

STATUS OF THESIS

Title of Thesis

| |
|--|
| Genetic Fuzzy Filter Based on MAD and ROAD to Remove Mixed Impulse Noise |
|--|

I, NUR ZAHRATI JANAH

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only
3. This thesis is classified as

Confidential

Non-confidential

If this thesis is confidential, please state the reason:

The contents of the thesis will remain confidential for _____ years.

Remarks on disclosure:

Endorsed by

Signature of Author

Permanent Address:
Munggur, RT 01, RW 14, Srimartani,
Piyungan, Bantul, Yogyakarta, 55792,
Indonesia

Date: _____

Signature of Supervisor

Name of Supervisor

Date: _____

UNIVERSITI TEKNOLOGI PETRONAS

GENETIC FUZZY FILTER BASED ON MAD AND ROAD TO REMOVE MIXED
IMPULSE NOISE

by

NUR ZAHRATI JANAHA

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree stated.

Signature: _____

Main Supervisor: _____

Signature: _____

Head of Department: _____

Date: _____

GENETIC FUZZY FILTER BASED ON MAD AND ROAD TO REMOVE MIXED
IMPULSE NOISE

by

NUR ZAHRATI JANAH

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

MASTER OF SCIENCE

COMPUTER AND INFORMATION SCIENCES

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR,

PERAK

APRIL 2010

DECLARATION OF THESIS

Title of Thesis

| |
|--|
| Genetic Fuzzy Filter Based on MAD and ROAD to Remove Mixed Impulse Noise |
|--|

I, NUR ZHRATI JANAH

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

Signature of Author

Permanent Address:
Munggur, RT 01, RW 14, Srimartani,
Piyungan, Bantul, Yogyakarta, 55792
Indonesia

Date: _____

Signature of Supervisor

Name of Supervisor

Date: _____

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Allah the Almighty for the blessings and guidance throughout my life, especially in the completion of this thesis. My special gratitude is due to my parents and brothers for their loving support and understanding.

I would like to express deep and sincere gratitude to my supervisor, Dr. Baharum Baharudin, for his guidance & support in my research works. I warmly thank Assoc. Prof. Dr. Ahmad Kamil Mahmood and Assoc. Prof. Dr. Fakhruddin Mohd Hashim for the guidance during my first steps in UTP that has made it possible for me to find this research topic.

My sincere thanks are due to Assoc. Prof. Dr. Zulkifli Mohamed Udin from Universiti Utara Malaysia (UUM), Prof. Dr. Nanna Suryana Herman from Universiti Teknikal Malaysia (UTeM) Melaka, Dr. Jafreezal Jaafar and Dr. Mohd Fadzil Hassan for the detailed and constructive comments that have been very helpful for this study.

I also wish to thank Mr. Afifi Mohd Akib from Electrical and Electronics Engineering department UTP for his essential assistance in translating the abstract into Bahasa Melayu. I am indebted to many of my colleagues who support me during my masters study in UTP, especially postgraduate fellows in Computer and Information Sciences department and all Indonesian students in UTP.

The financial support of the Universiti Teknologi PETRONAS is gratefully acknowledged.

ABSTRACT

In this thesis, a genetic fuzzy image filtering based on rank-ordered absolute differences (ROAD) and median of the absolute deviations from the median (MAD) is proposed. The proposed method 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. Our idea is to utilize MAD and ROAD as measures of noise probability of a pixel. Fuzzy inference system is used to justify the degree of which a pixel can be categorized as noisy. Based on the fuzzy inference result, the fuzzy switching scheme that adopts median filter as the main estimator is applied to the filtering. The GA training aims to find the best parameters for the fuzzy sets in the fuzzy noise detection.

From the experimental results, the proposed method has successfully removed mixed impulse noise in low to medium probabilities, while keeping the uncorrupted pixels less affected by the median filtering. It also surpasses the other methods, either classical or soft computing-based approaches to impulse noise removal, in MAE and PSNR evaluations. It can also remove salt-and-pepper and uniform impulse noise well.

ABSTRAK

Tesis ini mengkaji dan mencadangkan tapisan imej genetik fuzzy berdasarkan rank-ordered absolute differences (ROAD) dan median penyimpangan mutlak daripada median asal (MAD). Kaedah yang dicadangkan dalam tesis ini terdiri daripada tiga komponen utama, iaitu sistem pengesan gangguan fuzzy, skim penapis suis fuzzy, dan pengoptimuman parameter fuzzy menggunakan algoritma genetik (GA). MAD dan ROAD dimanfaatkan sebagai pengukur kebarangkalian gangguan daripada suatu piksel. Sistem kesimpulan fuzzy diambil pakai untuk mengesahkan darjah sesuatu piksel boleh dikategorikan sebagai gangguan. Skim penapisan suis fuzzy yang mengadaptasikan kaedah penapisan median sebagai penilai utama menapis piksel-piksel dalam imej mengikut keputusan sistem kesimpulan fuzzy. Pelatihan algoritma genetik diguna pakai untuk mencari parameter terbaik yang akan digunakan oleh set-set fuzzy di dalam sistem pengesan gangguan fuzzy.

Berdasarkan keputusan eksperimen, kaedah yang dicadangkan telah berjaya menghapuskan gangguan impuls campuran dalam kebarangkalian yang rendah sampai menengah, sambil memastikan piksel yang normal tidak terkesan dengan kaedah penapisan median. Prestasi keseluruhan yang ditunjukkan oleh kaedah ini adalah lebih baik berbanding kaedah-kaedah lain, samada ianya kaedah klasik ataupun kaedah berdasarkan pendekatan soft computing, dalam evaluasi MAE dan PSNR. Ianya juga boleh menghapuskan gangguan ‘salt-and-pepper’ dan gangguan impuls sekata dengan baik.

In compliance with the terms of the Copyright Act 1987 and the IP policy of the university, the copyright of this thesis has been reassigned by the author to the legal entity of the university,

Institute of Technology PETRONAS Sdn Bhd.

Due to acknowledgement shall always be made of the use of any material contained in, or derived from, this thesis.

© Nur Zahrati Janah, 2010
Institute of Technology PETRONAS Sdn Bhd
All rights reserved.

TABLE OF CONTENTS

| | |
|---|------|
| STATUS OF THESIS..... | i |
| UNIVERSITI TEKNOLOGI PETRONAS | ii |
| TITLE PAGE | iii |
| DECLARATION OF THESIS | iv |
| ACKNOWLEDGEMENTS | v |
| ABSTRACT..... | vi |
| ABSTRAK..... | vii |
| COPYRIGHT PAGE | viii |
| TABLE OF CONTENTS..... | ix |
| LIST OF TABLES..... | xi |
| LIST OF FIGURES | xii |
| LIST OF ABBREVIATIONS..... | xiv |
| | |
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 Overview..... | 1 |
| 1.2 Problem Statement..... | 3 |
| 1.3 Objectives | 3 |
| 1.4 Scope of Study | 4 |
| 1.5 Thesis Organization | 4 |
| 1.6 List of Publications | 5 |
| | |
| CHAPTER 2 RELATED WORKS..... | 6 |
| 2.1 Median Filter and Its Modifications | 6 |
| 2.2 Soft Computing in Image Enhancement..... | 8 |
| 2.2.1 Neural Network..... | 8 |
| 2.2.2 Genetic Algorithms..... | 9 |
| 2.2.3 Fuzzy Filters..... | 10 |
| 2.3 Summary | 13 |
| | |
| CHAPTER 3 THEORETICAL BACKGROUNDS | 14 |
| 3.1 Fuzzy Systems | 14 |
| 3.1.1 Fuzzy sets..... | 14 |
| 3.1.2 Membership functions | 15 |
| 3.1.3 Fuzzy operators..... | 17 |
| 3.1.4 Fuzzy rules | 18 |
| 3.1.5 Fuzzy inference systems | 19 |
| 3.1.5.1 Fuzzification | 19 |
| 3.1.5.2 Rule evaluation | 19 |
| 3.1.5.3 Aggregation..... | 19 |
| 3.1.5.4 Defuzzification..... | 20 |
| 3.2 Genetic Algorithms..... | 20 |

| | | |
|--|---|----|
| 3.2.1 | Chromosome representation | 21 |
| 3.2.2 | Fitness function | 22 |
| 3.2.3 | Population | 23 |
| 3.2.4 | Parent selection | 23 |
| 3.2.5 | Crossover | 24 |
| 3.2.6 | Mutation | 25 |
| 3.2.7 | Stopping criteria | 26 |
| 3.2.8 | Constraint handling | 26 |
| 3.3 | Genetic-fuzzy systems | 27 |
| 3.4 | Digital Image Representation | 28 |
| 3.5 | Impulsive Noise Models | 29 |
| 3.6 | Median of the Absolute Deviations (MAD) | 30 |
| 3.7 | Rank-Ordered Absolute Differences (ROAD)..... | 32 |
| 3.8 | Switching scheme filtering | 33 |
| CHAPTER 4 METHODOLOGY | | 35 |
| 4.1 | Fuzzy Noise Detection System | 35 |
| 4.1.1 | MAD and ROAD Application | 36 |
| 4.1.2 | Fuzzy Inference System | 37 |
| 4.2 | Fuzzy Switching Scheme Filtering | 41 |
| 4.3 | Fuzzy Parameter Optimization using Genetic Algorithms | 41 |
| 4.3.1 | Gene Encoding | 41 |
| 4.3.2 | Genetic Learning Scheme | 42 |
| CHAPTER 5 RESULTS AND ANALYSIS..... | | 45 |
| 5.1 | Obtaining the optimal parameter set | 48 |
| 5.2 | Filter performance in preserving image details | 51 |
| 5.3 | Filter performance in impulsive noise removal | 54 |
| 5.3.1 | Filter performance in removing mixed impulse noise | 54 |
| 5.3.2 | Filter performance in removing salt-and-pepper noise | 60 |
| 5.3.3 | Filter performance in removing uniform impulse noise | 62 |
| CHAPTER 6 CONCLUSIONS AND FUTURE WORK..... | | 68 |
| 6.1 | Accomplishing the Goal of This Research | 68 |
| 6.2 | Research Contributions | 69 |
| 6.3 | Future Works | 70 |
| REFERENCE..... | | 71 |
| APPENDIX A..... | | 74 |

LIST OF TABLES

| | | |
|------------|--|----|
| Table 5.1 | GA training results | 50 |
| Table 5.2 | Test to find the optimal parameter for general cases | 50 |
| Table 5.3 | The final GA training results | 50 |
| Table 5.4 | MAE and PSNR values of noise-free Lena image..... | 53 |
| Table 5.5 | MAE values of Lena images corrupted by mixed impulse noise | 55 |
| Table 5.6 | PSNR values of Lena images corrupted by mixed impulse noise | 55 |
| Table 5.7 | MAE values of the standard test images corrupted by mixed impulse noise | 59 |
| Table 5.8 | PSNR values of the standard test images corrupted by mixed impulse noise | 59 |
| Table 5.9 | MAE values of Lena images corrupted by salt-and-pepper noise | 60 |
| Table 5.10 | PSNR values of Lena images corrupted by salt-and-pepper noise | 61 |
| Table 5.11 | MAE values of Lena images corrupted by uniform impulse noise | 63 |
| Table 5.12 | PSNR values of Lena images corrupted by uniform impulse noise ... | 63 |
| Table 5.13 | MAE values comparison of filtered Lena images corrupted by mixed impulse, salt-and-pepper, and uniform impulse noise | 66 |
| Table 5.14 | PSNR values comparison of filtered Lena images corrupted by mixed impulse, salt-and-pepper, and uniform impulse noise | 67 |

LIST OF FIGURES

| | | |
|-------------|--|----|
| Figure 3.1 | Trapezoidal membership function | 16 |
| Figure 3.2 | The pseudocode of GA process | 21 |
| Figure 3.3 | The pseudocode of roulette wheel algorithm..... | 24 |
| Figure 3.4 | Schematic diagram of a genetic-fuzzy system..... | 28 |
| Figure 3.5 | Deviation image between noisy image x_{ij} and its median image (in inverted color)..... | 31 |
| Figure 3.6 | MAD image of the noisy Lena image for 5G5 window size (in inverted color)..... | 32 |
| Figure 4.1 | The diagram of the system..... | 36 |
| Figure 4.2 | Histogram comparison of d values of noisy and noise-free pixels from Lena image corrupted by mixed impulse noise 0.2..... | 37 |
| Figure 4.3 | Histogram comparison of MAD values of noisy and noise-free pixels from Lena image corrupted by mixed impulse noise 0.2 | 38 |
| Figure 4.4 | Histogram comparison of ROAD values of noisy and noise-free pixels from Lena image corrupted by mixed impulse noise 0.2 | 38 |
| Figure 4.5 | Fuzzy sets HIGH, LOW, and LARGE for d , MAD, and ROAD, respectively | 40 |
| Figure 4.6 | The gene encoding | 42 |
| Figure 4.7 | The genetic learning process..... | 43 |
| Figure 5.1 | Standard test images | 48 |
| Figure 5.2 | The fitness evolution in the GA training..... | 49 |
| Figure 5.3 | The application of the proposed filter and median filter to a noise-free Lena image..... | 52 |
| Figure 5.4 | (a) 128×128 8-bit grayscale Lena image, (b) image corrupted by 0.1 mixed impulse noise, (c) noisy image (b) filtered by median filter in 5×5 windows, and (d) noisy image (b) filtered by the proposed method..... | 53 |
| Figure 5.5 | MAE graph comparison among Lena image corrupted by 0.1 mixed impulse noise, filtered by 5x5 median filter, filtered by the proposed filter and its original image. | 54 |
| Figure 5.6 | MAE curves of Lena images corrupted by mixed impulse noise | 56 |
| Figure 5.7 | PSNR curves of Lena images corrupted by mixed impulse noise..... | 56 |
| Figure 5.8 | Noisy 256x256 8-bit grayscale Lena image corrupted by 0.2 mixed impulse noise filtered by: (a) the proposed filter, (b) 3x3 average filter, (c) 5x5 average filter, (d) 3x3 median filter, (e) 5x5 median filter, (f) EIE, (g) WFM, (h) GFIF, and (i) FRINR..... | 57 |
| Figure 5.9 | Noisy standard test images corrupted by 0.2 mixed impulse noise filtered by the proposed filter..... | 58 |
| Figure 5.10 | MAE curves of Lena images corrupted by salt-and-pepper noise..... | 62 |
| Figure 5.11 | Noisy 256x256 8-bit grayscale Lena image corrupted by 0.2 salt-and-pepper noise filtered by: (a) the proposed filter, (b) 3x3 | |

| | | |
|-------------|--|----|
| | average filter, (c) 5x5 average filter, (d) 3x3 median filter, (e) 5x5 median filter, (f) EIE, (g) WFM, (h) GFIF, and (i) FRINR..... | 61 |
| Figure 5.12 | PSNR curves of Lena images corrupted by salt-and-pepper noise..... | 62 |
| Figure 5.13 | MAE curves of Lena images corrupted by uniform impulse noise | 65 |
| Figure 5.14 | PSNR curves of Lena images corrupted by uniform impulse noise ... | 65 |
| Figure 5.15 | Noisy 256x256 8-bit grayscale Lena image corrupted by 0.2 uniform impulse noise filtered by: (a) the proposed filter, (b) 3x3 average filter, (c) 5x5 average filter, (d) 3x3 median filter, (e) 5x5 median filter, (f) EIE, (g) WFM, (h) GFIF, and (i) FRINR..... | 64 |
| Figure 5.16 | MAE curves comparison of filtered Lena images corrupted by mixed impulse, salt-and-pepper, and uniform impulse noise | 66 |
| Figure 5.17 | PSNR curves comparison of filtered Lena images corrupted by mixed impulse, salt-and-pepper, and uniform impulse noise | 66 |

LIST OF ABBREVIATIONS

| | |
|-------|---|
| AWFM | Adaptive weighted fuzzy mean |
| CWM | Center weighted median |
| EIE | Evolutionary image enhancement |
| FRINR | Fuzzy random impulse noise reduction |
| GA | Genetic algorithms |
| GFIF | Genetic-fuzzy image filtering |
| MAD | Median of the absolute deviations from the median |
| MAE | Mean absolute error |
| PSNR | Peak-signal-to-noise ratio |
| ROAD | Rank-order absolute differences |
| ROLD | Rank-ordered logarithmic differences |
| VLSI | Very-large-scale integration |
| WFM | Weighted fuzzy mean |

CHAPTER 1

INTRODUCTION

1.1 Overview

Images are required as sources of information for interpretation and analysis by many applications. Noise can contaminate images in many cases, especially when they are transferred or converted from one form to another [1]. The most commonly found noises in images are impulse noise and additive noise. The main characteristic of impulse noise is that only parts of pixels are corrupted while the others remain free from noise. Salt-and-pepper noise is the most popular model of impulse noise, while Gaussian noise is an additive noise that has a Gaussian amplitude distribution.

The quality of input images plays a key role in image-based measurement. Many image enhancement techniques exist due to the needs of noiseless and highly defined images for better interpretation and analysis. The goal of image enhancement is to improve the image quality so that the processed image is better than the original image for a specific application or a set of objectives such as analysis, detection, segmentation, and recognition. Applications of digital image processing are mainly rooted from two principal areas: improvement of pictorial information for human interpretation, and processing of scene data for autonomous machine perception [1]. Enhancing image quality can be done by removing noise, amplifying image contrast and amplifying the level of a detail [2].

Various methods in image enhancement can be categorized into two broad categories: spatial domain methods and frequency domain methods [1]. Approaches in the spatial domain methods are based on manipulation of pixels in an image.

Frequency domain methods are based on modification to the Fourier transform of an image. Both approaches have their strengths and weaknesses.

Classical spatial filters, such as averaging or low-pass spatial filter and median filter, have been used to filter images for years. They provide a reasonable amount of noise removal performance by removing thin lines, distorting edges and blurring image details even at low densities [3]. Edges are features that define the shape of an object. Shape provides important information that triggers identification that is extremely important in visual recognition of objects [4].

Soft computing has become a new computing paradigm which is very relevant for pattern recognition and image analysis because of its tolerance to imprecision, uncertainty, approximate reasoning, and partial truth. Although soft computing is mostly applied in classification and recognition tasks, it can also be applied in image enhancement tasks [5]. Some soft computing methods that have been implemented in image enhancement are neural network, cellular neural network, pulse coupled neural network, rough set, fuzzy filter, genetic algorithms (GA), and the hybrids of those methods, such as fuzzy-GA and fuzzy network [6].

Salt-and-pepper noise is the most popular impulse noise model where corrupted pixels are replaced with maximum or minimum intensity values [1]. Abreu et al. [7] introduce random-valued impulse noise of which noise intensities randomly range from the minimum to the maximum intensity values, that is 0 to 255 for 8-bit grayscale images. A new model of impulse noise proposed by Petrović [8] incorporates both aforementioned impulse noise models. Half of the corrupted pixels follow the salt-and-pepper noise model, and the other half are contaminated by noise in random amplitudes, while the rest are free from noise. Let x_{ij} be an image containing mixed impulse noise,

$$x_{ij} = \begin{cases} n_{ij}^{unif}, & \text{with probability } \frac{p}{2} \\ n_{ij}^{snp}, & \text{with probability } \frac{p}{2} \\ f_{ij}, & \text{with probability } 1-p \end{cases} \quad (1.1)$$

where $n_{ij}^{unif} \in [L_{\min}, L_{\max}]$ and $n_{ij}^{snp} \in \{L_{\min}, L_{\max}\}$. L_{\min} and L_{\max} are the minimum and maximum intensity values allowed in an image. Also, f_{ij} denotes a noise-free image

pixel at the location (i, j) . This type of noise model is claimed to be more realistic because the impulse noise occurs as a result of disturbances induced by noise signals with random amplitudes. The amplitudes of the impulse noise could fall into the dynamic range allowed as well as out of the range. The noise will be saturated to the maximum or minimum value if it is out of the range. Otherwise, it will appear as uniform impulse noise if it is within the dynamic range. Therefore, this noise model is more suitable for evaluating the performance of impulse noise filters.

1.2 Problem Statement

Image filter design mainly focuses on removing one type of noise only, e.g. salt-and-pepper, the most popular one [9-17]. Most the existing methods can remove salt-and-pepper well, but they fail to remove mixed impulse noise [8], which is a more natural model for impulsive noise. It is of less usefulness if our filter has outstanding simulation performance in removing salt-and-pepper noise, but it fails completely when implemented in the real world application. Therefore, it is a great challenge to develop algorithms that can remove mixed impulse noise, a more natural model of impulsive noise, from an image while preserving the details in the image and keeping the computational cost as low as possible.

1.3 Objectives

The objectives of this research are:

1. To analyze current noise removal methods based on soft computing.
2. To develop an image filter based on median of the absolute deviations from the median (MAD) [18] and rank-order absolute differences (ROAD) [19] that applies the concept of fuzzy filtering to remove mixed impulse noise.
3. To conduct experiments for assessing the performance of the proposed filter in removing mixed impulse, salt-and-pepper, and uniform impulse noise in images.

1.4 Scope of Study

Some scope definitions are set to keep the research works focused and done within the time specified.

1. The proposed filter and all filtering methods discussed in this thesis are concentrated to the filters that work on spatial domain only.
2. It focuses on removing impulsive type of noise in grayscale images. Impulse noise is one of the most commonly found noises in images.
3. Experiments conducted to assess the filter's performance include performance comparison with the following filters:
 - a. Classical spatial filters: median and average filters
 - b. Evolutionary image enhancement (EIE) [9]
 - c. Weighted fuzzy mean filter (WFM) [11, 12]
 - d. Genetic-fuzzy image filter (GFIF) [13]
 - e. Fuzzy random impulse noise reduction (FRINR) [20]

1.5 Thesis Organization

The thesis is organized as follows. Chapter 2 discusses some works related to the field of discussion. It contains critical evaluation of other related researches. It discusses what other researchers have done in the field and also the issues and challenges faced.

Chapter 3 presents theoretical backgrounds to support the proposal of the new method. It covers information about digital image representation, impulsive noise models, ROAD and MAD image statistical properties, fuzzy filters, and genetic algorithms.

The methodology is proposed in chapter 4, including fuzzy noise detection system, fuzzy switching scheme filtering, and fuzzy parameters optimization using genetic algorithms, to picture the complete system.

Experiment results over various cases and the analysis are discussed in chapter 5. It is explained how to obtain the parameter used for the filtering. The filter is tested on noise-free image to know the performance in detail preservation. It is also tested on three types of impulse noise to assess its performance in removing those types of noise.

Finally, the conclusions and the future works are drawn in chapter 6. This chapter also highlights the contributions made in this thesis.

1.6 List of Publications

Several publications have been resulted while the research was conducted. There were one journal submitted and three conferences attended. The details of the publications can be seen in Appendix A.

CHAPTER 2

RELATED WORKS

2.1 Median Filter and Its Modifications

The interest in the development of nonlinear techniques for image processing continues to increase in recent years. In spite of the existence of linear filters with their simplicity, nonlinear filters are demanded because they offer benefits that most linear filters cannot gain. The human visual system naturally involves nonlinear effects that must be considered to get better visual results. Generally, images are signals that do not satisfy the hypotheses of Gaussianity and stationarity that are usually assumed to validate linear models and filtering techniques. Moreover, the advances of computers and digital signal processors, in terms of speed, size, and cost, make the implementation of more sophisticated algorithms possible. The list of classes in nonlinear image filtering includes homomorphic filters, nonlinear mean filters, morphological filters, order-statistics filters, polynomial filters, fuzzy filters, and nonlinear operators modeled in terms of nonlinear partial differential equations [21]. Most of our discussion in this thesis covers rank-order statistics filters, fuzzy filters, and other filters based on soft computing techniques.

Order-statistics filters are nonlinear spatial filters of which response is based on ranking the pixels in the filtering window, and then replacing the value of the center pixel with the value determined by the ranking result. The most widely used filter in this class is the median filter [1]. To compute the output of the median filter, an odd number of sample pixels from a filtering window are sorted, and the median value is used as the filter output. Median filter is successfully used in noise removal application for years despite of its simplicity. Two intrinsic properties that contribute to the success of median filter are edge preservation and efficient noise attenuation

with robustness against impulsive noise types. Edge preservation is very important in image processing due to the nature of human visual perception [4, 14].

However, median filter application can cause edge jitter, streaking problem and loss of small details from the images. The reason for this is that the median filter uses only the rank-order information from the input data, and discards the original spatial-order information data. In the median filter, each pixel inside the filter window has the same influence on the output. One approach to overcome the problems is to give more emphasis to pixels in specific window position, e.g. to the center of the filter window. This idea has led to the development of the weighted median filters [14, 15, 22] in which each pixel in a filter window has a specific weight that determines the effect of the current pixel to the filter output. It can satisfy user's predefined set of feature types to remove or retain by modifying the weights of the filter. The weights can be integer or real valued weights [1, 14].

Center-weighted median filter is a special case of weighted median filters. It gives more weight only to the central value of a filtering window, and thus it is easier to design and implement than the general weighted median filters [10]. The effect of the filtering can be adjusted by setting the filtering window size and the weight of the central pixel.

Although these filters can preserve more details than the median filter, they are still implemented uniformly across the image without considering whether the current pixel is noise-free or not. They have been proven as powerful methods to remove impulse noise, but at the same time they remove edges and other details in uncorrupted pixels. To overcome this limitation, noise removal methods with noise detectors are proposed. The main idea is to treat pixels differently based on the existence of noise in the pixels. Noise removal process is done in two steps: detection of impulses, and replacement of detected impulses with estimates, where the median is commonly used as the estimator [23].

Noise adaptive soft-switching median filter is a switching-based median filter with incorporation of fuzzy set concept aiming robustness in combating noise density variations. It consists of two stages: the detection scheme and the filtering scheme. The detection scheme categorizes each pixel to be uncorrupted pixels, isolated

impulse noise, nonisolated impulse noise, or edge pixel. Three types of filtering, i.e. identity filter, standard median filter, and fuzzy weighted median filter, are applied according to the results of the noise-detection scheme [24]. This approach can minimize edge and detail loss after filtering because the filter only replaces the value of pixels detected as noisy while the rest are kept at their original value. It is of great potential for future design of image filter that aims at removing noise and at the same time preserving more image features.

2.2 Soft Computing in Image Enhancement

Soft computing has become a new computing paradigm, which is very relevant for pattern recognition and image analysis because of its tolerance to imprecision, uncertainty, approximate reasoning, and partial truth. Although soft computing is mostly applied in classification and recognition tasks, it can also be applied in image enhancement [5]. Some soft computing methods that have been implemented in image enhancement are neural network, cellular neural network, pulse coupled neural network, rough set, fuzzy filter, genetic algorithms, and the hybrids of those methods: fuzzy-GA, fuzzy network [6].

2.2.1 Neural Network

More than 200 applications of neural networks in image processing have been reviewed by M. Egmont-Petersen et al. [5]. Since the earliest neural network, McCulloch-Pitts neuron, was invented in 1943 neural networks continue to grow and be implemented in many fields. Improved computational capabilities and hardware implementations on VLSI and optical media are some reasons why neural networks continue growing although there was a quiet period of neural networks development in 1970s. Neural networks are being used in signal processing, control, pattern recognition, medicine, speech production, speech recognition, business, etc.

Satyanarayana et al. [25] implemented neural network for video color enhancement to hardware. This intelligent system can automatically adjust the color saturation on a field-by-field basis. Implementations of artificial neural networks in

image enhancement are actively developed. New generations of neural networks, such as cellular neural network and pulse coupled neural network, are proposed to yield better result while keeping the computational complexity as low as possible [5, 26, 27].

Russo has employed fuzzy network that combines sharpening and noise reduction for images [6]. He proposed a multiple-output system consists of three fuzzy networks. Each fuzzy network operates on 3x3 window. The filtering is done recursively over all pixels in the image. Instead of complicated parameters needed by fuzzy systems, this method only needs one parameter for its execution, that is α . This parameter determines the level of noise cancellation.

2.2.2 Genetic Algorithms

Genetic algorithms have been applied to image enhancement by several authors. Various tasks in image processing solved by genetic algorithms range from basic image contrast and level of detail enhancement, to complex filters and deformable models parameters. Genetic algorithms are used to construct new filters, to optimize parameters of existing filters, and to look for optimal sequence of existing filters. The approach of genetic algorithm for each problem is unique, with different information encoding types, reproduction and selection schemes [2].

In his paper, Cho [9] employed evolutionary image enhancement (EIE) to optimize a set of proper filters for noise reduction in images. The output of the GA training is a set of optimum combination of five filters. There are 70 filters made available to be combined, which consists of histogram-based filters, mask morphology filters, and identity filter. For each generation, the fitness of the population is evaluated using fitness function derived from mean absolute error (MAE). This method outperformed applications of single spatial filter to images.

Petrović et al. [8] implemented genetic programming in universal impulse noise filter to detect mixed impulse noise in an image. Universal impulse noise filter is mixed impulse noise filter based on the switching scheme with two cascaded detectors and two corresponding estimators. The detectors are built using genetic programming

based on the robust estimators of location and scale: median and MAD. The estimators are the center weighted median (CWM) and the recursive variant of an α -trimmed mean estimate. Because it is designed to be able to remove mixed impulse noise, its performance in removing all types of impulse noise is good. Choosing a suitable noise model is important in the process of designing image filters.

Choosing a suitable set of filter is crucial yet difficult task in image filtering. There are no explicit rules defining the relationship between filtering methods and the type of noise. In this case, GA is very applicable to find the optimal filter set without prior knowledge on the rules.

GA approaches need extensive cost, in terms of time and computational resource, to obtain the training results. Therefore, it is common practice to separate the GA training process apart from the main filtering function. There is no need to perform GA training each time the filtering is performed. The training results usually can be stored and used for several filtering task.

2.2.3 Fuzzy Filters

Fuzzy filters are also an active research area of soft computing in image enhancement. Fuzzy filters in general are able to represent human's knowledge in a more comprehensive way than classical spatial filters, and they have the ability to combine edge preservation and smoothing [28]. A fuzzy system is a nonlinear system formed by a set of fuzzy rules (rule base) and an appropriate inference mechanism [21].

Fuzzy rules are implemented in fuzzy random impulse noise reduction (FRINR) method [29]. The method consists of two separated phases: two-step fuzzy detection mechanism and a fuzzy filtering. The first detection operates in local neighborhoods, while the second gathers more information from extended neighborhoods. Several fuzzy rules are fired simultaneously for each filtered pixel. The fuzzy filter only processes the pixels detected by both detector. This method is intended to remove random-valued impulse noise from corrupted images. It can detect and filter most noisy pixels quite well, but it still leaves some visible noisy pixels in the image. It

needs more than an iteration to gain optimum noise removal. Using the same concept, new methods are derived to remove impulsive noise in color images [30-32].

Kang and Wang proposed a directional median filter based on fuzzy reasoning to remove uniform impulse noise [33, 34]. It inspects differences between the current pixel and the neighbors in four directions. Applying fuzzy reasoning techniques, the filter categorizes pixels as impulse noise pixels, detailed pixels, or noise-free pixels. The filter work better on smooth images than on complicated images with small details.

Another filter based on directional weighted median and fuzzy logic was proposed by Hussain et al [35]. It consists of two stage noise detection and fuzzy filtering processes to detect and remove uniform impulse noise. The noise detection process includes fuzzy based noise detection and direction based noise detection that consider four edge directions in windows of size 5×5 . Fuzzy filtering processes are applied to each detected noisy pixel: calculation of direction based weighted median, construction of fuzzy set homogeneous, and noise removal. Both detection and filtering processes are repeated several times until the noise become negligible. Therefore, this filter requires high computational cost. It can remove impulse noise well, especially in low noise condition and smooth images.

In [11], Sugeno fuzzy inference system is applied in Weighted Fuzzy Mean (WFM) filter to remove heavy salt-and-pepper noise in grayscale images. The fuzzy sets DARK, MEDIUM, and BRIGHT are derived from the histogram of a reference image, and saved as the knowledge base for the filtering process. The output pixels are the mean value of the corrupted pixels weighted by a membership grade of an associated fuzzy set. This method is designed to simplify the implementation of the algorithm in either software or hardware. In the condition when noise probability higher than 0.3, WFM can remove most of the noisy pixels. Despite its performance in heavy noise environment, WFM filtered image loses many details in intensity gradient and edges. By simple visual observation, the output images look unnatural.

Lee enhanced the WFM method becomes Adaptive Weighted Fuzzy Mean Filter (AWFM) to do better in lower noise probability [12, 16]. While the three fuzzy sets are maintained, a dynamic selector is added to the system to decide whether a pixel is

corrupted or undecided. Thus, the AWFM has the advantage of WFM in removing heavy impulse noise and can do better in low noise environment. Compared to WFM, the details are more preserved and the output image looks slightly more natural. As the effect of the addition of dynamic selector in the system, the high computational requirement of WFM raises even higher in AWFM.

Further improvement of AWFM can be found in Genetic-based Fuzzy Image Filtering (GFIF) [13] where the number of fuzzy sets involved become five sets. Another significant improvement is the fuzzy sets parameter optimization using genetic algorithm. Specific parameter encoding and genetic learning are defined to solve the optimization problem. These changes bring significant improvement in the filtering results. It can work very well in removing salt-and-pepper noise, ranging from low to high noise probability. The filtered images look natural, because it can preserve most of the details in the images. These advantages come with a drawback: the computational cost that is undoubtedly high when filtering a corrupted image and even more in genetic optimization.

The advanced feature in GFIF is implemented in an intelligent agent to handle impulse noise in color images [17]. A color image can be defined as three layers of intensity value in red, green, and blue color. Each layer is filtered individually as if it is a grayscale image. The filtered color image is the aggregation of the partial results of filtering in every color layer. Having advanced features inherited from GFIF, this intelligent image agent can remove salt-and-pepper noise in color images from light to heavy noise condition. This system needs very high computational requirements to work.

Fuzzy system is suitable to model the uncertainty that occurs when both noise removal and detail preservation are required. A rulebase structure is designed to define the probability of a pixel become noisy based on its relationship with the neighborhood. But when the images are highly corrupted, discovering the rulebase structure become quite difficult. To overcome this problem, many techniques based on neuro-fuzzy system are proposed that make use of the neural networks ability to learn from examples [36, 37]. Another alternative is implementing GA in the fuzzy filters [13]. GA may help in defining rulebase structure or finding the optimal parameter setting of a fuzzy filter. Because of its high computational cost, the GA

training is set offline to the main fuzzy filter operation. It means that the training is executed apart from the filtering. The results are then stored and retrieved whenever the process of filtering is done.

2.3 Summary

Despite of its simplicity, the robustness of median filter is proven by its continuous implementation and improvement. Two intrinsic properties that contribute to the success of median filter are edge preservation and efficient noise attenuation with robustness against impulsive noise types. Meanwhile, soft computing techniques are also applied in image enhancement tasks. The most widely applied is fuzzy filters that is able to model human's cognition into a rulebase structure to detect the appearance of noise in images. In most cases, the rulebase is harder to define as the noise corruption rate is higher. To solve this problem, collaboration of fuzzy filters with other techniques in soft computing, such as neural network and genetic algorithms, is inducted. Unlike embedded neural network in fuzzy filter that increases filtering time significantly, genetic-fuzzy filter can have more or less the same filtering time as ordinary fuzzy filter do. It is because the GA training process is separated from the main filtering function.

CHAPTER 3
THEORETICAL BACKGROUNDS

3.1 Fuzzy Systems

Unlike two-valued classical Boolean logic, fuzzy logic is multi-valued. It can handle the concept of partial truth that enables the modeling of uncertainties in natural language. Furthermore, it can express the vagueness in natural language by linguistic terms used to describe objects or situations. It enables the computing systems to understand vague terms, and to reason with the terms [38, 39].

3.1.1 Fuzzy sets

Classical (crisp) set theory imposes a sharp boundary on the set and gives each member distinct status of the membership. Let X be a classical set and x an element. The element x either belongs to X ($x \in X$) or does not belong to X ($x \notin X$). Crisp set A of X is defined as function $f_A(x)$, that is the characteristic function of A .

$$f_A(x) : X \rightarrow 0,1 \tag{3.1}$$

where

$$f_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$

Unlike classical set, elements of a fuzzy set have membership degree to that set. The membership degree of a fuzzy set indicates the certainty (or uncertainty) we have that the element belongs to the set. Fuzzy set A of universe X is defined by function $\mu_A(x)$ called the membership function of set A

$$\mu_A(x) : X \rightarrow [0,1] \quad (3.2)$$

where

$$\mu_A(x) = 1 \text{ if } x \text{ is totally in } A;$$

$$\mu_A(x) = 0 \text{ if } x \text{ is not in } A;$$

$$0 < \mu_A(x) < 1 \text{ if } x \text{ is partly in } A.$$

Fuzzy sets can be defined for finite or infinite domains. For a finite domain X , the fuzzy set can be expressed in the form of an n -dimensional vector. If $X = \{x_1, x_2, \dots, x_n\}$, then fuzzy set A is denoted as

$$A = \{(\mu_A(x_i) / x_i) \mid x_i \in X, i = 1, \dots, n\} \quad (3.3)$$

where the separating symbol / is used to indicate that A is a set of ordered pairs of the membership degrees and their respective elements of the domain [38, 39].

3.1.2 Membership functions

A membership function or characteristic function of the fuzzy set defines the fuzzy set. The function associates a membership degree of each element in the domain to the corresponding fuzzy set. Membership functions for fuzzy sets can be any shape or type determined by experts in the domain over which the sets are defined. Nevertheless, the membership functions must satisfy the following constraints:

- The range of a membership function must be the range $[0,1]$.
- For each $x \in X$, $\mu_A(x)$ must be unique. It means that the same element cannot map to different degrees of membership for the same fuzzy set [38].

There are several types of membership functions. It ranges from the simplest one, triangular membership function, to the ones that include more complex mathematical operation such as Gaussian and sigmoid membership functions. The choice of suitable membership function depends on the nature of the problem that is being solved. The most widely used function is trapezoidal membership function because of its ability to represent human's perception in spite of its simple formulas and computational efficiency [40].

Figure 3.1 shows a trapezoidal membership function, which has four parameters $\{a, b, c, d\}$. The membership function $\mu_A(x)$ can be expressed as follows:

$$\mu_A(x) = \max(\min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}, 0), 0) \quad (3.4)$$

It can also be expressed as

$$\mu_A(x) = \begin{cases} 0, & x \leq a \quad \text{or} \quad x \geq d \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \end{cases} \quad (3.5)$$

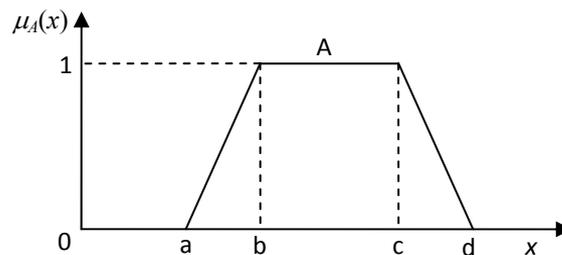


Figure 3.1 Trapezoidal membership function

3.1.3 Fuzzy operators

Relations and operators are defined for fuzzy logic. They enables reasoning about vague information. Let X be the domain or universe, and A and B are sets defined over the domain X . The relations and operations of fuzzy logic are defined as follows [38].

- Equality of fuzzy sets

For fuzzy sets, equality can be concluded if the two sets have the same elements and the membership degree of the sets are equal. That is, the membership functions of the two sets must be the same. Two fuzzy sets A and B are equal ($A=B$) if and only if the sets have the same domain, and $\mu_A(x) = \mu_B(x)$ for all $x \in X$.

- Containment of fuzzy sets

Fuzzy set A is a subset of fuzzy set B ($A \subset B$) if and only if $\mu_A(x) \leq \mu_B(x)$ for all $x \in X$.

- Complement of a fuzzy set (NOT)

The complement of the set A consists of all the elements of set A , but the membership degree differ. Let \bar{A} denote the complement of set A . For all $x \in X$,

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (3.6)$$

- Intersection of fuzzy sets (AND)

The intersection of two fuzzy sets (A AND B) is the set of all elements in the two fuzzy sets, but with membership degree of the new set determined by one of the following operators.

$$\mu_{A \cap B}(x) = \min \{ \mu_A(x), \mu_B(x) \}, \forall x \in X \quad (3.7)$$

or

$$\mu_{A \cap B}(x) = \mu_A(x) * \mu_B(x), \forall x \in X \quad (3.8)$$

- Union of fuzzy sets (OR)

The union of two fuzzy sets (A OR B) is the set of all elements in the two fuzzy sets, but with membership degree of the new set determined by one of the following operators.

$$\mu_{A \cup B}(x) = \max \{ \mu_A(x), \mu_B(x) \}, \forall x \in X \quad (3.9)$$

or

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x), \forall x \in X \quad (3.10)$$

3.1.4 Fuzzy rules

The dynamic behavior of fuzzy systems is characterized by a set of linguistic fuzzy rules. The rules are based on the knowledge and experience of human experts within the domain. A fuzzy rule can be defined as a conditional statement in the form:

IF x is A
THEN y is B

Where x and y are linguistic variables, and A and B are linguistic values determined by fuzzy sets on the universe of discourses X and Y , respectively.

Fuzzy reasoning consists of two distinct parts: evaluating the rule antecedent, and implication or applying the result to the consequent. If the antecedent is true to some degree of membership, then the consequent is also true to the same degree.

A fuzzy rule can have multiple antecedents. All parts of the antecedents are calculated simultaneously and resolved in a single number, using fuzzy operations mentioned in the previous section. A fuzzy rule can also has multiple parts of consequent. All parts of the consequent are affected equally by the antecedent [39].

3.1.5 Fuzzy inference systems

Usually a fuzzy system incorporates several rules that describe expert's knowledge and play off one another. The output of each rule is a fuzzy set, but usually we need to obtain a single number representing the fuzzy system output. To obtain the solution, all outputs of the fuzzy sets are aggregated into a single fuzzy set output, and then the result of the aggregation is defuzzified into a single number.

The Mamdani method is the most commonly used fuzzy inference technique [39]. It is performed in four steps: fuzzification of the input variables, rule evaluation, aggregation of the rule outputs, and defuzzification.

3.1.5.1 Fuzzification

The fuzzification process applies membership functions associated with each fuzzy set. This process is concerned with finding a fuzzy representation of non-fuzzy input values. Input values from the universe of discourse are assigned membership values to fuzzy sets [38].

3.1.5.2 Rule evaluation

In rule evaluation process, the fuzzified inputs are mapped to the rule base to produce a fuzzified output for each rule. The membership degrees of the output sets are determined based on the membership degrees of the input sets and the relationships between them. Fuzzy operators are used to define the relationship between sets [38].

3.1.5.3 Aggregation

Aggregation is the process of unification of the outputs of all rules. We take the membership functions of all rule consequent previously evaluated and combine them into a single fuzzy set [39].

3.1.5.4 Defuzzification

Defuzzification is the last step in fuzzy inference process. The most popular defuzzification method is the centroid technique that finds the point where a vertical line would slice the aggregate set into two equal masses [39]. This center of gravity (COG) is denoted as follows:

$$COG = \frac{\int_{x \in X} \mu_A(x) x dx}{\int_{x \in X} \mu_A(x) dx} \quad (3.11)$$

where X is the universe of discourse. For a discrete domain of a finite number of values, n , the output of the fuzzification process can be denoted as follows.

$$COG = \frac{\sum_{i=1}^n \mu_A(x_i) x_i}{\sum_{i=1}^n \mu_A(x_i)} \quad (3.12)$$

3.2 Genetic Algorithms

The concept of genetic algorithms (GA) was introduced in the early 1970s by John Holland. GA models genetic evolution. The GA can be represented by a sequence of procedural steps for moving from one population of artificial chromosomes to a new population. It includes nature-inspired operations: selection, crossover, and mutation. Each chromosome consists of a number of genes, and each gene is represented by 0 or 1 [39].

Genetic algorithms, along with evolution strategies and genetic programming, are optimization methods in the field of evolutionary computing. They simulate natural evolution by creating a population of individuals, evaluating their fitness, generating a new population through genetic operations, and repeating this process a number of times. The main components of evolutionary algorithms are:

- **encoding** scheme to map solution candidates to chromosomes,

- **fitness function** to evaluate the survival strength of individuals,
- **initialization** of the initial population,
- **selection** operators, and
- **reproduction** operators.

GA utilizes a population of individuals, where each individual represents a candidate solution to the problem. The characteristics of an individual are represented by a chromosome. Each chromosome represents a point in search space. It consists of a number of genes, where each gene represents one characteristic of the individual. A gene represents one parameter of the optimization problem [38].

The mechanism of GA follows the following pseudocode [41].

```

BEGIN
  INITIALIZE population with random candidate solution;
  EVALUATE each candidate;
  REPEAT UNTIL (TERMINATION CONDITION is satisfied) {
    1 SELECT parents;
    2 RECOMBINE pair of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidate;
    5 SELECT individuals for the next generation;
  }
END

```

Figure 3.2 The pseudocode of GA process

3.2.1 Chromosome representation

Finding an appropriate chromosome representation is a very important step in GA. It affects the efficiency and complexity of the search algorithm. It involves defining the genotypes and the mapping from genotype to phenotype [38, 41].

Binary representation

Classically, the representation scheme for GA is binary vectors of fixed length. Although binary vector is frequently used, it has Hamming cliffs disadvantage. A Hamming cliff happens when two numerically adjacent values have bit

representations that are far apart. Gray coding is an alternative bit representation, where the Hamming distance between the representations of successive numerical values is one.

Integer representation

If the problem more naturally maps onto a representation where different genes can take one of a set of values, the binary representations might not be suitable. An example of this is the problem of finding the optimal values for a set of variables that all take integer values.

Real-valued or floating-point representation

This representation is chosen when the values that we want to represent in genes come from a continuous rather a discrete distribution. Each gene is represented by a floating-point value.

Permutation representations

An ordinary GA string allows numbers to occur more than once. Thus, it will not represent valid permutations. For problems that naturally take the form of deciding on the order in which a sequence of events should occur, the most natural representation is as a permutation of a set of integers. As a consequence, we need a new variation operators to preserve the permutation validity, so that each possible allele value occurs exactly once in the solution.

3.2.2 Fitness function

Fitness function represents the task to solve in the evolutionary context. It is a function or procedure for assigning a quality measure to the genotypes. The fitness function should include all criteria to be optimized. It can also reflect the constraints of the problem through penalization of those individuals that violate constraints [38, 41].

3.2.3 Population

Population is a multiset of genotypes. It holds the representation of possible solutions. A population is defined by setting the population size μ , which defines the number of individuals in a population. The diversity of a population is a measure of the number of different solutions present. The first population is started by generating random individuals.

There are two different GA models: the generational model and the steady-state model. In the generational model, a mating pool of parents is selected from a population. The new offspring are created from the selected parents by applying crossover and mutation operations. They are evaluated by a fitness function. After each generation, the whole population is replaced by the new offspring. In the steady state model, only a part of the population is changed at once. The percentage of the population that is replaced is called the generational gap [41].

3.2.4 Parent selection

Ideally, the mating pool of parents taking part in recombination would have exactly the same proportions as the selection probability distribution. In practice, this is not possible because of the finite set of the population. The mating pool of parents is sampled from the selection probability distribution, but will not accurately reflect it.

Roulette wheel algorithm is one of the ways to implement this sampling. The probability distribution can be associated with a roulette wheel, where each slice has a width corresponding to the selection probability of an individual. Selection can be visualized as the spinning of the wheel and testing which slice ends up at the top. The outline of the algorithm is represented in the pseudocode shown in Figure 3.3 [41].

```

BEGIN
  set current_member = 1;
  WHILE ( current_member ≤  $\mu$  ) {
    Pick a random value r uniformly from [0,1];
    set i = 1;
    WHILE (  $a_i < r$  ){
      set i = i + 1;
    }
    set mating-pool[current_member] = parents[i];
    set current_member = current_member + 1;
  }
END

```

Figure 3.3 The pseudocode of roulette wheel algorithm

3.2.5 Crossover

Crossover, or recombination, is considered as one of the most important features in GA. It is a GA operator that uses information from two (or more) parents to create a new individual (child). Crossover operators are usually applied probabilistically according to a crossover rate p_c , which is usually in the range [0.5,1.0]. The crossover probability determines the chance that a chosen pair of parents undergoes the crossover operation. Two parents are selected and then a random variable is drawn from [0,1). If the value is lower than p_c , two offspring are created via recombination of two parents; otherwise they are created asexually by copying the parents [41].

For integer representations, it is normal to use the same set of operators as for binary representations although each gene in integer representations has a higher number of possible allele values than binary representations. Three operators in binary representations are applicable in integer representations. Generally two parents are chosen and then two children are created. However, the operators have been extended to the more general case where a number of parents may be used.

One-point crossover

This is the most basic crossover operator for binary representations. It works by choosing a random number in the range [0, $l-1$], with l is the length of the encoding, and then splitting both parents at this point and creating two children by exchanging the tails.

***N*-point crossover**

N-point crossover is a generalization of one-point crossover. It works by breaking the genes into more than two segments of contiguous genes, and then creating offspring by taking alternative segments from the two parents.

Uniform crossover

Unlike the previous two operators, uniform crossover works by treating each gene independently and making a random choice as to which parent it should be inherited from. In the implementation, a string of l random variables from a uniform distribution over $[0,1]$ is generated. For each position, if the value is below a parameter p_c , the gene is inherited from the first parent; otherwise from the second. The second offspring is the inverse mapping of the first offspring.

3.2.6 Mutation

Mutation is a GA operator that uses only one parent and creates one child by applying some kind of randomized change to the genotype. The form of mutation operation depends on the choice in chromosome representation and a parameter called mutation rate. Different forms of operations based on the representations are described as follows [41].

There are two forms of mutation operators used for integer representations. They can mutate each gene independently with user-defined probability p_m . It is possible to use both forms of operator consecutively for integer-based problems.

Random resetting

This method is analogous to bit-flipping for binary encodings. With probability of p_m a new value is chosen at random from the set of permissible values in each position. It is suitable for encoding of cardinal attributes.

Creep mutation

This operator works by adding a small (positive or negative) value to each gene with probability p_m . It is best applied on ordinal attributes.

3.2.7 Stopping criteria

The formation of new generation is stopped after some criteria are met [41]. Commonly used criteria to stop a GA process are:

- The maximally allowed CPU time elapses.
- The total number of fitness evaluation reaches a given limit.
- The fitness improvement remains under a threshold value for a given period.
- The population diversity drops under a given threshold.

3.2.8 Constraint handling

We need constraint handling because many practical problems are constrained. Naturally, GA operator work without considering the constraints of the candidate solutions. It is possible that the offspring generated by GA operators do not satisfy the constraints for the problem addressed. We can distinguish two types of constrained problems: constrained optimization problems and constraint satisfaction problems.

The following are common options in constraint handling [41].

- The use of penalty functions that reduce the fitness of infeasible solutions, so that the fitness is reduced proportionate to the number of the constraints violated, or to the distance from the feasible region.
- The use of repair functions that take infeasible solutions and generate feasible solutions based on them.

- Restricting search to the feasible region by the use of specific problem representations, suitable initialization, recombination, and mutation operators that. The feasibility of the solutions is ensured and the mapping from genotype to phenotype is not ambiguous.
- The use of decoder functions that manage the mapping from genotype to phenotype so that solutions (phenotypes) are guaranteed to be feasible. Typically, a number of different genotypes may be mapped onto the same phenotype.

3.3 Genetic-fuzzy systems

The combinations of genetic algorithms (GA) and fuzzy systems have come in two ways. The use of fuzzy logic to improve the performance of a GA is known as fuzzy-genetic algorithms, whereas the use of GA to improve the performance of a fuzzy system is called genetic-fuzzy system.

The performance of a fuzzy system depends on its knowledge base that consists of data base and rule base. GA improves the performance of fuzzy systems by tuning their knowledge base. The tuning can be done in one of the following actions: tuning the data base only, tuning the rule base only, or tuning both the data base as well as the rule base. Since optimization process based on GA is computationally expensive, the GA-based tuning is generally carried out off-line, separated from the main process of the fuzzy system. The general structure of a genetic-fuzzy system can be seen in Figure 3.4, which is taken from [42].

The main goal of genetic algorithms integration with fuzzy systems is improving the design process and the performance of the fuzzy systems. Improved fuzzy systems' performance implies higher accuracy and efficiency. If manual selection of fuzzy membership function parameters becomes difficult or takes too much time to attain, GA can be used to find the optimal ones as long as an appropriate fitness function that serves an adequate representative of the optimization process can be defined [43].

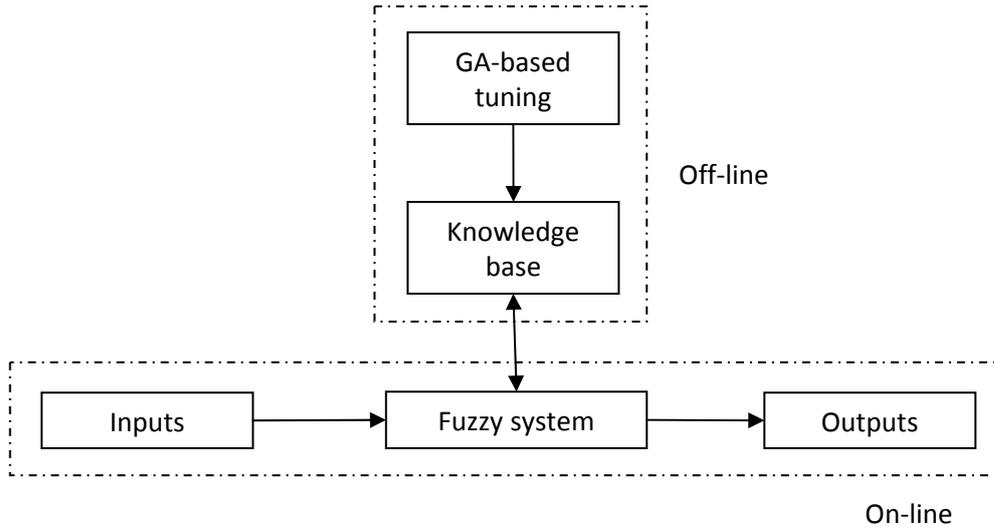


Figure 3.4 Schematic diagram of a genetic-fuzzy system

3.4 Digital Image Representation

An image can be represented in a two dimensional function, $f(i, j)$, where i and j are spatial (plane) coordinates, and the value of f at any pair of coordinates (i, j) is the intensity or gray level of the image at that point. An image is called a digital image when the i, j and the amplitude values of f are all in discrete quantities. A digital image consists of finite elements that have a particular location and value as the results of sampling and quantization of sensed data. They are called picture elements, image elements, pels, or pixels [1]. In the other words, an image can be represented by a two dimensional array of picture elements (pixels), which contain information about locations and intensity values.

A digital image in $M \times N$ size can be written in the following matrix form:

$$f(i, j) = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,2) & \cdots & f(M,N) \end{bmatrix} \quad (3.13)$$

or in the other matrix notation:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \quad (3.14)$$

Equation (3.13) and (3.14) are identical matrices, so $f(i, j)$ and $a_{i,j}$ both equally denote a pixel element of the image.

3.5 Impulsive Noise Models

Digital images are often corrupted by impulse noise due to transmission errors, malfunctioning pixel elements in the camera sensors, faulty memory locations, or timing errors in analog-to-digital conversion [1]. Impulse noise in an image has two major properties: only a certain percentage of image pixels are contaminated, and the intensity value of image corrupted pixel is significantly different from other pixels in its neighborhood. An image corrupted by impulse noise can be described as follows:

$$x_{ij} = \begin{cases} n_{ij}, & \text{with probability } p \\ f_{ij}, & \text{with probability } 1-p \end{cases} \quad (3.15)$$

where x_{ij} denotes a pixel in a noisy image and f_{ij} denotes a noise free image pixel at the location (i, j) . Also, $n_{ij} \in [L_{\min}, L_{\max}]$ is a noisy impulse at the location (i, j) , where L_{\min} and L_{\max} denote the lowest and the highest pixel luminance values within the dynamic range, respectively.

Salt-and-pepper noise is the simplest and the most frequently used impulse noise model, where noisy pixels take either minimal or maximal value, that is $n_{ij} \in \{L_{\min}, L_{\max}\}$ [1]. Abreu et al. [7] introduce random-valued impulse noise of which noise intensities randomly range from the minimum to the maximum intensity values,

that is 0 to 255 for 8-bit grayscale images, that is $n_{ij} \in [L_{\min}, L_{\max}]$. A new model of impulse noise proposed by Petrović [8] incorporates both aforementioned impulse noise models. The corrupted image is contaminated by salt-and-pepper and uniform impulse noise in the same probabilities. Let x_{ij} be an image containing mixed impulse noise,

$$x_{ij} = \begin{cases} n_{ij}^{unif}, & \text{with probability } \frac{p}{2} \\ n_{ij}^{snp}, & \text{with probability } \frac{p}{2} \\ f_{ij}, & \text{with probability } 1-p \end{cases} \quad (3.16)$$

where $n_{ij}^{unif} \in [L_{\min}, L_{\max}]$ and $n_{ij}^{snp} \in \{L_{\min}, L_{\max}\}$. Also, f_{ij} denotes a noise-free image pixel at the location (i, j) . This type of noise model is claimed to be more realistic because the impulse noise occurs as a result of disturbances induced by noise signals with random amplitudes. The amplitudes of the impulse noise could fall into the dynamic range allowed as well as out of the range. The noise will be saturated to the maximum or minimum value if it is out of the range. Otherwise, it will appear as uniform impulse noise if it is within the dynamic range. Therefore, this noise model is more suitable to evaluate the performance of impulse noise filters.

3.6 Median of the Absolute Deviations (MAD)

Median of the absolute deviations from the median (MAD), a robust estimator of the variance, was used to separate noisy pixels from image details [23, 44]. MAD has been successfully used for the estimation of local image variance in the presence of impulse noise [18]. Let x_{ij} denote pixels with coordinates (i, j) in noisy image, and X_{ij} denote the set of pixels in $(2K+1) \times (2K+1)$ neighborhood window W centered at x_{ij} .

$$X_{ij} = \{x_{i-K, j-K}, \dots, x_{ij}, \dots, x_{i+K, j+K}\} \quad (3.17)$$

The median of the image is defined as

$$m_{ij} = \text{median}(X_{ij}) \quad (3.18)$$

The deviation image is the absolute difference between the noisy and its median image. It is defined by

$$d_{ij} = |x_{ij} - m_{ij}| \quad (3.19)$$

MAD presents the median of absolute deviations from the median. It is defined by

$$MAD_{ij} = \text{median}(|X_{ij} - \text{median}(X_{ij})|) \quad (3.20)$$

Figure 3.5 and Figure 3.6 show the deviation image (d) and the median of the absolute deviations (MAD) from 256×256 Lena image corrupted by 20% mixed impulse noise. Sharp dots visible in Figure 3.5 depict the noise removed by median filter, while the softer dots indicate fine details lost because of the median filtering. As can be seen from Figure 3.6, despite the impulse noise in the image, MAD can retain the details information of the image very well. We can exploit MAD to distinguish noisy and noise-free pixels in an image.

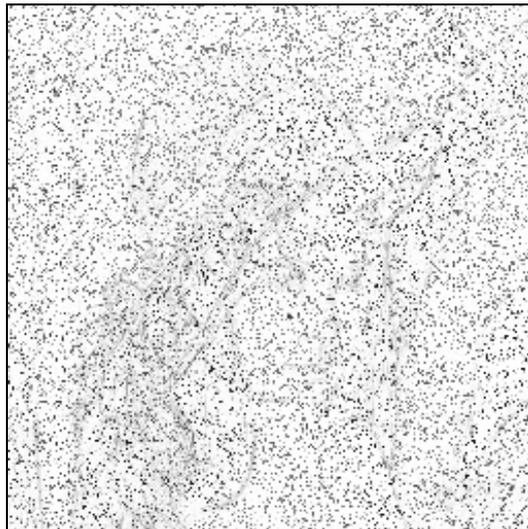


Figure 3.5 Deviation image between noisy image x_{ij} and its median image (in inverted color)



Figure 3.6 MAD image of the noisy Lena image for 5×5 window size (in inverted color)

3.7 Rank-Ordered Absolute Differences (ROAD)

Rank-ordered absolute differences (ROAD) is an image statistic for detecting random-valued impulse noise [19]. Let X_{ij}^0 denote the set X_{ij} with x_{ij} excluded. Define D_{ij} as the absolute difference between the gray-level values in X_{ij} and x_{ij} .

$$D_{ij} = |X_{ij}^0 - x_{ij}| \quad (3.21)$$

Sort D_{ij} values in increasing order, and let r_{ij}^k be the k th smallest value in D_{ij} . Then, ROAD can be defined by

$$ROAD_{ij}^m = \sum_{k=1}^m r_{ij}^k \quad (3.22)$$

where $2 \leq m \leq (2K+1)^2 - 2$. If the noise ratio is higher than 25%, it is suggested that we use the 5-by-5 windows and $m = 12$. Otherwise, use 3-by-3 windows and $m = 4$ [45].

ROAD can identify most of the noisy pixels in the images corrupted by random-valued impulse noise. In [45], ROAD is incorporated into a filter to remove Gaussian and impulse noise. It performs well in removing Gaussian and also impulse noise, but when the noise level is high, it blurs images seriously. In [19], it is combined with an edge-preserving regularization to build a two-stage method for denoising random-valued impulse noise up to 60% probability. An experiment done by Gangadhar [46] shows the effectiveness of ROAD in detecting impulse noise. Combined with simple filtering methods, such as median filter, weighted median filter, and center weighted median filter, the nonlinear filtering techniques are found to be effective in eliminating the impulse noise besides preserving the image features quite satisfactorily.

A local statistic, rank-ordered logarithmic differences (ROLD) is derived from the ROAD method. It amplifies the difference between noisy and noise-free pixels so that the noisy pixels can be detected, even when the noisy pixels have close value to the noise-free ones [19]. Because ROAD and ROLD are not accurate at image edges although they perform well in detecting most impulse noise, Yu [47] proposed a rank-ordered relative differences, an impulse noise detector that is more accurate than ROAD and ROLD even at image edges. The local statistic is calculated from the difference between a noisy image and a reference image. The main weakness of this method is its need of a reference image, which is the noise-free version of the filtered image. Therefore, this method might not be useful in the real case.

3.8 Switching scheme filtering

Median filter is one of the most popular approaches for impulse noise removal where every pixel in an image is replaced by its median of certain window size [1]. In median filter, all pixels are treated in the same way, whether they are noisy or not. It has been proven as a powerful method to remove impulse noise, but at the same time it removes edges and other details in uncorrupted pixels [22]. Adaptive median filter, weighted median filter, center weighted median filter, etc. are some modifications to median filter to minimize the detail loss caused by the filtering.

One of the strategies to preserve edges and details is the switching scheme. Let x_{ij} and y_{ij} denote the pixels with coordinates (i, j) in a noisy and a filtered image, respectively. The switching filter concept is defined by

$$y_{ij} = M_{ij} \cdot \xi(x_{ij}) + (1 - M_{ij}) \cdot x_{ij} \quad (3.23)$$

where $\xi(x_{ij})$ is the estimated value of the corrupted pixel x_{ij} , and M_{ij} is the binary noise map, with 1's indicating noisy pixels on respected coordinates. The filter removes noise in two steps: detecting impulse noise in an image and replacing the detected pixels with estimated values. The noise-free pixels are kept unfiltered. Therefore, excessive filtering that removes edges and details can be avoided. Generally, the detection employs some local statistics of the neighborhoods around the processed pixels, such as median, weighted median, rank-order thresholding, normalized mean, fuzzy reasoning, or neural networks [23].

CHAPTER 4

METHODOLOGY

We propose a fuzzy switching scheme filtering method based on MAD and ROAD statistical properties. Fuzzy inference system is incorporated to detect the noise level of each pixel in an image. The parameters included in the fuzzy inference system are optimized using genetic algorithm (GA). Since GA training is a resource-intensive process, we arrange it separated from the main process of filtering. After the training is completed, the results are then stored in a knowledge base. The filtering process obtains the training results, i.e. the fuzzy parameters, from the knowledge base without redoing the training process each time we want to filter.

This filter is designed to remove mixed impulse noise, which means that it can also be used to remove salt-and-pepper and uniform impulse noise. We define three main processes in the method: fuzzy inference system, fuzzy switching scheme filtering, and fuzzy parameters optimization. Figure 4.1 depicts the overall process of the proposed method.

4.1 Fuzzy Noise Detection System

In order to avoid excessive filtering that can cause detail loss in images, noisy and noise-free pixels should be treated differently. If a pixel is detected as noisy, its value will be replaced by an estimate. Otherwise, it will be left as is. We use fuzzy inference system to determine whether a pixel can be categorized as noisy. The detection system yields the pixels' degree to be categorized as noisy, which ranges in $[0,1]$.

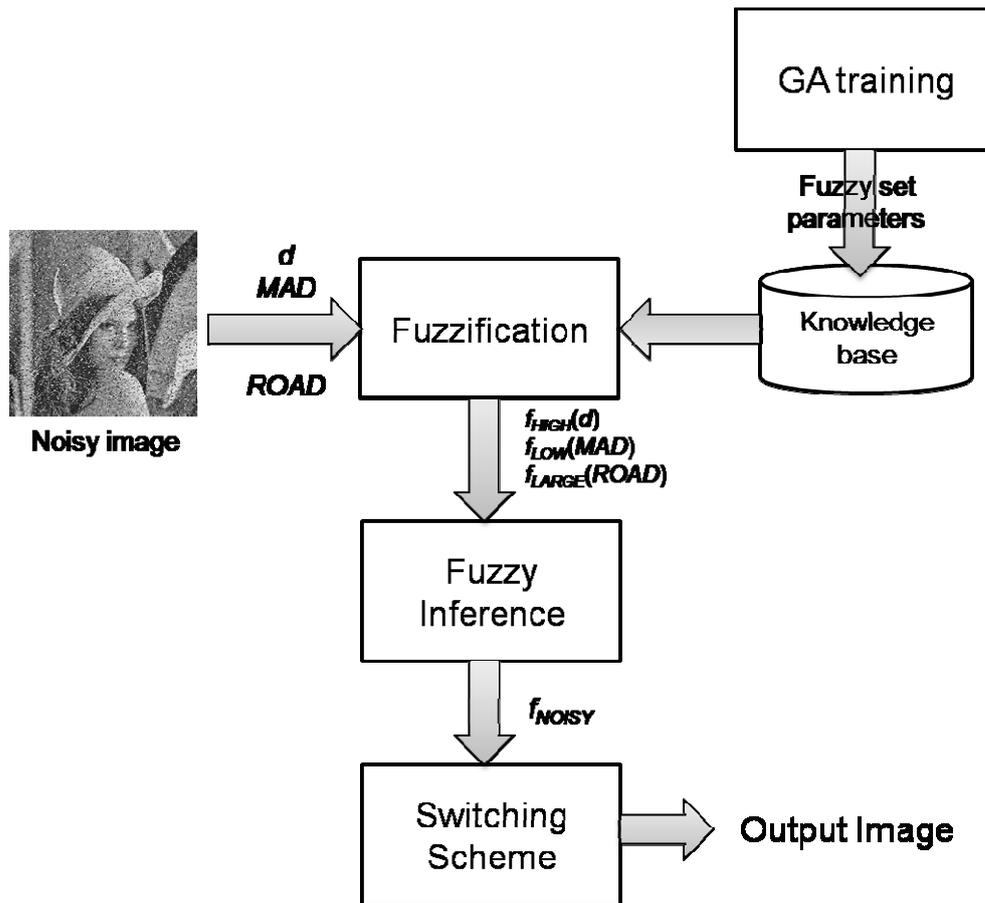


Figure 4.1 The diagram of the system

4.1.1 MAD and ROAD Application

The first step in detecting the noise is extracting the value of MAD and ROAD from a noisy image. We will also use the deviation image d in this process. To obtain d , MAD, and ROAD values from an image, we use the formulas in equation (3.19), (3.20), and (3.22), respectively. The window size for calculating d , MAD, and ROAD depends on the noise level of the input. Generally, for low to medium noise probability, we can use 3×3 or 5×5 windows. They are the size in which median filters can remove noise in low to medium probability without losing too many details in the filtered image.

4.1.2 Fuzzy Inference System

Fuzzification process maps MAD into LOW, d into HIGH, and ROAD into LARGE fuzzy sets. In this process, the trapezoidal membership function is chosen to define the fuzzy sets. As can be seen in equation (3.4) and (3.5), each fuzzy set has four parameters $\{a, b, c, d\}$ that determine the mapping to fuzzy membership functions.

Figure 4.2, Figure 4.3 and Figure 4.4 indicate the histogram comparison between d , MAD, and ROAD values of noise-free and noisy pixels from Lena images corrupted by mixed impulse noise in 0.2 probability. To obtain these histograms, first, we generated 0.2 mixed impulse noise in a 256×256 grayscale Lena image. The calculation of d , MAD, and ROAD are then applied to the image. Since we have the original version of 256×256 grayscale Lena image, we can locate the noisy pixels out of the noise-free ones. The histogram analysis is done separately for the noisy pixels as well as the noise-free pixels.

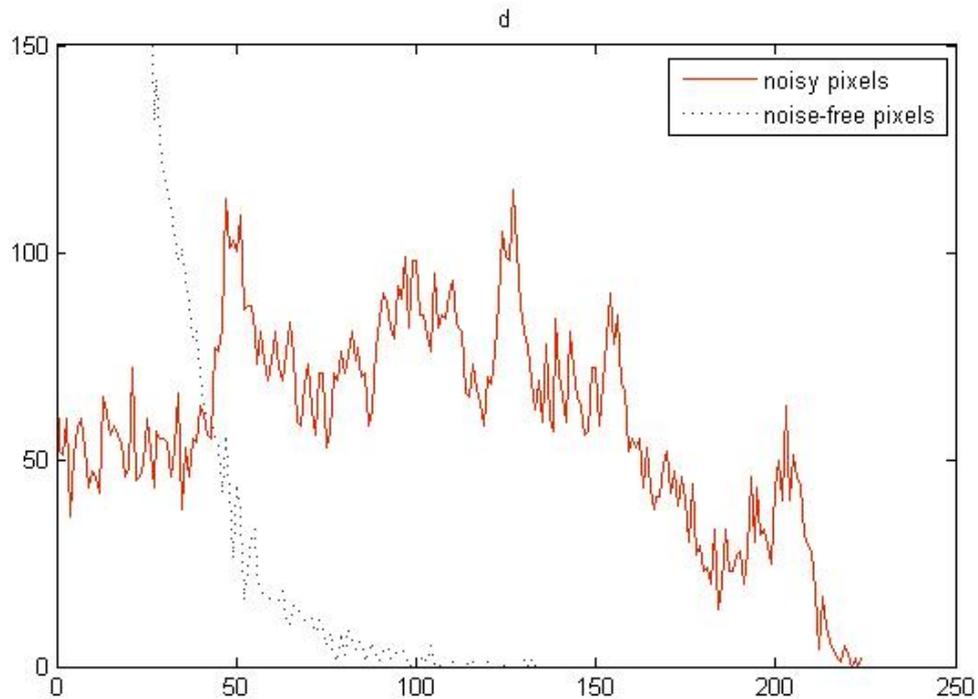


Figure 4.2 Histogram comparison of d values of noisy and noise-free pixels from Lena image corrupted by mixed impulse noise 0.2

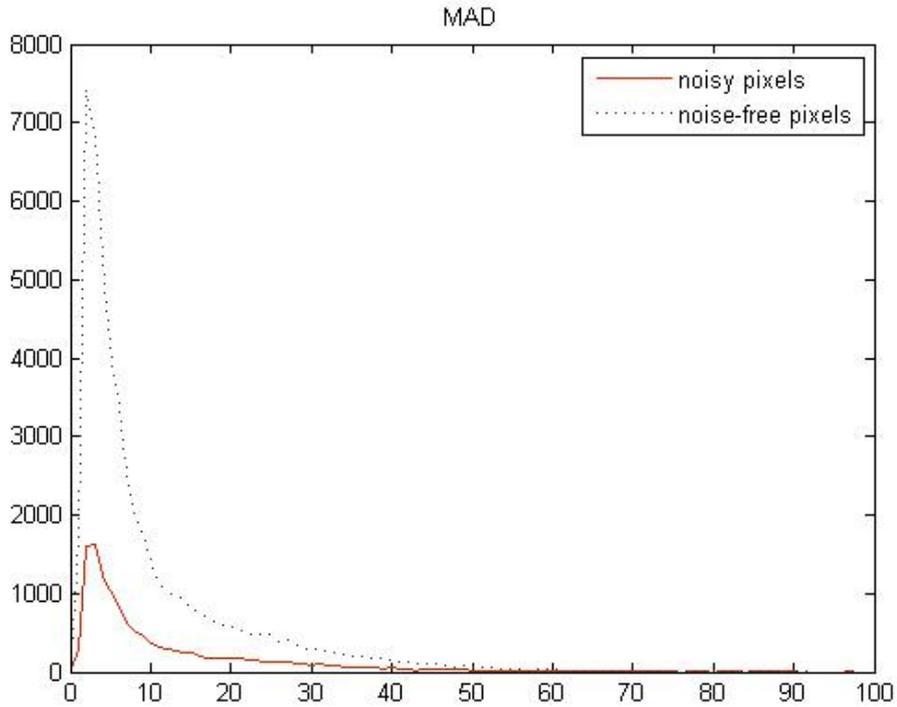


Figure 4.3 Histogram comparison of MAD values of noisy and noise-free pixels from Lena image corrupted by mixed impulse noise 0.2

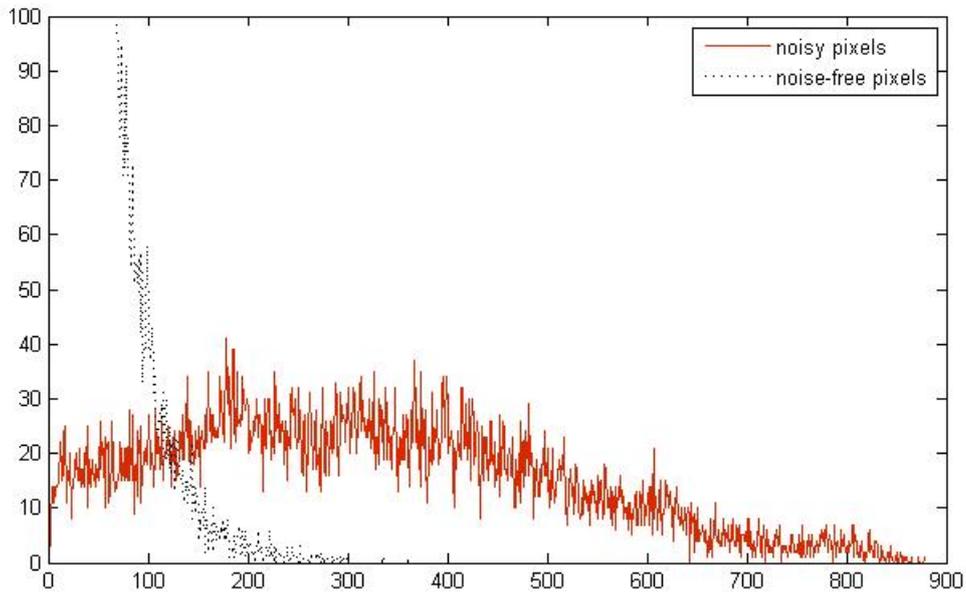


Figure 4.4 Histogram comparison of ROAD values of noisy and noise-free pixels from Lena image corrupted by mixed impulse noise 0.2

Normally for noise free pixels, the values of d mostly fall in the range 0-50. Otherwise, noisy pixels has higher values of d . Lower value on d mostly came from the image details loss caused by median filtering. The higher ones come from the relatively high difference between noisy image and its median filtered image.

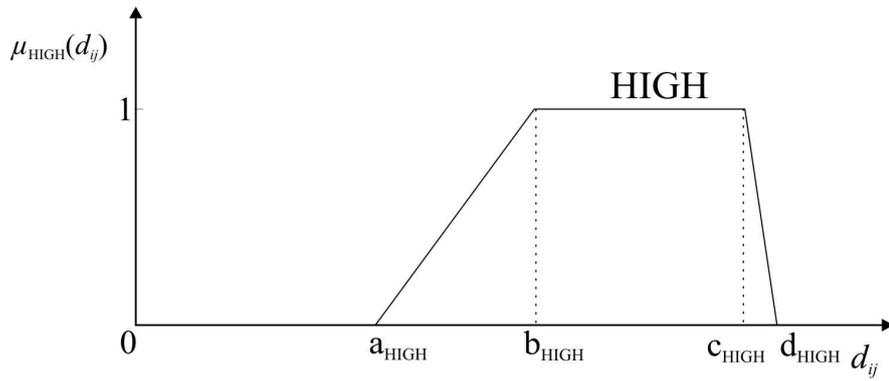
MAD value is highly related to the details in an image. The MAD values of noisy pixels are not much different from the noise-free ones. MAD can retain details and edges information from an image although the image becomes noisy until a certain degree. Its value is relatively low regardless of the noise conditions. MAD and d are incorporated to detect the noise and leave the details of an image [23]. If a pixel has high d value, it is most probably a noisy pixel. However, if a pixel has low MAD value, it can be considered as an image details that we do not want to filter.

Figure 4.4 shows the comparison of ROAD values of noise-free and noisy pixels from Lena image corrupted by mixed impulse noise in 0.2 probability. It has quite similar features with Figure 4.2, the histogram of d values. Noise free pixels tend to have lower ROAD value, while the corrupted ones have higher ROAD value.

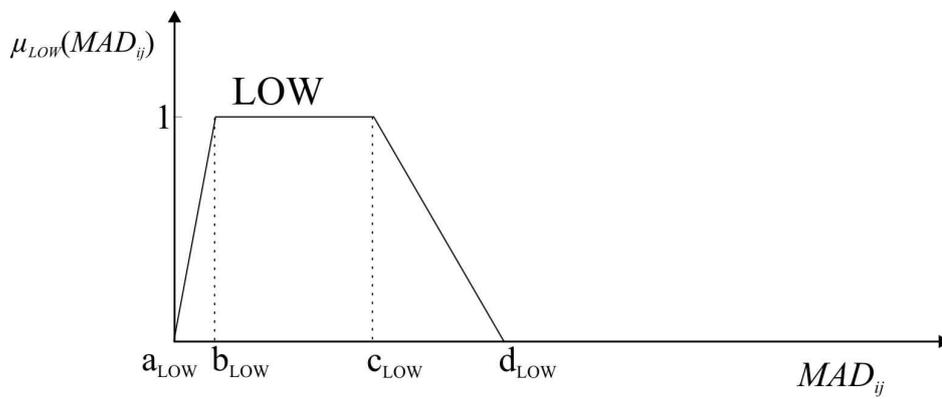
Based on the above rationale, we can map the value of d , MAD, and ROAD to the suitable fuzzy sets shown in Figure 4.5. Let x_{ij} be a pixel of an input image located in (i, j) , d_{ij} , MAD_{ij} , and $ROAD_{ij}$ are the values of d , MAD and ROAD of x_{ij} , then we can translate the above information into the following fuzzy rule.

IF ((d_{ij} is HIGH) AND (MAD_{ij} is LOW)) OR ($ROAD_{ij}$ is LARGE) THEN x_{ij} is NOISY

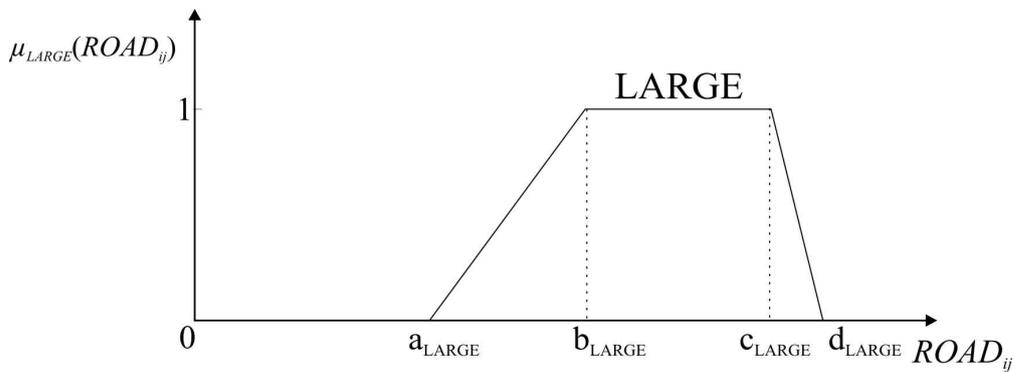
The new fuzzy set $\mu_{NOISY}(x_{ij})$ resulted from the above inference is the noise detector. Its membership value ranges in $[0,1]$. Higher value of $\mu_{NOISY}(x_{ij})$ indicates higher probability of x_{ij} become noisy pixel. The degree of this noise probability determines the effects of median filtering applied to the respective pixel.



(a)



(b)



(c)

Figure 4.5 Fuzzy sets HIGH, LOW, and LARGE for d , MAD, and ROAD, respectively

Normally, a fuzzy filter needs to have several rules fired on each pixel and its neighboring pixels. By employing d , MAD, and ROAD we only need one rule to be fired because those image statistical properties can resume useful pixel information in the neighborhood.

4.2 Fuzzy Switching Scheme Filtering

Switching scheme filtering treats noisy and noise-free pixels differently [18]. The switching scheme filtering combined with the output of fuzzy noise detection result in a fuzzy switching scheme filtering. Originally, as can be seen in equation (3.23), the value of noise detection $\xi(x_{ij})$ is either 1 or 0, i.e. representations of noisy and noise-free conditions. If a pixel detected as noisy, its value will be replaced by an estimate value. In fuzzy noise detection system, the results of detection is in fuzzy membership function that have values in $[0,1]$. Therefore, the filtered pixels are switched softly based on the detection values. The filtered pixel y_{ij} is defined as follows

$$y_{ij} = \mu_{NOISY}(x_{ij}) \cdot m_{ij} + (1 - \mu_{NOISY}(x_{ij})) \cdot x_{ij} \quad (4.1)$$

A pixel x_{ij} will be fully replaced with its median m_{ij} if it is detected as noisy ($\mu_{NOISY}(x_{ij}) = 1$). Otherwise, it will be kept at its original value. The pixels are filtered proportional to the value of $\mu_{NOISY}(x_{ij})$.

4.3 Fuzzy Parameter Optimization using Genetic Algorithms

In this section, we apply the genetic algorithms to find the best parameter for fuzzification process as explained in the previous section. Defining a fixed threshold of d , MAD, and ROAD to differentiate noisy pixels from the noise-free ones is not trivial. For noise in low amplitudes, the value of d , MAD, and ROAD are relatively similar to their values for noise-free pixels in edges or small details. GA is used to approach the optimal parameter set for the fuzzification process.

4.3.1 Gene Encoding

The genes represent 12 parameters of trapezoidal membership functions in fuzzy sets HIGH, LOW, and LARGE. Figure 4.6 depicts the gene encoding in a chromosome. The first four genes are parameters of HIGH fuzzy set. The second and the last four genes are parameters of LOW and LARGE fuzzy sets, respectively.

| | | | | | | | | | | | |
|------------|------------|------------|------------|-----------|-----------|-----------|-----------|-------------|-------------|-------------|-------------|
| a_{HIGH} | b_{HIGH} | c_{HIGH} | d_{HIGH} | a_{LOW} | b_{LOW} | c_{LOW} | d_{LOW} | a_{LARGE} | b_{LARGE} | c_{LARGE} | d_{LARGE} |
|------------|------------|------------|------------|-----------|-----------|-----------|-----------|-------------|-------------|-------------|-------------|

Figure 4.6 The gene encoding

GA operators applied over the genes may cause the gene values to fall outside the allowed value of fuzzy membership function parameters. For each trapezoidal membership function, there are four parameters $\{a, b, c, d\}$. The membership function is meaningful only if a , b , c and d are in ascendant order. Thus, to maintain meaningful fuzzy membership functions, the following restrictions on the parameters are applied.

$$\begin{aligned}
a_{HIGH} < b_{HIGH} \leq c_{HIGH} < d_{HIGH} \\
a_{LOW} < b_{LOW} \leq c_{LOW} < d_{LOW} \\
a_{LARGE} < b_{LARGE} \leq c_{LARGE} < d_{LARGE}
\end{aligned} \tag{4.2}$$

We use special representations and operators to maintain feasible solutions in the GA [48]. The genes are coded using integer codification because all the possible values of d , MAD, and ROAD are integers and it is easier to implement the restrictions mentioned by equation (4.2) in integer gene representation.

4.3.2 Genetic Learning Scheme

Figure 4.7 shows the genetic learning process to obtain the optimal set of fuzzy membership function parameters. We adopted the genetic training in Evolutionary Image Enhancement (EIE) by Cho [9]. Initial population is formed by generating random numbers within the range of possible values of d , MAD, and ROAD. Since the first four genes are parameters of HIGH fuzzy set for d , they must be in the range of [0,255]. The second four ones are parameters of LOW fuzzy set for MAD. So, their values are in the range of [0,127]. The last four genes represents the parameters of LARGE fuzzy set for ROAD, of which maximum values are $m*255$, where m is the number of highest absolute difference taken into calculation of ROAD in equation (3.22).

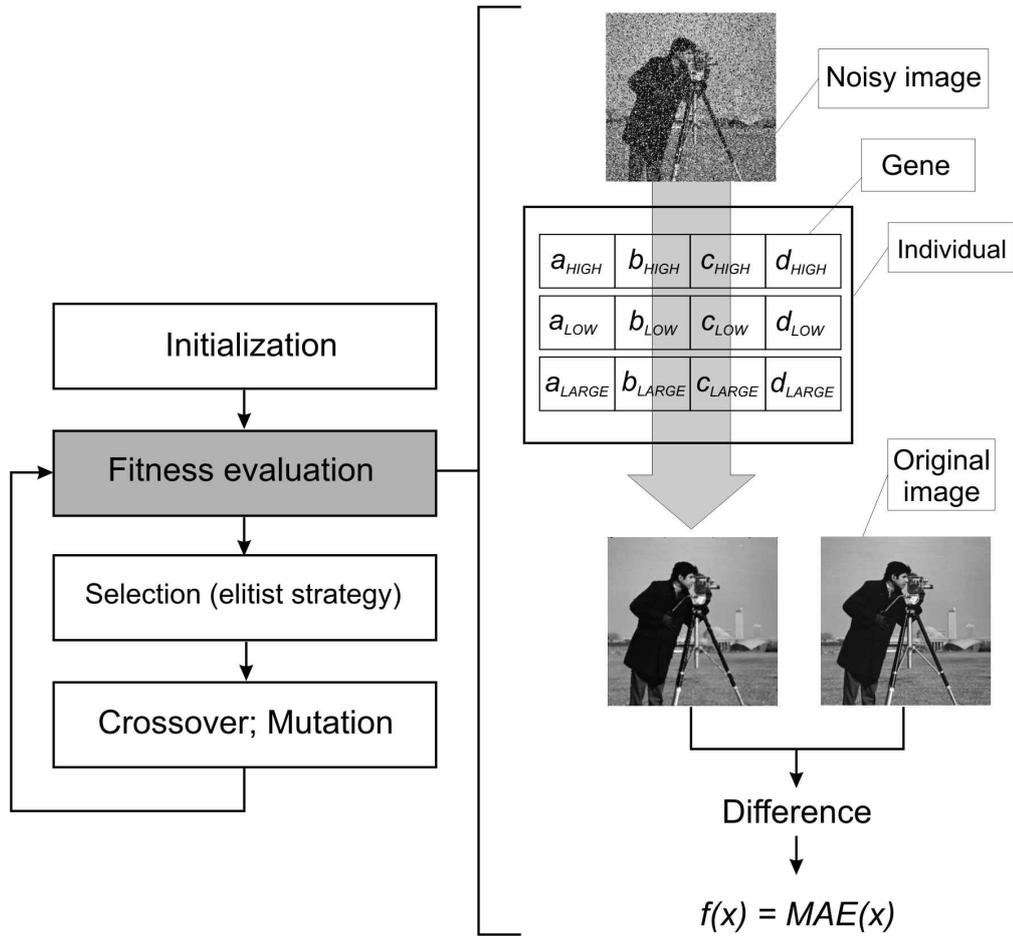


Figure 4.7 The genetic learning process

The fitness of the chromosomes are tested by applying them into the fuzzy sets in the proposed filtering system to filter a noisy image. The output of the filtering is then compared to the original version of the noisy image. The mean absolute error (MAE) is adopted as the fitness function to evaluate the difference between the filtered image and the original image. The training process aims at minimizing the value of the fitness function, because smaller MAE means smaller difference between two images. Equation (4.3) shows the MAE adopted in the fitness function for each gene.

$$f(x) = MAE(x) \quad (4.3)$$

where MAE of two images can be defined as:

$$MAE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |y_{i,j} - x_{i,j}| \quad (4.4)$$

The image used in the training process may be different from the one that will be actually filtered. Once the training results obtained, they can be used many times for filtering by saving them in a knowledge base.

Parents that are selected to produce new offsprings are chosen using the roulette wheel method. We also apply the elitist strategy, in which some of the best chromosomes are reserved to be used in the next generation. The crossover operator is single-point crossover, which means a crossover site is randomly chosen within the gene length. The mutation is done by adding or subtracting the gene values with small numbers [42].

Applying GA operators may cause the genes to produce unfeasible solutions. To maintain feasible solutions, we apply repair algorithm that make changes to unfeasible individuals so that the gene values are kept in the allowed values [48]. After operators applied, for each fuzzy set, genes representing its parameters are sorted ascending from a to d . If values after operations fall out of the allowed range, the values are saturated to the maximum or minimum allowed values.

CHAPTER 5

RESULTS AND ANALYSIS

To evaluate the performance, the method has been tested in MATLAB R2007a that runs under Windows XP SP3 operating system. There were three MATLAB toolboxes involved: fuzzy logic toolbox, genetic algorithm and direct search toolbox, and image processing toolbox. GA training to obtain the optimal parameter set was implemented using genetic algorithm and direct search toolbox. Image processing toolbox was utilized to handle some basic image operations. Because only simple fuzzy inference involved, the fuzzy logic toolbox was not used in the implementation of the proposed method. Hard-coded fuzzy inference system was used in implementation of the proposed filter.

We also considered some outstanding filters based on soft computing as well as median and average filters as samples of classical filtering techniques to evaluate the proposed filter's performance. The filters involved in the tests were median filters, average filters, evolutionary image enhancement (EIE) [9], weighted fuzzy mean filter (WFM) [11, 12], genetic-fuzzy image filter (GFIF) [13], and fuzzy random impulse noise reduction (FRINR) [20]. Fuzzy logic toolbox was used to apply other filters based on Sugeno fuzzy inference system, such as WFM and GFIF.

Median and average filters were involved in the test to know the performance of the proposed method compared to the classical methods of filtering, i.e. median and average filters. The filtering windows size chosen were 3×3 and 5×5 for each median and average. Those are the most common window size to filter low to medium noise corruptions.

Evolutionary image enhancement (EIE) [9] is a filtering method based on genetic algorithms. The GA chooses the best combination of five spatial filters from 70

spatial filters available. GA training is done for every noise probability. Therefore, if there are five different noise probabilities to be filtered, five different trainings must be done for each noise probability. The image used as the training image is the cameraman image in 256×256 8-bit grayscale. The GA training parameters were as follows: 500 generations, 30 populations, 5 gene length, 0.7 crossover rate, and 0.05 mutation rate.

Lee's weighted fuzzy mean filter (WFM) [11, 12] is a filtering method based on Sugeno fuzzy inference system. Pixels in image are categorized into fuzzy sets based on their intensity, such as dark, medium, and bright. It needs one Sugeno inference system for each fuzzy set and one more for fuzzy estimator. In WFM, there is no training needed. It only generates parameters from a sample image to be kept in image knowledge base. These parameters are retrieved from the image knowledge base during the filtering process. For this experiment, a cameraman image in 256×256 8-bit grayscale has been used to generate the WFM parameters.

Genetic-based fuzzy image filtering (GFIF) is a further improvement of AWFM and WFM [13] where the number of fuzzy sets involved become five sets. Another significant improvement is the fuzzy sets parameter optimization using genetic algorithm. Specific parameter encoding and genetic learning are defined to solve the optimization problem. The training image used for this experiment is a 256×256 8-bit grayscale cameraman image. The GA training parameters were 100 generations, 20 population size, 2 elite counts, 0.8 crossover fractions, and 0.01 mutation probabilities.

Fuzzy rules are implemented in fuzzy random impulse noise reduction (FRINR) method [29]. The method consists of two separated phases: two-step fuzzy detection mechanism and a fuzzy filtering. The first detection operates in local neighborhoods, while the second gathers more information from extended neighborhoods. The fuzzy filter only processes the pixels detected by both detector. This method is intended to remove random-valued impulse noise from corrupted images. There is no need to define any parameters nor trainings to make the filter work.

The filter performance is evaluated in terms of noise attenuation, overall visual quality, detail and edge preservation. We evaluate the filtering results based on

observed visual quality as well as quantitative error measurements Mean Absolute Error (MAE) and Peak Signal-to-Noise Ratio (PSNR).

For an output image y and an input image x , both of them have $m \times n$ size, the value of MAE and PSNR can be defined as follows [44].

$$MAE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |y_{i,j} - x_{i,j}| \quad (5.1)$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{\max(x)^2}{MSE} \right) \quad (5.2)$$

where

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \|y_{i,j} - x_{i,j}\|^2 \quad (5.3)$$

Images included in the tests are standard test images, such as Lena, cameraman, and other images depicted in Figure 5.1.

These standard test images are widely available in the Internet. One of the source is the homepage of fuzzy image processing, University of Waterloo [49]. The page contains 12 standard test images shown above. It was last accessed on November 2nd, 2009.



(a) walkbridge



(b) cameraman



(c) livingroom



(d) jetplane



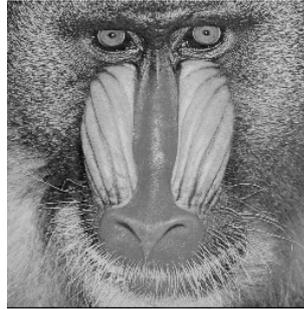
(e) lake



(f) house



(g) Lena



(h) mandrill



(i) woman blonde



(j) pirate



(k) woman dark hair



(l) peppers

Figure 5.1 Standard test images

5.1 Obtaining the optimal parameter set

Fuzzy set parameters were trained using 256×256 8-bit grayscale cameraman image. The GA training parameters were 100 generations, 20 population size, 2 elite counts, 0.8 crossover fractions, and 0.01 mutation probabilities. The training results are the parameters of the fuzzy sets HIGH, LOW and LARGE.

Figure 5.2 shows the evolution of the best and average fitness values over generations for 0.2 noise probability. With the current GA settings, the best solution's

fitness starts to converge after the 50th generation. It means that there is no significant change in the best fitness value from 50th generation onwards. We can conclude that the maximum number of generation set in 100 is sufficient for GA training to reach its convergence, i.e. to find the optimal solutions.

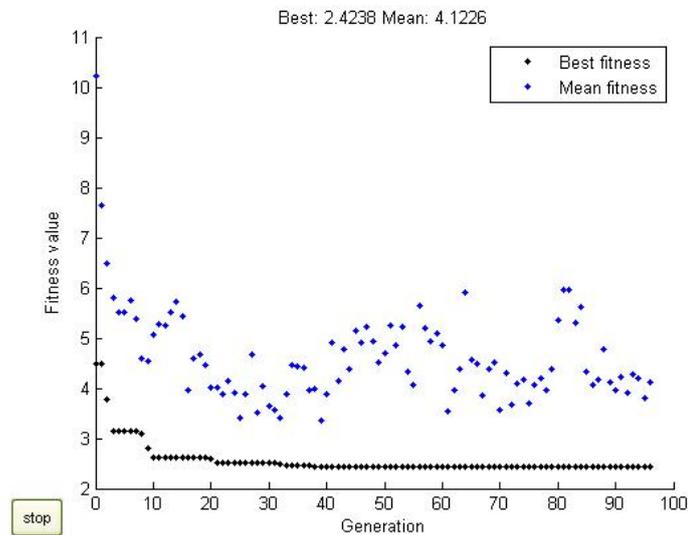


Figure 5.2 The fitness evolution in the GA training

GA training was done for each noise level. For each noise level, the training was repeated three times to make sure the results obtained are the best ones. Table 5.1 shows the training results for each noise level. The best results for each noise level can already be used for filtering. It is user preference to use different parameter sets for different noise levels or to use the same parameter set for all noise probabilities. Adjusting parameter set for handling different noise condition may lead to better filtering results, but sometimes it is not practical to change the parameters over time, especially in the real case when we do not know exactly the value of the noise probability.

The simpler and more practical way is to choose one parameter set to be used in all noise probabilities. The best results for each noise level are tested to assess their performance in removing noise from 0.1 to 0.5 probabilities. Table 5.2 shows the testing results. The shaded cells are the best results for each noise level. Set #4 got the most shaded cells that means it can handle most noise probabilities. Therefore, it has

been chosen as the parameter set to filter all noise probabilities. The final training results are shown in Table 5.3.

Table 5.1 GA training results

| Noise prob. | Training | Fuzzy set parameters | | | | | | | | | | | | Fitness | Remark |
|-------------|----------|----------------------|-----|-----|-----|-----|-----|-----|-----|------|-----|-----|------|---------|--------|
| | | MAD | | | | d | | | | ROAD | | | | | |
| 0.1 | 1st | 0 | 0 | 186 | 188 | 140 | 212 | 255 | 255 | 103 | 155 | 846 | 864 | 1.26 | |
| | 2nd | 0 | 0 | 78 | 125 | 101 | 162 | 236 | 236 | 121 | 157 | 777 | 832 | 1.24 | Set#1 |
| | 3rd | 0 | 0 | 185 | 206 | 131 | 149 | 156 | 225 | 116 | 151 | 959 | 971 | 1.24 | |
| 0.2 | 1st | 0 | 0 | 55 | 255 | 67 | 110 | 225 | 255 | 96 | 140 | 459 | 748 | 2.43 | |
| | 2nd | 0 | 0 | 99 | 133 | 104 | 104 | 255 | 255 | 106 | 137 | 707 | 1006 | 2.42 | Set#2 |
| | 3rd | 144 | 224 | 249 | 255 | 5 | 52 | 185 | 201 | 91 | 142 | 933 | 936 | 2.48 | |
| 0.3 | 1st | 0 | 0 | 187 | 255 | 64 | 135 | 208 | 255 | 98 | 134 | 730 | 923 | 3.83 | |
| | 2nd | 0 | 0 | 49 | 147 | 76 | 87 | 215 | 255 | 113 | 117 | 833 | 865 | 3.78 | Set#3 |
| | 3rd | 0 | 0 | 182 | 213 | 23 | 127 | 255 | 255 | 89 | 103 | 593 | 662 | 3.88 | |
| 0.4 | 1st | 0 | 0 | 134 | 255 | 44 | 69 | 255 | 255 | 65 | 130 | 314 | 713 | 5.27 | |
| | 2nd | 0 | 0 | 123 | 255 | 23 | 105 | 254 | 255 | 85 | 99 | 638 | 1020 | 5.31 | |
| | 3rd | 0 | 0 | 94 | 255 | 43 | 69 | 228 | 255 | 67 | 127 | 628 | 817 | 5.27 | Set#4 |
| 0.5 | 1st | 0 | 0 | 123 | 178 | 26 | 46 | 244 | 255 | 92 | 130 | 168 | 178 | 7.19 | Set#5 |
| | 2nd | 0 | 0 | 142 | 235 | 43 | 67 | 201 | 225 | 75 | 100 | 872 | 951 | 7.31 | |
| | 3rd | 0 | 0 | 141 | 219 | 15 | 69 | 255 | 255 | 73 | 147 | 547 | 1020 | 7.32 | |

Table 5.2 Test to find the optimal parameter for general cases

| Set | Noise probability | | | | | | | | | |
|-----|-------------------|-------|------|-------|------|-------|------|-------|------|-------|
| | 0.1 | | 0.2 | | 0.3 | | 0.4 | | 0.5 | |
| | MAE | PSNR | MAE | PSNR | MAE | PSNR | MAE | PSNR | MAE | PSNR |
| #1 | 1.24 | 31.78 | 2.51 | 28.75 | 4.16 | 26.12 | 6.41 | 23.65 | 9.84 | 21.14 |
| #2 | 1.27 | 31.60 | 2.42 | 29.12 | 3.84 | 26.88 | 5.66 | 24.73 | 8.31 | 22.56 |
| #3 | 1.33 | 31.07 | 2.43 | 29.08 | 3.78 | 27.01 | 5.43 | 25.18 | 7.89 | 23.15 |
| #4 | 1.64 | 30.12 | 2.65 | 28.89 | 3.86 | 27.31 | 5.27 | 25.88 | 7.30 | 24.12 |
| #5 | 2.02 | 28.81 | 2.97 | 27.75 | 4.08 | 26.60 | 5.33 | 25.41 | 7.19 | 24.04 |

GA as a stochastic process could generate different solutions for different runs. In the GA training, we use one sample image that has more or less similar features with the test images. The training is done independently for each noise level. The best solutions for each level of noise are then assessed to get the best parameter set that works on all noise probability.

Table 5.3 The final GA training results

| a _{HIGH} | b _{HIGH} | c _{HIGH} | d _{HIGH} | a _{LOW} | b _{LOW} | c _{LOW} | d _{LOW} | a _{LARGE} | b _{LARGE} | c _{LARGE} | d _{LARGE} |
|-------------------|-------------------|-------------------|-------------------|------------------|------------------|------------------|------------------|--------------------|--------------------|--------------------|--------------------|
| 0 | 0 | 94 | 255 | 43 | 69 | 228 | 255 | 67 | 127 | 628 | 817 |

5.2 Filter performance in preserving image details

When a filter is applied to an image, it is not only removing noise but also removing some informative details from an image. We apply the proposed filter to a noise-free image to know the filter's 'destructive' effects when it is applied to original, noise-free image. An 8-bit 256×256 Lena image is chosen to be the test image. The result is compared to the one filtered by median filter in 5×5 windows. The median filter in 5×5 windows is the estimator for our proposed filter. It means that originally the image is filtered by the median filter. Our filter has a detector that is able to measure the probability that a pixel can be categorized as noisy. This detector determines how much the estimator, that is the median filter, affects the output value. So, by comparing the results of our proposed filter and the median filter, we can observe the details and edges preserved by the proposed filter.

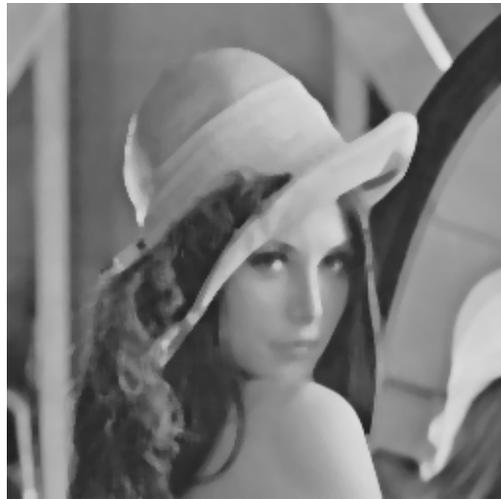
The filtered images and the difference images in Figure 5.3 shows how much details are lost after the filter operation. Figure 5.3a is an original, noise-free 8-bit Lena image in 256×256 pixels. Without adding any noise, it is filtered by the proposed filter in Figure 5.3b and median filter in Figure 5.3c. The difference between the filtered images and their original images are shown in Figure 5.3d and Figure 5.3e. The image filtered by the proposed filter shows more detail preservation and higher image visual quality than the median filtered image. The MAE and PSNR values in Table 5.4 also affirm this result. The proposed filter can reach MAE value of 0.73 while median filter can only reach 5.58. The PSNR value of the proposed filter is 19.29% higher than median filter that means more detail preservation gained by the proposed method compared to median filter.



(a) Original 256x256 8-bit grayscale Lena image



(b) original image filtered by the proposed filter



(c) original image filtered by median in 5x5 windows



(d) difference between (a) and (b) in inverted and enhanced color



(e) difference between (a) and (c) in inverted and enhanced color

Figure 5.3 The application of the proposed filter and median filter to a noise-free Lena image

Table 5.4 MAE and PSNR values of noise-free Lena image

| | MAE | PSNR |
|------------|------|-------|
| Proposed | 0.73 | 32.04 |
| Median 5×5 | 5.58 | 26.86 |

Figure 5.4 shows the comparison among the original image, noisy image, median 5×5 filtered noisy image, and the noisy image filtered by the proposed method. The effect of median filter in 5×5 windows is obvious. The noise is removed, but the image also lost many of its details. We can see that our detection can selectively replace noisy pixels with its median, while keeping the rest less affected by the median filter. The detection scheme in the proposed filter significantly reduces detail loss caused by the median filters.

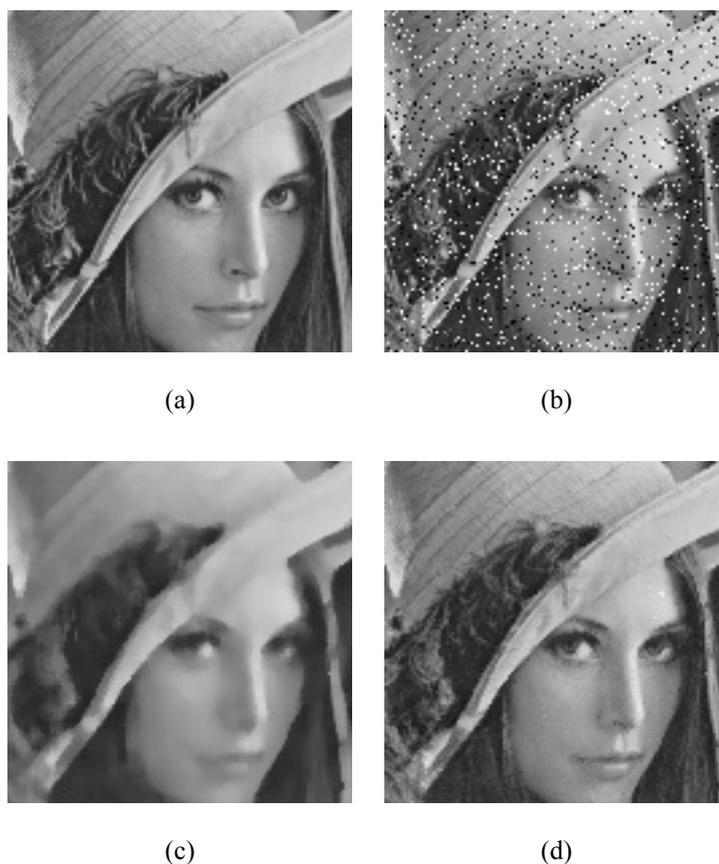


Figure 5.4 (a) 128×128 8-bit grayscale Lena image, (b) image corrupted by 0.1 mixed impulse noise, (c) noisy image (b) filtered by median filter in 5×5 windows, and (d) noisy image (b) filtered by the proposed method.

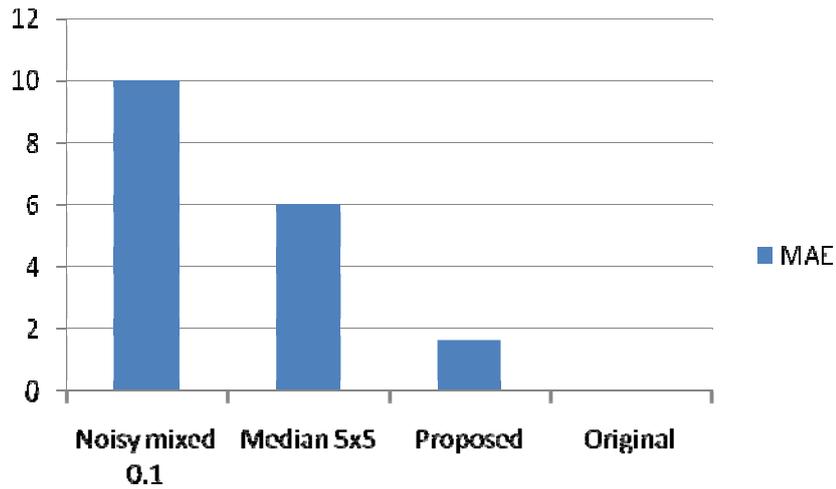


Figure 5.5 MAE graph comparison among Lena image corrupted by 0.1 mixed impulse noise, filtered by 5x5 median filter, filtered by the proposed filter and its original image.

Figure 5.5 confirms the ability of the proposed filter to remove noise and retain more detail information of the image rather than the median filter. MAE value of two similar images is zero. Greater MAE value means greater difference between two images. The MAE value of Lena image corrupted by mixed impulse noise in 0.1 probability compared to the original Lena image is 9.99. The median filter in 5x5 can restore the image to the MAE value of 6.02. The proposed filter restored even more image information to the MAE value of 1.64, very close to 0, that indicates better performance in image restoration.

5.3 Filter performance in impulsive noise removal

5.3.1 Filter performance in removing mixed impulse noise

We tested the performance of the proposed filter to 8-bit Lena image in 256x256 pixels. Mixed impulse noise was generated on the image in 0.1 to 0.5 probabilities, with 0.1 increments. The proposed filter and the other filters mentioned in the previous section filtered the noisy images. The optimal parameter set from the GA

training section was used in this test. We analyze the results by observing the image visual quality and evaluating the MAE and PSNR error measurements.

Table 5.5 MAE values of Lena images corrupted by mixed impulse noise

| Filter Name | Noise Probability | | | | |
|-------------|-------------------|-------|-------|-------|-------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Proposed | 1.64 | 2.65 | 3.86 | 5.27 | 7.30 |
| Median 3x3 | 4.40 | 5.23 | 6.56 | 8.99 | 13.02 |
| Median 5x5 | 6.02 | 6.44 | 7.02 | 7.74 | 8.99 |
| Average 3x3 | 11.43 | 16.46 | 20.86 | 25.22 | 28.82 |
| Average 5x5 | 11.34 | 14.84 | 18.31 | 22.09 | 25.04 |
| GFIF | 2.04 | 4.30 | 7.13 | 10.33 | 14.76 |
| WFM | 7.13 | 8.83 | 11.06 | 13.66 | 17.18 |
| FRINR | 2.32 | 2.82 | 4.67 | 9.33 | 18.79 |
| EIE | 3.32 | 4.49 | 5.74 | 7.92 | 8.96 |

The proposed method outranked the other methods in MAE and PSNR evaluation. The graphs of MAE and PSNR values of the method are shown in Figure 5.6 and Figure 5.7. Lower MAE value and higher PSNR value indicate higher performance of a filter. It gained the lowest MAE and the highest PSNR for every noise probability tested. The complete comparison of filtering results among the methods can be seen in Figure 5.8.

Table 5.6 PSNR values of Lena images corrupted by mixed impulse noise

| Filter Name | Noise Probability | | | | |
|-------------|-------------------|-------|-------|-------|-------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Proposed | 30.12 | 28.89 | 27.31 | 25.88 | 24.12 |
| Median 3x3 | 28.53 | 27.32 | 24.89 | 21.90 | 19.10 |
| Median 5x5 | 26.40 | 25.98 | 25.29 | 24.51 | 23.73 |
| Average 3x3 | 23.43 | 20.85 | 18.98 | 17.49 | 16.39 |
| Average 5x5 | 23.49 | 21.78 | 20.19 | 18.76 | 17.73 |
| GFIF | 29.60 | 24.69 | 24.79 | 23.79 | 21.44 |
| WFM | 25.17 | 23.44 | 21.66 | 20.21 | 18.49 |
| FRINR | 29.45 | 28.13 | 24.20 | 19.49 | 15.46 |
| EIE | 30.06 | 26.39 | 26.47 | 23.59 | 22.90 |

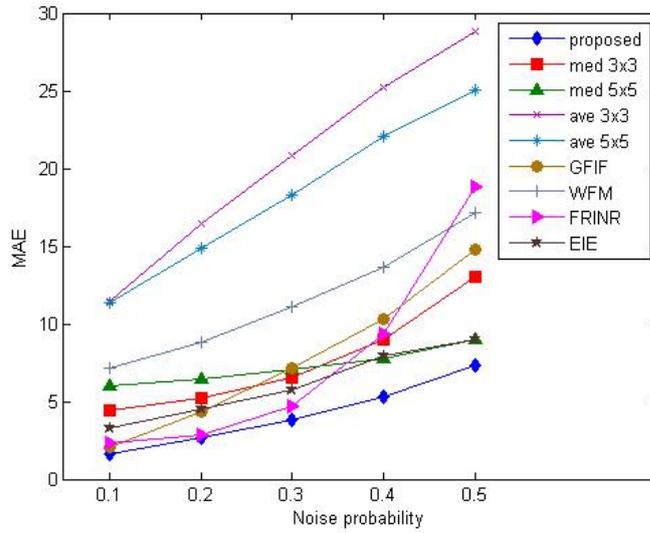


Figure 5.6 MAE curves of Lena images corrupted by mixed impulse noise

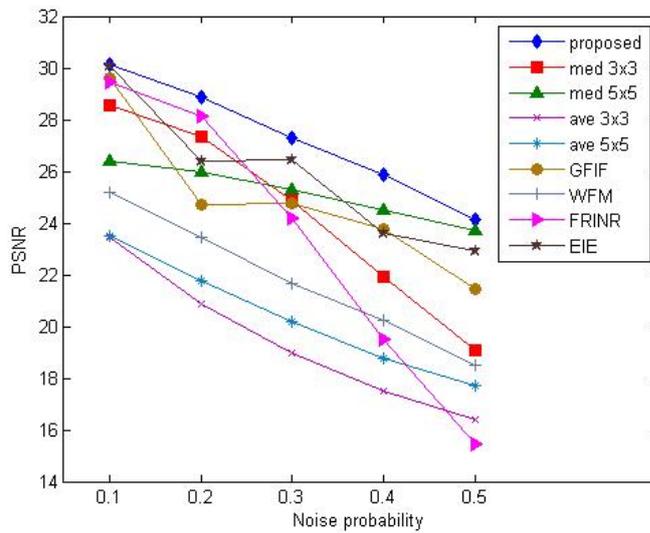


Figure 5.7 PSNR curves of Lena images corrupted by mixed impulse noise

It can be seen from Figure 5.8 that the proposed filter can remove mixed impulse noise while preserving more details than median filters do. The average filters are not suitable to remove this kind of noise. They can barely reduce the noise appearance. There are still many noises left in images filtered by WFM and GFIF. Both methods are originally designed to remove salt-and-pepper impulse noise. They detect the appearance of noise in extreme minimum and maximum values, but failed to detect noise values between them. EIE is a flexible filter that can be trained for any type of noise, including mixed impulse noise, while FRINR is designed to remove uniform

impulse noise. EIE and FRINR seem to be able to preserve the image features, but they still leave some noticed noise.

Another set of tests was also done to other standard test images. Mixed impulse noise in 0.1 to 0.5 probabilities were added to the images. Then, all the images were filtered by the proposed filter. Images in Figure 5.9 depict standard test images corrupted by mixed impulse noise in 0.2 probability. They have been filtered by the proposed filter. The noise is successfully removed with minimum edge and detail loss.



Figure 5.8 Noisy 256x256 8-bit grayscale Lena image corrupted by 0.2 mixed impulse noise filtered by: (a) the proposed filter, (b) 3x3 average filter, (c) 5x5 average filter, (d) 3x3 median filter, (e) 5x5 median filter, (f) EIE, (g) WFM, (h) GFIF, and (i) FRINR.



(a)



(b)



(c)



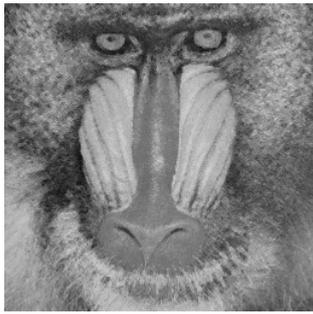
(d)



(e)



(f)



(g)



(h)



(i)



(j)



(k)

Figure 5.9 Noisy standard test images corrupted by 0.2 mixed impulse noise filtered by the proposed filter

There are differences in MAE and PSNR values for each picture in Figure 5.9. It is observed that higher MAE comes from image that has small details as thin as one pixel, such as mandrill's feather in mandrill image and small branches in walkbridge image. It implies that the proposed filter can remove noise and retain image details better in less complicated images. Despite the various MAE and PSNR values, the filter produces more or less similar visual quality for wide variety of images.

Table 5.7 MAE values of the standard test images corrupted by mixed impulse noise

| Image name | Noise probability | | | | |
|----------------|-------------------|------|------|-------|-------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| walkbridge | 4.77 | 6.54 | 8.68 | 10.86 | 13.75 |
| livingroom | 3.10 | 4.56 | 6.05 | 7.75 | 9.90 |
| jetplane | 2.78 | 3.78 | 5.08 | 6.80 | 9.05 |
| lake | 3.79 | 5.12 | 6.74 | 8.59 | 11.20 |
| house | 0.82 | 1.59 | 2.73 | 3.82 | 5.56 |
| lena | 1.64 | 2.65 | 3.86 | 5.27 | 7.30 |
| mandril | 5.77 | 7.56 | 9.56 | 11.73 | 14.06 |
| woman_blonde | 2.23 | 3.16 | 4.19 | 5.51 | 7.15 |
| pirate | 2.12 | 3.32 | 4.71 | 6.38 | 8.63 |
| woman_darkhair | 0.66 | 1.32 | 2.16 | 3.14 | 5.00 |
| peppers | 1.39 | 2.34 | 3.53 | 5.03 | 7.07 |

Table 5.8 PSNR values of the standard test images corrupted by mixed impulse noise

| Image name | Noise probability | | | | |
|----------------|-------------------|-------|-------|-------|-------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| walkbridge | 24.85 | 23.87 | 22.75 | 21.81 | 20.68 |
| livingroom | 26.68 | 25.43 | 24.53 | 23.51 | 22.41 |
| jetplane | 26.28 | 25.63 | 24.59 | 23.38 | 22.05 |
| lake | 25.33 | 24.45 | 23.51 | 22.47 | 21.25 |
| house | 35.10 | 32.28 | 29.32 | 27.75 | 25.49 |
| lena | 30.12 | 28.89 | 27.31 | 25.88 | 24.12 |
| mandril | 23.21 | 22.32 | 21.78 | 21.20 | 20.89 |
| woman_blonde | 27.03 | 26.39 | 25.70 | 24.80 | 23.79 |
| pirate | 28.52 | 27.19 | 26.16 | 24.95 | 23.39 |
| woman_darkhair | 35.01 | 32.45 | 30.99 | 29.42 | 26.47 |
| peppers | 30.93 | 29.09 | 27.67 | 25.86 | 24.03 |

5.3.2 Filter performance in removing salt-and-pepper noise

Mixed impulse noise is a mix of salt-and-pepper and uniform impulse noise. We also wanted to know the filter's performance in removing salt-and-pepper noise as well as uniform impulse noise. This section and the next section discuss the experiment results and analysis of the filter's performance to remove two aforementioned types of noise.

From 0.1 to 0.5 salt-and-pepper noise were added to 8-bit grayscale Lena image. These images were filtered by the proposed filter as well as the other filters. The results were evaluated in terms of subjective evaluation on image visual quality, MAE and PSNR error measurements.

Based on MAE and PSNR values in Table 5.9 and, it is shown that GFIF outranked the proposed filter and the others. GFIF, a genetic-fuzzy filter that originally designed to remove salt-and-pepper noise, has done very well in removing noise in maximum and minimum possible pixel values. In the highest noise probability, WFM, as the core process of GFIF, also outranked the proposed filter. Nonetheless, the proposed filter still outranked the other methods and had a stable performance over various noise probabilities tested.

Table 5.9 MAE values of Lena images corrupted by salt-and-pepper noise

| Filter Name | Noise Probability | | | | |
|-------------|-------------------|-------|-------|-------|-------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Proposed | 1.54 | 2.59 | 3.99 | 5.87 | 8.88 |
| Median 3x3 | 4.38 | 5.21 | 7.17 | 11.39 | 18.78 |
| Median 5x5 | 5.94 | 6.30 | 6.78 | 7.52 | 9.10 |
| Average 3x3 | 12.99 | 18.64 | 23.24 | 27.88 | 31.87 |
| Average 5x5 | 12.11 | 15.82 | 19.16 | 23.05 | 26.22 |
| GFIF | 0.93 | 1.69 | 2.57 | 3.41 | 4.42 |
| WFM | 5.91 | 5.98 | 6.18 | 6.36 | 6.68 |
| FRINR | 2.13 | 2.76 | 6.62 | 15.74 | 30.57 |
| EIE | 3.37 | 4.94 | 5.61 | 8.81 | 10.32 |

Table 5.10 PSNR values of Lena images corrupted by salt-and-pepper noise

| Filter Name | Noise Probability | | | | |
|-------------|-------------------|-------|-------|-------|-------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Proposed | 30.68 | 28.73 | 26.76 | 24.66 | 21.77 |
| Median 3x3 | 28.98 | 26.77 | 22.57 | 18.43 | 15.14 |
| Median 5x5 | 26.51 | 25.99 | 25.75 | 24.59 | 22.17 |
| Average 3x3 | 22.55 | 19.89 | 18.14 | 16.91 | 15.74 |
| Average 5x5 | 23.10 | 21.27 | 19.83 | 18.38 | 17.34 |
| GFIF | 32.83 | 30.85 | 29.04 | 29.78 | 26.64 |
| WFM | 26.50 | 26.40 | 26.02 | 25.78 | 25.30 |
| FRINR | 30.08 | 27.17 | 20.58 | 15.76 | 12.45 |
| EIE | 29.42 | 24.23 | 26.68 | 21.30 | 19.76 |



Figure 5.10 Noisy 256x256 8-bit grayscale Lena image corrupted by 0.2 salt-and-pepper noise filtered by: (a) the proposed filter, (b) 3x3 average filter, (c) 5x5 average filter, (d) 3x3 median filter, (e) 5x5 median filter, (f) EIE, (g) WFM, (h) GFIF, and (i) FRINR.

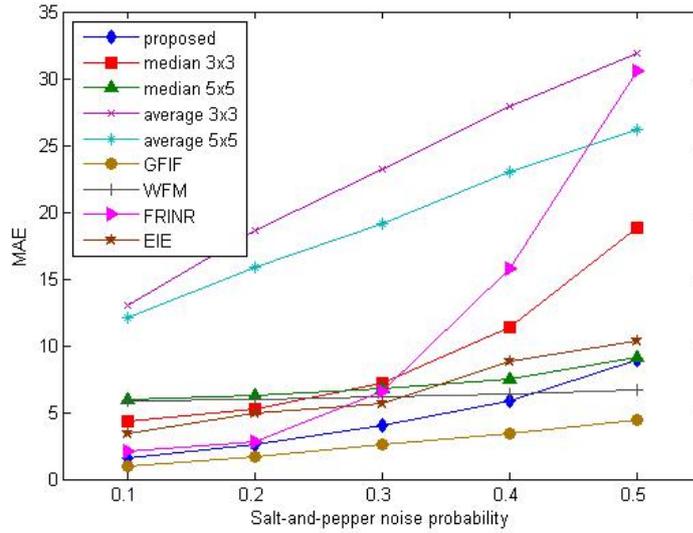


Figure 5.11 MAE curves of Lena images corrupted by salt-and-pepper noise

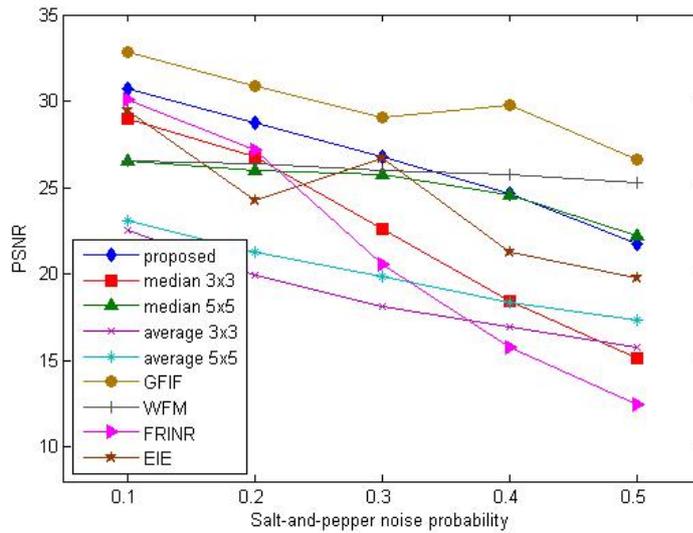


Figure 5.12 PSNR curves of Lena images corrupted by salt-and-pepper noise

5.3.3 Filter performance in removing uniform impulse noise

Unlike salt-and-pepper that has extreme value of noise, uniform impulse noise corrupts images with random values of noise. Uniform impulse noise from 0.1 to 0.5 noise probabilities were added to 8-bit Lena image. The filtering results of the proposed filter as well as the other filters were analyzed. The MAE and PSNR values of the filtering results can be seen in the Table 5.11 and Figure 5.14. They are

visualized in the graphs in Figure 5.14 and Figure 5.15. The filtering results in 0.2 uniform noise probability is shown in Figure 5.13.

Table 5.11 MAE values of Lena images corrupted by uniform impulse noise

| Filter Name | Noise Probability | | | | |
|-------------|-------------------|-------|-------|-------|-------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Proposed | 1.68 | 2.83 | 4.09 | 5.68 | 8.24 |
| Median 3x3 | 4.43 | 5.28 | 6.65 | 8.81 | 12.44 |
| Median 5x5 | 6.06 | 6.60 | 7.35 | 8.44 | 10.48 |
| Average 3x3 | 9.92 | 13.76 | 17.78 | 21.56 | 25.42 |
| Average 5x5 | 10.65 | 13.64 | 16.99 | 20.34 | 23.77 |
| GFIF | 3.17 | 6.90 | 11.40 | 16.27 | 21.80 |
| WFM | 8.22 | 11.15 | 14.60 | 18.55 | 22.91 |
| FRINR | 2.43 | 3.02 | 4.46 | 7.41 | 13.01 |
| EIE | 3.35 | 4.42 | 6.01 | 8.19 | 10.07 |

Table 5.12 PSNR values of Lena images corrupted by uniform impulse noise

| Filter Name | Noise Probability | | | | |
|-------------|-------------------|-------|-------|-------|-------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Proposed | 30.14 | 28.29 | 26.90 | 25.31 | 23.37 |
| Median 3x3 | 28.61 | 27.08 | 25.04 | 22.78 | 20.12 |
| Median 5x5 | 26.45 | 25.90 | 25.14 | 24.19 | 22.52 |
| Average 3x3 | 24.53 | 22.17 | 20.29 | 18.79 | 17.45 |
| Average 5x5 | 23.82 | 22.31 | 20.82 | 19.44 | 18.15 |
| GFIF | 25.61 | 24.62 | 22.63 | 22.43 | 20.33 |
| WFM | 24.12 | 21.72 | 19.70 | 18.07 | 16.68 |
| FRINR | 28.94 | 27.63 | 25.43 | 22.19 | 18.72 |
| EIE | 29.82 | 26.95 | 26.34 | 23.68 | 22.28 |

In removing uniform impulse noise, the proposed filter outranked the other methods, including the FRINR that is specially designed to remove uniform impulse noise. FRINR can remove most of the noise, but the visual quality is not satisfactory. It still left some noticeable noise after filtering. The proposed filter also surpassed the GFIF and WFM that completely failed to remove this type of noise. Its detection scheme can detect impulse noise in random values. Average filters are not suitable to remove any types of impulse noises, while median filters and EIE are able to remove uniform impulse noise with noticeable detail loss.



Figure 5.13 Noisy 256x256 8-bit grayscale Lena image corrupted by 0.2 uniform impulse noise filtered by: (a) the proposed filter, (b) 3x3 average filter, (c) 5x5 average filter, (d) 3x3 median filter, (e) 5x5 median filter, (f) EIE, (g) WFM, (h) GFIF, and (i) FRINR.

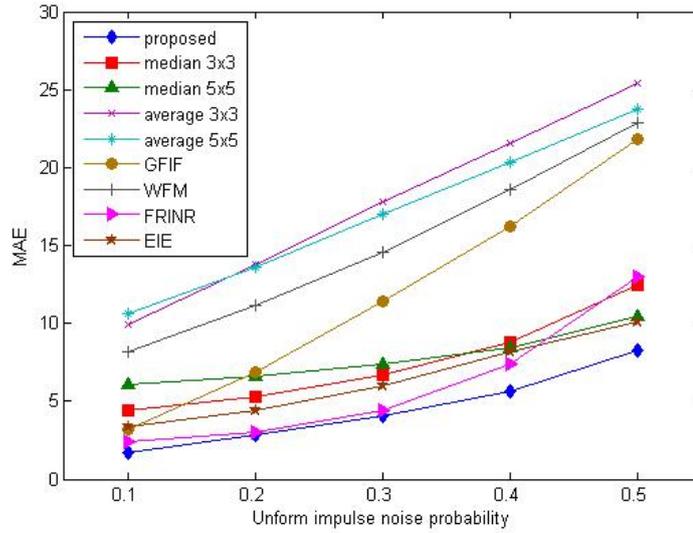


Figure 5.14 MAE curves of Lena images corrupted by uniform impulse noise

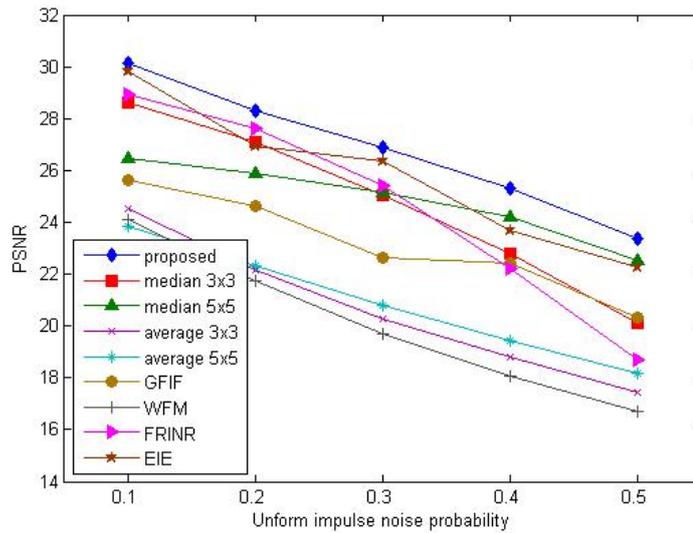


Figure 5.15 PSNR curves of Lena images corrupted by uniform impulse noise

The proposed filter is originally intended to remove mixed impulse noise. Its performance in removing other types of impulse noise, such as salt-and-pepper and uniform impulse noise, is as good as in removing mixed impulse noise. The comparison of filter's performance in removing three types of impulse noise can be seen in Table 5.13 and that contain the MAE and PSNR values, and visualized in graph in Figure 5.16 and Figure 5.17. The performance is stable regardless the type of impulse noise.

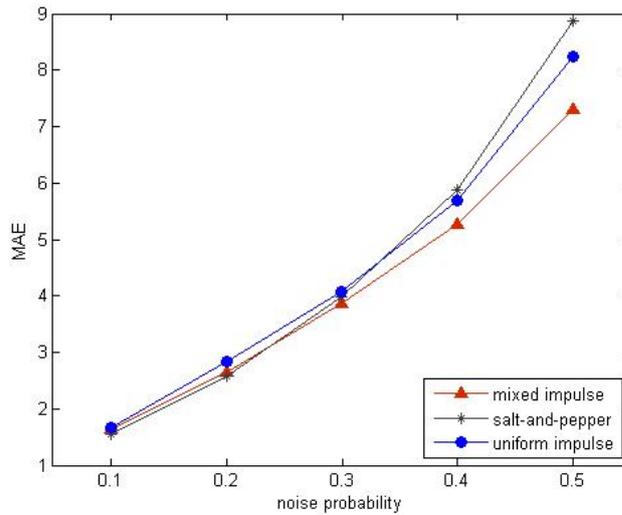


Figure 5.16 MAE curves comparison of filtered Lena images corrupted by mixed impulse, salt-and-pepper, and uniform impulse noise

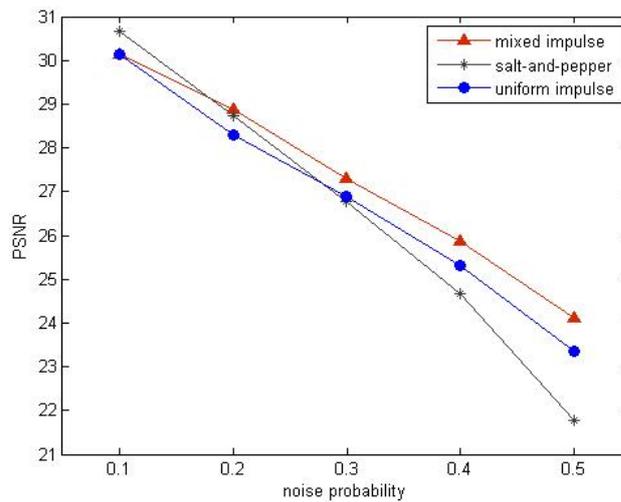


Figure 5.17 PSNR curves comparison of filtered Lena images corrupted by mixed impulse, salt-and-pepper, and uniform impulse noise

Table 5.13 MAE values comparison of filtered Lena images corrupted by mixed impulse, salt-and-pepper, and uniform impulse noise

| Noise type | Noise Probability | | | | |
|-----------------------|-------------------|------|------|------|------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Mixed impulse noise | 1.64 | 2.65 | 3.86 | 5.27 | 7.30 |
| Salt-and-pepper noise | 1.54 | 2.59 | 3.99 | 5.87 | 8.88 |
| Uniform impulse noise | 1.68 | 2.83 | 4.09 | 5.68 | 8.24 |

Table 5.14 PSNR values comparison of filtered Lena images corrupted by mixed impulse, salt-and-pepper, and uniform impulse noise

| Noise type | Noise Probability | | | | |
|-----------------------|-------------------|-------|-------|-------|-------|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Mixed impulse noise | 30.12 | 28.89 | 27.31 | 25.88 | 24.12 |
| Salt-and-pepper noise | 30.68 | 28.73 | 26.76 | 24.66 | 21.77 |
| Uniform impulse noise | 30.14 | 28.29 | 26.90 | 25.31 | 23.37 |

Several tests has been done to expose the proposed filter's characteristics. The proposed filter is able to remove mixed impulse noise in low to medium noise condition successfully. The detection scheme in the filter is able to differentiate noisy pixels out of the noise-free ones. Therefore, it saves more details than median filters do. Its performances in removing salt-and-pepper and uniform impulse noise are as good as its performance in removing mixed impulse noise.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Accomplishing the Goal of This Research

In this thesis, a fuzzy image filtering based on ROAD and MAD to remove mixed impulse noise has been presented. The proposed method consists of three components, including fuzzy noise detection system, fuzzy switching scheme filtering, and fuzzy parameters optimization using GA to perform efficient and effective noise removal. The noise are detected by the fuzzy noise detection system, and then filtered using fuzzy switching scheme filtering to minimize degrading effect of median filtering on noise-free pixels. The GA training aims to find the best parameters for the fuzzy sets in fuzzy noise detection.

From the experimental results, we observe that the proposed filter can preserve more detail than median filter in 5×5 , its estimator. When it was applied to noise free Lena image, it gained PSNR value 19.29% higher than median filter did. Based on the experiments in section 5.3, PSNR and MAE value of the proposed method achieve the best results compared to the other approaches, including median filters, average filters, EIE, WFM, GFIF, and FRINR, when removing impulse noise in 0.1 to 0.5 noise probability. Subjective evaluation on visual quality of the filtering results also shows its superior performance on removing mixed impulse noise in low to medium noise probabilities of mixed impulse noise. The performance of the filter to remove other types of impulse noise, such as salt-and-pepper and uniform impulse noise, is as good as its performance in removing mixed impulse noise.

6.2 Research Contributions

The contributions of this research lie on the design of the new filter. They are:

1. The integration of ROAD and MAD image statistical properties into a simple fuzzy inference system to detect mixed impulse noise in corrupted images. It includes the fuzzy sets and fuzzy rule design and development as parts of the fuzzy inference system. It results in less complicated design of fuzzy filter that is able to outperform several existing methods.
2. The collaboration of fuzzy set concept with switching scheme filtering results in a soft-switching scheme filtering. Instead of treating each pixel as pure noisy or noise-free, it treats each pixel uniquely based on the value of noise detection that ranges in $[0,1]$.
3. Genetic learning scheme is specially designed to complete the fuzzy set design. It provides the system with the optimal parameter of fuzzy sets. The genes are encoded in integer codification instead of the common binary or float codification. This encoding scheme requires an operator set that conforms to the encoding scheme. These operators are derived from those for binary encoding. Mean absolute error (MAE) is adopted as the fitness function to evaluate the fitness values over generations. The nature of the variables encoded as the genes also demands a set of restrictions and repair algorithms to maintain the fuzzy membership functions.
4. This filter is purposely designed to remove mixed impulse noise. There are not many filtering methods built based on this noise type yet. As a new model of impulse noise that is more realistic, designing filter to remove this noise type could also lead to good performances in removing other impulse noise types, such as salt-and-pepper and uniform impulse noise.

6.3 Future Works

The median filter as the filtering estimator is a classical yet quite powerful filtering method for impulse noise. Nowadays, many modified median filters have more advanced features than their ancestor has. Finding a better estimator other than median filter surely could improve the filtering results.

Fuzzy logic is now extended to a higher order, i.e. fuzzy type-2, which can handle uncertainties in fuzzy membership functions. Combined with Human Evolutionary Model as an optimization method, a powerful hybrid system can be built [40]. It is of high potential to the improvement of the genetic-fuzzy filter system.

Since most of the images that people deal with today mostly come in color, it is a good idea to modify the method to remove impulse noise in color images. The filtering processes in color images take the same concept as filtering in grayscale images, but in color images, there are multiple layers as well as different color modes. Some modifications on the method are needed to remove impulse noise in color images.

REFERENCE

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. New Jersey: Prentice Hall, 2002.
- [2] M. Paulinas and A. Usinkas, "A survey of genetic algorithms applications for image enhancement and segmentation," *Information Technology and Control, Kaunas, Technologija*, vol. 36, pp. 278-284, 2007.
- [3] M. Nachtegaele, D. Van der Weken, E. Kerre, and W. Philips, *Soft Computing in Image Processing: Recent Advances*: Springer, 2007.
- [4] J. C. Russ, *The image processing handbook*, 5th ed. Boca Raton: CRC Press, 2006.
- [5] M. Egmont-Petersen, D. d. Ridder, and H. Handels, "Image processing with neural networks - a review," *Journal of Pattern Recognition*, vol. 35, pp. 2279-2301, 2002.
- [6] F. Russo, "An image enhancement technique combining sharpening and noise reduction," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, pp. 824-828, 2002.
- [7] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of Impulse noise from highly corrupted images," *IEEE Transactions on Image Processing*, vol. 5, pp. 1012-1025, 1996.
- [8] N. Petrović and V. Crnojević, "Universal impulse noise filter based on genetic programming," *IEEE Transactions on Image Processing*, vol. 17, pp. 1109-1120, 2008.
- [9] U.-K. Cho, J.-H. Hong, and S.-B. Cho, "Evolutionary image enhancement for impulsive noise reduction," in *Intelligent Computing*, 2006, pp. 678-683.
- [10] S.-J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Transactions on Circuits and Systems*, vol. 38, pp. 984-993, 1991.
- [11] C.-S. Lee, Y.-H. Kuo, and P.-T. Yu, "Weighted fuzzy mean filters for image processing," *Fuzzy Sets and Systems*, vol. 89, pp. 157-180, 1997.
- [12] C.-S. Lee and Y.-H. Kuo, "The important properties and applications of the adaptive weighted fuzzy mean filter," *International Journal of Intelligent Systems*, vol. 14, pp. 253-274, 1999.
- [13] C.-S. Lee, S.-M. Guo, and C.-Y. Hsu, "Genetic-based fuzzy image filter and its application to image processing," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, pp. 694-711, 2005.
- [14] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: a tutorial," *IEEE Transactions on Circuits and Systems*, vol. 43, 1996.
- [15] D. R. K. Brownrigg, "The weighted median filter," *Image Processing and Computer Vision*, vol. 27, pp. 807-818, August 1984.

- [16] C.-S. Lee and Y.-H. Kuo, "Adaptive weighted fuzzy mean filter," in *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, 1996, pp. 2110-2116 vol.3.
- [17] S.-M. Guo, C.-S. Lee, and C.-Y. Hsu, "An intelligent image agent based on soft-computing techniques for color image processing," *Expert Systems with Applications*, vol. 28, pp. 483-494, 2005.
- [18] V. Crnojević, "Impulse noise filter with adaptive MAD-based threshold," in *IEEE International Conference on Image Processing, ICIP 2005*, 2005, pp. 337-340.
- [19] Y. Dong, R. H. Chan, and S. Xu, "A detection statistic for random-valued impulse noise," *IEEE Transactions on Image Processing*, vol. 16, pp. 1112-1120, 2007.
- [20] S. Schulte, V. De Witte, M. Nachtegael, D. Van der Weken, and E. E. Kerre, "Fuzzy random impulse noise reduction method," *Fuzzy Sets and Systems*, vol. 153, pp. 270-283, 2006.
- [21] S. K. Mitra and G. L. Sicuranza, "Nonlinear Image Processing," London: Academic Press, 2001, p. 455.
- [22] J. Astola, *Fundamentals of Nonlinear Digital Filtering*: CRC Press, 1997.
- [23] V. Crnojević, V. Šenk, and Ž. Trpovski, "Advanced impulse detection based on pixel-wise MAD," *IEEE Signal Processing Letters*, vol. 11, pp. 589-592, 2004.
- [24] H.-L. Eng and K.-K. Ma, "Noise adaptive soft-switching median filter," *IEEE Transactions on Image Processing*, vol. 10, pp. 242-251, 2001.
- [25] S. Satyanarayana and S. Dalal, "Video Color Enhancement Using Neural Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 295-307, 1996.
- [26] L. O. Chua and L. Yang, "Cellular Neural Networks: Theory," *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 1257-1272, 1988.
- [27] L. O. Chua and L. Yang, "Cellular Neural Networks: Applications," *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 1273-1290, 1988.
- [28] M. Nachtegael, D. Van der Weken, D. Van De Ville, E. E. Kerre, W. Philips, and I. Lemahieu, "An overview of fuzzy filters for noise reduction," in *IEEE International Fuzzy Systems Conference*, 2001, pp. 7-10.
- [29] S. Schulte, V. De Witte, M. Nachtegael, D. Van der Weken, and E. E. Kerre, "Fuzzy random impulse noise reduction method," *Fuzzy Sets and Systems*, vol. 153, pp. 270-283, 2007.
- [30] 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 Transactions on Image Processing*, vol. 15, pp. 3568-3579, November 2006.
- [31] S. Schulte, V. De Witte, and E. E. Kerre, "A fuzzy noise reduction method for color images," *IEEE Transactions on Image Processing*, vol. 16, pp. 1425-1436, May 2007.
- [32] S. Schulte, V. De Witte, M. Nachtegael, D. Van der Weken, and E. E. Kerre, "A New Fuzzy Multi-channel Filter for the Reduction of Impulse Noise," in *Pattern Recognition and Image Analysis*, 2005, pp. 368-375.
- [33] C.-C. Kang and W.-J. Wang, "A novel directional median filter based on fuzzy reasoning," in *Consumer Electronics, 2008. ISCE 2008. IEEE International Symposium on*, 2008, pp. 1-4.

- [34] C.-C. Kang and W.-J. Wang, "Fuzzy reasoning-based directional median filter design," *Signal Processing*, vol. 89, pp. 344-351, 2009.
- [35] A. Hussain, M. Jaffar, and A. Mirza, "A hybrid image restoration approach: fuzzy logic and directional weighted median based uniform impulse noise removal," *Knowledge and Information Systems*, 2009.
- [36] F. Russo, "Evolutionary neural fuzzy systems for noise cancellation in image data," *IEEE Transactions on Instrumentation and Measurement*, vol. 48, October 1999.
- [37] M. E. Yüksel, A. Baştürk, and E. Beşdok, "Detail-preserving restoration of impulse noise corrupted images by a switching median filter guided by a simple neuro-fuzzy network," *EURASIP Journal on Applied Signal Processing*, vol. 16, pp. 2451-2461, 2004.
- [38] A. P. Engelbrecht, *Computational intelligence: an introduction*. West Sussex: John Wiley & Sons, Ltd, 2002.
- [39] M. Negnevitsky, *Artificial Intelligence: a guide to intelligent system*, 2nd ed.: Addison-Wesley, 2005.
- [40] O. Castillo and P. Melin, *Type-2 fuzzy logic: theory and applications*. Berlin Heidelberg: Springer, 2008.
- [41] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*, 1st ed.: Springer, 2003.
- [42] D. K. Pratihari, *Soft Computing*. Oxford: Alpha Science International Ltd., 2008.
- [43] F. O. Karray and C. De Silva, *Soft computing and intelligent systems design*. Harlow, England: Addison Wesley, 2004.
- [44] G. Upton and I. Cook, *A dictionary of statistics*: Oxford University Press, 2002.
- [45] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *IEEE Transactions on Image Processing*, vol. 14, pp. 1747-1754, 2005.
- [46] H. Gangadhar and E. Srinivasan, "Performance Comparison of ROAD Statistic Based Nonlinear Filters for Image Denoising," in *IEEE Region 10 Colloquium and the third ICIS Kharagpur, India: IEEE*, 2008.
- [47] H. Yu, L. Zhao, and H. Wang, "An efficient procedure for removing random-valued impulse noise in images," *IEEE Signal Processing Letters*, vol. 15, pp. 922-925, 2008.
- [48] A. C. C. Carlos, "A survey of constraint handling techniques used with evolutionary algorithms," Laboratorio Nacional de Informatica Avanzada, Xalapa, Veracruz, Mexico 1999.
- [49] "Standard Test Images," Homepage of Fuzzy Image Processing, University of Waterloo. Available: <http://pami.uwaterloo.ca/tizhoosh/images.htm>.

APPENDIX A

LIST OF PUBLICATIONS

- [1] N. Z. Janah, B. Baharudin, "Genetic fuzzy filter based on MAD and ROAD to remove mixed impulse noise", *Majlesi Journal of Electrical Engineering*, 2010 <submitted>
- [2] N. Z. Janah, B. Baharudin, "Mixed impulse fuzzy filter based on MAD, ROAD, and genetic algorithms," in *International Conference on Soft Computing and Pattern Recognition (SoCPaR 2009)*, 2009.
- [3] N. Z. Janah, B. Baharudin, "A review on neural network, genetic algorithm and fuzzy filters for noise reduction," in *National Postgraduate Conference on Engineering, Science and Technology (NPC 2009)*, 2009.
- [4] N. Z. Janah, B. Baharudin, "A Review on Soft Computing in Image Enhancement Techniques: Neural Network, Genetic Algorithms, and Fuzzy Filters," in *International Graduate Conference on Engineering and Science 2008*, 2008.