and M. A. Karim, "*A new impulse detector for switching median filters,"* IEEE Signal Processing Letter, vol. 9, no. 4, pp. 360–363, Nov. 2002.

[11] H.-L. Eng and K.-K. Ma, *"Noise adaptive soft-switching median filter,"* IEEE Transaction Image Processing, vol. 10, no. 2, pp. 242–251, Feb. 2001.

[12] T. Sun and Y. Neuvo, *"Detail-preserving median based filters in image processing",* Pattern Recognition Letters. Vol. 15, pp. 341-347, Apr. 1994.

[13] W. Luo, "*An Efficient Detail-Preserving Approach for Removing Impulse Noise in Images",* IEEE Signal Processing Letters, Vol. 13, No. 7, July 2006.

[14] G. Pok, J. Liu and A. S. Nair, *"Selective removal of impulse noise based on homogeneity level information",* IEEE Trans. on Image Processing, Vol. 12, pp. 85-92, Jan. 2003.

[15] F. Russo, *"Noise cancellation using nonlinear fuzzy filters*", Proc. of IEEE Instrumentation and Measurement Technology Conference, Vol. 2, pp. 77-2777, 1997.

[16] F. Russo, *"A technique for image restoration based on recursive processing and error correction"*, Proc. of IEEE Instrumentation and Measurement Technology Conference, Vol. 3, pp. 1232- 1236, 2000.

[17] S. Md. Mansoor Roomi and T. Pandy Maheswari and V. Abhai Kumar, "*A Detail Preserving Filter for Impulse Noise Detection and Removal*", GVIP Special Issue on Denoising, pp. 35-40, 2007.

[18] P.-E. Ng and K.-K. Ma, *"A switching median filter with boundary discriminative noise detection for extremely corrupted images,"* IEEE Transaction Image Processing., vol. 15, no. 6, pp. 1506–1516, Jun. 2006.

[19] K.S. Srinivasan, D.Ebenezer, *"A New Fast and Efficient Decision Based Algorithm for removal of High Density Impulse Noises",* IEEE Signal Processing Letters, Volume 14, Issue 3, 2007, pg 189-192.

[20] Nair, Revathy, Tatavarti, *"An Improved Decision Based Algorithm for Impulse Noise Removal"*, Congress on Image and Signal Processing, 2008, Volume 1, pg 426-431.

[21] Madhu S. Nair, K. Revathy, Rao Tatavarti, *"Removal of Salt and Pepper Noise in Images: A New Decision Based AlgorIthm"*, International MultiConference of Engineers and Computer Scientists 2008, Volume I.

[22] Wei Li, Yanxia Sun, Shengjian Chen, *"A New Algorithm for removal of High Density Salt and Pepper Noises"*, 2nd International Congress on Image and Signal Processing, 2009, Pg 1-4.

[23] Smail Akkoul, Roger Ledee, Remy Leconge and Rachid Harba, *"A New Adaptive Switching Median Filter"*, IEEE Signal Processing Letters, 2010, pg 587-590.

[24] Fei Duan and Yu-Jin Zhang, *"A Highly Effective Impulse Noise Detection Algorithm for Switching Median Filter"*, IEEE Signal Processing Letters, Vol. 17, No. 7, July, 2010.

[25] Tripathi, Ghanekar, Mukhopadhyay, *"Switching Median Filter: advanced boundary discriminative noise detection algorithm",* Image Processing, IET, Volume 5 Issue 7, pg. 598-610, 2011.

[26] K. Arojawa, *"Median Filter based on Fuzzy Rules and its Application to Image Restoration"*, Fuzzy Sets and Systems, Vol. 77, pp. 3-13, 1996.

[27] K. Arojawa, Edited by E.E. Kerre and M. Nachtegael*, "Fuzzy Ruled-Based Image Processing with Optimization",* Springer- Verlag, pp. 222-247, 2000.

[28] F. Russo, "*FIRE operators for Image Processing*", Fuzzy Sets and Systems, Vol. 103, pp. 265-275, 1997.

[29] F. Russo, "*Noise Cancellation using Non-linear Fuzzy Filters*", Proc. of IEEE Instrumentation and Measurement Technology Conference, pp. 772-777, May 1997.

[30] F. Russo and G. Ramponi, *"A noise smoother using cascaded FIRE filters"*, Proc. of 4th Intl. Conf. on Fuzzy Systems, Vol. 1, pp. 351-358, 1995.

[31] J.Y. Jiu, *"Multilevel Median Filter based on Fuzzy Decision*", DSP IC E.E. NTU., 1996.

[32] Wang J. H. Yang, Chiu H.C., *"An Adaptive Fuzzy Filter for Restoring Highly Corrupted Image by Histogram Estimation"*, Proc. Natl. Sci. ROC(A), Vol. 23, No. 5, pp. 630-643, 1999.

[33] D. Androutsos, K.N. Plataniotis, and A.N. Venetsanopoulos, *"Colour image processing using vector rank filters"*, International Conf. on Digital Signal Processing, Vol. 2, pp. 614-619, 1995.

[34] L. Khriji, M. Gabbouj, "A Class of Multichannel Image Processing Filters", IEICE Trans. on Information and System, Vol. E82-D, No.12, pp.1589-1596, 1999.

[35] S. Schulte, V. De Witte, M. Nachtegael, D. Van der Weken, and E.E. Kerre, *"Fuzzy Two-Step Filter for Impulse Noise Reduction From Color Images",* IEEE Trans. on Image Processing, Vol. 15, No. 11, November 2006.

[36] Li Song, Wang Caizhu, Li Yequi, Wang Ling, Sakata, H. Sekiya, S. Kuroiwa, *"A fuzzy switching filter for removing impulse noise",* 11th IEEE International Conference on Communication Technology (ICCT), 2008, pg: 684-687.

[37] N.Z. Janah, B. Baharudin, *"Mixed Impulse Fuzzy Filter Based on MAD, ROAD and Genetic Algorithms"*, International Conference of Soft Computing and Pattern Recognition (SOCPAR), 2009, pg: 82-87.

[38] M.A. Soyturk, A. Basturk, M.E. Yuksel, "Rule based optimization of impulse noise remover fuzzy inference system", National Conference on Electrical, Electronics and Computer Engineering (ELECO), 2010, pg:613-616.

[39] K.K.V. Toh, N.A.M. Isa, *"Cluster-based adaptive fuzzy switching median filter for universal impulse noise reduction",* IEEE Transactions on Consumer Electronics, 2010, Volume 56, Issue 4, pg: 2560-2568.

[40] S.K. Meher, P. Patel, "*Fuzzy impulse detector for efficient image restoration"*, IEEE Conference on Recent Advances in Intelligent Computational Systems (RAICS), 2011, pg: 701-705.

[41] A. Hussain, Q. Javaid, M. Siddique, *"Impulse noise removal using fuzzy logic and alpha-trimmed mean"*, IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 2011, pg: 216-220.

[42] Madhu S. Nair, G. Raju, *"A new fuzzy-based decision algorithm for high-density impulse noise removal"*, Signal, Image and Video Processing, 2010 Springer-Verlag. DOI:0.1007/s11760-010-0186-4.

[43] S. Muthukumar, G.Raju, *"A Non-Linear Image Denoising Method for salt and pepper noise removal using fuzzy based approach",* International Conference on Image Information Processing, ICIIP 2011.

[44] Aborisade, D.O., "A Novel Fuzzy Logic Based Impulse Noise Filtering Technique", International Journal of Advanced Science and Technology, Vol. 32, July 2011, Pg. 79-88.

[45] Om Prakash Verma, Madasu Handmandlu, Anil Singh Parihar, Vamsi Krishna Madasu, *"Fuzzy Filters for Noise Reduction in Color Images"*, ICGST-GVIP Journal, Volume 9, Issue 5, September 2009, ISSN: 1687-398X. Pg 29-43.

[46] Recep Demirci, *"Rule-based automatic segmentation of color images",* AEU-International Journal of Electronics and Communications, Volume 60, Issue 6, 2006, pp 435- 442.

[47] Om Prakash Verma, *"Fuzzy Edge Detection Based Similarity Measure in Color Image"*, India Conference (INDICON), 2010 Annual IEEE.

[48] *"Fuzzy Image Processing Tutorial",* University of Waterloo.

[49] L.A. Zadeh, Fuzzy sets, *Information Control* 8 (1965) 338-353.

[50] J. Yen, R. Langari, *"Fuzzy logic: intelligence, control and information",* Prentice-Hall, New Jersey 1998.

[51] Werner Backhaus, Reinhold Kliegl, "Color Vision: Perspective from Different Disciplines", John Simon Werner, 1998.

[52] M. Tkalcic and J. F. Tasic, *"Color spaces: Perceptual, historical and application background",* in Proceeding s of IEEE EUROCON, vol. 1, Sep. 2003, pg 304–308.