IMPULSE NOISE REMOVAL IN COLOR IMAGES USING FUZZY LOGIC

A MAJOR PROJECT REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING
in
Electronics and Communication Engineering

By

BABITA

Under the supervision of

MR. O.P.VERMA

(ASSISTANT PROFESSOR,HOD I.T.)

Department of Electronics and Communication Engineering

Delhi College Of Engineering

2010

Delhi College of Engineering

NEW DELHI

CERTIFICATE

This is to certify that the report entitled, "**IMPULSE NOISE REMOVAL IN COLOR IMAGES USING FUZZY LOGIC "** submitted by **Ms. BABITA** in partial fulfillment of the requirements for the award of Master of Engineering Degree in **ELECTRONICS AND CMMUNICATION** at Delhi College Of Engineering (University Of Delhi) is an authentic work carried out by her under my supervision and guidance. To the best of my/our knowledge, the matter embodied in the thesis has not been submitted to any other University/ Institute for the award of any degree or diploma.

(PROJECT GUIDE)

MR. O.P.VERM
ASSISTANT PROFESSOR
HEAD OF THE I.T. DEPARTMENT
DELHI COLLEGE OF ENGINEERING
NEW DELHI-110042

PROF. RAJIV KAPOOR
HEAD OF THE ELECTRONICS &
COMMUNICATION DEPARTMENT
DELHI COLLEGE OF ENGINEERING
NEW DELHI-110042

ACKNOWLEDGEMENT

I express my sincere gratitude and indebtedness to the guide **MR. O.P.VERMA,** for their initiative in this field of research, for his valuable guidance, encouragement and affection for the successful completion of this work. Their sincere sympathies and kind attitude always encouraged me to carry out the present work firmly. I express my thankfulness to **Prof. RAJIV KAPOOR, Head of the Department** of Electronics and Communication Engineering, Delhi college of Engg. New Delhi, for providing me with best facilities in the Department and his timely suggestions. I would like to thank Matlab Software for providing the simulation package.

At last but not least I would like to thank all teaching and non-teaching staff of DCE, and all my friends who were involved directly or indirectly in successful completion of the present work.

BABITA

(M.E. ELECTRONICS & COMMUNICATION ENGG.)

COLLEGE ROLL NO. 22/E&C/07

UNIVERSITY ROLL NO. 8803

DELHI COLLEGE OF ENGG.

ABSTRACT

In this project impulse noise reduction method for color images is presented. Color images that are corrupted with impulse noise are generally filtered by applying a gray scale algorithm on each color component separately or using a vector-based approach where each pixel is considered as a single vector. The first approach causes artifacts especially on edge and texture pixels. Vector-based method were successfully introduced to overcome this performance while much less artifacts are introduced. The main difference between the proposed method and other classical noise reduction methods is that the correlated color information id taken into account to develop

a) A better impulse noise detection method.
b) A noise reduction method that filters only the corrupted pixels while preserving the color and the edge sharpness.

Experimental results show that the proposed problem. Nevertheless, they tend to cluster the noise and to receive a lower noise reduction performance. In this project, we discuss an alternative technique which gives a good noise reduction method provides a significant improvement on other existing filters.

CONTENTS

ABSTRACT

In this work, a two step fuzzy filters is presented to remove the impulse in the color image. For removal of the noise the R,G,B color space is chosen. In the proposed method first impulse noise is detected by the method based on the following two assumptions

This nonlinear filtering technique contains two separate steps:

1) impulse noise detection &

2) noise removal using fuzzy technique while preserving the fine details. Based on the concept of Laplacian operator method & median method, impulse noise is detected. The detected noise partially removed by switching action than further improved by application of fuzzy membership function. Experimental results show that the two step fuzzy filter provides a significant improvement on other existing filters available in the literature.

Chapter 1

Introduction

1.1 Introduction to Image Processing

Image Processing is a technique in which the data from an image are digitized and computed. The Image processing includes digital image processing, scanning, indexing images, digital edition image processing, image assessment service using the latest technology. Our Image processing team provides you with high quality, accurate, and reliable service. Utilizing our most modern digital image processing software and experienced professionals, we provide you with proven technology solutions. Our scanning and imaging experts provides services to both large scale as well as small scale image processing projects that require quick data processing turnaround and high quality deliverables.

Digital Image processing is a rapidly growing area of computer science & engineering. Processing of Digital image by means of a digital computer is known as digital image processing. An image may be defined as a two dimensional function, f(x, y), where x and y are spatial coordinates. These elements are called pixels and the amplitude of a pixel is the intensity or gray level of the image at that pixel in the gray image. Each pixel is a scalar proportional to the brightness. The minimum brightness is called black, and the maximum brightness is called white. A typical example is given in Figure 1.1. A color image measures the intensity and chrominance of light.

Figure 1.1: A typical grayscale image of Lena of resolution 256 ×256

For storage purposes, pixel values need to be quantized. The brightness in grayscale images is usually quantized to L levels so; $f(x, y) \in \{0, 1, -----, L-1\}$. If quantization level L has the form of 2^L the image is referred to as having L bits per pixel. Many common grayscale images use 8 bits per pixel, giving 256 distinct grey levels. In medical applications often use 12–16 bits per pixel, as their accuracy could be critically important.

1.2 Fuzzy Logic in Image Processing

1.2.1 Introduction to fuzzy logic

Fuzzy logic was initiated in 1965 by Lotfi A. Zadeh, professor for computer science at the University of California in Berkelay.

Basically, fuzzy logic is a multivalued logic that in a narrow sense, fuzzy logic refers to a logical system that generalizes classical two valued logical for reasoning under uncertainty. In a broad sense, fuzzy logic refers to all of the theories and technologies that employ fuzzy sets, which are classes with unsharp boundaries. Notions like rather tall or very fast can be formulated mathematically and processed by computer, in order to apply a more human-like way of thinking in the programming of computers.[2]

Fuzzy system is an alternative to traditional notion of set membership and logic that has its origin in ancient Greek philosophy. Fuzzy logic has emerged as a profitable tool for the controlling and steering of the systems and complex industrial processes as well as for household and entertainment electronics, as well as other expert systems and applications.

Fuzzy Fundamentals

Fuzzy logic technique is based on four basic concepts:

(1) Fuzzy Sets	:	Sets which smooth boundaries.
(2) Linguistic variables	:	Variables whose values are both qualitatively and quantitatively described by a fuzzy set.
(3) Possibility distributions	:	Constraints on the value of a linguistic variable imposed

by assigning it a fuzzy set.

(4) Fuzzy-if-then rules : A knowledge representation scheme for describing functional mapping or logical formula.

Example

Fuzzy set theory 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 as a means 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 elements 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 members 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 Figure 1.2)

Fig. 1.2 Representative of “dark gray-levels” with a crisp and a fuzzy set

1.2.3 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.

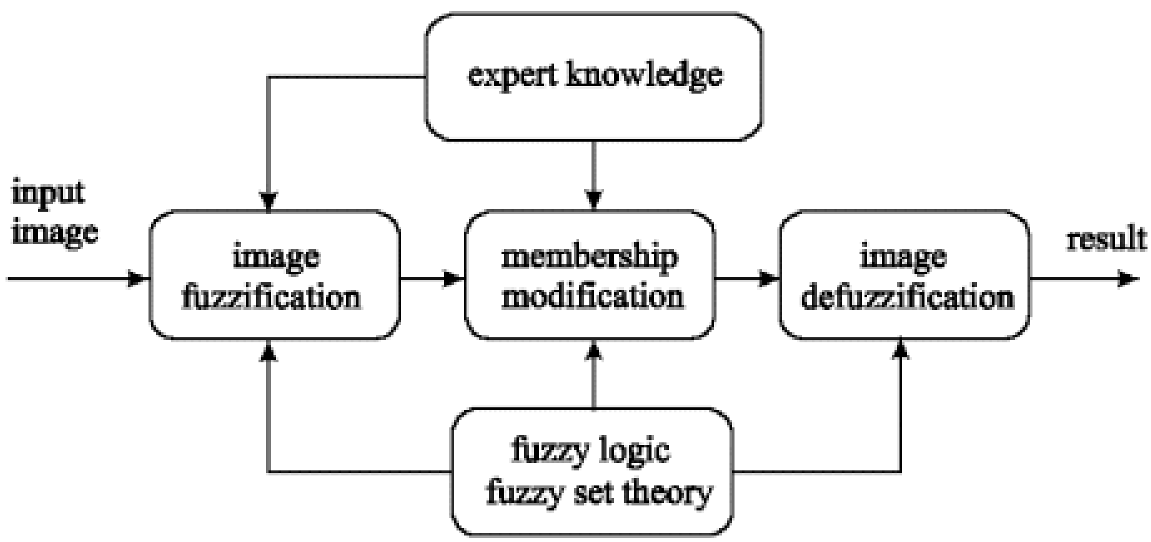

Fig 1.3 The general structure of fuzzy image processing.

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 (fuzzification) 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). 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. These steps are shown in Figure 1.3.

Fuzzification

The fuzzification comprises the process of transforming crisp values into grades of membership for linguistic terms of fuzzy sets. The membership function is used to associate a grade to each linguistic term.

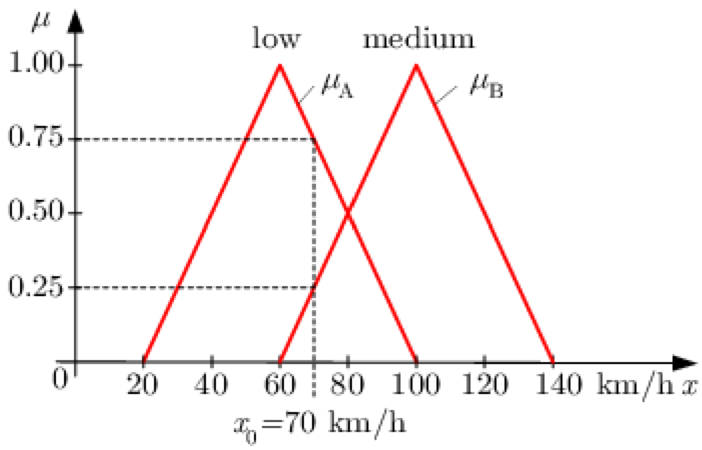

Figure 1.4: Fuzzification of a car speed

Example For the fuzzification of the car speed value $x_0 = 70\,\mathrm{km/h}$ the two membership functions μ_A and μ_B from Figure 1.4 can be used, which characterise a low and a medium speed fuzzy set, respectively. The given speed value of $x_0 = 70\,\mathrm{km/h}$ belongs with a grade of $\mu_A(x_0) = 0.75$ to the fuzzy set ``low'' and with a grade of $\mu_B(x_0) = 0.25$ to the fuzzy set ``medium''.

Defuzzification

Once the rules have been composed the solution, as has been seen, is a fuzzy set. However, for most applications (in particular in control) there is a need for a single action or `crisp' solution to emanate from the inferencing process. This will involve the `defuzzification' of the solution set. There are various techniques available. Lee [16]describes the three main approaches as the max criterion, mean of maximum and the centre of area. The *max criterion* method finds the point at which the membership function is a maximum.

The *mean of maximum* takes the mean of those points where the membership function is at a maximum. The most common method is the *centre of area* method which finds the centre of gravity of the solution fuzzy sets. For a discrete fuzzy set this is

``Unfortunately, there is no systematic procedure for choosing a defuzzification strategy.''. Although the process of reducing the final fuzzy set to a crisp value does seem appropriate for control problems much information is lost by doing this and further work needs to be done on how to use the information available in the solution fuzzy set.

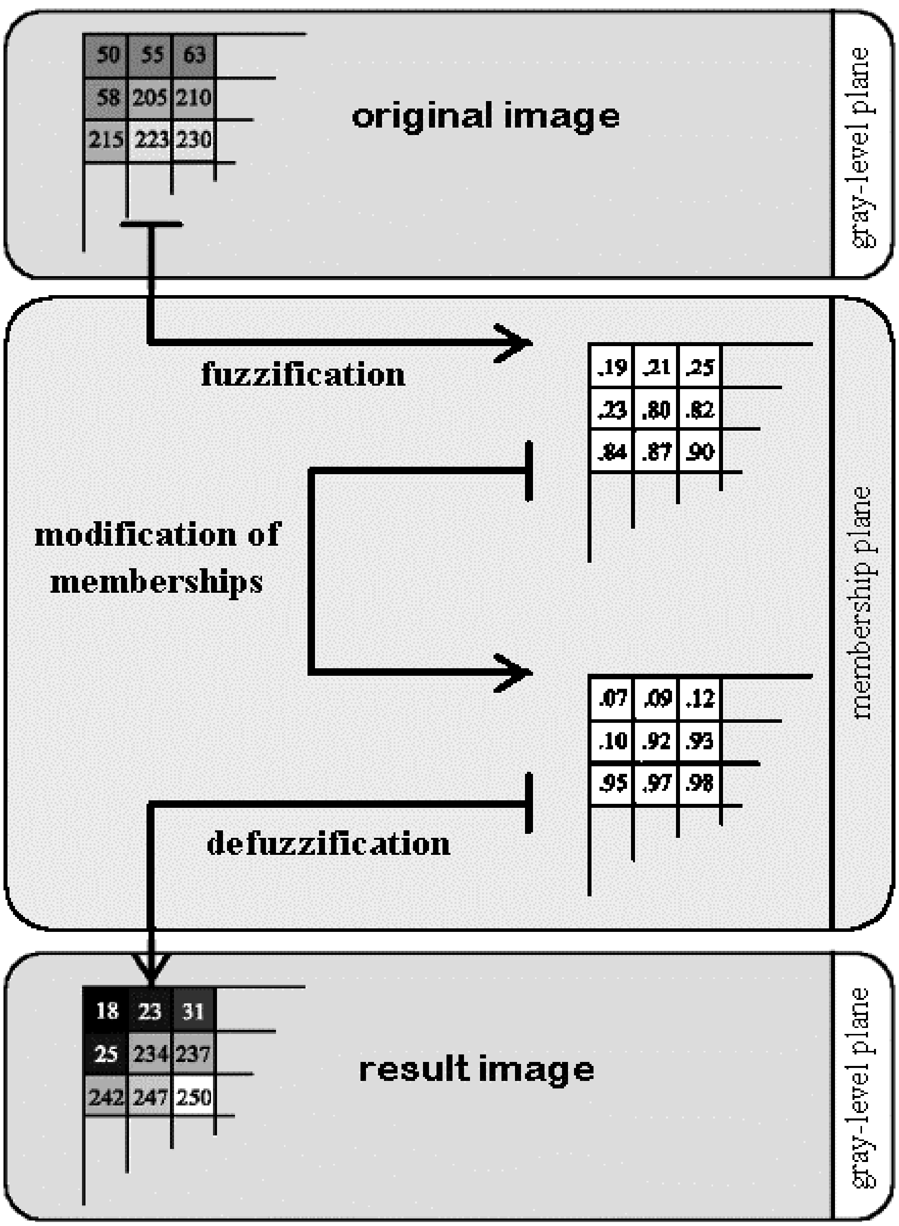

Fig.1.4 Steps of fuzzy image processing

1.2.4 Advantages of Fuzzy Image Processing

We use Fuzzy techniques in image processing. There are many reasons to do this. The most important of them are as follows:

Fuzzy techniques are powerful tools for knowledge representation and processing.

Fuzzy techniques can manage the vagueness and ambiguity efficiently.

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 represented in fig(1.5)

- 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 images 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.

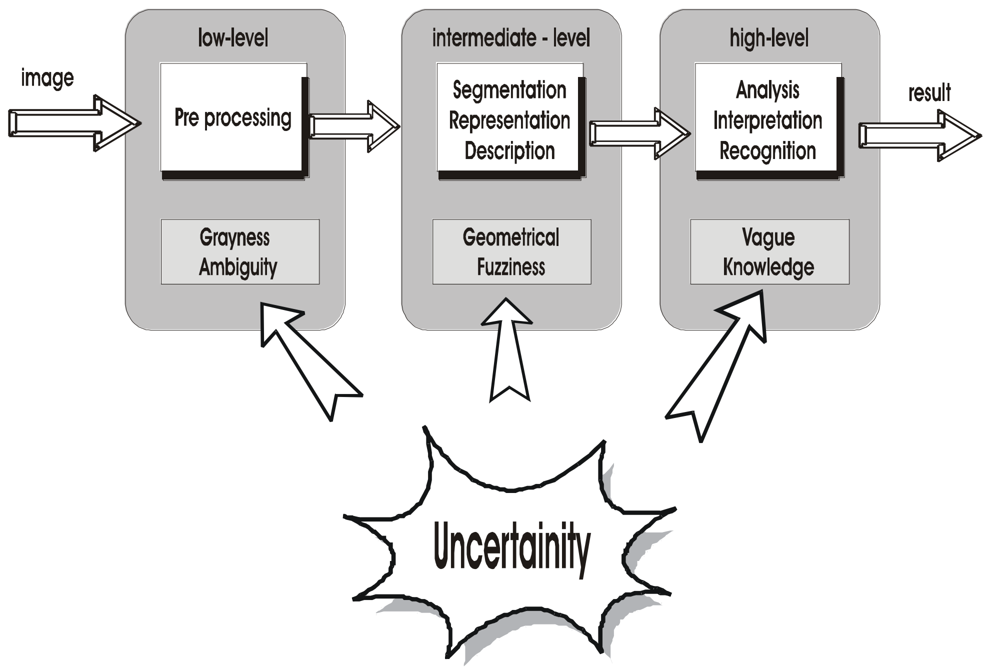

Fig.1.5 Uncertainty / imperfect knowledge in image processing.

1.2.5 Classification of Fuzzy Filters

The fuzzy logic in image processing is discussed as above . For a meaningful classification of fuzzy filter we will consider on the following criteria:

Noise Type : for which noise type is the filter designed.

Degree of fuzziness : whether the filter a pure fizzy filter or is it rather a modification of one or more classical filters.

Fuzzy rules : which criteria are used by the fuzzy rules to determine whether a pixel is a noise pixel or not .

Algorithm : whether the algorithm iterative or recursive?

Complexity : how complex is the construction.

1.3 Noise reduction

Noise, discussed in the next chapter, reduction is an important step in many image processing applications. The image may easily corrupt during storage, scanning, transmission etc. Corruption of image by impulse noise is a frequent problem in image processing. It is necessary to remove noise from the image before some subsequent processing such as edge detection, image segmentation and pattern reorganization. Non linear techniques for noise reduction are very effective due to their specific ability to perform an effective nose cancellation without degrading the image structure.

A large number of filters have been proposed to remove impulse noise while preserving image details. In the literature median based filter and its modification is used widely because of their effective noise suppressive capability and their simplicity. However most of the median filters are implemented uniformly across the image, they tend to modify both noise pixel and undisturbed good pixels.

The basic principle of noise reduction is replacing the gray level of every pixel with a new value depending on the local window. The filtering action should distinguish between unwanted noise and image information. Hence ideal algorithm for noise cancellation should vary from pixel to pixel depending upon local context. For example if window region is free from noise, then average of window values must be taken for the new pixel value. On the other hand if window region contains edge or non impulse noise pixel, different type of algorithm should be used to preserve image information but it is very difficult.

Fuzzy filters have been suggested as on means of solving above problem It succeeded in removing impulse noise while preserving image detail such as edge. For example consider the following fuzzy rule for impulse noise detection.

If pixel value in the region is large.

Then pixel is impulse noise pixel.

The antecedents of fuzzy rules are constructed by using two important local characteristics.

Improvement in the central pixel by applying fuzzy rule / other equation.

Apply fuzzy membership function.

Since the fuzzy filter is able to change its property according to local characteristics in given window, it can remove Impulse noise while preserving image details.

Two important features as presented above estimates a Improvement in the central pixel which is sensitive to local variation due to image structure such as edges; and then apply the membership function accordingly to the noise level to perform as “fuzzy filter” for noise reduction. These rules examine each pixel that is processed and determine a correlated term.

The evaluation of the performance of classical and fuzzy filters for impulse noise will be supported by numerical and visual experiments. Among other things, we will compare whether fuzzy filters perform better than classical filter. We will also compare other available fuzzy filters and whether good numerical results are also confirmed by good visual results.

Since most effective approaches of Impulse noise reduction are non linear and adaptive nature. We are trying to develop fuzzy filter for impulse noise reduction in images.

The fuzzy filter removes ‘salt and pepper’ and ‘random’, noise from the image. This filter suppresses impulse noise effectively while preserving details of the image information. Here depending on the fuzzy rule, we use S-function to remove noise by taking the parameter “α”,”β,”and “γ”. The parameters are evaluated from the trial and error procedure from image depending upon image information.

Chapter 2
NOISE

2.1. Introduction

Generally, noise in signal processing is interpreted as ‘unwanted signals’. However, in the context of image processing, noise is termed as the displacement of the signal intensities from their original values. A fundamental problem of image analysis is to effectively remove noise from an image while keeping its fundamental structure constituting of edges, corners, etc., intact. The nature of the noise removal problem depends on the type of the noise corrupting the image. The two most commonly occurring types of noise are: Additive noise (e.g. Gaussian and Impulse noise) and Multiplicative noise (e.g. Speckle noise).

In this chapter, the degradation process modeled, which included noises such as uniform noise, salt and pepper noise and Gaussian noise etc. Figure 2.1 shows how these types of noise affect a typical grayscale image.

Figure 2.1: Different types of noise: (a) original image; (b) additive noise; (c) multiplicative noise; (d) impulse noise.

2.2 Noise Model

2.2.1 Additive noise

Let $g(x,y)$ be the noisy image of the ideal image, $f(x,y)$, and $n(x,y)$ be a “noise function” which gives random values coming from an arbitrary distribution. Then additive noise can be described by Equation 2.1.

$$g(x,y) = f(x,y) + n(x,y) \tag{2.1}$$

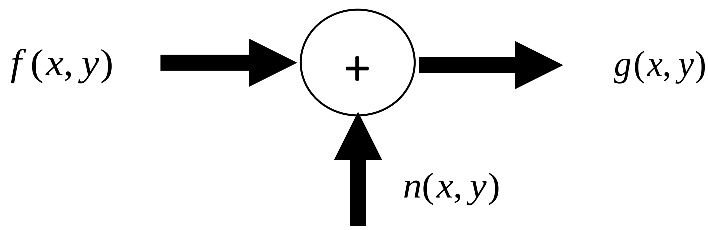

Figure 2.2 shows the degradation of image by additive noise

Additive noise is independent of the pixel values in the original image. Typically, additive noise $n(x,y)$ is symmetric about zero. This has the effect of not altering the average brightness of the image, or large parts thereof. A good example of additive noise is the thermal noise within photo-electronic sensors.

Here level of additive noise generally express by its variance. Given $g(x,y)$ and some knowledge about the additive noise term, the objective of restoration is to obtain an estimate $f(x,y)$ of the original image; which to be as close as possible to the original input image and it require more we know about $n(x,y)$. In order to compare the performance with respect to original input image, the signal to noise ratio is often used for the characterization of signal.

$$SNR \text{ in } d\text{B} = 10\log_{10} \frac{\sigma_f^2}{\sigma_n^2} \tag{2.2}$$

Where σ_f = variance in the signal

σ_n = variance in the noise

The mean square error (MSE) between the original image $f(x, y)$ and the processed image $\overline{f}(x, y)$ is given by:

$$MSE\left[f(x, y), p(x, y)\right] = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[f(x, y) - \overline{f}(x, y)]^2}{M \times N} \tag{2.3}$$

Where N,M = image dimensions

In practice, images easily get corrupted with noise, e.g. due to the circumstances of recording for example digital image acquired with digital electronic camera corrupted due to dust on a lens, electronic noise in the cameras and sensors etc., transmission (e.g. electromagnetic interaction, transmission over channel etc.) for example video images transmitted via satellite are susceptible to the electromagnetic interference. Also the digital image also easily corrupted during storage on secondary memory, during copying and scanning etc.

The noise component may be considered random variables characterized by a probability density function (PDF). The mean and variance are useful parameters to represent noise. Mean value M gives the average brightness of the noise and square root of variance σ gives the average peak-to-peak gray level deviation of noise. μ (mean) and σ (variance) are given by:-

$$\mu = \sum_{k=0}^{g_{\max}} k.p_n(k) \tag{2.4}$$

$$\sigma^2 = \sum_{k=0}^{g_{\max}} \left[k.p_n(k) - \mu\right]^2 \tag{2. 5}$$

Where $p_n(k)$ = frequency of occurrence of noise amplitude k

$g_{\max}$ = maximum gray level

Where k, ideally varies from – ∞ to ∞. Since pixel level are limited to range (0, N–1) the noise amplitude level k also varies to range (0, N–1).

2.2.2 Multiplicative noise

A signal dependent form of noise, whose magnitude is related to the value of the original pixel is know as Multiplicative noise, or speckle noise, and Equation 2.6 describes one simple form it can take, but a more complex function of the original pixel value is also possible.

$$g(x,y) = f(x,y) + n(x,y)f(x,y) = f(x,y)[1+n(x,y)] \qquad (2.6)$$

2.2.3 Impulsive noise

The PDF of impulsive noise is given by:

$$p(k) = \begin{cases} p_a & \text{for } z = a \\ p_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \qquad (2.7)$$

Where p_a, is the probability of salt is noise and p_b is the probability of pepper noise

Impulse noise has the property of either leaving a pixel unmodified with probability 1-p or replacing it altogether with probability p.

If b > a, gray level b will appear as a light dot in the image. While level a will appear like dark dot. If either p_a or p_b is zero, the impulse noise is called unipolar. If neither probability is zero, especially if a and b are approximate equal, impulse noise will resemble salt and pepper granules randomly distributed over the image. For this reason, bipolar noise is also called salt and pepper noise. Other name of impulse noise is shot and spike noise. Noise impulse can be positive or negative.

2.2.4 Gaussian noise

Gaussian noise models are used frequently in practice due its mathematical tractability in both the spatial and frequency domains.

The PDF of a Gaussian random variable k is given by:

$$p_n(k) = \frac{1}{\sigma\sqrt{2\pi}} \; e^{\frac{-(k-\mu)^2}{2\sigma^2}} \tag{2.8}$$

Where k = represent gray value.

μ = mean average value of k

σ = standard deviation and its square σ^2 is called the variance of k

When k is described by above equation approximate 70% of its values will be in the range [μ–σ], [μ +σ)], and approximate 95% will be in the range [μ – 2σ), μ + 2σ)]

2.2.5 Uniform noise

The PDF of uniform noise with equal probability in the range from g_{max} to g_{min} is given by:-

$$p_n(k) = \begin{cases} \dfrac{1}{g_{\max} - g_{\min}} & \text{if} \quad g_{\min} \le k \le g_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

The mean and standard deviation can be computed as:

$$\mu = \frac{g_{\min} + g_{\max}}{2}$$

$$\sigma^2 = \frac{(g_{\min} - g_{\max})^2}{12}$$

2.2.6 Rayleigh noise

The PDF of Rayleigh noise is given by:

$$p(k) = \begin{cases} \frac{2}{g_{\max}}(k - g_{\min})e^{-(k-g_{\min})^2 / g_{\max}} & for\ k \geq g_{\min} \\ 0 & for\ k \langle\ g_{\min}\ . \end{cases}$$

(2.10)

The mean and variance of this density are given by:

$$\mu = g_{\min} + \sqrt{\frac{\pi\, g_{\max}}{4}}$$

$$\sigma^2 = \frac{g_{\max}(4 - \mu)}{4}$$

Rayleigh density can be quite useful for approximation of skewed histograms.

2.2.7 Enlong (Gamma) noise

The PDF of England Noise is given by :

$$p(k) = \begin{cases} \frac{g_{\min}^{\ g_{\max}} \times k^{g_{\max}-1}}{(g_{\max} - 1)!} & for\ k \geq 0 \\ 0 & for\ k < 0 \end{cases}$$

(2.11)

Where parameters are such that $g_{min} > 0$, g_{max} is a positive integer and mean and variance of this density are given by :

$$\mu = \frac{g_{\max}}{g_{\min}}$$

and

$$\sigma^2 = \frac{g_{\max}}{g^2_{\min}}$$

Although above equation is referred to as the gamma density. Strictly this is correct only when the denomination is gamma function.

2.2.8 Exponential noise

The PDF of exponential noise is given by :

$$p(k) = \begin{cases} g_{\min} e^{-g \min k} & for k \geq 0 \\ 0 & for k < 0 \end{cases} \qquad (2.12)$$

Where $g_{min} > 0$

The mean and variance of the Exponential noise function are :

$$\mu = \frac{1}{g_{\min}}$$

and

$$\sigma^2 = \frac{1}{g^2{}_{\min}}$$

2.2.9 Periodic noise

Electrical or electromechanical interference during image acquisition in an image is known as periodic noise. This type of noise can be reduced significantly via frequency domain filtering.

2.2.10 Quantization noise

Noise due to quantization of pixel values during the analog to digital conversion is known as quantization noise. For example, consider an analog image with brightness values ranging from 0 to 10. If it is quantized to accuracy 0.1, the digitized image will have 101 distinct grey levels. A given intensity could have originally been anywhere in the range ± 0.05. Quantization noise is this uncertainty in the true value of intensity.

Chapter 3

Impulse Noise Removal Using Fuzzy Logics

3.1 Introduction

The focus of this work is on Impulse noise removal. Image filters exist in three domains: spatial, frequency and fuzzy domain. This study deals with fuzzy filters which offer several advantages over classical filters even as they preserve the image structure. Moreover, fuzzy filters are easy to realize by means of simple fuzzy rules that characterize a particular noise. A brief review of well known fuzzy filters is presented in the following paragraphs. In the field of image noise reduction, several linear and nonlinear filtering methods have been proposed. Linear filters are not able to effectively eliminate impulse noise as they have a tendency to blur the edges of an image. On the other hand, nonlinear filters like median filters are better suited for dealing with impulse 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 is a modification to the classical median filter. We have used fuzzy operator that attempts to cancel the noise while preserving the image structure.

Noise reduction from images is an important step in many image-processing applications. In the presence of noise, filtering of noisy pixel and its correction is an important pre- processing step if they are to be followed by other tasks such as edge detection, feature extraction and object / pattern recognition. The most effective approaches of noise reduction are non-linear and adaptive in nature. The ability of the filter unwanted impulse noise and the random noise while preserving edges and details of an image is a non trivial task. Various nonlinear filters based on classical techniques and fuzzy techniques have emerged in the few years for this challenging task.

Since filtering a region might destroy an edge and sharpening edges might lead to unnecessary noise. The type of algorithm to be used depends upon the objective to be achieved by the filtering process as well as the particular application. The objective of this paper is development of fuzzy filter, which is adaptive in nature and able to remove salt and pepper noise.

Most of the classical filters that remove noise simultaneously also blur the edges, but fuzzy filters have the ability to combine preservation of image detail while filtering. Compared to other non-linear techniques, fuzzy filter are able to represent knowledge in a comprehensible way.

The two step fuzzy filters have been shown to be faster and yet more effective than uniformly applied methods. This nonlinear filtering technique contains two separated steps: an impulse noise detection step and a reduction step that preserve edge sharpness. Based on the concept of Laplacian operator method[6,7], impulse noise detected. The detected noise partially removed by switching action[7] than further improvement can be achieved by application of fuzzy membership function[6]. Experimental results show that the two step fuzzy filter provides a significant improvement on other existing filters available in the literature. This filter is not only very fast, but also very effective for reducing little as well as very high impulse noise.

3.1 Prosposed Fuzzy Two Step Filter

Images corrupted with impulse noise contain pixels affected by some probability. This implies that some of the pixels may not have a trace of any noise at all. Moreover, a pixel can have either all or one or two of its R,G,B components corrupted with impulse noise. Mathematical modeling of impulse noise in color images is as follows

$$f(x, y, z) \;\; as\ original\ image\ where\ z = 1,2,3 \qquad ..3.1$$

$$\bar{f}(x, y, z) \;\; as\ noisy\ image\ where\ z = 1,2,3 \qquad ..3.2$$

where 1, 2 & 3 represents red, green & blue components .

In the proposed method first impulse noise has been detected by the method based on the following two assumptions: A noise-free image consists of locally smoothly varying areas separated by edges and a noise pixel takes a gray value substantially larger or smaller than those of its neighbors.

3.1.1 The two step fuzzy filters

The two step fuzzy filters have been shown to be faster and yet more effective than uniformly applied methods. This nonlinear filtering technique contains two separated steps: an impulse noise detection step and a reduction step that preserve edge sharpness. Based on the concept of Laplacian operator method, impulse noise has been detected. The detected noise partially removed by switching action than further improvement can be achieved by application of fuzzy membership function. Experimental results show that the two step fuzzy filter provides a significant improvement on other existing filters available in the literature. This filter is not only very fast, but also very effective for reducing little as well as very high impulse noise.

First step: Impulse noise Detection

Let $f(x, y)$ is the pixel value at center position (x, y) in the window of size 5×5 .

The input image is first convolved with a set four one-dimensional Laplacian operators as shown in fig.3.1 to detect the impulse noisy pixels

0	0	0	0	–1
0	0	0	–1	0
0	0	4	0	0

0	–1	0	0	0
–1	0	0	0	0

0	0	–1	0	0
0	0	–1	0	0
0	0	4	0	0
0	0	–1	0	0
0	0	–1	0	0

–1	0	0	0	0
0	–1	0	0	0
0	0	4	0	0
0	0	0	–1	0
0	0	0	0	–1

0	0	0	0	0
0	0	0	0	0
–1	–1	4	–1	–1
0	0	0	0	0
0	0	0	0	0

Fig..3.1 Four 5 x 5 convolutions Laplacian kernels

each of which is sensitive to edges in a different orientation. Then, the minimum absolute value of these four convolutions (denoted as $w(x,y)$) is used for impulse detection, which can be represented as:

$$w(x,y,1) = \min\{\left| f(x,y) \quad \otimes K_t \right| : t = 1 \text{ to } 4 \qquad \ldots (3.3)$$

Where K_t is the pth kernel, and $\otimes$ denotes a convolution operation.

The value of $w(x,y)$ detects the impulse noise due to the following reasons:

(1) $w(x,y)$ is large when the current pixel is an isolated impulse because the four convolutions are large and almost the same.

(2) $w(x,y)$ is small when the current pixel is a noise-free, flat region pixel because the four convolutions are close to zero.

(3) $w(x, y)$ is small also when the current pixel is an edge (including thin line) pixel because one of the convolutions is very small (close to zero) along with the other three might be large.

Median calculation

An algorithm to determine the median of noise-free pixels in the neighborhood of a pixel under interest is now presented. The median of the noise free pixels is utilized to modify the pixel corrupted with impulse noise. This median is computed separately for each color component in the following steps:

Step1: Take a window of size $w \times w$ centered on the pixel of interest in the corrupted image. The window of 5X5 has been taken from the noisy image.

$$U(x, y, 1) = \bar{f}(x + p, y + q) \quad where\ p = -2\ to\ 2\ \&\ q = -2\ to\ 2 \qquad ...(3.4)$$

Step2: Arrange all the pixels of the window as a vector. Sort the vector in an increasing order and compute the median of the sorted vector.

Step3: Calculate the difference between each window pixel and the median of the vector.

$$D(x, y, 1) = \bar{f}\ (x, y, 1) - Med(x, y, 1) \qquad .(3.5)$$

Step4: Arrange all the window pixels having the differences less than or equal to a parameter T in a vector.

if $D(x, y, 1) > T$ then the pixel is noisy and if $D(x, y, 1) \leq T$ then the pixel is noise free, the threshold value has been calculated after simulation[1].

Step5: Sort the new vector and obtain the median *med* of the sorted vector then this median value have been used to get improved image, the equation (3.7).

The improvement of the current pixel without fuzzy logic

From the above analysis $w(x, y)$ is large when $f(x, y)$ is corrupted by noise, and $w(x, y)$ is small when $f(x, y)$ is noise-free whether or not it is a flat-region, edge, or thin-line pixel. So, we can compare $w(x, y)$ with a threshold, T, to determine whether a pixel is corrupted[1], i.e.

$$\delta(x, y) = \begin{cases} 1, & w(x, y) > T \\ 0, & w(x, y) \leq T \end{cases} \qquad (3.6)$$

Obviously, the threshold T affects the performance of impulse detection. It is not easy to derive an optimal threshold through analytical formulation. But we can determine a reasonable threshold using computer simulations.

If $\delta(x, y) = 1$, then the current pixel is noisy pixel otherwise its noise free

Then impulse pixel has been isolated to get the noisy pixel and the determined value of $\delta(x, y)$ has been use in the improved current pixel equation given below.

$$E(x, y) = \delta(x, y) * Med(x, y) + [1 - \delta(x, y)] * \bar{f}(x, y) \quad .. \quad (3.7)$$

Second Step: Impulse Noise removing through Fuzzy Logic

The impulse noise is reduced through Fuzzy logic. The membership function has been defined

$$\mu(x, y, 1) \in \ (1,0)$$

is the membership function of $w(x, y)$ that indicates how much a pixel corrupted with an impulse noise, we can give the following fuzzy rules.

Fuzzyfication:

The membership function introduced

$$\mu[w(x, y, 1)] = \begin{cases} 0 & if\ w(x, y, 1) \leq \alpha \\ \left(\frac{w(x,y,1)-\alpha}{\beta-\alpha}\right) & if\ (w(x, y, 1) > \alpha) \& (w(x, y, 1) \leq \beta) \\ 1 & if\ w(x, y, 1) > \beta \end{cases} \quad ..(3.8)$$

The following values have been taken according to the simulation result α=30/255 & β= 60/255 . The improved pixels are selected one by one and making the complete image with the noise free pixels. If the membership function is 0, it means that the current pixel is a noise-free pixel and hence it is no need to filter. The filter will output the original pixel and preserve the image information. If the membership function is 1, it describe that the current pixel has been corrupted by impulse noise and it needs to filter. Here used in the filter output is the improvement of current pixel. If the membership function is in between 0 and 1, it means that the current pixel has been corrupted somewhat by impulse noise and will select the function

$\left(\frac{w(x,y,1)-\alpha}{\beta-\alpha}\right)$. After the selection through fuzzy logic the pixels have been implemented into the equation

$$I(x,y,1) = \bar{f}(x,y,1) + \mu(x,y,1) * (E(x,y,1) - \bar{f}(x,y,1)) \qquad .(3.9)$$

for the red image. Similarly the impulse noise has been removed for the green and blue images. Then all results of R,G,B images have been combined to get the colored improved image.

3.1.2 . Simulation of Results

Computer simulations are carried out to assess the performance of the proposed method Lena image. The proposed method was tested

Chapter 4

RESULTS

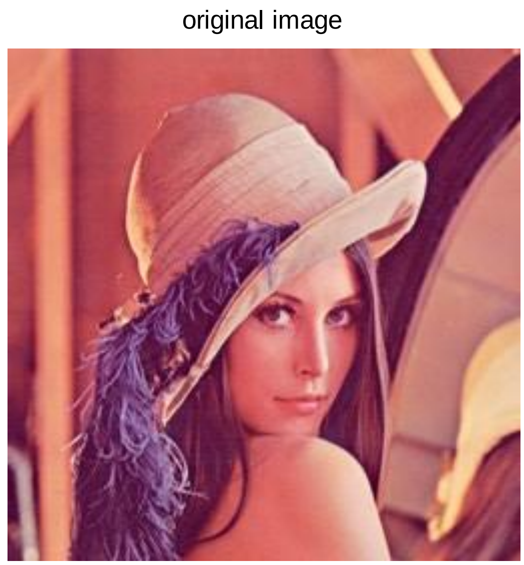

fig. (1) Original image

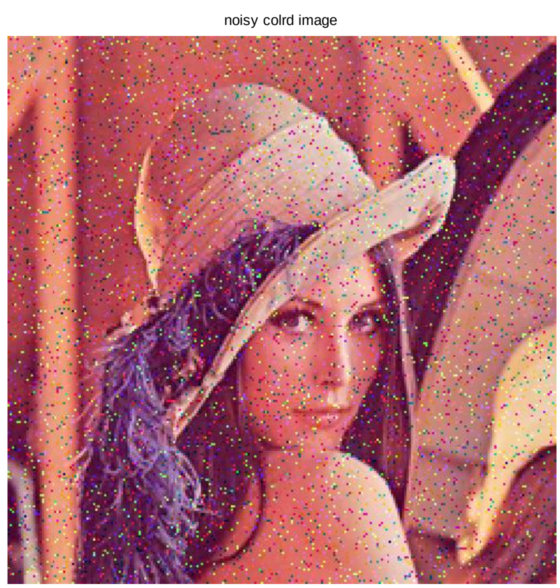

fig. (1.a) With 5% Impulse noise

filtered image

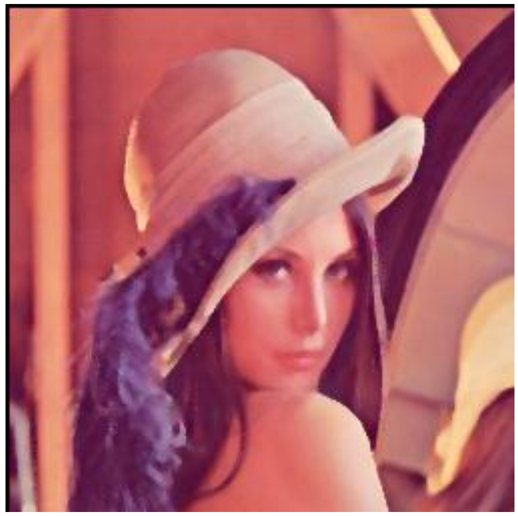

fig. (1.b) Improved filtered Image

filtered fuzzy image

fig. (1.c) Fuzzy filtered Image

noisy colrd image

fig. (2.a) With 5% Impulse noise

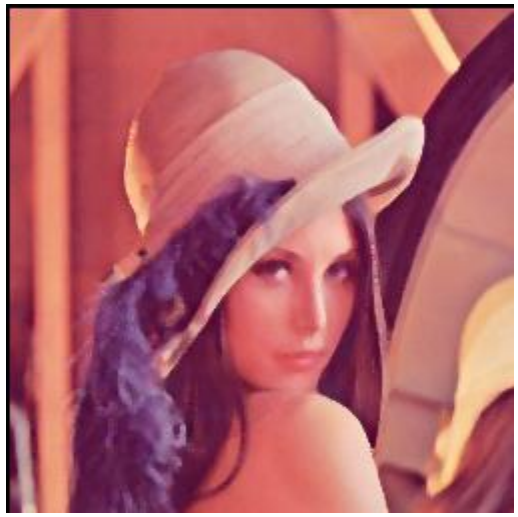

fig. (2.b) Improved filtered Image

filtered fuzzy image

fig. (2.c) Fuzzy filtered Image

original image

fig. (3) Original image

noisy colrd image

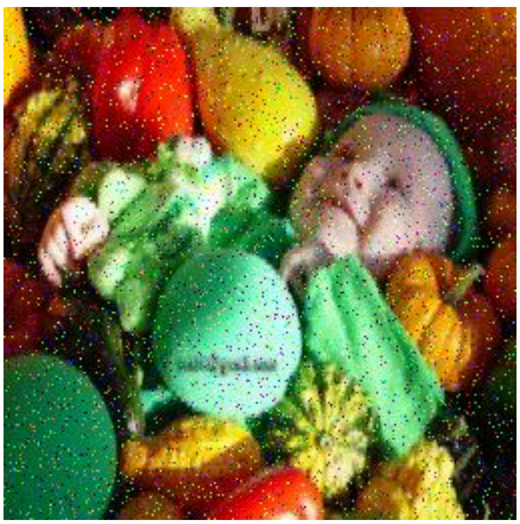

fig. (3.a) With 5% Impulse noise

fig. (2.b) Improved filtered Image

filtered fuzzy image

fig. (2.c) Fuzzy filtered Image

noisy colrd image

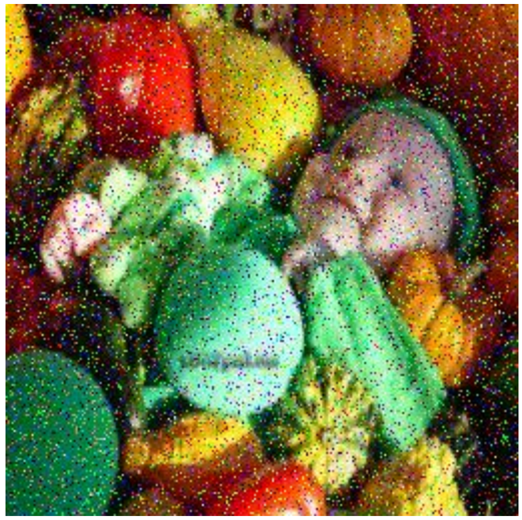

fig. (3.a) With 10% Impulse noise

filtered image

fig. (2.b) Improved filtered Image

filtered fuzzy image

fig. (2.c) Fuzzy filtered Image

original image

fig. (3) Original image

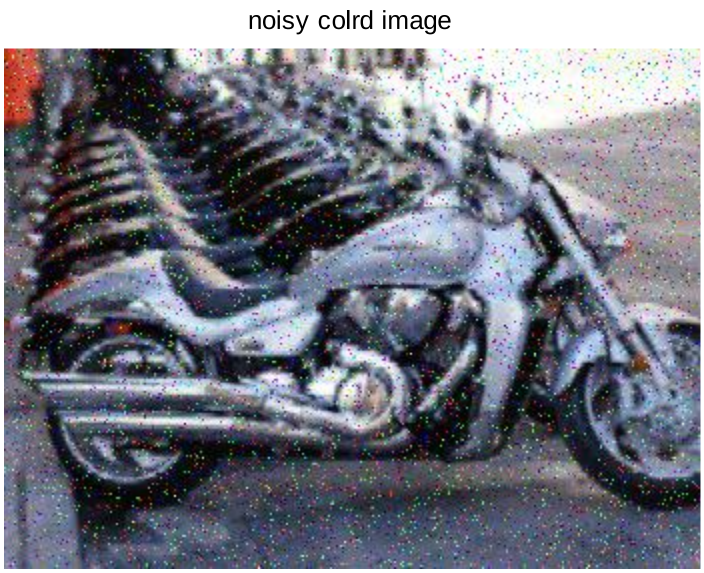

fig. (3.a) With 5% Impulse noise

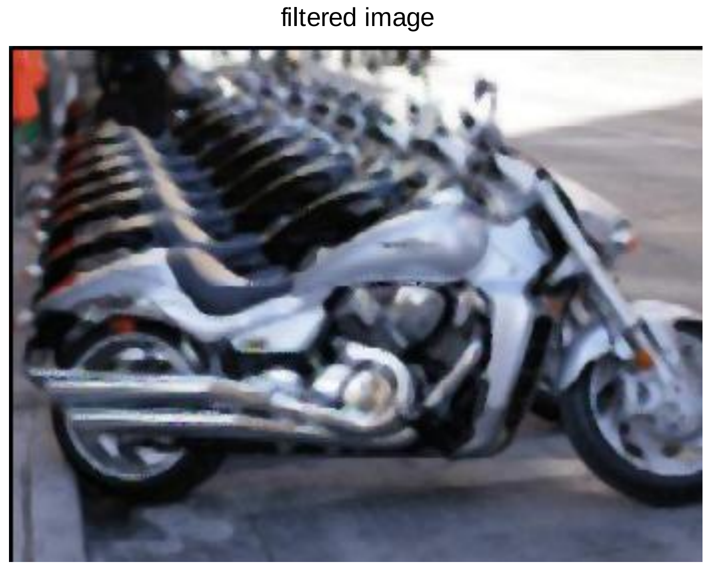

fig. (2.b) Improved filtered Image

fig. (2.c) Fuzzy filtered Image

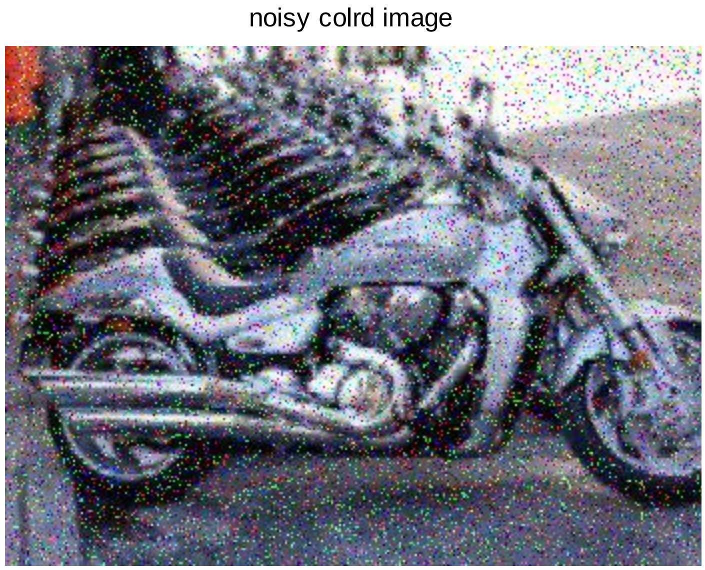

fig. (3.a) With 10% Impulse noise

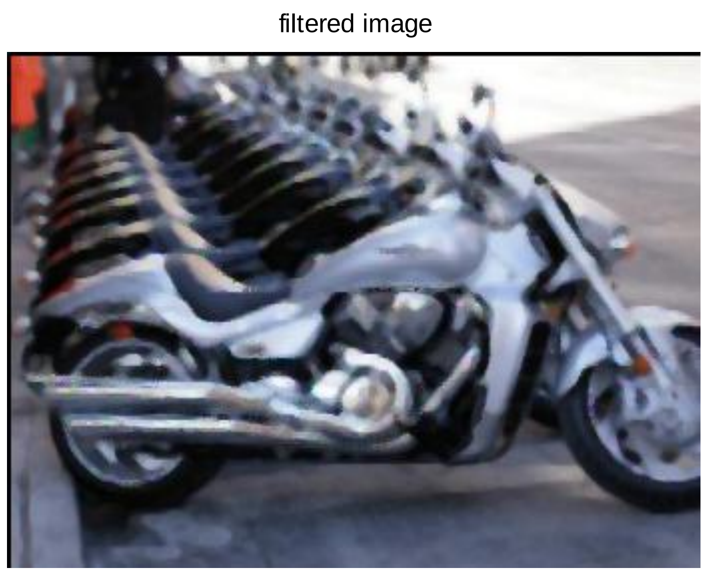

fig. (2.b) Improved filtered Image

filtered fuzzy image

fig. (2.c) Fuzzy filtered Image

MSE Results with 5% & 10 % of Impulse noise

S.N.	Noise / Filter	5%			10 %		
		R	G	B	R	G	B
1	Noisy image	0.0160	0.0149	0.0134	0.0328	0.0328	0.0272
2	Improved pixel image	0.0096	0.0037	0.0035	0.0098	0.0098	0.0036
3	Proposed	0.0090	0.0030	0.0028	0.0091	0.0091	0.0029

	filter						

Conclusion

In this thesis we have introduced the two step fuzzy filter that can remove impulse noise effectively while preserving details of the image. As shown by illustrative examples both numerical and visual the performance, the proposed filter is better than many of the existing filters such as the median filter and its variant and other fuzzy filters such as Gaussian fuzzy filter with median center (GMED),the symmetrical triangular fuzzy filter with median center (TMED), the asymmetrical triangular fuzzy filter with median center (ATMED), the Gaussian fuzzy filter with moving average center (GMAV), the symmetrical triangular fuzzy filter with moving average center (TMAV), the asymmetrical triangular fuzzy filter with moving average center (ATMAV), and the moving average center filter (MAV) .

The advantages of the proposed method are:-

1) It reduces Impulse noise in a colored image (low level impulse noise).
2) It preserves edge sharpness

3) It does not introduce blurring artifacts or new colors artifacts in comparison to other state of the art metods.

Chapter 6

Codes

```
close all;clear all; clc;
Y=imread('c:\lena.jpg');
figure;imshow(Y);title('original image');
y=imnoise(Y,'salt & pepper',0.1);
figure;imshow(y);title('noisy colrd image');
r=y(:,:,1);
l=y(:,:,2);
b=y(:,:,3);
r=im2double(r);
l=im2double(l);
b=im2double(b);
```

```
%figure;imshow(r);title('red image');
%figure;imshow(l);title('green image');
%figure;imshow(b);title('blue image');
alpha=30/255;
beta=60/255;
%FOR RED IMAGE-----------------------------------------------------------
R=imnoise(r,'salt & pepper',0.1); %insert imge noise
%R=im2double(R);
%figure;imshow(R);title('noisy red image');
w1=[ 0   0   0   0   0;
     0   0   0   0   0;
    -1  -1   4  -1  -1;
     0   0   0   0   0;
     0   0   0   0   0];
w2=[ 0   0  -1   0   0;
     0   0  -1   0   0;
     0   0   4   0   0;
     0   0  -1   0   0;
     0   0  -1   0   0];
w3=[ 0   0   0   0  -1;
     0   0   0  -1   0;
     0   0   4   0   0;
     0  -1   0   0   0;
    -1   0   0   0   0];
w4=[-1   0   0   0   0;
     0  -1   0   0   0;
     0   0   4   0   0;
     0   0   0  -1   0;
     0   0   0   0  -1];
cr1=abs(conv2(R,w1,'same'));
cr2=abs(conv2(R,w2,'same'));
```

```
cr3=abs(conv2(R,w3,'same'));
cr4=abs(conv2(R,w4,'same'));
[c d]=size(a);
for i=3:c-2
    for j=3:d-2
        k=1;
        for p=-2:2
            for q=-2:2
                U(k)=a(i+p,j+q)
                k=k+1;
            end
        end
        s=sort(U,'ascend')
        M=median(s)
 g=1;
        for p=-2:2
            for q=-2:2

                D(p+3,q+3)=abs(a(i+p,j+q)-M)

                if D(p+3,q+3)<3
                    Di(g)=D(p+3,q+3)
                    g=g+1;

                end
            end
        end
        Ds=sort(Di,'ascend')
        Med(i,j)=median(Di)

    end
```

```
end



[c d]=size(R);

for i=3:c-2
    for j=3:d-2
        k=1;
        for p=-2:2
            for q=-2:2
                UR(k)=R(i+p,j+q);
                k=k+1;
            end
        end
        S=sort(UR,'ascend');
        M=median(S);
        g=1;
        for p=-2:2
            for q=-2:2
                DR(p+3,q+3)=abs(R(i+p,j+q)-M);
                if DR(p+3,q+3)<=0.3
                    RR(g)=R(i+p,j+q);
                    g=g+1;
                end
            end
        end

        Cr(i,j)=min(min(cr1(i,j),cr2(i,j)),min(cr3(i,j),cr4(i,j)));

        %without fuzzy-------------------------
```

```
        if(Cr(i,j)<0.3)
            delta(i,j)=0;
        else
            delta(i,j)=1;

        end
        difR=sort(RR,'ascend');
        medR(i,j)=median(difR);

        ER(i,j)=delta(i,j)*medR(i,j)+(1-delta(i,j))*R(i,j);

        %with fuzzy-----------------------------
        if Cr(i,j)<=alpha
            mr(i,j)=0;
        elseif (Cr(i,j)>alpha)&&(Cr(i,j)<=beta)
            mr(i,j)=((Cr(i,j)-alpha)/(beta-alpha));
        elseif (Cr(i,j)>beta)
            mr(i,j)=1;
        end
        IR(i,j)=R(i,j)+mr(i,j)*(ER(i,j)-R(i,j));

    end
end



%figure;imshow(ER);title('noise free red image ');
%figure;imshow(IR);title('noise free red image ER fuzzy ');

 g1=imfilter(g,w1,'conv');
g2=imfilter(g,w2,'conv');
g3=imfilter(g,w3,'conv');
```

```
g4=imfilter(g,w4,'conv');
c1=min(g1,g2);
c2=min(g3,g4);
c4=min(c1,c2);
c3=abs(c4);%implse noise detection
k_size=3;
d=medfilt2(g,[k_size,k_size]);
figure;imshow(d);title('median image');
 c5=im2uint8(c3);
 alpha=c5>40;
 row1=size(g,1);
col1=size(g,2);
for i=1:row1
    for j=1:col1
        y2(i,j)=alpha(i,j)*d(i,j)+(1-alpha(i,j))*g(i,j);
    end
end
 figure;imshow(y2);title('improvement in current pixel');
  y3=im2uint8(y2);
  k=im2uint8(g);
  c=im2uint8(c4);
  d=im2uint8(d);
  y3=double(y3);
  l=double(k);
  c=double(c);
 d=double(d);
 %application of median rule
row1=size(g,1);
col1=size(g,2);
for m=2:250
    for n=2:250
```

```
          %u=g(m-1:m+1,n-1:n+1);
          %u1=reshape(u,9,1);
          u=[g(m,n-1); g(m,n+1) ;g(m-1,n); g(m+1,n); g(m+1,n+1); g(m+1,n-1); g(m-1,n+1); g(m-
1,n-1); g(m,n)];

         u2=sort(u,'ascend');
      mm= median(u2);




    end
end
 %application of fuzzy rule
 p1=u(4)
p2=u(5)
p3=mm
row1=size(g,1);
col1=size(g,2);
for i=1:row1
    for j=1:col1
       if c(i,j)<=p1
       s(i,j)=0;
         elseif (c(i,j)>p1)& (c(i,j)<p3)
             s(i,j)=2*((c(i,j)-p1)/(p2-p1))^2;
               elseif (c(i,j)>p3)& (c(i,j)<p2)
                s(i,j)=1-(2*((c(i,j)-p2)/(p2-p1))^2);
                 else (c(i,j)>=p2)
                 s(i,j)=1;
       end
  y4(i,j)=s(i,j)*y3(i,j)+(1-s(i,j))*k(i,j);
    end
```

```
end
d=uint8(d);
y5=uint8(y4);
%error detection with median filter
row1=size(y,1);
col1=size(y,2);
for i=1:row1
    for j=1:col1
        r1(i,j)=((y(i,j)-d(i,j))^2);
    end
end

k1=sum(sum(r1));
mse1=k1/(size(y,1)*size(y,2))
%error detection with switching median filter
y2=im2uint8(y2);
row1=size(y,1);
col1=size(y,2);
for i=1:row1
    for j=1:col1
        r2(i,j)=((y(i,j)-y2(i,j)).^2);
    end
end

k2=sum(sum(r2));
mse2=k2/(size(y,1)*size(y,2))
%error detection with fuzzy filter
row1=size(y,1);
col1=size(y,2);
for i=1:row1
    for j=1:col1
```

```
        r3(i,j)=((y(i,j)-y5(i,j)).^2);
    end
end

k3=sum(sum(r3));
mse3=k3/(size(y,1)*size(y,2))
%error detection with noisy image
for i=1:row1
    for j=1:col1
        r4(i,j)=((y(i,j)-k(i,j)).^2);
    end
end
k4=sum(sum(r4));
mse4=k4/(size(y,1)*size(y,2))
display(p1)
display(p2)
display(u2)
display(mm)
figure;imshow(h);title('noisy image with 5% noise');
figure;imshow(y5);title('filtered image');



%FOR GREEN IMAGE-----------------------------------------------------------
G=imnoise(l,'salt & pepper',0.1); %insert imge noise
%G=im2double(G);
%figure;imshow(G);title('noisy green image');
cg1=abs(conv2(G,w1,'same'));
cg2=abs(conv2(G,w2,'same'));
cg3=abs(conv2(G,w3,'same'));
cg4=abs(conv2(G,w4,'same'));
```

```
[c d]=size(G);

for i=3:c-2
    for j=3:d-2
        k=1;
        for p=-2:2
            for q=-2:2
                UG(k)=G(i+p,j+q);
                k=k+1;
            end
        end
        S=sort(UG,'ascend');
        M=median(S);
        g=1;
        for p=-2:2
            for q=-2:2
                DG(p+3,q+3)=abs(G(i+p,j+q)-M);
                if DG(p+3,q+3)<=0.3
                    GG(g)=G(i+p,j+q);
                    g=g+1;
                end
            end
        end

        Cg(i,j)=min(min(cg1(i,j),cg2(i,j)),min(cg3(i,j),cg4(i,j)));

        %without fuzzy-------------------------
        if(Cg(i,j)<0.3)
            delta(i,j)=0;
        else
```

```
            delta(i,j)=1;

        end
        difG=sort(GG,'ascend');
        medG(i,j)=median(difG);

        EG(i,j)=delta(i,j)*medG(i,j)+(1-delta(i,j))*G(i,j);

        %with fuzzy------------------------------
        if Cg(i,j)<=alpha
            mg(i,j)=0;
        elseif (Cg(i,j)>alpha)&&(Cg(i,j)<=beta)
            mg(i,j)=((Cg(i,j)-alpha)/(beta-alpha));
        elseif (Cg(i,j)>beta)
            mg(i,j)=1;
        end
        IG(i,j)=G(i,j)+mg(i,j)*(EG(i,j)-G(i,j));

    end
end


%figure;imshow(EG);title('noise free green image ');
%figure;imshow(IG);title('noise free green image EG fuzzy ');

%FOR BLUE IMAGE------------------------------------------------------------
B=imnoise(b,'salt & pepper',0.1); %insert imge noise
%B=im2double(B);
% figure;imshow(B);title('noisy blue image');
cb1=abs(conv2(B,w1,'same'));
cb2=abs(conv2(B,w2,'same'));
```

```
cb3=abs(conv2(B,w3,'same'));
cb4=abs(conv2(B,w4,'same'));

[c d]=size(B);

for i=3:c-2
    for j=3:d-2
        k=1;
        for p=-2:2
            for q=-2:2
                UB(k)=B(i+p,j+q);
                k=k+1;
            end
        end
        S=sort(UB,'ascend');
        M=median(S);
        g=1;
        for p=-2:2
            for q=-2:2
                DB(p+3,q+3)=abs(B(i+p,j+q)-M);
                if DB(p+3,q+3)<=0.3
                    BB(g)=B(i+p,j+q);
                    g=g+1;
                end
            end
        end

        Cb(i,j)=min(min(cb1(i,j),cb2(i,j)),min(cb3(i,j),cb4(i,j)));

        %without fuzzy-------------------------
        if(Cb(i,j)<0.3)
```

```
            delta(i,j)=0;
        else
            delta(i,j)=1;

        end
        difB=sort(BB,'ascend');
        medB(i,j)=median(difB);

        EB(i,j)=delta(i,j)*medB(i,j)+(1-delta(i,j))*B(i,j);

        %with fuzzy------------------------------
        if Cb(i,j)<=alpha
            mb(i,j)=0;
        elseif (Cb(i,j)>alpha)&&(Cb(i,j)<=beta)
            mb(i,j)=((Cb(i,j)-alpha)/(beta-alpha));

        elseif (Cb(i,j)>beta)
            mb(i,j)=1;
        end
        IB(i,j)=B(i,j)+mb(i,j)*(EB(i,j)-B(i,j));

    end
end

%figure;imshow(EB);title('noise free BLUE image ');
%figure;imshow(IB);title('noise free BLUE image EB fuzzy ');

fil(:,:,1)=ER;
fil(:,:,2)=EG;
fil(:,:,3)=EB;
figure;imshow(fil);title(' filtered image');
```

```
fil(:,:,1)=IR;
fil(:,:,2)=IG;
fil(:,:,3)=IB;
figure;imshow(fil);title(' filtered fuzzy image');

%error detection with noisy RED image
for i=1:c-2
    for j=1:d-2
        red(i,j)=((r(i,j)-R(i,j)).^2);
    end
end
redS=sum(sum(red));
mseR=redS/(c*d)

%error detection with noisefree RED image
for i=1:c-2
    for j=1:d-2
        redE(i,j)=((r(i,j)-ER(i,j)).^2);
    end
end
redSE=sum(sum(redE));
mseER=redSE/(c*d)
%error detection with fuzzy noisefree RED image
for i=1:c-2
    for j=1:d-2
        redI(i,j)=((r(i,j)-IR(i,j)).^2);
    end
end
redSI=sum(sum(redI));
```

```
mseIR=redSI/(c*d)

%error detection with noisy GREEN image
for i=1:c-2
    for j=1:d-2
        gren(i,j)=((l(i,j)-G(i,j)).^2);
    end
end
grenS=sum(sum(gren));
mseG=grenS/(c*d)

%error detection with noisefree GREEN image
for i=1:c-2
    for j=1:d-2
        grenE(i,j)=((l(i,j)-EG(i,j)).^2);
    end
end
grenSE=sum(sum(grenE));
mseEG=grenSE/(c*d)
%error detection with fuzzy noisefree GREEN image
for i=1:c-2
    for j=1:d-2
       grenI(i,j)=((l(i,j)-IG(i,j)).^2);
    end
end
grenSI=sum(sum(grenI));
mseIG=grenSI/(c*d)
%error detection with noisy BLUE image
for i=1:c-2
    for j=1:d-2
        blue(i,j)=((b(i,j)-B(i,j)).^2);
```

```
    end
end
blueS=sum(sum(blue));
mseB=blueS/(c*d)

%error detection with noisefree BLUE image
for i=1:c-2
    for j=1:d-2
        blueE(i,j)=((b(i,j)-EB(i,j)).^2);
    end
end
blueSE=sum(sum(blueE));
mseEB=blueSE/(c*d)
%error detection with fuzzy noisefree BLUE image
for i=1:c-2
    for j=1:d-2
        blueI(i,j)=((b(i,j)-IB(i,j)).^2);
    end
end
blueSI=sum(sum(blueI));
blueIR=blueSI/(c*d)
```

Chapter 7

References

[1] Om Prakash Verma1, Madasu Hanmandlu2, Anil Singh Parihar3 and Vamsi Krishna Madasu4 " Fuzzy Filters for Noise Reduction in Color Images"

[2] Gonzalez, R.C., Woods, R.E., Book on “Digital Image Processing”, 2nd Ed, Prentice-Hall of India Pvt. Ltd.

[3] R. Vorobel, "Contrast Enhancement of Remotely-Sensed Images," *in 6th Int. Conf. Math. Methods in Electromagnetic Theory*, Lviv, Ukraine, Sept 1996, pp.

[4] Tizhoosh, “Fuzzy Image Processing”, © Copyright Springer, 1997.

[5] Farzam Farbiz, Mohammad Bager Menhaj, Seyed A. Motamedi, and Martin T. Hagan, “A new Fuzzy Logic Filter for image Enhancement” IEEE Transactions on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 30, No. 1, February 2000

[6] Haisiang Xu, Guangsi Zhu, Fuyuan Peng, Haijun Gao “Spatially Adaptive Fuzzy Filter for Image Processing”, IEEE 2004.

[7] Shuqun Zhang, Mohammad A.Karim “A New Impulse Detector for Switching Median Filters.

[8] C.C. Lee. Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):419--435, 1990.

[9] H.J. Bremermann. Cybernetic Functionals and Fuzzy Sets. *IEEE Systems, Man and Cybernetics Group Annual Symposium*, pages 248--254, 1971.

[10] H.J. Bremermann. Pattern Recognition. *Systems Theory in the Social Sciences*, pages 116--159, 1976.

[11] D.G. Burkhardt and P.P. Bonissone. Automated Fuzzy Knowledge Base Generation and Tuning. *Proc. of the First IEEE Fuzzy Systems Conference*, pages 179-188, 1992.

[12] E. Cox. Fuzzy Fundamentals. *IEEE Spectrum*, pages 58--61, 1992.

[13] E. Cox. *The Fuzzy Systems Handbook*. Academic Press, 1994.

[14] C. Czarnecki, R. John, and S. Bennett. The Application of Fuzzy Logic To Real Time Multiple Robot Collision Avoidance. *Proc. ICSC Fuzzy Logic Symposium 1995*, pages C116--C121, 1995.

[15] Carl Steven Rapp, “Image Processing and Image Enhancement”, Texas, 1996.

[16] M.A. Lee and H. Takagi. Integrating Design Stages of Fuzzy Systems using Genetic Algorithms. *Second IEEE Int'l Conference on Fuzzy Systems*, 1:612--617, 1993.

[17] L. Davis. *Handbook of Gentic Algorithms*. Van Nostrand Reinhold, 1991. D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[18] H. Ishibuchi and K. Moriaka. Determination of Type II Membership Functions by Fuzzified Neural Networks. *EUFIT'95*, pages 529--533, 1995.

[19] H. Ishibuchi, K. Nozaki, and H. Tanaka. Distributed Representation of fuzzy rules and its application to pattern classification. *Fuzzy Sets and Systems*, 52:21--32, 1992.

[20] H. Ishibuchi, K. Nozaki, and N. Yanamoti. Selecting Fuzzy Rules by Genetic Algorithms for Classification Problems. *Second IEEE Intl. Conf. on Fuzzy Systems*, 2:1119--1124, 1993.

[21] C.J. Kim and D. Russell. Automatic Generation of Membership Function and Fuzzy Rule Using Inductive Reasoning. *IEEE??*, page ??, 1993.

[22] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*. Wiley, 1994.

[23] D.L. Meredith, C.L. Karr, and K. Krishna Kamur. The use of genetic algorithms in the design of fuzzy logic controllers. *3rd Workshop on Neural Networks WNN'92*, pages 549--545, 1992.

[24] M. Mizumoto and K. Tanaka. Some properties of Fuzzy Sets of Type 2. *Information and Control*, 31:312--340, 1976.

[25] M. Mizumoto and H.J. Zimmerman. Comparison of Fuzzy Reasoning Methods. *Fuzzy Sets and Systems*, 8:253--283, 1982.

[26] T.J. Schwarz. Fuzzy Systems in the Real World. *AI Expert*, 1990.

[27] I.B. Turksen. Measurement of membership functions and their acquisition. *Fuzzy Sets and Systems*, 40:5--38, 1991.

[28] S. Wang. Generating fuzzy membership functions: A monotonic neural network model. *Fuzzy Sets and Systems*, 61:71--81, 1994.

[29] N. Watanabe. Statistical Methods for Estimating Membership Functions. *Japanese Journal of Fuzzy Theory and Systems*, 5(4), 1979.

[30] L.A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338--353, 1965.