Estimation, Suppression and Optimization of Impulses in Digital Imagery

Thesis submitted for the Degree of Doctor of Philosophy (Engineering) in the Faculty of Engineering, Technology and Management, University of Kalyani

By

Somnath Mukhopadhyay

Under the Supervision of

Prof. Jyotsna Kumar Mandal

Department of Computer Science and Engineering
University of Kalyani
Kalyani, West Bengal, India

October, 2014

To My Beloved Sana

Copyright (c) 2014, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this thesis work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

University of Kalyani

FACULTY OF ENGINEERING, TECHNOLOGY & MANAGEMENT

Prof.(Dr.) J. K. Mandal
PROFESSOR
DEPT. OF CSE, FACULTY OF ENGINEERING, TECHNOLOGY & MANAGEMENT
UNIVERSITY OF KALYANI

Kalyani-741235, Nadia
West Bengal, India
Phone: (033) 25809617 (O)
Mobile: +91- 9434352214
e-mail: jkm.cse@gmail.com

Certificate

This is to certify that the thesis entitled "**ESTIMATION, SUPPRESSION AND OPTIMIZATION OF IMPULSES IN DIGITAL IMAGERY**" submitted by Mr. Somnath Mukhopadhyay, who got his name registered on 30.05.2012 (Ref. No: Ph.D/Regn. /N.Rgl./Eg-35/Com.Sc./SM/2012 dated July 25, 2012) for the award of Ph.D. (Engineering) degree of the University of Kalyani is absolutely based upon his own work under my supervision and that neither his thesis nor any part of the thesis has been submitted for any degree or diploma or any other academic award anywhere before. I recommend that Mr. Somnath Mukhopadhyay has fulfilled all the requirements according to rules of this University regarding the work embodied in this thesis.

_____

(Dr. Jyotsna Kumar Mandal)
Professor,
Department of Computer Science and Engineering,
University of Kalyani,
Kalyani, India

Date:
Place: Kalyani

Acknowledgment

A research work is not an obligation done by one person toward a predetermined and specific goal. Rather, it is a realization conducted from many elements, some administered directly and others indirectly, to the planning and implementation of a long-term effort. Such a project requires hard work, perseverance and dedication, and above all a lot of supports, as only teamwork can make good better and better best. This work has been carried out as a university research scholar in without fellowship basis in the department of computer science of the University of Kalyani, India.

First, I would like to express my sincere gratitude to Dr. J. K. Mandal of Computer Science and Engineering Department of University of Kalyani for his careful supervision of my doctoral thesis and for all his support from the very beginning right after the thesis idea was conceived. He was always available with new ideas, advises and suggestions during the difficult phases of the thesis work. I would also thankful and express my indebtedness to all the reporters, examiners and jury members of the thesis for accepting to review this report and for their careful evaluation of the thesis presentation.

I would like to express my thanks and gratitude to Dr. Tandra Pal of Department of Computer Science and Engineering of National Institute of Technology, Durgapur for guiding and advising me very sincerely by giving different ideas on the thesis work which has led to the betterment of the result.

This auspicious occasion gives me a wonderful opportunity and great pleasure to express my thanks to Dr. Anirban Mukhopadhyay of University of Kalyani and Prof. Sripati Mukhopadhyay of University of Burdwan for motivating me to work in the field of research. Also, I render my respect to Mrs. Shyamali Mandal for giving her support and help all the way.

I am thankful and have pleasure to express my deep friendship to all the scholars of the department of Computer Science and Engineering that supported my work and entertained me with love; that helped me a lot to progress rapidly in my work on daily basis.

I would like to express my thanks to Prof. Paramartha Dutta of Viswa Bharti University, Computer Science and Engineering department for supporting me by giving different ideas on the thesis work which has led to the betterment of the result. I am very much grateful to him for his support.

I would also like to thank my closed friends Rudra Narayan Mazumdar and Tamal Dev for supporting and motivating me a lot throughout the research work since 2009.

Last but not the least, before going to the implementation detail, I would like to express all my sincere love to my brother like, Dr. Kartick Chandra Mandal, Assistant Professor of Computer Science and Engineering, Jadavpur University, India, for his valuable advises, guidance and unending support right from the beginning. I have no word to express my

gratitude for his constant motivation and inspiration.

Also a lot of thanks and love to my mother, my brother, my wife and all my family members who always have supported me in every possible way. I will always be thankful for their love and confidence on me, which has constantly motivated me to work healthy and happily. I sincerely appreciate all their love, support and confidence that made this work possible.

So, right at the end I would like to express my apologies to those whose names I have missed out to mention.

Somnath Mukhopadhyay

Abbreviations

ANDWP All Neighbor Directional Weighted Pixels

ACWM Adaptive Center Weighted Median Filter

ATPRO Adaptive Two Pass Rank Order

BTS Binary Tournament Selection

BDND Boundary Discriminative Noise Detection

CWM Center Weighted Median Filter

DIP Digital Image Processing

DSP Digital Signal Processing

DPSO Different length Particle Swarm Optimization

DWM Directional Weighted Median

DWMD Directional Weighted Minimum Deviation

DFT Discrete Fourier Transform

DWT Discrete Wavelet Transform

EA Evolutionary Algorithm

EC Evolutionary Computation

EDBA Efficient Decision Based Algorithm

EIE Evolutionary Image Enhancement

EPR Edge Preserving Restoration

FF Fuzzy Filter

FSMF Fuzzy Switching Median Filter

FPSO Fixed length Particle Swarm Optimization

FBDA Fuzzy Based Decision Algorithm

GA Genetic Algorithm

GASIR Genetic Algorithm based Sub-Image Restoration

GANDFS Genetic Algorithm based Noise Detection and Filtering Scheme

IDN Image De Noising

IDBA Improved efficient Decision-Based Algorithm

ID Impulse Detector

IF Image Fidelity

IDFT Inverse Discrete Fourier Transform

ITR Iteration, Threshold and decreasing Rate of threshold in each iteration

MWB Modified Weighted Based

MAD Median of the Absolute Deviations from the median

SIR Sub-Image Restoration

SM Standard Median

MDWMF Modified Directional Weighted Median Filter

MF Miss and False

MSM Multi State Median

MRDH Median Rational Hybrid Filter

MDWMF Modified Directional Weighted Median Based Filter

MWB Modified Weighted Based Filter

NFF Neuro Fuzzy Filter

PDF Probability Density Function

PSP Percentage of Spoiled Pixels

PSO Particle Swarm Optimization

PSOSIR Particle Swarm Optimization based Sub-Image Restoration

PWMAD Pixel Wise modifications of the MAD

PSM Progressive Switching median

PMF Partition based Median Filter

PVF Peak and Valley Filter

PSNR Peak Signal to Noise Ratio

RVIN Random Valued impulse Noise

RVINGA Random Valued Impulse Noise using GA

RVINPSO Random Valued Impulse Noise using PSO

SPN Salt and Pepper Noise

SD-ROM Signal Dependent Rank Ordered Mean

SI Swarm Intelligence

SMF Switching Median Filter

SNR Signal to Noise Ratio

SODM Second Ordered Difference based Median

SODM Second Order Difference based Median filter

TSM Tri State Median

TBF Threshold Boolean filter

UC Uniform Crossover

VMM Variable Mask Median

WMF Weighted Median Filter

WM Weighted Median

Notations

Symbol	Significance
Mathematical operators :	
$\neq$	Inequality
$\geq$	Greater than or equal to
$\leq$	Less than or equal to
Set theory operators :	
$\in$	Membership
$\notin$	Absence of membership
$\subset$	Subset inclusion
$\subseteq$	Subset inclusion or equality
$\not\subseteq$	Not a proper subset
$\bigcup$	Generalized union
$\bigcap$	Generalized intersection
$\exists$	Existence
$\forall$	Universality
Set and logic symbols :	
$\emptyset$	Null set
σ	Standard deviation
ρ	Set of pixels
λ	Set of pixels
$\diamond$	Repetition operator
Σ	Sum operator
μ	Membership function
$\mathbb{R}$	Set of real
$\dashv$	Implication
ψ	Learning factor
γ	Connection function
Γ	Closure operator

List of Publications

Journals

1. Mukhopadhyay, S., Mandal, J.K.: A Fuzzy Switching Median Filter of Impulses in Digital Imagery (FSMF). Circuits, Systems and Signal Processing, 33(2), pp.1-24 (2014), Springer Science + Business Media New York. DOI 10.1007/s00034-014-9739-z

2. Mukhopadhyay, S., Mandal, J.K.: Denoising of digital images through PSO based pixel classification. Central European Journal of Computer Science 3(4), pp.158-172 (2013), Springer Vienna. DOI 10.2478/s13537-013-0111-3

3. Mandal J. K., Mukhopadhyay S.: A Novel Technique of Filtering High Random Valued Impulse Noise and Optimization through Genetic Algorithm (HRVINGA). AMSE Journal of Signal Processing and Pattern Recognition, France, Volume 55, Issue 2, No. 1-2, 2012, pp.37

4. Mandal J. K., Mukhopadhyay S.: Image Filtering using All Neighbor Directional Weighted Pixels: Optimization using Particle Swarm Optimization. In Signal and Image Processing: An International Journal (SIPIJ) Vol.2, No.4, pp. 187-200, December 2011, Academy and Industry Research Collaboration Center (AIRCC)

5. Mukhopadhyay, S., Mandal, J.K.: Image Denoising based on Sub-image Restoration through Threshold Optimization. **accepted** in International Journal of Computational Intelligence Studies (IJCIStudies), Inderscience Publishers, Geneva, Switzerland

Book Chapters

6. Mandal J. K., Mukhopadhyay S.: Adaptive Median Filtering based on Unsupervised Classification of Pixels. (Handbook of Research on Computational Intelligence for Engineering, Science and Business, IGI Global, 701 E. Chocolate Ave., Hershey, PA 17033, USA)

International Conferences

7. Mandal J. K., Mukhopadhyay S.: A Novel Directional Weighted Minimum Deviation (DWMD) based Filter for Removal of Random-Valued Impulses in Digital Images. International Conference on Computing and Systems (ICCS 2010), pp. 214-220, Burdwan University, India, November 2010

8. Mandal J. K., Mukhopadhyay S.: A Novel Technique for Removal of Random Valued Impulse Noise using All Neighbor Directional Weighted Pixels (ANDWP). International Conference on Advances in Parallel, Distributed Computing: Communications in Computer and Information Science (CCIS), Springer, vol. 203, pp. 102-111, Tirunelveli, Tamil Nadu, India, September 2011

9. Mandal J. K., Mukhopadhyay S.: GA based Denoising of Impulses (GADI). 10th International Conference on Computer Information Systems and Industrial Management Applications (CISIM 2011), Springer CCIS, vol. 245, pp. 212-220, Calcutta University, India, December 2011

10. Mandal J. K., Mukhopadhyay S.: A Novel Technique for Removal of Random Valued Impulses using Variable Mask Median Filter (VMM). International Symposium on Electronic System Design (ISED 2011), IEEE Computer Society, pp. 302-306, December 2011, Kochi, India

11. Mandal J. K., Mukhopadhyay S.: Edge Preserving Restoration of Random Valued Impulse Noises in Digital Images (EPRRVIN). International Conference on Recent Trends in Information Systems (ReTIS-11), IEEE Calcutta Section, pp. 309-314, Jadavpur University, India

12. Mandal J. K., Mukhopadhyay S.: PSO based Edge Keeping Suppression of Impulses in Digital Imagery. Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012), Visakhapatnam, India, January 2012: Springer AISC, vol.132, pp. 395-403

13. Mukhopadhyay, S., Mandal J. K.: Wavelet based Denoising of Medical Images using Sub-band Adaptive Thresholding through Genetic Algorithm. Procedia Technology, Elsevier Ltd., 10(0), pp.680-689 (2013). First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA), University of Kalyani, India, 2013

14. Mukhopadhyay, S., Mandal, P., Pal, T., Mandal J. K.: Image Clustering based on Different length Particle Swarm Optimization (DPSO), **accepted** In International Conference on Frontiers of Intelligent Computing: Theory and applications (FICTA) 2014, Springer AISC, Bhubaneswar, India

Publication Indexing Database

The list of publications are indexed/abstracted in the following Databases which are mentioned in the following table serially.

Publication Serial No.	**Database**
1	SCI, SCOPUS, INSPEC, ACM Digital Library, Gale, SCImago, Google Scholar etc.,
2	DBLP, Zentralblatt Math, OCLC, Summon by ProQuest, Google Scholar etc.,
3	SCOPUS, IEEE-INSPECT, Elsevier BV, Institute for Scientific Information etc.,
4	DBLP, ProQuest, EBSCO, Scribd, Google Scholar etc.,
5	Academic OneFile (Gale), ACM Digital Library, Expanded Academic ASAP (Gale), Inspec.,
6	Thomson Reuters, SCOPUS, DBLP, Compendex, PsycINFO, INSPEC, Cabell etc.,
7	DBLP, Google Scholar etc.,
8	SCOPUS, SCImago etc.,
9	SCOPUS, DBLP etc.,
10	SCOPUS, SCImago etc.,
11	SCOPUS, SCImago, DBLP etc.,
12	SCOPUS, Google Scholar etc.,
13	SCOPUS, INSPEC, Google Scholar, ACM Digital Library, Gale, SCImago etc.,
14	ISI Proceedings, DBLP, Ulrich's, EI-Compendex, SCOPUS, Zentralblatt Math, MetaPress etc.,

Table 1: Indexing Database of Publications

List of Presentations

International Conferences

1. Mandal J. K., Mukhopadhyay S.: A Novel Directional Weighted Minimum Deviation (DWMD) based Filter for Removal of Random-Valued Impulses in Digital Images. International Conference on Computing and Systems (ICCS 2010), pp. 214-220, Burdwan University, India, November 2010

2. Mandal J. K., Mukhopadhyay S.: GA based Denoising of Impulses (GADI). 10th International Conference on Computer Information Systems and Industrial Management Applications (CISIM 2011), Springer CCIS, vol. 245, pp. 212-220, Calcutta University, India, December 2011

3. Mandal J. K., Mukhopadhyay S.: A Novel Technique for Removal of Random Valued Impulses using Variable Mask Median Filter (VMM). International Symposium on Electronic System Design (ISED 2011), IEEE Computer Society, pp. 302-306, December 2011, Kochi, India

4. Mandal J. K., Mukhopadhyay S.: Edge Preserving Restoration of Random Valued Impulse Noises in Digital Images (EPRRVIN). International Conference on Recent Trends in Information Systems (ReTIS-11), IEEE Calcutta Section, pp. 309-314, Jadavpur University, India

5. Mandal J. K., Mukhopadhyay S.: PSO based Edge Keeping Suppression of Impulses in Digital Imagery. Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012), Visakhapatnam, India, January 2012: Springer AISC, vol.132, pp. 395-403

6. Mukhopadhyay, S., Mandal J. K.: Wavelet based Denoising of Medical Images using Sub-band Adaptive Thresholding through Genetic Algorithm. Procedia Technology, Elsevier Ltd., 10(0), pp.680-689 (2013). First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA), University of Kalyani, India, 2013

Abstract

Digital image processing plays a vital role in various applications like medical, satellite, underwater, robot vision, etc. Image denoising is the primary preprocessing to almost all image analysts since impulse noise is an unwanted signal that mixed with the original signal in different situations, such as during image acquisition, storage and transmission. There are various types of impulses based on their characteristics and way of combining with the original signal. One of the most important impulse type is Salt and Pepper noise (SPN) which degrades the digital image by replacing some pixels with maximum or minimum gray value of the image. Random Valued noise (RVN) replaces some pixels with random value of the intensity range of the digital image. Generally medical images have poor contrast and several types of noises. The suppression of noise in medical images corrupted by mixed Gaussian and Poisson noise is a major issue in diverse image processing and computer vision problems.

In the thesis, novel approaches for digital and medical image denoising have been proposed. The problem of image denoising is categorized into two major parts, first one is the denoising of digital images in spatial domain and the other one is denoising the medical images in frequency domain. All the algorithms proposed here are two-step method for noise detection and filtering operations. In the spatial domain, the impulse noises are suppressed, estimated and optimized by the proposed algorithms. In frequency domain based filtering the noisy medical images are restored. A set of five algorithms are proposed for suppressing the noises in the digital images which are based on simple statistical and mathematical operations. Those techniques are directional weighted minimum deviation (DWMD) filter, all neighbor directional weighted pixels (ANDWP) based filter, variable mask median filter (VMM), edge preserving restoration (EPR) filter and sub-image restoration (SIR) based filter. These five algorithms use three user parameters such as the number of iterations (I), threshold value (T) and decreasing rate of threshold value in each iteration (R). Those parameters can be tuned to get the optimal solutions by the proposed algorithms. Two recent stochastic and randomized search algorithms such as Genetic algorithm (GA) and Particle Swarm Optimization (PSO) based optimizations of the noise filtering operators are performed. Here GA and PSO based optimization techniques are proposed to search the optimal values of the three parameters for which global optimal solutions are obtained. An adaptive median filter is proposed for denoising of digital images. In this technique, the noise density is estimated towards suggesting an empirical window for noise detection and filtering operations. Fixed and different length particle swarm optimization algorithm based image clustering techniques are devised for the image denoising purpose. A fuzzy switching median filter is also proposed for denoising the impulses in the digital images. In the frequency domain, two dimensional discrete wavelet transformation based image denoising has been done on the medical images.

In the DWMD and ANDWP filters, the random valued impulse noise is detected and filtered on each 5 × 5 window where the center pixel of the window is under consideration. The ANDWP filter considers all pixels in the 5 × 5 window for noise detection and filtering. The center pixel is classified as noisy or noise-free based on simple arithmetic and statistical operations. If the center pixel of the 5 × 5 window is detected as noisy then it is restored by a standard deviation filter. In detection step, simple arithmetic operations are performed on the neighboring pixels to classify the center pixel of the test window. During restoration, the proposed operator filters only the noisy pixels based on minimum variance of four directional pixels.

Variable mask median filtering (VMM) has been proposed for denoising the digital images which are corrupted by random valued impulse noises. The detection of noisy pixels is done using variable weights of all neighbor directional pixels using 5 × 5 mask. The operator performs simple arithmetic absolute differences along with some other arithmetic operations on the pixels aligned in the four directions with the center pixel to classify test pixel. For noise filtering, an improved median based filtering technique has been proposed which uses variable mask consisting of nine and twenty five pixels respectively. Certain pixels in the 3 × 3 window are selected prior to compute median operation. Three user supplied parameters are varied in a wide range to obtain the optimal results.

The edge preserving restoration (EPR) filter is proposed to remove random valued impulse noise from the digital images, a novel two step method has been proposed. The first step is to classify whether the center pixel in the 5 × 5 window is noisy or not, which is done using all neighbor directional weighted pixels in a 5 × 5 mask. The proposed algorithm performs arithmetic absolute differences on the pixels aligned in the four directions with the center pixel. On detection of noisy pixels an advance median filter has been proposed for restoration of noisy pixel where a variable window consisting of 3 × 3 and 5 × 5 respectively have been used. To optimize the results, a three dimensional search space has been used on user supplied parameters in a wide range.

In sub-image restoration (SIR) based filter, the detection and filtering of impulses in digital images have been done through median filtering technique. The noisy image is logically partitioned into 5 × 5 sub images. Two restored versions are generated through proposed technique using 3 × 3 and 5 × 5 masks separately of which better one in terms of PSNR (dB) is chosen. The proposed detection method is based on all neighbor directional pixels. The absolute differences between each of four directional pixels with the center pixel are calculated for detection. The filtering is based on weighted median approach. Among the four directions, positive weights are assigned to the direction containing minimum standard deviation considering the fact that they have more intensity value similarities. Thus certain pixels are selected for median filtering. The median value replaces the center pixel to restore it.

Three user parameters such as the number of iterations (I), threshold value (T) and decreasing rate of threshold value in each iteration (R) of the algorithms are searched in a three dimensional search space. The first parameter of the algorithm is number of iterations (I) which is varied in a wide range to get the optimal restoration results. The most important parameter which is used in the noise detection rule is the Threshold (T) value. Depending upon the value of the threshold any pixel in the image is classified as noisy or noise free. Lastly, in each iteration, the threshold value is decreased by a multiplying factor as generation wise the noise density is decreased by an amount. These three most relevant parameters are searched in a wide range to get the optimal restoration results. Two optimization algorithms which are stochastic and randomized in nature are used for optimization of the noise removal operators. Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) techniques based optimization algorithms are proposed here.

The noise density of any digital image should be estimated first before the denoising algorithm works. The K-Means based noise density estimation towards image denoising is done in this technique. This method deals with a novel approach which aims at detection and filtering of impulses in digital images through clustering of pixels. This approach coagulates directional weighted median filtering with clustering based adaptive window selection toward detection and filtering of impulses in digital images. In this technique, noise density of any digital image is estimated first then an empirical window size is suggested for noise detection and filtering operations. Adaptive median filtering approach has been proposed to obtain best possible restoration results.

The image clustering towards image denoising technique has been proposed to suppress the digital images corrupted by two types of impulse noises. This method proposes a denoising method where the detection and filtering is based on clustering of image pixels. The noisy image is grouped into subsets of pixels with respect to their intensity values and spatial distances. A novel Euclidean distance function has been proposed in this clustering algorithm. Using a novel fitness function the image pixels are classified using the fixed length and different length Particle Swarm Optimization (PSO) techniques. The distance function measured similarity/dissimilarity among pixels using not only the intensity values, but also the positions of the pixels.

The fuzzy switching median filter aims at detection and filtering of impulse noises from digital images. Two types of noise models are used to obtain the noisy images. In this two-step process, the noise-free pixels are remained unchanged. The proposed detection algorithm uses 5×5 window, based on all neighboring pixels on the center of the window of a noisy pixel. Two weighted median filters are devised, and a particular one is applied selectively to the noisy pixel based on the characteristics of the neighboring pixels within the window. Instead of a single threshold, two threshold values are used in the proposed fuzzy membership function to partition the noise level, and accordingly, a filtering method is applied to restore

the corrupted pixel.

In frequency domain based image denoising, a sub-band adaptive wavelet thresholding is done to denoise the medical images. This algorithm proposed a denoising method of medical images through thresholding and optimization using Genetic Algorithm (GA). The noisy image is partitioned into fixed sized blocks and then transformed it into wavelet domain. Some important parameters of the two dimensional discrete wavelet transformation such as the decomposition level and the threshold value are searched and optimized in a wide range in the proposed technique. The Bayesian shrinkage method has been selected for thresholding based of its sub band dependency property.

Experimental results and discussions show the performances of the different algorithms of image denoising and compare them to others state-of-the-art algorithms for noise estimation, suppression and optimization of the proposed operators. The efficiency of the algorithms have been tested using standard images on applying several types noise models such as Salt-and-Pepper noise, Random Valued noise, Poisson and Gaussian noises. The noisy images are generated by applying the noise models with 10 % to 100% density. On analysis, it is seen that for all types of noise models, the proposed algorithms perform excellent in terms of PSNR (dB), Image Enhancement Factor (IEF), Structure Similarity Index Measure (SSIM) and visual effects. It has been established through experimental analysis that the proposed operators give excellent restoration results compared to conventional operators, such as standard median filter (SMF), adaptive median filter (AMF) etc., and some state-of-the-art methods such as directional weighted median filter (DWMF), efficient decision-based algorithm (EDBA), improved efficient decision-based algorithm (IDBA), boundary discriminative noise detection (BDND), boundary discriminative noise detection by elimination (BDNDE), fuzzy-based decision algorithm (FBDA), NASRI., etc. Time complexity of each of the proposed algorithm has also been studied and reported in the results section of the thesis work.

Wavelet based denoising algorithm has been validated through an ultrasound image corrupted by a variety of noise densities through Gaussian and Poisson noises in terms of peak signal to noise ratio and visual effects. Simulation results show that the proposed method outperforms the existing denoising methods.

Contents

I Introduction **1**

1 Introduction **3**
1.1 Image Denoising 4
1.2 Types of Noises 10

2 State of the Art **17**
2.1 Denoising in Spatial Domain 19
2.2 Denoising in Frequency Domain 37
2.3 Background and Motivation 55
2.4 Contributions 57
2.5 Layout of the Thesis 59

II Proposed Denoising Techniques **61**

3 Noise Suppression **63**
3.1 Variance based Filter 64
3.1.1 Directional Weighted Minimum Deviation Filter 65
3.1.2 All Neighbor Directional Weighted Pixels Filter 71
3.2 Weighted Median Filter 75
3.2.1 Variable Mask Median Filter 77
3.2.2 Edge Preserving Restoration Filter 80
3.2.3 Sub-Image Restoration Filter 84
3.3 Discussions 88

4 Filter Optimization **93**
4.1 Based on Genetic Algorithm 94
4.1.1 The Proposed Technique 95
4.2 Based on Particle Swarm Optimization 100
4.2.1 The Proposed Technique 100
4.3 Discussions 104

5 Noise Density Estimation for Adaptive Median Filter 107
5.1 Proposed Technique for Noise Detection . . . 108
5.2 Proposed Technique for Noise Density Estimation . . . 108
5.3 Proposed Technique for Adaptive Median Filter . . . 111
5.4 Illustration and Examples . . . 111
5.5 Discussions . . . 115

6 PSO based Clustering for Noise Detection and Restoration 117
6.1 Based on Fixed length Particles . . . 118
6.1.1 Proposed Technique for Fixed length PSO based Clustering . . . 119
6.2 Based on Different length Particles . . . 123
6.2.1 Proposed Technique for Different length PSO based Clustering . . . 124
6.3 Proposed Technique for Noise Detection and Restoration . . . 126
6.4 Discussions . . . 129

7 Fuzzy Switching Median Filter 131
7.1 Proposed Technique for Noise Detection . . . 133
7.2 Proposed Technique for Noise Restoration . . . 134
7.3 Proposed Technique for Fuzzy Switching . . . 137
7.4 Discussions . . . 138

8 Denoising in Frequency Domain 141
8.1 Proposed Technique for Denoising . . . 142
8.2 Discussions . . . 144

III Experimental Results and Comparisons 145

9 Experimental Results and Comparisons 147
9.1 Noise Suppression . . . 149
9.1.1 DWMD Filter . . . 149
9.1.2 ANDWP Filter . . . 150
9.1.3 VMM Filter . . . 153
9.1.4 EPR Filter . . . 156
9.1.5 SIR Filter . . . 158
9.1.6 Comparisons . . . 163
9.2 Filter Optimization . . . 171
9.2.1 Based on Genetic Algorithm . . . 171
9.2.2 Based on Particle Swarm Optimization . . . 172

9.2.3 Comparisons . . . 174
9.3 Noise Density Estimation for Adaptive Median Filter . . . 184
9.3.1 Results . . . 184
9.3.2 Comparisons . . . 185
9.4 PSO based Clustering for Noise Detection and Restoration . . . 200
9.4.1 Results . . . 200
9.4.2 Comparisons . . . 201
9.5 Fuzzy Switching Median Filter . . . 215
9.5.1 Results . . . 215
9.5.2 Comparisons . . . 220
9.6 Denoising in Frequency Domain . . . 250
9.6.1 Results . . . 250
9.6.2 Comparisons . . . 250
9.7 Discussions . . . 251

IV Conclusive Discussions 255

10 Conclusion 257
10.1 Limitations and Future Scope . . . 260

V Bibliography 263

Bibliography 265

Index 285

List of Figures

1.1 (a)Subimage representation of wavelet decomposition on level three (b)Wavelet decomposition on *Lena* image . . . 7
1.2 (a) Original Image (b) Hard Thresholding (c) Soft Thresholding . . . 9
1.3 Probability Density Function of Gaussian Distribution . . . 11

2.1 Image denoising techniques . . . 18
2.2 A 3×3 neighbor about a pixel (x,y) in an image. . . . 20
2.3 3 × 3 smoothing filters of an image. . . . 21
2.4 A noisy sine wave and its denoised version (solid line) . . . 38
2.5 Signal f and its noisy version y . . . 39

3.1 Four directions for impulse detection and restoration . . . 66
3.2 Four directions include all neighboring pixels . . . 71
3.3 Four directions of an 3 × 3 window where the center pixel is under consideration 82
3.4 Process diagram for each of the 5 × 5 sub image . . . 85

4.1 Context level diagram of optimization technique . . . 95
4.2 Uniform crossover . . . 98
4.3 Mutation . . . 99
4.4 Change of velocity and position of particles . . . 102

6.1 Four directions of an 3 × 3 window where the center pixel is under consideration 127

7.1 5 × 5 Window . . . 133
7.2 (a)Fuzzy membership function for a noisy pixel [Ima] . . . 138

8.1 Diagram of proposed denoising method . . . 142

9.1 Graph for **DWMD** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises . . . 151
9.2 Graph for **DWMD** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises . . . 152
9.3 Graph for **DWMD** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises . . . 153
9.4 Graph for **ANDWP** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises . . . 155

9.5 Graph for **ANDWP** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises 156
9.6 Graph for **DWMD** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises 157
9.7 Graph for **VMM** in PSNR for *Lena*, *Cameraman*, *Boat*, *Liftingbody* and *Barbara* images for 10% to 100% random valued noises 158
9.8 Graph for **VMM** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises 159
9.9 Graph for **VMM** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises 160
9.10 Graph for **EPR** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises 161
9.11 Graph for **EPR** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises 162
9.12 Graph for **EPR** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises 163
9.13 Graph for **SIR** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises 164
9.14 Graph for **SIR** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises 165
9.15 Graph for **SIR** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises 166
9.16 Graph showing average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filters . 168
9.17 Graph showing average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filters . 169
9.18 Graph showing average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filters . 169
9.19 a-b represent the original and 50% noisy color *Lena* images with RVN. Figures c-h represent the corresponding restored color images by SMF, AMF, DWM, EDBA, IDBA and BDND. Figures i-m show the restored images by proposed **DWMD, ANDWP, VMM, EPR and SIR** 170
9.20 Graph for **GASIR** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% impulses on RVN 173
9.21 Graph for **GASIR** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% impulses on RVN 174
9.22 Graph for **GASIR** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 175

9.23 Graph for **PSOSIR** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 176
9.24 Graph for **PSOSIR** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% impulses on RVN 177
9.25 Graph for **PSOSIR** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% impulses on RVN 178
9.26 Graph for average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filter . . . 180
9.27 Graph of average IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filter 181
9.28 Graph of average SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filter 182
9.29 a-b show original and 50% noisy color *Lena* images with RVN. Figures c-h represent the corresponding restored images by SMF, AMF, DWM, EDBA, IDBA and BDND. Figures i-o show restored images by proposed DWMD, ANDWP, VMM, EPR, SIR, **GASIR and PSOSIR** 183
9.30 Graph for **NDE** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs 186
9.31 Graph for **NDE** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 187
9.32 Graph for **NDE** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs 189
9.33 Graph for **NDE** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 190
9.34 Graph for **NDE** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs 191
9.35 Graph for **NDE** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 192
9.36 Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and NDE 192
9.37 Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and NDE filter 194
9.38 Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and NDE filter 195
9.39 Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and the NDE filter . . . 196
9.40 Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and NDE filter 197

9.41 Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and NDE filter 199
9.42 Graph for **DPSO** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs 203
9.43 Graph for **DPSO** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 204
9.44 Graph for **DPSO** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs 206
9.45 Graph for **DPSO** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 207
9.46 Graph for **DPSO** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs 207
9.47 Graph for **DPSO** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 208
9.48 Graph of *Average PSNR* values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and DPSO filter 208
9.49 Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and DPSO filter 210
9.50 Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and DPSO filter 210
9.51 Graph for average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and DPSO filter 212
9.52 Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and DPSO filter 212
9.53 Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and DPSO filter 214
9.54 Graph for **FSMF** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs 217
9.55 Graph for **FSMF** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 218
9.56 Graph for **FSMF** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs 220
9.57 Graph for **FSMF** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 221
9.58 Graph for **FSMF** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% on SPNs 222
9.59 Graph for **FSMF** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs 223

9.60 Graph representing the regression line of restored pixels by DWMD filter against original pixels for 3 × 3 sub-image of *Lena* image corrupted by SPN with 30% noise density 228

9.61 Graph representing the regression line of restored pixels by DWMD filter against original pixels for 3 × 3 sub-image of *Lena* image corrupted by RVN with 30% noise density 228

9.62 Graph representing the regression line of restored pixels by DPSO filter against original pixels for 3 × 3 sub-image of *Lena* image corrupted by SPN with 30% noise density 229

9.63 Graph representing the regression line of restored pixels by DPSO filter against original pixels for 3 × 3 sub-image of *Lena* image corrupted by RVN with 30% noise density 230

9.64 50% noisy color Lena image corrupted with SPN formed by Y, Cb and Cr components from left to right. First row shows the components of original image. Similarly second and third rows show the components of noisy and restored images by DPSO filter respectively 231

9.65 50% noisy color Lena image corrupted with RVN formed by Y, Cb and Cr components from left to right. First row shows the components of original image. Similarly second and third rows show the components of noisy and restored images by DPSO filter respectively 232

9.66 First and second rows show the components of restored images by DWMD filter with SPN and RVN respectively. Similarly third and fourth rows show the components of restored images by FSMF filter 236

9.67 Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and **FSMF** filter 237

9.68 Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and **FSMF** filter 238

9.69 Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and **FSMF** filter 239

9.70 Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and **FSMF** filter 240

9.71 Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and **FSMF** filter 241

9.72 Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and **FSMF** filter 243

9.73 a and b represent the original and 50% noisy color *Lena* images respectively with SPN. Figures c-j represent the corresponding restored color images by the filters SMF, AMF, DWM, EDBA, IDBA, BDND, FBDA and NASRI. Figures k-m show the restored images by the proposed filters **NDE, DPSO** and **FSMF** 244
9.74 a-h represent restored images from 50% noisy color *Lena* image corrupted with RVN, by the filters SMF, AMF, DWM, EDBA, IDBA, BDND, FBDA and NASRI respectively. Figures i-r show the restored images by the proposed filters DWMD, ANDWP, VMM, EPR, SIR, GASIR, PSOSIR, NDE, DPSO and FSMF respectively 245
9.75 Comparisons of execution time (secs) for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images having 50% SPN 247
9.76 Comparisons of execution time (secs) for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images having 50% RVN 249
9.77 First row shows the original ultrasound image. Second row shows *noisy ultrasound image(σ=30)* and *restored image* from left to right, third row shows the *noisy ultrasound image(σ=60)* and *restored image* and fourth row show *noisy ultrasound image(σ=90)* and *restored image respectively.* 253
9.78 Graph representing comparison in *PSNR* for *Ultrasound* image corrupted by Gaussian noise and Poisson noise or Shot noise 254
9.79 a, b, c and d are restored images by VisuShrink, SureShrink, BayesShrink and the proposed shrinking method respectively while the noisy image has σ=50. 254

List of Tables

1 Indexing Database of Publications xix

4.1 Sample results of fitness evaluation 97

5.1 Suggested filtering window size against noise density (ND) 111

9.1 PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DWMD** filter for varying noises with RVN 150
9.2 IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DWMD** filter for varying noises with RVN 150
9.3 SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DWMD** filter for varying noises with RVN 151
9.4 PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **ANDWP** filter for varying noises with RVN 153
9.5 IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **ANDWP** filter for varying noises with RVN 154
9.6 SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **ANDWP** filter for varying noises with RVN 154
9.7 PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **VMM** filter for varying noises with RVN 155
9.8 IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **VMM** filter for varying noises with RVN 156
9.9 SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **VMM** filter for varying noises with RVN 157
9.10 PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **EPR** filter for varying noises with RVN 159
9.11 IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **EPR** filter for varying noises with RVN 160
9.12 SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **EPR** filter for varying noises with RVN 161
9.13 PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **SIR** filter for varying noises with RVN 162
9.14 IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **SIR** filter for varying noises with RVN 163
9.15 SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **SIR** filter for varying noises with RVN 164

9.16 Comparison of restoration results for the proposed algorithms in *Average PSNR* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density with RVN 167
9.17 Comparison of restoration results for the proposed algorithms in *Average IEF* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density with RVN 167
9.18 Comparison of restoration results for the proposed algorithms in *Average SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density with RVN 168
9.19 PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **GASIR** filter for varying random valued noises 172
9.20 IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **GASIR** filter for varying random valued noises 172
9.21 SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **GASIR** filter for varying random valued noises 173
9.22 PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **PSOSIR** filter for varying random valued noises 175
9.23 IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **PSOSIR** filter for varying random valued noises 176
9.24 SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **PSOSIR** filter for varying random valued noises 177
9.25 Comparison of proposed techniques in *Average PSNR* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with varying random valued noises . . 179
9.26 Comparison of results for the proposed filters in *Average IEF* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with varying RVNs 179
9.27 Comparison of results for proposed filters in *Average SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* with varying RVNs 180
9.28 PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **NDE** filter for varying SPNs and RVNs 185
9.29 IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **NDE** filter for varying SPNs and RVNs 188
9.30 SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **NDE** filter for varying SPNs and RVNs 188
9.31 Comparison of results for the proposed filters in *Average PSNR* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs 189
9.32 Comparison of the proposed filters in *Average PSNR* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs 193

9.33 Comparison of the proposed filters in *Average IEF* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs 193
9.34 Comparison of proposed filters in *Average IEF* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images RVNs 195
9.35 Comparison of the proposed filters in *Average SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs 196
9.36 Comparison of results for the proposed algorithms in *Average SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs 198
9.37 PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DPSO** filter for varying SPNs and RVNs 202
9.38 IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DPSO** filter for varying SPNs and RVNs 205
9.39 SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DPSO** filter for varying SPNs and RVNs 205
9.40 Comparison in *Average PSNR* of proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs 206
9.41 Comparison in *Average PSNR* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs 209
9.42 Comparison of *Average IEF* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs 209
9.43 Comparison of *Average IEF* for the proposed filter for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images RVNs 211
9.44 Comparison of *Average SSIM* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs 211
9.45 Comparison of *Average SSIM* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs 213
9.46 PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **FSMF** filter for varying SPNs and RVNs 216
9.47 IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **FSMF** filter for varying SPNs and RVNs 219
9.48 SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **FSMF** filter for varying SPNs and RVNs 219
9.49 Results on Bit Error Rate on 30% and 90% noisy Lena image corrupted with SPN and RVN by the proposed filters 224
9.50 Results on Bit Error Rate on 30% and 90% noisy Cameraman image corrupted with SPN and RVN by the proposed filters 225
9.51 Restoration results in terms of PSNR on Quantization noise on Lena and Cameraman images by proposed filters 226

9.52 Restoration results in PSNR (dB) on 50% anisotropic noise on Lena and Cameraman images by the proposed filters 226
9.53 Restoration results in terms of UIQI on SPN and RVN on Lena and Cameraman images corrupted by 50% noise density by proposed filters 227
9.54 Comparison of *Average PSNR* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs 230
9.55 Comparison of *Average PSNR* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs 233
9.56 Comparison of *Average IEF* for proposed filters in for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs 234
9.57 Comparison of *Average IEF* for the proposed filters in for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs 235
9.58 Comparison of *Average SSIM* for the proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs 237
9.59 Comparison of *Average SSIM* for the proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs 242
9.60 Comparison of **Execution Time** in Seconds for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPN 246
9.61 Comparison of **Execution Time** in Seconds for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVN 248
9.62 Comparisons of qualitative results in *PSNR (dB)* for *Ultrasound* image corrupted by Gaussian noise and Poisson noise or Shot noise 251

List of Algorithms

1 DWMD noise detector . . . 67
2 DWMD filter . . . 69
3 ANDWP noise detector . . . 72
4 ANDWP filter . . . 74
5 VMM noise detector . . . 78
6 VMM filter . . . 81
7 EPR noise detector . . . 90
8 SIR noise detector . . . 91
9 SIR filter . . . 92
10 PSO pseudo code . . . 101
11 Noise detection technique . . . 109
12 Noise density estimation technique . . . 110
13 Impulse filter . . . 112
14 PSO based clustering . . . 121
15 DPSO algorithm . . . 125
16 Noise detection and restoration technique . . . 128
17 Noise detection algorithm . . . 135
18 Filtering algorithm . . . 136
19 Fuzzy switching algorithm . . . 139

Part I

Introduction

Chapter 1

Introduction

Contents

1.1 Image Denoising 3

1.2 Types of Noises 9

The image is defined as a two dimensional function, f(x,y), where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y and the amplitude values of f are all finite, discrete quantities, then it is called a digital image. The field of digital image processing refers to processing of digital images by means of a digital computer.

Today digital image processing has a great impact on almost all areas of technical endeavor in some way [Chanda 2002]. The areas of application of digital image processing are so varied that some form of organization is desirable in attempting to capture the breadth of this field. The most easy way to develop the image processing applications is to divide images based on their source. The main source for images is the electro magnetic energy. Other includes acoustic, ultrasonic, and electronic. The image generation are categorized and the areas where they are applied, are given below.

Digital image processing is classified into various sub-branches [Gonzalez 1992] based on methods where:

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

The process of image processing and analysis are divided into these stages. They are, discretization and representation, processing and analysis. Here the input is a scene and the output is corresponding digital image. Then both the input and the output are images and the output is an better version of input. Finally, the input is again an image but the output is about the contents of the image [Gonzalez 1992].

1.1 Image Denoising

Image denoising means process of making a clean image when a noisy image is given. The noisy image is actually the sum of a clean image with a noise. So denoising is the process of decomposition. The objective is to divide a noisy image into a noise free image and the noise particle. Since an infinite number of such decompositions exist, one is interested in finding a plausible clean image, given a noisy one. The notion of plausibility is that the denoised image should look like an image, whereas the noise component should look noisy. Without prior knowledge, image denoising is very difficult. One can think of an image as a point lying in a high-dimensional space. Hence, image denoising involves moving from one point in a high-dimensional space (the noisy image), to a different point in the same space which is unknown a priori. Usually, it is impossible to find the clean image exactly. One is therefore interested in finding an image that is close to the clean image.

The goal of enhancement is to process an image where the resultant image is more suitable than the original image for a specific application. The word specific is important, because the techniques are very much problem oriented. For example, a method which is useful for enhancing X-ray images may not be efficient for enhancing remote sensing images. Image denoising method divided into two broad sections: one is spatial domain methods and another is frequency domain methods.

The spatial domain is the image plane itself, and techniques in this area are dependent on direct dealing with the pixels in that image but on frequency domain the operations are done on the transformed image. The techniques involved in spatial domain filtering are described in next chapter.

Frequency domain filters [1] process an image in the frequency domain. The image is Fourier transformed, multiplied with the filter function and then re-transformed into the spatial domain. Attenuation of high frequency signal makes a blurred image in the spatial domain and attenuating low frequency makes enhancement of the edges. The frequency domain filters can be converted to spatial domain when there exists a kernel for the output filter, it is easy to perform the filtering operation in spatial domain. Frequency domain filtering is more suitable when no kernel is present in spatial domain. That is also more efficient.

Frequency filtering is based on the transformation techniques. The operator usually takes an image and a filter function in the frequency domain. This image is then multiplied with the filter function in a pixel-by-pixel fashion, given in (1.1), where F(k,l) is the input image in the Fourier domain, H(k,l) the filter function and G(k,l) is the filtered image. To obtain the resulting image in the spatial domain, G(k,l) has to be reverse transformed using the

[1] http://homepages.inf.ed.ac.uk/rbf/HIPR2/freqfilt.htm

inverse transform equation.

$$G(k,l) = F(k,l)H(k,l) \tag{1.1}$$

Since the multiplication in the Fourier space is identical to convolution in the spatial domain, all frequency filters can be implemented as a spatial filter theoretically. However, in practice, the Fourier domain filter function can only be approximated by the filtering kernel in spatial domain. The form of the filter function determines the effects of the operator. There are basically three kinds of filters: *low pass, high pass and bandpass* filters.

1. A *low-pass filter* attenuates high frequencies and retains low frequencies unchanged. The result in the spatial domain is equivalent to that of a smoothing filter; as the blocked high frequencies correspond to sharp intensity changes, i.e. to the fine-scale details and noise in the spatial domain image.

2. A *high pass filter*, on the other hand, yields edge enhancement or edge detection in the spatial domain, because edges contain many high frequencies. Constant gray level areas have low frequencies and so it is suppressed.

3. A *bandpass filter* attenuates very low and very high frequencies, but retains a middle range band of frequencies. Bandpass filtering can be used to enhance edges (suppressing low frequencies) while reducing the noise at the same time (attenuating high frequencies).

The low pass filter suppresses all frequencies higher than the cut-off frequency *f*, (1.2) and leaves smaller frequencies unchanged. In most implementations, f is a fraction of the highest frequency represented in the Fourier domain image. The drawback of this filter function is a ringing effect that occurs along the edges of the filtered spatial domain image.

$$H(k,l) = \begin{cases} 1 : \sqrt{(k^2) + (l^2)} < f \\ 0 : \sqrt{(k^2) + (l^2)} > f \end{cases} \tag{1.2}$$

Better results can be achieved with a Gaussian shaped filter function. The advantage is that the Gaussian has the same shape in the spatial and Fourier domains and therefore does not incur the ringing effect in the spatial domain of the filtered image. A commonly used discrete approximation to the Gaussian is the *Butterworth filter*. Applying this filter in the frequency domain shows a similar result to the Gaussian smoothing in the spatial domain. One difference is that the computational cost of the spatial filter increases with the standard deviation (i.e. with the size of the filter kernel), whereas the costs for a frequency filter are independent of the filter function. Hence, the spatial Gaussian filter is more appropriate for narrow low pass filters, while the Butterworth filter is a better implementation for wide low

pass filters.

From 1990s, wavelets were found to be a perfect method of eliminating noises from a variety of signals. It can analyze the noise depth individually at each wavelet subband and can adapt the algorithm accordingly. In wavelet based thresholding method the wavelet coefficients are thresholded to remove their noisy positions. They do not need any certain knowledges about the nature of the signal and it allows discontinuities and spatial variation in the signal. The wavelet uses the spatially adaptive multiresolution analysis of the transformation. During the 1990s, the field was dominated by wavelet shrinkage and wavelet thresholding methods, including several papers by Donoho and Johnstone [Donoho 1994, Donoho 1995b, Donoho 1995a, Donoho 1995c]. The wavelet denoising method consists of four steps. These are, forward discrete wavelet transformation (DWT), determining the threshold, application of the thresholding using specific rule and inverse discrete wavelet transformation (IDWT).

The forward wavelet transformation converts the image from spatial based domain to wavelet based domain using (1.3) and (1.4), respectively. For inverse transformation the (1.5) is used.

$$O_l[k] = \sum_n x[n].h[2k-n] \tag{1.3}$$

$$O_h[k] = \sum_n x[n].g[2k-n] \tag{1.4}$$

$$G[n] = \sum_{k=-\infty}^{+\infty} O_h[k].g[2k-n] + O_l[k].h[2k-n] \tag{1.5}$$

where G[n] is the original one. The h[x] and g[x] are the half band low pass and high pass filters respectively. $O_l[k]$ and $O_h[k]$ are the outputs of low pass and high pass filters when sub sampling by 2.

The matrix is multiplied with scaling factors and the wavelet coefficients to have the forward transformation wavelet coefficients. Forward Haar transformation coefficients and wavelet coefficients are H_0=0.5, H_1=0.5, G_0=0.5 and G_1 = -0.5. For reverse transformation the column transformation is done first after the row transformation. For that, coefficients are H_0=1, H_1=1, G_0=1 and G_1=-1.

Thresholding is done forward transformed image prior to noise suppression. There exists a variety of approaches for thresholding the wavelet coefficients [Chang 2000a, S.Chang 2000, Donoho 1994, Donoho 1995b, Donoho 1995a].

The discrete wavelet transformation decomposes the noisy image into different frequency sub bands, labeled as LL_j, LH_k, HL_k and HH_k, where k=1,2,...,j. The implementation of 2-D discrete wavelet decomposition (one level) is shown in Fig. 1.1. The subscript are the

kth frequency level and j is highest scale in the process of decomposition. The subbands defines different properties about the image. The lowest frequency band LL_j is used to a coarse approximation of the image. The LH_k, HL_k and HH_k sub bands represents to the horizontal, vertical and diagonal information of the image signal, respectively. The highest frequency band is HH_k. The LL_k sub band is further decomposed in recursive manner into the sub bands LH_{k+1}, HL_{k+1} and HH_{k+1}.

On finding the threshold value, the wavelet coefficients are changed according to a shrinkage function T, given in (1.6).

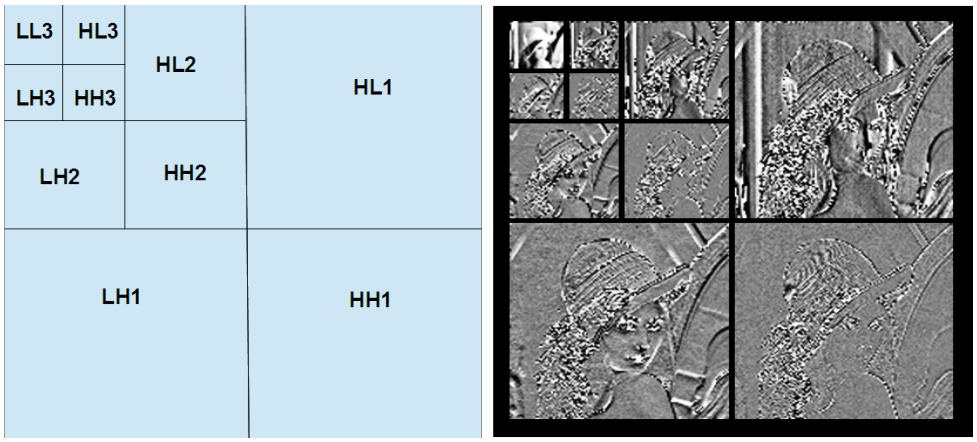


Figure 1.1: (a)Subimage representation of wavelet decomposition on level three (b)Wavelet decomposition on *Lena* image

$$F = T(G) \tag{1.6}$$

At end of shrinkage of the wavelet coefficients, it is transformed inverse to the original image domain given in (1.7).

$$R' = I^{-1}(F) \tag{1.7}$$

where I^{-1} is the inverse discrete wavelet transformation function and R' is the restored image.

The most complex task in wavelet denoising is to find out the optimal value of the threshold. A small value can intact the maximum portion of coefficients and that makes a signal and which cannot remove noise at all. When it is large, it may eliminate maximum portions of that. It can blur the signal which cannot preserve important details of the image. There are mainly three techniques to estimate the value of this thresholds, such as, VisuShrink [Donoho 1994], SureShrink [Donoho 1995b] and BayesShrink [H.Chipman 1997, S.Chang 2000].

VisuShrink [Donoho 1994] is applied on each level of the wavelet coefficients and this it is universal. This is shown in (1.8).

$$\lambda = \sigma\sqrt{2\log M} \tag{1.8}$$

where M is the number of pixels of the image and σ is standard deviation. The threshold is high and suffers from excessive blurring.

SUREShrink [Donoho 1995b] is sub-band adaptive thresholding method. It applies a different threshold on each sub-band depending on Stein's unbiased risk estimator (SURE). This is given in (1.9).

$$\lambda = \arg\min_{p\geq 0} SURE(p, N) \tag{1.9}$$

The Stein's unbiased risk is minimized in (1.10)

$$SURE(m, N) = y - 2\{j : |N_j| \leq p\} + \sum_{j=1}^{y} min(|N_j|, p)^2 \tag{1.10}$$

where N and y are the coefficients and number of coefficients in subband N. The technique is simple because the coefficients are sorted in terms of magnitude and then the threshold is selected to minimize the risk.

BayesShrink [H.Chipman 1997, S.Chang 2000] is also a subband adaptive thresholding technique. The thresholding method depends on normal distribution in each subband. The threshold given in (1.11).

$$\lambda = \frac{\sigma^2_{noise}}{\sigma_{signal}} = \frac{\sigma^2_{noise}}{\sqrt{max(\sigma^2_X - \sigma^2_{noise}, 0)}} \tag{1.11}$$

where $\sigma^2_X = \frac{1}{p}\sum_{i=1}^{p} Z_i^2$ and p is number of coefficients in $X_{i,j}$. This method uses a robust estimation on noise variance as median absolute value of the coefficients [Donoho 1994]. The noise variance is given in (1.12).

$$\sigma_{noise} = \frac{median(|X_{i,j}|)}{0.6745} \tag{1.12}$$

where $Y_{i,j} \in$ sub band HH, where $Y_{i,j}$ holds the coefficients in sub band HH which is the finest decomposition level.

Shrinking method defines the rules of applying the threshold to the wavelet coefficients. The threshold is compared to all coefficients of the wavelet domain and when the coefficients are less than the threshold value they are assigned zero values. On the other case they are untouched because very small coefficients are supposed to be not a part of signal elements

and so they can be modified to zeros. The large values are assumed to be a part signal elements. A number of work are done by Weaver et al. [J.Weaver 1991], Donoho and Johnstone [Donoho 1994, Donoho 1995b, Donoho 1995a, Donoho 1995c], Jansen [Jansen 2001] and Antoniadis [Antoniadis 2007]. They have proposed an extensive wavelet thresholding methods for image denoising.

In hard thresholding, the coefficients w=W_{xy} which are less than a threshold λ are assigned to zeros. Otherwise they are not touched. The hard thresholding technique is given in (1.13). One sample original image and corresponding hard and soft thresholding methods

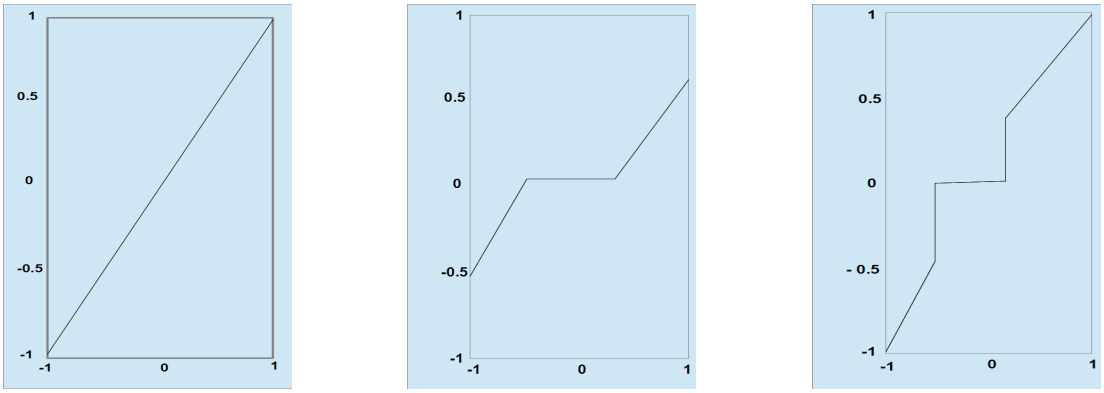


Figure 1.2: (a) Original Image (b) Hard Thresholding (c) Soft Thresholding

are presented in Fig.1.2, where the threshold (λ) is assumed 0.4.

$$Hard(w,\lambda) = \begin{cases} w & : |w| > \lambda \\ 0 & : |w| \leq \lambda \end{cases} \tag{1.13}$$

In soft thresholding, the coefficients which are higher than the threshold are reduced by an amount equal to the value of threshold. Otherwise they are assigned with zero values. The equation for soft thresholding technique is given in (1.14).

$$Soft(w,\lambda) = \begin{cases} sgn(w)(|w| - \lambda) & : |w| \geq \lambda \\ 0 & : -\lambda \leq |w| < \lambda \\ sgn(w)(|w| + \lambda) & : |w| < -\lambda \end{cases} \tag{1.14}$$

where sgn(w) returns the sign of the w.

Hard thresholding can abruptly discontinue which makes artifacts in the output image. Soft thresholding can make the output image over blurred. There are also several thresholding schemes in between the hard and soft thresholding.

1.2 Types of Noises

During any physical measurement, it is likely that the signal is corrupted by some amount of noise. The sources as well as the types of impulse depends on the physical instrument. It often arises from a source that is of different measurement, but sometimes it is because of the measurement process itself (e.g. shot noise/Poisson noise). Sometimes, it is caused because of the mathematical manipulation of a signal, like image deconvolution or image compression. Often, a noise is corrupted through several sources and it is a difficult task to characterize all of them completely. In all situations, impulse is an undesirable particle of the signal.

The characteristics of the impulse noise completely based on the acquisition process. It can be acquired by many ways, including, but not limited to: Digital and analog cameras, magnetic resonance imaging (MRI), computed tomography (CT), positron emission tomography (PET), ultrasonography, electron microscopy and radar imagery such as synthetic aperture radar (SAR).

Noise characteristics may be considered random variables, characterized by a probability density function (PDF). Most common PDFs found in image processing applications are Photon shot noise or Poisson noise, Thermal noise, Gaussian noise and compression artifacts.

Photon noise[2], also known as Poisson noise or Shot noise , is a basic form of uncertainty associated with the measurement of light, inherent to the quantized nature of light and the independence of photon detections. The magnitude of the signal makes the dominant source of the noisy image except in low light conditions. Images are sometimes affected by shot noise. This is due to the discrete nature of light. During a given exposure time, only a finite number of photons reach the imaging sensor. The number of photons going the sensor maintains a Poisson distribution. The standard deviation of distribution is square root of its expected value.

In probability theory and statistics, the Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known average rate and independently of the time since the last event [3].

Thermal noise arises due to the thermal energy of electrical and electronic devices. Thermally generated electrons accumulate in the chip's wells and are indistinguishable from photoelectrons. Thermal noise occurs even in the absence of light and is therefore sometimes referred to as dark-current noise. This type of noise is strongly dependent on the temperate of the sensor, but also on exposure time as well as the ISO-setting of the camera. Each pixel

[2] http://people.csail.mit.edu/hasinoff/pubs/hasinoff-photon-2012-preprint.pdf
[3] http://en.wikipedia.org/wiki/Poisson_distribution

can be approximately modeled as a Gaussian. Thermal noise is an example of noise which can be reduced by modifying the signal acquisition process: Cooling the camera's sensor reduces thermal noise.

Gaussian noise [4] in digital images arises during acquisition for example sensor noise caused by poor illumination and/or high temperature, and/or transmission for example electronic circuit noise. It is evenly distributed in the signal. The pixel is constructed with the true pixel value and a random Gaussian distributed noise value. Because of its mathematical relationship in both the spatial and frequency domains, Gaussian noise models are used frequently research. The probability distribution function of a Gaussian random variable, z, is given by

$$p(z) = \frac{1}{\sqrt{2\Pi\sigma}} * e^{\frac{-(z-z')^2}{2\sigma^2}} \tag{1.15}$$

where z represents gray level, z' is the mean of average value of z, and σ is its standard deviation. The standard deviation squared, σ^2, is called the variance of z. A graph is shown in Fig. 1.3. Digital images are usually stored in a compressed format such as JPEG. The

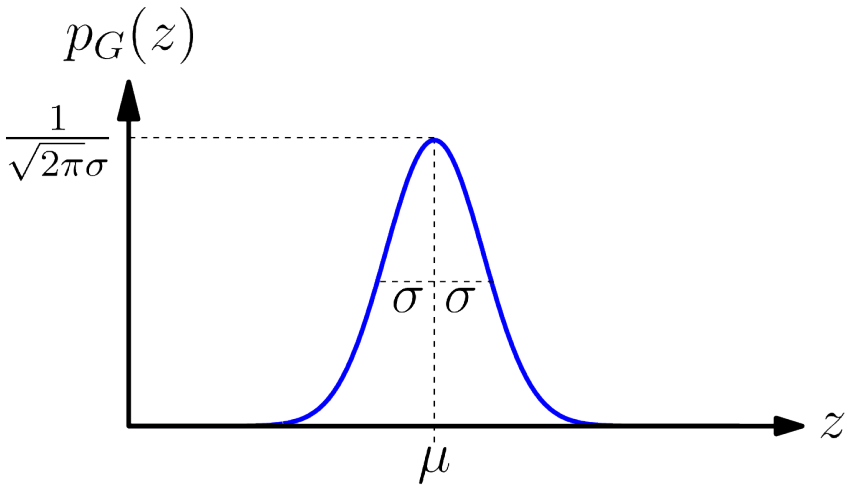


Figure 1.3: Probability Density Function of Gaussian Distribution

algorithm of compression rises to artifacts, which is assumed a type of noise. The algorithm is the most frequently used technique and which causes blocking artifacts. And also the information is lost due to the compression algorithm.

There are two noise models available in literature. These are salt and pepper and random valued noise. These standard impulse models [Nair 2010] are described as follows:

[4] http://en.wikipedia.org/wiki/Gaussian_noise

Noise is modeled as salt-and-pepper impulse noise [Wang 1999], pixels are randomly corrupted by two fixed extreme values, 0 and 255 (for 8-bit monochrome image), generated with the same probability. That is, for each image pixel at location (i, j) with intensity value S_{ij}, the corresponding pixel of the noisy image will be x_{ij}, in which the probability density function of x_{ij} is

$$f(x) = \begin{cases} \frac{p}{2} & : for\ x = 0 \\ 1 - p & : for\ x = S_{i,j} \\ \frac{p}{2} & : for\ x = 255 \end{cases} \tag{1.16}$$

Salt and pepper noise can also be found with unequal probabilities, that is,

$$f(x) = \begin{cases} p1 & : for\ x = 0 \\ 1 - p & : for\ x = S_{i,j} \\ p2 & : for\ x = 255 \end{cases} \tag{1.17}$$

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

Instead of two fixed values , impulse noise could be more realistically modeled by two fixed ranges that appear at both ends with a length of m each, respectively. For example, if m is 30, noise will equal likely be any values in the range of either [0, 3] or [226, 255]. That is,

$$f(x) = \begin{cases} \frac{p}{2m} & : for\ 0 \leq x < m \\ 1 - p & : for\ x = S_{ij} \\ \frac{p}{2m} & : for\ 255 - m < x \leq 255 \end{cases} \tag{1.18}$$

p is the noise density.

Random valued noise can be modeled with unequal probabilities, that is,

$$f(x) = \begin{cases} \frac{p1}{m} & : for\ 0 \leq x < m \\ 1 - p & : for\ x = S_{ij} \\ \frac{p2}{m} & : for\ 255 - m < x \leq 255 \end{cases} \tag{1.19}$$

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

Image quality [5] metrics are paramount to provide quantitative data on the fidelity of rendered images. Typically the quality of an image synthesis method is evaluated using

[5] http://en.wikipedia.org/wiki/Image_quality

numerical techniques which attempt to quantify fidelity using image to image comparisons often comparisons are made with a photograph of the scene that the image is intended to depict). That may generate some of distortion in the signal, so the quality judgment is an important problem.

The visible differences between each pair of images are determined by image quality metrics. The simple one is mean squared error (MSE), but which is not enough powerful. So better metrics are required.

By defining image quality in terms of a deviation from the ideal situation, quality measures become technical in the sense that they can be objectively determined in terms of deviations from the ideal models. The measures are Mean Square Error (MSE), Peak-Signal-to-Noise-Ratio (PSNR), Image Enhancement Factor (IEF) and Structure Similarity Index Measure (SSIM).

In statistics, the mean squared error or MSE of an estimator is one of many ways to quantify the amount by which an estimator differs from the true value of the quantity being estimated. It is used to calculate the difference between a original image with a restored image. PSNR (dB) analysis uses a standard mathematical model to measure an objective difference between two images. It calculates the quality of a output image with respect to an input image. The objective is to find a number that reflects the quality of the restored image. Restored images with higher Peak SNR (dB) are weighted [Brownrigg 1984].

Given an original image I_1 of size (M $\times$ N) pixels and a reconstructed image I_2, the PSNR (dB) is defined in (1.20).

$$f(I_1, I_2) = PSNR(dB) = 10 * \log_{10}(\frac{255^2}{\frac{1}{M*N}\sum_{m,n}(I_1m, n - I_2m, n)^2}) \tag{1.20}$$

where M and N are the number of rows and columns in the input images, respectively, and I_1 and I_2 are original and enhanced images respectively.

Image enhancement factor (IEF) is defined as

$$IEF(I_1, I_2, I_3) = \frac{\sum_{m,n}[I_1m, n - I_3m, n]^2}{\sum_{m,n}[I_1m, n - I_2m, n]^2} \tag{1.21}$$

The structural similarity (SSIM) index [Nair 2012] is a method for measuring the similarity between two images [6]. SSIM is an efficient operator over conventional methods like peak SNR (PSNR) and mean squared error (MSE). They are proven to be inconsistent with the human visual system. The difference with respect to other techniques mentioned previously such as MSE or PSNR is that these approaches estimate perceived errors, on the other hand, SSIM considers image degradation as perceived change in structural information. Structural

[6] http://en.wikipedia.org/wiki/Structural_similarity

information is based on the idea of strong inter dependencies particularly when they are spatially nearer. This idea is important about the structure of the objects in the visual system. Some results are also given in terms of Universal Image Quality Index (UIQI) [Wang 2002].

The SSIM value between images O and R having size N × N is given in (1.22). The SSIM value is a decimal value between -1 and 1.

$$SSIM = L(O,R) * C(O,R) * S(O,R) \tag{1.22}$$

where

$$L(O,R) = \frac{2\mu_O\mu_R + C_1}{\mu_O^2 + \mu_R^2 + C_1} \tag{1.23}$$

and

$$C(O,R) = \frac{2\sigma_O\sigma_R + C_2}{\sigma_O^2 + \sigma_R^2 + C_2} \tag{1.24}$$

and

$$S(O,R) = \frac{\sigma_{OR} + C_3}{\sigma_O^2\sigma_R + C_3} \tag{1.25}$$

where

- $C_1 = (K_1 * G)^2$, $C_2 = (K_2 * G)^2$ and $C_3 = \frac{C_2}{2}$
- G=255
- $K_1, K_2 \ll 1$
- $K_1 = 0.001$ and $K_2 = 0.002$
- O and R are the original and restored images respectively.
- μ_O is the mean of O and μ_R is the mean of R
- σ_O^2 is the variance of O and σ_R^2 is the variance of R
- σ_{OR} is the covariance of O and R respectively
- L represents *luminance* comparison, C represents *contrast* comparison and S represents *structure* comparison

The contribution of the thesis in this regard is to propose efficient algorithms for denoising the digital and medical images corrupted by various types of impulsive noises. The main objective and contribution of the thesis is to deal with random valued noises with high density. A set of algorithms have been proposed to deal with high random valued noises.

Some techniques outperforms in terms of noise suppression ability and computational cost of those algorithms. Characteristics of the techniques are that they can also suppress salt and pepper noise in addition to random valued impulse noise with variable probabilities of noise density. Some algorithms are based on standard deviation operation where as some are based on median operation of the neighboring pixels of the test pixel. A variety of median filters have been developed to deal for variable noise density. Most of the median filters developed are weighted. Some pixels within the test window are given extra weight when restorations are performed using the median operation. Filter optimization is performed by using two stochastic search algorithms such as genetic algorithm and particle swarm optimization techniques. Noise density estimation for noise detection and restoration is done in the thesis. Fuzzy switching median filter has been developed for digital image denoising. Medical image denoising is also done by using the two dimensional discrete wavelet transformation. The detail description of the proposed algorithms is outlined in the corresponding chapter.

Chapter 2

State of the Art

Contents

2.1 Denoising in Spatial Domain 19
2.2 Denoising in Frequency Domain 37
2.3 Background and Motivation 55
2.4 Contributions 57
2.5 Layout of the Thesis 59

Noise removal from a contaminated image signal is a prominent field of research and many researchers have suggested a large number of algorithms and compared their results. The main thrust on all such algorithms is to remove impulsive noise along with preserving image details. The methods are different in their methodologies which are applied to remove the noises. Some techniques utilize detection of impulsive noise followed by filtering where as other filters all the pixels irrespective of corruption [Brownrigg 1984]. In this section a detail literature review has been done on the reported articles. There are two basic approaches for image denoising, spatial filtering and transform domain filtering, as given in Fig. 2.1. The spatial filtering is a traditional way to remove noise from image data. Spatial filters can be further classified into non-linear and linear filters. With non-linear filters, the noise is removed without any attempts to explicitly identify it. Spatial filters employ a low pass filtering on groups of pixels with the assumption that the noise occupies the higher region of frequency spectrum. Generally spatial filters remove noise to a reasonable extent but at the cost of blurring images which in turn makes the edges in pictures invisible. Linear filters too tend to blur sharp edges, destroy lines and other fine image details, and perform poorly in the presence of signal-dependent noise. The transform domain filtering methods can be subdivided according to the choice of the basic functions. The basic functions can be further classified as adaptive and non-adaptive. Spatial-frequency filtering refers use of low pass filters using Fast Fourier Transform (FFT). In frequency smoothing methods the removal of the noise is achieved by designing a frequency domain filter and adapting a cut-off frequency when the noise components are decorrelated from the useful signal in the frequency domain. Wavelet domain filtering operations in the frequency domain can be subdivided into linear

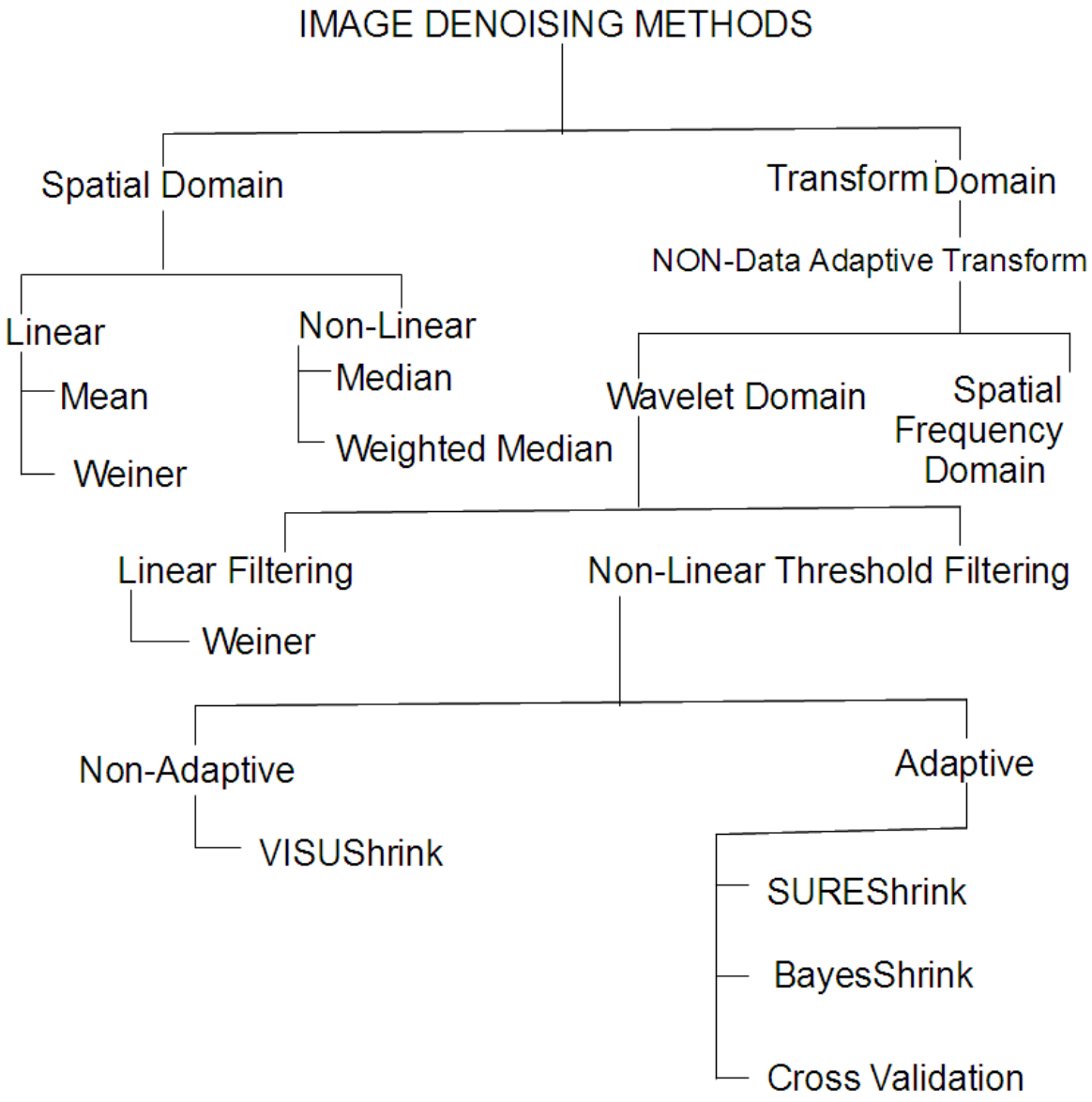


Figure 2.1: Image denoising techniques

and nonlinear methods. The Linear filters such as Wiener filter in the wavelet domain yields optimal results when the signal corruption can be modeled as a Gaussian process and the accuracy criterion is the mean square error (MSE). However, designing a filter based on this assumption frequently results in a filtered image that is more visually displeasing than the original noisy signal, even though the filtering operation successfully reduces the MSE.

The most investigated frequency domain in denoising using Wavelet Transform which is a non-linear coefficient thresholding based method. The procedure exploits sparsity property of the wavelet transform and the fact that the discrete wavelet transformation maps white noise in the signal domain to white noise in the transform domain. Thus, while signal energy becomes more concentrated into fewer coefficients in the transform domain, the noise energy does not. It is the logic that enables the separation of signal from noise. Most of the thresholding techniques are based on the methods for finding the optimal value of threshold which can would be adaptive or non-adaptive to the digital image. The non-adaptive thresholding method *VISUShrink* is non-adaptive universal threshold, which depends only on number of data points. It has asymptotic equivalence suggesting best performance in terms of MSE when the number of pixels reaches infinity. The adaptive thresholds like *SUREShrink* uses a hybrid of the universal threshold and the SURE (Stein's Unbiased Risk Estimator) threshold and performs better than VisuShrink. BayesShrink minimizes the Bayes' Risk Estimator function assuming Generalized Gaussian prior and thus yielding data adaptive threshold. BayesShrink defeats the SureShrink in all most every cases. Cross Validation replaces wavelet coefficient with the weighted average of neighborhood coefficients to minimize generalized cross validation (GCV) function providing optimum threshold for every coefficient.

2.1 Denoising in Spatial Domain

The spatial domain [Gonzalez 1992] refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image, based on neighboring pixels within a specific mask.

A spatial filter consists of a *neighborhood* and *a predefined operation* that is performed on the surrounded neighboring pixels. Filtering operation creates a new pixel which replaces the test pixel. A filtered image is produced as the center of the filter visits each pixel in the input image. Fig. 2.2 shows a random point on a digital image. The 3×3 neighbor of the center pixel is shown in the figure. If the operation performed on the image pixels is *linear*, then the filter is called a linear spatial filter. Otherwise the filter is *nonlinear*.

For example, in a 3×3 window, for any input pixel (x,y) the output pixel g(x,y) is given by sum of products of the coefficients and the neighboring pixels of the image. The output pixel g(x,y) is given in (2.1).

$$g(x,y) = \omega(-1,-1)i(x-1,y-1) + \omega(-1,0)i(x-1,y) + ... + \omega(0,0)i(x,y) + ... + \omega(1,1)i(x+1,y+1) \quad (2.1)$$

Here the coefficient $\omega(0,0)$ coincides with image value i(x,y), indicating that the mask is centered at (x, y) when the computation of the sum of products takes place. For a window

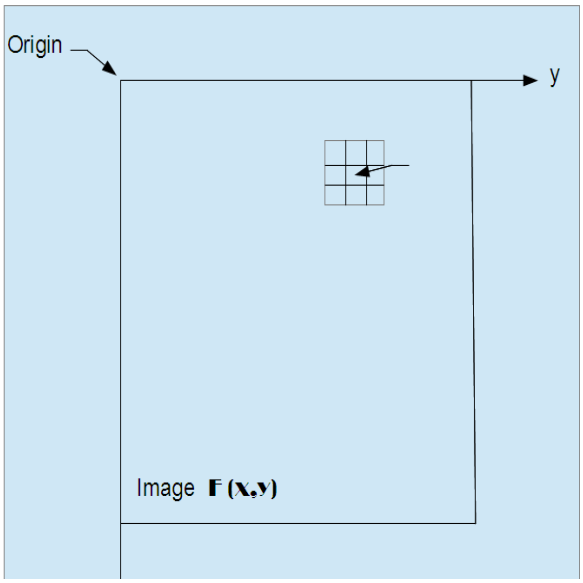


Figure 2.2: A 3×3 neighbor about a pixel (x,y) in an image.

of size m × n, it is assumed that m=2a+1 and n=2b+1, where a and b are nonnegative integers. Window will be of odd sizes, with the smallest meaningful size being 3 × 3, where the trivial case is 1 × 1 mask.

In general, linear filtering of an image f of size M × N with a filter mask of size m × n is given in (2.2).

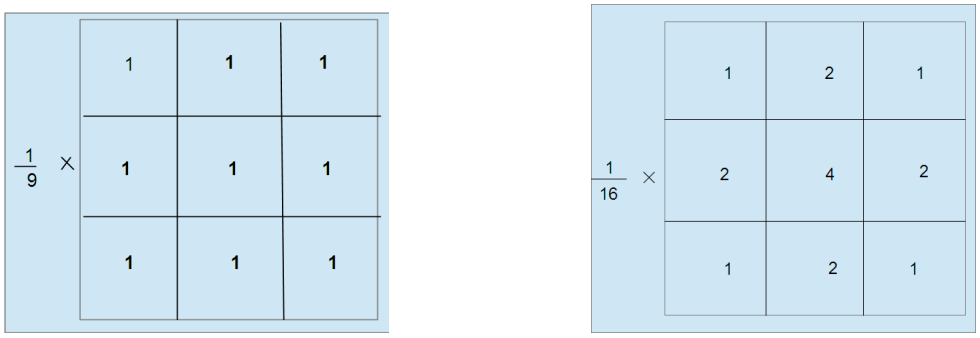
$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t) f(x+s, y+t) \tag{2.2}$$

where x and y are varied so that each pixel in w visits every pixel in f.

Smoothing filters can be used for blurring purpose and also noise reduction. Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering.

The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters are sometimes called averaging filters.

The idea behind smoothing filters is straightforward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask, this process results in an image with reduced "sharp" transitions in gray levels. Figure 2.3 (a) shows a 3 × 3 smoothing filter. The filter yields the standard average of the pixels under the mask. This can best be seen by substituting the coefficients of the mask into (2.3).



(a) (b)

Figure 2.3: 3 × 3 smoothing filters of an image.

Figure 2.3 (b) shows another 3 × 3 smoothing filter. They also yield the standard average of the pixels under the mask. This can be best seen by substituting the coefficients of the mask into (2.4).

$$R = \frac{1}{9}\sum_{i=1}^{9} z_i \tag{2.3}$$

$$R = \frac{1}{16}\sum_{i=1}^{16} z_i \tag{2.4}$$

A m × n mask would have a normalizing constant equal to $\frac{1}{mn}$. The mask shown in Fig. 2.3 (b) is much more interesting. It makes so-called weighted average and the technique used to indicate that pixels are multiplied by different coefficients, thus giving more weight to some pixels at the expense of others. In case of the mask presented in Figure 2.3 (b), the pixel at the center of the mask is multiplied by the higher value than any other, thus giving this pixel more importance in the calculation of the average.

The general implementation for filtering an M × N image with a weighted averaging filter of size m × n (m and n odd) is given by (2.5). For x=0, 1, 2, p, M-1 and y=0, 1, 2, p, N-1, the denominator in (2.5) is simply the sum of the mask coefficients and, therefore, it is a constant that needs to be computed only once. Typically, this scale factor is applied to all the pixels of the output image after the filtering process is completed.

$$g(x,y) = \frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b}\omega(s,t)f(x+s,y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b}\omega(s,t)} \tag{2.5}$$

Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the

value of the center pixel with the value determined by the ranking result. The best-known example in this category is the median filter, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel. Median filters are useful because they obtain good restoration results with less smoothing effects than the linear filters. Median filters are generally useful when the image has salt-and-pepper noise.

In order to perform median filtering at a point in an image, the center pixel and its neighbor values are sorted first then their median is determined, lastly this value is assigned to that pixel. For example, in a 3 × 3 neighborhood the median is the 5th largest value, in a 5 × 5 neighborhood the 13th largest value, and so on. For example, suppose that a 3 × 3 neighborhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100). They are ordered as (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in a median of 20. Thus, the principal function of median filters is to force points with distinct gray levels to be more like their neighbors.

Various algorithms exist in literature regarding spatial domain filtering. A few of them are surveyed in this section.

There exists some filters which perform filtering operation without detection . Here, a test window is moved throughout the image. The window has size 2M + 1 × 2M + 1, M is positive integer. The center pixel is under consideration. The window is moved from the top left position to the bottom right position. It performs some operations without classifying the pixel. The technique does not detect the noisy pixels.

The averaging filter [Gonzalez 1992] is simplest linear filter. The average value of neighboring pixels in the window replaces the center pixel using in (2.6).

$$R_{i,j} = \frac{1}{p * q} \sum_{(u,v) \in W_{pq}} N_{c,d} \tag{2.6}$$

where, N and R are noisy and restored images respectively and W_{pq} is test window having size p × q centered around (i,j). Using this technique, the restoration results is very poor in terms of quantitative and qualitative measures.

The median filters [Brownrigg 1984] are nonlinear filters. The equation shown in (2.7) is very popular nonlinear filter. The filter can be implemented easily and is efficient also. The disadvantage is it smoothes the restored image and fine textures may be lost. It performs like a low pass filter which prevents the high frequencies of image such as the edges and noise, and thus smoothes the image.

$$R_{i,j} = median_{(u,v) \in W_{pq}}(N_{c,d}) \tag{2.7}$$

Based on the window size, many variations of median filters exist. Two of them are surveyed here. Median (3×3) filter operates on 3×3 mask, where the center element is replaced by the median value of all 9 neighboring pixels within the window. This is useful because the noisy pixels are scattered in the image matrix. In (5×5) median filter the center pixel is replaced by the median value of all 25 neighboring pixels within the test window. This type of filter is useful when noise comes in blotch in the noisy image. In case of variants the filter does not perform well.

Some other filters exist which uses the principal of detection followed by filtering technique [Sun 1994a] during noise suppression . This technique involved in two steps. The first step identifies the noisy pixels and in second step it filters those pixels. A mask is moved across the image and some arithmetic operations are carried out to detect the noisy pixels. Then filtering operation is performed only on those pixels which are found to be noisy in the previous step, keeping the noise free pixels intact. Some of the methods are described here.

The rank-ordered mean [E. Abreu 1996] filter is another two step method which performs noise detection and then filtering operation on the noisy pixels. This filter is adaptive where restoration is performed on the current state of the algorithm. It is defined by the output of an operator which calculates the differences between the current pixel and the other sorted pixels where the center pixel is under question. So it is robust and simple. The does not preserve the image fine textures during restoration.

Eduardo Abreu *et al.* [E. Abreu 1996] has proposed a technique which eliminate the noises from images where noise types are Gaussian and mixed Gaussian noises. This method operates using a simple two-step method where it does switching between the output of an identity filter and a rank-ordered mean (ROM) filter. It always try to get the tradeoff between noise suppression and image fine texture preservation. It has low computational complexity as well. Simulation results show that it performs better in terms of noise suppression and preservation of fine details than other nonlinear filters with maximum 40% noise density. The filter performs well on both monochrome [E. Abreu 1996] and color [Moore 1999] images.

Russo and Rampani [Russo 1996] proposed a fuzzy filter for enhancement of digital images corrupted by impulse noises. This technique adopts a two step fuzzy reasoning method where the first subunit is *action detection module* which detects the noises by considering luminance differences among neighboring pixels. The second subunit is *action adaptation module* which modifies the value of the correction to preserve the image fine textures. This filter performs better in terms of MSE than the other mentioned filter in [E. Abreu 1996].

Kong and Guan [Kong 1996] proposed a neural network based adaptive filter which used pixel classification by a self-organizing neural network to detect the positions of the noisy pixels. The filter is able to effectively eliminate the impulses while retaining image integrity. In this method, a self organizing neural network is used to detect the positions of the noisy pixels. After the noise detection, a noise exclusive median filter is proposed to eliminate the noises. The noise exclusive method is used to exclude the noisy pixels while the median operation is done. This operator performs significantly better than the traditional median filter.

The minimum-maximum exclusive mean [Han 1997] filter is simple nonlinear filter. It is also robust filter where it centers around two window regions having sizes 3×3 and 5×5 respectively. It first takes for a certain range of intensity values within the 3×3 window and then if it fails it takes the larger window 5×5. If the average value of the pixels of the particular range is greater than a threshold then it is replaced by the average value, otherwise is kept unchanged. The method is useful because it is simple and can be implemented easily.

Beaurepaire et al. [Beaurepaire 1997] proposed a denoising method where the *histogram* information of the digital image is for consideration. The histogram of homogeneous regions are considered to characterize and to detect the noisy pixels. The homogeneity level can be found in [Pok 2003].

The MRHF-1, MRHF-2 and MRHF-3 [Khriji 1998] are the class of non-linear filters which is also known as Median Rational Hybrid Filters. These are depended on a particular rational function. The output of the filter is taken by the output of a rational operation which takes three sub functions. Among the three filters, the MRHF-2 performs better the other two.

The histogram information of the input image is used by Wang [Wang 1998] to determine the parameters of the membership functions of an adaptive fuzzy filter. Then the filter is used for denoising the digital images. An adaptive vector filter exploiting histogram information is presented for restoration of color images [Ma 2003].

A switching median filter has an impulse detector to classify the center pixel. If the center pixel of the test window is detected a noisy then it is replaced by the median value of the window. As a consequence, various impulse detection methods and switching median filters are presented [Sun 1994b, Chen 2001a, Zhang 2002, Pok 2003, Crnojevic 2004, Emin Yüksel 2004, Yüksel 2004].

The Tri-State Median Filtering [Chen 1999] is a combined filter which comprises of stan-

dard median filter, identity filter, center weighted median filter and a switching logic. Noise detection is done by impulse detection method, which takes the outputs from the standard median and center weighted median filters and compares them with the center pixel value in order to make a tri-state decision. The switching method is based on a threshold value. Based on that value, the center pixel is restored by the output of either Simple Median filter or Center Weighted Median filter.

Progressive-Switching Median [Wang 1999] is a median based filter, which works in two stages. Initially an impulse detector is used to generate a sequence of binary flag images. This binary flag image predicts the location of noise in the observed image. Later the noise filtering method is performed through several iterations. This filter is a very good filter for fixed valued impulsive noise but for random values the performance is very poor.

Tao Chen and Wu [Chen 2001b] presented an adaptive impulse detection method using center-weighted median filters to deal fixed valued impulses and random valued impulses. This method devises an adaptive operator which forms estimates based on the differences between the current pixel and the outputs of center-weighted median (CWM) filters with varied center weights. This filter outperforms the existing operators in terms of PSNR (dB) but the filter does not perform well when the images have high density of fixed valued impulses or random valued impulses.

The Peak and Valley [Windyga 2001] filter is a nonlinear filter which operates recursively. This filter is composed of two conditional rules. It checks the test pixel with neighboring eight pixels against some conditions. Then replaces the noisy pixel by the most suitable neighboring pixel. The method is efficient than others, but it also hampers noise free pixels.

A modified peak and valley filter [Alajlan 2004] has also been proposed which provides good detail preserving performance but suffers from the earlier version of peak-and-valley in terms of computational cost.

The adaptive center weighted median filter [Chen 2001a] is an improved median filter. It operates on the estimations on the differences between the current pixel and the outputs of the center weighted median filters with different weights. The estimation amount helps to switch between the center pixel value and the median of the neighboring pixels. The method is robust for a wide variety of images, but it fails to find the most suitable value to replace the noisy pixel.

Tao Chen *et al.* [Chen 2001d] proposed a filter which is based on space variant median

operations to restore the impulse noise. It is a general method of median based switching schemes, which is known as multi-state median (MSM) filter. By using simple thresholding rule, the output is adaptive to a group of CWM filters which use distinct center weights. Consequently the filter is equivalent to an adaptive CWM filter with a varying center weight. The filter is based on local signal informations. The filter performs better than the existing filters in the literature, however cannot provide good restoration results having high density of RVN in the images.

The partition based median filter [Chen 2001c] is an adaptive median filter that classifies the sample vector at each pixel position into one of the mutually exclusive clusters. The partitions are obtained by using of a number of center weighted median filters with variable center weights. Estimation at each pixel position is computed as a linear combination of the outputs of the center weighted median filters. The weights are optimized using LMS algorithm. This algorithm is capable of denoising both the impulse noise and Gaussian noise.

Cristian and Agostinho [Munteanu 2001] proposed an evolutionary image enhancement method based on user behavior modeling. This method is presented for image enhancement of gray-scale images based on the simulation of evolution. It employs Genetic Algorithm to evolve the shape of the contrast curve in the image, while attempting to partially automate the subjective process of image evaluation (e.g. user behavior) by performing multiple regressions on fitness values. It employs image enhancement using real coded GA rather than binary coded GA. Real-coded GAs use real coded genes in the chromosome, which means that each free parameter of the optimization problem is represented by one real-valued gene in the chromosome. The initial population is generated randomly, that is N random chromosomes. Following the generation of the initial population, the chromosomes are evaluated by calculating their fitness function, and selection is applied to choose for the better fitted chromosomes. The output of the selection has a population of N chromosomes which are also used to perform crossover and mutation to the next generation. The main advantage of the algorithm is that it aims at systematizing and easing the process of human interpretation of images. An attempt may be done on this work to extend the mechanism of user behavior modeling by including other complex paradigms such as machine learning models. These models will be expected to further reduce the number of images the user has to evaluate during the run of the GA, while maintaining a high level of image enhancement.

Adaptive Two-Pass Median [Xu 2002] deploys median filter on the noisy image twice. This algorithm tries to accommodate for false substitutions generated by the first step of median filtering technique. Depending on the estimated distribution of the impulse, the pixels replaced by first median filter are also changed by their original values and kept unaltered

in the second phase. In the second round it filters out the remaining impulses. However the filter obtains good restoration results in terms of noise suppression parameters but it spoils the noise free pixels and also it performs poorly.

Accurate Noise Detector [Kondo 2002] is a filter based on Progressive Switching Median Filter, which generates an edge flag image to classify the pixels of noisy image into ones in the flat regions and edge regions. The two types of pixels are processed by different noise detector. When noise is very high prevention of false-detection and non-detection becomes difficult. Therefore, one iteration is dedicated for verification of the noise flag image. This scheme exhibits good results not only with low noise density but also with high percentage of corruption. But computational complexity is very high and not applicable in real time situations.

The SMF (5×5), SMF (7×7) and SMF (9×9) [Zhang 2002] are also a two stage process, where in the first stage noise detection is carried out and in the second stage filtering is done. The noisy image is gets convolution with a set of image kernels. Each of the kernels is sensitive to edges in a different orientation. The minimum absolute value of these four values is used in impulse detection by checking with a threshold value. By varying the size of kernel different variations of SMF may be obtained. As the characteristics of these four kernels it classifies the noise efficiently when the edge density is much more. But when the kernel size increases to 7×7 and 9×9 it fails to obtain good results. Also it fails in preserving image finer details.

Bogdan Smolka [Smolka 2003] proposed an algorithm for noise reduction in color images where the noise types are impulsive and Gaussian. This filter has an added advantage that it can enhance the sharpness in the restored images. This operator is a smoothing one which is based on a random walk model and a fuzzy similarity measure between pixels connected by digital geodesic paths. Experimental results or this filter show that it is useful for segmentation of digital images some extent, which is also useful for reducing the mixed impulsive noises and Gaussian noises in the digital images. The technique suffers from high computational complexity.

Vladimir Crnojevic *et al.* proposed an impulse removal method [Crnojevic 2004] based on pixel-wise median of the absolute deviations from the median is used to detect the noisy pixels from the digital images. The algorithm is free from varying parameters, requires no previous training or optimization, and successfully removes all types of impulse noise. The pixel-wise MAD concept is straightforward, low in complexity, and achieves high filtering results. The median of the absolutes deviations from the median-MAD given by Hubber

[Hubber 1981] is the single most useful ancillary estimate of scale, is used to estimate the presence of image details, thus it separates the noisy image pixels efficiently. An iterative pixel-wise modification of MAD provides excellent removal of impulses which are may be randomly distributed impulse noises.

Nikolova et al. proposed a variational method [Nikolova 2004], is capable of preserving details which restoring impulse noise. It uses non smoothing data fitting with edge preserving regularization functions. A combination of this variational method with a noise detector was proposed by Chan [Chan 2004] for removal of random valued noises. It performs better than existing filters but computational cost is larger than other methods.

The differential ranked impulse detector [Butakoff 2004] is another nonlinear technique which also works in two stages. It tries to filter only noisy pixels. Such pixels are identified by comparing signal samples within a rank mask by using both rank and absolute value. The first estimation method is dependent on the comparison among the rank of the pixel of consideration and rank of the median value. The next estimate is based on the brightness value which is analyzed using the median. It is a good filter in low noise conditions but the results slightly degrades if noise level increases above 20%. It also leaves noise blotch as uncorrected.

The two-output nonlinear filter [Russo 2004] is based on the subsequent activation of two recursive filtering algorithms that operates on different subsets of input data. One subset is the right-bottom 3×3 sub-window and the other one is left-top 3×3 sub windows of a 4×4 sliding window. The center pixels of the 3×3 window regions are updated at each stage. This is a good technique for impulsive noise and gives very good result under fixed valued impulsive noise conditions. But under RVN it fails miserably.

The enhanced ranked impulse detector [Butakoff 2004] is an alternative of the technique described above. The brightness is measured by calculating the difference of pixels of consideration with its closest neighboring pixels in the variational series. Its performance is very good at low noise denoising but fails miserably for noise density more than 20%.

The advanced impulse detection based on pixel-wise MAD [Crnojevic 2004] which is modified median of absolute deviation from median (MAD). It is useful in estimating the presence of image fine details. An iterative pixel wise modification of MAD is performed which gives a reliable elimination of impulses. It is more than average and fails when the edge density is more.

The two output non-linear filter [Russo 2004] is a rank order filter based on subsequent

aviation of two recursive filtering algorithms that operate on several sets of input data. The pixel values are updated at each iteration. A non-linear method is also given for error correction to avoid blurring. The filter performs well in terms of detail preservation.

The detail preserving impulsive noise removal [Alajlan 2004] technique is unlike thresholding techniques, it detects noisy pixels non-iteratively using the surrounding pixel values. It is based on a recursive minimum maximum method of Peak and Valley scheme. When the image contains numerous edges like Baboon, Clown etc., it suffers.

An adaptive partition fuzzy median (PFM) filter based on fuzzy rules for image restoration [Lin 2004] proposed by Tzu-Chao *et al.* to improve the median filter to preserve image fine details while effectively suppressing impulsive noises in the digital images. The filter obtains the effect through a summation of the weighted output of the median filter and the weighted input signal. The weights are assigned based on the fuzzy rules. In order to design this weight function, a method for partitioning of the observation vector space and a learning approach are proposed so that the mean square error of the filter output can be minimum. Based on the least mean square algorithm, an iterative learning method is devised and the corresponding convergence method is investigated. Training the filter over a refinance image with the constrained LMS algorithm obtains the optimal weight coefficients. This filter is capable of getting desirable results in noise suppression.

Kundra *et al.* [Harish 2005] proposed a filter for removal of salt-and-pepper noise by using fuzzy logic. Fuzzy image processing is a collection of all approaches that understand, represent and process the images, their segments and features as fuzzy sets. The representation and processing is dependent on the fuzzy based technique and on the problem to be concurred. This method combines the features of image enhancement and fuzzy logic. It also deals with fuzzy inference system (FIS) which help to take the decision about the pixels of the image under consideration. This technique focuses on the removal of the impulse noise with the preservation of edge sharpness and image details along with improving the contrast of the images.

The threshold boolean filter [Aizenberg 2005] makes use of boolean functions denoising the digital images. The noisy image is decomposed into number of binary images by gray level intensity thresholding. The detection and restoration of impulse noise are performed in these binary images by using the boolean functions which were specially made for such purpose. Finally the resultant boolean images are merged back to obtain the restored digital image.

An operator for restoring digital images corrupted by impulse noise has been performed by Yuksel [Yuksel 2006]. The operator is a hybrid filter obtained by appropriately combining a median filter, an edge detector, and a neuro-fuzzy network. The internal parameters used in the neuro fuzzy technique are adaptively optimized by the training method. The training is easily accomplished by using simple artificial images that can be generated in a computer. The most distinctive feature of the operator over most other operators is that it offers good line, edge and texture preservation while noise suppressing, at the same time, effectively removing noise from the input image. The need of training of the system and higher number of arithmetic operations, the computational cost of the operator is higher than the most of the other filters.

Pei-Eng Ng *et al.* proposed a switching median filter with boundary discriminative noise detection for extremely corrupted digital images [Ng 2006]. To determine whether the current pixel is corrupted, the algorithm classifies the pixels in a localized window, centering on the current pixel, into three groups as lower intensity impulse noise, un-corrupted pixels and higher intensity impulse noise. The center pixel will then be considered as un-corrupted provided that it belongs to the un-corrupted pixel group, or corrupted. Two boundaries are taken which discriminate the three groups. It works fine up to 70% noise density, for the two noise models. The main advantage of this algorithm is, it is simple for implementation, but shortcoming is the initial window size which is 21×21, which is not affordable. For higher noise density the algorithm also suffers from false alarm rate.

Ung-Keun *et al.* [Cho 2006] proposed an evolutionary algorithm for image enhancement and impulse noise reduction. The method uses the genetic algorithm to find a filter set for reducing impulsive noises in digital images. The salt-and-pepper noise is used to corrupt the digital images while the algorithm works to suppress the noise from *Lena* image. There are m number of filters available, the number of possible ordered subsets of n filters is m^n. This technique finds the proper type and order of filters using genetic algorithm. The operators of the genetic algorithm such as encoding, fitness evaluation, selection, cross over, mutation and elitism strategy are properly maintained in the algorithm to find the optimal solution. To find the filters which will be used for the search algorithms is difficult to determine. That is why the algorithm is not so meaningful.

To remove random-valued noises from the digital images a directional weighted median filter (DWM) [Dong 2007] is proposed by Dong and Xu. This filter uses a new impulse detector, which is based on the differences between the current pixel and its neighbors aligned with four main directions. In the filtering stage, it continues to use the information of the directions to weigh the pixels in the window to keep the image details. It considers a 5×5

window. Then it considers the four directions: horizontal, vertical and two diagonals. Each direction there has 5 pixel points. It then calculates the weighted difference in each direction and takes the minimum of them. The minimum value is thresholded against a value and if it is larger than the value then it is detected as noisy pixel otherwise it is noise free. In filtering phase, it finds the standard deviation in all four directions. As the standard deviation defines the tightness of the pixels around the mean in the set of pixels, so the pixels aligned in the direction are the nearest to each other. So the center value needs to close to them. Now it finds the weighted median value, assigning extra weight on the direction where standard deviation is minimum and replaces the noisy one with this median value. It is an iterative method. This method requires 8 to 10 iterations. It gives the good performance when noise level is also high.

Srinivasan et al. [Srinivasan 2007] proposed a decision-based algorithm for removal of impulse noises in digital images. The algorithm removes corrupted pixel by the median value or by its neighboring pixel value. The proposed method removes the noise effectively at noise level as high as 90% and preserves the edges. In the detection phase it decides for a pixel by checking whether the value of the center test pixel lies within the minimum and maximum intensity range of the selected window. If it lies within this range then it is said to be noise free pixel otherwise it is classified as noisy. if the test pixel is detected as noisy then it is replaced by median value of the test window or by its neighboring value. If the noise density is high, then the pixel processed is replaced by the previously adjacent neighbor pixel value in position of the median value. The algorithm shows stable performance across noise densities up to a high range. The limitations of the algorithm is that is it cannot perform well for both the noises types fixed valued noise and random valued noises.

Low *et al.* [Teck 2008] proposed an extended model on removing Gaussian noise in mechanical engineering drawing images using median filters. An extended study is performed on the effect of different factors on the quality of vector data. For the noise cleaning methods, it uses median filters and its variants. In case of vectorization factor, the best available commercial raster to vector software is namely VPstudio, which is used to convert raster images into the vector format. The performance is tested based on the objective evaluation criteria. But it cannot deal with other types of noises.

Borisov [Borisov 2008] proposed a optimal filtration method against noises in signal frequency at the frequency detector output against flicker noises with the power spectral density is considered. In a stationary approximation, a simple fractionally rational model of the spectra of frequency finite-power flicker noises is given. The optimal impulse response function of the device, which ensures an effective, unbiased, and consistent estimate of the parameter

with regard to the edge effects at the ends of the measurement interval and the optimal frequency estimate variance are obtained.

Sa et al. [Sa 2009] proposed an operator for removal of random-valued noise based on second order difference based median filter (SODM). This filter uses a 3 x 3 window and which is based on the differences between the current pixel and its neighbors aligned with four main directions. After impulse detection, it not only simply replace noisy pixels identified by outputs of median filter but also continue to use the information of the four directions to weight the pixels in the window in order to preserve the details as removing noise. It considers the four directions as horizontal, vertical and two diagonals. In each direction there are two end pixel points. It then calculates the absolute differences in each direction from the test pixel and takes the minimum of them. The minimum value is thresholded to detect a noisy pixel. In filtering phase, it again calculates the absolute differences in each direction between the end pixels in four directions. In the similar way like DWM it selects a direction for which absolute difference is minimum among the four and the gives more weight to those directional pixels. As the center value should also be close to them. Now it calculates the weighted median, giving extra weight on that direction in which absolute difference is small and replaces the noisy pixel with this median value. It performs well and operates iteratively under low noise level.

Haixiang et al. [Xu 2009] proposed an adaptive fuzzy switching filter that adopts a fuzzy logic approach for the enhancement of images corrupted by impulse noise. In order to achieve optimal detail preservation, the maximum-minimum exclusive median method has been adopted to estimate the current pixel. By making convolution operation with the *Laplace* operator it first gets four convolutions. The minimum one is selected and compared with a threshold value to classify the test pixel. Then it estimates the corrupted pixel by using maximum-minimum exclusive median method to the current pixel. Eventually a fuzzy switching method is proposed by the authors to filter the noisy pixels to a great extent. This method can effectively suppress the salt-and-pepper noises in digital images. Although the method is not tested for other types of impulsive noises.

Xuan et al. [Guo 2009] proposed an adaptive hybrid filtering method for removing impulse noise in color images. It combines a group of sigma vector median filters with different thresholds with a filter based on neuro-fuzzy system for color image processing. The first subunit of the filter is six sigma vector median filters, outputs oh which are used as optimum initial points to input the second subunit constituted by a simple *Sugeno-type* neuro-fuzzy system, and then the optimized result is obtained from the output of the second subunit. The parameters of the neuro-fuzzy model are automatically tuned and fixed by a learning

method based on genetic algorithm. From thr results it is seen that the design of the proposed hybrid filter has met the requirement of removing impulse noise along with preserving details.

Mandal et al. [Mandal 2010] proposed a modified directional weighted median filter for removal of random valued noises in digital images. This operator calculates the differences between the center pixel and its neighboring pixels in the four main directions in the 5×5 window of the test pixel. The minimum of these four direction indices are used for impulse detection. On classifying the test pixel as noisy, the filtering method replaces the noisy pixels by median values of the window depending upon four directions. Standard deviation followed by median operations are performed on the 5×5 window based pixels to get the resultant value. This performs well but when the noise density is higher than 60% the operator suffers to outperform the other operators in the literature.

An improved neural network based filter proposed by Chao *et al.* [Deng 2010] for removal of impulses, where a neural network is used to detect the noisy pixels and then distinguish it from noise-free pixels. The noisy pixels are then replaced by the suitable pixel value which has the most local similarity. Finally the output is the mixture of the noise free pixels and the suitable pixel. The noise is experimented in this algorithm is salt-and-pepper noise with uniform distribution. So each pixel has equal chance of getting 255 to 0 or 0 to 255. A 3×3 window is selected to pass through the entire image region. The median value of the window is used as input values to the neural networks. A different back propagation method is used in this algorithm rather than gradient descent optimization approach. The noisy pixels are classified based on the local similarity measure.Based on the classification the noisy pixel is replaced by the original pixel value and the median value by a fuzzy switching method. The algorithm performs excellent for salt-and-pepper type impulsive noise but may be tested for other types of impulse models.

Nasimudeen et al. [Nasimudeen 2010] proposed a noise detection algorithm for impulse noise removal, called the boundary discriminative noise detection by elimination (BDNDE), which retains the good characteristics of the BDND filter [Ng 2006] while suppressing noise. To determine whether a pixel is corrupted or not, the algorithm first sets the minimum and maximum boundary (threshold) values based on the localized window centered on the pixel. The thresholding method tries to achieve low false alarm and miss detection rate when the noise is RVN. The experiments are conducted on gray scale images under a wide range (from 10 to 90%) of noise corruption. this algorithm outperforms the BDND median-based filters, in terms of suppressing impulse noise while preserving image details. The method is algorithmically simple and faster, compared to existing BDND. Firstly it imposes a 21×21 test window where the center pixel is the test pixel. It then finds the maximum and minimum

However it may be tested with other noise models also.

Moacir et al. [Ponti-Jr 2013] proposed a three dimensional image restoration algorithm based on extrapolation and deconvolution. The fluorescence microscopic images have to deal the effect of blurring and photon noise simultaneously. Their combined effects corrupt the images by inserting elements that do not belong to the real object and distort the contrast of the image apparently. This obstructs the way of using the images for visualization, recognition and analyzing to use the three dimensional data. The algorithms developed in this citation are to restore the lost frequencies and perform band extrapolation in an additive noise circumstance. This algorithm presents a restoration approach through band extrapolation and deconvolution that deals with the noise.

Nasri *et al.* [Nasri 2013] proposed a switching non-local means (SNLM) filter for high-density salt and pepper noise reduction. First the noises are classified based on the their values which must be the extreme gray scale of the image. Then, at the filtering stage, the noise-free pixels remain unchanged and noisy pixels are restored using a modified non-local mean filter. To calculate the weights of the filter, only noise-free pixels are considered.

Some most recent and powerful techniques deal with all types of impulses, including fast and efficient decision-based algorithm (EDBA) for removal of high density of impulsive noises [Srinivasan 2007], an improved decision based algorithm for impulse removal (IDBA) [Nair 2008], a switching median filter with boundary discriminative noise detection for extremely corrupted images (BDND) [Ng 2006], directional switching median filter using boundary discriminative noise detection by elimination (BDNDE) [Nasimudeen 2010] and a fuzzy-based decision algorithm for high-density impulse noise removal (FBDA) [Nair 2012]. EDBA takes less processing time and shows good edge preserving quality but smooth transition among pixels are lost. IDBA has been proposed to overcome this problem, which also suffers when the noise density is high. To overcome this problem for higher noise level BDND has been devised which also suffers from high miss and false detection rate at random valued impulse noise with higher time complexity. BDNDE has been proposed to overcome the drawbacks of BDND which eliminates the problems but cannot preserve the fine edge details. FBDA was proposed to eliminate all the above problems of the existing filters but still the noise detection rule of FBDA is not strong and subsequently it suffers from sensitivity and specificity issues.

Although there exists number of median and mean based filters discussed so far, a number of soft computing based filters have also been proposed in the literature such as *FIRE* operators [Russo 1999b], Recursive neuro-fuzzy filter [Russo 1999a], topological median fil-

ter [Senel 2002], Fuzzy based noise reduction filter [Van De Ville 2003], Neuro-fuzzy operator based filter [Yuksel 2003], Adaptive fuzzy switching filter, Resilient neural network based filter [Besdok 2004], Adaptive neuro-fuzzy inference interpolant filter [Besdok 2005], Neuro-fuzzy edge detection based noise filter [Yuksel 2005], Fuzzy logic based robust filter [Choi 1997], etc.

Various filters discussed so far have the disadvantage because of removing thin lines, removing thin edges, blurring image fine textures in the image during the noise removal operation. Moreover, the performance of the filters depend on the user parameters, tuning parameters such as noise density, pixel weighting value, impulse detection threshold, etc. These values are heuristically determined and externally supplied by the user for each individual image or noise density as there is no analytical method to get the optimal value of such parameters for a particular denoising experiment.

Although all the spatial filters perform well on digital images but they have some constraints regarding resolution degradation. These filters operate by smoothing over a fixed window and it produces artifacts around the object and sometimes causes over smoothing thus causing blurring of image. Wavelet transform is most suitable for performance because of its features like sparsity, multiresolution and multiscale natures. All frequency filters can also be implemented in the spatial domain and, if there exist a simple kernel for the desired filter effect, it is computationally less expensive to perform the filtering in the spatial domain. Frequency filtering is more appropriate if no straightforward kernel can be found in the spatial domain, and may also be more efficient.

2.2 Denoising in Frequency Domain

Signals do not exist without noises. There are many cases in which the noise corrupts the signals in a significant manner, and it must be removed from the data in order to proceed with further data analysis. The process of noise removal is termed as signal denoising. Example of a noisy signal and its denoised version can be seen in Fig. 2.4. It is seen that the noise adds high-frequency components to the original signal which is smooth. This is a characteristic effect of noise. Although the term signal denoising is usual, it is devoted to the recovery of a digital signal that has been contaminated by additive white Gaussian noises (AWGN) and Poisson noises. There is a wide range of fields where denoising is very important. Examples are medical image or signal analysis, data mining, radio astronomy and there are many more. For example, noise removal in medical signals requires specific care, since denoising which involves smoothing of the noisy signal (e.g., using low-pass filter) may cause to loose of fine details. There are many approaches in the literature for the task of denoising in frequency domain.

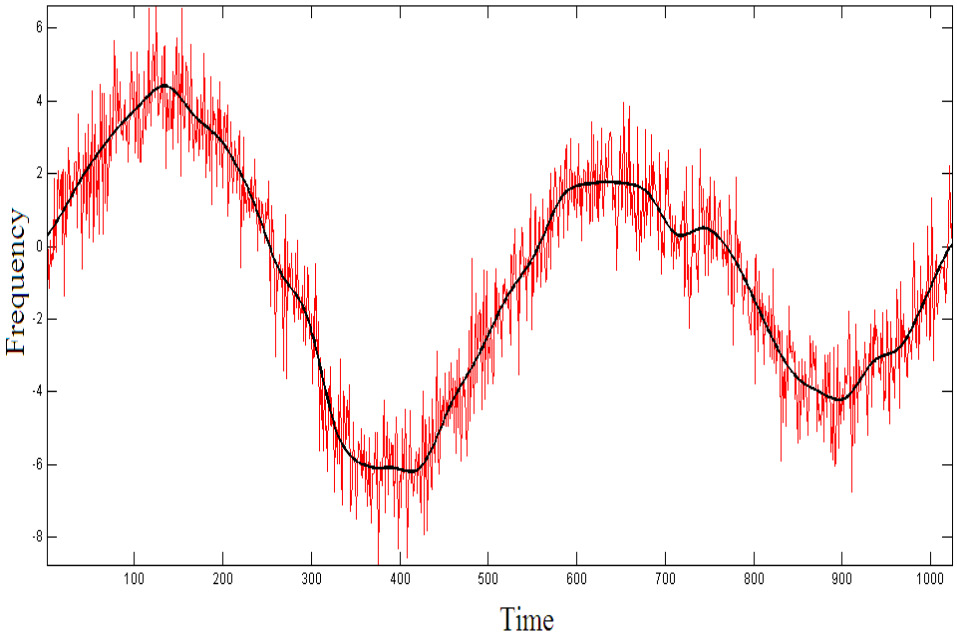


Figure 2.4: A noisy sine wave and its denoised version (solid line)

The *Fourier transform* is analyzed to find the frequency components of the signal. However, the Fourier transformation is taken over the whole time axis, it is not possible to say in what instant a particular frequency rises. Short time Fourier transformation (STFT) operates on sliding window to find the spectrogram, to give the information on both time and frequency. But the length of window minimizes the resolution in frequency domain. *Wavelet transform* seems to be a solution to this problem. Wavelet transforms are tiny wavelets with limited time duration. In the thesis, the wavelet based denoising techniques are surveyed and a novel algorithm based on wavelet transformation has been proposed.

The first literature that relates to the wavelet transform is Haar wavelet, proposed by the Alfrd Haar in 1909. Before 1985, Haar was only the orthogonal wavelet transform. Many researchers felt the situation that no orthogonal wavelet except Haar wavelet exists. As more and more scholars joined in this field, the first international conference was held in France in 1987. In 1988, Stephane Mallat and Meyer devised the theory of *multiresolution* analysis. At the same year, Daubechies constructed the compact support orthogonal wavelet. In 1989, Mallat proposed the fast wavelet transformation.

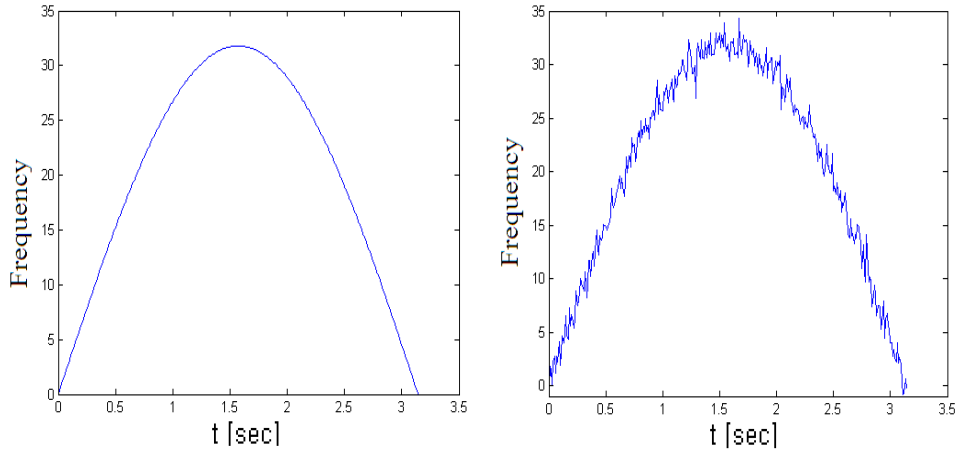
The development of wavelet transforms over last decades revolutionized modern signal

and image processing, especially in the field of signal denoising. During the 1990s, the field was dominated by wavelet shrinkage and wavelet thresholding methods by Donoho *et. al* [Donoho 1995c, Donoho 1994, Donoho 1995a, Donoho 1995b, Donoho 1995d, Mallat 2009, Tsai 2002, Mallat 1989].

The denoising problem in wavelet domain is formulated as follows:

$$\hat{f} = \min_{f} ||\hat{f} - f||_2 \tag{2.8}$$

Suppose, there are n noisy samples of a function f given in (2.9)



(a) (b)

Figure 2.5: Signal f and its noisy version y

$$y_i = f(t_i) + \sigma\varepsilon_i, = 1...n \tag{2.9}$$

where ε_i are *iid* N(0,1) and the noise level σ may be known or unknown. The visual representation of this is given in Figure 2.5. The goal is to recover the underlying function f from the noisy data, $y = (y_1, ..., y_n)$, with small error, when the criterion is the mean squared error. In other words, it is needed to find a function $\hat{f}$ which satisfies the equation given in (2.8), where $\hat{f} = \hat{f}(\mathrm{y})$. Wavelet thresholding is commonly used for denoising because of its ease of implementation and simplicity.

Numbers of techniques based on transform domain have been reported in the literature.

Zhong Hua et. al [Hua 1997] proposed a wavelet based denoising algorithm to remove the background noise in medical images. The forward wavelet transformation is applied to the original image. Donoho's soft thresholding method is applied to the wavelet coefficients to suppress the noise. Subsequently the Wavelet coefficients are reconstructed through inverse wavelet transformation. An adaptive least squared error clustering is applied to the inverse image to get a bi-level image. Then the contour and mask of the image are obtained by the chain coding of the object. The mask is applied to the original image where the pixels outside the mask are assumed as background noises and those are removed eventually. The algorithm is applied to various medical images and an efficient comparative analysis is presented. Several medical images such as heart and bone images are tested by the algorithm and hence algorithm is effectively validated.

Crouse et al. [Crouse 1998] developed a framework for statistical signal processing based on wavelet domain hidden markkov models (HMM). This model described the non Gaussian statistics of each wavelet coefficients and relate the statistical dependencies between the coefficients effectively.

Kingsbury et al. [Kingsbury 1999] proposed the 2D dual tree complex wavelet which satisfies these requirements effectively. But this method is less efficient for motion estimation since the motion information is related to the coefficient phase, which is nonlinear function of estimation.

P.E Tikkanen [Tikkanen 1999] proposed a nonlinear wavelet and wavelet packet denoising algorithm for electrocardiogram (ECG) signal. It is a non-linear denoising approach through soft and hard thresholding methods simultaneously, in which thresholds are chosen using four different techniques. Coiflet wavelet and wavelet packet functions are used to build up the dyadic wavelet and optimized wavelet packet decompositions. The localization of the denoising error value within the cardiac cycle was given by visual inspection of the denoised signal and extracting the error measures during the QRS-complex. This method involves the quantitative comparison of several denoising approaches by means of optimized error measures and inspection of the denoised ECG signal and the error signal. Results given in the paper shows that the wavelet based denoising algorithm is more efficient than wavelet packet approaches in all cases, but with the heuristic SURE threshold function as hard thresholding for white noises performs better in terms of Mean Square Error. Here the denoising errors are concentrated within the QRS-area when the wavelet approach is employed but soft and hard thresholding show balances in denoising the high-frequency parts of an ECG.

Cai and Silverman [Cai 2001] proposed a wavelet thresholding method which takes the

immediate neighboring coefficients into account to form the threshold. This method obtained better denoising results than the conventional pixel by pixel method, as reported in the paper.

Xiao-Ping [Zhang 2001] proposed a type of thresholding using neural network for adaptive noise reduction. An improved version of soft and hard thresholding functions are devised to serve as the activation function of the Thresholding Neural Network. Proposed thresholding functions are infinitely differentiable. By using this, gradient-based learning algorithm becomes more effective. The optimal solution of the thresholding neural network in a mean square error (MSE) sense is discussed. The performances of both soft and hard thresholding are analyzed in the algorithm in linear noise reduction method. Gradient-based adaptive learning algorithms are used and presented to obtain the optimal solutions for noise reduction. The algorithms adopted supervised and unsupervised batch learning as well as supervised and unsupervised stochastic learning. The thresholding neural network with the stochastic learning algorithms is used as a nonlinear adaptive filter. The results given for the thresholding neural network show optimal solutions of thresholding methods in terms of MSE and outperform other noise reduction methods.

Vladimir *et al.* [Cherkassky 2001] proposed an comparative study of several wavelet-denoising methods applied to the problem of removing myopotential noise from the observed noisy ECG signal. It compares the denoising accuracy of several wavelet thresholding methods such as VISU, SURE and soft thresholding algorithms and a new thresholding approach based on Vapnik-Chervonenkis (VC) [Vapnik 1995] learning theory. It indicated that the VC-based wavelet approach is superior to the standard thresholding methods as it achieves higher denoising accuracy in terms of both Mean Square Error measure and visual fidelity.

Lin *et al.* [Lin 2003] proposed an image compression and denoising method through a non-separable wavelet approximation. In this algorithm an application of two-dimensional non separable wavelet approximation to is applied the task of image denoising and image compression. The algorithm consists of two-dimensional non separable scaling function interpolation to perform image compression and image denoising. The comparison results with the separable counterparts it shows that the algorithm have some improvements and advantages of two-dimensional non separable scaling function interpolation. Extensive experimental results are studied in this algorithm based on the test images *Lena* and *Goldhill* and the restoration results show better performance in terms of PSNR (dB) compared to most of the state-of-the-art methods.

The idea of neighboring wavelet thresholding has been developed by Chen and Bui [Chen 2003] in the multiwavelet scheme. From the results it is seen that neighbor mul-

tiwavelet denoising outperforms the neighbor single wavelet denoising [T.D.Bui 1998] for some test images and real time signals.

Kim *et al.* [Kim 2004] proposed a wavelet based algorithm to detect shared congestion. This technique proposes a signal processing method i.e., wavelet denoising, to separate queuing delay caused by network congestion from various delay variations. It shows that when detecting shared congestion of paths with a common endpoint, this technique provides faster convergence and higher accuracy while using fewer packets than previous techniques, and that it also accurately determines when there is no shared congestion. Congestion control helps the endpoints fairly and efficiently to share network resources. Better utilization of network resources through congestion management algorithms is done by determining two different flows between shared congested links. This can be used to implement cooperative congestion control or improve the overlay topology of a P2P system. The earlier techniques to detect shared congestion were either assuming a common source or destination node, drop-tail queuing, or a single point of congestion. It proposes a technique which is applicable to any pair of paths on the Internet, without such limitations.

Cheng *et al.* [Cheng 2004] proposed a image denoising method using wavelet and support vector regression. This algorithm described a method for the suppression of noise in image by fusing the wavelet denoising technique with support vector regression (SVR). Based on the least squares support vector machine (LS-SVM), this denoising operator is used in the wavelet domain. The simulated noisy images are used to evaluate the denoising performance of the devised algorithm along with the other wavelet-based denoising algorithms. The results show that the denoising method outperforms standard wavelet denoising techniques on the basis of the signal-to-noise ratio (PSNR (dB)) and the edge information is preserved in most the cases. It also obtains better performance than the median filter.

Neelamani proposed *ForWardD* [Neelamani 2004] method which obtains better denoising results than the traditional denoising methods based on Wiener filtering. It can perform in boxcar smoothing while the images are with low noise density. This method does not perform well for other noises.

Donoho proposed *WaveD* [Donoho 2004] which deals with the natural representation of the convolution operator in the Fourier domain as well as the typical characterization of Besov classes in the wavelet domain. This method has the disadvantage of not performing well on tuning parameters independently.

Ghazel *et al.* [Ghazel 2005] proposed a localized wavelet thresholding strategy which

adopts context-based thresholding operators. The earlier wavelet thresholding methods, such as VisuShrink [Donoho 1994], LevelShrink [Chang 2000b] and BayesShrink [H.Chipman 1997, S.Chang 2000], apply the conventional hard and soft thresholding operators and only differ in the selection of the threshold. The earlier soft and hard thresholding operators are point operators in the sense that only the value of the processed wavelet coefficient is taken into consideration before thresholding. In this work, it is shown that the performance of some of the standard wavelet thresholding methods can be improved by applying a localized, context-based, thresholding strategy instead of the conventional thresholding operators. Four different test images such as *Lena, Boat, Peppers and Sanfrancisco* images are corrupted by AWGN with varying noise density $\sigma_w = 10$ to 40 with the bayes shrinking method. Results obtained with the improved Bayes shrinking methods are satisfactory.

Chen et al. [Chen 2005] proposed a noise suppression method which considered a square neighborhood window to customize the wavelet filter threshold for image denoising. They remove impulses from the images effectively.

Toprak *et al.* [Toprak 2006] proposed a suppression technique of impulse noises in the medical images based on fuzzy adaptive median filter. It is a rule based fuzzy filter for removal of highly impulse noises in medical images known as rule based fuzzy adaptive median (RBFAM) Filter. The RBFAM filter is a version of adaptive median filter (AMF) and it is presented to reduce noises of images corrupted with additive impulse noise. The filter works in three stages. Two of those steps are fuzzy rule based and last one ios based on standard median and adaptive median filters. The filter can preserve image details better then AMF while suppressing additive salt and pepper or impulsive noises. In the algorithm the preference is based on the bell shaped membership function rather than triangular membership function obtain better results. the results given indicates that the proposed filter is further promising with improved fuzzy based rules to reduce more noises and to remove salt and pepper noises effective than AMF filter.

Qi-ming [Ma 2006] presented a Genetic Algorithm (GA) based on wavelet transform threshold shrinkage (WTS) method. The translation invariant thresholding and shrinking (TIS) is proposed here with the method of noise reduction, as the parameters used in WTS and TIS, such as wavelet function, decomposition levels, hard or soft threshold and threshold can be selected automatically. This method finds better solution by comparing two noise reduction methods in terms of their denoising performances, computation time, etc. The effectiveness of these methods introduced in this algorithm to validate the results of the simulated and real signals.

Bacchelli *et al.* [Bacchelli 2006] proposed an image denoising technique using the principal component analysis in the wavelet domain. It is proposed for removal of the Gaussian noise from digital images, based on the combination of the wavelet packet transform and the principal component analysis. The aim of denoising method is to retain the energy of the signal while discarding the energy of the noise, so to construct powerful tailored filters by applying the Karhunen Loeve transformation in the wavelet packet domain, so it obtains a compaction of the signal energy into a few principal components, while the noise is spread over all the transformed coefficients. To act with a suitable shrinkage function on these new coefficients to remove the noise without blurring the edges and the image fine textures.

Schulte *et al.* [Schulte 2006] proposed a fuzzy based wavelet shrinking for image denoising method. The usage of a fuzzy set theory in the domain of image enhancement using wavelet thresholding is widely used now-a-days. It devises a simple but efficient new fuzzy wavelet shrinkage method where a fuzzy variant of a probabilistic shrinkage method for reducing adaptive Gaussian noise from digital grey scale images is used. The results show that the proposed method can be efficiently and rapidly remove additive Gaussian noise from digital grey scale images. The quantitative and qualitative restoration results obtained using the proposed method outperforms the state-of-the-art fuzzy and non-wavelet denoising methods and the method is also is compared with some of the recent but more complex wavelets methods.

Feng *et al.* [Feng 2006] proposed the application of wavelet neural network (WNN) with orthonormal bases in digital image denoising. By considering the image fuzzy generated in the process of image denoising in spatial field the image denoising method based on wavelet neural network with orthonormal bases is proposed in this algorithm. The denoising principle and construction method of orthonormal wavelet network is also discussed in this algorithm. In is algorithm the neural network based optimization has been done through wavelet multiresolution analysis theory and the adaptive image denoising method is accomplished. Multiresolution learning algorithm followed by wavelet neural network based denoising method on multiresolution learning is performed. The simulation experiments based on median filtering, adaptive median filtering and wavelet neural network with orthonormal bases are used separately in the denoising performance for contaminated images. The restoration results show that the image denoising method based on orthonormal wavelet neural network improves to the traditional denoising methods on the basis of the image quality and it also decreases the image ambiguity to some extent.

Eslami *et al.* [Eslami 2006] proposed a translation invariant contourlet transformation and its application to image denoising. Most sub-sampled filter banks suffer with the feature

of translation invariant, which is used in denoising applications. The methods have been developed to convert a general multichannel, multidimensional filter bank to a corresponding translation-invariant (TI) framework. In particular, a generalized algorithm has been proposed which is an extension of the algorithm introduced for 1-D wavelet transforms. The TI contourlet transform (TICT) has been constructed. To reduce the high redundancy and complexity of the TICT, semi translation-invariant contourlet transform (STICT) has also been used. An adaptive bivariate shrinkage scheme to the STICT has been used to achieve an efficient image denoising approach. Experimental results demonstrate the benefits and potential of the denoising technique.

Li *et al.* [Li 2007] proposed subband and spatially adaptive thresholding method for removing noises as well as to preserve the image fine textures. One of the imaging applications such as the ultrasound or Synthetic Aperture Radar (SAR) images, the texture features are used to illustrate the geological formation or biological tissue at that spatial location. Therefore to remove noises from these texture images, it is essential that the local texture details characterizing the geological formation or tissue type are not lost. For preprocessing of these images, the operator usually should have prior knowledge of the type of textures to preserve it in the image. This prior knowledge is exploited to implement a spatially-adaptive and subband-adaptive wavelet thresholding method that denoising the texture images while preserving the characteristic features in the textures as well. The proposed algorithm involves three stages: firstly the texture characterization, secondly the texture region identification system training and lastly the texture region identification and denoising. In the first stage, the texture features to be preserved are characterized by the subband energies of the wavelet decomposition details at each level. Then the energies of the characteristic subband are used as inputs to train the adaptive neural-fuzzy inference system (ANFIS) classifier for texture region identification. Finally, the texture regions are identified by the ANFIS and the subband-adaptive Bayes Shrink threshold is adjusted locally to obtain the proposed spatially-adaptive and subband-adaptive threshold.

Zhang *et al.* [Zhang 2007] proposed an image denoising method based on wavelet with a support vector machine. The least square support vector machine (LS-SVM) is a modified version of SVM, which uses the equality constraints to replace the original convex quadratic programming problem. Consequently, the global minimizer is much easier to obtain in LS-SVM by solving the set of linear equation. LS-SVM has shown to exhibit excellent classification performance in many applications. In this algorithm a wavelet-based image denoising using LS-SVM is proposed. The wavelet coefficients are divided into two classes (noisy coefficients and noise free coefficients) by LS-SVM training model. Finally, all noisy wavelet coefficients are relatively well denoised by soft-thresholding method. The experimental re-

sults demonstrate that this method obtains better performances in terms of both subjective and objective evaluations than those of the state-of-the-art denoising techniques.

Bruni *et al.* [Bruni 2008] proposed an image denoising method using the similarities in the time-scale plane. It presents a denoising method that recognizes similarities in the image through the time scale behavior of wavelet coefficients. The details of the wavelet are represented as linear combination of atoms whose center of mass traces trajectories in the time scale plane. The distances among the atoms, the ratios of their amplitudes and the differences of scales are the parameters which are used for defining similarities in the image.

Chen *et al.* [Chen 2008] proposed a denoising of a natural image corrupted by the Gaussian white noise is a classical problem in image processing. Here, a new image denoising method is proposed by using three scales of dual-tree complex wavelet coefficients. The dual tree wavelet transformation is suitable for its shift invariance and directional selectivity features, which are important in image denoising. Results show that the technique is very competitive in nature when compared with other denoising methods.

Deng *et al.* [Deng 2009] proposed an impulse noise removal technique which is based on wavelet based neural network, is applied to restore digital images corrupted by impulse noise. Firstly the wavelet neural network is used to detect the noisy pixels and to distinguish it from noise free pixels. Secondly the noisy pixels are classified again by corresponding thresholding and assigning the coefficients. Lastly the median filter is combined with the coefficients for the restoration of the noisy pixels. This method can used to effectively eliminate the impulse noise while preserving image fine textures.

Krommweh *et al.* [Krommweh 2009] proposed directional haar wavelet frames on triangles for denoising the Gaussian noise in digital images. The earlier wavelet based filters are not very effective in dealing with images which contain oriented discontinuities (edges) in the images. To achieve a more efficient representation the algorithm used basis elements with much higher directional sensitivity. In the recent past years, several operators like *curvelets* and *shearlets* have been devised to provide essentially optimal approximation properties for images that are piecewise smooth and have discontinuities along C^2 curves. While *curvelets* and *shearlets* have compact support in frequency domain, it constructs the directional wavelet frames generated by functions with compact support in time domain. This *Haar wavelet transformation* or constructions can be used as a special composite dilation wavelets, which are being based on the generalized *multiresolution analysis (MRA)* associated with a dilation matrix and a finite collection of *'shear'* matrices. Based on the *multiresolution analysis (MRA)* structure it provides an efficient filter bank algorithm for

image denoising.

Alajlan [Alajlan 2010] proposed a novel recursive algorithm for denoising the digital images while preserving the image fine details. To remove the impulse noises from the digital images without removing the fine details of the image is a major problem in the field of image processing. There are a large number of filter set in the literature those deal with the problem of image denoising. Most of the state-of-the-art filters which removes the impulse noise is usually performed in two main stages: *detection* and *estimation*. It shows that the algorithm can take any impulse noise removal approach where the noise detection and estimation processes grow separately and then enhance its detail preservation capability in the digital images. Unlike the traditional recursive implementations which perform sequentially in row by row scanning, the denoising algorithm maximizes the contribution of noise-free neighbors in detecting and correcting the noisy pixels.

Binh *et al.* [Binh 2010] proposed a computationally efficient approach for denoising the images which are corrupted by additive Gaussian noise. Here, a relatively recent category of stochastic global optimization technique i.e., particle swarm optimization (PSO) technique has been devised for learning the parameters of adaptive thresholding function required for optimization of image denoising algorithm. The PSO-based denoising approach can speed up the optimization algorithm and can improve the performance the denoising performance in comparison with wavelet transform based thresholding neural network (WT-TNN) algorithm. The results obtained by the algorithm shows good noise removal and edge preservation performances in bior 6.8 wavelet filter when compared to db8 wavelet filter.

Aravind *et al.* [Aravind 2010] proposed a wavelet based image denoising method with optimal thresholding technique. Among the VisuShrink and SureShrink for removing the Gaussian noises from the digital images, the VisuShrink is a simple and effective methods but it has the disadvantage that it smoothes the images because of the big universal threshold value. The disadvantage of the SureShrink is that it chooses subband independent threshold value for each subband. The NeighShrink is better thresholding technique than the VisuShrink and SureShrink in terms of the thresholding. The NeighShrink is improved in this algorithm by determining the optimal threshold and neighboring window size for each of the subband using Stein's Unbiased Risk Estimator. The results show that the algorithm performs better than the unmodified version of NeighShrink, VisuShrink and SureShrink.

Binh *et al.* [Binh 2010] proposed an adaptive wavelet technique for medical image denoising. The filter is based on the second derivative of Gaussian filters methods, and in particular generalizes an adaptive method proposed earlier by R.Duits [Duits 2003]. It devises

a method which is based on the steerable complex wavelet construction. This is an improvement over the operators such as ForWaRD [Neelamani 2004] and WaveD [Donoho 2004].,etc.

Wang *et al.* [Wang 2010] proposed a wavelet-based image denoising using least squares support vector machine. The least squares support vector machine (LS-SVM) is a modified version of SVM, which uses the equality constraints to replace the original convex quadratic programming problem. LS-SVM has shown to exhibit excellent classification performance in many applications. The wavelet-based image denoising using LS-SVM is proposed here. Firstly, the noisy image is decomposed into different subbands of frequency and orientation responses using the wavelet transform. Secondly, feature vector for the noisy pixels are formed by the spatial regularity in the transformed domain, and the model is obtained by training process. Then the wavelet coefficients are divided into two classes (noisy coefficients and noise-free ones) by LS-SVM training model. Finally, all noisy wavelet coefficients are relatively well denoised by soft-thresholding method.

Yang *et al.* [Yang 2010] proposed an adaptive non-local means filter for denoising live cell images and to improve the particle detection strategy. The fluorescence imaging of dynamical processes in live cells often results in a low signal-to-noise ratio (PSNR (dB)). The algorithm presents a feature-preserving non-local method to denoise such fluorescence images to improve feature recovery and particle detection performance. Most commonly used non-local filter is not the optimal algorithm for noisy biological images which contains small features of interest because the noise image prevents accurate determination of the correct coefficients for averaging, leading to over-smoothing and other artifacts. The adaptive method of solving this problem by constructing a particle feature probability image, which is based on Haar-like feature extraction. The particle probability image is then used to improve the estimation of the correct coefficients for averaging. It justifies that the filter achieves better peak signal-to-noise ratio (PSNR (dB)) while denoising the images and it achieves better capability in identifying weak particles when applied to synthetic data. This approach of denoising to live-cell images results in enhanced detection performance of end binding protein 1 foci on dynamically extending microtubules in photo-sensitive Drosophila tissues. It shows that the feature-preserving non-local filter can reduce the threshold of imaging conditions required to obtain meaningful data.

Chou *et al.* [Chou 2011] proposed a thresholding technique based on wavelet denoising method for Hydrological Data Modeling. A novel framework has been developed for considering wavelet denoising in linear perturbation models (LPMs) and a simple linear model (SLMs). Rainfall and runoff time series data are decomposed using discrete wavelet transformation to acquire approximate and detailed rainfall and runoff signals, respectively,

at multiresolution levels. At each subband of the wavelet decomposition level, the threshold quantifications are performed by setting the values of a detailed signal below a certain threshold to zero. The denoised rainfall and runoff time series data are obtained from the approximation at the final resolution level and processed detailed signals using threshold quantification at all resolution levels of rainfall and runoff, respectively, by wavelet reconstruction. The data are applied to the SLM and then regarded as the smooth seasonal mean employed in the LPM. The noise, i.e., original time series value minus denoised time series value, is employed as the perturbation term in the LPM. Moreover, a linear relationship between input and output noise is assumed. The denoised runoff and estimated noise of runoff are summed to estimate overall runoff in the LPM. The method is verified by the accuracy of the method, daily rainfall runoff data are analyzed for an upstream area of the Kee-Lung River. The quantitative results obtained demonstrate that wavelet denoising enhances rain fall runoff modeling precision for the LPM.

Hilton *et al.* [de Oliveira Mota 2011] proposed a partial discharge signal denoising method with spatially adaptive wavelet thresholding and support vector machines. In this algorithm an improved method for denoising partial discharge (PD) signals is presented. The method is based on the wavelet transformation (WT) and support vector machines (SVM). It is different from other WT-based denoising strategies in the sense that it exploits the high spatial correlations presented by PD wavelet decompositions as a way to identify and to select the relevant coefficients. PD spatial correlations are characterized by Wavelet modulus maxima propagation along decomposition levels (scales), which is a strong indication of the time-of-occurrence. The task of denoising is performed by identification and separation of partial discharge signal related to maximum lines by an SVM pattern classifier.

Dong *et al.* [Guo 2011] proposed the application of *Contourlet* transformation for image denoising. *Coutourlet* is a real representation of two dimensional image. The transformation is consisted with two layer filters: *Laplacian pyramid transform* is used to realize multiscale or multiresolution decomposition and multi direction decomposition can be implemented by directional filter bank. The *Contourlet transformation* can realize multiresolution and multi-direction decomposition flexibly and grasp essential character of the images. Compared to the *curvelet* transformation, the contourlet transformation presented by digital conversion of discrete domain first, and then extended to continuous domain and analyzed on its properties.

Ming *et al.* [He 2011] proposed an anisotropic wavelet based image denoising using multi scale products. The denoising algorithm is based on anisotropic wavelet transform (AWT) using multiscale products. Before denoising, the noisy images are first decomposed by the anisotropic wavelet. The multi decomposition product threshold is then applied to the mul-

tiscale products of the AWT coefficients instead of directly to the AWT coefficients. Since the multiplication operation amplifies the significant features and dilute noise, the method reduces speckle effectively while preserving edge structures.

Xiao *et al.* [Xiao 2011] proposed a comparative study on thresholding methods in wavelet based image denoising. The study explored properties of several representative thresholding techniques in wavelets denoising, such as VisuShrink, SureShrink, BayesShrink and Feature Adaptive Shrinkage.

Junfeng *et al.* [Li 2011a] proposed an adaptive image denoising algorithm based on correlativity. The first method distinguishes the noise spots according to the relations between the grey coefficients of incidence matrix which are calculated between the noise image and the mean image and the threshold value. The second one calculates the values of the noise spot according to the correlativity coefficients which is calculated between the noisy image and the mean image on the 3×3 template and realizes the adaptive weighted image denoising. The results given in this article show that the method reduces the image fuzziness, preserves the integrity of edge and detail information and it has the good denoising effect as well.

CAO *et al.* [Ke 2011] proposed a novel image denoising algorithm which is based on *Crank Nicholson Semi-implicit Difference(C-N)* Scheme [Crank 1947]. In this algorithm, the semi-implicit *C-N* scheme is applied to discrete the *ROF* (Rudin-Osher-Fatemi) [Rudin 1992] and the *G-S* iterative method to solve the linear equations using matrix form.

Ming *et al.* [Li 2011b] proposed an algorithm for Remote Sensing Image (RIS) denoising which is based on the combination of the improved BiShrink [Sendur 2002] and dual tree complex wavelet transfer (DTCWT) [Kingsbury 1998b, Kingsbury 2001, Kingsbury 1998a]. The technique considered the strong correlation between wavelet coefficients of the actual image when the bivariate model is only a statistical model for the inter scale dependency of wavelet coefficient with parent coefficient without taking into account the correlation of adjacent coefficient. Based on the shift-invariance and better directionality of the dual tree complex wavelet transformation (DTCWT) and incorporating neighboring wavelet coefficients with BiShrink, a BiShrink threshold and DTCWT remote sensing image denoising method is presented.

During the last decade, several novel methods have been devised for removing the additive white Gaussian noise from medical images. Two filters in the domain of wavelet have been devised for image denoising are Nguyen et al. [Binh 2010] and Silva et al. [Silva 2012].

Other methods combine impulse removal filters with local adaptive filtering in the transform domain to remove noise. Cross-validation techniques are used for thresholding parameter selection [N 1998]. Bayesian procedure [H.Chipman 1997] combine inference from data with prior information to estimate thresholding parameters.

Anitha [Anitha 2012] proposed switching median filter for denoising the two dimensional patterned textile images. Wallpaper fabric, a part of patterned fabric, is a new decorative material covered with fabric or other natural fiber materials. Wallpaper fabrics are different from other indoor decorative materials and are designed for daily use with easy maintenance instructions. A common type of noise that affects patterned fabric images is the impulse noise. Here a modified switching median filter is devised. The algorithm contains two stages, namely, noise detection stage and noise filtering stage. During noise detection a modified adaptive switching median filter with three new thresholds are used to remove the impulse noise. During filtering stage, an adaptive switching median filter is used to determine the size of the sliding window. To detect noisy pixels which are close to its neighbors that are normally missed with the traditional threshold of switching median filter, a modified threshold that considers the ranking order of the pixels in the sliding window is used. The enhanced stages make improvement the noise removal process while maintaining image details. The denoising model is evaluated using three performance metrics, namely, PSNR (dB), FoM and execution time. The experimental results shows that the model is efficient in removing impulse noise with low and high noise density while preserving important image and edge details.

V. Naga *et al.* [Raj 2012] proposed a denoising method of medical images using Dual Tree Complex Wavelet Transformation [Kingsbury 1998b, Kingsbury 2001, Kingsbury 1998a]. The technique proposed the method which uses dual tree complex wavelet transform to decompose the image and shrinkage operation to eliminate the noise from the noisy image. In the shrinkage step semi-soft and Stein thresholding operators are used along with traditional hard and soft thresholding operators and verified the suitability of dual tree complex wavelet transform for the denoising of medical images. The results obtained ensured that denoised image using DTCWT (Dual Tree Complex Wavelet Transform) have a better balance between smoothness and accuracy than the DWT and less redundant than UDWT (Undecimated Wavelet Transform) [Fodor 2001].

Chui *et al.* [Chui 2012] proposed an image denoising method with adaptive Bayes threshold in non sub sampled contourlet domain. In this algorithm, an image denoising method has been given which is based on a nonlinear thresholding function with adaptive Bayes thresholding in non sub-sampled contourlet transform (NSCT). In overcoming the shortcomings of

the same threshold, the noise deviation of the different sub-band are estimated based on the coefficients of different directional and level sub-bands in NSCT domain, and the thresholds of every sub-band is estimated by Bayesian threshold estimation method. After choosing the thresholds, a nonlinear thresholding function is chosen to overcome the shortcomings of the soft and the hard thresholding function. The simulation results show that the proposed method here can remove Gaussian white noise more effectively and get a higher PSNR value and keep image texture and detail information more clearly which also has a better visual effect.

Chen *et al.* [Chen 2012] proposed an image denoising algorithm using the three scales of wavelet coefficients. The denoising of an image corrupted by the Gaussian white noise is a classical problem in image processing. An image denoising method is proposed by using three scales of dual-tree complex wavelet coefficients here. The dual-tree complex wavelet transform is known to all for its approximate shift invariance and better directional selectivity, which are very important in image denoising. Results given in the paper show that the method is competitive when compared with other existing denosing methods.

Biswas *et al.* [Biswas 2012] proposed a wavelet based soft-thresholding technique based image denoising method. In this algorithm, they propose a method of noise removal from an image corrupted with Gaussian noise using soft-thresholding. This method is simple and adaptive since the estimation of thresholding parameters depends on the data of wavelet coefficients. According to the experimental results, this method has higher Peak Signal-to-Noise Ratio (PSNR (dB)) and visual effects than the VisuShrink [Donoho 1994]. Lena, Cameraman and Barbara test images are used for validation of the algorithm.

Silva *et al.* [Silva 2012] proposed an adaptive edge-preserving image denoising algorithm using wavelet transformation. This algorithm proposed an image denoising method based on wavelet transforms to preserve edges. An adaptive thresholding scheme based on edge strength is used to effectively reduce noise while preserving important features of the original image. The simulation and experimental results obtained and those are compared to other approaches, proved that the method is suitable for different classes of images contaminated by Gaussian noise.

Wang *et al.* [Wang 2012b] proposed a nonlinear filtering method which is based on three dimensional wavelet transform for magnetic resonance images (MRI) denoising. In this study, the algorithm is based on a three dimensional extension of the wavelet transform (WT) which is bilateral filtering for Rician noise removal. Due to delineating capability of wavelet, three dimensional WT is proposed to provide effective representation of the noisy

coefficients. Bilateral filtering of the approximation coefficients in a modified neighborhood can improve the denoising efficiency and can effectively preserve the relevant edge and image features. The detailed subbands of the wavelet decomposition are processed with an enhanced NeighShrink thresholding algorithm. The validation operation is performed on both simulated data and clinical data. Using the peak signal-to-noise ratio (PSNR (dB)) to quantify the amount of noise of the MR images, it has achieved an average PSNR (dB) enhancement of 1 to 32 times with simulated data.

Mehmet *et al.* [Mehmet 2012] proposed a denoising algorithm for weak electrocardiogram (ECG) signal by using the wavelet based fuzzy thresholding. This algorithm presented a weak ECG signal denoising method based on fuzzy thresholding and wavelet packet analysis. Firstly, the weak ECG signal is decomposed into various sub bands/levels by wavelet packet transformation. Subsequently the threshold value is determined using the fuzzy s-function. The reconstruction of the ECG signal from the retained coefficients is achieved by using inverse wavelet packet transformation. Various experiments were performed to show the effectiveness of the proposed method and compared the results with the traditional wavelet packet based soft and hard thresholding methods for weak signal denoising.

Amudha *et al.* [Amudha 2012] surveyed digital image denoising performances in the wavelet domain. The objective of this research work is to introduce digital image restoration to the reader, who has just begun their research work in this field, and to provide a review and analysis for the reader who may already be familiar in image restoration. The perspective on the topic offered here is to let the readers and researchers to know various methods of digital image restoration.

Liu *et al.* [Liu 2012] proposed an algorithm and implementation for heart sound denoising because of the most important signal is heart signal. Although the process of getting heart sound signal can be interfered with many factors outside. The heart sound is a weak electric signal and even weak external noise may corrupt it to the misjudgment of pathological and physiological information about the signal, thus it causes the misjudgment of disease diagnosis. In this algorithm, a more systematic research and analysis which is involved in heart sound denoising based on matlab has been made. Multi-level wavelet transform is adopted to deal with each level of detail coefficients thresholding to eliminate noise and interference of every frequency band. Meanwhile, the design of a notch filter is chosen to filter out frequency interference and the body's electrical and mechanical interference. The study of heart sound denoising based on matlab firstly use the powerful image processing function of matlab to transform heart sound signals with noise into the wavelet domain through wavelet transform and decomposition of these signals into multilevel signal having different

frequencies. For the detail coefficient, soft thresholding is made using wavelet transform thresholding to eliminate noise, so that a signal denoising is significantly improved. The reconstructed signals are gained with stepwise coefficient reconstruction for the processed detail coefficient. *Coif wavelet* decomposition of eight layers is chosen after analysis and the corresponding comparisons. Lastly, 50HZ power frequency and 35 Hz mechanical and electrical interference signals are eliminated using a notch filter.

Wang *et al.* [Wang 2012a] proposed a model on medical image denoising and enhancement based on *GSM* model (Gaussian scale mixture) for wavelet coefficients. For the complexity and application diversity of medical *CT* image, this algorithm presents a medical CT Image enhancing algorithm based on Gaussian Scale Mixture model for wavelet coefficient in the study of wavelet multi-resolution analysis. The noisy image is firstly denoised in auto adapted Wiener filter. Secondly, through the qualitative analysis and classification of wavelet coefficients for the signal and noise, the wavelet's approximate distribution and statistical characteristics are described, by combining GSM (Gaussian scale mixture) model for wavelet coefficient in this algorithm.

Tyagi *et al.* [Tyagi 2013] proposed an algorithm for removing of partial discharge (PD) signals using genetic algorithm (GA) which is based on wavelet shrinking methods. The measurements are seriously affected by the interference signals resulting from different sources. In order to enhance the sensitivity of a PD monitoring system, many digital signal processing methods have been proposed for removal of noises. But to obtain the optimal denoising Genetic Algorithm is used.

Jaiswal *et al.* [Jaiswal 2014] proposed an image denoising and quality measurement techniques by using filtering and wavelet based techniques respectively. In this algorithm the researchers work with denoising of salt pepper and Gaussian noise.

A major amount of filters based on discrete wavelet transformation discussed so far have the disadvantage of finding the optimal values of the threshold to be used on the wavelet coefficients in the transformed image before the inverse transformation is made during the image denoising method. The most difficult problem in wavelet based denoising approach is to find out the exact value of the threshold. A small threshold can keep the maximum portion of coefficients related to the noisy signal and that results a signal which is still noisy. When the threshold is a large value then it will shrink maximum portion of coefficients. That results blurring of the signal which causes losing of important textures in the image.

Good estimation of the wavelet parameters such as wavelet function, decomposition level

and threshold value is important to the success of wavelet based denoising. These parameters are usually estimated in empirical or semi-empirical manner during the denoising the corrupted images. This procedure does not guarantee to achieve the optimal restoration results. To overcome this problem, some algorithm can be developed in this area. Any randomized search algorithm can perform this operation. A method may be added to the conventional wavelet based denoising algorithms as a wavelet denoising technique which is based on BayesShrink threshold technique and which uses the optimization algorithm of the wavelet decomposition level as well. The sub band adaptive thresholding technique using the BayesShrink may perform excellently to obtain good restoration results. But a good tuning to threshold value and estimation of the optimal value of the decomposition level outperform the BayesShrink thresholding technique.

2.3 Background and Motivation

From available approaches for restoration of digital image it is seen that the problem can be divided into two parts: image restoration/ enhancement in the spatial or frequency domain. In spatial filtering techniques, the image pixels directly accessed and altered to restore the noisy pixels. In the frequency domain, the image pixels are transformed into spectral domain using Fourier transformation or by Wavelet transformation etc., and then those transformed coefficients are altered for restoration followed by reverse transformation to regenerate the original noise free images.

In the early development of image filtering, the linear filters were widely used. However in presence of noise the performance of linear filter is very poor because it blurs the image pixels to a great extent. So the image fine textures are hardly preserved using this technique.

For poor performance of the linear filters, some new types of filters have been introduced which is called *non linear filter*. Order statistic filters are the most popular set of non linear filters such as median filter or stack filter. These filters are widely used in the application areas of digital image processing.

Adaptive filters have also taken the advantage of characteristics of non linear filters. This type of filter changes its behavior depending upon the characteristics of the digital image. The non-adaptive linear filters are generally designed to filter the digital images having a particular type of noise in the images. Adaptive filters generally outperform the non-adaptive non-linear filters.

Several types of methods have been proposed by different authors in the recent past. Most of the methods are based on the median filters, where the rank order information of the pixels of in test window is used. The standard median filter [Brownrigg 1984] removes the noisy pixels by replacing test pixel with the median value of the pixels of the filtering window. This technique performs well but also removes the thin lines, dots, distorts edges

and blurs image fine textures even at very low noise density. A huge improvement has been done recently over the standard median filters. In early stage of filtering techniques, the method of filtering was adopted as non-switching techniques where the filtering has been made in single step, where all pixels were filtered irrespective of noise contamination. In case of switching filters, the noisy pixels are only altered. The pixels of the images are first classified as noisy or noise free then the noisy pixels are restored using its neighboring pixels.

The techniques available in the literature of digital image processing mainly targeted to perform on some particular type of images and specific type of noise models.

Algorithms are lacking from selection of proper window size and noise detection technique. Available methods do not include all the pixels of the test window to classify the center pixel. Some of them simply compare the center pixel of the window with the maximum and minimum intensity values within the test window for noise detection rule. Moreover those algorithms use only single threshold value for comparison to define the noisy pixel, as a result the decision is very rigid.

So the motivation of the thesis is to develop algorithms which can deal with all types of noise models. Another important issue is to deal with the images corrupted with impulsive noises with high density. A major problem of image denoising is to suppress the random valued noise in the digital images. The random valued impulse noise may take any value in the range of minimum intensity value to maximum intensity value of the image pixels. Detection of those pixels is very much difficult for the digital image.

Unless and until the noise density is estimated a proper window size cannot be suggested because it is seen that when the noise density is low then there is no need to select a large filtering window. A low density noisy image can perfectly be denoised by taking a small window size. But if the noise density is more than 60-70% then it is needed to select a larger window size for suppressing the noises. Hence a proper window selection is a major point prior to filter a image. The problem can be solved if the noise density of the input digital image is estimated and followed a corresponding filtering window size is empirically suggested.

Another approach of the thesis is to filter the images through image clustering techniques. An image can be perfectly clustered if the image is noise free. In other words an image can be made noise free progressively if the image is clustered iteratively. So noise reduction and cluster generation of digital images can be done simultaneously.

Two dimensional Wavelet can also used for image denoising particularly for medical images with Gaussian or Poisson noises in the images. Wavelet can decompose the image matrix into four different frequency sub bands. Noises are actually the higher frequency elements of the image matrix. Using this characteristic of the Wavelet, the noisy frequencies can be analyzed and removed from the images.

2.4 Contributions

The objective of this thesis is to design efficient techniques for digital image denoising. The digital images may be corrupted with any type of noises and those may have any level of noise density. There are various types of impulses based on their characteristics and how they combined with the original images. Two important types of noises are mixed with the digital images. Salt-and-Pepper (SPN) noise and Random valued noise (RVN) are used for such purpose. The SPN and RVN are having both equal and unequal probabilities of densities. There exist a lot of operators which can deal with the Salt-and-Pepper noise but there exist a limited number of algorithms to remove the Random Valued noise from the digital images. The Random Valued noises are the type of noise where any pixel of the image can take any value in the range of minimum intensity value and maximum intensity value of the gray level intensity range. It is also very much difficult to suppress the noises from digital images which are corrupted with high density of random valued noise.

The main objective and contribution of the thesis is to deal with random valued noises with high density. A set of algorithms have been proposed to deal with high random valued noises. Some techniques outperforms in terms of noise suppression ability and computational cost of those algorithms. Characteristics of the techniques are that they can also suppress salt and pepper noise in addition to random valued impulse noise with variable probabilities of noise density. Some algorithms are based on standard deviation operation where as some are based on median operation of the neighboring pixels of the test pixel. A variety of median filters have been developed to deal for variable noise density. Most of the median filters developed are weighted. Some pixels within the test window are given extra weight when restorations are performed using the median operation.

Medical images often corrupted with Gaussian Noise and Poisson Noise due to the malfunctioning from of machineries in the clinical laboratories. To deal with these noises in medical images novel optimization based noise suppression algorithm have been proposed. This process can suppress the noises from the medical images as well as it optimizes the performance of the algorithm. This algorithm obtained better restoration results compared to the state of the art algorithms in the literature.

From study of algorithms surveyed in this thesis work it is observed that a hypothetical window is selected prior to detect and restore the test pixel. Some algorithms uses 3×3 test window where as some of them used 21×21 test window. There is no such technique which can suggest a filtering window size depending upon the noise density of the image. The filtering window size must be compatible with the noise density of the image otherwise selection of pixels from the neighborhood would not be perfect and hence median operation will return a false value while restoring the noisy pixels. It may happen that a noisy pixel is replaced by a noisier pixel by the median operation.

In the proposed methodology of the study, the noise density of any image is determined at the beginning. On estimation of noise density, an empirical filtering window size is suggested and the algorithm proposed is deployed.

Optimization is another objective to obtain the optimal results. To achieve this goal the proposed techniques consists of two steps. First in the detection step some user parameters are used to classify the test pixel. Genetic Algorithm and Particle Swarm optimization based techniques are proposed to search the user parameters in a wide range to obtain the optimal restoration results. The results obtained using the optimization algorithms outperforms the most of the state-of-the-art techniques in the literature.

Clustering algorithms have also been proposed in this thesis for noise detection and filtering operations of the digital images. K-means based clustering algorithm has been utilized to estimate the noise density in the digital images. K-means algorithm has been validated by Silhouette validity index. PSO based clustering algorithm has been proposed to classify the digital image by unsupervised way. This algorithm is capable of suppressing salt-and pepper noise as well as random-valued-noise with good restoration results. It uses the general property of a noise-free image to detect the noisy pixels which leads to a conclusion that the proposed technique may be treated as generalized noise removal approach for the digital images. The technique is also capable to deal with high density of impulses in the digital images. PSO-based image clustering has been proposed over the genetic algorithm based clustering because of convergence of PSO is very fast and it's another advantage is ease of implementation. Three evaluation criteria are used to form the fitness function in the PSO-based clustering. The fitness function in PSO-based clustering has been proposed in such a way to give more compact clusters and larger inter-cluster distance compared to K-means. A novel Euclidean distance function has been proposed in the thesis where the spatial as well as positional distances are computed to compare the similarity or dissimilarity of any two pixels during grouping of pixels.

Several types of statistical, mathematical and soft computing based tools are utilized to make the proposed operator efficient. Statistical tools such as standard deviation, variance, median, mean, etc are used for noise detection. Soft computing tools such as Fuzzy membership function, Genetic algorithm, K-means, Particle Swarm optimization algorithms are utilized for noise suppression, estimation and optimization purposes for the denoising methods.

Fuzzy refinement process has been applied on each pixel depending on the noise level of the center pixel which is actually dependent on the weighted sum calculated in detection stage. For noise detection purpose, instead of a single threshold, two threshold values are used in the proposed fuzzy membership (MF) function to partition the noise level and accordingly a filtering method is applied to restore the corrupted pixel. Two different weighted median filters are devised and a particular one is applied selectively to the noisy pixel based on the

characteristics of the neighboring pixels within the test window.

Denoising in medical images corrupted by Gaussian noise is important in diverse image processing techniques and computer vision problems. Image denoising using discrete wavelet transform is well established process in image processing because it can separate the noisy signal from the image signal. This algorithm proposed a denoising method of medical images through thresholding and optimized using a recent stochastic random search algorithm i.e., Genetic Algorithm (GA). Firstly the noisy image is partitioned into fixed sized blocks and then transformed into wavelet domain. Some important parameters present in the two dimensional discrete wavelet transform such as the decomposition level and the threshold value have been searched and optimized in a wide range in the proposed technique. The Bayesian shrinkage method has been selected for thresholding because of its sub band dependence property.

To validate the proposed algorithms extensive analysis of the results has been done. A number of bench mark images have been trained to obtain noisy images and then proposed algorithms are applied on these images for the purpose of testing of the algorithms. These are *Lena*, *Barbara*, *Cameraman*, *Circuit*, *Boat*, *Ultrasound*, *Liftingbody.*, respectively.

The result set is also made uniform that is, a homogeneous set of results have been made for each algorithm to make the comparisons easily understandable. Bench mark color images are also applied to the proposed algorithm to validate the proposed algorithms in the thesis work.

The performances of the proposed filters have been evaluated quantitatively and qualitatively through simulation results and analysis. Eight bit gray scale bench mark images are used to evaluate the performance of the proposed algorithm. The results obtained are compared with the existing filter against the noise models. Various graphical representations of the results using different attributes are also given to substantiate the performance of the proposed technique. Bench mark images are trained using low (10%) to high (100%) noise density using all noise models.

Various performance metrics are used to find the algorithmic performances of the proposed algorithms in the thesis. Different statistical measures are used in this thesis to validate the proposed algorithms by comparing it with most of the state of the art methods in the literature. Some measures of image quality such as *Peak Signal to Noise Ratio (PSNR (dB))*, *Mean Square Error (MSE)*, *Image Enhancement Factor (IEF)*, *Structure Similarity Index Measure (SSIM)*, etc are used for the purpose mentioned above.

2.5 Layout of the Thesis

The thesis has been organized in five sections. The details of the sections are as follows.

Section I presents an introduction and the state of the art on topics related to this work.

The first chapter of the thesis gives a brief idea about the problem domain and definitions in the related area. In the second chapter, different areas of image denoising are studied in the literature. Different statistical approaches towards noise suppression have been reviewed. Mean filter or averaging filter, order statistics filter, adaptive filter, hybrid filter, histogram based filter, median filter, variance based filter., etc are surveyed in this chapter. Various soft computing tools based filters such as fuzzy membership filter, genetic algorithm based filter, particle swarm optimization filter, and various clustering algorithm tools based filters are also studied. Filtering operators in the frequency domain have also been reviewed in this chapter. Extensive study on wavelet based denoising is also presented.

Section II focused on the algorithmic and data structure development of the thesis. The first chapter of this section focuses to present some simple algorithms towards digital image denoising based on statistical tools. The methods to optimize the above algorithms by searching the user parameters in a wide range using genetic algorithm and particle swarm optimization techniques are presented in the fourth chapter of the thesis. In the fifth chapter, the noise density estimation and noise suppression techniques are developed through K-means algorithm. A novel noise suppression technique based on pixel clustering of digital image has been performed and reported in the sixth chapter. A novel fuzzy filter has been proposed in seventh chapter, where the membership function has been proposed and used to detect the noise level on the test pixel. Based on the noise level a particular filtering method is selected in the algorithm. A wavelet based medical image denoising technique has been reported at chapter eight of the thesis.

Section III presents an extensive experimental results and discussions. In this section, both the filtering domains such as spatial and frequency are evaluated for digital and medical image denoising. In this section, various comparisons have been made on salt-and-pepper noise for all the proposed algorithms with the state of the art operators. Comparisons have also been made in this section where the salt and pepper noise has unequal probability distribution over the noisy image. The images having random valued noises with equal and unequal probability distribution have also been applied to all the proposed algorithms and the existing algorithms. Some comparisons on the medical images have also been performed in the frequency domain based denoising. Several performance metrics are used in this section to validate the proposed algorithms quantitatively and qualitatively.

Conclusions, applications, limitations and future perspectives of the proposed techniques have been reported in section IV.

Section V contains bibliographical references cited in the thesis.

Part II

Proposed Denoising Techniques

Chapter 3

Noise Suppression

Contents

3.1 Variance based Filter 64

3.1.1 Directional Weighted Minimum Deviation Filter 65

3.1.2 All Neighbor Directional Weighted Pixels Filter 70

3.2 Weighted Median Filter 75

3.2.1 Variable Mask Median Filter 76

3.2.2 Edge Preserving Restoration Filter 80

3.2.3 Sub-Image Restoration Filter 83

3.3 Discussions 87

In this chapter, techniques for image denoising have been proposed. The working principal of these techniques are based on statistical and mathematical operators. The Salt-and-Pepper and Random-Valued-noises are used to train the digital images with a variable noise density which varies from 10% to 100%. The techniques proposed are switching filters. The noisy pixels in each test window are detected using noise detection operator and then only the noisy pixels in each window are restored by a noise filtering operator. The test window size is $2m + 1 \times 2m + 1$, where m is a positive integer. The test window slides to next pixel to consider the center pixel of the window in row major order. Each time the center pixel of the window is classified by the proposed noise detector and then the noisy pixels are filtered by using the neighboring pixels of the test pixel. The noise free pixels are left unchanged.

The objective of this chapter is to develop techniques for noise suppression especially for salt-and-pepper and random-valued-noises in digital images while preserving the image fine details and fine textures which includes the thin edges, dots etc. The process works in two steps. First a strong impulse detector is proposed which uses simple arithmetic and statistical operators for detection of impulses in the images. An odd size window is selected and the center pixel of the window is classified as noisy or noise free iteratively. The direction of the test window is hypothetically considered as an edge in the image. The pixels on the edge

must be made closely clustered so that they preserve the edges during the noise restoration operation.

The standard median based filters are not so strong to preserve the image fine details while suppressing the noises. In this context, variance based filter performs better to some extent. Blurring is a major effect that results from standard median filter. The demerits of the median based filters can be eliminated if the proposed filter is used. The computational overhead of any algorithm is also a major factor to concern about. By minimizing the iteration in the algorithm the computational complexity is minimized. Another point regarding the median filter is about the degree of noise density. Most of the median based filters suffer from blurring the image fine textures even at low noise density. When a high density of noises are integrated into the digital images, the performance of median based filter is very poor. This problem of median based filters is also resolved in the proposed techniques.

Simple variance and median based filters are developed. In the first phase of the technique it finds the noisy pixels by simply calculating the absolute differences between the center pixel and all other pixels in each directional pixels in the test window. After detecting the noisy pixels the filtering operator is applied to those pixels which are detected as noisy. In the second phase, the operator restores the noisy pixel by minimizing the variance based equation to align the center pixel most closely to all other pixel in the direction of the test window.

The chapter is organized as follows. The techniques are described in details sequentially in this chapter. The foremost section contains the techniques based on simple variance based operations. The first scheme termed as *Directional Weighted Minimum Deviation (DWMD)* based Filter is described in section 3.1.1. Another algorithm based on *All Neighbor Directional Weighted Pixels (ANDWP)* based filter is given in section 3.1.2. Three median based filters are also included in this chapter. The first method based on *Variable Mask Median (VMM)* filter is described in section 3.2.1. An *Edge Preserving Restoration (EPR)* algorithm is given in section 3.2.2. Section 3.2.3 discusses *Image Denoising based on Sub-Image Restoration (SIR)*. The chapter is concluded with a brief discussions of the algorithms given in section 3.3.

3.1 Variance based Filter

In this section, two directional weighted minimum variance based filters are proposed. The algorithms have two stages. The first stage detects the noisy pixels and in the second stage, only the noisy pixels are restored. Random valued noise models are incorporated to corrupt the original images. The noise models have already described in section 1.2. This algorithms assumes that a noise free flat image consist of locally and smoothly varying areas separated

by edges. The pixels in a noise free regions change their intensity values compared to each other gradually or smoothly but in case of noisy pixels they change their intensity values drastically. It is also assumed that two pixels whose spatial distance is small, their gray level values will also be close to each other. Usually the pixels located in the neighborhood of a test pixel are correlated to each other and they possess almost similar characteristics. The schemes are simple but efficient and work alternatively in two phases: detection of noisy pixels followed by deviation based filtering of the noisy pixels.

A 5×5 window of the noisy image is considered as a mask and verified whether the test pixel (center pixel) is noisy or not. Four directions along the center pixel are considered as main contributor for the technique, because four directions can represent the four edges in the test window. The absolute differences of pixel intensity values over an edge are considered very low as there is no transition between them. But in the four directions or edges, the gray values may have large differences as they are on different edges. In the four directions, four absolute differences are calculated between the test pixel and neighborhood pixels. The minimum of these is selected and measured against a threshold for detecting impulse noise. The filtering methodology is based on minimum variance of the pixels in a single direction. As in the case when the density of random valued noise on the digital image is high then the median filter may blur the image. Because in this case, most of the pixels are corrupted. So it cannot simply restore the center pixel using the median value within the window. This may increase the complexity too. So rather than median, this algorithm chooses to utilize the variance of the pixels in a direction to restore the noisy pixel within the test window. This will decrease the computational complexity and restores the noisy pixels using the pixels which have maximum local similarity within the test window. The variance describes how tightly all the values are clustered around the mean in the set of pixels. The pixels aligned in this direction are the closest to each other. Therefore the center pixel should also be close to them in order to keep the edges (even thin lines) intact in the direction. Here a new method has been introduced to make the center pixel (test pixel) as close as possible to the remaining four pixels aligned in a direction.

The directional weighted minimum deviation filter (DWMD) is presented in section 3.1.1 and the all neighbor directional weighted pixels filter (ANDWP) is given in section 3.1.2.

3.1.1 Directional Weighted Minimum Deviation Filter

The methodology used for impulse detection is based on simple arithmetic operation like the absolute differences between the center pixel and all other pixels in the test window. Then the minimum of the differences are selected and compared with a threshold (user parameter) to classify the center pixel. In this scheme the edges aligned with four directions shown in

Fig. 3.1. In the 5 × 5 test window, four directions are considered as they are four edges of the image in the test window. The objective is to preserve the edges of the image during the course of denoising.

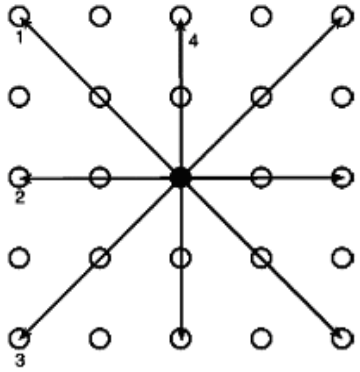


Figure 3.1: Four directions for impulse detection and restoration

3.1.1.1 Impulse Detector

The DWMD impulse detection algorithm is given in Algorithm 1. The algorithm runs for a total of n times in step 2 where n is an user parameter. In steps 3 and 4, the for loops iterate from 0 to maximum rows and maximum columns of the image matrix. Steps 5 and 6 iterate a total of twenty five times to contain the twenty five pixels of the test window. In step 7, four sets are formed each with five pixels. Let D_k (k=1 to 4) denotes a set of coordinates aligned with the k^{th} direction centered at (0, 0), i.e.,

- D_1= (-2,-2), (-1,-1), (0, 0), (1, 1), (2, 2)
- D_2= (0,-2), (0,-1), (0, 0), (0, 1), (0, 2)
- D_3= (2,-2), (1,-1), (0, 0), (-1, 1), (-2, 2)
- D_4= (-2,0), (-1, 0), (0, 0), (1, 0), (2, 0)

Let $D_k^0 = D_k/(0,0)$, $\forall$ k =1 to 4. In step 8, the absolute differences are calculated and a weight is multiplied to it. The $d_{i,j}^{(k)}$ is the sum of all absolute differences of gray level values between $y_{i+s,j+t}$ and $y_{i,j}$ with (s,t)$\in D_k^0$ (k= 1 to 4), given in (3.1). In steps 9-13, it multiplies the absolute differences between the two closest pixels with a large value ω_m, before it calculates the sum.

$$d_{i,j}^{(k)} = (\sum_{(s,t)\in D_k^0} \omega_{s,t}|y_{i+s,j+t} - y_{i,j}|, 1 \leq k \leq 4) \tag{3.1}$$

Algorithm 1 DWMD noise detector

Input: Image Matrix g_{xy}, Threshold (T), Maximum Number of Iterations (n), Decreasing Rate of Threshold (R)

Output: Window's Center Pixel Classified

begin
for g=0 to n-1 **do**
  for x=0 to MaxRow-1 **do**
    for y=0 to MaxCol-1 **do**
      for i=-2 to 2 **do**
        for j=-2 to 2 **do**
          Initialize D_k to contain seven pixel in each direction with i and j, k= 1 to 4, Say $y_{i,j}$ be the center pixel
          $d_{i,j}^{(k)} = \omega_{s,t}|y_{i+s,j+t} - y_{i,j}|$
          if D $\in \omega^3$ (3.3) **then**
            $\omega_{s,t}$=2
          else
            $\omega_{s,t}$=1
          end if
          $r_{i,j} = \min\{d_{i,j}^{(k)}\}$
          if $r_{i,j} > T$ **then**
            $y_{i,j}$ is Noisy
          else
            $y_{i,j}$ is Noise-Free
          end if
        end for
      end for
    end for
  end for
  T= T * R
end for
end

where $\omega_{s,t}$ is given in (3.2).

$$\omega_{s,t} = \begin{cases} 2 & : (s,t) \in \omega^3 \\ 1 & : otherwise \end{cases} \tag{3.2}$$

where ω^3 is given in (3.3).

$$\omega^3 = \{(s,t) : -1 \leq (s,t) \leq 1\} \tag{3.3}$$

$d_{i,j}^{(k)}$ is known as direction index. Each direction index is sensitive to the edge aligned with a given direction. In step 14, the minimum of the four direction indices is used for impulse detection, which is given in (3.4).

$$r_{i,j} = \min\{d_{i,j}^{(k)} : 1 \leq k \leq 4\} \tag{3.4}$$

There may be three cases for the value of $r_{i,j}$ done in step 14 as described below:

1. When the current pixel is in a noise free flat region, $r_{i,j}$ is small because of the four small direction indices.

2. When the current pixel is an edge pixel then $r_{i,j}$ is small because at least one of the direction index is small.

3. When the current pixel is a pixel containing impulse noise, $r_{i,j}$ is large because the four large direction indices.

In definition of $r_{i,j}$, it makes full use of the information aligned with four directions. So from the above analysis it can find a noisy pixel by employing a threshold T. It can identify the impulse from the noise free pixels, no matter which are in a flat region, edge or thin line. Then it can define the impulse detector by making complete use of the information aligned with four directions, as in (3.5). The value of the *Threshold (T)* is supplied by the user during run time. T is varied in a wide range to get the best possible restoration results. In step 29, the threshold is decreased in each iteration by a rate of decreasing threshold (R).

$$y_{i,j} = \begin{cases} Noisy\ Pixel & : r_{i,j} > T \\ Noise\ Free\ Pixel & : r_{i,j} \leq T \end{cases} \tag{3.5}$$

In this technique, the value of n is varies from 4 to 6, so the algorithm iterates a maximum of 6 times, i.e., constant time. In steps 3 and 4, the loops iterate a total of (rows * columns) times. Steps 5-6 iterate for a constant time. The internal steps are performed for a constant time also. As a whole the algorithm iterates for c_1*(rows*cols)*c_2 which is equivalent to O (rows*cols) times, which is quadratic polynomial, where c_1 and c_2 are constants.

3.1.1.2 Impulse Filter

The DWMD filtering technique is given in Algorithm 2. Step 2 iterates to maximum of n times where n varies from 4 to 6 times. In steps 3 and 4, the for loops iterate to cover the maximum rows and maximum columns. Steps 5 and 6 iterate twenty five times to contain twenty pixels of the test window. In step 7, the variances are calculated and the minimum is searched in step 8. The pixels are collected to form the set S in the direction where the

Algorithm 2 DWMD filter

Input: Image Matrix g_{xy}, Maximum Number of Iterations (n)
Output: Window's Center Pixel Restored

begin
for g=0 to n-1 **do**
  for x=0 to MaxRow-1 **do**
    for y=0 to MaxCol-1 **do**
      for i=-2 to 2 **do**
        for j=-2 to 2 **do**
          Calculate $\sigma_{i,j}^{(k)}$ using the i and j values
          $l_{i,j} = \min(\sigma_{i,j}^{(k)})$
          S = pixels at $l_{i,j}$ direction
          Let S={a,b,c,d,e}
          Replace c by x
          while (true)
          $\sigma_1 = \sigma(x = \mu)$
          $\sigma_1 = \sigma(x = \mu + 5)$
          $\sigma_1 = \sigma(x = \mu - 5)$
          if σ_1 is minimum **then**
            exit while
          else
            if $\sigma_2 < \sigma_3$ **then**
              Assign x=x + 5 and continue
            else
              Assign x=x - 5 and continue
            end if
          end if
          end while
        end for
      end for
    end for
    Find an t: abs$(x - t)$ is minimum, $\forall t \in S$.
    Replace center pixel $y_{i,j}$ by t
  end for
end for
end

standard deviation is minimum in steps 9 and 10. The center pixel of the set is replaced by an unknown variable x in step 11. The mean (μ) of the set is calculated. Three variances are calculated in steps 13-15. If for x=mean, the variance is minimum then the loop terminates. Otherwise, between μ_2 and μ_3, the minimum is selected and loop continues in steps 20 or 22. In step 28, the center pixel $y_{i,j}$ is replaced by x. The detail description of the algorithm is given below.

In the technique a novel scheme has been introduced based on minimum variance of the four directional pixels. The variance $\sigma_{i,j}^{(k)}$ of gray values of all $y_{i+s,j+t}$ with (s,t) $\in D_k^0$ (k= 1 to 4) are calculated. Then it finds the minimum of $\sigma_{i,j}^{(k)}$, where k= 1 to 4, given in (3.6).

$$l_{i,j} = argmin_k\{\sigma_{i,j}^{(k)} : k = 1 \; to \; 4\} \tag{3.6}$$

where the operator *argmin* is used to find the minimizer of a function. Since the variance describes how tightly all the values are clustered around the mean in the set of pixels, $l_{i,j}$ shows that the four pixels aligned with this direction are closest to each other. Therefore the center pixel should also be close to them in order to keep the edges (even thin lines) intact in the $l_{i,j}$ direction. Here a novel method has been introduced to make the center pixel (test pixel) as close as possible to the rest of the four pixels aligned in the direction. Let the set in the direction is S= {a, b, c, d, e}. It first replaces the center pixel of the set by x, to generate the set a, b, x, d, e. Now calculate three variances using x= mean, mean+5 and mean-5. Here 5 is used as the set length is 5. The x value is also selected for which the variance of the set is minimum among the three variances. It is clear that variance is minimum for x=mean, then it just replaces by the mean value and terminates, else it proceeds by assigning x by either (x+5) or (x-5). The technique iterates to minimize the variance by either decreasing x by 5 or by increasing x by 5. The iteration terminates when it becomes steady.

$$Let \; t_{i,j} = x. \tag{3.7}$$

. The algorithm finds the element $t_{i,j}$ in the initial set S ($l_{i,j}$ directional) and just replace $y_{i,j}$ by $t_{i,j}$. If not found, it assigns $t_{i,j}$ to its nearest element in the set S, and then $y_{i,j}$ is replaced by $t_{i,j}$.

In this technique, the value of n is varies from 4 to 6, so the algorithm iterates a maximum of 6 times, i.e., constant time. In steps 3 and 4, the loops iterate a total of (rows * columns) times. Steps 5-6 also iterate for a constant time. The internal steps run for a constant time as well. The total iteration is c_1*(rows*cols)*c_2 which is equivalent to O (rows * cols), which is quadratic polynomial, where c_1 and c_2 are constants.

3.1.2 All Neighbor Directional Weighted Pixels Filter

In this algorithm, an *All Neighbor Directional Weighted Pixels (ANDWP)* based filter has been proposed for removal of highly random valued noises (RVN) in the digital images. The proposed approach works in two phases. The first phase detects the contaminated pixels by making the differences between the test pixel and its all neighbor pixels aligned in four directions in the 5×5 window. The 5×5 test window is given in Fig. 3.2 which includes all the neighboring pixels in the test window. The second phase filters only the noisy pixels based on minimum variance of the four directional pixels. The advancement is done in terms of considering all the neighboring pixels in the 5×5 window in both the stages of detection and filtering, secondly the variance based filter is optimized by the principal of finding the minima by using an algebraic equation. An equation is constructed on the basis of the variance and then the equation is used to find the minimum value of the unknown variable x for which the variance of the set of seven points in any direction is minimum. All twenty five pixels are considered for noise detection and filtering. As a result, maximum possible information of the neighborhood is considered in order to improve the quality of the reconstructed image.

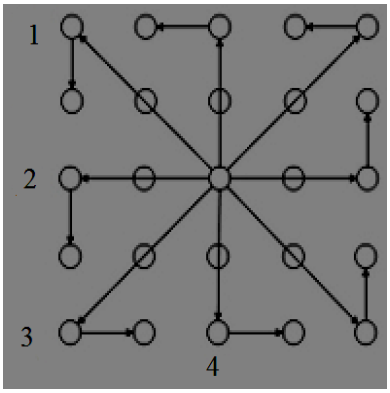


Figure 3.2: Four directions include all neighboring pixels

3.1.2.1 Impulse Detector

The ANDWP impulse detection technique is given in Algorithm 3. Step 2 iterates for a maximum of 6 times as n varies in range [4, 6]. In the steps 3 and 4, the for loops grow to cover the maximum rows and maximum columns of the noisy image. Steps 5-7 iterate to form the set D, which contain all the twenty five pixels of the window. That is described as follows:

Algorithm 3 ANDWP noise detector

Input: Image Matrix g_{xy}, Threshold (T), Maximum Number of Iterations (n), Decreasing Rate of Threshold (R)

Output: Window's Center Pixel Classified

begin
for g=0 to n-1 **do**
  for x=0 to MaxRow-1 **do**
    for y=0 to MaxCol-1 **do**
      for i=-2 to 2 **do**
        for j=-2 to 2 **do**
          Initialize D_k to contain seven pixel in each direction with i and j, k= 1 to 4, Say $y_{i,j}$ be the center pixel
          $d_{i,j}^{(k)} = \omega_{s,t}|y_{i+s,j+t} - y_{i,j}|$
          if D $\in \omega^3$ (3.10) **then**
            $\omega_{s,t}$=2
          else
            if D $\in \omega^2$ (3.11) **then**
              $\omega_{s,t}$=1
            else
              $\omega_{s,t}$=0.5
            end if
          end if
          $r_{i,j} = \min(d_{i,j})$
          if $r_{i,j} > T$ **then**
            $y_{i,j}$ is Noisy
          else
            $y_{i,j}$ is Noise-Free
          end if
        end for
      end for
    end for
  end for
  T=T*R
end for
end

Let D_k (k=1 to 4) denotes a set of seven pixels aligned in the k^{th} direction centered at (0, 0), i.e.,

. D_1 = {(-1,-2), (-2,-2), (-1,-1), (0, 0), (1, 1), (2, 2),(1, 2)}.

. D_2 = {(1,-2), (0,-2), (0,-1), (0, 0), (0,1), (0, 2), (-1, 2)}.

. D_3 = {(2,-1), (2,-2), (1,-1),(0, 0), (-1, 1), (-2, 2), (-2, 1)}.

. D_4 = {(-2,-1), (-2, 0), (-1, 0), (0, 0), (1, 0), (2, 0), (2,1)}.

Then let $D_k^0 = D_k/(0,0)$, $\forall$ k =1 to 4. In step 8, in each direction of the 5 $\times$ 5 window centered at (i,j), it defines $d_{i,j}^{(k)}$ as the sum of all absolute differences of gray values between $y_{i+s,j+t}$ and $y_{i,j}$ with (s,t)$\in D_k^0$ (k= 1 to 4), given in (3.8). In steps 9-17, in each direction, it multiplies the absolute differences between two closest pixels from the center pixel with a large value ω_m, multiplies the absolute differences between the center pixel and the corner pixels by ω_n and also multiplies the absolute differences between two end pixels from the center pixel with a small value ω_o, before it calculates the sum. It assigns ω_m =2, ω_n=1 and ω_o=0.5. The value of $\omega_{s,t}$ is given in (3.14).

$$d_{i,j}^{(k)} = \sum_{(s,t)\in D_k^0} \omega_{s,t}|y_{i+s,j+t} - y_{i,j}|, 1 \leq k \leq 4 \tag{3.8}$$

$$\omega_{s,t} = \begin{cases} \omega_m & : (s,t) \in \omega^3 \\ \omega_o & : (s,t) \in \omega^2 \\ \omega_n & : \text{otherwise} \end{cases} \tag{3.9}$$

where

$$\omega^3 = \{(s,t) : -1 \leq s,t \leq 1\}, and \tag{3.10}$$

where

$$\omega^2 = \{(s,t) : (s,t) = \pm\{(-2,-1),(-1,-2),(1,-2),(2,-1)\}\}. \tag{3.11}$$

In step 18, the minimum of $d_{i,j}^{(k)}$ is calculated. The minimum of these four direction indices are used for impulse detection, which is denoted as

$$r_{i,j} = min\{d_{i,j}^{(k)} : 1 \leq k \leq 4\} \tag{3.12}$$

In steps 19-22, the noisy pixel is identified efficiently from the window by employing an user parameter T. T is decreased in step 28 by the rate of R.

In this algorithm, the value of n is varies from 4 to 6, so the algorithm iterates for a maximum of 6 times, i.e., constant time. In steps 3 and 4, the loops iterate a total of (rows * columns) times. Steps 5-6 iterate for a constant time. The internal steps are performed for a constant time as well. As a whole the algorithm iterates for c_1*(rows*cols)*c_2 which is equivalent to O (rows*cols) times, which is quadratic polynomial, c_1 and c_2 are constants.

3.1.2.2 Impulse Filter

Algorithm 4 ANDWP filter

Input: Image Matrix g_{xy}, Maximum Number of Iterations (n)
Output: Window's Center Pixel Restored

1: **begin**
2: **for** g=0 to n-1 **do**
3: **for** x=0 to MaxRow-1 **do**
4: **for** y=0 to MaxCol-1 **do**
5: **for** i=-2 to 2 **do**
6: **for** j=-2 to 2 **do**
7: Calculate $\sigma_{i,j}^{(k)}$ using the i and j values
8: $l_{i,j} = \min(\sigma_{i,j}^{(k)})$
9: S = pixels at $l_{i,j}$ direction
10: Let S={a,b,c,d,e,f,g}
11: Replace d by x
12: Compute mean (μ)
13: Form f(x)=(a-μ)^2) + (b-μ)^2) + (c-μ)^2) + (x-μ)^2) + (e-μ)^2) + (f-μ)^2) + (g-μ)^2)
14: Compute f'(x) and f"(x)
15: Solve f'(x)=0
16: Find an t: abs($x - t$) is minimum, $\forall t \in S$.
17: **end for**
18: **end for**
19: **end for**
20: Replace center pixel $y_{i,j}$ by t
21: **end for**
22: **end for**
23: **end**

The ANDWP filtering technique is given in Algorithm 4. A novel scheme has been introduced based on minimum variance of all the four directional pixels. Starting with a noisy image and a threshold value (T), in row major order it scans each 5×5 window in the noisy image. If any pixel is detected as noisy, the filtering scheme restores it to a pixel which is most suitable in the 5×5 window. Step 2 iterates for n times. In steps 3 and 4, the for loops iterate for the whole image matrix. The lines 5 and 6 iterate to contain all the pixels of the test window of size 5×5. In line 7, the standard deviations are computed. The minimum standard deviation is selected in step 8. In lines 9-10, the seven pixels are selected to form the set where standard deviation is minimum. The center pixel of the set is replaced by x

in line 11. The mean is computed in step 12. In step 13, an equation is formed on the basis of standard deviation. The first order and second order derivatives are calculated in step 14. In step 15, f'(x) is solved to find the value of the x. Here f"(x) always returns a positive value because S contains positive values in [0, 255]. In line 17, a t is searched so that the absolute difference between t and x is minimum where t $\in$ S. To make the edges intact it selects such an element from S by which the similarity measure maintained efficiently. In step 20, the center pixel is replaced by t.

In this algorithm, the value of n is varies from 4 to 6, so the algorithm iterates a maximum of 6 times, i.e., constant time. The inner and outer loops iterate a total of (rows * columns) times. Steps 5-6 also iterate for a constant time. The internal steps run for a constant time as well. The total iteration is c_1*(rows*cols)*c_2 which is equivalent to O (rows*cols) times, which is quadratic polynomial, where c_1 and c_2 are constants.

3.2 Weighted Median Filter

The schemes for eliminating impulsive noise from digital images are presented in this section. Random Valued Noise (RVN) is considered for denoising. The *Variable Mask Median* filter, *Edge Preserving Restoration* filter and *Sub-image based Restoration* filter are described in this section.

In the previous section, some noise removal operators have been proposed where the noise detection technique is based on simple arithmetic operations like absolute differences of the center pixel with its neighboring pixels and the restoration of noise is done by standard deviation operation on the directional pixels in the test window. The noise detection operation is empowered in the following approach to make the sensitivity and specificity issues of the detection operator. To eliminate the *false alarm rate* of the noise detection operator, novel approach is proposed in this section. The noise restoration operation is based on the standard median filter in this section. The proposed median based filters are improved median filters. They weigh the neighboring pixels to select for the median operation. The most relevant pixels on the test window are selected prior to median operation. This scheme is simple but efficient and works in two phases: detection of noisy pixels followed by median based filtering.

Three techniques are proposed in this section based on improved impulse detection method followed by weighted median filtering technique. The first of which is *variable mask median filter* uses all neighbor directional pixels in the 5 $\times$ 5 test window for impulse noise detection. Simple arithmetic absolute differences are computed and then the minimum among them is selected to compare with an user parameter (*Threshold* T) for classification of the center pixel in the test window. To filter the noisy pixels both the 5 $\times$ 5 and 3 $\times$ 3

masks are used and a median based filter has been proposed. For filtering it uses an improved median based filtering technique which uses variable mask consisting of nine and twenty five pixels respectively. Certain pixels in the 3×3 window are selected prior to compute median operation. Three user parameters viz., *number of iterations(I), threshold(T) and decreasing rate (R) of threshold in each iteration* are varied in three dimensional space to obtain optimal results.

Another algorithm concentrates on *miss alarm rate* of the noise detection rigorously. The first step is to classify whether the center pixel in the 5×5 window is noisy or not, which is done using all neighbor directional weighted pixels in a 5×5 mask. The proposed algorithm performs simple arithmetic absolute differences on the pixels aligned in the four directions with the center pixel. On detection of noisy pixels an advance median filter has been proposed for restoration of noisy pixel where a variable window consisting of 3×3 and 5×5 respectively have been used. To optimize the results, a three dimensional search space has been used on user supplied parameters in a wide range.

A novel approach which aims at detection and filtering of impulses in digital images through median filtering has also been proposed here. The proposed detection method is based on all neighbor directional pixels. The absolute differences between each of four directional pixels with the center pixel are calculated for detection. The filtering technique is based on weighted median approach. Among four directions positive weights are assigned to the direction containing minimum standard deviation considering the fact that they have more intensity value similarity. The median value replaces the center pixel to restore it. The training image is logically partitioned into several 5×5 window sub regions. Each of the window separately goes through two different masks such as 3×3 and 5×5 and produces two different restored versions of that window. Each of the restored version is compared with the original to compare the fitness in terms of PSNR (dB). The correction obtaining maximum PSNR is accepted and incorporated in to the selected window. PSNR is calculated at end of restoration of all 5×5 sub window regions of the test image. Both 3×3 and 5×5 window regions based detection and filtering operators use all neighbor directional weighted pixels to include all the pixels in the window. Four directions of the center pixel within are considered to classify the center pixel. Only noisy pixel goes through the filtering operator, the remaining pixels are not altered.

The *Variable Mask Median Filter (VMM)*, *Edge Preserving Restoration (EPR) filter* and *Sub-image based Restoration (SIR) filter* are given in sections 3.2.1, 3.2.2 and 3.2.3, respectively.

3.2.1 Variable Mask Median Filter

The technique deals with *all neighbor directional weighted pixels (ANDWP)* for removal of highly random valued noise (RVN). The proposed approach works in two phases. The first phase detects the contaminated pixels by making the differences between the test pixel and its all neighbor pixels aligned in four directions in the 5 × 5 window. The 5 × 5 test window is given in Fig. 3.2 which includes all the neighboring pixels in the test window. The second phase filters only the noisy pixels based on median values of the four directional pixels which utilizes all the neighborhood pixels in both the 5 × 5 and 3 × 3 windows. The advancement done compared to the previous method are considering all the neighboring pixels in the 5 × 5 window in both the stages of detection and filtering, secondly the median operation selects relevant pixels and rejects irrelevant pixels in the 5 × 5 and 3 × 3 window regions by finding the minimum and maximum standard deviations among the directional pixels of the window.

3.2.1.1 Impulse Detector

The VMM impulse detection algorithm is given in Algorithm 5. In step 2, the algorithm runs for a total of n times where n is an user parameter. In the steps 3 and 4, the for loops grow to cover the maximum rows and maximum columns of the original image. Steps 5-7 iterate to form the set D_k, where k = 1 to 4 which contain all the twenty five pixels of the window. Let D_k (k=1 to 4) denotes a set of seven pixels aligned in the k^{th} direction centered at (0, 0), i.e.,

. D_1 = {(-1,-2), (-2,-2), (-1,-1), (0, 0), (1, 1), (2, 2),(1, 2)}.

. D_2 = {(1,-2), (0,-2), (0,-1), (0, 0), (0,1), (0, 2), (-1, 2)}.

. D_3 = {(2,-1), (2,-2), (1,-1),(0, 0), (-1, 1), (-2, 2), (-2, 1)}.

. D_4 = {(-2,-1), (-2, 0), (-1, 0), (0, 0), (1, 0), (2, 0), (2,1)}.

Then let $D_k^0 = D_k/(0,0)$, ∀ k =1 to 4. In step 8, in each direction of the 5 × 5 window centered at (i,j), it defines $d_{i,j}^{(k)}$ as the sum of all absolute differences of gray values between $y_{i+s,j+t}$ and $y_{i,j}$ with (s,t)∈ D_k^0 (k= 1 to 4), given in (3.13). In steps 9-17, in each direction, it multiplies the absolute differences between two closest pixels from the center pixel with a large value ω_m, multiplies the absolute differences between the center pixel and the corner pixels by ω_n and also multiplies the absolute differences between two end pixels from the center pixel with a small value ω_o, before it calculates the sum. It assigns ω_m =2, ω_n=1 and ω_o=0.5. The value of $\omega_{s,t}$ is given in (3.14).

$$d_{i,j}^{(k)} = \sum_{(s,t)\in D_k^0} \omega_{s,t}|y_{i+s,j+t} - y_{i,j}|, 1 \le k \le 4 \tag{3.13}$$

Algorithm 5 VMM noise detector

Input: Image Matrix g_{xy}, Threshold (T), Maximum Number of Iterations (n), Decreasing Rate of Threshold (R)

Output: Window's Center Pixel Classified

begin
for g=0 to n-1 **do**
  for x=0 to MaxRow-1 **do**
    for y=0 to MaxCol-1 **do**
      for i=-2 to 2 **do**
        for j=-2 to 2 **do**
          Initialize D_k with i and j, k= 1 to 4, Say $y_{i,j}$ be the center pixel
          $d_{i,j}^{(k)} = \omega_{s,t}|y_{i+s,j+t} - y_{i,j}|$
          if D $\in \omega^3$ (3.15) **then**
            $\omega_{s,t}$=2
          else
            if D $\in \omega^2$ (3.16) **then**
              $\omega_{s,t}$=1
            else
              $\omega_{s,t}$=0.5
            end if
          end if
          $r_{i,j} = \min d_{(i,j)}^{(k)}$
          if $r_{i,j} > T$ **then**
            $y_{i,j}$ is Noisy
          else
            $y_{i,j}$ is Noise-Free
          end if
        end for
      end for
    end for
  end for
  T= T * R
end for
end

$$\omega_{s,t} = \begin{cases} \omega_m & : (s,t) \in \omega^3 \\ \omega_o & : (s,t) \in \omega^2 \\ \omega_n & : \text{otherwise} \end{cases} \tag{3.14}$$

where

$$\omega^3 = \{(s,t) : -1 \leq s,t \leq 1\}, and \tag{3.15}$$

where

$$\omega^2 = \{(s,t) : (s,t) = \pm\{(-2,-1),(-1,-2),(1,-2),(2,-1)\}\}. \tag{3.16}$$

In step 18, the minimum of $d_{i,j}^{(k)}$ is calculated. The minimum of these four direction indices are used for impulse detection, which is given in (3.17).

$$r_{i,j} = min\{d_{i,j}^{(k)} : 1 \leq k \leq 4\} \tag{3.17}$$

In steps 19-22, the noisy pixel is identified efficiently from the window by employing an user parameter T. In step 28, the threshold value is decreased in each iteration by a rate of decreasing threshold rate (R).

In this algorithm, the value of n is varies from 4 to 6, so the algorithm iterates for a maximum of 6 times, i.e., constant time. In steps 3 and 4, the loops iterate a total of (rows * columns) times. Steps 5-6 iterate for a constant time. The internal steps are performed for a constant time as well. As a whole the algorithm iterates for c_1*(rows*cols)*c_2 which is equivalent to O (rows * cols) times, which is quadratic polynomial, c_1 and c_2 are constants.

3.2.1.2 Impulse Filter

This technique is capable of restoring image fine details and textures even when the image is highly noisy. For detection of noisy pixels it uses all the neighborhood pixels in the 5 × 5 window. As the detection operator uses the four directions as four edges so it does well in terms of edge keeping performance. And for filtering the noisy pixels it uses variable mask, both the 3 × 3 and 5 × 5. Initially it uses 5 × 5 mask for finding the maximum and minimum standard deviations but eventually to restore the noisy pixels it does median filtering on the 3 × 3 neighborhood pixels.

The VMM impulse restoration technique is given in Algorithm 6. For detection of noisy pixels the algorithm uses 5 × 5 test window. But filtering algorithm uses 3 × 3 window to restore the noisy pixel in the test window. Step 2 iterates for n times. In steps 3 and 4, the for loops iterate for the whole image matrix. The lines 5 and 6 iterate to contain all the pixels of the test window of size 5 × 5. In line 7, the standard deviations are computed. The directions where the minimum and maximum standard deviations present are selected in steps 8 and 9. In line 10, the median is calculated using the (3.18) by using the 3 × 3 window region pixels. The filter uses the 3 × 3 window and in which it does not use the directional pixels for which the standard deviation is maximum. Rather it uses the directional pixels two times denoted by the repetition or selection operator ⋄ [Brownrigg 1984] for which standard deviation is minimum. And center pixel is excluded from the median operation because it is

detected as noisy pixel. In steps 11-19, the weights are assigned to the 3×3 window region based pixels for selection of pixels for median operation. Eventually in step 23, the noisy pixel is replaced by $\hat{f}(x, y)$.

The time requirement for the algorithm is c_1 * (rows * cols) * c_2 which is equivalent to O (rows * cols) times, which is quadratic polynomial, where c_1 and c_2 are constants.

$$\hat{f}(x, y) = \underset{(s,t)\in\omega^4(3.19)}{\text{median}} \{\omega_{s,t} \diamond g(s, t)\} \tag{3.18}$$

where ω^4 is given in (3.19).

$$\omega^4 = \{(s, t) : (s, t) = \pm\{(-1, -1), (-1, 0), (-1, 1), (0, -1)\}\}. \tag{3.19}$$

3.2.2 Edge Preserving Restoration Filter

Edge Preserving Filter is a two step method proposed to denoise the digital images corrupted by high random valued noises. The first step is to classify whether the center pixel in the 5×5 window is noisy or not, which is done using all neighbor directional weighted pixels in 5×5 mask. The proposed technique performs simple arithmetic absolute differences on the pixels aligned in the four directions with the center pixel. On detection of noisy pixels an advance median filter has been proposed for restoration of noisy pixel where a variable window consisting of 3×3 and 5×5 respectively have been used. To optimize the results, a three dimensional search space has been used on user supplied parameters in a wide range. The strength of the algorithm is the detection algorithm which is proposed here. Here a novel scheme has been proposed which uses a two step method to remove random valued impulse noises in the digital images. It utilizes all the neighborhood pixels in the 5×5 window. The noise removal phase uses variable size of mask 3×3 and 5×5 to achieve good quality in restoring the noisy images. Three user parameters viz., number of iterations (I), threshold (T) and decreasing rate (R) of threshold in each of the iteration are varied in a wide three dimensional space to obtain optimal restoration results. Firstly the center pixel is classified as noisy or noise free in the noise detection step. The 5×5 test window is considered for this reason. In the test window, in each direction seven pixels are contained in it. The pixel under consideration goes through two level of detection to conform it as a noise free. The *false alarm rate* decreases to a very low value by the approach. The noise free pixels should not be detected as noisy to preserve the image fine textures by the algorithm. That objective is achieved to some extent by the algorithm. In the filtering stage the proposed algorithm restores the noisy pixels by weighted median filtering by the help of 3×3 and 5×5 window regions. At first it finds the most relevant directions in the 5×5 to take the help

Algorithm 6 VMM filter

Input: Image Matrix g_{xy}, Maximum Number of Iterations (n)
Output: Window's Center Pixel Restored

begin
for g=0 to n-1 **do**
  for x=0 to MaxRow-1 **do**
    for y=0 to MaxCol-1 **do**
      for i=-2 to 2 **do**
        for j=-2 to 2 **do**
          Calculate $\sigma_{i,j}^{(k)}$ D_k values, k= 1 to 4
          $l_{i,j} = \min(\sigma_{i,j}^{(k)})$
          $m_{i,j} = \max(\sigma_{i,j}^{(k)})$
          Find $\hat{f}(x,y) = \underset{(s,t)\in\omega^4(3.19)}{\text{median}} \{\omega_{s,t} \diamond g(s,t)\}$
          if $(s,t) \in D^0_{m_{i,j}}$ **then**
            $\omega_{s,t} = 0$
          else
            if $(s,t) \in D^0_{l_{i,j}}$ **then**
              $\omega_{s,t} = 2$
            else
              $\omega_{s,t} = 1$
            end if
          end if
        end for
      end for
    end for
    Replace center pixel $y_{i,j}$ by $\hat{f}(x,y)$
  end for
end for
end

of maximum possible information from the neighbor then it calculates the median value by the help of pixels in the 3 × 3 window. The 3 × 3 window used only for median filtering is given in Fig. 3.3. The closest pixels from the center pixel are taken zero or more times for median filtering because they have maximum local similarity with the center pixel.

3.2.2.1 Impulse Detector

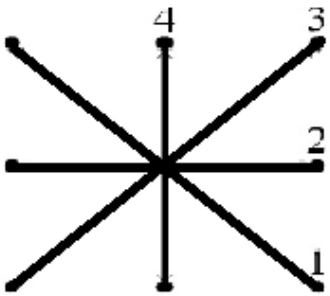


Figure 3.3: Four directions of an 3 × 3 window where the center pixel is under consideration

The edge preserving restoration of digital images algorithm is presented in Algorithm 7. The proposed impulse detection algorithm uses maximum possible information in the 5 × 5 window. A simple decision rule is used here to define a noisy pixel. At the beginning it finds the minimum and maximum intensity values of the twenty four pixels within the test window around the center pixel, $y_{i,j}$. If the intensity value of the central pixel lies within the intensity range spread of its neighbors, it is passed to the next level of detection. Otherwise the center pixel is detected as corrupted pixel. Let $y_{i,j}$, W_{min} and W_{max} are intensity values of the center pixel, maximum and minimum gray values, respectively within the selected window around $y_{i,j}$. Then the decision rule is given in (3.20). Initially this decision function is performed to classify the center pixel as noisy.

$$y_{i,j} = \begin{cases} Undetected\ Pixel & : y_{i,j} > W_{min}\ \&\&\ y_{i,j} < W_{max} \\ Noisy\ Pixel & : y_{i,j} \leq W_{min}\ \&\&\ y_{i,j} \geq W_{max} \end{cases} \tag{3.20}$$

The technique given in Algorithm 7. Step 2 iterates for a maximum of 6 times as the n varies in range [4,6]. Steps 3-4 iterate to maximum row and maximum column dimensions of the image matrix. Then it iterates to contain the 5 × 5 window in row major order where the center pixel, $y_{i,j}$ is under consideration. Steps 5, 6 and 7 iterate to form the set D_k, where k = 1 to 4, which contain all the twenty five pixels of the window. Let D_k (k=1 to 4) denotes a set of seven pixels aligned in the k^{th} direction centered at (0, 0), i.e.,

. D_1 = {(-1,-2), (-2,-2), (-1,-1), (0, 0), (1, 1), (2, 2),(1, 2)}.

. D_2 = {(1,-2), (0,-2), (0,-1), (0, 0), (0,1), (0, 2), (-1, 2)}.

. D_3 = {(2,-1), (2,-2), (1,-1),(0, 0), (-1, 1), (-2, 2), (-2, 1)}.

. D_4 = {(-2,-1), (-2, 0), (-1, 0), (0, 0), (1, 0), (2, 0), (2,1)}.

Then let $D_k^0 = D_k/(0,0)$, ∀ k =1 to 4. In step 8, in each direction of the 5 × 5 window centered at (i,j), it defines $d_{i,j}^{(k)}$ as the sum of all absolute differences of gray values between

$y_{i+s,j+t}$ and $y_{i,j}$ with (s,t)$\in D_k^0$ (k= 1 to 4), given in (3.13). In steps 9-17, in each direction, it multiplies the absolute differences between two closest pixels from the center pixel with a large value ω_m, multiplies the absolute differences between the center pixel and the corner pixels by ω_n and also multiplies the absolute differences between two end pixels from the center pixel with a small value ω_o, before it calculates the sum. It assigns ω_m =2, ω_n=1 and ω_o=0.5. In step 18, it finds the minimum of $d_{i,j}^{(k)}$ and $r_{i,j}$ is obtained. In steps 20-22, the algorithm defines a noisy pixel by the definition of a noisy pixel discussed above. In steps 23-27, the algorithm defines a noise free pixel by the complete decision rule given in (3.21). In this algorithm the noise detection algorithm permits a pixel to be a noise free if it passes two levels of detection stages. T is decreased in step 32 by the rate of R. The total iteration in the algorithm is c_1*(rows*cols)*c_2 which is equivalent to O(rows*cols) times, which is quadratic polynomial, where c_1 and c_2 are constants.

$$y_{i,j} = \begin{cases} Noise\ Free\ Pixel & : r_{i,j} \ \leq \ T \ \&\& \ y_{i,j} > W_{min} \ \&\& \ y_{i,j} < W_{max} \\ Noisy\ Pixel & : y_{i,j} \leq W_{min} \ \&\& \ y_{i,j} \geq W_{max} \end{cases} \tag{3.21}$$

3.2.2.2 Impulse Filter

The impulse filter for the edge preserving restoration technique is given in Algorithm 6. In the proposed noise detection and filtering algorithms, it uses three user supplied parameters viz., I, T and R. These parameters are varied in the following ranges of values to obtain the restoration results in terms of PSNR (dB) given in this algorithm.

The filter utilizes an advance switching median filter for restoration of images corrupted with high density of impulse noise in which the filtering is applied to only the corrupted pixels in the image while the uncorrupted pixels are kept unchanged. For impulse detection it utilizes all neighbor directional weighted pixels in the test window. The detection of noisy pixels in the 5 $\times$ 5 test window is done by a simple comparison and arithmetic absolute differences from the pixel under question with the other pixels in that specified window region. To restore a noisy pixel detected by the impulse detector the algorithm devises an enhanced median filter which utilizes both the 3 $\times$ 3 and 5 $\times$ 5 window regions. Out of four directions of the 5 $\times$ 5 test window three directions are used in the median filtering. Simple standard deviations are computed on the four directional pixels to select the most important and most unimportant directions. Repetition operator $\diamond$ is used to signify that operation. The median value is calculated on 3 $\times$ 3 window regional pixels. The center pixel which is detected as noisy is not used to find the median value. Three user supplied parameters are varied manually in a wide range to get the maximum possible fitness value by the proposed algorithm given in this algorithm.

3.2.3 Sub-Image Restoration Filter

The technique aims at detection and filtering of impulses in digital images through median filtering technique. The image is logically partitioned into 5×5 sub images. Two restored versions are generated through proposed technique using 3×3 and 5×5 masks separately of which better one in terms of PSNR (dB) is chosen. The proposed detection method is based on all neighbor directional pixels. The absolute differences between each of four directional pixels with the center pixel are calculated for detection. The filtering is based on weighted median approach. Among four directions positive weights are assigned to the direction containing minimum standard deviation considering the fact that they have more intensity value similarity. Thus certain pixels are selected for median filtering. The median value replaces the center pixel to restore it. The three user parameters used for detection and filtering are varied in a wide range to obtain the optimal solutions.

This algorithm proposes a weighted median filter based impulse detection and suppression techniques in digital images. The training image is logically partitioned into several 5×5 window sub regions. Each of the window separately goes through two different masks such as 3×3 and 5×5 and produces two different restored versions of that window. Each of the restored versions is compared with the original to compare the fitness in terms of PSNR (dB). The correction obtaining maximum PSNR is accepted and incorporated in to the selected window. PSNR (dB) is calculated at end of restoration of all 5×5 sub window regions of the test image. Both 3×3 and 5×5 window based detection and filtering operator uses all neighbor directional pixels to include all the pixels in the window. Four directions of the center pixel within are considered to classify the center pixel. Only noisy pixel goes through the filtering operator, the remaining pixels are not altered. The detection operator computes arithmetic operations on the directional pixels with the center pixel to define a noisy pixel. The filtering operator selects the most relevant directional pixels among the four to calculate the median value and that replaces the center pixel. Four standard deviations are calculated in four directions to select pixels of more importance. Three user parameters are used for detection and filtering operations, viz., threshold (T), maximum number of iterations (I_{max}) and decreasing rate of threshold in each iterations (R) are varied in a range to get the optimal restoration results. The proposed noise removal operator performs noise denoising on every pixels of 5×5 sub image using 3×3 and 5×5 neighboring pixels of the pixel under question separately. The process diagram is shown in Fig. 3.4. The proposed impulse detection and filtering operator consists of two main processes which illustrate the impulse detection and filtering. Each 5×5 subimage is restored using best optimal result obtained out of 3×3 and 5×5 masks. The 3×3 and 5×5 window regions are already shown in this chapter. In both figures, four directions are considered based on center pixel to classify that pixel. The second one shows 5×5 mask which includes all the neighborhood pixels of the center pixel. These pixels will be referred as all neighboring directional weighted pixels (ANDWP).

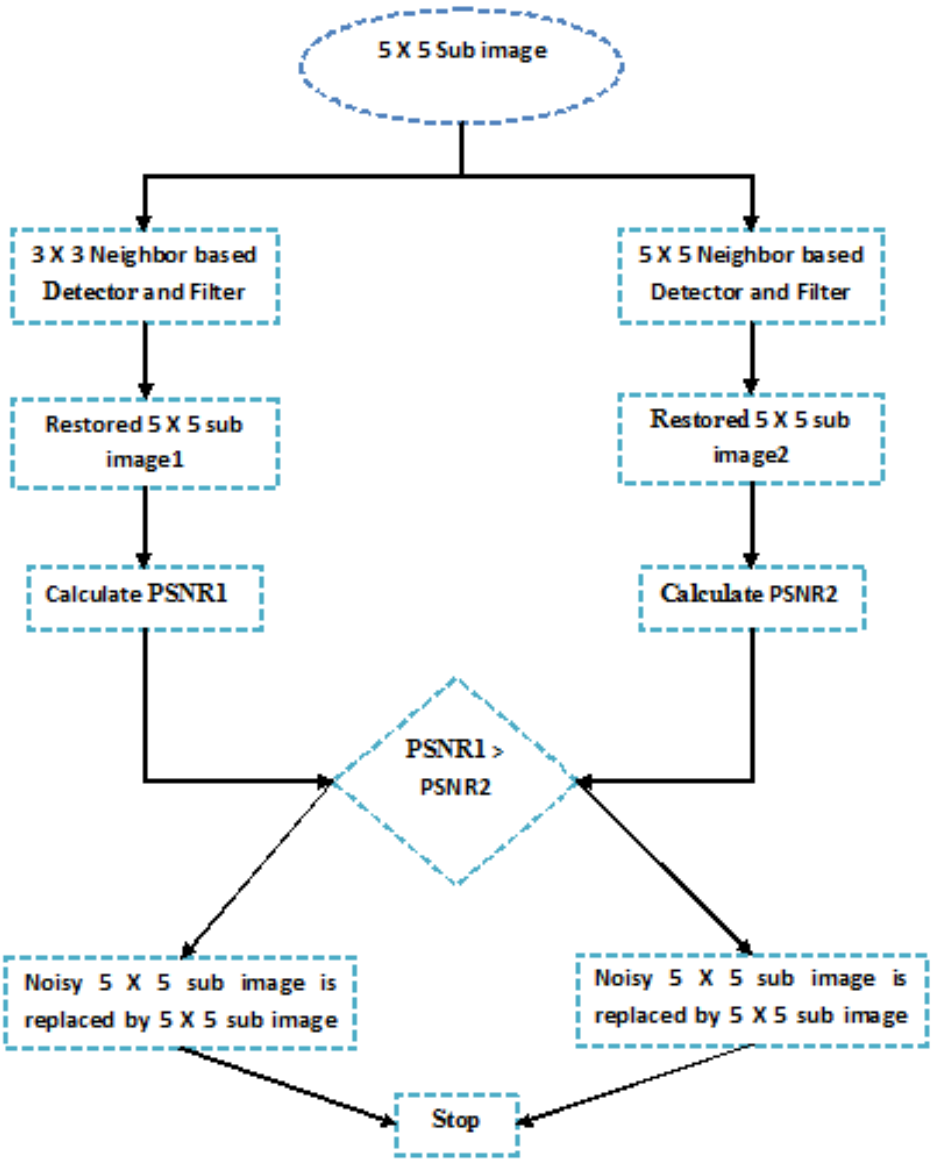


Figure 3.4: Process diagram for each of the 5 × 5 sub image

The impulse detection rule is given in subsection 3.2.3.1 and that of filtering rule is given in subsection 3.2.3.2.

3.2.3.1 $W_{d1} \times W_{d2}$ based Impulse Detector

The 3×3 and 5×5 window regions based detection operators are almost identical except their sizes. The maximum and minimum gray values in each of the 3×3 and 5×5 window regions are determined. When the center pixel of the test window does not lie in the range of maximum and minimum gray values, it is detected as noisy. Otherwise the absolute differences between the center pixel and all other pixels in each direction are calculated. These are known as direction indices. The minimum of these indices are determined and compared with a threshold value. Eventually the complete detection rule is formed. The impulse detection rule is outlined in Algorithm 8. Step 2 iterates for a maximum of 6 times as the n varies in range [4, 6]. Steps 3-4 iterate to maximum rows and maximum columns to contain all the pixels in the $W_{d1} \times W_{d2}$ window. Steps 5-7 iterate to form the set D_k where D_k (k=1 to 4) denotes the set of coordinates aligned in the four directions centered at (0, 0) in the window, i.e.,

D_1={(-y,-y),...,(-1,-1), (0,0), (1,1),...,(y, y)}.

D_2={(0,-y),...,(0,-1), (0,0), (0,1),...,(0,y)}.

D_3={(y,-y),...,(1,-1), (0,0), (-1,1),...,(-y, y)}.

D_4={(-y,0),...,(-1,0), (0,0), (1,0),...,(y,0)}.

For 5×5 window two end pixels in each direction included for impulse detection and filtering purposes are as follows:

D_{1a}= {(-2,-1), (-2, 1)}.

D_{2a}= {(-1,-2), (-1, 2)}.

D_{3a}= {(1,-2), (1, 2)}.

D_{4a}= {(2,-2), (2, 1)}.

In steps 8-17, $d_{i,j}^{(k)}$ is calculated as the absolute differences between the center pixel and all other directional pixels around the center pixel and then the differences in each direction are multiplied by weighting factors. In step 18, the minimum of the direction indices are computed. The impulse detection is done by the decision rule given in (3.21). T is decreased in step 32 by the rate of R. The time complexity of the algorithm is c_1 * (rows*cols) * c_2 which is equivalent to O (rows * cols) times, which is quadratic polynomial, where c_1 and c_2 are constants.

3.2.3.2 $W_{d1} \times W_{d2}$ Window based Impulse Filter

The filtering rule using 5 × 5 neighbor is almost similar to 3 × 3 neighbor based filtering. It uses a statistical tool, i.e., standard deviation for selecting the pixels in the window to compute the median operation. The directions where the standard deviations are on the minimum side are determined and taken for the median operation. Some of the pixels of the test window are given more weights and they are selected number of times and some pixels are not taken at all to form the set prior to the median operation. The pixels which have minimum spatial distances with the center pixel are given larger weight. The technique is outlined in Algorithm 9. In the algorithm, step 2 iterates for n times. Steps 3-4 iterate to maximum rows and maximum columns to contain all pixels in the $W_{d1} \times W_{d2}$ window. Steps 5-7 iterate to form the set D_k where D_k (k=1 to 4) denotes the set of coordinates aligned in the four directions centered at (0, 0) in the window. In step 7, four standard deviations are obtained in four directional pixels. In steps 8-9, the directions and pixels are taken at the directions where standard deviations are minimum and second minimum. In steps 10-20, the median is calculated based on the weights of the pixels, which are used to select by the repetition operator $\diamond$. In step 24, the center pixel is replaced by the median value.

$$\omega^1 = \{(s,t) : -1 \leq (s,t) \leq 1\} \tag{3.22}$$

$$\omega^2 = \{(u,v) : -1 \leq (u,v) \leq 1\} \tag{3.23}$$

$$\omega^3 = \{(p,q) : -2 \leq (p,q) \leq 2 \; and \; (p,q) \neq [-1,0,1]\} \tag{3.24}$$

The algorithm for restoration of impulses is proposed where the digital images are contaminated by any level or density of noises. The total iteration count of the algorithm is c_1 * (rows * cols) *c_2 which is equivalent to O (rows * cols) times, which is a quadratic polynomial, where c_1 and c_2 are constants.

The simple noise suppression techniques presented in this chapter are evaluated through results and comparisons with the state of the art methods in section 9.1 of chapter 9. In this case the digital images are corrupted with *random valued noises*. The section contains results and comparisons on five algorithms such as *Directional Weighted Minimum Deviation (DWMD)*, *All Neighbor Directional Weighted Pixels (ANDWP)*, *Variable Mask Median (VMM)*, *Edge Preserving Restoration (EPR)* and *Sub-Image Restoration (SIR)* based filters. The *liftingbody*, *boat*, *lena*, *barbara* and *cameraman* images are used to evaluate the proposed algorithms. Color *lena* image is also used to evaluate the proposed algorithms for noisy color image. Those images are corrupted using minimum (10%) to maximum noise density (100%) using the noise models. The results given by the proposed algorithms are presented by three popular and useful performance metrics such as Peak-Signal-Noise Ratio, Image Enhancement Factor, Structure Similarity Index and visual effects.

3.3 Discussions

In this chapter, techniques for digital image denoising have been presented which are based on simple arithmetic operations. Absolute differences are computed in the noise detection stage. Standard deviations are also used for such purpose. The maximum and minimum pixel intensity values within a test window are used to detect the noisy pixels. 3×3 and 5×5 window regions are considered in row major order and slides to next window to cover the whole input image. In both the test window regions all the pixels in the window are utilized for detection and filtering purposes in this chapter. In 5×5 test window all the twenty five pixel are utilized in order to classify and restore the center pixels. Those pixels are referred to as all neighbor directional weighted pixels. In the test window four directions are associated in calculation to have the edges intact. The objective of any image de noising algorithm is to suppress the noise as much possible while to preserve the image fine textures as well. The image should not be blurred at all. Hence the pixels within the test window are considered in directional positions. The direction where standard deviation is minimum may be considered as an edge in the image. In the test window the four directional pixels are preserved to have the edges intact. The center pixel within each test window are classified as noisy or noise free by the help of the above operations. The time complexities of the proposed algorithms are also, studied and proved to be in order of polynomial.

In the filtering stage, either standard deviation based approach or median filtering approaches have been incorporated in the algorithms presented in this chapter. In case of standard deviation based approach, the noisy center pixel is replaced by the pixel which has maximum similarity within the test window. As it is known that the standard deviation is the measure of how closely the set of pixels are clustered. The median operations are also done in the algorithms for restoring the pixels which are noisy. The pixels which are most relevant are selected first and then they are applied to median filtering. The pixels which are most relevant are selected more than one time and the pixel which has not such similarity with the test window are rejected for median operations. Thus weighted median filtering approach has been incorporated in this chapter to restore the noisy pixels.

The first method presented in this chapter is based on standard deviation based filtering. In this method, a directional weighted minimum deviation (DWMD) based filter has been proposed for removal of high random valued impulse noise (RVN). The proposed approach based on standard deviation works in two phases. The first phase detects the contaminated pixels by differencing between the test pixel and its neighboring pixels aligned with four directions. The second phase filters only those pixels keeping others intact. The filtering scheme is based on minimum standard deviation of the four directional pixels.

A variance based filter has also been proposed for removing high random valued impulse noise from digital images. In the proposed algorithm, all the twenty four neighbors of the

center pixel in the 5 × 5 window are included and used for noise detection. As a result it gives very less miss and false alarm values compared to other filters. The fundamental superiority of the proposed operator over most other operators is that it efficiently removes impulse noises from highly corrupted images while successfully preserves the thin lines, edges and fine details in the enhanced image.

The variable mask median filter (VMM) of suppressing random valued impulses in the digital images is simple and robust. This method is capable of preserving fine details and textures of images even when the image is corrupted with high density of impulses. For detection of noisy pixels all the neighborhood pixels in the 5 × 5 window are taken into considerations. As the detection operator uses the four directions as four edges so it performs well in retaining of edges. The VMM uses variable mask, both the 3 × 3 and 5 × 5 to filter a noisy pixel. Initially it uses 5 × 5 mask for finding the directions which have the maximum and minimum standard deviations but eventually to restore the noisy pixels it does median filtering on the 3 × 3 mask. Certain pixels of the 3 × 3 window are selected and repeated prior to median filtering.

EPR filter proposes an advanced switching median filter for restoration of images corrupted with high density of impulse noise in which the filtering is applied to only the corrupted pixels in the image while the uncorrupted pixels are kept unchanged. For impulse detection it utilizes all neighbor directional weighted pixels in the test window. The detection of corrupted pixels in the 5 × 5 test window is done by a simple comparison and arithmetic absolute differences from the pixel under question with the other pixels in that specified window region. To restore a noisy pixel detected by the impulse detector it proposes an enhanced median filter which utilizes both the 3 × 3 and 5 × 5 window regions. Out of four directions of the 5 × 5 test window three directions are used in the median filtering. Simple standard deviations are computed on the four directional pixels to select the most important and irrelevant directions. The median value is calculated on 3 × 3 window regional pixels. The center pixel which is detected as noisy is not used to find the median value.

Sub image restoration filter is an efficient algorithm for restoration of impulses in digital images which are contaminated by any level or density of noises. It is an improved switching median filter based on 3 × 3 and 5 × 5 neighborhood pixels where the filter is applied to only the corrupted pixels in the training images while the noise free pixels are left unchanged. The full length image is broken into several 5 × 5 sub images. Each of the sub image is restored using two different masks such as 3 × 3 and 5 × 5. Among the two different restored sub images better results in terms of PSNR is taken. The detection and filtering algorithms are very simple but efficient. The computational cost of each of the algorithm presented in this chapter are studied and presented in the results section.

Algorithm 7 EPR noise detector

Input: Image Matrix g_{xy}, Threshold (T), Maximum Number of Iterations (n), Decreasing Rate of Threshold (R)

Output: Window's Center Pixel Classified

begin
for g=0 to n-1 **do**
  for x=0 to MaxRow-1 **do**
    for y=0 to MaxCol-1 **do**
      for i=-2 to 2 **do**
        for j=-2 to 2 **do**
          Initialize D_k with i and j, k= 1 to 4, Say $y_{i,j}$ be the center pixel
          $d_{i,j}^{(k)} = \omega_{s,t}|y_{i+s,j+t} - y_{i,j}|$
          if D $\in \omega^3$ (3.15) **then**
            $\omega_{s,t}$=2
          else
            if D $\in \omega^2$ (3.16) **then**
              $\omega_{s,t}$=1
            else
              $\omega_{s,t}$=0.5
            end if
          end if
          $r_{i,j} = \min d_{(i,j)}^{(k)}$
          Say W_{min} and W_{max} be the minimum and maximum gray values in 5×5 window
          if $y_{i,j} \leq W_{min}$ && $y_{i,j} \geq W_{max}$ **then**
            $y_{i,j}$ is Noisy
          end if
          if $y_{i,j} > W_{min}$ && $y_{i,j} < W_{max}$ **then**
            if $r_{i,j} \leq T$ **then**
              $y_{i,j}$ is Noise-Free
            end if
          end if
        end for
      end for
    end for
  end for
  T=T*R
end for
end

Algorithm 8 SIR noise detector

Input: Image Matrix g_{xy}, Threshold (T), Maximum Number of Iterations (n), Decreasing Rate of Threshold (R)

Output: Window's Center Pixel Classified

begin
for g=0 to n-1 **do**
  for x=0 to MaxRow-1 **do**
    for y=0 to MaxCol-1 **do**
      for i=-(2*m+1) to (2*m+1) **do**
        for j=-(2*m+1) to (2*m+1) **do**
          Initialize D_k, k= 1 to 4, Say $y_{i,j}$ be the center pixel
          $d_{i,j}^{(k)} = \omega_{s,t}|y_{i+s,j+t} - y_{i,j}|$
          if D $\in \omega^3$ (3.15) **then**
            $\omega_{s,t}$=2
          else
            if D $\in \omega^2$ (3.16) **then**
              $\omega_{s,t}$=1
            else
              $\omega_{s,t}$=0.5
            end if
          end if
          $r_{i,j} = \min(d_{i,j})$
          Say W_{min} and W_{max} be the minimum and maximum intensity values within the $W_{d1} \times W_{d2}$ test window around the center pixel $y_{i,j}$
          if $y_{i,j} \leq W_{min}$ && $y_{i,j} \geq W_{max}$ **then**
            $y_{i,j}$ is Noisy
          end if
          if $y_{i,j} > W_{min}$ && $y_{i,j} < W_{max}$ **then**
            if $r_{i,j} \leq T$ **then**
              $y_{i,j}$ is Noise-Free
            end if
          end if
        end for
      end for
    end for
  end for
  T=T*R
end for
end

Algorithm 9 SIR filter

Input: Image Matrix g_{xy}, Maximum Number of Iterations (n)
Output: Window's Center Pixel Restored

begin
for g=0 to n-1 **do**
  for x=0 to MaxRow-1 **do**
    for y=0 to MaxCol-1 **do**
      for i=-(2*m+1) to (2*m+1) **do**
        for j=-(2*m+1) to (2*m+1) **do**
          Calculate $\sigma_{i,j}^{(k)}$ using D_k values, k= 1 to 4
          $min1_{i,j} = \min(\sigma_{i,j}^{(k)})$, k= 1 to 4
          $min2_{i,j} = \min(\sigma_{i,j}^{(k)})$, k= 1 to 3
          $\hat{f}(x,y)$= $\underset{(s,t)\in\omega^1,(p,q)\in\omega^2,(u,v)\in\omega^3}{\text{median}}\{\omega_{s,t} \diamond g(s,t)\ and\ \omega_{p,q} \diamond g(p,q)\ and\ \omega_{u,v} \diamond g(u,v)\}$
          ω^1, ω^2 and ω^3 are given in (3.22), (3.23) and (3.24) respectively
          if $(s,t) \in D^0_{min1_{i,j}}$ **then**
            $\omega_{s,t} = 2$
          end if
          if $(p,q) \in D^0_{min1_{i,j}}$ **then**
            $\omega_{p,q} = 1$
          end if
          if $(u,v) \in D^0_{min2_{i,j}}$ **then**
            $\omega_{u,v} = 2$
          end if
        end for
      end for
    end for
    Replace center pixel $y_{i,j}$ by $\hat{f}(x,y)$
  end for
end for
end

Chapter 4

Filter Optimization

Contents

4.1 Based on Genetic Algorithm 94

4.1.1 The Proposed Technique 95

4.2 Based on Particle Swarm Optimization 100

4.2.1 The Proposed Technique 100

4.3 Discussions 104

The noise removal operators discussed in the previous chapter are optimized in this chapter to get the global optimal restoration results. Meta- heuristic optimization [Nash 1996, Osman 1996] methods are applied for optimization of the decision variables of the optimization problem. Some user parameters of the noise removal operators are searched using the meta-heuristic optimization techniques in this chapter. Three user parameters viz. number of iterations (I), threshold value (T) and decreasing rate of threshold value in each iteration (R) of the algorithms are searched in a three dimensional search space in this chapter. The first parameter of the algorithm is number of iterations (I) which will be varied in a fixed range to get the optimal restoration results. The most important parameter which is used in the noise detection rule is the Threshold (T) value. Depending upon the value of the threshold any pixel in the image is classified as noisy or noise free. Lastly, in each generation or iteration the threshold value is decreased by a multiplying factor as generation wise the noise density is decreased by an amount. In simple words, the pixel intensity values are changed in each generation of the algorithm. So the threshold value should be decreased generation wise. Three most relevant parameters are searched in a wide range to get the optimal restoration results. The restoration results are obtained on the basis of peak signal to noise ratio (PSNR (dB)). So the optimization algorithm that has a fitness or objective function which should always be *maximized to get larger PSNR (dB)* values by the generation.

Two optimization algorithms which are stochastic and randomized in nature are used for optimization of the noise removal operators discussed so far. *Genetic Algorithm (GA) [Holland 1992, D.E. 1989]* and *Particle Swarm Optimization (PSO)* based optimization algorithms are proposed in this chapter. Genetic algorithm based optimization algorithm

has been proposed here first because it can solve every optimization problem which can be described with the chromosome encoding. *Particle Swarm Optimization* is a relatively recent heuristic search method whose mechanics are inspired by the swarming or collaborative behavior of biological populations. This technique is proposed because of its ease of implementation and fast convergence rate. PSO is more computationally efficient (uses less number of function evaluations) than GA. In this chapter, both the algorithms are examined and investigated to optimize the fitness function of the noise removal operators depending upon the constraints viz., number of iterations (I), threshold value (T) and decreasing rate of threshold value in each iteration (R) of the noise detection and filtering techniques.

Genetic algorithm and Particle swarm optimization based optimization techniques are presented in sections 4.1 and 4.2, respectively. The chapter has a discussion in section 4.3.

4.1 Based on Genetic Algorithm

In this technique, a basic binary encoded GA [Williams 1998] with binary tournament selection, uniform crossover and low probability mutation rate are employed to solve the problem. The GA represents the design variables of each individual design with binary strings of 0's and 1's that are referred to as chromosome. Three parameters of the detection and filtering operators are used and searched in a three dimensional space to find the optimal solutions using GA. Fig. 4.1 shows the context level process diagram of the technique. Three main processes such as detection of noisy pixel, genetic algorithm and filtering of noisy pixels exchange arguments among themselves. The encoded chromosomes along with the noisy image are passed by the genetic algorithm to the detection process for detecting the noisy pixels. Subsequently the detection method returns a Boolean output to the GA process. Now if the Boolean value is true then the GA process calls the filtering process by sending the encoded chromosomes along with two images i.e., original image and noisy image. Finally the process returns the fitness values in terms of *PSNR*s for the corresponding chromosomes. As a result three user parameters viz. numbers of iterations (I), threshold value (T) and decreasing rate of threshold value in each of the iteration (R) are searched in a three dimensional search space for which the optimal solutions are obtained using genetic algorithm. These three parameters are estimated to get optimal restoration results by a search technique that mimic the evolution of populations of individuals i.e., GA. Using this technique, the detection and filtering algorithm does not require any parameter to be supplied by the user for any level of noise density in the image. The GA based optimization technique is discussed in section 4.1.1.

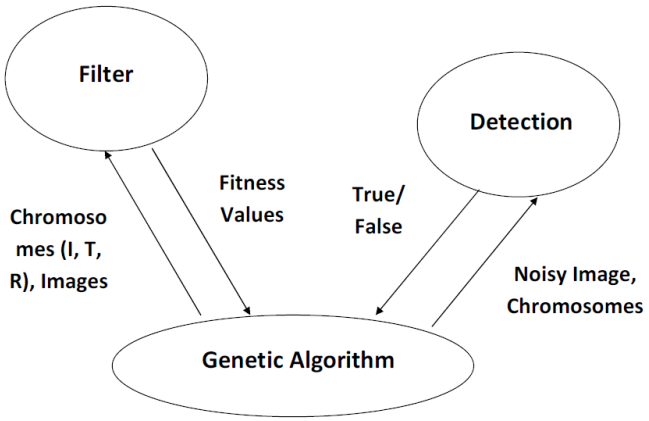


Figure 4.1: Context level diagram of optimization technique

4.1.1 The Proposed Technique

The performances of filters are tuned manually in general using a very limited search space using various parameters to optimize the performance. Unfortunately there is no analytical method to determine the optimal values for these parameters that yield the best results for a given filtering experiment. As a result, values of these parameters are heuristically determined and experimentally verified for each individual simulation experiment. In the optimization process three user parameters are taken using three dimensional search space using a randomized search and optimization technique.

Simple genetic algorithm based optimization technique works as follows. Initial population of individuals are encoded randomly followed by the fitness of all individuals are evaluated. Until a termination condition is obtained fittest individuals are selected for reproduction, crossover and recombination by generating new population in each cascading stage. Extensive simulation has been performed using six sub procedures of genetic algorithm. The process starts with encoding of initial population followed by the evaluation of fitness, elitism, selection, crossover and mutation. The parameters, attributes and detail techniques are discussed in the following subsections, 4.1.1.1 to 4.1.1.6.

4.1.1.1 Chromosome Encoding and Initial Population

In Genetic algorithm based noise detection and filtering scheme the binary chromosomes are used to encode the three parameters I, T and R as follows:

Encoded Chromosome:

P: Iteration (I) + Threshold (T) + Decreasing Rate of T (R)

The size of the encoded chromosome (P) is of 21 bits length out of which four bits are kept for iteration (I), ten bits for threshold (T) and seven bits for decreasing rate of T termed as R. First four bits of a chromosome represents the number of iterations (I) to afford a maximum of 15 iterations, intermediate 10 bits used to encode the maximum threshold value of 1024 and last 7 bits to encode the decreasing rate of threshold value in each iteration of up to a value of 0.95. Each parameter is converted to decimal and mapped into a specific range. Number of iterations (I) is considered in the range of 4 to 12 for computation in a single chromosome. If the decimal value corresponding to the first 4 bits of the chromosome is i, then the mapping is done as follows:

If ($i < 4$) then i=4;

If ($i > 12$) then i=12;

Threshold value (T) is also mapped into the range of 300 to 1000. Let the decimal value corresponding to the 10 bits of the chromosome is t, then the mapping is as follows:

If ($t < 300$) then t=300;

If ($t > 1000$) then t=1000;

For the decreasing rate of threshold value (R), last 7 bits of the chromosome is d, and then the mapping is done using a function given in (4.1). R is mapped into the range of [0.6, .95] using the (4.1).

$$R(d) = 0.6 + \frac{0.95 - 0.6}{127 - 0} * d \tag{4.1}$$

4.1.1.2 Fitness Evaluation

Fitness/objective function is associated with each chromosome. This indicates the degree of goodness of the encoded solution. Fitness of each chromosome is determined by the parameter such as *Peak Signal to Noise Ratio (PSNR (dB))* and *Image Enhancement Factor IEF)*. As enhancement of *PSNR* increases IEF, the PSNR is selected as the parameter to act as the objective function/fitness value of the chromosomes. So the system maximizes the *PSNR(dB)* of the restored image in each generation. The fitness function for the GA based optimization algorithm is given in (4.2), where M and N are the dimensions of the

input images respectively. I_1 and I_2 are the original and enhanced images respectively.

$$f(I_1, I_2) = PSNR(dB) = 10 * \log_{10}(\frac{255^2}{\frac{1}{M*N}\sum_{m,n}(I_1 m, n - I_2 m, n)^2}) \tag{4.2}$$

Table 4.1 shows the *PSNR* values obtained in simulation for four chromosomes.

Table 4.1: Sample results of fitness evaluation

Population (P=4) and Length (L) =21 bits	Decimal			Fitness Value
	I	T	R	PSNR
011010000011001011110	06	524	0.863	24.90
000110101010010011100	04	681	0.678	25.69
101000001011101111111	10	300	0.950	26.39
011001110001010011010	06	453	0.672	27.29

4.1.1.3 Elitism

In this technique one copy of the best chromosome is kept outside the population of each generation, which propagate for the selection, cross over and mutation. At the end of each generation the worst has also been selected and if it is seen that it is better than the best chromosome of the previous generation then it survives in the population otherwise is replaced by the best chromosome of the previous generation.

4.1.1.4 Selection

In each generation a mating pool is formed with the same size of the population consisting of better chromosomes and for the purpose, the Binary Tournament Selection (BTS) method [D.E. 1989] is used. Best chromosome between the two has been selected and copied into the mating pool, and the process repeated until the mating pool is full. Using this method, the chromosome with the lowest fitness value can never be copied into the mating pool. If there is a tie, then the chromosome, for which the number of iterations is lower, is selected for the mating pool.

4.1.1.5 Crossover

Crossover means exchange of genetic information. It takes place among randomly selected parent chromosomes from the mating pool. Single point crossover and uniform crossover

[Z. 1996] are the most commonly used schemes. It is a probabilistic operation. It occurs with high probability (μ_c) in each generation. A real value is taken randomly and if it is seen that it is less than (μ_c) then cross over occurs. Otherwise the two parents are copied to the next pool for mutation directly. The uniform crossover method is done here. The process has been iterated n/2 times for a mating pool of size n. First two parent chromosomes are selected from the mating pool followed by generation of a binary mask of length 21 bits. Depending upon the mask bit value (either 0 or 1) the crossover has been made. If mask bit value of i^{th} position is 1 then the i^{th} bits of the parent chromosomes are swapped. Otherwise crossover does not occur for that bit position. From n number of mating pool chromosomes n numbers of off springs have been made for the next generation. One example of uniform cross over is shown in Fig. 4.2.

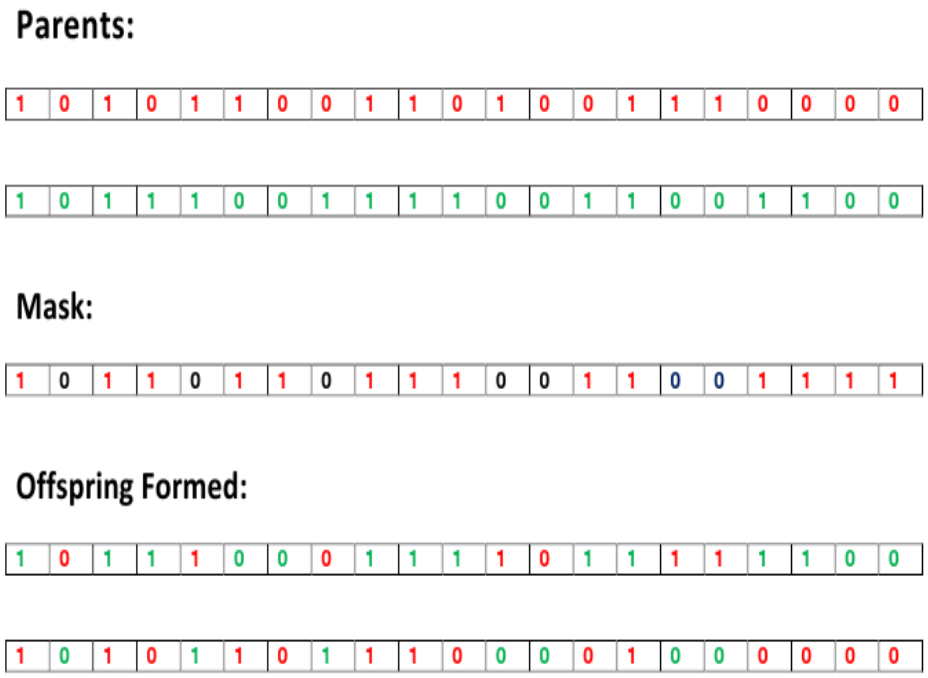


Figure 4.2: Uniform crossover

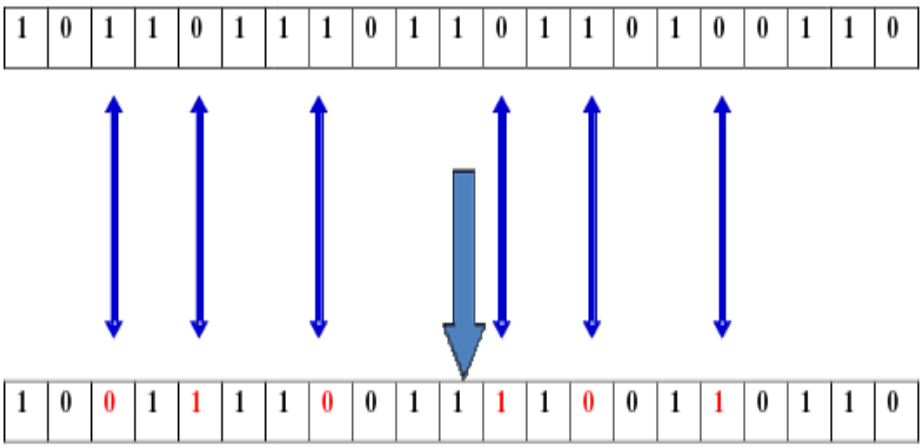


Figure 4.3: Mutation

4.1.1.6 Mutation

It is the process of random alteration in the genetic structure. It introduces genetic diversity into the population. It explores new search areas in the populations. Mutating a binary gene involves simple negation of the bit. It's a probabilistic operation. It occurs with very low probability (μ_m) in each generation. A random value is taken first and if it is seen that it is less than (μ_m) then mutation occurs for a single bit position in the parent chromosome. Otherwise the mutation does not occur for the particular bit position. Mutation on a single chromosome is shown in Fig. 4.3. Typically (μ_m) is between $\frac{1}{pop_{size}}$ and $\frac{1}{chromosome_{length}}$.

The following parameters are used for genetic algorithm for the proposed denoising technique.

- ◇ Population size (P) - variable. In this algorithm P is chosen in the range [5, 15].
- ◇ Chromosome length (L)- fixed (21 bits)
- ◇ Probabilities of crossover μ_c, and mutation μ_m. μ_c is kept high and μ_m is kept low. μ_c = [0.6-0.9] and μ_m = [0.01-0.1]
- ◇ The simulation scheme executed the program for [30, 50] generations on the various images along with different types of noise ratios.

4.2 Based on Particle Swarm Optimization

Particle swarm optimization [J. 1995, Eberhart 1995] based algorithm has also been applied for operator optimization. The noise removal operator has three parameters for which the optimization or maximization of PSNR (dB) is done. PSO [1] is similar to the Genetic Algorithm (GA) in the sense that these two evolutionary heuristics are population-based [Si] search methods. The drawback of the GA is the computational cost. In this technique, a basic binary encoded PSO with fixed length particle and a fixed length swarm of particles are employed to solve the problem. The PSO represents variables of each individual design with binary strings of 0's and 1's are referred to as the candidate solutions. In PSO, the algorithm maintains a population of particles, where each particle represents a potential solution of the optimization. Each particle is assigned a randomized velocity. The particles are then flown through the problem space. The aim of PSO is to find the particle position that results in the best evaluation of a given fitness function. Each particle keeps track of the following information in the problem space: x_i, the current position of the particle; v_i, the current velocity of the particle; and y_i, the personal best position of the particle which is the best position that it has achieved so far. This position yields the best fitness value for that particle. The fitness value of this position is called pBest. There is another parameter simulated by PSO, called global best (gBest). For gBest, the best particle is determined from the entire swarm. The best value tracked by the global version of the PSO is the overall best value (gBest), obtained so far by any particle in the population. The PSO changes the velocity of each particle at each time step so that it moves toward its personal best and global best locations. This is described in section 4.2.1. The fitness of each particle is considered as the PSNR (dB) value of the restored image by optimization of the noise removal operator. The PSNR (dB) values of each particle are updated or maximized in each generation of the PSO run. The PSO based optimization technique is presented in section 4.2.1.

4.2.1 The Proposed Technique

The particle swarm optimization pseudo code is given in Algorithm 10. The pseudo code is simplified for implementing the global version of PSO is as follows:

1. Initialize population of particles with random positions and velocities on a d-dimensional problem space. The three dimensional search space represented by I, T and R as parameters and are initialized as 1 to 10, 300 to 1000 and 0.6 to 0.95 respectively. Particles are initialized randomly at x_p in a fixed size of swarm. Here 'p' represents particle number in a swarm.

[1] http://www.mnemstudio.org/particle-swarm-introduction.htm

Algorithm 10 PSO pseudo code

```
1: begin
2: for each particle do
3:    Initialize particle
4: end for
5: for maximum iterations or minimum error criteria do
6:    Calculate Particle fitness value
7:    if the fitness value is better than pBest then
8:       Set pBest = current fitness value
9:    end if
10:   if pBest is better than gBest then
11:      Set gBest = pBest
12:   end if
13: end for
14: for each particle do
15:   Calculate particle's velocity
16:   Update particle's position
17: end for
18: end
```

2. For each particle, evaluate the desired optimization fitness function of d variables. As the maximum value of PSNR to be estimated using (4.3), same equation is used as fitness function f for the particles in PSO based optimization technique.

$$f(I_1, I_2) = PSNR(dB) = 10 * \log_{10}(\frac{255^2}{\frac{1}{M*N}\sum_{m,n}(I_1 m, n - I_2 m, n)^2}) \tag{4.3}$$

where M and N are the dimensions of the input images respectively. I_1 and I_2 are the original and enhanced images respectively. The detection of noisy pixels depends upon the threshold value T, which is decreased by a rate R and the finite numbers of iterations are required to obtain the optimum fitness value depending upon the parameter R and I respectively. The problem under consideration is to find the particles having the best fitness value (i.e., maximum PSNR) and that has been implemented. The fitness function given in the (4.3) is used for PSO.

3. Compare particle's fitness with its personal best value (pBest). If the current fitness is better than pBest, then the pBest value is set equal to the current value and the pBest location equal to the current location in the multidimensional space.

4. Compare fitness evaluation with overall previous best value of populations. If the

current value is better than the global best (gBest), set gBest to the current value of current particle and set the global best position to the position of current particle.

5. Change the velocity and position of the particle according to (4.4) and (4.5), respectively.

$$v_p(i+1) = h(i) * v_p(i) + \Psi_p * r_p * (x_{pbp}(i) - x_p(i)) + \Psi_g * r_g * ((x_{gbp}(i) - x_p(i)) \quad (4.4)$$

$$x_p(i+1) = x_p(i) + v_p(i+1) \quad (4.5)$$

Ψ_p and Ψ_g are the positive learning factors respectively. r_p and r_g are random numbers in [0, 1]. i is the generation number in [1, I_{MAX}]. I_{MAX} is the maximum number of generations. $h(i) \in [0, 1]$ is the inertia factor. $f_{pB}(i)$ and $f_{gB}(i)$ are the pBest value and gBest values at i^{th} generation, respectively. $x_{pB}(i)$ and $x_{gB}(i)$ are the personal and global best positions of p^{th} particle at i^{th} generation respectively.

6. Loop to step 2 until a termination criterion is met. The criterion is usually a sufficiently good fitness or a maximum number of iterations.

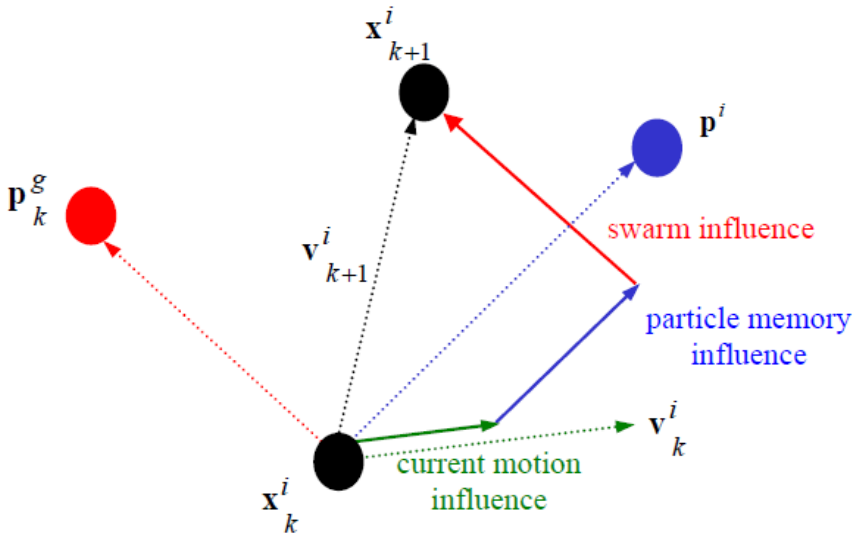


Figure 4.4: Change of velocity and position of particles

The position of each particle is updated using its velocity vector as given in (4.5) and depicted in Fig. 4.4. Three values which effect the new search direction, namely, *current motion*, particle *own memory*, and *swarm influence*, are incorporated via a summation approach as

shown in (4.4) with three weight factors, namely, inertia factor, h, self confidence factor, Ψ_p and swarm confidence factor Ψ_g respectively. The original PSO algorithm [J. 1995] uses the values of 1, 2 and 2 for h, Ψ_p and Ψ_g respectively, and suggests upper and lower bounds on these values as shown below.

- Inertia weight h $\in$ [0.4,1.4]
- Self confidence factor $\Psi_p \in$ [1.5,2]
- Swarm confidence factor $\Psi_g \in$ [2,2.5]

The research presented in this chapter reveals that setting the three weight factors h, Ψ_p and Ψ_g at 0.5, 1.5, and 1.5 respectively provides the best convergence rate for all test problems considered. For complete set of experiments following parameters are used:

- Number of particles =[15,20].
- Stopping criteria is considered as the convergence of particle global best value gBest.
- Learning factors Ψ_p =1.5 and Ψ_g= 1.5 for all images.
- Inertia factor h = 0.5
- The three constraints for which the optimization of the PSNR (dB) is done are having the following upper and lower limit

    1. The threshold value T $\in$ [300, 1000]
    2. Maximum numbers of iteration (I_{max}) $\in$ [1,10]
    3. Decreasing rate of Threshold (R) $\in$ [.06.,0.95]

The operator optimization techniques presented in this chapter are evaluated through results and comparisons with the state of the art methods in section 9.2 of chapter 9. In this case the digital images are corrupted with *random valued noise* and *salt and pepper noise*. The optimizations of the *Sub-Image Restoration (SIR)* filter are done using *Genetic Algorithm (GA)* and *Particle Swarm Optimization (PSO)* techniques. Three user parameters of the SIR filter are searched in a wide range by the two optimization algorithms. Best method of the noise suppression techniques i.e., *SIR* is optimized in the section by using GA and PSO. Other methods are not used for optimization here but they can also be optimized. The section contains results and comparisons on the algorithms such as *GASIR* and *PSOSIR* based filters. The *liftingbody*, *boat*, *lena*, *barbara* and *cameraman* images are used to evaluate

the proposed algorithms. Color *lena* image is also used to evaluate the proposed algorithms for noisy color image. Those images are corrupted using minimum (10%) to maximum noise density (100%) using the noise models. The results given by the proposed algorithms are presented by three popular and useful performance metrics such as Peak-Signal-Noise Ratio (PSNR(dB)), Image Enhancement Factor (IEF), Structure Similarity Index Measure (SSIM) and visual effects.

4.3 Discussions

The noise removal operators proposed in the previous chapter are optimized in this chapter. The noise removal operators have some parameters which required to be supplied during suppression of noises in digital images. Those parameters are heuristically supplied by the user and varied in a range to obtain the restoration results satisfactory. But those methods do not guarantee optimal restoration results. Those methods give semi-optimal or sub-optimal restoration results. Another factor is the different bench mark images have different characteristics in the image pixels. So there is a requirement of parameter searching and optimization during the program run independent to the benchmark images. Beside that when noise density is varied, the values of the parameter set need to be changed. For such reasons the parameters of the algorithms discussed in the previous chapter are searched in a wide range by two heuristic search algorithms in this chapter. Two recent stochastic randomized search algorithms are proposed for optimization the operators in this chapter. Genetic algorithm based optimization algorithm is proposed first because it can solve every optimization problem which can be described with the chromosome encoding. Whereas *Particle Swarm Optimization* is a relatively recent heuristic search method whose mechanics are inspired by the swarming or collaborative behavior of biological populations. Both the algorithms are examined and investigated to optimize the fitness function of the noise removal operators depending upon the constraints viz., number of iterations (I), threshold value (T) and decreasing rate of threshold value in each of the iteration (R) of the noise detection and filtering techniques. Maximization of PSNR (dB) of the restored image is the fitness function of the proposed optimization algorithms.

Elitism model of genetic algorithm based optimization algorithm is proposed first for the optimization algorithm. The operators of the genetic algorithm such as encoding, fitness calculation, crossover and mutation are done properly to get the global optimal solutions. Chromosome encoding is done in terms of binary strings. Binary tournament selection is used as the fitness calculation. Uniform crossover method is presented to make the mating pool from the parent chromosomes. The crossover method is done on the basis of high crossover probability. The mutation is done lastly with each chromosome with low probability.

Due to high computational cost of the genetic algorithm, the particle swarm optimization

is also used for optimization of the operators in this chapter. Besides that, the ease of implementation and faster convergence rate of the PSO algorithm has made it a better algorithm compared to GA. Fixed length particles are encoded during the optimization method by the PSO algorithm.

Chapter 5

Noise Density Estimation for Adaptive Median Filter

Contents

5.1 Proposed Technique for Noise Detection 108
5.2 Proposed Technique for Noise Density Estimation 108
5.3 Proposed Technique for Adaptive Median Filter 111
5.4 Illustration and Examples 111
5.5 Discussions 115

In this chapter, the density of noises in the digital image is estimated prior to denoising. Depending upon the noise density, an empirical window size is suggested for restoration of the noisy pixels. To estimate the noise percentage on the digital image, pixel clustering is done in the proposed algorithm. In this technique, one of the important task is to cluster the pixels in the images into homogeneous regions, each of which corresponds to some intensity space type. The problem of pixel clustering is often modeled as clustering in the intensity space [Bandyopadhyay 2001, Maulik 2003]. Clustering [Everitt 1993, Jain 1988, Moulik 2002, Tou 1974] means the method of grouping samples so that the samples belong to same group are similar. The groups are called cluster [Jain 1999, Tan 2006, Gose 1996]. It is an unsupervised pattern classification tool in which a given data set of n objects is partitioned into K distinct groups based on some similarity or dissimilarity metric where the value of K may or may not be *a priori*. The proposed clustering based noise removal algorithm consists of three main steps: noise detection, noise density estimation and noise filtering.

The adaptive median filtering has been applied widely as an advanced method compared with standard median filtering. The adaptive median filter performs spatial processing to determine which pixels in an image have been affected by impulse noise. The adaptive median filter classifies pixels as noisy by comparing each pixel in the image to its surrounding neighbor pixels. The size of the neighborhood is adjustable, as well as the threshold for the comparison. A pixel that is different from a majority of its neighbors, as well as being not

structurally aligned with those pixels to which it is similar, is labeled as impulse noise. These noisy pixels are then replaced by the median pixel value of the pixels in the neighborhood that have passed the noise labeling test. Adaptive median filter changes the size of S_{xy} (the size of the neighborhood) during operation. The size of the window can be varied depending upon the density of the image automatically during the program run. Density of noises in the image is first estimated. Then an empirical window size is suggested in the program. On the suggested test window of neighbors, noise detection and filtering rules are applied. A popular clustering technique is used for clustering of pixels in the proposed chapter for noise density estimation.

The noise density estimation technique based on clustering of image pixels is given in section 5.2. The noise detection and adaptive median filtering rules are discussed in section 5.1 and section 5.3, respectively. The discussions of the issue has been done in section 5.5.

5.1 Proposed Technique for Noise Detection

The proposed impulse detection scheme is based on the clustering results and on some simple arithmetic comparisons. Considering that two pixels whose spatial distance is small, their gray level values should also be closed. Accordingly their cluster level vectors would be similar. A 5×5 region which belong to same cluster must be noise free flat region. The pixels present on an edge should also belong to same cluster. An edge is detected using Laplace operator [1]. If the pixels on an edge belong to dissimilar clusters then one or more pixels on that particular direction are corrupted pixel(s). Corrupted pixels on each direction are determined by counting the belongingness of the pixels to the cluster centers where the test pixel $y_{i,j}$ does not belong to. The remaining eight pixels in the 5×5 window are classified by comparing the intensity value of each pixel with the average intensity values of the surrounded directional pixels of the pixel under question. The noise detection technique is given in Algorithm 11.

5.2 Proposed Technique for Noise Density Estimation

The total number of corrupted pixels in the four directions within a 5×5 window are determined by ensuring the pixels over the edges excluding $y_{i,j}$, belong to the cluster center where the center pixel does not belong. The remaining eight pixels are detected as noisy by ensuring that the intensity values of these pixels do not lie in between the average intensity values of the surrounded directional pixels. The proposed noise density estimation technique is given in Algorithm 12.

[1] http://mathworld.wolfram.com/Laplacian.html

Algorithm 11 Noise detection technique

1: Make clustering on the training image into K number of clusters by K-Means algorithm
2: Select a pixel $y_{i,j}$ and it's 5×5 neighborhood pixels around $y_{i,j}$
3: If all the 24 pixel along with the test pixel belong to same cluster then it is noise free flat region. Proceed to the next pixel i.e., $y_{i,j+3}$ in row major order
4: Otherwise there may exist edge(s). It is also known that the pixels present on any edge must belong to same cluster because of their intensity value similarities
5: Among four directional pixels around $y_{i,j}$, shown in Fig. 3.1, the edge(s) containing the directional pixels which return *Laplacian* value equal to zero is considered is/are edge(s)
6: If any edge found within the test window and the corresponding pixels present on the particular direction (edge) belong to same cluster then the test pixel along with all other pixels on that direction are noise free. Proceed to the next pixel i.e., $y_{i,j+1}$ in row major order.
7: If no edge found within the test window then determine whether the test pixel is noisy or not.
8: To classify the center pixel, it finds the minimum and maximum gray values of the 5×5 test window around it. If the intensity value of the center pixel does not lie within the intensity range spread of its neighbors, it is declared as noisy. Otherwise it is noise less. For example let $y_{i,j}$ be the center pixel and W_{min} and W_{max} are the maximum and minimum intensity values respectively within the test window around $y_{i,j}$. The detection rule is given in (5.1).

$$y_{i,j} = \begin{cases} Noise\ Free & : W_{min} < y_{i,j} < W_{max} \\ Noisy & : Otherwise \end{cases} \tag{5.1}$$

If $y_{i,j}$ is detected as noise free then proceed to the next pixel in row major order.

$$ND = \frac{N_c}{25} * 100 \tag{5.2}$$

where N_c is the number of corrupted pixels within the test window.

$$a_{i,j}^{(k)} = \frac{\sum_{(s,t) \in S_k^0} y_{i+s,j+t}}{4}, 1 \leq k \leq 4 \tag{5.3}$$

Algorithm 12 Noise density estimation technique

1: If $y_{i,j}$ is detected as noisy then estimate noise density (ND) within the window.
2: Let $y_{i,j} \in$ cluster C_z. Then any pixel in the four main directions apart from $y_{i,j}$, is detected as noisy which does belong to C_z. Count the number of pixels in the four main directions which have that property.
3: Detect whether the remaining pixels (eight pixels), which are not present on any direction are noisy or not. Let the set of four pixels centered at (0, 0) in the k^{th} direction denoted by S_k, (k=1 to 4), i.e.,
   1. S_1={(-2,-2),(-1,-1),(1,1),(2,2)}
   2. S_2={(0,-2),(0,-1),(0,1),(0,2)}
   3. S_3={(2,-2),(1,-1),(-1,1),(-2,2)}
   4. S_4={(-2,0),(-1,0),(1,0),(2,0)}

   The remaining eight pixels are as follows
   1. S_5={(-2,-1),(-2,1)}
   2. S_6={(-1,-2),(-1,2)}
   3. S_7={(1,-2),(1,2)}
   4. S_8={(2,-2),(2,1)}
4: In each direction of a 5 × 5 window, define $a_{i,j}^{(k)}$ as the average of intensity values of S_k^0 (k= 1 to 4), given in (5.3).
5: Make comparisons as follows to classify the eight remaining pixel
   1. if $y_{\pm(i+2,j+1)}$ lies between $a_{i,j}^{(1)}$ and $a_{i,j}^{(4)}$, then it/these is/are noise free.
   2. if $y_{\pm(i+1,j+2)}$ lies between $a_{i,j}^{(1)}$ and $a_{i,j}^{(2)}$, then it/these is/are noise free.
   3. if $y_{\pm(i+1,j-2)}$ lies between $a_{i,j}^{(2)}$ and $a_{i,j}^{(3)}$, then it/these is/are noise free.
   4. if $y_{\pm(i+2,j-1)}$ lies between $a_{i,j}^{(3)}$ and $a_{i,j}^{(4)}$, then it/these is/are noise free.
6: Define Noise Density (ND) within the 5 × 5 test window given in (5.2).
7: Empirically suggest a filtering window ($W_{d1} \times W_{d2}$) given in table 5.1.
8: Let D_k (k=1 to 4) denote the set of coordinates aligned in the four directions centered at (0, 0) in the filtering window ($W_{d1} \times W_{d2}$), i.e.,
   1. D_1= {(-y,-y),...,(-1,-1), (0,0), (1,1),...,(y,y)}
   2. D_2= {(0,-y),...,(0,-1), (0,0), (0,1),...,(0,y)}
   3. D_3= {(y,-y),...,(1,-1), (0,0), (-1,1),...,(-y,y)}
   4. D_4= {(-y,0),...,(-1,0), (0,0), (1,0),...,(y,0)}

Table 5.1: Suggested filtering window size against noise density (ND)

Noise Density (ND)	Window Size ($W_{d1} \times W_{d2}$)
$1 \leq ND \leq 30$	3×3
$31 \leq ND \leq 60$	5×5
$61 \leq ND \leq 80$	7×7
$81 \leq ND \leq 100$	9×9

5.3 Proposed Technique for Adaptive Median Filter

The adaptive median filtering scheme uses a simple rule to find the most significant pixels within the selected filtering window to restore the noisy pixel. Standard deviations are done on each directional pixel and from which the directions are selected where standard deviations are minimum, maximum and not any of them. In order to keep the edges intact within the test window, the corresponding direction where the standard deviation is minimum, is given extra weight. Except the center pixel all other pixels over that direction and some pixels within 3×3 window are used for median filtering. Within the 3×3 window some pixels are given extra weight during median filtering. The adaptive median filtering rule is given in Algorithm 13.

5.4 Illustration and Examples

To identify the noisy pixels within the 5×5 window, the clustering of the pixels of the image plays the most important role. Considering that two pixels whose spatial distance is small, their gray level values should also be close. Three facts are considered to detect the noisy pixels and to estimate the noise density within the 5×5 window. When any pixel is on a noise free flat region then the probability outcomes of belongingness of those pixels to the same cluster is high. Next is if any pixel exists on the edge then at least one of the directional pixels will belong to same cluster. Finally any noisy pixel in a noise free region will belong to a dissimilar cluster where the noise free pixels belong. These all facts have very high probability of occurrence.

Noise estimation is done by counting the noisy pixels within the 5×5 window. High noise percentage involves less number of noise free pixels within the window initially used. Low percentage of noise density within the window involves more number of noise free pixels. The proposed weighted median filter replaces the noisy pixel by the median value, computed of some particular pixels within the filtering window. To restore a noisy pixel in a high noise

Algorithm 13 Impulse filter

1: Calculate the standard deviations $\sigma_{i,j}^{(k)}$ of the intensity values of all $y_{i+s,j+t}$ within (s,t) $\in D_{i,j}^{(k)}$ (k= 1 to 4) with $(W_{d1} \times W_{d2})$.

2: Find the minimum of $\sigma_{i,j}^{(k)}$, where k= 1 to 4, as

$$l_{i,j} = \min_k \{\sigma_{i,j}^{(k)} : k = 1 \; to \; 4 \tag{5.4}$$

and let the corresponding direction is l.

3: Find the maximum of $\sigma_{i,j}^{(k)}$, where k= 1 to 4, as

$$m_{i,j} = \max_k \{\sigma_{i,j}^{(k)} : k = 1 \; to \; 4\} \tag{5.5}$$

and let the corresponding direction is m.

4: Select the directions where standard deviations are maximum, minimum and also not any of them. Use repetition operator $\diamond$ [Brownrigg 1984] for which standard deviation is minimum.

5: Form a set containing the all the pixels at direction l, except the center pixel in the $W_{d1} \times W_{d2}$ window and also some pixels from 3×3 window.

6: Calculate the median using (5.6) with assigning $\omega_m = 0, \omega_l = 1$ and $\omega_n = 1$.

$$med = median\{\omega_{s,t} \diamond y_{i+s,j+t} : (s,t) \in \; \Omega^4 \; and \; y_{i+p,j+q} : (p,q) \in \Omega^5\} \tag{5.6}$$

where

$$\Omega^4 = \{(s,t) : -1 \leq s,t \leq 1\} \; and \; (s,t) \neq (0,0). \tag{5.7}$$

where

$$\Omega^5 = \{(p,q) : (p,q) \neq (0,0) \; and \; where \; y_{i+p,j+q} \in l \; direction \; within \; W_{d1} \times W_{d2}\}. \tag{5.8}$$

and where

$$\omega_{s,t} = \begin{cases} \omega_m & : (s,t) \in s^0_{m_{i,j}} \\ \omega_l & : (s,t) \in s^0_{l_{i,j}} \\ \omega_n & : \text{otherwise} \end{cases} \tag{5.9}$$

7: Replace $y_{i,j}$ by med.

density region within a low size window may grab the noisy pixels prior to calculate the median. Eventually there may be a high risk that a noisy pixel is replaced by a noisy pixel again during computing of the median of the pixels in a shorter window. Larger size window

size eliminates this risk to some extend by involving number of pixels so that the probability of the participation of less number of noisy pixels for calculating the median will be high.

Salt and pepper and random valued impulse noise models drastically alters the image pixels to an intensity value that is either 0 or 255 or any value within that range. With unequal probability there are two other noise models as well. The center pixel is compared with the maximum and minimum gray level intensity values spread of its neighbor within the 5 × 5 window to classify it. The probability of the event that a noise free pixel will belong to that range is very high. That is why such clustering or detection rule is proposed for the center pixel within the window.

The proposed filtering rule is based on selection of the direction where the variance is least among the four directions. Since the variance describes how tightly all the values are clustered around the mean in the set of pixels. The directional pixels whose variance is minimum shows the direction where each pixels are closest to each other. Therefore the center pixel should also be close to them in order to keep the edges even thin lines intact. The directional pixels whose variance is minimum are given more importance than the directional pixels whose variance is maximum. The weighted median filter replaces the noisy pixel by a pixel value which is most suitable within the filtering window.

It is considered as a n × m matrix consisting of set of elements/pixels S = a_{11} ... a_{nm} = $\sum\sum a_{ij}$. The pixels are grouped into K number of clusters each consisting of variable number of elements where values of K varies from 1 to K and number of elements varies from 0 to n × m (maximum). As the property of K-Means is that it uses similarity measurement, the mapping of elements/pixels will be based on similarity index. In clustering the original set is divided into K number of sub sets S_1, S_2, ... S_k each consisting of variable number of elements, similar pixels moves to same cluster in the process of clustering where spatial distance of elements of the set are minimum as similarity index is maximum. A group is called noise free region in a test window if the pixels move to same subset. In other words if the pixels are not grouped to the same cluster in a specific test window then there exist edges/impulses. Now probability that corrupted pixels may coagulate to same region then noise density will be either maximum or minimum depending on existence of edges during detection. Hence methodology of applying detection technique will results in detecting maximum number of pixels within a cluster and probability of altering uncorrupted pixels is minimum. As a result improvement of the technique is maximum. Again size of the window selection is made according to the density of noise which leads to better correction. Again there is no dependency of the model on benchmark images as consistent PSNR and IEF are obtained from experimental results in all benchmark images.

The noise density estimation for denoising technique presented in this chapter are evaluated through results and comparisons with the state of the art methods in section 9.3 of chapter 9. In this case the digital images are corrupted with *random valued noises* and *salt and pepper noise*. The *liftingbody*, *boat*, *lena*, *barbara* and *cameraman* images are used to evaluate the proposed algorithm. Color *lena* image is also used to evaluate the proposed algorithm for noisy color image. Those images are corrupted using minimum (10%) to maximum noise density (100%) using the noise models. The results given by the proposed algorithm are presented by three popular and useful performance metrics such as Peak-Signal-Noise Ratio (PSNR(dB)), Image Enhancement Factor (IEF), Structure Similarity Index Measure (SSIM) and visual effects.

To visualize the algorithm, a 5×5 sub image is used as an example for illustrating the proposed impulse detection, density estimation and filtering schemes as follows:

Case 1:

$$\text{Let } W_1 = \begin{pmatrix} \mathbf{210} & 150 & 61 & 200 & 155 \\ 90 & \mathbf{215} & 30 & 61 & 95 \\ 85 & 200 & \mathbf{205} & 210 & 170 \\ 96 & 195 & 200 & \mathbf{210} & 190 \\ 170 & 93 & 254 & 95 & \mathbf{215} \end{pmatrix}$$

1. All the 25 pixels within the 5×5 window of W do not belong to same cluster, so there is/are noisy pixel(s).

2. In the four directions the edge(s) is/are detected by Laplace operator. Here only direction 1 is an edge with that property. Say all the pixels on direction 1 i.e., 210, 215, 205, 210 and 215 belong to cluster C_1, then these pixels are detected as noise free. No filtering is needed. The next pixel is processed.

Case 2:

$$\text{Let } W_2 = \begin{pmatrix} 116 & 122 & 146 & 117 & 135 \\ 125 & 110 & 63 & 46 & 72 \\ 47 & 140 & \mathbf{173} & 132 & 124 \\ 64 & 137 & 114 & 41 & 153 \\ 24 & 132 & 150 & 23 & 44 \end{pmatrix}$$

1. Here no edge is found by the Laplace operator. So there is a need to classify the center pixel, i.e., 173.

2. Let the center pixel 173 belong to cluster C_1 and this is a noisy pixel because of $23 < 173 \geq 153$. So there is a need to estimate the Noise Density (ND) within W to select filtering window for median filtering. Say, among the main four directional pixels

within W, pixels 150 and 146 also belong to the cluster C_1. So these are also corrupted pixels.

3. The status of the remaining eight pixels i.e., 122, 117, 125, 72, 64, 153, 132 and 23 are determined by comparing the intensity value of that pixel with average intensity values of the surrounded directional pixels. Center pixel is not considered for averaging as this is a noisy pixel. Here the average intensity values within directions 1 and 2 are 78 and 111. As 125 and 153 do not lie between 78 and 111, so these are corrupted pixels.

4. Similarly all other pixels are classified i.e., 117, 125, 72, 64, 153, 132 and 23. In this way the noise density is estimated by counting the total number of corrupted pixels within W, to select corresponding filtering window. Here noise density is 28%. So the 3×3 window is selected for median filtering.

5. The direction 2 has minimum standard deviation i.e., 21. So the median of the set is calculated

   S = $\{140, 132, 140, 132, 110, 41, 63, 114\}$, i.e., 123.

6. Replace 173 by 123.

5.5 Discussions

In this chapter, a clustering based decision algorithm has been proposed for restoration of digital images which were contaminated by any level of noises. It is an improved switching median filter based on k-means in which the filtering is applied to only the corrupted pixels in the training image while the uncorrupted pixels are left unchanged. Two type noise models are used to train different types of bench mark images. In each iteration, clustering of the image pixels are done and based on the clustering indices, the noise detection is performed. If any noisy pixel found, noise density is estimated to suggest one empirical filtering window. Filtering is done through median operation of the neighborhood pixels by calculating variance of the four directional pixels. Certain pixels of the neighborhood are given extra importance and taken more than single time. 3×3, 5×5, 7×7 and 9×9 filtering window regions are used for such purpose. The performance of the proposed operator has been conformed using several benchmark images which reveals that it produces better results in terms of quantitative measures such as PSNR, SSIM and IEF. From the qualitative restoration results it is seen that the technique preserves image fine details and textures well.

Chapter 6

PSO based Clustering for Noise Detection and Restoration

Contents

6.1 Based on Fixed length Particles 118

6.1.1 Proposed Technique for Fixed length PSO based Clustering 119

6.2 Based on Different length Particles 123

6.2.1 Proposed Technique for Different length PSO based Clustering 124

6.3 Proposed Technique for Noise Detection and Restoration 126

6.4 Discussions 129

Image pixel clustering can be done for image denoising. The pixels containing similar pixel values can be grouped together to make the clusters and then the information can be utilized for the detection and filtering of the noisy pixels in a digital image. The performance of the noise detection rule and restoration of noisy pixel are completely powered by the clustering algorithm. How beautifully the clustering of pixels are done determines the performance of the noise detection and noise restoration performances.

In this chapter, two clustering algorithms are proposed based on *Particle Swarm Optimization* algorithm. The first algorithm partitions the images into homogeneous regions based on a powerful optimization function which considers three evaluation criteria. The particles which are potential solutions of the optimization algorithm are kept fixed length. In the second case, the particles are of different lengths. In case of fixed length particles, the parameter, number of cluster center is supplied by the user. The shortcoming is solved in the second algorithm where the particles have different lengths. In that case there is no need to supply the parameter, number of cluster center to the program, rather the program automatically determine the particle length on the basis of randomized encoding of number of cluster centers for the clustering algorithm.

In this chapter, a new distance function is proposed for obtaining dissimilarity between two pixels. In case of image processing, the spatial distance is a factor when determining their similarity/dissimilarity during image clustering. The spatial distance is given a relevant factor during the operation. The gray level distance of two pixels is the principle similarity/dissimilarity measure which is done for generating the clusters. The weighted *Euclidean* distance function proposed here, considers both the distance measures while determining the similarity/dissimilarity of the image pixels. By using two coordinates along with the pixel intensity value, the distance function is constructed.

Rest of the chapter describes the detail proposals of *PSO based clustering* towards denoising. The fixed length particles along with the algorithm is described in section 6.1. The different length particle based optimization algorithm is presented in section 6.2. The proposed noise detection and restoration rules are presented in section 6.3. The chapter summary has been given in section 6.4.

6.1 Based on Fixed length Particles

In this chapter, a de-noising method is proposed where the detection and filtering is based on clustering of image pixels. The noisy image is grouped into subsets of pixels with respect to their intensity values and spatial distances. Using a novel fitness function the image pixels are classified using Particle Swarm Optimization (PSO) technique. The distance function measures similarity/dissimilarity among pixels using not only the intensity values but also the positions of the pixels. The detection technique enforced the *PSO* based clustering technique, which is very simple and robust. Filtering operator restored only the noisy pixels keeping noise free pixels intact.

There exist several algorithms for denoising based on clustering algorithms such as [L.B. 2009], some algorithms based on gray-level clustering algorithm for image segmentation [Kavitha 2010] proposed in the literature. Here a novel technique is proposed to detect and suppress two types of noises in the digital images for a varying noise density from 10% to 100%. In PSO based clustering, the design of good fitness functions for PSO is important to ensure the quality of clustering. The proposed operator detects and restores the noisy pixels in a test image where the noise detection operator depends on the PSO based clustering algorithm completely. A novel distance function in conjunction with the conventional Euclidean distance function has been proposed for PSO based pixel clustering which can provide more compact clusters and larger separation between the cluster centers compared to K-means clustering. After the image clustering the noise detection and restoration stages work. A 3×3 window based noise detection and then an efficient filtering method has been proposed to restore the noisy pixels. Some user parameters of the clustering algorithm have

been empirically determined.

6.1.1 Proposed Technique for Fixed length PSO based Clustering

Fixed length particles are encoded first in this case which contains the randomized points in the image which describes the cluster centers. In this case, the number of cluster centers is supplied by the user and then the proposed algorithm encodes swarm particles. Particle length remains fixed for each particle. All the data points in the image pixels are then applied to the proposed *Euclidean* distance function to find the distance between the cluster centers and each data point. The data point assigned to the cluster center which has closest distance to the data point. The three evaluation criteria such as *inter cluster distance*, *intra cluster distance* and *quantization error function* are used as similarity or dissimilarity measures in the proposed algorithm. The inter cluster distance is maximized because this distance is referred to as the inter cluster separation distance between two cluster centers. The intra cluster distance is minimized during the clustering algorithm because this implies the distance of center point to all other data points in a cluster. The weighted quantization error function is also minimized to contain the maximum pixels in the clusters having minimum intra cluster distances and maximum inter cluster distances.

The most relevant parameter of the clustering algorithm is the number of cluster centers which is supplied during the program execution. The value of the parameter is varied in a range to get best possible clustering results which eventually produces good denoising results.

6.1.1.1 Particle Swarm Optimization

In PSO, the algorithm maintains a population of particles, where each particle represents a potential solution of the optimization. Each particle is assigned a randomized velocity. The aim of PSO is to find the particle position that results in the best evaluation of a given fitness function. Each particle keeps track of the following information in the problem space: x_i, the current position of the particle; v_i, the current velocity of the particle; and y_i, the personal best position of the particle which is the best position that it has achieved so far. This position yields the best fitness value for each particle. The fitness value of this position is called pBest. There is another parameter simulated by PSO, called global best (gBest). For gBest, the best particle is determined from the entire swarm. The best value tracked by the global version of the PSO is the overall best value (gBest), obtained so far by any particle in the population.

The PSO changes the velocity of each particle at each step so that it moves toward personal best and global best locations. The algorithm for implementing the global version of PSO is as follows:

1. Initialize population of particles with random positions and velocities on a d-dimensional problem space.

2. For each particle, evaluate the desired fitness value.

3. Compare particle's fitness with its personal best value *(pBest)*. If the current fitness is better than pBest, then the pBest value is set equals to the current value and the pBest location equal to the current location.

4. Compare fitness evaluation with overall previous best value of population. If the current value is better than the global best *(gBest)*, set gBest to the current value of current particle and set the global best position to the position of current particle.

5. Change the velocity and position of the particle according to (6.1) and (6.2), respectively.

$$v_p(i+1) = h(i) * v_p(i) + \Psi_p * r_p * (x_{pbp}(i) - x_p(i)) + \Psi_g * r_g * ((x_{gbp}(i) - x_p(i)) \quad (6.1)$$

$$x_p(i+1) = x_p(i) + v_p(i+1) \quad (6.2)$$

Ψ_p and Ψ_g are the positive learning factors respectively. r_p and r_g are random numbers in [0, 1]. i is the generation number in [1, I_{MAX}]. I_{MAX} is the maximum number of generations. *h(i)* $\in$ [0,1] is the inertia factor. $f_{pB}(i)$ and $f_{gB}(i)$ are the pBest value and gBest values at i^{th} generation, respectively. $x_{pB}(i)$ and $x_{gB}(i)$ are the personal and global best positions of p^{th} particle at i^{th} generation respectively.

6. Loop to step 2 until a termination criterion is met. The criterion is usually a sufficiently good fitness or a maximum number of iterations.

6.1.1.2 PSO based Clustering of Pixels

In this technique, a PSO based clustering method has been proposed prior to image denoising. To describe PSO based image clustering method various notations are used, for different attributes. N_p, N_c, z_p, m_j, C_j and $|C_j|$ denote the number of image pixels to be clustered, total number of cluster centers, p-th pixel of the image, mean or center of cluster j, set of pixels of cluster j and the number of pixels in cluster j respectively. In PSO-based image clustering, a single particle represents the N_c cluster centers. Each particle x_i is constructed as $x_i = \{m_{i1}, ...,m_{ij}, ..., m_{iN_c}\}$ where m_{ij} refers to the j-th cluster centroid vector of the i-th

particle. The Euclidean distance function is proposed to find the distance and/or similarity between the pixels in the image. Vector containing the X-coordinate (say x), Y-coordinate (say y) and pixel intensity value (say z) are used to obtain the Euclidean distance between each pair of pixels. The distance between each pair of pixels termed as gray level distance as well as their spatial distance is calculated in the proposed distance function. The proposed Euclidean distance function between the i-th pixel and j-th pixel in the image matrix is given in (6.3).

$$d(x, y, z) = \sqrt{w_1((x_i - x_j)^2 + (y_i - y_j)^2) + w_2(z_i - z_j)^2} \tag{6.3}$$

In the Euclidean distance function, two different weights are assigned to two different distances, the spatial distance and gray level distance. The spatial distance has been assigned with less weight over the gray level distance. The intensity value similarity/dissimilarity of any two pixels is given with more weight over the positional similarity/dissimilarity of the pixels. The intensity value difference between the pixels are multiplied by a weight factor w_2=0.5 whereas coordinate level difference is multiplied by w_1=0.1.

The PSO-based algorithm for clustering is given in Algorithm 14. The fitness function

Algorithm 14 PSO based clustering

Input: Gray Scale Image Matrix
Output: Partition matrix with N_c number of clusters
Symbols: t_{max} = Maximum number of iterations
d(z_p, m_{ij}) = Euclidean distance given in (6.3) between the p-th pixel z_p and the centroid of j-th cluster of particle i.
Z = a matrix of pixels assigned to clusters of particle i
Begin
Initialize each particle to contain N_c randomly selected cluster centers
for $t = 1$ to t_{max} **do**
  for all particle $\in$ i **do**
    for all pixel $\in z_p$ **do**
      Calculate d(z_p, m_{ij}) for all clusters C_{ij}
        d(z_p, m_{ij})=min$_{\forall c=1,\ldots,N_c}${d($z_p$,$m_{ic}$)}
      Assign z_p to C_{ij}
      Calculate the fitness function f(x_i(t),Z)
  Update the personal best and global best positions
  Update the cluster centers using (6.1) and (6.2)
End

proposed by Omran and Salman [Omran 2005] has been associated in the proposed clustering

based on three evaluation criteria: inter cluster distance, intra cluster distance and weighted quantization error.

Inter cluster separation distances for all clusters are measured and the minimum distance between any two clusters is calculated using (6.4). A large value of d_{min} means that the clusters are well separated.

$$d_{min}(x_i) = \min_{\forall j1,j2,j1\neq æ2} \{d(m_{ij_1}, m_{ij_2})\} \tag{6.4}$$

The intra cluster distances of all the clusters are measured and the maximum one among all the clusters is selected in d_{max}, which is defined in (6.5). Here Z is a matrix representing the assignment of pixels to clusters of particle i. A smaller value of d_{max} means that the clusters are more compact.

$$d_{max}(Z, x_i) = \max_{j=1 \ to \ N_c} \{ \sum_{\forall z_p \in C_{ij}} d(z_p, m_{ij}) / \ |C_{ij}| \} \tag{6.5}$$

Recently Esmin et al [Esmin 2008] used the quantization error function in the clustering algorithm of image pixels which calculates the average distance of the pixels of a cluster to its cluster center, followed by the average distances of all clusters and hence calculates new average. The problem of Esmin et al [Esmin 2008] is that any cluster with one pixel would affect the final result with another cluster containing many pixels. Suppose i-th particle in a cluster which has only one pixel and very close to the center and there is another cluster that has many pixels which are not so close to the centroid. The problem has been resolved by assigning the less weight to the cluster containing only a single pixel with another cluster containing many pixels. The weighted quantization error function proposed by Esmin et al. is given in (6.6).

$$Q_e = \{ \sum_{\forall j=1 \ to \ N_c} [(\sum_{\forall z_p \in c_{ij}} d(z_p, m_{ij}) / \ |C_{ij}| . (N_0 / \ |C_{ij}|)] \} \tag{6.6}$$

where N_0 is the total number of data points to be clustered. The fitness function is constructed by intra cluster distance d_{max}, inter cluster distance d_{min} along with the quantization error Q_e function. The fitness function used to minimize $f(x_i, Z)$ [Wong 2011] which is given in (6.7). Here z_{max} is the maximum intensity value of the digital images which is 255 for eight bit gray scale images. In the optimization function, equal weights are assigned to the three distance functions. The fitness function is given to the PSO based optimization technique and which minimizes the value of f in each generation to make the image well clustered.

$$f(x_i, Z) = d_{max}(z, x_i) + (z_{max} - d_{min}(x_i)) + Q_e \tag{6.7}$$

The performance of the proposed filter has been evaluated quantitatively and qualitatively through simulation and analysis. Eight bit gray scale bench mark images Boat, Lena, Cameraman, Liftingbody and Barbara are used to evaluate the performance of the proposed algorithm. The fitness function given in the (6.7) is used for PSO-based clustering. For complete set of experiments following parameters are used:

- Number of particles for Cameraman image = [15, 20].
- Number of particles for Boat, Liftingbody, Barbara and Lena images = [10, 15].
- Maximum number of generations (I_{max}) for Boat, Liftingbody, Barbara and Lena=[5,8].
- Maximum number of generations (I_{max}) for Cameraman =[10,12].
- Number of cluster centers = [15, 25].
- Learning factors Ψ_p and Ψ_g =2 for all images clustered.
- Inertia factor h(i) $\in$ [0,1].

Maximum number of particles chosen for the simulation of Cameraman image is 20, because this image is of size up 256 $\times$ 256. The number of particles chosen for Boat, Liftingbody, Barbara and Lena images is 15, because of the image matrix is of size 512 $\times$ 512. Less number of particles has been chosen because of time complexity and during real life applications the size may vary as per convenience of the user. The maximum number of iterations (I_{max}) is chosen in the range of 10 to 12 in the proposed PSO-based noise removal algorithm. The algorithm evaluates a maximum of 240 fitness functions because it executes the 20 particles for 12 iterations and then the restoration result is generated. The maximum number of cluster centers used for the PSO based algorithm is 15 to 25. Two important user parameters such as the number of iterations and number of cluster centers are empirically determined through the observations and experiments of the restoration results. The value of the cluster center is varied in the specified range and the best results are taken. The inertia factor h (i) is initialized as 1 and it decreases to 0.1 when it reached final iteration. The inertia factor has been minimized in each iteration to obtain good global search ability at the beginning of the iteration and eventually to obtain good local search.

6.2 Based on Different length Particles

In any fixed length clustering algorithm for example in the K-means algorithm, the clustering is obtained by iteratively minimizing a fitness function that is dependent on the distance of the pixels to the cluster centroids. However, the K-means algorithm like most of the existing

clustering algorithms assume a priori knowledge of the number of centers, K, while in many practical situations, this information cannot be determine in advance. Also it is sensitive to the selection of the initial cluster centers and may converge to the local optima.

A lot of approaches [Katari 2007, Srikanth 1995, Qiu 2009, Javier 2010, Pakhira 2005, Pakhira 2004, Maulik 2003] towards the image clustering based on variable length chromosome based genetic algorithm and variable length particle swarm optimization have been done in the recent past years. When the pixels are grouped the noises can be detected easily. In section 6.2.1, a new approach called different length particle swarm optimization (DPSO) is proposed. In DPSO algorithm, a swarm of particles of different length, automatically determines the number of clusters and simultaneously clusters the data set with minimal user interference. DPSO starts by partitioning the image into different partitions encoded in each particle of the swarm. By using the same fitness function as discussed in (6.7) which contains three evaluation criteria such as inter cluster distance given in (6.4), intra cluster distance (6.5) and the quantization error function (6.6). The weighted *Euclidean* distance function given in (6.3) is used for distance function between two pixels in the image matrix. The algorithm terminates when the *gBest* converges to optimal solution or for a finite number of iterations. After the clustering algorithm executes, the noise removal operator proposed in section 6.3 in Algorithm 16 starts executing to remove the noises while preserving the image fine textures.

6.2.1 Proposed Technique for Different length PSO based Clustering

The novel different length particle swarm optimization algorithm for image pixel clustering is proposed. It is an automatic image clustering algorithm which is a modified version of general particle swarm optimization algorithm. The proposed algorithm can make the clusters being well separated from each other, well compact and quantization error is minimized compared to the fixed length particle swarm optimization algorithm without having any knowledge of number of clusters. The algorithm operates on a swarm of particles having different number of cluster centers, which are encoded randomly during the operation.

In this section, an automatic version of PSO called Different length PSO (DPSO) is proposed which is a minimal interactive algorithm is compared to fixed length PSO (FPSO). Let Z = (z_1, z_2, z_3... z_n) be the digital image with n number of pixels. The DPSO maintains a swarm of particles, where each particle represents a potential solution to the clustering problem. Each particle encodes an entire partition of the image Z. DPSO tries to find an optimal partition of N_c number of compact and well separated clusters. The DPSO based

image clustering method has various parameters, for different attributes. N_p, N_c, z_p, m_j, C_j and $|C_j|$ denotes the denotes the number of image pixels to be clustered, total number of cluster centers, p-th pixel of the image, mean or center of cluster j, set of pixels of cluster j and the number of pixels in cluster j respectively. In PSO-based image clustering, a single particle represents the N_c cluster centers. Each particle x_i is constructed as $x_i = \{m_{i1}, ...,m_{ij}, ..., m_{iN_c}\}$ where m_{ij} refers to the j-th cluster centroid vector of the i-th particle. The swarm of particles, is initialized such that each particle contains N_c number of randomly selected cluster centers from the range $[K_{min}, K_{max}]$, where K_{min} is usually assigned to 2, and K_{max} describes the maximum particle length, which means the maximum number of possible cluster centers, which is completely image size dependent. The framework of the

Algorithm 15 DPSO algorithm

Input: Gray Scale Image Matrix

Output: Partition Matrix

begin
Initialize the maximum cluster number K_{max} and all the constant parameters
Initialize each particle with K randomly selected cluster centers and initial velocities
Initialize each particle x_i with the $pBest_i$ and gBest
while gBest converges or for n iterations **do**
for i= 1 to NOP **do**
  for x= 1 to rows **do**
    for y= 1 to cols **do**
      Say p^{th} pixel has coordinate (x,y)
      Find *Euclidean distance* using (6.3) from p^{th} pixel to all center of i^{th} particle
      Assign p^{th} pixel to j^{th} center of the i^{th} particle if they contribute minimum distance
    end for
  end for
  Compute intra cluster distance of i^{th} particle using (6.5)
  Compute inter cluster distance of i^{th} particle using (6.4)
  Compute weighted quantization error of i^{th} particle using (6.6)
  Compute the fitness value of i^{th} particle using (6.7)
  Update pBest position $x_{pB}(i)$ and pBest value $f_{pB}(i)$ of i^{th} particle
end for
Update gBest
Accelerate particles using (6.1) and (6.2)
Update inertia weight
end while
end

DPSO algorithm is given in Algorithm 15. In line 2 of the algorithm the maximum number of cluster centers is supplied by the user. The value of this constant is completely dependent on the image size. In this line all the constant parameters such as the inertia weight, Ψ_p and Ψ_g (positive learning factors) are also initialized. In step 3, the initialization of each particle with K randomly selected cluster centers are done and also the initial velocities are assigned randomly during program run. In step 4, each particle x_i is initialized with the $pBest_i$ and gBest values. Lines 5-23 iterate until the gBest converges or for n iterations. Step 6 loops for each particle, here NOP refers to number of particles in the swarm population. Steps 7-8 iterate to contain each p^{th} pixel in the image matrix. In step 10, the *Euclidean distance* using (6.3) from p^{th} pixel to all centers of i^{th} particle are determined. In step 11, the p^{th} pixel is assigned to j^{th} center of the i^{th} particle if they contribute minimum distance. In step 14, the intra cluster distance of i^{th} particle is calculated by using the (6.5). In step 15, the inter cluster distance of i^{th} particle is calculated by using the (6.4). In step 16, the quantization error value of i^{th} particle is calculated by using (6.6). In step 17, the fitness value of i^{th} particle is obtained using (6.7). At step 18, the pBest position $x_{pB}(i)$ and pBest value $f_{pB}(i)$ of i^{th} particle are updated. At line 20, the gBest value is updated by comparing with the previous gBest. At line 21, the particles are moved using (6.1) and (6.2). At line 22, the inertia weight is updated. Based on the partition matrix, the noise detection and restoration operations are done and presented in section 6.3.

6.3 Proposed Technique for Noise Detection and Restoration

Based on clustering of the pixels of the noisy image, detection and filtering rules are applied to detect and restore the noisy pixels. Some assumptions and property of a noise free digital image are made in the proposed algorithm for detection and filtering operations. The noisy image matrix and the partition matrix are used for noise detection. An 3×3 window or neighborhood of the pixel under consideration is given in Fig. 6.1. There are four directions in the 3×3 test window where in each direction, the center pixel is to be classified as noisy or noise-free. If the center pixel is classified as noisy by the proposed rule based technique then it is restored using its 3×3 window based neighboring pixels. When the pixels in a particular direction of the 3×3 test window belong to a common cluster then it is considered that they lie on an edge. As the pixels present on an edge have similar intensity values. The detection and filtering rule is proposed and outlined in Algorithm 16. The probabilities of various options based on 3×3 are as follows:

- If nine pixels belong to nine different clusters, total number of pixels (N_p): total number of dissimilar cluster centers (N_c)=9:9. The case may be interpreted as extremely noisy.

- When N_p: N_c= 9:8, then only two pixels among nine belong to same cluster. Under this circumstance the test pixel is also noisy.

- If N_p: N_c= 9:7, then only three pixels among nine are similar. If these three pixels lie on any direction in the 3 × 3 test window given in Fig. (6.1), then the center pixel is noise-free. Otherwise it is treated as noisy. When all the three pixels of any direction of a 3 × 3 window belong to same cluster then it is obvious that they lie on an edge.

- Consider situation where N_p: N_c= 9:6, four among nine pixels are similar. In this case, at first an operation is performed to find whether there exists any edge in the test window. If edge is found then the test pixel is noise free, otherwise it is checked whether the test pixel is one of the pixels of the four similar pixels or not. If it belongs to the set of similar pixels then the test pixel is treated as noise free.

- If N_p: N_c= 9:5, five pixels among nine of the test window are similar. All other possible cases are as follows: N_p: N_c= 9:4 or 9:3 or 9:2. For all these cases, if any edge found in the 3 × 3 test window then the test pixel is noise free. If no edge found as well as the center pixel belong to the cluster center where the maximum number of pixels of the test window do not belong to then it is noisy.

- Consider the case where N_p: N_c= 9:1, the window is homogeneous. All the pixels of the test window are similar and hence the test pixel is noise-free.

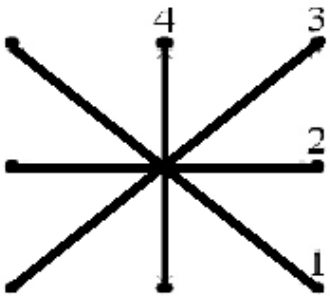


Figure 6.1: Four directions of an 3 × 3 window where the center pixel is under consideration

The image clustering for denoising techniques presented in this chapter are evaluated through simulation results and comparisons with the state of the art methods in section 9.4 of chapter 9. The *fixed length particle swarm optimization (FPSO)* and *different length particle swarm optimization (DPSO)* algorithms are implemented and results are generated in the section. In this case the digital images are corrupted with *random valued noise* and *salt and pepper noise*. The *liftingbody*, *boat*, *lena*, *barbara* and *cameraman* images are used

Algorithm 16 Noise detection and restoration technique

1. **Detection :** All nine pixels of the 3×3 window belong to nine different clusters. Then the center pixel is noisy. This type of window is called as heterogeneous window.
   Restoration : As all pixels are detected as noisy, averaging or smoothing filtering is used to restore the noisy pixel. In the 3×3 window, select the direction in which the standard deviation is minimum, as in this case the pixels are most closely clustered. Compute the average of the three pixels present in the direction of the test window. Replace the center pixel of the window by the average value.

2. **Detection :** All nine pixels of the test window belong to same cluster. Then the center pixel is noise free. This type of window is known as homogeneous window.
   Restoration : No filtering method is applied in this case. The center pixel is left unchanged.

3. **Detection :** Four directions are considered in the test window as four edges may exist around the center pixel. If all the three pixels present on any particular direction belong to same cluster then that directional pixels are considered to be present on any edge. In that case the center pixel is noise free.
   Restoration : No filtering method is applied in this case. The center pixel is kept unchanged.

4. **Detection :** The center pixel is not present on any edge and only neighboring three pixels belong to same cluster. In this case, if the center pixel belongs to the cluster center in which the neighboring three pixels also belong to then it is not noisy. Otherwise it is noisy.
   Restoration : Replace the noisy pixel by the center of the cluster where three neighboring pixels belong to.

5. **Detection :** Four or more neighboring pixels around the test pixel belong to same cluster and the test pixel also does this then the test pixel is noise free. When four or more pixels belong to cluster center C_k and the test pixel belong to cluster center C_l, and $C_k \neq C_l$, then the test pixel is noisy.
   Restoration : Find the centroid value C_k, where the four or more pixels belong to. Replace the test pixel by C_k.

to evaluate the proposed algorithm. Color *lena* image is also used to evaluate the proposed algorithm for noisy color image. Those images are corrupted using minimum (10%) to maximum noise density (100%) using the noise models. The results given by the proposed

algorithm are presented by three popular and useful performance metrics such as Peak-Signal-Noise Ratio (PSNR (dB)), Image Enhancement Factor (IEF), Structure Similarity Index Measure (SSIM) and visual effects.

6.4 Discussions

A generalized method for noise detection and filtering are proposed which enhanced the noise restoration quality quantitatively and qualitatively. Proposed algorithm is capable of suppressing salt-and pepper noise as well as random-valued-noise with good restoration results. It uses the general property of a noise-free image to detect the noisy pixels which leads to a conclusion that the proposed technique may be treated as generalized noise removal approach for the digital images. The technique is also capable to deal with high density of impulses in the digital images.

PSO-based image clustering has been used over the genetic algorithm based clustering because of convergence of PSO is very fast and it's another advantage is ease of implementation. PSO needs less number of user intervention then the genetic algorithm. Three evaluation criteria are used to form the fitness function in the PSO-based clustering. Using the fitness function in PSO-based clustering DPSO can give more compact clusters and larger inter cluster distance when compared to K-means. In Euclidean distance function, the spatial distance as well as positional distances are computed to compare the similarity or dissimilarity of any two pixels during grouping of pixels. The weakness of the method is also resolved to some extent by different length particle swarm optimization algorithm to determine the values of the user parameter, number of cluster centers.

Limitation of the different length swarm particles is, the number of cluster centers which is assigned for a particle does not change. This limitation is resolved to some extent by using a large number of particles, which are used in simulation purpose to get diverse swarm because in that case the particles may have all possible type cluster centers. A large number of particle set represents a maximum diversity in the swarm of particles. To work over the limitations of the fixed length particles, variable length PSO based clustering can be done to remove the limitations.

Chapter 7

Fuzzy Switching Median Filter

Contents

7.1 Proposed Technique for Noise Detection 133
7.2 Proposed Technique for Noise Restoration 134
7.3 Proposed Technique for Fuzzy Switching 137
7.4 Discussions 138

In this chapter, a fuzzy based switching technique is proposed which aims at detection and filtering of impulse noises from digital images. Two different types of noise models are used to obtain the noisy images. In this two step process, the noise free pixels are remained unchanged. The proposed detection algorithm uses 5×5 window, based on all neighboring pixels on the centered of the window of a noisy pixel. Two different weighted median filters are devised and a particular one is applied selectively to the noisy pixel based on the characteristics of the neighboring pixels within the window. Instead of a single threshold, two thresholds are used in the proposed fuzzy membership (MF) function to measure the noise level and accordingly a filtering method is applied to restore the corrupted pixel.

Most of the existing methods cited in the proposed thesis for comparisons, lacking from selection of proper window size and noise detection technique. Those methods do not include all the pixels of the test window to classify the center pixel. Some of them simply compare the center pixel of the window with the maximum and minimum intensity values within the test window for noise detection rule. Other problem is that those algorithms use only one threshold value for comparison to define the noisy pixel. In that case the decision is very rigid. Rather a noise detection rule is proposed which is based on the maximum and minimum intensity values within the test window as well as on the differences between the center pixel and all other pixels in the main four directions around the center pixel. After that the level of noise is also determined on the test pixel using a fuzzy membership function [FDM]. It calculates the membership value as the noise level on the center pixel based on its minimum absolute difference. In the filtering rule two different cases are considered prior to median filtering. The directions where standard deviations are on the minimum side are

selected and those directional pixels are used for median filtering. The pixels having minimum spatial distances in the neighborhood of the test pixel are given more weight (w) before the median operation. The fuzzy technique does the rest. Both of the intensity values, the noisy pixel and the pixel value after the median operation are used for restoration of the noisy pixel. When the noise level on the test pixel is low, the pixel value of the noisy image is more used than the median value. When the noise level is high, a large portion of the median value is used for restoration of the noisy pixel. These are the strengths and novelties of the proposed method.

In this chapter, the fuzzy logic [Zadeh 1965] [1] is applied to select a type of filter determining the noise level on the pixel under question. The fuzzy decision rule [2] is applied in the proposed fuzzy switching median filter (FSMF) to measure the noise level on the center pixel of the 5×5 test window. The noise detection rule, the median filtering techniques along with the fuzzy switching technique performs excellent when they are applied to the noisy images corrupted by two type impulse noises. The fuzzy based switching algorithm is used for refinement to the restoration technique. The noise removal technique is proposed here contains three main parts. First the noise detection rule which detects a pixel as noisy in the 5×5 window by applying a noise detector. The detection and filtering rule uses all neighbor directional weighted pixels in the 5×5 window. The noise detection rule is based on the maximum and minimum intensity values within the test window as well as on the differences between the center pixel and all other pixels in the four directions around the center pixel. After that the level of noise is also determined on the test pixel using a fuzzy rule. The membership value is calculated as the noise level on the center pixel based on its minimum absolute difference. In the filtering rule two different cases are considered prior to median filtering. The directions where standard deviations are on the minimum side are selected and those directional pixels are used for median filtering. The pixels having minimum spatial distances in the neighborhood of the test pixel are given more weight (w) before the median operation. Both of the intensity values, the test pixel and the pixel value after the restoration of the noisy pixel are used for filtering. If the noise level on the test pixel is low, the pixel value of the noisy image is given more weight than the median value. If the noise level is high, the maximum amount of the median value is used for restoration of the noisy pixel.

Proposed noise detection, filtering and fuzzy based switching algorithms are outlined in sections 7.1, 7.2 and 7.3 respectively. Finally, the chapter is summarized in section 7.4.

[1] http://en.wikipedia.org/wiki/Fuzzy_logic
[2] http://plato.stanford.edu/entries/logic-fuzzy/

7.1 Proposed Technique for Noise Detection

The proposed technique is a switching median filter which detects the noisy pixels and then applies the proposed weighted median filtering technique to those pixels. The size of the test window for noise detection and filtering techniques is kept same and is shown in Fig. 7.1. Four directions are considered around the center pixel of the 5 × 5 test window to classify the test pixel. The figure shows the window which includes all the neighborhood pixels around the center pixel. These pixels are referred as all neighboring directional weighted pixels (ANDWP). The proposed noise detection steps are given in Algorithm 17. Steps 2-3

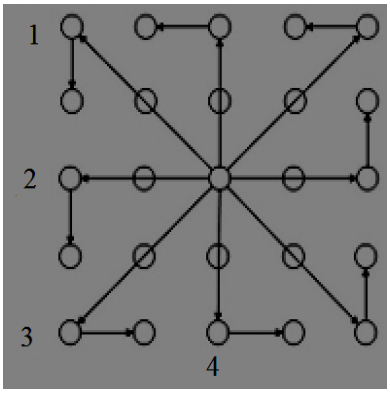


Figure 7.1: 5 × 5 Window

iterate to contain each pixel of the image matrix each time. Steps 4-5 iterate the 5 × 5 test window each time where the center pixel $y_{i,j}$ is the pixel under consideration. In steps 6-8, the minimum and maximum gray values of the test window around the center pixel $y_{i,j}$ are selected as g_{min} and g_{max} respectively. If $y_{i,j}$ does not lie in $[g_{min}, g_{max}]$ then that pixel is treated as highly noisy. To do that, $r_{i,j}$ is assigned with the value T_2, which is used in fuzzy switching algorithm in section 7.3. Otherwise it calculates the value of $r_{i,j}$. In step 10, the four directional pixels are gathered in D_k (k=1 to 4) as follows:

Let D_k (k=1 to 4) denotes a set of coordinates aligned in the four main directions centered at (0,0) in the test window, i.e.,

. D_1={(-1,-2),(-2,-2),(-1,-1),(0,0),(1,1),(2,2),(1,2)}.

. D_2={(1,-2),(0,-2),(0,-1),(0,0),(0,1),(0,2),(-1,2)}.

. D_3={(2,-1),(2,-2),(1,-1),(0,0),(-1,1),(-2,2),(-2,1)}.

. D_4={(-2,-1),(-2,0),(-1,0),(0,0),(1,0),(2,0),(2,1)}.

It is assumed that there exist four directions which are considered as edges along the center pixel in the 5 × 5 test window. Each of the directional pixels are stored in four sets such as D_1, D_2, D_3 and D_4, (D_k, where k= 1 to 4). When the coordinate of the center pixel of the test window is (i, j), then all four directional pixels are represented by $D_k^{(i,j)}$ (k=1 to 4). And $D_k^{(0,0)}$ (k=1 to 4) represents the four sets of pixels when the coordinate of the center pixel of the test window is (0,0). In steps 11-21, in each direction, the absolute differences are calculated. The three different types of absolute differences between the center pixel and each three types of pixels in each direction are weighted by three constants such as ω_l, ω_m and ω_s to make the differences equivalent. The method is to weigh the absolute differences between the two closest pixels from the center pixel with a large ω_l, weigh the absolute differences between the center pixel and the two corner pixels by ω_m and weigh the absolute differences between the two end pixels from the center pixel with a small value ω_s in each direction, before calculating the sum. Assign ω_s =0.25, ω_m=0.5 and ω_l=1. The values of $\omega_{s,t}$ is assigned in (7.1). In step 22, the minimum of $d_k^{(i,j)}$ is calculated as $r_{i,j}$.

$$\omega_{s,t} = \begin{cases} \omega_l & : (s,t) \in \omega^1 \\ \omega_s & : (s,t) \in \omega^2 \\ \omega_m & : \text{otherwise} \end{cases} \tag{7.1}$$

The value of ω^1 is given in (7.2) and ω^2 is given in (7.3)

$$\omega^1 = \{(s,t) : -1 \leq s,t \leq 1\} \tag{7.2}$$

$$\omega^2 = \{(s,t) : (s,t) = \pm\{(-2,-1),(-1,-2),(1,-2),(2,-1)\}\} \tag{7.3}$$

7.2 Proposed Technique for Noise Restoration

For the purpose of denoising, 3 × 3 and 5 × 5 window regions are used as neighboring pixels around the center pixel. The 3 × 3 neighboring pixels are given more weights than the 5 × 5 because they have minimum spatial distances to the center pixel, so their gray level differences should also be close. The operations of median filtering are done in (7.6) and (7.7). In four directions, four standard deviations are computed at first. Standard deviation signifies how closely the center pixel and all other pixels in a particular direction are clustered. Among these the minimum and second minimum directional pixels are selected because in those directions the pixels are most closely related. The pixels present in those directions are given more weight and thus selected repeatedly to calculate the median operations. The pixels which are nearest to the center pixel and belong to the first minimum direction are

Algorithm 17 Noise detection algorithm

Input: Image Matrix g_{xy}, Threshold (T)
Output: Window's Center Pixel Classified

begin
for x=0 to MaxRow-1 **do**
  for y=0 to MaxCol-1 **do**
    for i=-2 to 2 **do**
      for j=-2 to 2 **do**
        g_{min} and g_{max} be the minimum and maximum intensity values within the window around $y_{i,j}$
        if $y_{i,j} \leq g_{min}$ && $y_{i,j} \geq g_{max}$ **then**
          $y_{i,j} = T_2$
        else
          Initialize D_k to contain seven pixels in each direction
          $d_{i,j}^{(k)} = \omega_{s,t}|y_{i+s,j+t} - y_{i,j}|$
          if D $\in \omega^1$ (7.2) **then**
            $\omega_{s,t}$=ω_l
          else
            if D $\in \omega^2$ (7.3) **then**
              $\omega_{s,t}$=ω_m
            else
              $\omega_{s,t}$=ω_s
            end if
          end if
          $r_{i,j} = \min d_{(i,j)}^{(k)}$
        end if
      end for
    end for
  end for
end for
end

selected twice and the far distance pixels are selected once. From the second minimum direction, the closest pixels of the center pixel are taken only single time. W_{st}, W_{uv} and W_{pq} are used to represent the selection operators prior to compute the median operations. This shows the pixels aligned along D_k are closest to each other and the center value should be close to them. These pixels are assigned with an extra weight (w) while restoring the noisy pixels. The proposed weighted median filtering method is illustrated in Algorithm 18.

Algorithm 18 Filtering algorithm

1: Calculate the standard deviations $\sigma_{i,j}^{(k)}$ of the intensity values of all $y_{i+s,j+t}$ with (s,t) $\in D_k^0$ (k= 1 to 4).

2: Find the minimum of $\sigma_{i,j}^{(k)}$, where k= 1 to 4, as

$$min1_{i,j} = \min_k\{\sigma_{i,j}^{(k)} : k = 1 \; to \; 4\} \tag{7.4}$$

3: Find the second minimum of $\sigma_{i,j}^{(k)}$, where k= 1 to 3, as

$$min2_{i,j} = \min_k\{\sigma_{i,j}^{(k)} : k = 1 \; to \; 3\} \tag{7.5}$$

4: Form a set S_1 from the pixels located at directions $min1_{i,j}$ and $min2_{i,j}$, which lie in [g_{min}, g_{max}] but not the center pixel as it is a noisy pixel. Repetition operator $\diamond$ [Brownrigg 1984] works with the neighboring pixels for selecting them zero or more times depending upon their distance and alignment from and with the center pixel. If the cardinality of S_1 is not null then calculate median of S_1 using (7.6).

5: If S_1 has null cardinality, then calculate median of set S_2. This set is formed from the same directional pixel like the previous one by excluding the rule of belonging to [g_{min}, g_{max}] of the pixels using (7.7).

6: Calculate the first median as

$$med1 = median\{\omega_{s,t} \diamond y_{i+s,j+t} \; and \; \omega_{p,q} \diamond y_{i+p,j+q} and \; \omega_{u,v} \diamond y_{i+u,j+v}\} \tag{7.6}$$

(s,t) $\in$ {-1,0,1}, (s,t)$\neq$ (0,0) and $y_{i+s,j+t} \in min1_{i,j}$ direction. (p,q) include all pixels in 5 $\times$ 5 window and $y_{i+p,j+q} \in min1_{i,j}$ direction and (p,q)$\neq (0,0)$. (u,v) $\in$ {-1,0,1}, (u,v)$\neq (0,0)$ and $y_{i+u,j+v} \in min2_{i,j}$ direction. Assign $\omega_{s,t}$=2, $\omega_{u,v}$=1 and $\omega_{p,q}$=1.

7: The second median (for the case given step 5), is calculated as

$$med2 = median\{\omega_{s,t} \diamond y_{i+s,j+t} \; and \; \omega_{u,v} \diamond y_{i+u,j+v}\} \tag{7.7}$$

(s,t) $\in$ {-1,0,1}, (s,t)$\neq$ (0,0) and $y_{i+s,j+t} \in min1_{i,j}$ direction. (u,v) includes all pixels in 5 $\times$ 5 window and $y_{i+u,j+v} \in min1_{i,j}$ direction and (u,v)$\neq (0,0)$. Assign $\omega_{s,t}$=2 and $\omega_{u,v}$=1.

8: R= $med1$ or $med2$ by the above rule.

7.3 Proposed Technique for Fuzzy Switching

The $r_{i,j}$ has been calculated in Algorithm 17 for impulse detection. When $r_{i,j}$ is small the noise level on the 5×5 window is also small. But large $r_{i,j}$ implies that the center pixel has a high chance of noisy. Rather than crisp thresholding to classify the center pixel as noisy or noise free pixel, a fuzzy switching method has been proposed which restores the center pixel depending upon the value of the $r_{i,j}$. It calculates the membership value as the noise level on the center pixel based on its minimum absolute difference. When $r_{i,j}$ is small but noisy then a large portion is taken of the noisy pixel and a small portion of the restored pixel value to restore the noisy pixel. For the opposite case the reverse technique is adopted. Two threshold values such as T_1 and T_2 have been used for the purpose. The value of $r_{i,j}$ is compared with these threshold values to make the restoration more perfect. If the value of $r_{i,j}$ is less than T_1 then the corresponding pixel is detected as noise free. When $r_{i,j}$ is greater than T_2 then it is highly corrupted and restoration is done on the basis of the restored pixel. For the intermediate value of $r_{i,j}$, the proposed algorithm restores the noisy pixel through the help of noisy and restored pixel value. When the difference value $r_{i,j}$ is small then its membership value is also small that implies noise effect on that pixel is also small. If the difference value $r_{i,j}$ is high then its corresponding membership value is also high. That means there is high noise impact on the center pixel. In between very high membership and very low membership values there exits some kind of uncertainty. The fuzzy membership function along with the switching method are described in the Algorithm 19.

The r_{ij} is computed as the minimum of the four direction indices. The threshold values are compared with r_{ij} given in (7.8). Direction indices are calculated using the absolute differences from the center pixel to all other pixels in each direction. A total of six differences are calculated as there are six pixels in each direction apart from the center pixel. A fractional part is taken into account to calculate the direction indices. Out of six times, each of the one fourth (W_s=0.25) and one half (W_m=0.5) portions are taken two times while calculating the absolute differences. And for the two closest pixels these are calculated two times and full portions (W_l =1) are taken. But in that case the differences are assumed to very small as they are adjacent pixels (spatial distance is minimum). Eventually the minimum of the four direction indices is taken to calculate the r_{ij}. So r_{ij} becomes a small value despite the wide range of noise level used. That is why the T_1 and T_2 are changed marginally. Two user parameters viz., T_1 and T_2 are used as threshold values to partition the fuzzy membership function. They are kept constant in a range to generalize all types of bench mark images.

The fuzzy switching median filter presented in this chapter are evaluated through results and comparisons with the state of the art methods in section 9.5 of chapter 9. In this case the digital images are corrupted with *random valued noise* and *salt and pepper noise*. The

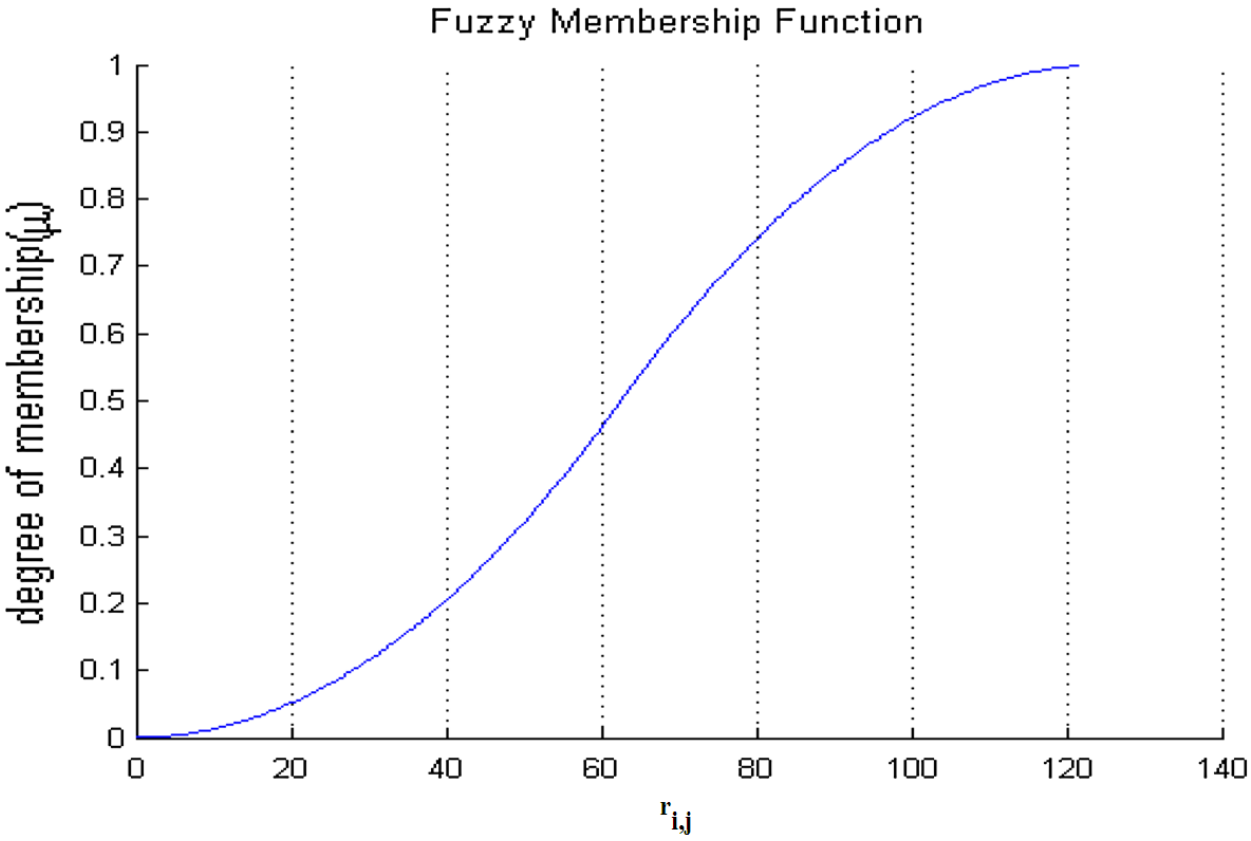


Figure 7.2: (a)Fuzzy membership function for a noisy pixel [Ima]

liftingbody, *boat*, *lena*, *barbara* and *cameraman* images are used to evaluate the proposed algorithm. Color *lena* image is also used to evaluate the proposed algorithm for noisy color image. Those images are corrupted using minimum (10%) to maximum noise density (100%) using the noise models. The results given by the proposed algorithm are presented by three popular and useful performance metrics such as Peak-Signal-Noise Ratio (PSNR(dB)), Image Enhancement Factor (IEF), Structure Similarity Index Measure (SSIM) and visual effects.

7.4 Discussions

A fuzzy switching decision technique for suppression of impulses in digital imagery proposed in this chapter. The images which are corrupted by any level of impulses i.e., from 10% to 100%, are trained using two noise models such as salt and pepper noises (SPN) and random valued noises (RVN). Only the corrupted pixels are restored using the proposed operator,

Algorithm 19 Fuzzy switching algorithm

1: The membership function for a noisy pixel is shown in Fig. 7.2. Using two threshold values T_1 and T_2, the fuzzy membership function is

$$\mu(r_{i,j} : T_1, T_2) = \begin{cases} 0 & : r_{i,j} < T_1 \\ 2[\frac{r_{i,j}-T_1}{T_2-T_1}]^2 & : T_1 \leq r_{i,j} < \frac{T_1+T_2}{2} \\ 1 - 2[\frac{r_{i,j}-T_2}{T_2-T_1}]^2 & : \frac{T_1+T_2}{2} \leq r_{i,j} < T_2 \\ 1 & : T_2 \leq r_{i,j} \end{cases} \tag{7.8}$$

2: Using R (given in algorithm 18) and pixel value $y_{i,j}$, the restored pixel is computed as

$$Restored\ pixel\ value = y_{i,j} + \mu(r_{i,j}) * (R - y_{i,j}) \tag{7.9}$$

3: The user parameters T_1 and T_2 are searched in wide ranges for various bench mark images with different noise density and observed that the following ranges give best result by the proposed algorithm. The values of T_1 in [55, 65] and T_2 in [115, 125] give the restoration results given by the algorithm.

the noise free pixels are kept unchanged. Proposed noise removal algorithm works on 5 $\times$ 5 neighboring pixels of the center pixel which is to classify and restore. Simple arithmetic operations are used on the neighboring pixels of the center pixel to detect the noisy pixel in the test window. An arithmetic weighted sum has been computed during detection. The filtering operator is based on weighted median operation of the neighboring pixels. Among three main stages of the noise removal operator, the backbone is the fuzzy switching or fuzzy refinement algorithm. Fuzzy refinement has been applied on each pixel depending on the noise level of the center pixel which is actually dependent on the weighted sum calculated in detection stage.

Chapter 8

Denoising in Frequency Domain

Contents

8.1 Proposed Technique for Denoising 142

8.2 Discussions 144

The performance of the spatial domain filters are well on digital images, although they suffer with resolution degradation in restored images. They use smoothing filters in a fixed test window, produces artifacts and sometimes blurred the images. Wavelet transform is preferred because of its sparsity, multiresolution and multiscale natures. Frequency domain filters can be converted to spatial domain by using a simple kernel for the desired filter. Frequency domain filters are useful and more efficient when no simple kernel is present in the spatial domain. Denoising in wavelet domain is useful because it analyzes the signal at different frequencies with different resolutions. This is known as multi resolution analysis (MRA). The denoising problem is modeled as follows:

Let $o(t)$ and $n(t)$ are the original and noisy images respectively with independent and identically distributed (i.i.d.) zero mean, Gaussian Noise $g(t)$, given in (8.1), where $g(t)$ has Gaussian distribution N(0,1) and σ_v is noise variance.

$$n(t) = o(t) + \sigma_v g(t) \tag{8.1}$$

In this chapter, the original image $o(t)$ is recovered by removing the *Gaussian* and *Poisson* noises from $n(t)$. In wavelet based denoising, first the noisy image is transformed into the wavelet domain and then the wavelet coefficients are thresholded, and finally inverse transformation is done. Thresholding is done on each pixel [Strela 1999, Downie 1998, Coifman 1994]. The simple method involves manipulation of the coefficients [Malfati 1997] efficiently. The coefficients which are noisy are replaced by a suitable value. Inverse transformation is performed to get restored noise free image.

In this chapter, wavelet domain based denoising algorithm is proposed. A sub-band adaptive wavelet thresholding is done in the proposed algorithm. Some parameters of wavelet

decomposition are searched using *Genetic algorithm*. The noisy images have mixed Gaussian and Poisson noises with a wide range of noise density. The noisy image is partitioned into fixed sized blocks and then transformed into wavelet domain. In this algorithm, binary encoded GA [Williams 1998] with binary tournament selection, uniform crossover and low probability mutation rate is employed. *Elitism* model is proposed to preserve the best chromosome of each generation and eventually to get the global optimal solutions.

Rest of the chapter is organized as follows. The proposed method is presented in section 8.1 and summary of the chapter is given in section 8.2.

8.1 Proposed Technique for Denoising

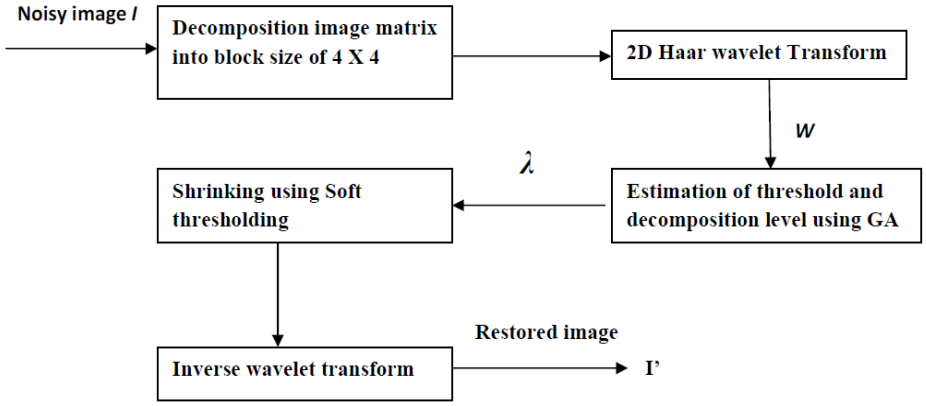


Figure 8.1: Diagram of proposed denoising method

The complete diagram of the proposed algorithm is given in Fig. 8.1. First the input image is decomposed into 4×4 sub images and then the two dimensional discrete forward wavelet transformation is done on each sub images. After that the threshold value and the value of the decomposition level are searched using GA. After the search operation, shrinking method is applied on the wavelet coefficients on each sub band using soft thresholding method. Finally the inverse transformation is done to obtain the restored sub image. The search operation using the GA is described below.

Genetic algorithm starts searching for the modified Bayesian threshold value and for the value of decomposition level by encoding the population randomly. A chromosome P in binary form is encoded to represent the threshold value and the decomposition level. In random seven bits are generated to make each chromosome. Four bits is used for correction of

the BayesShrink. A single bit precedes the four bits to make the positive/negative threshold. The 5 bits are taken to have a decimal correction of +15 to -15 to Bayesian threshold. This technique decomposes the test images up to level four. Remaining 2 bits are for the decomposition level. The '00' substring gives decimal representation of 4. The string '1101010' gives a corrected value of -10 to the threshold and level of decomposition to two. Another chromosome such as '0100000' defines a correction of +8 and 4 to the parameters specified. The objective is to maximize the value of PSNR as a result this is the fitness function f of each chromosome in the GA based optimization technique.

To preserve the elite chromosomes, the best chromosome of each iteration is taken to the next iteration with and without being involved subsequent stages of GA. Here, the worst one is also taken. If the worst one is better than the best one of the previous generation then it survives otherwise it is replaced by the best chromosome of the previous generation.

Binary tournament selection (BTS) [D.E. 1989] has been used for selecting the fittest chromosomes to make the mating pool with the same size as the population. 2 strings are taken randomly from the initial set of chromosomes and the best one is selected to the pool of mating until it becomes empty. When tie occurs the chromosome is taken randomly.

Uniform crossover method is done in the proposed scheme. It is iterated for n/2 times for a pool size of n. First two chromosomes are selected randomly from the pool. A string of similar length of the chromosome is taken randomly. The bit value is checked and when it is swapping is done in the bit positions of the two chromosomes. Otherwise it is not done for that position. The probability is (μ_c)

After crossover every offspring undergoes the mutation operation. It has been done with very low probability (μ_m).

The parameters of the algorithm are listed below:

- Population size: [5-10]
- Chromosome length: 7
- μ_c: [0.8-0.9]
- μ_m: [0.1-0.2]
- Total Number of generation: [5-10]

The algorithm initiates by encoding a population of chromosome. Let it takes a population of n binary chromosomes. Then the string is converted to decimal value and then the threshold value and the level of decomposition level is taken from the decimal value. The decimal threshold value is added to the Bayeshrink threshold value. After that for each chromosome the fitness is calculated using the new threshold and decomposition level. All stages of the genetic algorithm are performed to proceed to next iteration. After certain

number of iterations the optimal Bayesian threshold value and value of decomposition level is found. Using these parameter the corrupted image is restored.

Proposed frequency domain based filtering technique presented in this chapter is validated through experimental results and analysis in section 9.6 of chapter 9. Some state of the art filters like VisuShrink, SureShrink and BayesShrink are implemented and compared with the proposed GA based BayesShrink method. One ultrasound image is being used for the comparison purpose with other methods. Various graphical representations of the results are given to substantiate the performance of the proposed technique. The ultrasound image is corrupted with low (σ=30), medium (σ=60) and high noise density (σ=90) using the mixed Gaussian and Poisson noises. The performance of the algorithm is validated using peak signal to noise ratio (PSNR).

8.2 Discussions

A denoising algorithm has been proposed for the medical images which are corrupted with mixed Gaussian noise and Poisson noise. Traditional BasyesShrink operator has been modified and optimized in the proposed wavelet based denoising algorithm. In a specified range the Bayesian threshold has been varied and the corresponding decomposition level of wavelet transformation has been searched using Genetic Algorithm. Genetic algorithm has been used very effectively to search the pair of wavelet parameters, since these two are the most important parameters of the wavelet based denoising technique.

Part III

Experimental Results and Comparisons

Chapter 9

Experimental Results and Comparisons

Contents

9.1 Noise Suppression 149
9.1.1 DWMD Filter 149
9.1.2 ANDWP Filter 150
9.1.3 VMM Filter 153
9.1.4 EPR Filter 156
9.1.5 SIR Filter 158
9.1.6 Comparisons 163
9.2 Filter Optimization 171
9.2.1 Based on Genetic Algorithm 171
9.2.2 Based on Particle Swarm Optimization 172
9.2.3 Comparisons 174
9.3 Noise Density Estimation for Adaptive Median Filter 184
9.3.1 Results 184
9.3.2 Comparisons 185
9.4 PSO based Clustering for Noise Detection and Restoration 200
9.4.1 Results 200
9.4.2 Comparisons 201
9.5 Fuzzy Switching Median Filter 215
9.5.1 Results 215
9.5.2 Comparisons 220
9.6 Denoising in Frequency Domain 242
9.6.1 Results 242
9.6.2 Comparisons 242
9.7 Discussions 243

The experimental results and comparisons on the proposed algorithms are presented in the following sections. The first one represents the results and comparative analysis on the noise suppression techniques. In this section, filters such as *DWMD*, *ANDWP*, *VMM*, *EPR* and *SIR* are taken into account for generating the extensive results. In this case the digital images are corrupted with *random valued noises*. In section 9.2, the *genetic algorithm* and *particle swarm optimization* based filter optimizations are done. They are compared to each other and also with the state of the art methods. The *SIR* filter has been optimized in this section by using the two search algorithms and then the comparative results have been generated. Only the *SIR* filter has been optimized but the others can also be used for optimization. The *SIR* is the most efficient technique of the noise suppression section. In section 9.3, noise density estimation for adaptive median filter based results are generated and comparisons are presented. In this case, two noise models are considered. Along with the random-valued-noise, the salt-and-pepper noise is considered for denoising. All subsequent algorithms are proposed to deal with both the noises in the digital images. In section 9.4, PSO based clustering for noise restoration technique based results and comparisons are given. In section 9.5, fuzzy switching median filter based results and comparisons are presented. In section 9.6, results and comparative discussions are given for the technique denoising in frequency domain. In section 9.1 to section 9.5, the proposed filters are developed on the spatial domain which are used to suppress the *random-valued-noise and salt-and-pepper noise* from the digital images while the filter given in section 9.6 is based on the frequency domain and has been used to denoise the *Gaussian and Poisson* noises from the medical images. The spatial domain and frequency domain based discussions are given in section 9.7.

In this section, the proposed spatial domain based algorithms discussed in sections 3.1 and 3.2 in chapter 3, are validated through quantitative and qualitative restoration results. In section 9.1.6, the restoration results of the proposed algorithms are compared with the state-of-the-art methods in the literature. Popular benchmark digital images are contaminated by the random-valued-noise (RVN) and salt-and-pepper noise (SPN) models and then applied to the proposed algorithms. The *liftingbody* image of size 512 $\times$ 512, *boat* image of size 512 $\times$ 512, *lena* of size 512 $\times$ 512, *barbara* of size 512 $\times$ 512 and *cameraman* of size 256 $\times$ 256 are used to evaluate the proposed algorithms. Color *lena* image is also used in this section to evaluate the proposed algorithms for noisy color image. The results given by the proposed algorithms are presented by three popular and useful performance metrics such as Peak-Signal-Noise Ratio (PSNR (dB)), Image Enhancement Factor (IEF) and Structure Similarity Index Measure (SSIM). PSNR is used because it is most common metric to measure the quality of reconstructed images. The SSIM considers image degradation as perceived

change in structural information and the IEF measures the quality improvement done in the restoration by the algorithm. Various graphical representations of the results using different attributes are given to substantiate the performances of the proposed techniques. Bench mark images are trained using minimum (10%) to maximum noise density (100%) using the noise models. Comparative results and discussions of all these proposed algorithms with the state of the art algorithms such as SMF [Brownrigg 1984], AMF [Hwang 1995], BDND [Ng 2006], EDBA [Srinivasan 2007], IDBA [Nair 2008], FBDA [Nair 2012] and Nasri [Nasri 2013] are given in this chapter.

9.1 Noise Suppression

In the proposed noise detection and filtering algorithms, three parameters viz., I, T and R are supplied during the program run. These parameters are varied in the following ranges to obtain the restoration results, viz., *maximum number of iterations (I)* $\in$ [4, 6], *Threshold value (T)* $\in$ [300, 900] and *decreasing Rate of threshold value in each iteration (R)* $\in$ [0.7, 0.9]. These parameters are empirically determined. To get better restoration results using the proposed filters for a particular noise density, the values of the three parameters are varied widely in the specified ranges. For example, let the input image is 60% noisy *Lena* image and for that the *PSNR* is obtained by varying T from 300 to 900 with an increment of 50, R from 0.7 to 0.9 with an increment of 0.05 and I from 1 to 6 with an increment of 1 and then maximum PSNR (dB) and values of the other metrics are obtained.

9.1.1 DWMD Filter

The *DWMD* filter (section 3.1.1) based restoration results are presented here. The noise detection in directional weighted minimum deviation based filtering technique is done through simple absolute differences between the center pixel of the 5 $\times$ 5 test window. The noise filtering technique is based on finding the minimum standard deviation in the four directions. This is done by adding a constant value to the mean of optimal directional pixels iteratively. Tables 9.1, 9.2 and 9.3 show the restoration results of **DWMD** filter in terms of *PSNR*, *IEF* and *SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images respectively for varying noise densities from 10% (minimum) to 100% (maximum) contaminated by *Random Valued Noises (RVN)*. At the bottom of each table, the average values are shown to reflect the improvement of the results clearly. From the tables, it can be noticed that for complete range of noise densities, the proposed algorithm has given a consistent improvement in terms of *PSNR*, *IEF* and *SSIM* values. From the *IEF* table, it can be noticed that for low to high noise densities, the proposed algorithm has given excellent image enhancement values. The values given by the *DWMD* algorithm close to 1 but except the case of *Barbara* image. The

performances of the *DWMD* filter are satisfactory for all noise densities as per the restoration results. The performance of the proposed *DWMD* filter can also be visually noticed from the Fig. 9.1, Fig. 9.2 and Fig. 9.3 respectively where each plot of the graph represents the PSNR, IEF and SSIM values for individual image with varying noise densities from 10% (minimum) to 100% (maximum) contaminated by *RVN*. From the graph, it is seen that the PSNR for *Lena* image gradually decreases after noise density 60%.

Table 9.1: PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DWMD** filter for varying noises with RVN

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
Lena	34.65	33.36	32.00	31.38	29.18	26.89	22.43	20.54	18.75	17.42
Cameraman	28.36	27.12	25.78	23.77	20.34	19.30	18.21	16.39	15.87	13.53
Barbara	32.08	30.63	29.64	27.44	25.81	23.72	22.05	20.81	19.62	18.36
Liftingbody	37.56	36.09	35.18	33.27	31.17	29.48	28.11	26.79	26.74	25.67
Boat	31.42	30.11	29.51	28.15	27.08	26.59	25.06	24.41	23.22	21.79
Average	32.81	31.46	30.42	28.80	26.71	25.19	23.17	21.78	20.84	19.35

Table 9.2: IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DWMD** filter for varying noises with RVN

Image	*Noise Density/IEF*									
	10	20	30	40	50	60	70	80	90	100
Lena	294.87	266.54	235.87	227.68	196.19	157.65	108.98	79.34	67.71	59.71
Cameraman	140.17	122.65	113.92	101.55	92.52	84.50	62.13	72.12	54.81	47.01
Barbara	168.50	153.87	139.71	117.81	106.88	100.70	82.61	70.40	51.61	42.73
Liftingbody	460.30	452.21	436.70	412.08	376.72	329.79	276.95	262.91	237.48	208.54
Boat	153.61	150.22	132.24	111.50	97.25	92.57	63.81	51.95	50.25	47.91
Average	243.49	229.09	211.68	194.12	173.91	153.04	118.89	107.34	92.37	81.18

9.1.2 ANDWP Filter

In this section, the proposed *ANDWP* filter (section 3.1.2) based restoration results are presented. In this case, all neighboring pixels of the center pixel in the 5×5 window are considered for noise detection and filtering purposes. The directional weighted minimum deviation based filtering technique is further improved by finding the value of the center

Table 9.3: SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DWMD** filter for varying noises with RVN

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
Lena	0.9912	0.9877	0.9847	0.9821	0.9812	0.9773	0.9721	0.9672	0.9541	0.9501
Cameraman	0.9086	0.9072	0.9048	0.9028	0.9021	0.8990	0.8965	0.8934	0.8923	0.8907
Barbara	0.9110	0.9130	0.9106	0.9038	0.9010	0.9001	0.8959	0.8988	0.8932	0.8917
Liftingbody	0.9987	0.9932	0.9913	0.9857	0.9815	0.9773	0.9731	0.9712	0.9643	0.9613
Boat	0.9350	0.9287	0.9021	0.8995	0.8975	0.8941	0.8910	0.8868	0.8820	0.8811
Average	0.9489	0.9460	0.9387	0.9348	0.9327	0.9296	0.9257	0.9235	0.9172	0.9150

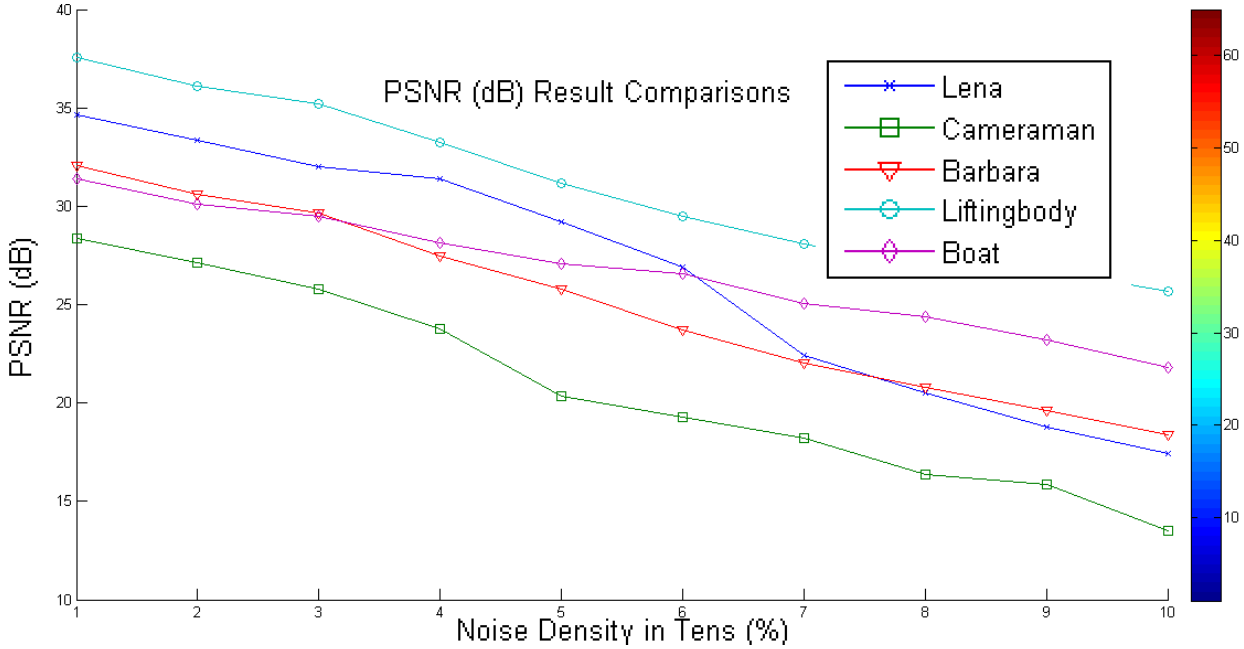


Figure 9.1: Graph for **DWMD** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises

pixel using the principal of maxima/minima. This is done by forming a quadratic equation by the directional pixels which represents the standard deviation and then minimizing the standard deviation. In this case it has obtained more appropriate value compared to the *DWMD* technique. Tables 9.4, 9.5 and 9.6 show the restoration results of **ANDWP** filter in terms of *PSNR*, *IEF* and *SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images respectively for varying noise densities from 10% (minimum) to 100% (maximum) contaminated by *Random Valued Noises*. At the bottom of each table, the average values

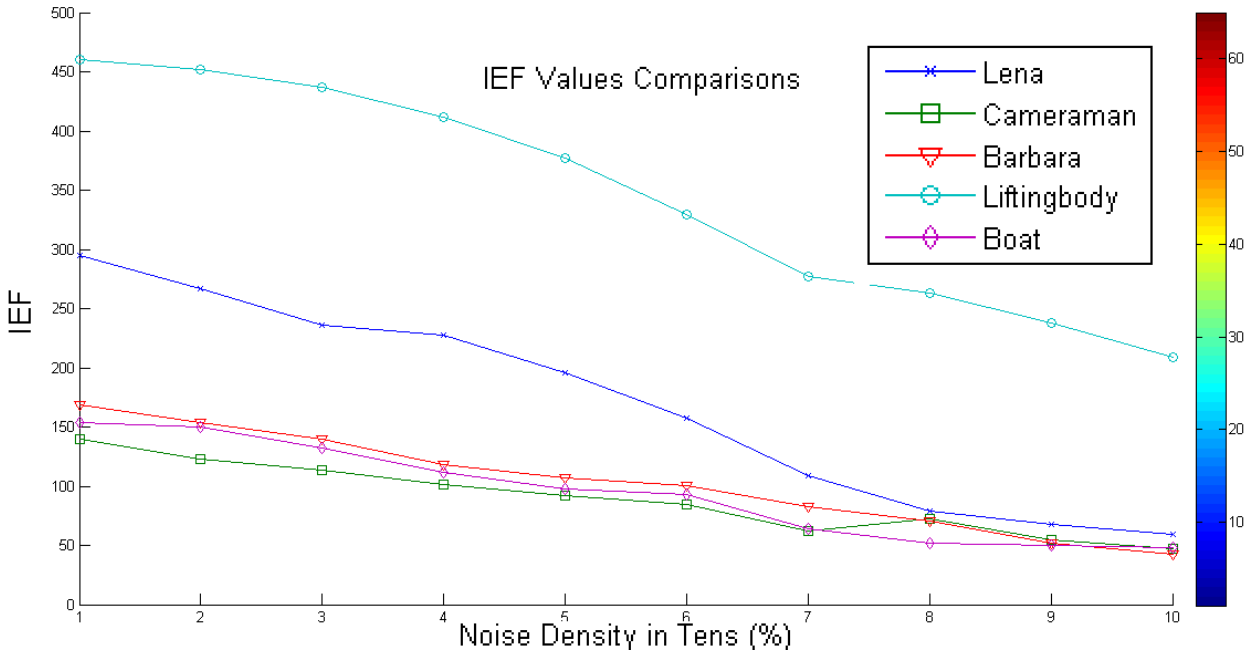


Figure 9.2: Graph for **DWMD** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises

are given to show the improvement of the results clearly. From the *PSNR* values of the table, it can be observed that, except the cameraman image, the proposed technique has generated excellent *PSNR* values for the other images. The *SSIM* values given by the algorithm are very close to 1. From the tables, it can be noticed that for low to high noise densities, the proposed algorithm has obtained a consistent improvement in terms of *PSNR*, *IEF* and *SSIM* values. From the *IEF* table, it can be noticed that for low to high noise densities, the proposed algorithm has obtained excellent image enhancement factor values. The values given by the *ANDWP* algorithm close to 1 for all the images considered but only the *Barbara* image has suffered. The performances of the *ANDWP* filter are satisfactory for all noise densities as per the restoration results obtained. Visual representation of the tables on the *ANDWP* restoration results can be noticed from the Fig. 9.4, Fig. 9.5 and Fig. 9.6 respectively where each plot of the graph represents the PSNR, IEF and SSIM values for individual image for varying noise densities from 10% (minimum) to 100% (maximum) contaminated by *RVN*.

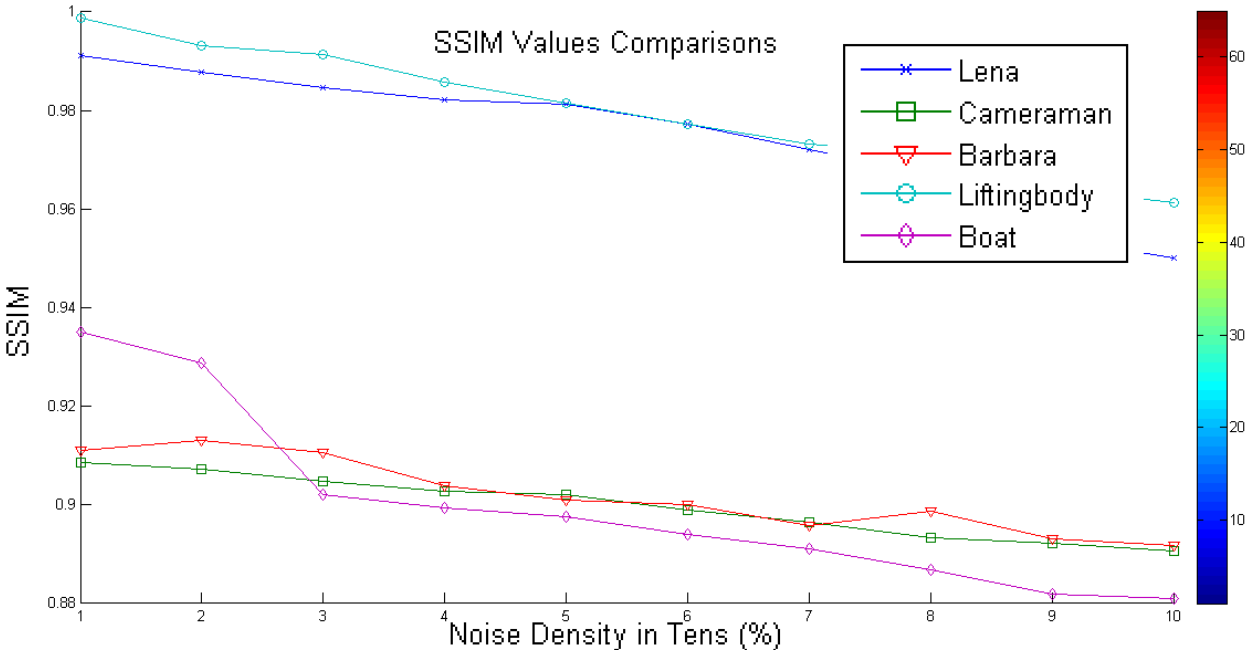


Figure 9.3: Graph for **DWMD** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises

Table 9.4: PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **ANDWP** filter for varying noises with RVN

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
Lena	34.97	33.81	32.74	31.54	29.70	27.22	22.98	20.66	19.92	17.87
Cameraman	28.87	27.91	26.18	24.21	20.82	19.87	18.67	16.98	16.37	14.21
Barbara	32.80	31.63	30.65	28.21	26.41	24.20	22.77	21.41	20.46	19.91
Liftingbody	38.33	36.76	35.49	33.89	31.70	30.01	28.78	27.37	27.14	26.36
Boat	33.16	32.01	30.33	29.23	28.72	26.95	24.71	22.91	21.56	20.50
Average	33.62	32.42	31.07	29.41	27.47	25.65	23.58	21.86	21.09	19.77

9.1.3 VMM Filter

In this section, the *VMM* filter (section 3.2.1) based restoration results are presented. The VMM filter has considered all neighboring pixels of the center pixel in the 3 × 3 and 5 × 5 window regions for noise detection and filtering purposes. The performance of *ANDWP* filtering technique has been improved by replacing the noisy pixel with the median value of

Table 9.5: IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **ANDWP** filter for varying noises with RVN

Image	*Noise Density/ IEF*									
	10	20	30	40	50	60	70	80	90	100
Lena	312.71	267.70	258.93	242.18	226.71	190.22	132.33	100.09	92.01	78.71
Cameraman	173.22	146.81	131.56	122.89	106.76	98.32	87.90	85.44	69.58	61.65
Barbara	206.33	183.71	179.66	151.66	138.12	133.50	103.44	96.65	58.30	52.38
Liftingbody	465.21	460.77	453.81	415.71	399.61	334.44	323.06	289.65	249.33	236.30
Boat	303.10	270.73	256.12	237.50	230.98	193.79	156.77	104.67	89.50	76.25
Average	292.11	265.94	256.01	233.98	220.43	190.05	160.70	135.30	111.74	101.05

Table 9.6: SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **ANDWP** filter for varying noises with RVN

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
Lena	0.9968	0.9903	0.9867	0.9855	0.9840	0.9813	0.9776	0.9725	0.9621	0.9523
Cameraman	0.9099	0.9087	0.9054	0.9031	0.9028	0.8996	0.8976	0.8954	0.8941	0.8931
Barbara	0.9130	0.9156	0.9134	0.9079	0.9066	0.9032	0.8969	0.8922	0.8918	0.8911
Liftingbody	0.9989	0.9948	0.9941	0.9878	0.9851	0.9776	0.9755	0.9750	0.9681	0.9636
Boat	0.9450	0.9359	0.9225	0.9195	0.9100	0.9003	0.8936	0.8807	0.8732	0.8615
Average	0.9527	0.9491	0.9444	0.9408	0.9377	0.9324	0.9282	0.9232	0.9179	0.9123

the neighboring pixels of the center pixel. Two window regions such as 3×3 and 5×5, are considered for median filtering to obtain a better restoration results than the *DWMD* and *ANDWP* techniques. Tables 9.7, 9.8 and 9.9 show the restoration results of **VMM** filter in terms of *PSNR*, *IEF* and *SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying noise densities from 10% (minimum) to 100% (maximum) contaminated by *Random Valued Noise*. The average values are shown at the bottom of the table to show the improvement of the results with the increasing noise percentage accordingly. From the *PSNR* values of the table, it can be observed that the proposed algorithm has given good *PSNR* values for the other images but not for the *Cameraman* image only. The *SSIM* values obtained by the algorithm are also close to 1. From the tables, it can be noticed that from low to high noise densities, the proposed algorithm has given a clear improvement in terms of *PSNR*, *IEF* and *SSIM* values. From the *IEF* table, it can be noticed that for low to high noise densities, the proposed algorithm has given an excellent image enhancement values. The values obtained by the *VMM* algorithm close to 1, but the *Barbara* image. The

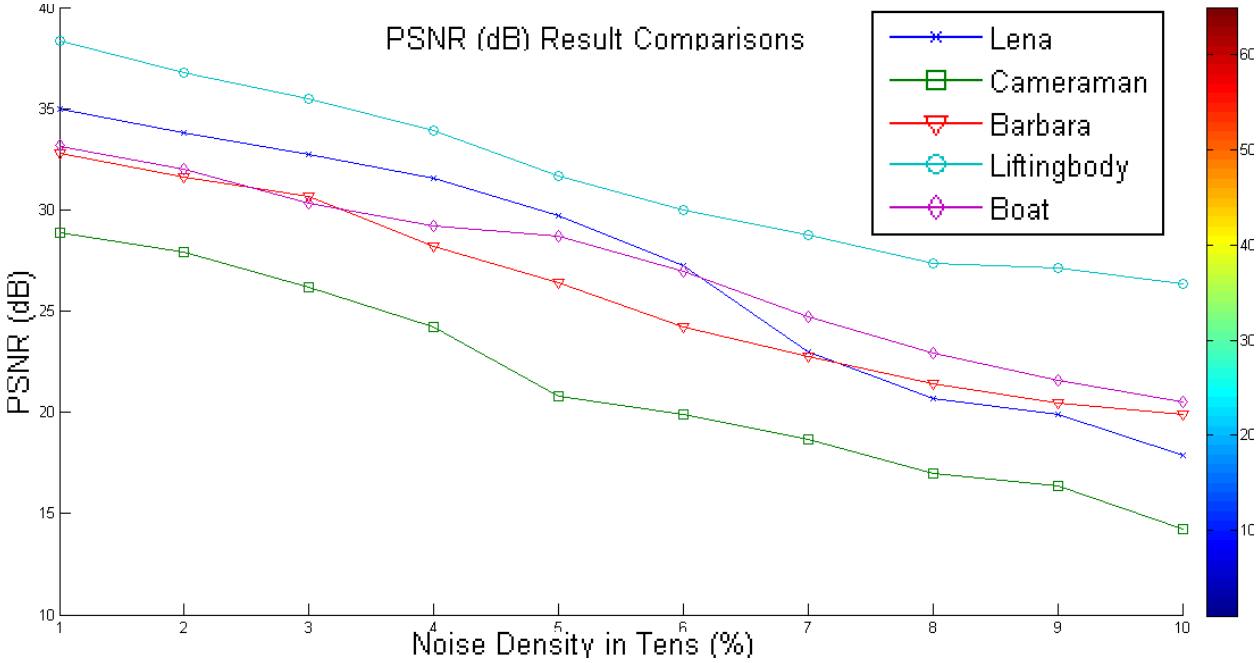


Figure 9.4: Graph for **ANDWP** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises

performances of the *VMM* filter are satisfactory for all noise densities as per the restoration results. Visual representation of the tables on the *VMM* restoration results are shown in Fig. 9.7, Fig. 9.8 and Fig. 9.9 respectively where each plot of the graph represents the PSNR, IEF and SSIM values for individual image for varying noise densities from 10% (minimum) to 100% (maximum) contaminated by *RVN*.

Table 9.7: PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **VMM** filter for varying noises with RVN

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
Lena	35.43	33.97	33.27	31.98	30.43	28.31	24.65	21.80	20.19	18.52
Cameraman	29.68	28.88	27.77	25.13	22.71	20.87	19.87	17.76	17.32	15.70
Barbara	33.22	31.98	31.26	28.87	26.91	24.79	23.43	21.76	20.95	20.45
Liftingbody	38.77	36.95	35.76	34.65	31.81	30.65	29.17	27.81	27.78	26.98
Boat	32.40	31.54	30.12	28.01	27.70	25.95	24.77	23.17	22.52	21.97
Average	33.90	32.66	31.63	29.72	27.91	26.11	24.37	22.46	21.75	20.72

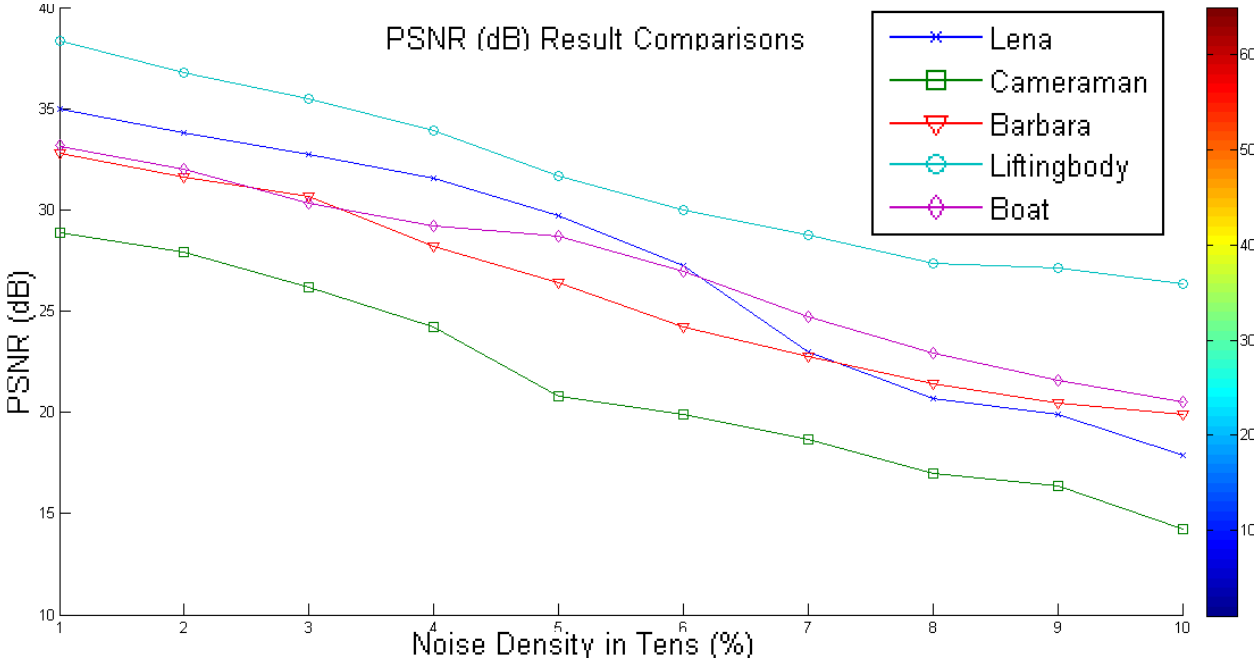


Figure 9.5: Graph for **ANDWP** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises

Table 9.8: IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **VMM** filter for varying noises with RVN

Image	*Noise Density/IEF*									
	10	20	30	40	50	60	70	80	90	100
Lena	347.78	298.78	286.33	282.80	256.22	220.25	172.79	136.15	102.79	93.66
Cameraman	196.66	167.43	145.33	142.65	126.50	109.25	107.25	93.55	84.23	76.25
Barbara	236.50	213.45	209.76	200.66	198.50	192.55	147.19	107.56	77.25	67.85
Liftingbody	489.50	477.65	459.50	423.35	407.45	397.50	376.50	341.25	287.98	254.50
Boat	235.76	213.69	202.50	196.71	188.43	181.25	152.50	118.71	70.39	64.90
Average	301.24	274.20	260.68	249.23	235.42	220.16	191.24	159.44	124.52	111.43

9.1.4 EPR Filter

In this section, the *EPR* filter (section 3.2.2) based restoration results are presented. This filter also considers all neighboring pixels of the center pixel in the 3×3 and 5×5 window regions for noise detection and filtering purposes. An advance weighted median filtering technique is proposed to obtain better restoration results than the DWMD, ANDWP and VMM methods. 3×3 and 5×5 window regions based median filtering approach obtained

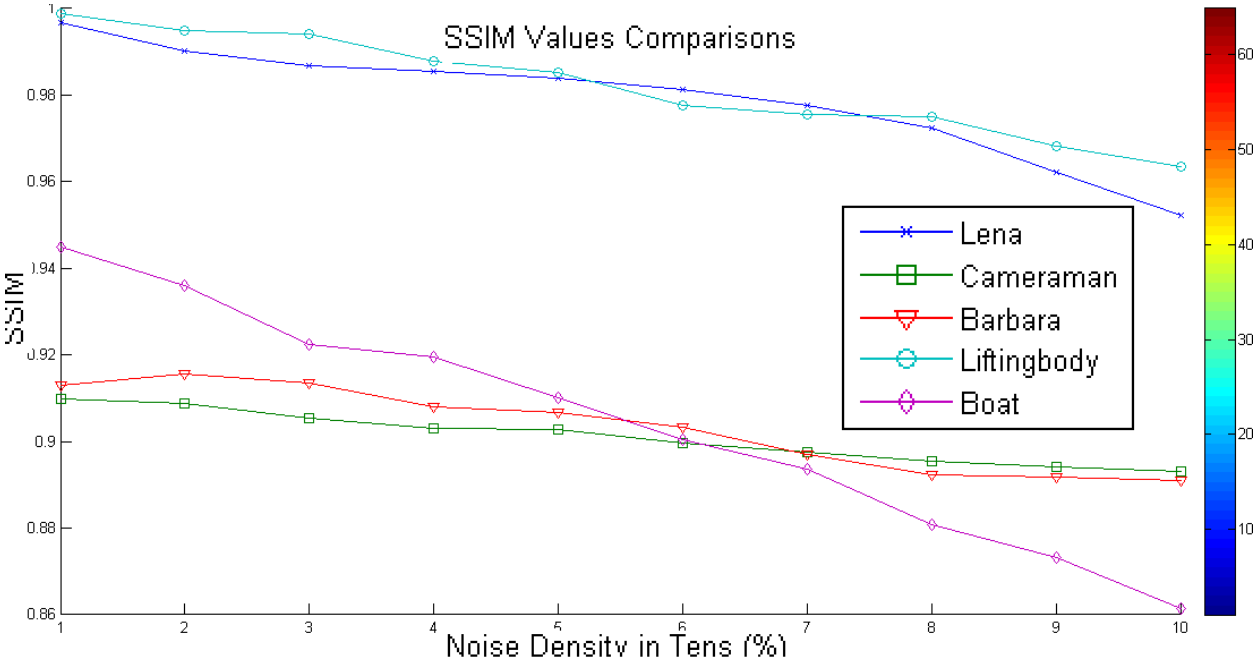


Figure 9.6: Graph for **DWMD** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying random valued noises

Table 9.9: SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **VMM** filter for varying noises with RVN

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
Lena	0.9972	0.9933	0.9877	0.9871	0.9856	0.9842	0.9832	0.9775	0.9678	0.9570
Cameraman	0.9132	0.9154	0.9078	0.9050	0.9021	0.8998	0.8970	0.8965	0.8955	0.8969
Barbara	0.9145	0.9175	0.9154	0.9091	0.9082	0.9049	0.8978	0.8965	0.8946	0.8941
Liftingbody	0.9994	0.9965	0.9954	0.9887	0.9865	0.9759	0.9757	0.9798	0.9695	0.9695
Boat	0.9641	0.9579	0.9522	0.9495	0.9307	0.9239	0.9216	0.9186	0.9127	0.9150
Average	0.9577	0.9561	0.9517	0.9479	0.9426	0.9377	0.9351	0.9338	0.9280	0.9265

better restoration results than the *DWMD*, *ANDWP* and *VMM* techniques. Tables 9.10, 9.11 and 9.12 show the restoration results of **EPR** filter in terms of *PSNR*, *IEF* and *SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for varying noise densities from 10% (minimum) to 100% (maximum) contaminated by *Random Valued Noises*. At the bottom of each table, the average values are given to show the improvement of the results with the noise percentage accordingly. From the *PSNR* values of the table, it can be

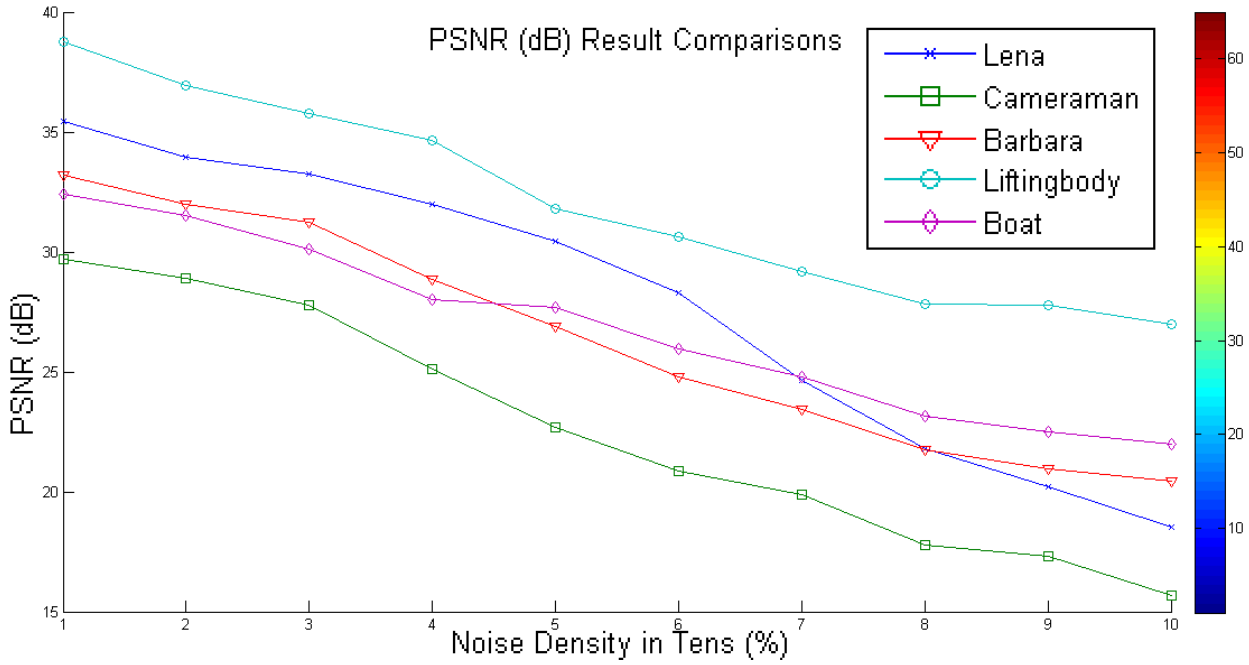


Figure 9.7: Graph for **VMM** in PSNR for *Lena*, *Cameraman*, *Boat*, *Liftingbody* and *Barbara* images for 10% to 100% random valued noises

observed that the proposed algorithm obtained excellent *PSNR* values for the other images but not for the *Cameraman* image. The *SSIM* values given by the algorithm are also close to 1. From the tables, it can be noticed that from low to high noise densities, the proposed algorithm obtained a consistent improvement in terms of *PSNR*, *IEF* and *SSIM* values. From the *IEF* table, it can be noticed that for low to high noise densities, the proposed algorithm obtained an excellent image enhancement values. The values obtained by the *EPR* algorithm close to 1, but the *Barbara* and *Cameraman* images suffer to do so. The performances of the *EPR* filter are satisfactory for all noise densities as per the restoration results. Visual representation of the tables on the *EPR* restoration results are shown in Fig. 9.10, Fig. 9.11 and Fig. 9.12 respectively where each plot of the graph represents the PSNR, IEF and SSIM values for individual image for varying noise densities from 10% to 100% contaminated by *Random Valued Noise*.

9.1.5 SIR Filter

In this section, the *SIR* filter (section 3.2.3) based restoration results are presented. The SIR filter logically partitioned the test image into 5×5 sub images. Two restored versions are generated through proposed technique using 3×3 and 5×5 masks separately of which

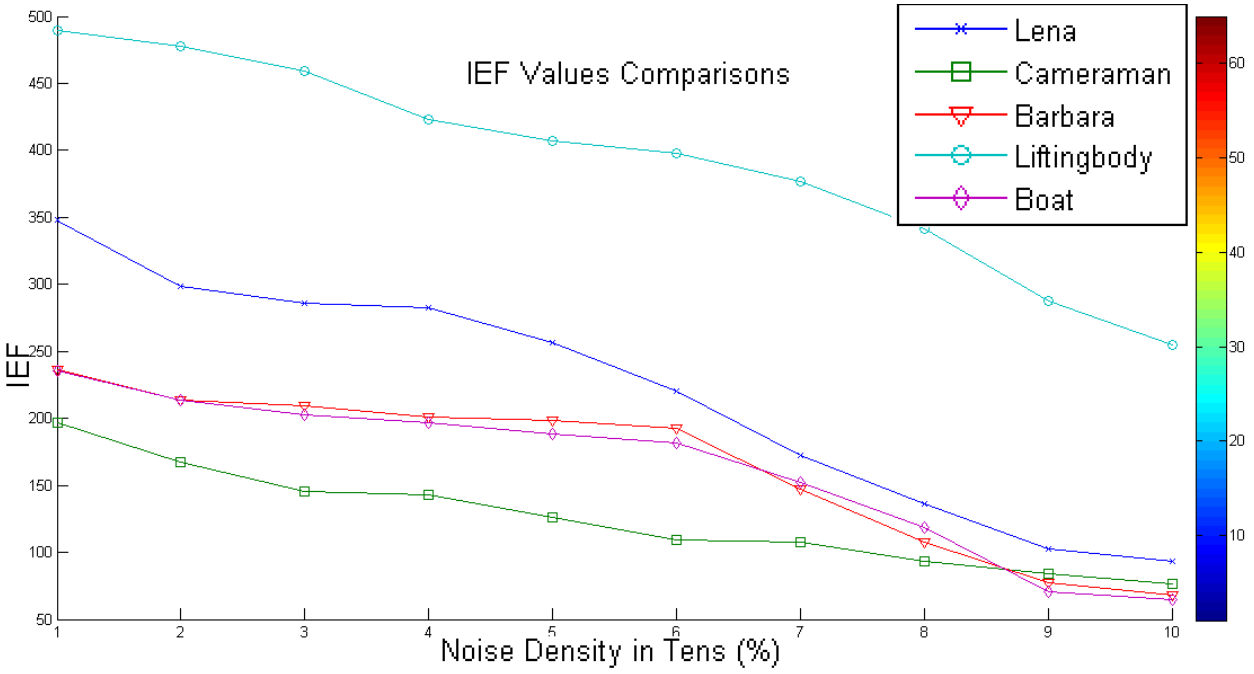


Figure 9.8: Graph for **VMM** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises

Table 9.10: PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **EPR** filter for varying noises with RVN

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
Lena	35.76	34.22	33.48	32.38	30.79	28.46	24.95	21.91	20.65	18.82
Cameraman	29.98	29.25	28.07	25.63	22.78	20.98	19.99	17.95	17.76	16.32
Barbara	33.75	32.25	31.75	29.32	27.43	25.43	23.85	21.98	21.32	20.50
Liftingbody	39.17	37.20	36.50	35.15	32.24	30.98	29.81	28.33	27.90	27.32
Boat	32.17	30.55	29.62	28.17	27.78	26.05	25.10	24.41	23.57	22.60
Average	34.16	32.69	31.88	30.13	28.20	26.38	24.74	22.91	22.24	21.11

better one in terms of PSNR (dB) is chosen. An advance weighted median filtering technique is proposed to obtain better restoration results than the previous methods. 3×3 and 5×5 window regions based median filtering approach obtained far better restoration results than the *DWMD*, *ANDWP*, *VMM* and *EPR* techniques. Tables 9.13, 9.14 and 9.15 show the restoration results of **SIR** filter in terms of *PSNR*, *IEF* and *SSIM* for *Lena*, *Cameraman*,

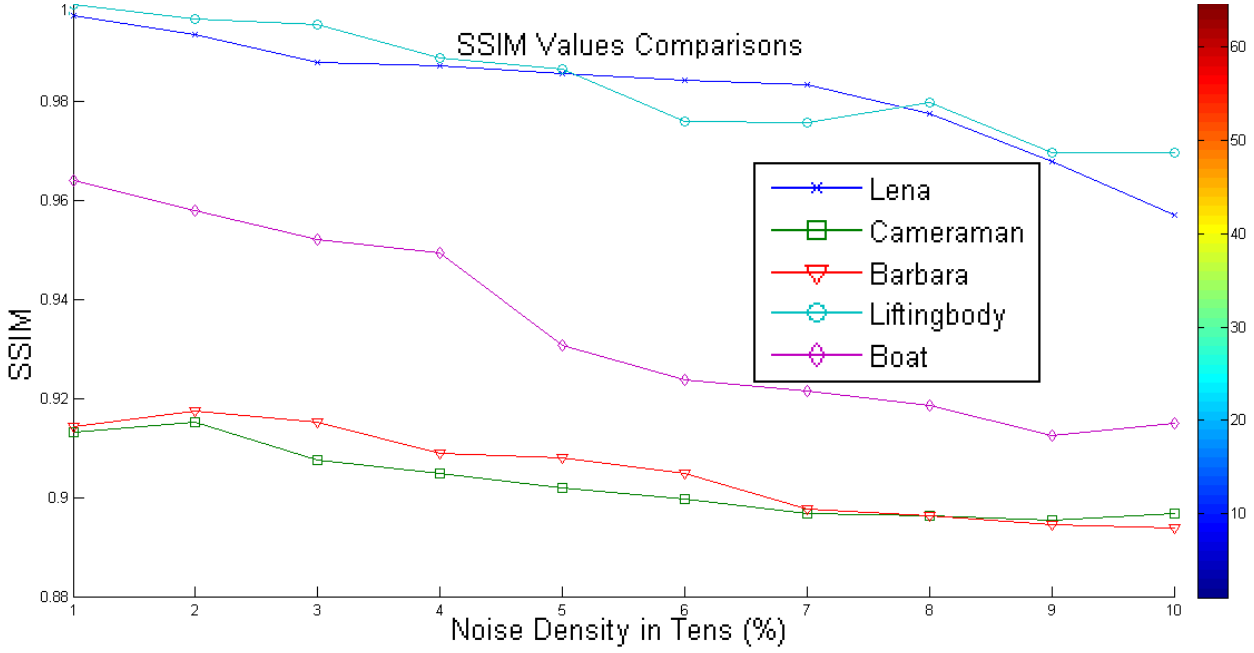


Figure 9.9: Graph for **VMM** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises

Table 9.11: IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **EPR** filter for varying noises with RVN

Image	*Noise Density/IEF*									
	10	20	30	40	50	60	70	80	90	100
Lena	387.50	305.75	298.25	296.22	279.47	252.75	234.35	189.25	146.25	113.50
Cameraman	214.25	198.25	175.50	163.55	152.20	149.55	137.60	119.95	100.65	87.39
Barbara	269.54	236.25	217.50	208.50	197.25	176.25	158.24	137.46	109.45	100.55
Liftingbody	497.25	487.21	467.25	451.25	437.33	401.12	387.75	362.76	312.50	274.76
Boat	255.70	240.91	207.25	200.69	188.62	168.50	147.25	138.09	111.20	89.28
Average	324.84	293.67	273.15	264.04	250.97	229.63	213.03	189.50	156.01	133.09

Liftingbody, *Boat* and *Barbara* images for varying noise densities from 10% (minimum) to 100% (maximum) contaminated by *Random Valued Noises*. The average values obtained are shown at the bottom of each table to show the improvement with the noise percentage accordingly. From the *PSNR* values of the table, it can be observed that the proposed algorithm has given excellent *PSNR* values for the images but not for the *Cameraman* image. The *SSIM* values given by the algorithm are close to 1. From the tables, it can be noticed that from low to high noise densities, the proposed algorithm obtained a consistent improve-

Table 9.12: SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **EPR** filter for varying noises with RVN

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
Lena	0.9989	0.9975	0.9890	0.9876	0.9872	0.9854	0.9850	0.9779	0.9670	0.9582
Cameraman	0.9187	0.9198	0.9118	0.9076	0.9065	0.9032	0.8971	0.8970	0.8963	0.8961
Barbara	0.9173	0.9206	0.9187	0.9113	0.9089	0.9079	0.8979	0.8969	0.8959	0.8944
Liftingbody	0.9996	0.9976	0.9959	0.9889	0.9871	0.9768	0.9759	0.9806	0.9712	0.9707
Boat	0.9787	0.9759	0.9742	0.9732	0.9607	0.9549	0.9532	0.9486	0.9412	0.9376
Average	0.9626	0.9623	0.9579	0.9537	0.9501	0.9456	0.9418	0.9402	0.9343	0.9314

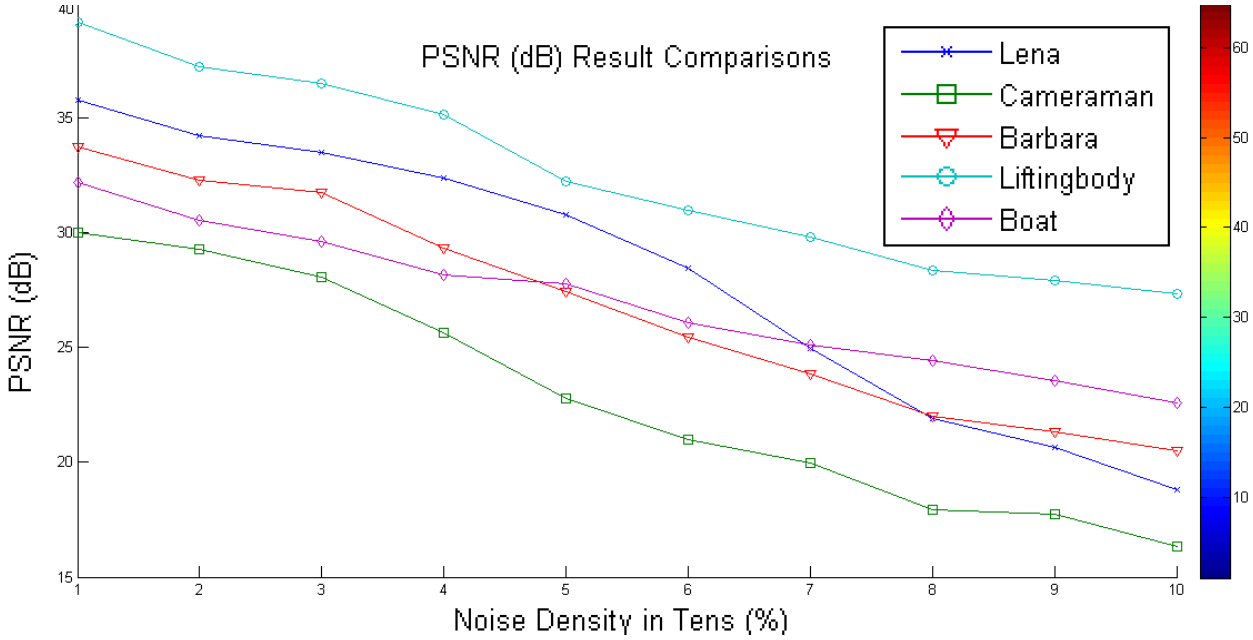


Figure 9.10: Graph for **EPR** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises

ment in terms of *PSNR*, *IEF* and *SSIM* values. From the *IEF* table, it can be noticed that for low to high noise densities, the proposed algorithm obtained an excellent image enhancement values. The values obtained by the *SIR* algorithm close to 1, but the *Barbara* and *Cameraman* images suffer. The performances of the *SIR* filter are satisfactory for all noise densities as per the restoration results. Visual representation of the tables on the *SIR* restoration results are shown in Fig. 9.10, Fig. 9.11 and Fig. 9.12 respectively where each plot of the graph represents the PSNR, IEF and SSIM values for individual image for varying

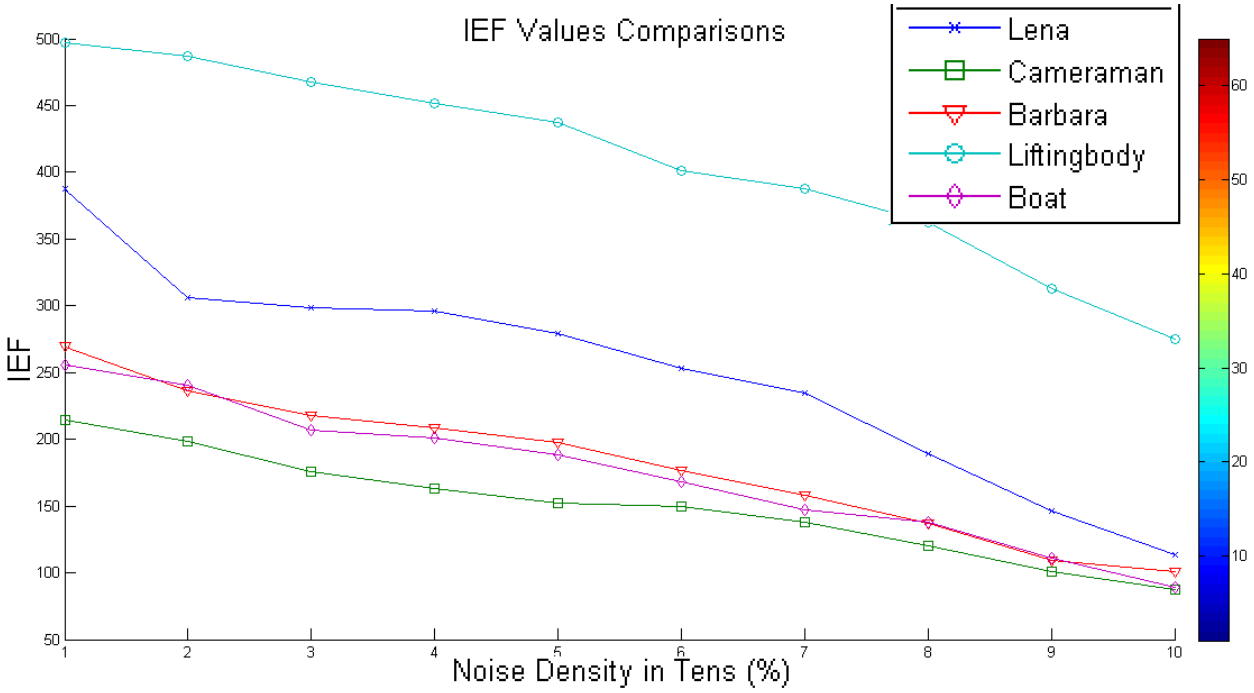


Figure 9.11: Graph for **EPR** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises

noise densities from 10% to 100% contaminated by *Random Valued Noise*.

Table 9.13: PSNR values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **SIR** filter for varying noises with RVN

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
Lena	35.93	34.74	33.65	32.52	31.52	28.75	25.33	22.32	21.52	19.76
Cameraman	30.34	30.25	28.57	25.98	23.37	21.22	20.22	18.37	18.12	17.23
Barbara	34.13	32.98	32.34	29.87	27.76	25.99	24.21	22.22	21.78	21.06
Liftingbody	39.99	37.56	37.55	36.11	33.55	31.34	30.16	28.79	28.32	27.91
Boat	33.28	31.91	29.75	28.21	27.99	26.70	25.34	24.80	23.67	22.95
Average	34.73	33.48	32.37	30.53	28.83	26.80	25.05	23.30	22.68	21.78

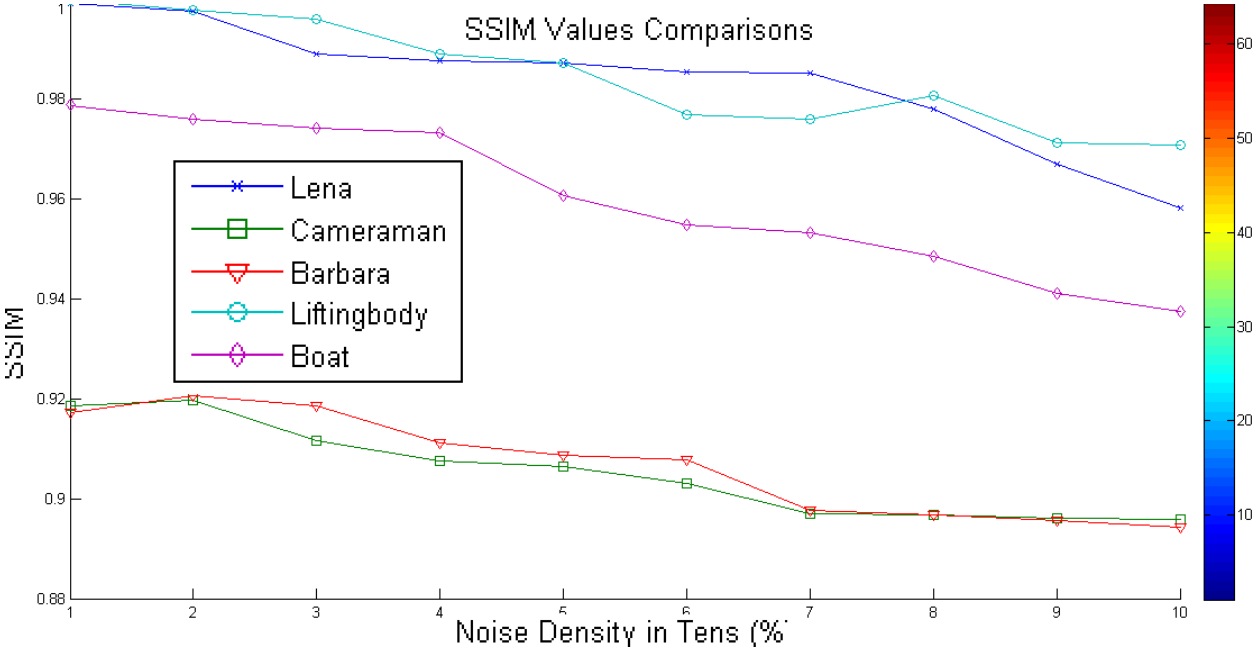


Figure 9.12: Graph for **EPR** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises

Table 9.14: IEF values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **SIR** filter for varying noises with RVN

Image	*Noise Density/IEF*									
	10	20	30	40	50	60	70	80	90	100
Lena	406.25	345.72	301.50	298.25	289.86	267.25	261.50	227.54	176.65	127.25
Cameraman	254.50	231.43	202.71	173.67	163.54	157.29	151.54	143.25	120.50	101.49
Barbara	276.25	242.25	231.51	228.23	201.54	193.49	167.44	146.23	141.43	122.50
Liftingbody	502.50	488.21	477.51	463.29	451.29	437.26	390.23	376.54	317.21	286.40
Boat	292.25	271.35	225.25	217.76	209.77	202.91	179.31	165.94	147.52	121.76
Average	346.35	315.79	287.69	276.24	263.20	251.64	230.00	211.90	180.66	151.88

9.1.6 Comparisons

The restoration results obtained from the five proposed methods such as *DWMD, ANDWP, VMM, EPR and SIR* are compared in terms of *PSNR (dB)*, *SSIM* and *IEF* metrics with the existing methods such as SMF, AMF, EDBA, IDBA and BDND. The average quantitative restoration results for *Lena, Cameraman, Liftingbody, Boat and Barbara* images by all the filters are calculated and compared to each other. Ten noise densities of different level

Table 9.15: SSIM values obtained for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **SIR** filter for varying noises with RVN

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
Lena	0.9991	0.9983	0.9891	0.9878	0.9886	0.9863	0.9878	0.9789	0.9689	0.9599
Cameraman	0.9206	0.9227	0.9187	0.912	0.9100	0.9067	0.8979	0.8979	0.8988	0.8976
Barbara	0.9200	0.9227	0.9183	0.9155	0.9101	0.9086	0.8998	0.8988	0.8965	0.8967
Liftingbody	0.9997	0.9980	0.9971	0.9902	0.9903	0.9787	0.9781	0.9860	0.9752	0.9741
Boat	0.9837	0.9809	0.9784	0.9755	0.9707	0.9689	0.9565	0.9506	0.9482	0.9400
Average	0.9646	0.9645	0.9603	0.9562	0.9539	0.9498	0.9440	0.9424	0.9375	0.9337

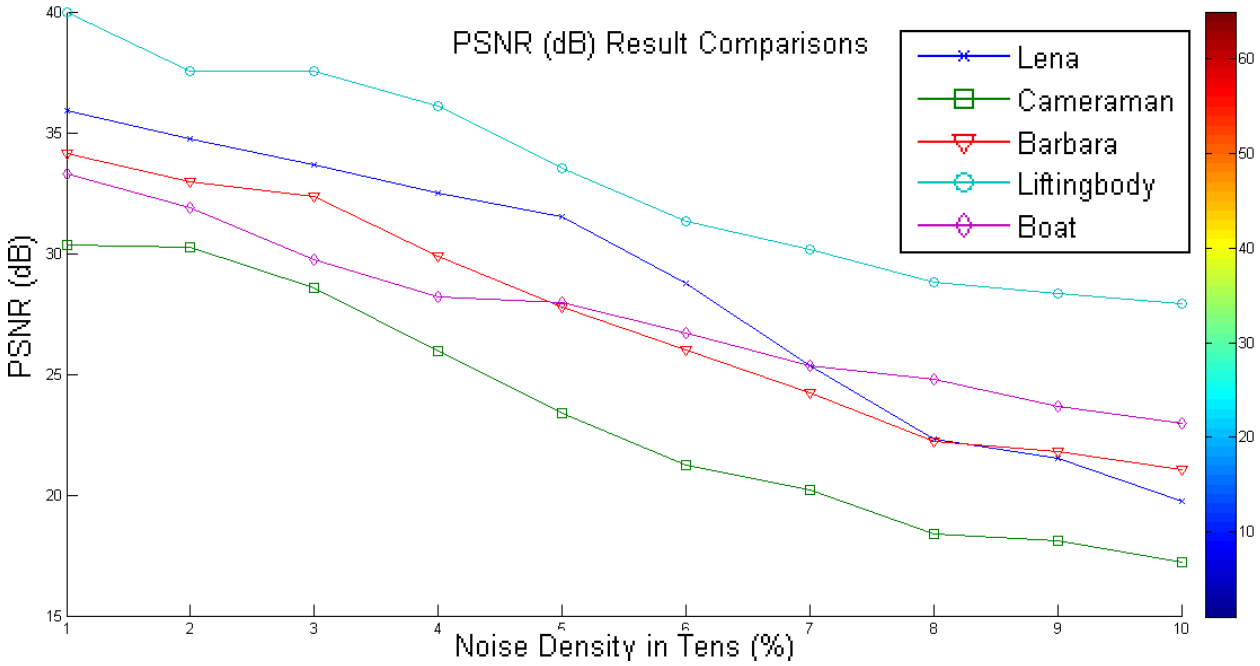


Figure 9.13: Graph for **SIR** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises

are considered for comparison purpose. Table 9.16 shows the average restoration results in terms of PSNR (dB) for the eleven filters. Five proposed filters and six existing filters are compared extensively in this table. SMF filter performs very poor compared to other filters. AMF and DWM have given similar results for the entire noise density range. EDBA, IDBA and BDND obtained similar PSNR values also. Although BDND performs best among them for the higher noise densities. The proposed DWMD performs better than the BDND

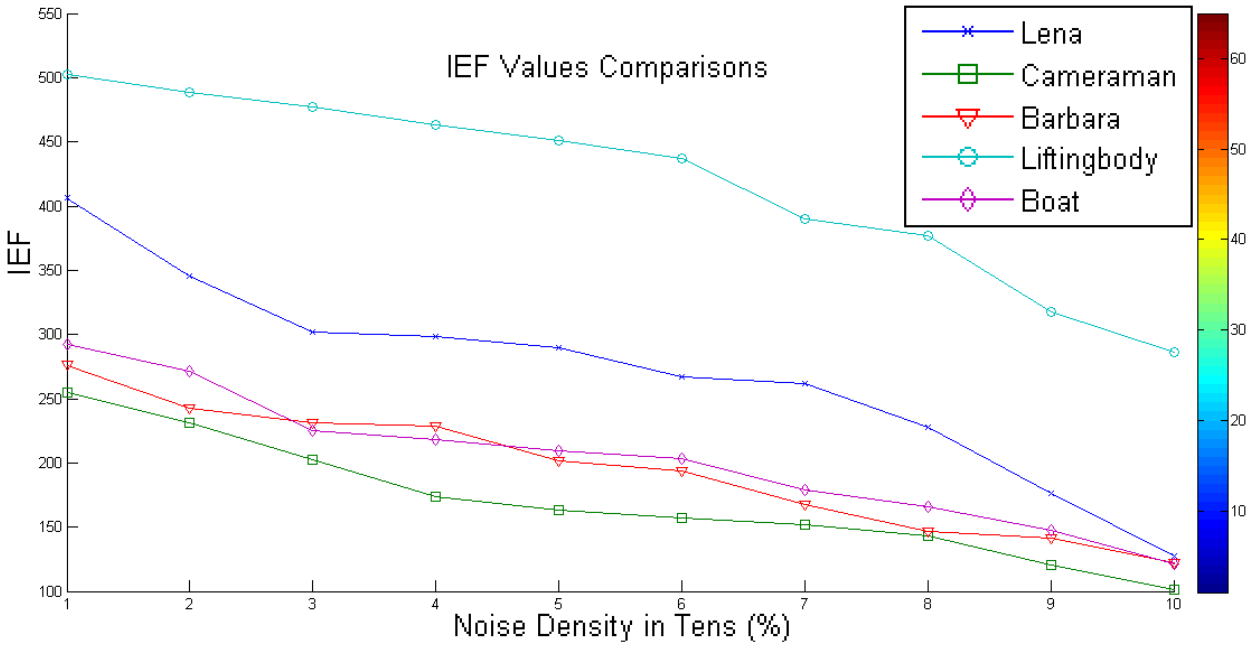


Figure 9.14: Graph for **SIR** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises

filter for low noise density and but performs better for higher noise density. The ANDWP performs better than the DWMD filter when the images are corrupted by low noise density but similar for high noise corruption. The VMM performs much better than the ANDWP for the entire noise density range. The EPR obtained similar results in terms of PSNR but performs much better for the highly noisy images. The **SIR** performs the best in the table in terms of PSNR for the entire noise density range considered. For the highly noisy images it performs excellent. The corresponding pictorial representation of the table is shown in Fig. 9.16. Table 9.17 shows the average restoration results in terms of IEF for five proposed filters and six existing filters. These filters are compared extensively in this table. SMF and AMF filters perform very poor compared to other filters. DWM and EDBA obtain similar results for the entire noise densities range. IDBA and BDND have given similar IEF values for the low noise density but the highly noisy images. The proposed DWMD generated small amount of better IEF values than the BDND filter for the entire noisy images. The ANDWP, VMM and EPR filters perform similarly for the 10% to 100% noise density for all the images. Among them the EPR performs better than the VMM for the high noise density. The **SIR** performs excellent in the table in terms of IEF for the entire noise density range. For the highly noisy images it performs simply outstanding. The corresponding graph the table 9.17 is shown in Fig. 9.17. Table 9.18 shows the average restoration results in

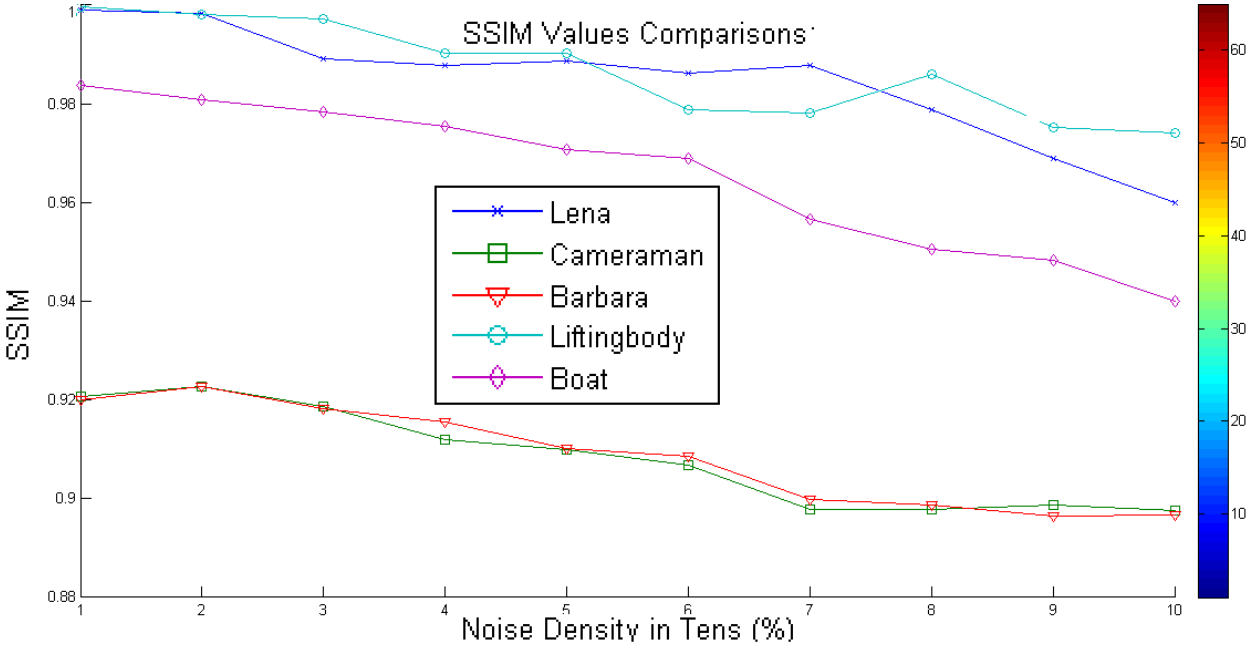


Figure 9.15: Graph for **SIR** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% random valued noises

terms of SSIM for all the filters considered. Five proposed filters and six existing filters are compared extensively in this table. SMF filter performs worse compared to other filters. AMF, DWM and EDBA have given similar results for the entire noise density range. IDBA and BDND obtained similar PSNR values also. However BDND performs best among them for the higher noise densities. The proposed DWMD performs not so good in terms of SSIM compared to the BDND and IDBA filters. It performs better than EDBA filter, whereas the ANDWP and VMM perform better than all the existing filters. The EPR performs much better than all the filters of the table. The **SIR** performs the best in the table in terms of SSIM for the entire noise density range considered. The corresponding graph of the table is shown in Fig. 9.18. The performances of proposed filters are compared with the existing filters for color image as well. Color *Lena* image is applied to the six existing filters and to the five proposed filters here. The noise percentage considered is 50% because of medium noise corruption by the random valued noise. Fig. 9.19 represents the restored images to show the qualitative performances of the filters for the image. In the figure, a and b represent the original and 50% noisy color *Lena* images respectively with Random Valued Impulse noise. Figures c, d, e, f, g and h represent the corresponding restored images using SMF, AMF DWM, AMF, EDBA, IDBA and BDND filters respectively. Figures i, j, k, l and m show the restored images by the proposed filters such as **DWMD, ANDWP,**

Table 9.16: Comparison of restoration results for the proposed algorithms in *Average PSNR* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density with RVN

Filter	**Noise Density/Average PSNR(dB)**									
	10	20	30	40	50	60	70	80	90	100
SMF	30.72	29.59	27.44	24.65	21.57	19.39	18.55	17.13	16.54	16.11
AMF	31.30	30.13	29.30	28.09	26.71	24.59	22.33	19.04	17.76	16.71
DWM	31.51	30.82	30.31	29.54	27.33	25.70	23.41	21.91	19.12	17.04
EDBA	32.09	30.54	30.10	29.12	27.01	24.66	23.81	21.71	19.32	17.23
IDBA	32.24	31.01	30.44	28.20	26.32	24.70	23.03	20.12	18.65	17.49
BDND	32.54	31.12	30.40	29.32	27.67	25.71	23.01	21.00	19.54	18.10
DWMD	32.81	31.46	30.42	28.80	26.71	25.19	23.17	21.78	20.84	19.35
ANDWP	33.62	32.42	31.07	29.41	27.47	25.65	23.58	21.86	21.09	19.77
VMM	33.90	32.66	31.63	29.72	27.91	26.11	24.37	22.46	21.75	20.72
EPR	34.16	32.69	31.88	30.13	28.20	26.38	24.74	22.91	22.24	21.11
SIR	34.73	33.48	32.37	30.53	28.83	26.80	25.05	23.30	22.68	21.78

Table 9.17: Comparison of restoration results for the proposed algorithms in *Average IEF* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density with RVN

Filter	**Noise Density/Average IEF**									
	10	20	30	40	50	60	70	80	90	100
SMF	51.77	34.56	28.98	17.65	14.44	10.91	8.01	7.88	7.12	5.01
AMF	82.87	67.99	67.12	51.81	38.73	35.13	27.39	23.91	19.91	15.19
DWM	131.99	120.82	104.83	81.73	59.71	39.70	21.39	20.11	18.97	18.89
EDBA	169.61	139.01	135.92	115.10	79.18	69.99	62.99	47.22	40.63	32.91
IDBA	201.50	187.88	165.61	139.40	116.33	100.49	81.71	62.81	59.51	51.99
BDND	231.90	220.11	201.91	169.40	142.01	109.11	90.90	86.25	82.10	76.51
DWMD	243.49	229.09	211.68	194.12	173.91	153.04	118.89	107.34	92.37	81.18
ANDWP	292.11	265.94	256.01	233.98	220.43	190.05	160.70	135.30	111.74	101.05
VMM	301.24	274.20	260.68	249.23	235.42	220.16	191.24	159.44	124.52	111.43
EPR	324.84	293.67	273.15	264.04	250.97	229.63	213.03	189.50	156.01	133.09
SIR	346.35	315.79	287.69	276.24	263.20	251.64	230.00	211.90	180.66	151.88

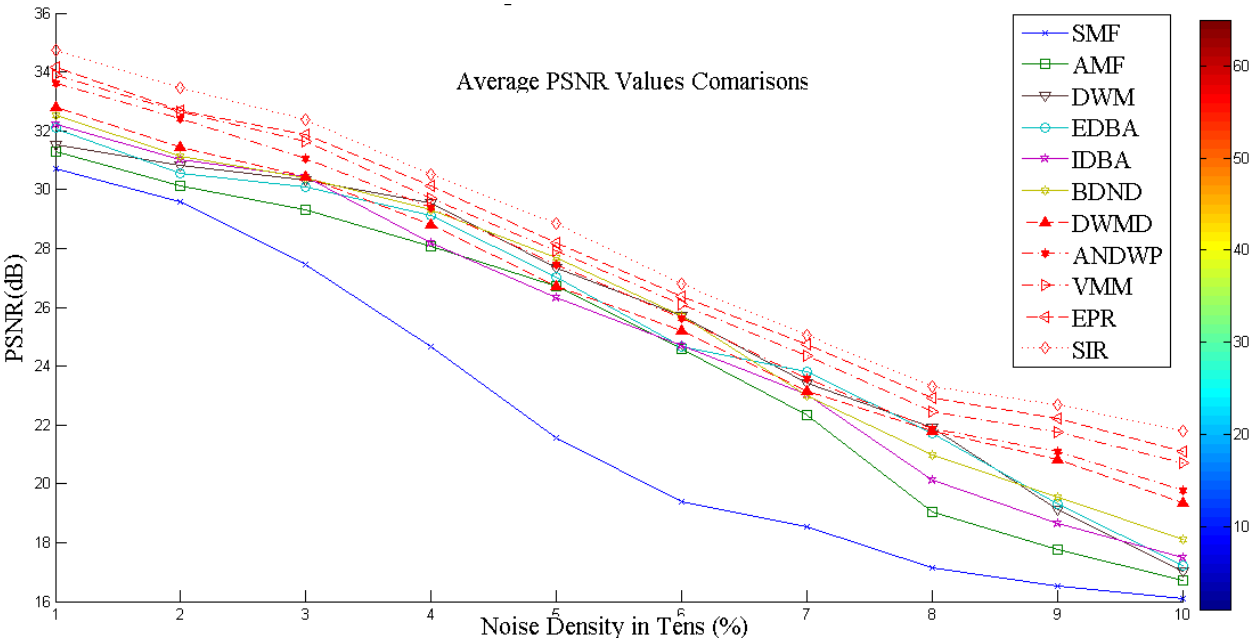


Figure 9.16: Graph showing average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filters

Table 9.18: Comparison of restoration results for the proposed algorithms in *Average SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density with RVN

Filter	**Noise Density/Average SSIM**									
	10	20	30	40	50	60	70	80	90	100
SMF	0.9362	0.9333	0.9319	0.9287	0.9259	0.9116	0.9087	0.9076	0.9012	0.8981
AMF	0.9412	0.9377	0.9329	0.9310	0.9254	0.9150	0.9076	0.9064	0.9059	0.9045
DWM	0.9420	0.9412	0.9343	0.9312	0.9240	0.9211	0.9150	0.9219	0.9136	0.9107
EDBA	0.9449	0.9440	0.9361	0.9322	0.9315	0.9243	0.9222	0.9217	0.9156	0.9130
IDBA	0.9510	0.9508	0.9455	0.9431	0.9319	0.9311	0.9276	0.9233	0.9176	0.9167
BDND	0.9512	0.9503	0.9430	0.9411	0.9355	0.9330	0.9310	0.9270	0.9235	0.9214
DWMD	0.9489	0.9460	0.9387	0.9348	0.9327	0.9296	0.9257	0.9235	0.9172	0.9150
ANDWP	0.9527	0.9491	0.9444	0.9408	0.9377	0.9324	0.9282	0.9232	0.9179	0.9123
VMM	0.9577	0.9561	0.9517	0.9479	0.9426	0.9377	0.9351	0.9338	0.9280	0.9265
EPR	0.9626	0.9623	0.9579	0.9537	0.9501	0.9456	0.9418	0.9402	0.9343	0.9314
SIR	0.9646	0.9645	0.9603	0.9562	0.9539	0.9498	0.9440	0.9424	0.9375	0.9337

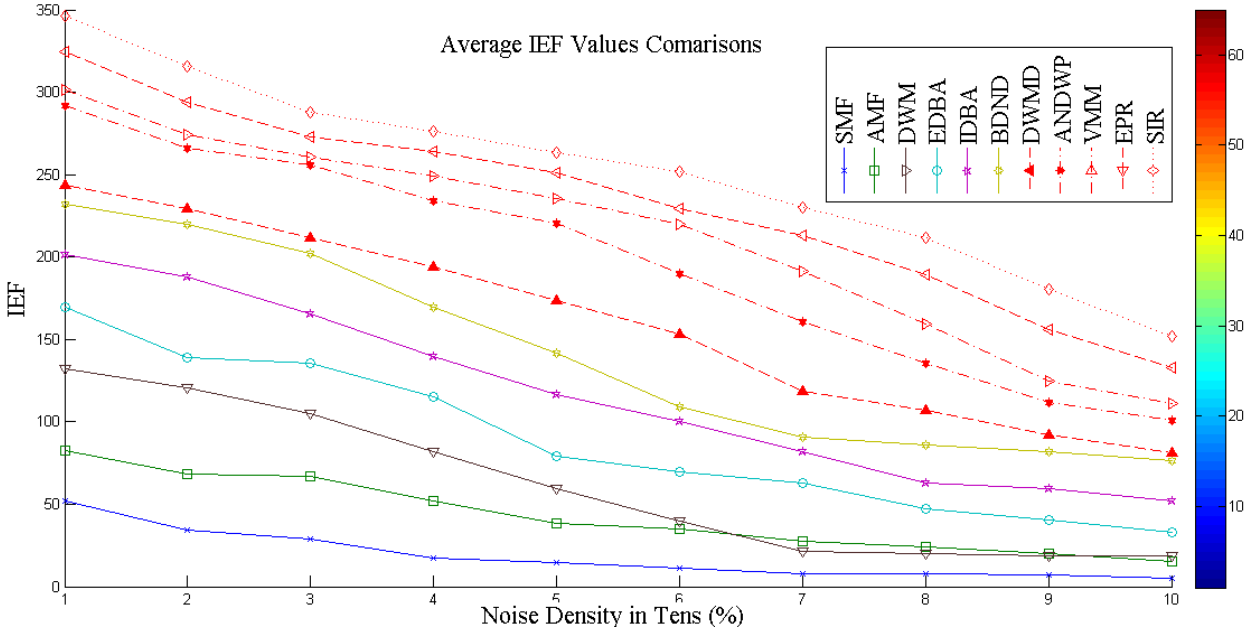


Figure 9.17: Graph showing average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filters

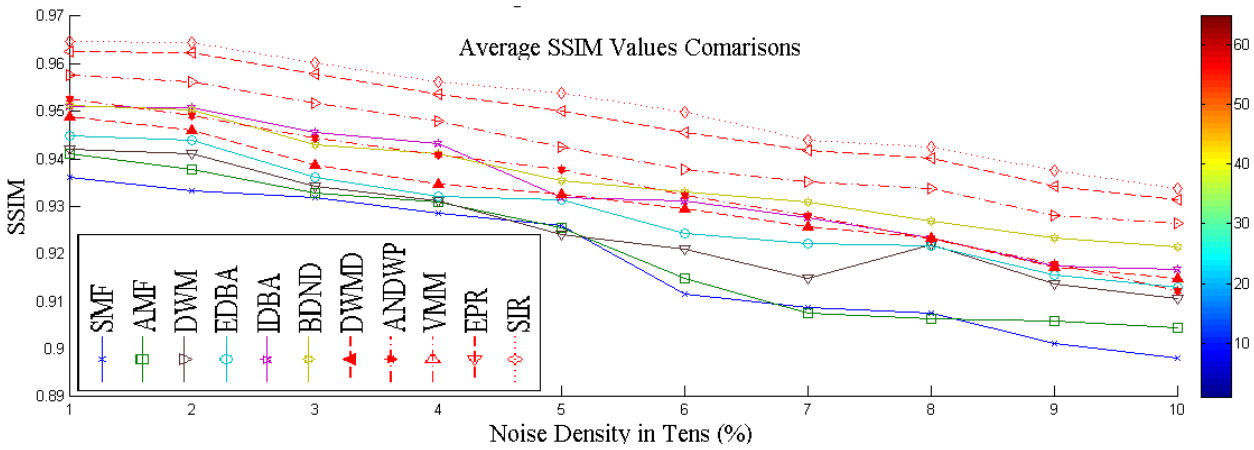


Figure 9.18: Graph showing average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filters

VMM, EPR and **SIR** respectively. From the figure, it can be observed that SMF and AMF perform very poor. A large amount of black patches exist there. The DWM, AMF, EDBA and IDBA filters improve the restoration results gradually. Although noises are not suppressed completely while the edges are well preserved. The proposed *DWMD, ANDWP,*

VMM, EPR and **SIR** filters also improve the quality of the restoration image. The EPR and SIR filters suppress the noises to some extent and preserves the image fine textures as well.

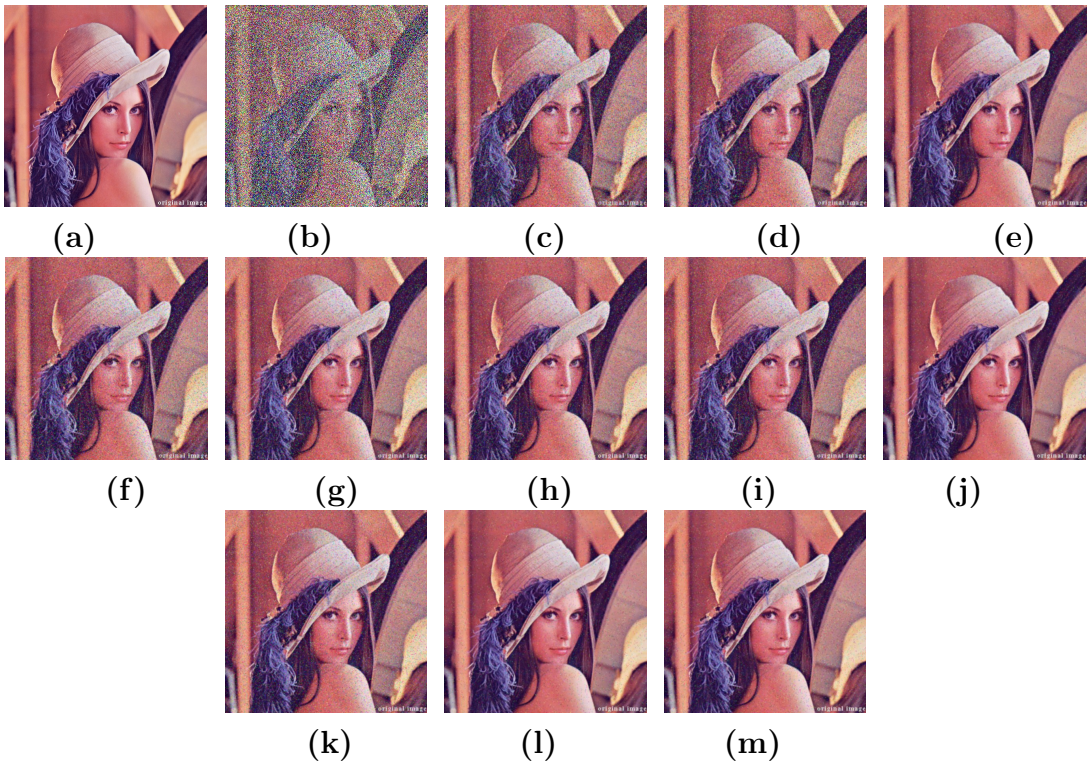


Figure 9.19: a-b represent the original and 50% noisy color *Lena* images with RVN. Figures c-h represent the corresponding restored color images by SMF, AMF, DWM, EDBA, IDBA and BDND. Figures i-m show the restored images by proposed **DWMD, ANDWP, VMM, EPR and SIR**

.

9.2 Filter Optimization

In this section, the optimizations of the *Sub-Image Restoration (SIR)* filter are validated using restoration results. The optimizations have been done using *Genetic Algorithm (GA)* and *Particle Swarm Optimization (PSO)* techniques. The *SIR* operator has been presented in section 3.2.3 and the optimization techniques using the *GA* and *PSO* are given in chapter 4. Three user parameters of the SIR filter are searched in a wide range by the two optimization algorithms. Best method of the previous section i.e., *SIR* is optimized in this section by using GA and PSO. Other methods are not used for optimization here but they can also be optimized. The user parameters for each of the proposed filter are discussed with the corresponding algorithm. In a specified range of values they can be searched using any optimization technique to get the optimal restoration results. By using any of the optimization algorithms, the program can determine the value the user parameters automatically or semi-automatically with less interaction with the user. In a specified range, the algorithm can search the optimal restoration results during the execution. The Genetic Algorithm based optimization and Particle Swarm Optimization algorithms are implemented and the corresponding result sets are generated to compare with the previously proposed filters and with the state of the art filters. The GA based restoration results are given in section 9.2.1 and of PSO based in section 9.2.2. The GA based and PSO based optimization algorithms obtained similar restoration results although the PSO is much efficient because its fast convergence rate. Other reason is, the PSO uses less number of parameters during execution. Using GA and PSO, the three user parameters are searched in the following ranges to obtain the global optimal solutions i.e., maximum number of Iterations (I) $\in$ [3, 8], Threshold value (T) $\in$ [300, 1000] and decreasing Rate of threshold (R) $\in$ [0.6, 0.95].

9.2.1 Based on Genetic Algorithm

Genetic Algorithm based Sub-Image Restoration (GASIR) filter (section 4.1.1) executes for five standard digital images which are corrupted by random valued noises with 10% to 100% densities and restoration results are generated. The restoration results in terms of PSNR (dB) for those images are presented in table 9.19. From the table, it is observed that the PSNR values for all the five images are consistent for the entire noise density range. From low to high noise density, the average PSNR values are also improving accordingly. The visual representation of the graph is given in Fig. 9.20. The restoration results in terms of IEF by the five bench mark images corrupted by random valued noise are given in table 9.20. Seeing the values from the table for the entire noise density range, it can be stated that, except the *Cameraman* image, the other images obtained good results in terms of IEF. The average IEF values for the noise densities can be noticed to conform the improvement of the result easily. The visual representation of the graph is given in Fig. 9.21. Lastly the

restoration results in terms of SSIM by the GASIR method for those images are presented in table 9.21. From the table, it is observed that the SSIM values for all the five images are consistent for the entire noise density range. Except Barbara and Cameraman images, the other images are performing well as the SSIM values are close to one. From low to high noise density, the average SSIM values are also improving. The visual representation of the graph is given in Fig. 9.22.

Table 9.19: PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **GASIR** filter for varying random valued noises

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
Lena	36.52	34.98	33.71	32.65	31.71	29.34	26.87	23.07	21.70	20.13
Cameraman	31.42	31.12	28.90	26.07	23.89	22.31	20.79	18.45	18.36	17.70
Barbara	34.69	33.45	32.79	30.22	28.09	26.31	24.83	22.67	21.97	21.43
Liftingbody	40.46	38.15	37.62	36.65	34.09	31.88	31.29	29.41	28.76	28.31
Boat	33.89	32.23	30.41	29.32	28.40	26.83	25.90	27.75	24.53	23.69
Average	35.39	33.98	32.68	30.98	29.23	27.33	25.93	24.27	23.06	22.25

Table 9.20: IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **GASIR** filter for varying random valued noises

Image	*Noise Density/IEF*									
	10	20	30	40	50	60	70	80	90	100
Lena	426.20	369.65	322.25	307.50	292.70	285.91	273.09	242.71	217.25	149.75
Cameraman	283.71	257.90	236.73	197.26	190.25	161.50	157.77	151.90	132.75	119.16
Barbara	312.76	286.76	252.09	231.22	216.45	202.40	179.51	166.37	149.50	137.78
Liftingbody	527.46	508.76	482.35	467.66	460.65	438.25	404.55	382.95	342.76	311.80
Boat	311.75	297.22	239.70	232.60	212.35	207.25	186.35	181.10	172.01	137.00
Average	372.37	344.05	306.62	287.24	274.48	259.06	240.25	225.00	202.85	171.09

9.2.2 Based on Particle Swarm Optimization

Particle Swarm Optimization based Sub-Image Restoration (PSOSIR) filter (section 4.2.1) runs for five standard digital images such as *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% noise density which are corrupted by random valued noises and restoration results are generated. The restoration results in terms of PSNR (dB) for those

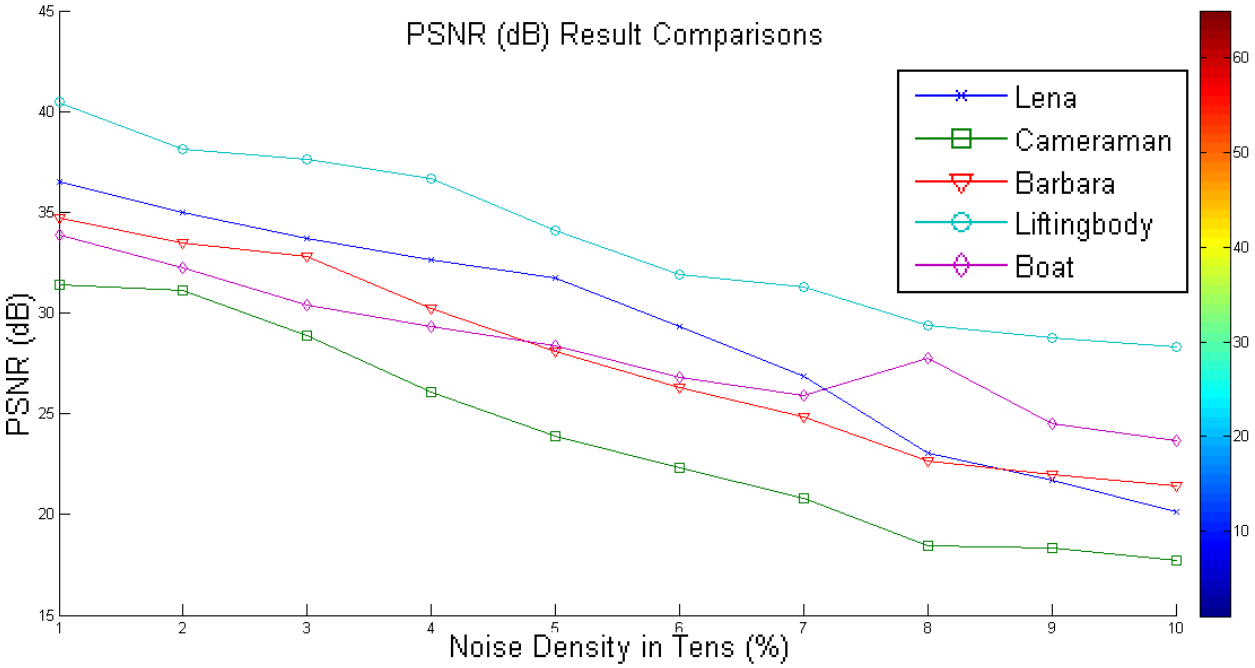


Figure 9.20: Graph for **GASIR** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% impulses on RVN

Table 9.21: SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **GASIR** filter for varying random valued noises

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
Lena	0.9993	0.9990	0.9962	0.9920	0.9900	0.9881	0.9879	0.9843	0.9722	0.9616
Cameraman	0.9339	0.9320	0.9277	0.9222	0.9167	0.9136	0.9112	0.9083	0.9006	0.8990
Barbara	0.9342	0.9317	0.9304	0.9267	0.9222	0.9103	0.9041	0.9003	0.8972	0.8970
Liftingbody	0.9997	0.9987	0.9976	0.9951	0.9942	0.9898	0.9878	0.9863	0.9757	0.9751
Boat	0.9846	0.9823	0.9817	0.9801	0.9786	0.9781	0.9705	0.9591	0.9534	0.9486
Average	0.9703	0.9687	0.9667	0.9632	0.9603	0.9560	0.9523	0.9477	0.9398	0.9363

images are presented in table 9.22. From the table, it is seen that the PSNR values for all the five images are consistent for the entire noise density range. From low to high noise density, the average PSNR values are also improving with a sharp interval. The visual representation of the graph is given in Fig. 9.23. The restoration results in terms of IEF by the five bench mark images corrupted by random valued noise are given in table 9.23. The values from the table prove that for the entire noise density, except the *Cameraman* image, the other images

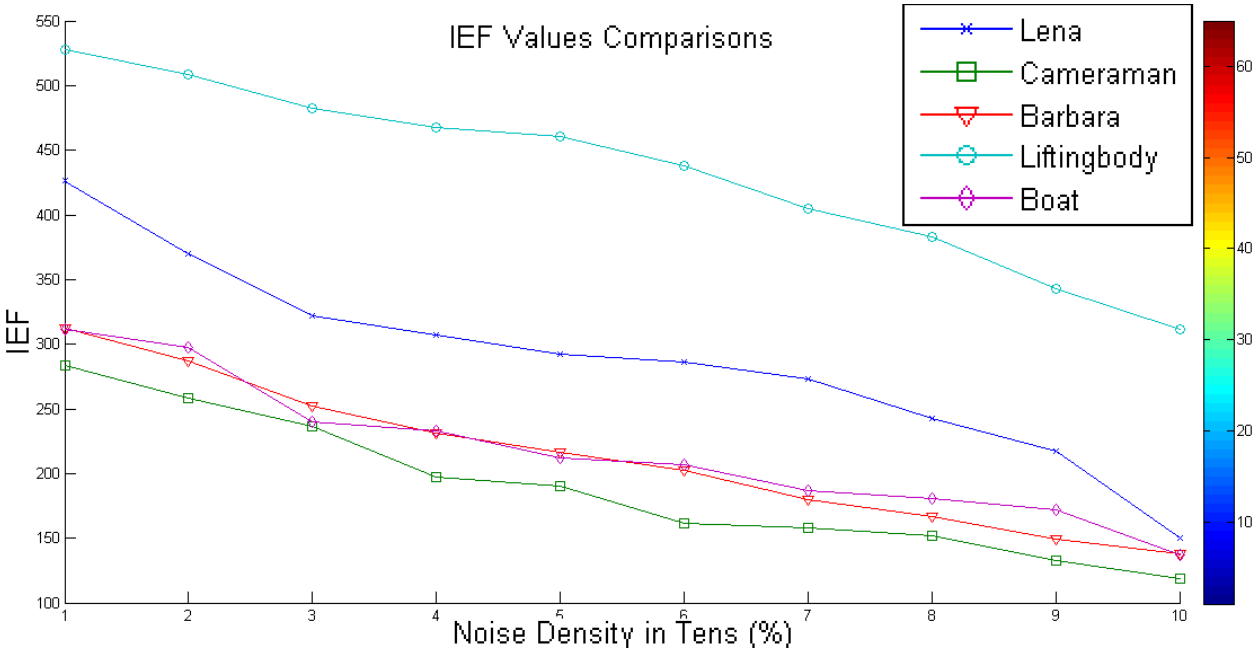


Figure 9.21: Graph for **GASIR** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% impulses on RVN

obtained good results in terms of IEF. The average IEF values for the noise densities can be noticed to understand the improvement of the result easily. The visual representation of the graph is given in Fig. 9.24. The restoration results in terms of SSIM by the PSOSIR method for those images are presented in table 9.24. From the table, it is observed that the SSIM values for all the five images are consistent for the entire noise density range. Except the Barbara and Cameraman images, the other images are performing well as the SSIM values are close to one. From low to high noise density, the average SSIM values are also improving. The visual representation of the graph is given in Fig. 9.25.

9.2.3 Comparisons

Table 9.25 shows the average restoration results in terms of PSNR for seven proposed filters and six existing filters. *GASIR and PSOSIR* based restoration results are under consideration in this section. The other proposed methods are already been compared and discussed in the previous section. These two filters are compared extensively which is given in this table. The average PSNR is also computed and given in the table. The *GASIR and PSOSIR* perform much better than the other proposed filters with an wide margin although the *GASIR and PSOSIR* obtained similar restoration results. The comparisons among thirteen

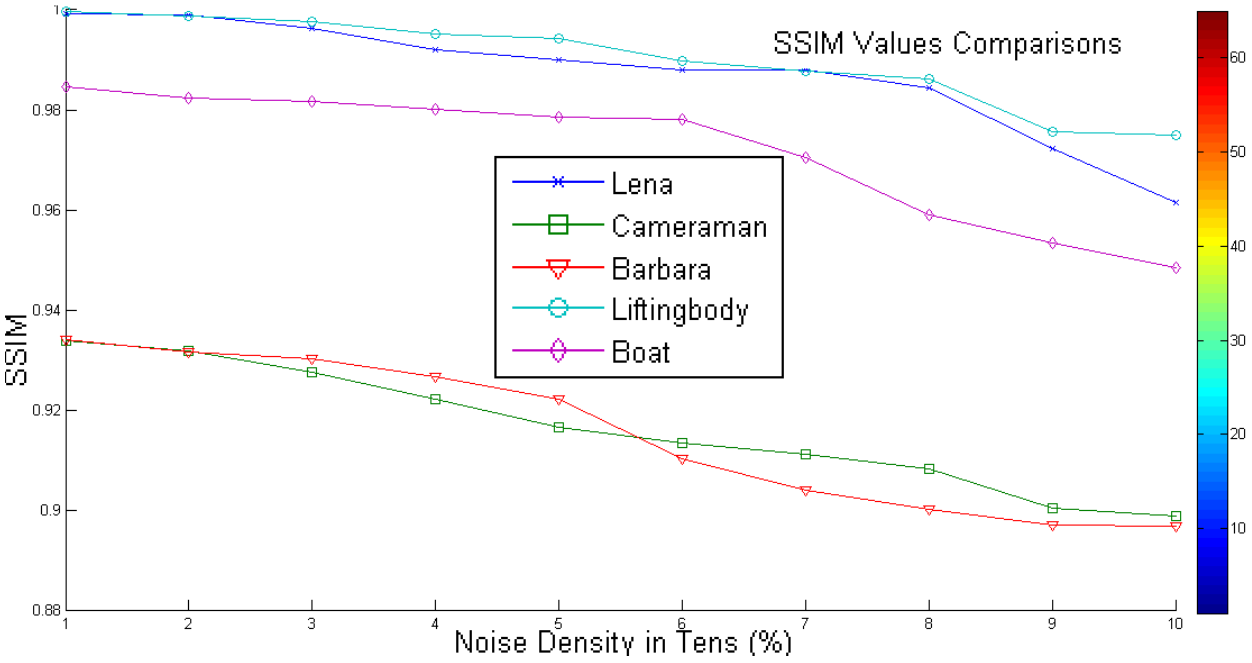


Figure 9.22: Graph for **GASIR** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

Table 9.22: PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **PSOSIR** filter for varying random valued noises

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
Lena	36.65	35.12	33.97	32.79	31.86	29.45	27.11	23.57	22.09	20.45
Cameraman	31.55	31.34	29.13	26.45	23.96	22.50	20.92	18.61	18.38	17.81
Barbara	34.73	33.50	32.80	30.32	28.13	26.37	24.91	22.88	22.12	21.67
Liftingbody	40.51	38.37	37.78	36.69	34.42	31.92	31.37	29.44	28.91	28.72
Boat	34.20	32.27	30.70	29.47	28.47	26.89	26.22	25.82	24.64	23.82
Average	35.52	34.12	32.87	31.14	29.36	27.42	26.10	24.46	23.22	22.49

filters are graphically represented in Fig. 9.26. Table 9.26 shows the average restoration results in terms of IEF for the proposed filters and the state of the art filters. *GASIR* and *PSOSIR* based restoration results are compared in this section. These two filters are compared extensively in this table. The *GASIR and PSOSIR* perform much better than the other proposed filters with a solid margin although they have obtained similar restoration

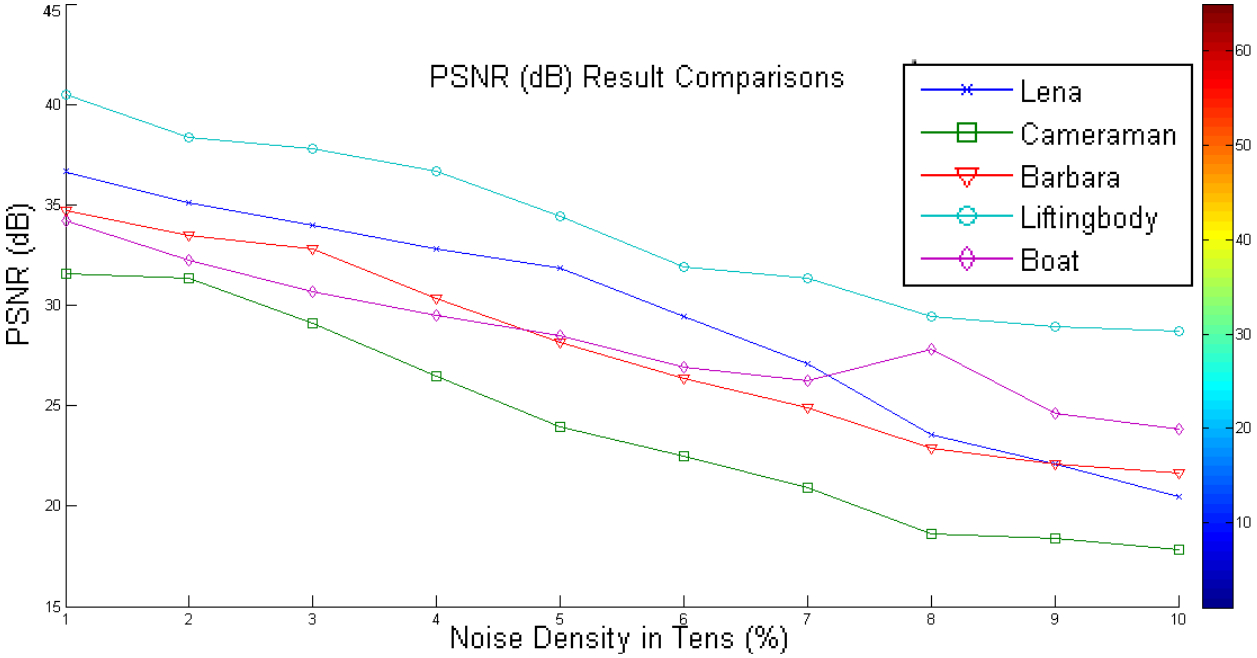


Figure 9.23: Graph for **PSOSIR** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

Table 9.23: IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **PSOSIR** filter for varying random valued noises

Image	*Noise Density/IEF*									
	10	20	30	40	50	60	70	80	90	100
Lena	430.45	377.60	328.52	317.51	296.02	291.22	278.22	244.22	234.17	157.50
Cameraman	287.17	264.50	243.25	202.54	191.87	168.65	161.79	157.36	138.21	121.50
Barbara	352.66	289.70	260.62	235.25	227.50	207.32	184.35	171.67	156.70	142.25
Liftingbody	529.70	511.86	482.66	478.25	464.11	440.50	407.87	387.44	351.25	322.43
Boat	323.17	307.50	256.16	248.22	245.54	222.83	195.71	191.17	176.60	143.73
Average	384.63	350.23	314.24	296.35	285.00	266.10	245.58	230.37	211.38	177.48

results. The comparisons among the thirteen filters are graphically represented in Fig. 9.27. Average SSIM values are also generated by the two algorithms in this section. Table 9.27 shows the average restoration results in terms of SSIM for the proposed filters and the state of the art filters. *GASIR* and *PSOSIR* based restoration results are compared to other in this section. The *GASIR* and *PSOSIR* perform better than the other proposed filters however they obtained similar restoration results. The comparison among the thirteen filters are

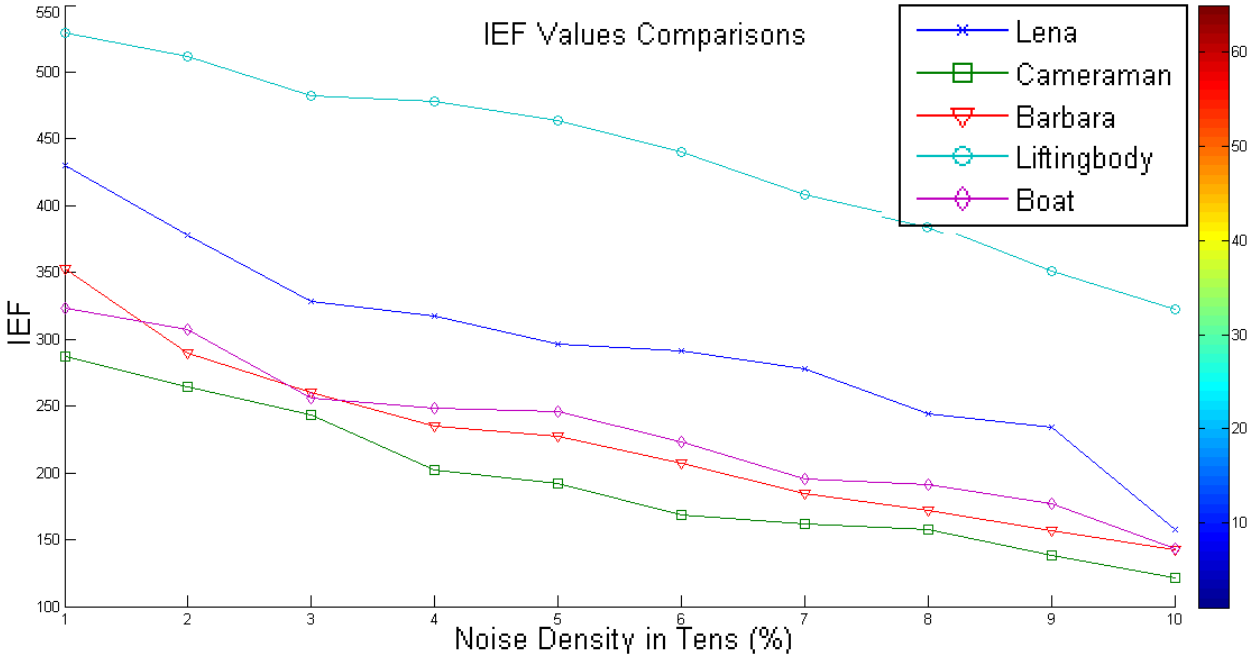


Figure 9.24: Graph for **PSOSIR** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% impulses on RVN

Table 9.24: SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **PSOSIR** filter for varying random valued noises

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
Lena	0.9994	0.9992	0.9986	0.9965	0.9957	0.9916	0.9900	0.9843	0.9812	0.9717
Cameraman	0.9359	0.9334	0.9313	0.9322	0.9270	0.9210	0.9179	0.9086	0.9041	0.9008
Barbara	0.9351	0.9343	0.9333	0.9298	0.9254	0.9210	0.9207	0.9136	0.9072	0.9065
Liftingbody	0.9997	0.9987	0.9976	0.9951	0.9942	0.9898	0.9878	0.9863	0.9757	0.9751
Boat	0.9846	0.9823	0.9817	0.9801	0.9786	0.9781	0.9705	0.9591	0.9534	0.9486
Average	0.9709	0.9696	0.9685	0.9667	0.9642	0.9603	0.9574	0.9504	0.9443	0.9405

graphically represented in Fig.9.28. The proposed filters are compared with the existing filters for color image also. Color *Lena* image is applied to the six existing filters and to seven proposed filters here. The noise density used is 50% with random valued noise. Fig. 9.29 shows the restored images to show the qualitative performances of the filters for the 50% noisy color *Lena* image. In figure 9.29, c to h represent the restored images by the filters SMF, AMF DWM, AMF, EDBA, IDBA and BDND. Figures i - m show the restored

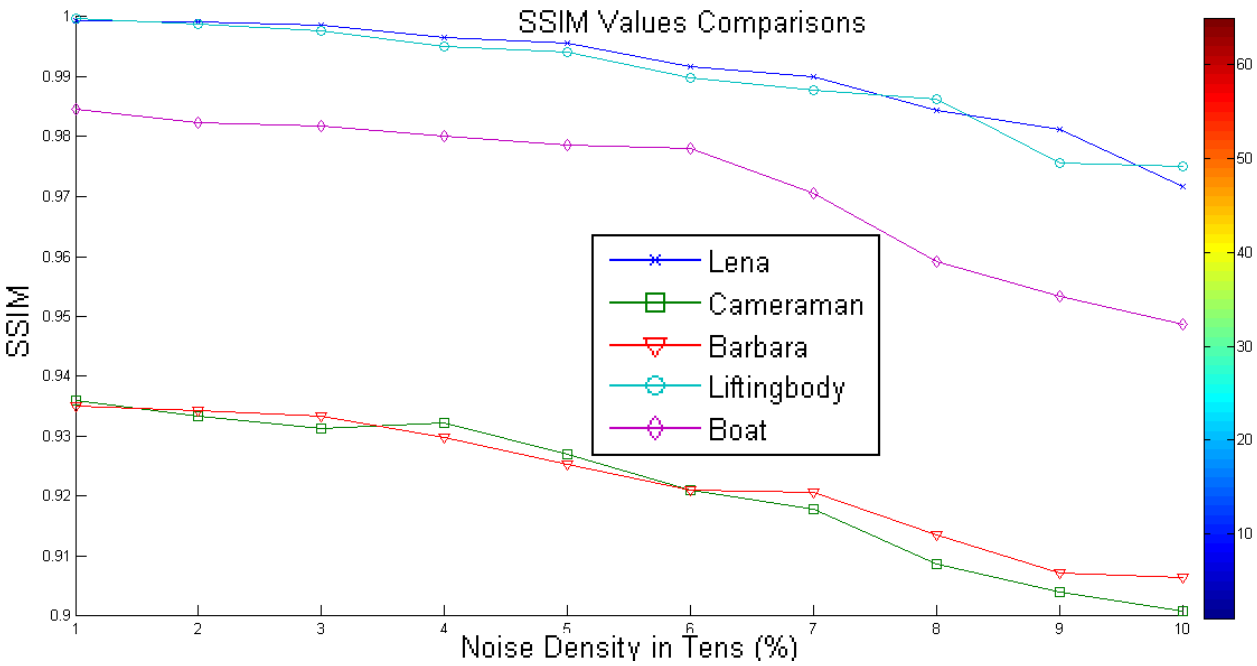


Figure 9.25: Graph for **PSOSIR** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% impulses on RVN

images by the proposed filters such as DWMD, ANDWP, VMM, EPR and SIR, discussed in previous sections. Figures n and o show the restored images by *GASIR* and *PSOSIR* respectively. It can be seen from these two figures that these filters can remove the noises very well and also preserves the image fine textures, compared to other outputs by the filters.

Table 9.25: Comparison of proposed techniques in *Average PSNR* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with varying random valued noises

Filter	**Noise Density/Average PSNR(dB)**									
	10	20	30	40	50	60	70	80	90	100
SMF	30.72	29.59	27.44	24.65	21.57	19.39	18.55	17.13	16.54	16.11
AMF	31.30	30.13	29.30	28.09	26.71	24.59	22.33	19.04	17.76	16.71
DWM	31.51	30.82	30.31	29.54	27.33	25.70	23.41	21.91	19.12	17.04
EDBA	32.09	30.54	30.10	29.12	27.01	24.66	23.81	21.71	19.32	17.23
IDBA	32.24	31.01	30.44	28.20	26.32	24.70	23.03	20.12	18.65	17.49
BDND	32.54	31.12	30.40	29.32	27.67	25.71	23.01	21.00	19.54	18.10
DWMD	32.81	31.46	30.42	28.80	26.71	25.19	23.17	21.78	20.84	19.35
ANDWP	33.62	32.42	31.07	29.41	27.47	25.65	23.58	21.86	21.09	19.77
VMM	33.90	32.66	31.63	29.72	27.91	26.11	24.37	22.46	21.75	20.72
EPR	34.16	32.69	31.88	30.13	28.20	26.38	24.74	22.91	22.24	21.11
SIR	34.73	33.48	32.37	30.53	28.83	26.80	25.05	23.30	22.68	21.78
GASIR	35.39	33.98	32.68	30.98	29.23	27.33	25.93	24.27	23.06	22.25
PSOSIR	35.52	34.12	32.87	31.14	29.36	27.42	26.10	24.46	23.22	22.49

Table 9.26: Comparison of results for the proposed filters in *Average IEF* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with varying RVNs

Filter	**Noise Density/Average IEF**									
	10	20	30	40	50	60	70	80	90	100
SMF	51.77	34.56	28.98	17.65	14.44	10.91	8.01	7.88	7.12	5.01
AMF	82.87	67.99	67.12	51.81	38.73	35.13	27.39	23.91	19.91	15.19
DWM	131.99	120.82	104.83	81.73	59.71	39.70	21.39	20.11	18.97	18.89
EDBA	169.61	139.01	135.92	115.10	79.18	69.99	62.99	47.22	40.63	32.91
IDBA	201.50	187.88	165.61	139.40	116.33	100.49	81.71	62.81	59.51	51.99
BDND	231.90	220.11	201.91	169.40	142.01	109.11	90.90	86.25	82.10	76.51
DWMD	243.49	229.09	211.68	194.12	173.91	153.04	118.89	107.34	92.37	81.18
ANDWP	292.11	265.94	256.01	233.98	220.43	190.05	160.70	135.30	111.74	101.05
VMM	301.24	274.20	260.68	249.23	235.42	220.16	191.24	159.44	124.52	111.43
EPR	324.84	293.67	273.15	264.04	250.97	229.63	213.03	189.50	156.01	133.09
SIR	346.35	315.79	287.69	276.24	263.20	251.64	230.00	211.90	180.66	151.88
GASIR	372.37	344.05	306.62	287.24	274.48	259.06	240.25	225.00	202.85	171.09
PSOSIR	384.63	350.23	314.24	296.35	285.00	266.10	245.58	230.37	211.38	177.48

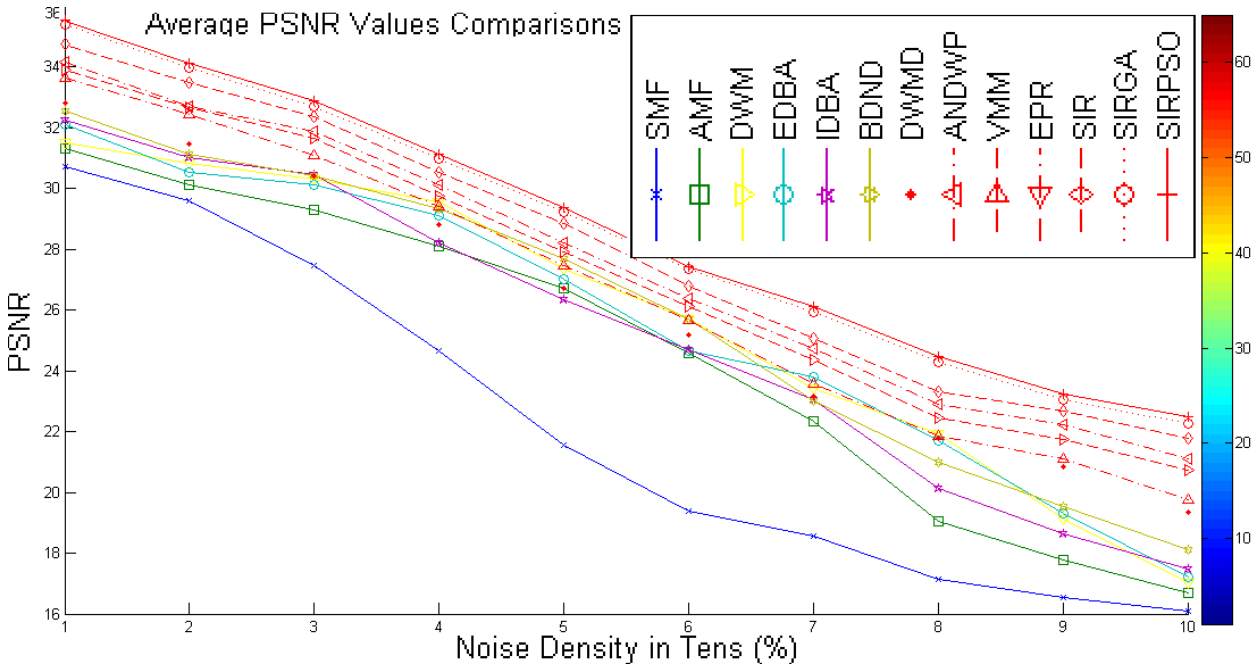


Figure 9.26: Graph for average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filter

Table 9.27: Comparison of results for proposed filters in *Average SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* with varying RVNs

Filter	**Noise Density/Average SSIM**									
	10	20	30	40	50	60	70	80	90	100
SMF	0.9362	0.9333	0.9319	0.9287	0.9259	0.9116	0.9087	0.9076	0.9012	0.8981
AMF	0.9412	0.9377	0.9329	0.9310	0.9254	0.9150	0.9076	0.9064	0.9059	0.9045
DWM	0.9420	0.9412	0.9343	0.9312	0.9240	0.9211	0.9150	0.9219	0.9136	0.9107
EDBA	0.9449	0.9440	0.9361	0.9322	0.9315	0.9243	0.9222	0.9217	0.9156	0.9130
IDBA	0.9510	0.9508	0.9455	0.9431	0.9319	0.9311	0.9276	0.9233	0.9176	0.9167
BDND	0.9512	0.9503	0.9430	0.9411	0.9355	0.9330	0.9310	0.9270	0.9235	0.9214
DWMD	0.9489	0.9460	0.9387	0.9348	0.9327	0.9296	0.9257	0.9235	0.9172	0.9150
ANDWP	0.9527	0.9491	0.9444	0.9408	0.9377	0.9324	0.9282	0.9232	0.9179	0.9123
VMM	0.9577	0.9561	0.9517	0.9479	0.9426	0.9377	0.9351	0.9338	0.9280	0.9265
EPR	0.9626	0.9623	0.9579	0.9537	0.9501	0.9456	0.9418	0.9402	0.9343	0.9314
SIR	0.9646	0.9645	0.9603	0.9562	0.9539	0.9498	0.9440	0.9424	0.9375	0.9337
GASIR	0.9703	0.9687	0.9667	0.9632	0.9603	0.9560	0.9523	0.9477	0.9398	0.9363
PSOSIR	0.9709	0.9696	0.9685	0.9667	0.9642	0.9603	0.9574	0.9504	0.9443	0.9405

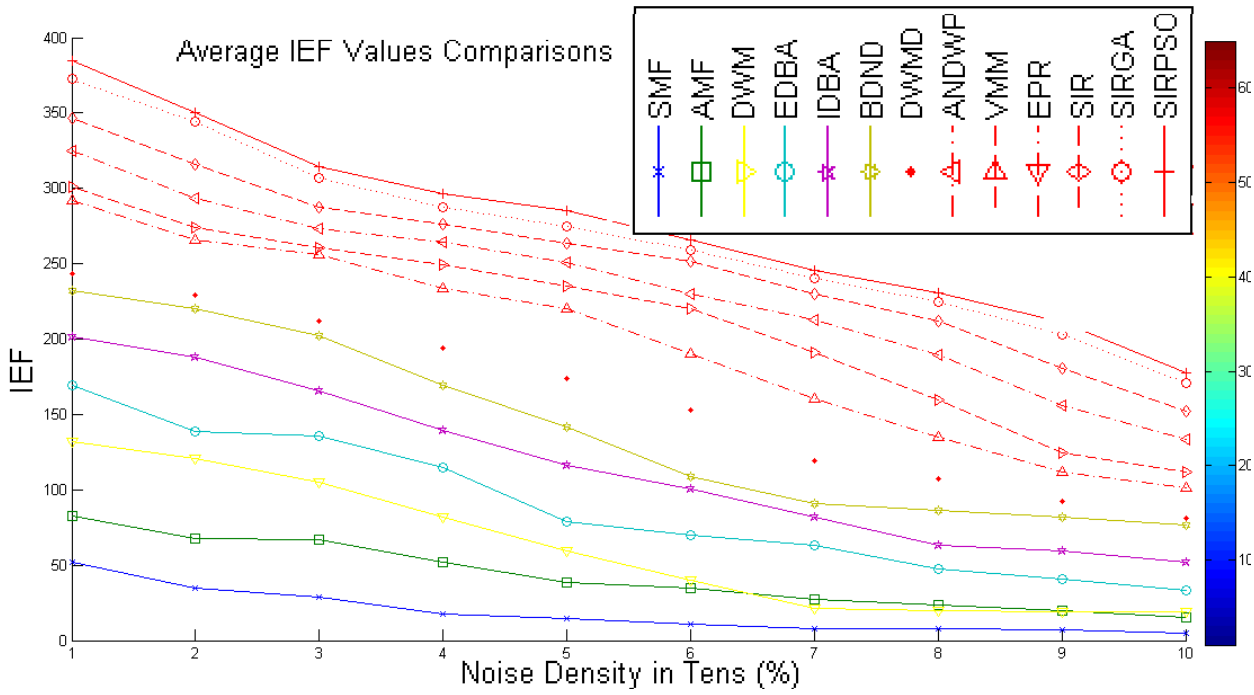


Figure 9.27: Graph of average IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filter

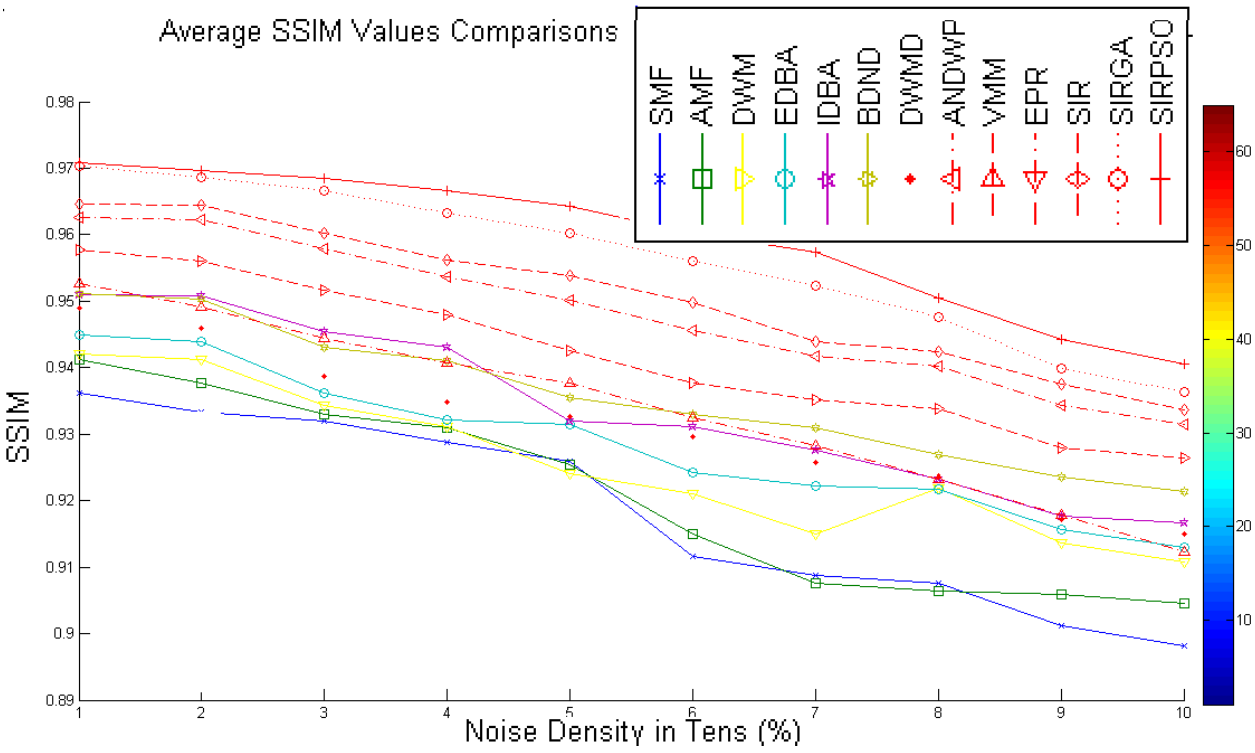


Figure 9.28: Graph of average SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and proposed filter

Figure 9.29: a-b show original and 50% noisy color *Lena* images with RVN. Figures c-h represent the corresponding restored images by SMF, AMF, DWM, EDBA, IDBA and BDND. Figures i-o show restored images by proposed DWMD, ANDWP, VMM, EPR, SIR, **GASIR and PSOSIR**

9.3 Noise Density Estimation for Adaptive Median Filter

In section 9.2, denoising with optimization algorithm is validated through experimental results. The SIR filter has been optimized for the purpose of optimal restoration results. Some user parameters are searched in a wide range using Genetic Algorithm and Particle Swarm Optimization techniques. The problem of all such algorithms is selection of constant window size for noise detection and filtering. The filtering window size should be varied to the density of the noise. A large window size may be useful for highly corrupted images, whereas for low noisy image small size window is applied. A constant window size is not an efficient filtering technique of noises in the digital images.

In chapter 5, the noise density estimation for denoising technique has been presented. In the section, the noise density of an image has been estimated and subsequently a proper window size is suggested for better restoration results. In this section, two noise models are considered for making the noisy images. Those are *Random Valued Noise (RVN)* and *Salt and Pepper Noise (SPN)*. The filters such as *DWMD, ANDWP, VMM, EPR, SIR, GASIR* and *PSOSIR* are proposed to denoise only *RVN* in the digital images. In section 9.3.1, restoration results on both the noise models are generated with the standard bench mark images. In section 9.3.2, the comparison of the proposed filter **NDE** with the existing algorithms in terms of PSNR (dB), IEF and SSIM are given. The comparative restoration results on the color *Lena* image is given at the end of the section.

9.3.1 Results

The **Noise Density Estimation for Denoising (NDE)** filter executes for five standard digital images which are corrupted by random valued noises and salt and pepper noises with a varying noise density of 10% to 100% and restoration results are generated. The restoration results in terms of PSNR (dB) for those images are presented in table 9.28. From the table 9.28((1) SPN), it is observed that the PSNR values for all the five images are consistent for the entire noise density range. From high to low noise density, the average PSNR values are also improving consistently. The visual representation of the graph is given in Fig. 9.30. From the table 9.28((2) RVN), it is observed that the PSNR values for all the five images are consistent for the entire noise density range. From high to low noise density, the average PSNR values are also improving consistently. In both the cases, the Cameraman image suffers with low PSNRs, but for other images the restoration results in terms PSNR are satisfactory. The visual representation of the graph is given in Fig. 9.31. The restoration results in terms of IEF by the five bench mark images corrupted by random valued noise and salt and pepper noise are given in table 9.29. Seeing the values given in table 9.29((1) SPN, (2) RVN) for the entire noise density, it can be stated that except the

Cameraman image, the other images obtained good results in terms of IEF. The average IEF values for the noise densities can be noticed to ensure the improvement of the result easily. The visual representation of the graph is given in Fig. 9.32 and Fig. 9.33. For Liftingbody image, the PSNRs obtained is maximum compared to other images. Lastly the restoration results in terms of SSIM using the **NDE** method for those images are presented in table 9.28. From both sections the table, it is observed that the SSIM values for all the five images are consistent for the entire noise density range for RVN and SPN. Except Barbara and cameraman images, for the other images the algorithm also performs well as the SSIM values are close to one. From high to low noise density, the average SSIM values are also improving persistently. The visual representations of the graph are given in Fig. 9.34 and Fig. 9.35. For Liftingbody and Lena images the restoration results in terms of SSIM, the proposed algorithm obtained excellent results.

Table 9.28: PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **NDE** filter for varying SPNs and RVNs

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
(1) SPN										
Lena	38.12	36.26	35.73	34.58	33.41	32.32	30.65	29.21	26.78	24.23
Cameraman	32.55	31.56	30.06	27.79	26.97	25.93	24.85	22.73	21.66	20.65
Barbara	36.23	34.09	32.97	31.25	30.78	27.50	26.21	24.56	24.21	22.77
Liftingbody	41.10	38.88	38.17	37.21	35.89	33.23	32.15	30.18	29.61	26.71
Boat	35.41	32.91	31.17	29.67	28.73	27.32	26.98	25.49	24.69	24.11
Average	36.68	34.74	33.62	32.10	31.15	29.26	28.16	26.43	25.39	23.69
(2) RVN										
Lena	37.42	35.87	34.52	32.98	32.23	30.22	27.67	24.25	22.76	21.12
Cameraman	31.75	31.42	29.77	27.13	24.92	22.76	21.21	19.72	18.65	18.05
Barbara	35.23	33.87	32.92	30.76	29.21	26.54	25.26	22.91	22.42	21.80
Liftingbody	40.87	38.65	37.87	36.70	34.62	32.33	31.97	29.67	29.17	28.86
Boat	34.50	32.65	31.13	29.52	28.52	27.21	26.76	25.46	24.67	23.90
Average	35.95	34.49	33.24	31.41	29.90	27.81	26.57	24.40	23.53	22.74

9.3.2 Comparisons

In this section, the proposed technique is compared with the existing algorithms in terms of PSNR(dB), IEF and SSIM. For both the noise models, the proposed algorithm **NDE** is

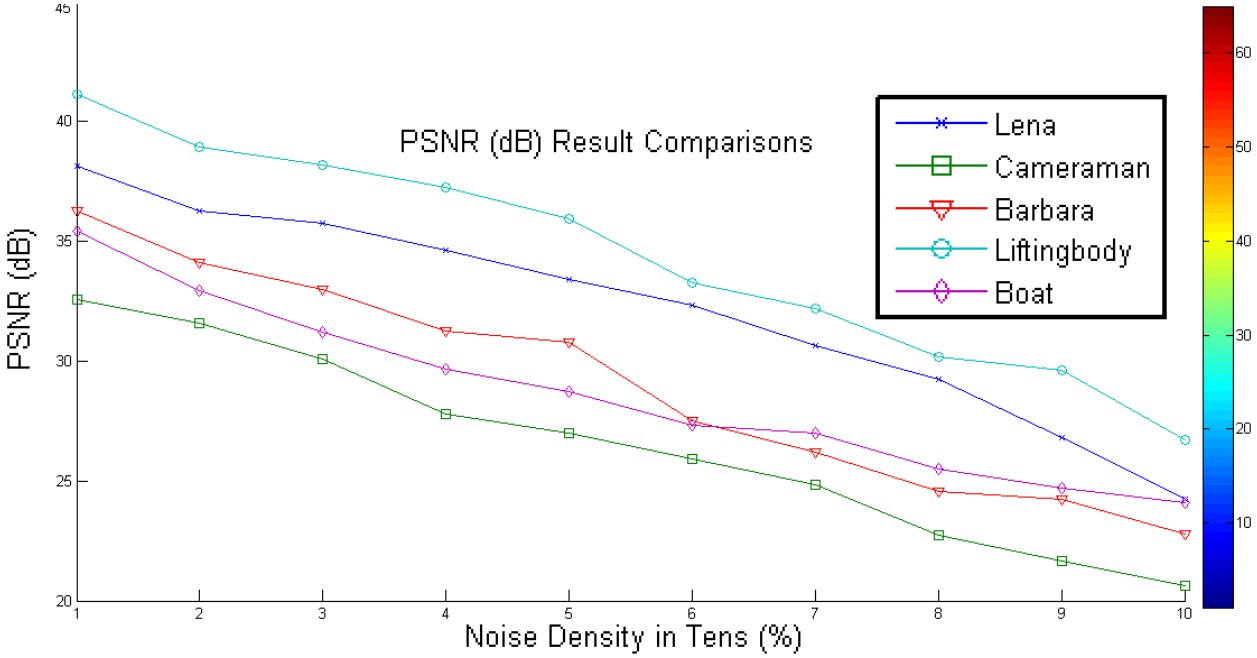


Figure 9.30: Graph for **NDE** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs

validated. In table 9.31, the values obtained from proposed algorithm is given from where a comparison can be made with SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table all results are generated on the images corrupted by SPN. The *Average PSNR* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it is seen that for the entire noise density, the proposed algorithm **NDE** performs the best in terms of PSNR. The comparative results of the **NDE** and existing seven algorithms can also be seen in the Fig. 9.36.

In table 9.32, the PSNR values obtained by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table, the other proposed algorithms which are compared in the previous sections are also kept in the table. In this table all results are generated on the images corrupted by RVN. The average PSNR values obtained are given in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with varying noise density from 10% to 100%. From this table, it can be shown that for the entire noise density, the proposed algorithm **NDE** performs the best in terms of PSNR. The comparative results of the **NDE** and existing seven algorithms can also be seen in the Fig. 9.37. The FBDA filter performs better than ANDWP filter. From the figure, it is clear that the **NDE** algorithm obtained much better restoration results than the existing algorithms

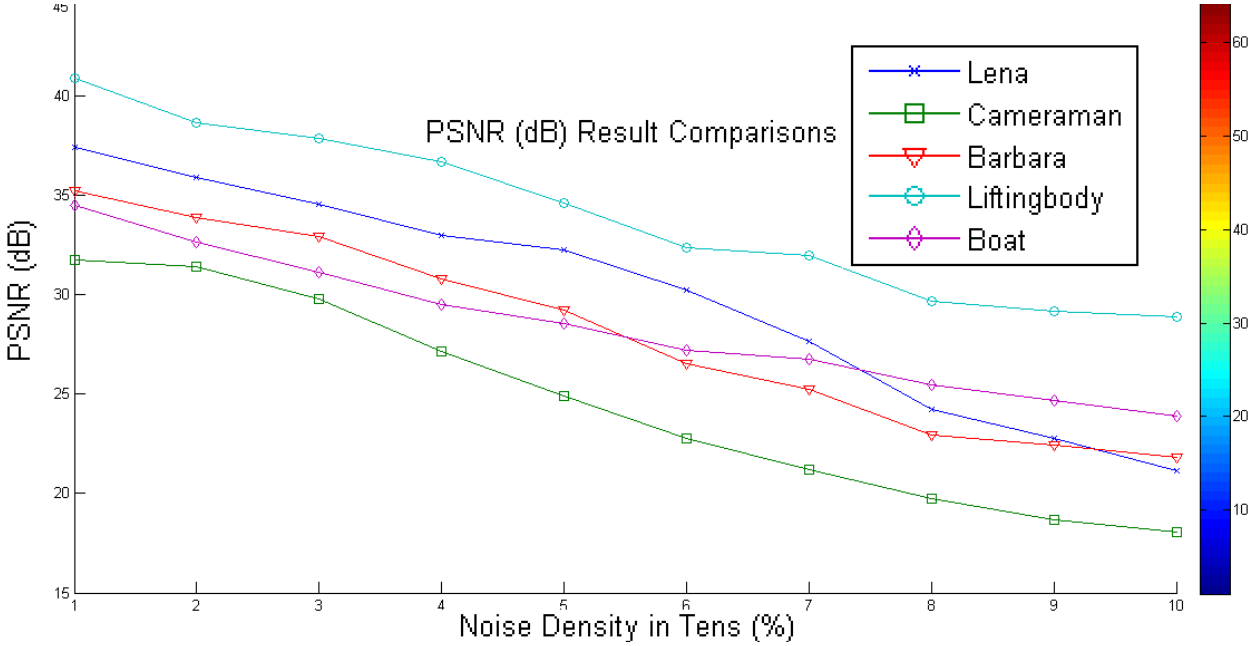


Figure 9.31: Graph for **NDE** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

for the entire noise density. The algorithm also obtained better PSNR than other proposed filters such as DWMD, ANDWP, VMM, EPR, SIR, GASIR and PSOSIR. In table 9.33, the restoration results obtained by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table all results are generated on the images corrupted by SPN. The *Average IEF* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it is noticed that for the entire noise density, the proposed algorithm **NDE** performs the best in terms of IEF. The SMF and AMF perform very poor. The comparative results of the **NDE** and existing seven algorithms can also be seen in the Fig. 9.38.

In table 9.34, the IEF values obtained from the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table, the other proposed algorithms which are compared in the previous sections are also kept in the table. In this table, all results are generated on the images corrupted by RVN. The *Average IEF* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it can be observed that for the entire noise density, the proposed algorithm **NDE** performs excellent in terms of IEF. The FBDA filter performs better than ANDWP filter. The results of comparison of the **NDE**

Table 9.29: IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **NDE** filter for varying SPNs and RVNs

Image	*Noise Density/IEF*									
	10	20	30	40	50	60	70	80	90	100
(1) SPN										
Lena	446.76	412.75	384.95	356.45	313.80	305.61	297.70	279.55	282.75	208.97
Cameraman	316.75	302.50	262.76	240.50	232.78	223.90	201.50	187.25	161.75	156.52
Barbara	387.59	303.50	289.25	276.50	245.25	237.95	206.71	192.75	184.88	167.88
Liftingbody	542.76	523.45	507.50	496.50	472.25	455.76	423.50	399.70	365.89	333.65
Boat	332.75	323.54	287.98	260.87	253.90	231.55	200.22	198.25	190.92	165.22
Average	405.32	373.14	346.48	326.16	303.59	290.95	265.92	251.50	237.23	206.44
(2) RVN										
Lena	437.30	383.71	353.50	331.99	297.81	294.91	286.44	276.05	243.80	161.36
Cameraman	292.41	267.22	252.00	213.81	203.60	192.00	176.61	160.86	141.27	149.25
Barbara	363.43	291.30	273.22	246.10	229.95	221.54	197.11	183.40	166.00	154.00
Liftingbody	532.03	517.50	487.08	483.88	468.76	446.00	411.31	393.11	357.39	328.93
Boat	329.50	313.71	263.11	255.63	249.46	227.91	199.50	197.35	181.32	153.87
Average	390.93	354.68	325.78	306.28	289.91	276.47	254.19	242.15	217.95	189.48

Table 9.30: SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **NDE** filter for varying SPNs and RVNs

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
(1) SPN										
Lena	0.9995	0.9992	0.9993	0.9981	0.9957	0.9932	0.9915	0.9874	0.9867	0.9840
Cameraman	0.9443	0.9412	0.9387	0.9356	0.9341	0.9336	0.9308	0.9323	0.9260	0.9260
Barbara	0.9462	0.9443	0.9422	0.9391	0.9362	0.9330	0.9309	0.9276	0.9233	0.9207
Liftingbody	0.9997	0.9990	0.9981	0.9965	0.9947	0.9906	0.9885	0.9884	0.9876	0.9844
Boat	0.9896	0.9866	0.9859	0.9822	0.9803	0.9791	0.9749	0.9721	0.9701	0.9680
Average	0.9759	0.9741	0.9728	0.9703	0.9682	0.9659	0.9633	0.9616	0.9587	0.9566
(2) RVN										
Lena	0.9994	0.9992	0.9987	0.9971	0.9954	0.9919	0.9904	0.9850	0.9832	0.9830
Cameraman	0.9364	0.9379	0.9351	0.9336	0.9312	0.9304	0.9279	0.9266	0.9243	0.9238
Barbara	0.9400	0.9387	0.9372	0.9365	0.9343	0.9300	0.9276	0.9243	0.9211	0.9200
Liftingbody	0.9997	0.9988	0.9980	0.9957	0.9941	0.9899	0.9883	0.9875	0.9861	0.9821
Boat	0.9851	0.9843	0.9838	0.9802	0.9795	0.9790	0.9723	0.9672	0.9643	0.9623
Average	0.9721	0.9718	0.9706	0.9686	0.9669	0.9642	0.9613	0.9581	0.9558	0.9542

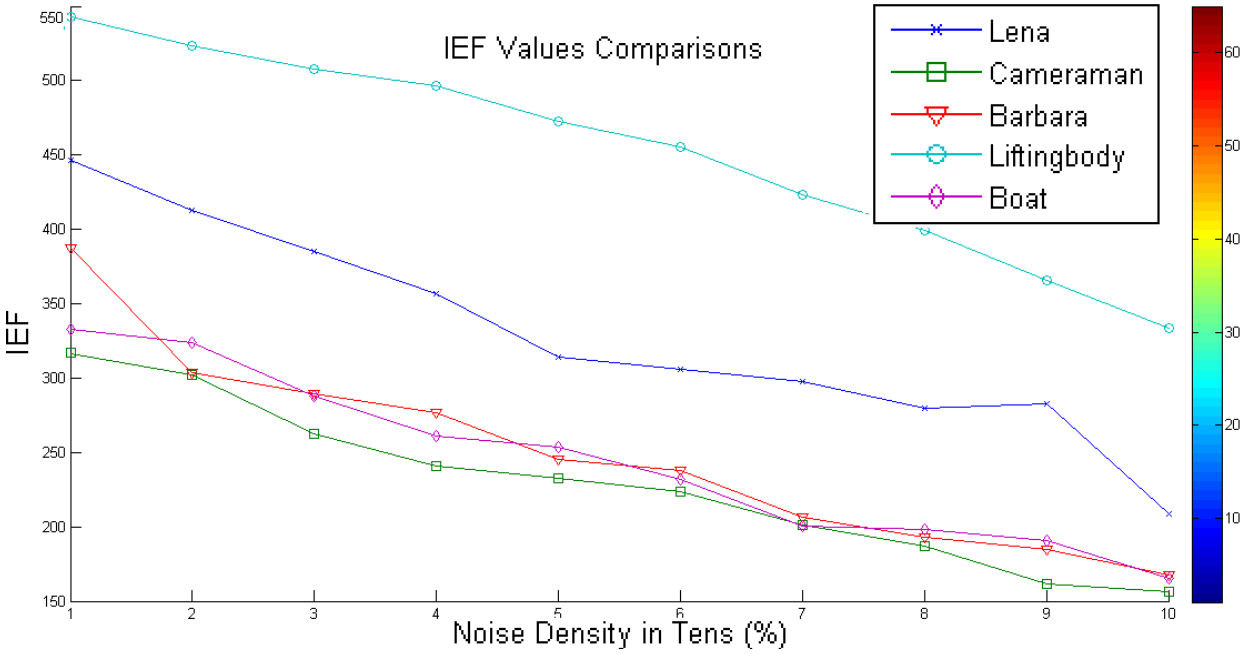


Figure 9.32: Graph for **NDE** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs

Table 9.31: Comparison of results for the proposed filters in *Average PSNR* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs

Filter	**Noise Density/Average PSNR(dB)**									
	10	20	30	40	50	60	70	80	90	100
SMF	32.22	30.56	28.51	26.76	24.59	22.76	20.40	18.89	17.57	16.87
AMF	32.87	30.79	28.76	28.45	26.89	24.81	22.47	19.19	17.88	16.90
DWM	33.65	31.35	30.87	29.66	27.63	25.93	24.09	22.11	19.45	17.63
EDBA	34.31	32.87	31.60	30.50	28.00	26.60	25.30	23.69	20.44	18.33
IDBA	35.29	33.70	31.79	31.10	29.50	27.11	25.54	24.25	20.32	19.06
BDND	35.41	33.69	31.50	31.32	30.79	30.00	26.32	25.21	21.25	20.62
FBDA	35.49	34.03	31.63	31.65	31.03	30.41	27.23	25.30	22.78	21.20
NDE	36.68	34.74	33.62	32.10	31.15	29.26	28.16	26.43	25.39	23.69

and existing seven algorithms are shown in the Fig. 9.39. From the figure, it can be cleared that the **NDE** algorithm obtained much better restoration results in terms of IEF than the existing algorithms for the entire noise density. The algorithm also has given better IEF than

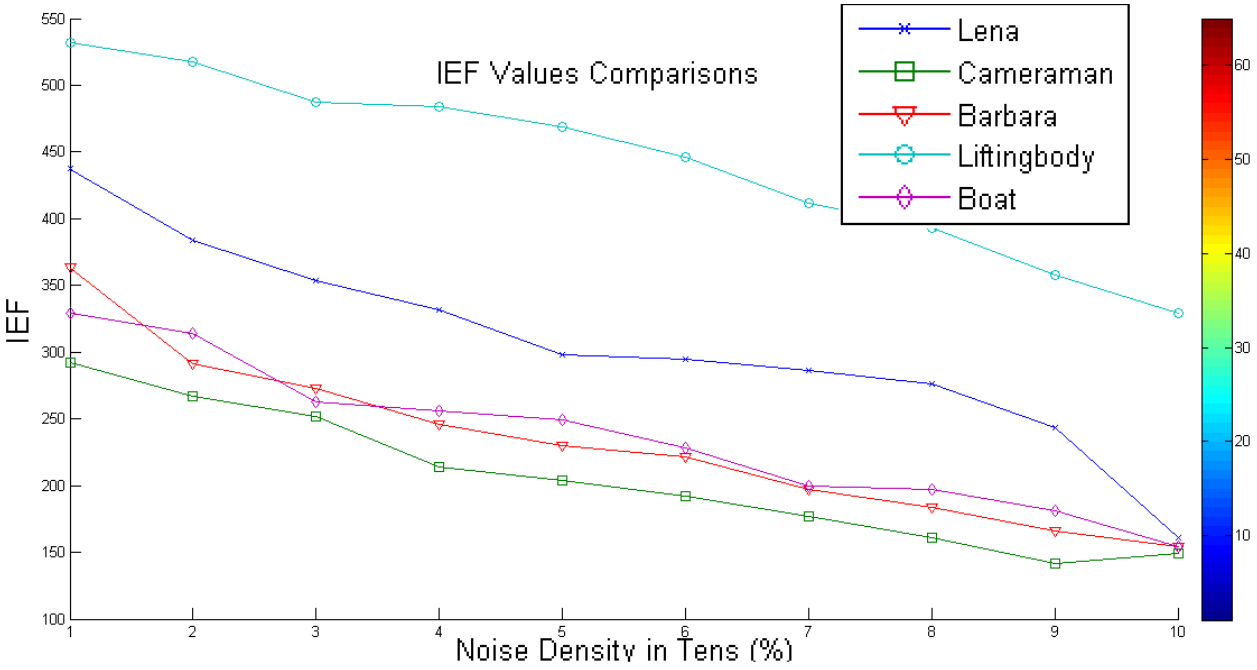


Figure 9.33: Graph for **NDE** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

other proposed filters such as DWMD, ANDWP, VMM, EPR, SIR, GASIR and PSOSIR. In table 9.35, the performances of the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table all results are generated on the images corrupted by SPN. The *Average SSIM* values for all the algorithms are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it can be noticed that for the entire noise density, the proposed algorithm **NDE** performs the best in terms of SSIM. The SMF and AMF perform very poor where as DWM, EDBA and IDBA perform similarly. The BDND and FBDA also have obtained similar SSIM values. The comparative results of the **NDE** and existing seven algorithms can also be seen in the Fig. 9.40.

In table 9.36, the results given by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table, the other proposed algorithms which are compared in the previous sections are also kept in the table. In this table, all results are generated on the images corrupted by RVN. The *Average SSIM* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it can be shown that for the entire noise density, the proposed **NDE** performs excellent in terms of SSIM. The FBDA filter performs

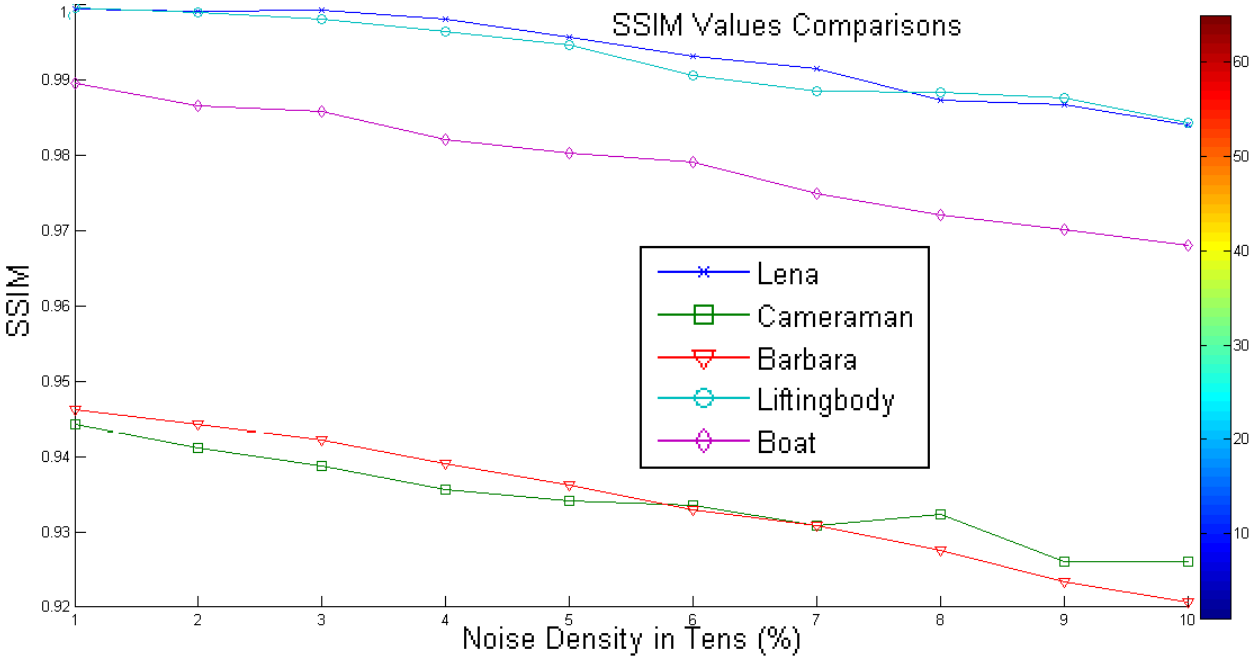


Figure 9.34: Graph for **NDE** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs

better than VMM filter. The comparative results of the **NDE** and existing seven algorithms can also be seen in the Fig. 9.41. From the figure, it can be cleared that the **NDE** algorithm obtained much better restoration results in terms of SSIM than the existing algorithms for the entire noise density. The algorithm has given better SSIM values than other proposed filters such as DWMD, ANDWP, VMM, EPR, SIR, GASIR and PSOSIR.

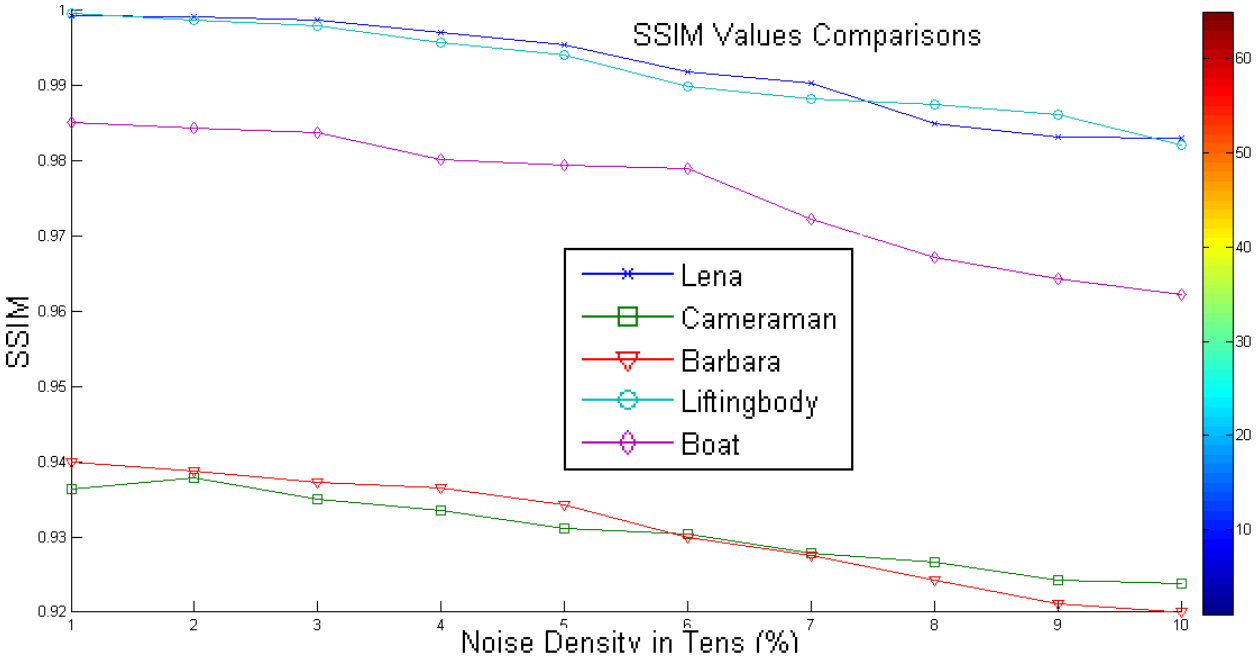


Figure 9.35: Graph for **NDE** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

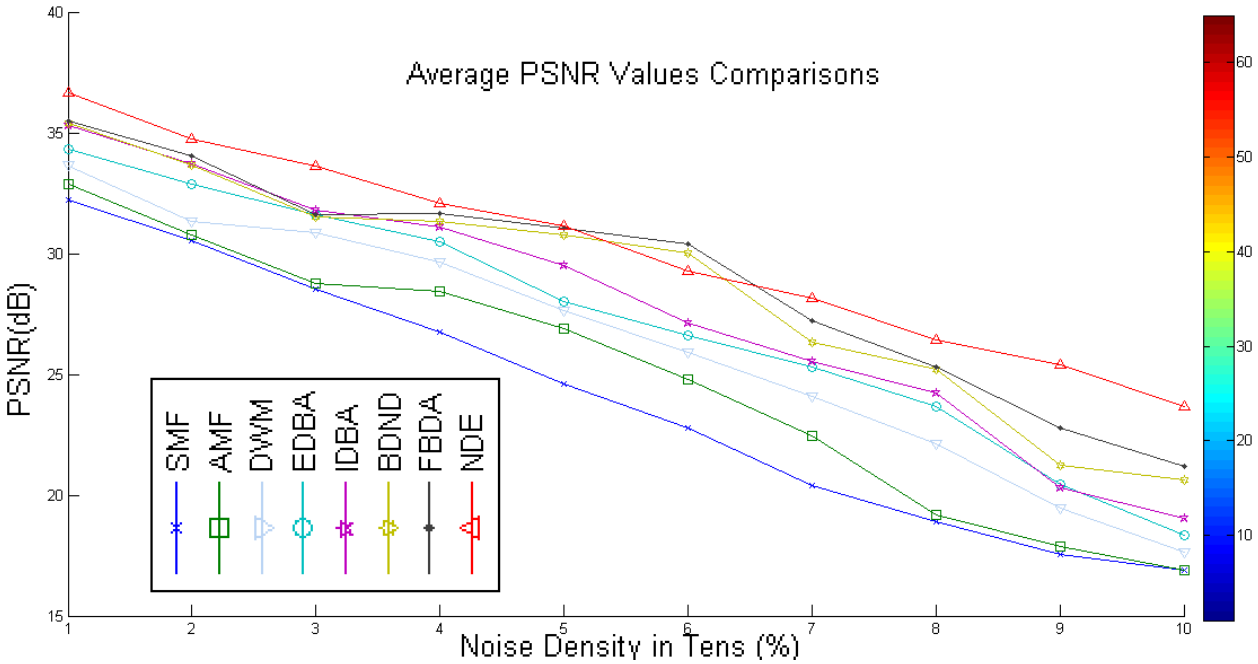


Figure 9.36: Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and NDE

Table 9.32: Comparison of the proposed filters in *Average PSNR* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs

Filter	**Noise Density/Average PSNR(dB)**									
	10	20	30	40	50	60	70	80	90	100
SMF	30.72	29.59	27.44	24.65	21.57	19.39	18.55	17.13	16.54	16.11
AMF	31.30	30.13	29.30	28.09	26.71	24.59	22.33	19.04	17.76	16.71
DWM	31.51	30.82	30.31	29.54	27.33	25.70	23.41	21.91	19.12	17.04
EDBA	32.09	30.54	30.10	29.12	27.01	24.66	23.81	21.71	19.32	17.23
IDBA	32.24	31.01	30.44	28.20	26.32	24.70	23.03	20.12	18.65	17.49
BDND	32.54	31.12	30.40	29.32	27.67	25.71	23.01	21.00	19.54	18.10
FBDA	33.62	31.90	31.22	30.40	28.76	26.79	24.92	23.55	21.79	19.70
DWMD	32.81	31.46	30.42	28.80	26.71	25.19	23.17	21.78	20.84	19.35
ANDWP	33.62	32.42	31.07	29.41	27.47	25.65	23.58	21.86	21.09	19.77
VMM	33.90	32.66	31.63	29.72	27.91	26.11	24.37	22.46	21.75	20.72
EPR	34.16	32.69	31.88	30.13	28.20	26.38	24.74	22.91	22.24	21.11
SIR	34.73	33.48	32.37	30.53	28.83	26.80	25.05	23.30	22.68	21.78
GASIR	35.39	33.98	32.68	30.98	29.23	27.33	25.93	24.27	23.06	22.25
PSOSIR	35.52	34.12	32.87	31.14	29.36	27.42	26.10	24.46	23.22	22.49
NDE	35.95	34.49	33.24	31.41	29.90	27.81	26.57	24.40	23.53	22.74

Table 9.33: Comparison of the proposed filters in *Average IEF* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs

Filter	**Noise Density/Average IEF**									
	10	20	30	40	50	60	70	80	90	100
SMF	56.82	48.89	41.76	32.50	27.25	21.00	17.60	14.76	10.00	7.75
AMF	63.81	50.50	46.00	41.56	37.09	31.20	29.22	25.50	23.10	18.00
DWM	144.75	127.25	114.56	86.33	69.70	47.89	40.25	29.17	21.00	18.91
EDBA	197.50	156.50	144.25	127.60	92.64	72.90	67.08	53.75	44.72	40.00
IDBA	223.55	204.32	190.75	165.80	145.90	123.43	93.25	77.94	65.60	54.10
BDND	256.80	249.96	221.09	202.95	178.60	176.25	123.55	111.17	103.04	92.81
FBDA	295.10	263.72	266.50	242.55	237.01	210.50	176.69	143.50	130.56	115.95
NDE	405.32	373.14	346.48	326.16	303.59	290.95	265.92	251.50	237.23	206.44

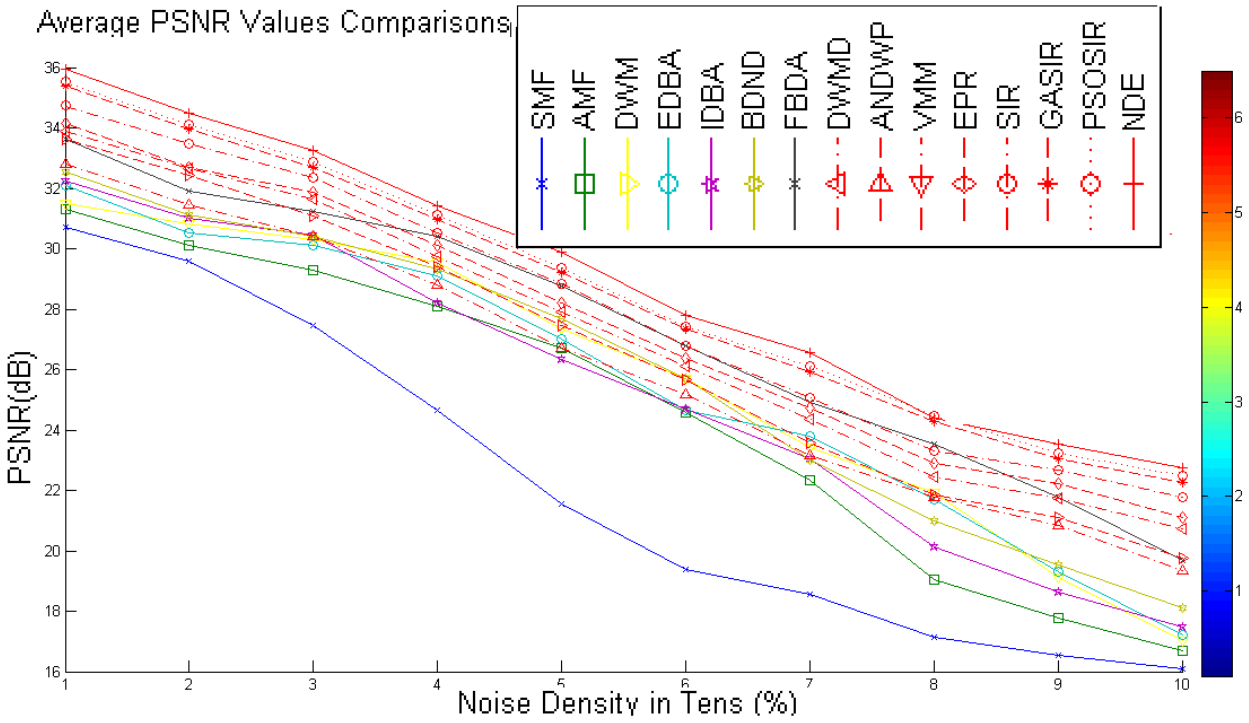


Figure 9.37: Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and NDE filter

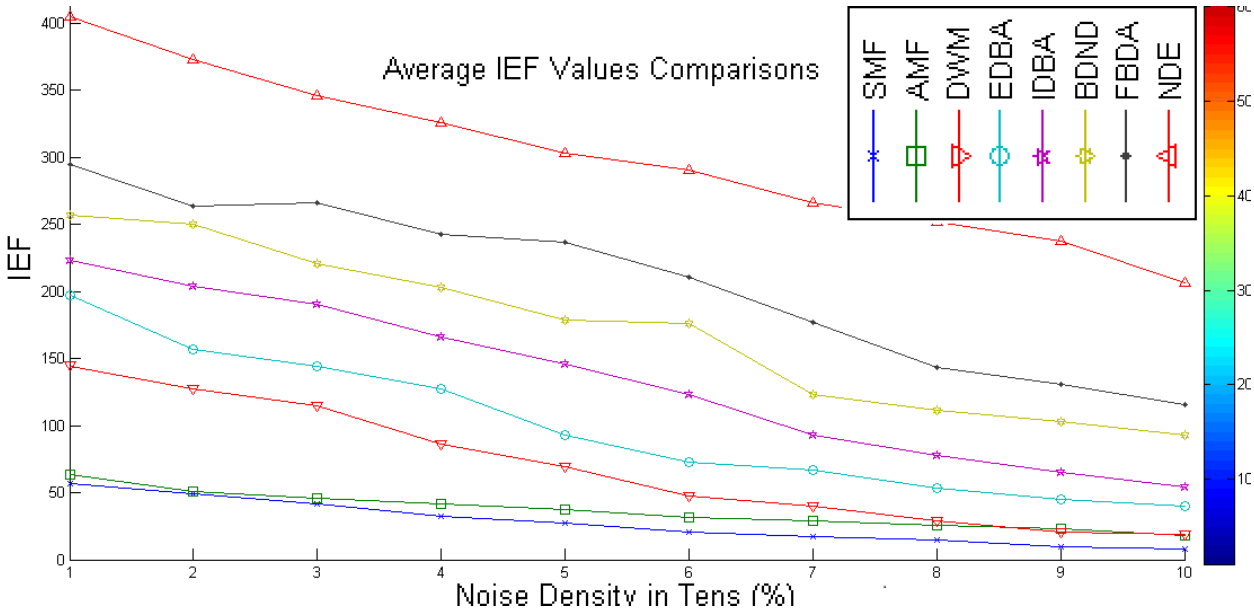


Figure 9.38: Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and NDE filter

Table 9.34: Comparison of proposed filters in *Average IEF* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images RVNs

Filter	**Noise Density/Average IEF**									
	10	20	30	40	50	60	70	80	90	100
SMF	51.77	34.56	28.98	17.65	14.44	10.91	8.01	7.88	7.12	5.01
AMF	82.87	67.99	67.12	51.81	38.73	35.13	27.39	23.91	19.91	15.19
DWM	131.99	120.82	104.83	81.73	59.71	39.70	21.39	20.11	18.97	18.89
EDBA	169.61	139.01	135.92	115.10	79.18	69.99	62.99	47.22	40.63	32.91
IDBA	201.50	187.88	165.61	139.40	116.33	100.49	81.71	62.81	59.51	51.99
BDND	231.90	220.11	201.91	169.40	142.01	109.11	90.90	86.25	82.10	76.51
FBDA	296.55	312.89	278.25	223.50	187.25	160.50	129.75	105.25	94.50	90.88
DWMD	243.49	229.09	211.68	194.12	173.91	153.04	118.89	107.34	92.37	81.18
ANDWP	292.11	265.94	256.01	233.98	220.43	190.05	160.70	135.30	111.74	101.05
VMM	301.24	274.20	260.68	249.23	235.42	220.16	191.24	159.44	124.52	111.43
EPR	324.84	293.67	273.15	264.04	250.97	229.63	213.03	189.50	156.01	133.09
SIR	346.35	315.79	287.69	276.24	263.20	251.64	230.00	211.90	180.66	151.88
GASIR	372.37	344.05	306.62	287.24	274.48	259.06	240.25	225.00	202.85	171.09
PSOSIR	384.63	350.23	314.24	296.35	285.00	266.10	245.58	230.37	211.38	177.48
NDE	390.93	354.68	325.78	306.28	289.91	276.47	254.19	242.15	217.95	189.48

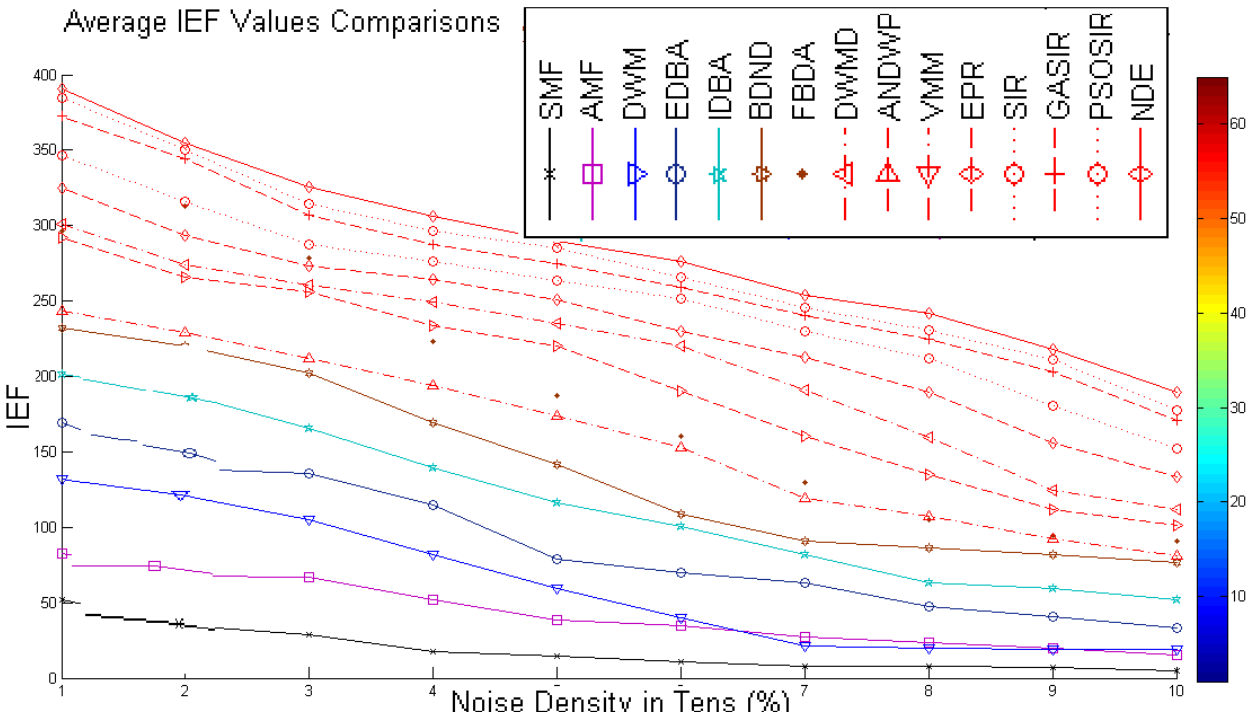


Figure 9.39: Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and the NDE filter

Table 9.35: Comparison of the proposed filters in *Average SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs

Filter	**Noise Density/Average SSIM**									
	10	20	30	40	50	60	70	80	90	100
SMF	0.9423	0.9417	0.9384	0.9328	0.9319	0.9265	0.9201	0.9156	0.9127	0.9103
AMF	0.9476	0.9425	0.9369	0.9333	0.9297	0.9265	0.9240	0.9222	0.9204	0.9178
DWM	0.9507	0.9477	0.9451	0.9445	0.9410	0.9378	0.9256	0.9210	0.9176	0.9165
EDBA	0.9565	0.9541	0.9501	0.9467	0.9420	0.9370	0.9342	0.9311	0.9267	0.9229
IDBA	0.9583	0.9567	0.9540	0.9427	0.9420	0.9413	0.9391	0.9358	0.9342	0.9311
BDND	0.9622	0.9607	0.9585	0.9571	0.9535	0.9515	0.9500	0.9457	0.9406	0.9376
FBDA	0.9687	0.9620	0.9607	0.9589	0.9554	0.9527	0.9501	0.9478	0.9430	0.9403
NDE	0.9709	0.9696	0.9685	0.9667	0.9642	0.9603	0.9574	0.9504	0.9443	0.9405

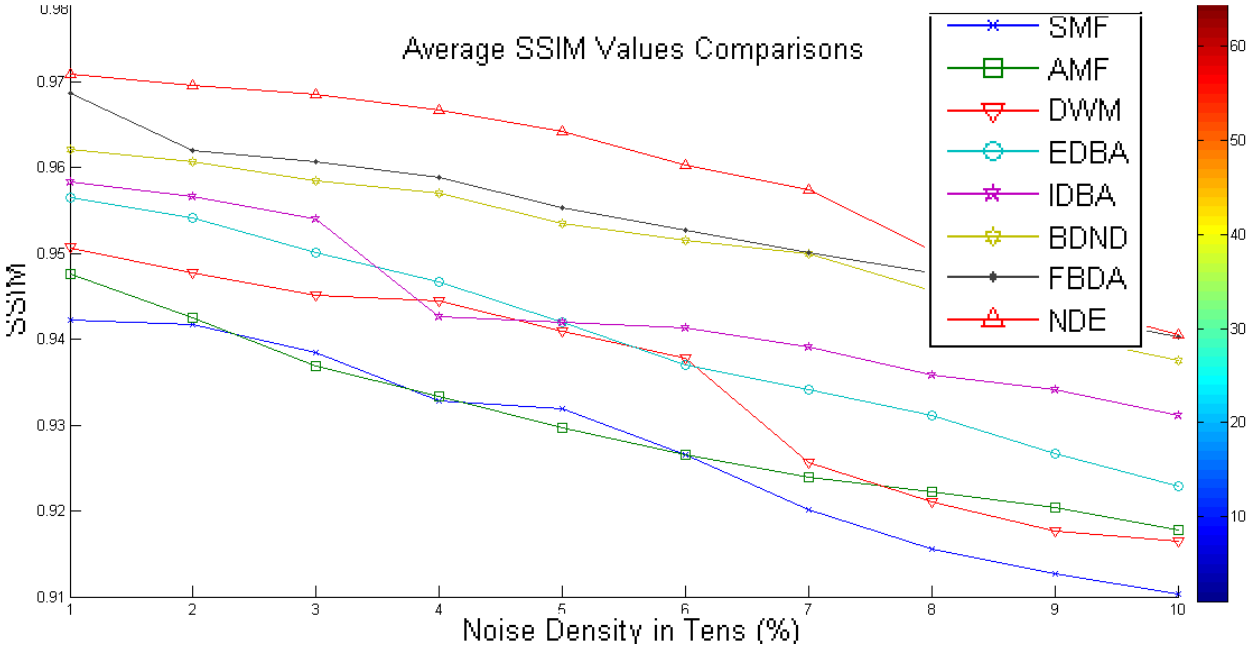


Figure 9.40: Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and NDE filter

Table 9.36: Comparison of results for the proposed algorithms in *Average SSIM* for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs

Filter	**Noise Density/Average SSIM**									
	10	20	30	40	50	60	70	80	90	100
SMF	0.9362	0.9333	0.9319	0.9287	0.9259	0.9116	0.9087	0.9076	0.9012	0.8981
AMF	0.9412	0.9377	0.9329	0.9310	0.9254	0.9150	0.9076	0.9064	0.9059	0.9045
DWM	0.9420	0.9412	0.9343	0.9312	0.9240	0.9211	0.9150	0.9219	0.9136	0.9107
EDBA	0.9449	0.9440	0.9361	0.9322	0.9315	0.9243	0.9222	0.9217	0.9156	0.9130
IDBA	0.9510	0.9508	0.9455	0.9431	0.9319	0.9311	0.9276	0.9233	0.9176	0.9167
BDND	0.9512	0.9503	0.9430	0.9411	0.9355	0.9330	0.9310	0.9270	0.9235	0.9214
FBDA	0.9598	0.9565	0.9542	0.9507	0.9429	0.9403	0.9376	0.9333	0.9317	0.9310
DWMD	0.9489	0.9460	0.9387	0.9348	0.9327	0.9296	0.9257	0.9235	0.9172	0.9150
ANDWP	0.9527	0.9491	0.9444	0.9408	0.9377	0.9324	0.9282	0.9232	0.9179	0.9123
VMM	0.9577	0.9561	0.9517	0.9479	0.9426	0.9377	0.9351	0.9338	0.9280	0.9265
EPR	0.9626	0.9623	0.9579	0.9537	0.9501	0.9456	0.9418	0.9402	0.9343	0.9314
SIR	0.9646	0.9645	0.9603	0.9562	0.9539	0.9498	0.9440	0.9424	0.9375	0.9337
GASIR	0.9703	0.9687	0.9667	0.9632	0.9603	0.9560	0.9523	0.9477	0.9398	0.9363
PSOSIR	0.9709	0.9696	0.9685	0.9667	0.9642	0.9603	0.9574	0.9504	0.9443	0.9405
NDE	0.9721	0.9718	0.9706	0.9686	0.9669	0.9642	0.9613	0.9581	0.9558	0.9542

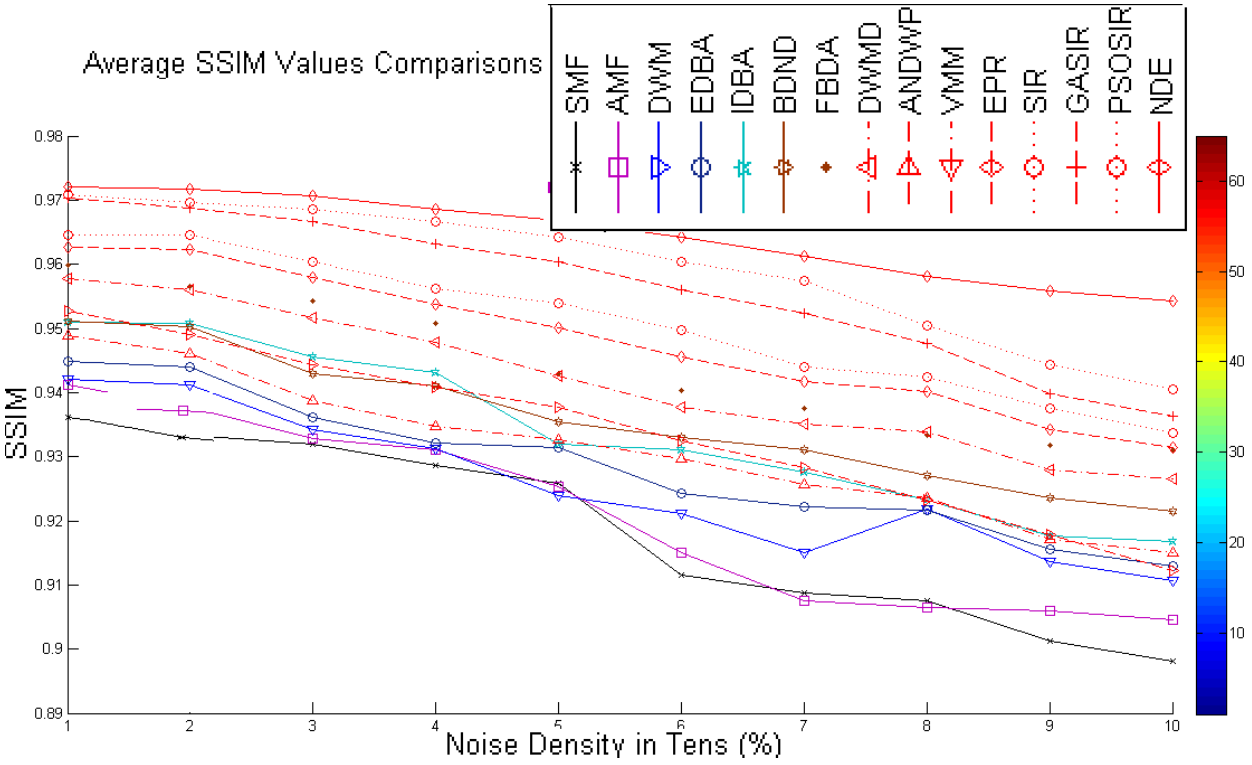


Figure 9.41: Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and NDE filter

9.4 PSO based Clustering for Noise Detection and Restoration

In section 9.3, noise density estimation based denoising filter has been validated through experimental results. Noise density is estimated to suggest an empirical window size for noise detection and filtering operations. For low noise density, a small size window and for high noise density a large size window is used. K-means clustering algorithm based noise density estimation is done in the section. A window of fixed size may not perform well for a variety of noise densities. The algorithm for denoising salt-and-pepper noise or random-valued noise in the image using the density estimation for denoising obtained excellent restoration results although the algorithm suffers from high computational cost. The execution times for all proposed algorithms along with existing algorithms are studied and presented towards the end of the chapter.

In chapter 6, fixed length and different length *PSO* based image clustering techniques for image denoising have been presented. In the section, a clustering of image pixels is done for image denoising. Fixed length particle and different length particle swarm optimization algorithms are proposed for image denoising. The different length particle swarm optimization (DPSO) algorithm performs much better than the fixed length particle swarm optimization (FPSO). The DPSO algorithm also requires less number of user parameters compared to FPSO. The results and comparisons using the DPSO algorithm are given in this section. In this section, two noise models are considered for making the noisy images. Those are *Random Valued Noise (RVN)* and *Salt and Pepper Noise (SPN)*. In section 9.4.1, some restoration results on both the noise models are generated with the standard bench mark images are given. In section 9.4.2, the comparison of the proposed **DPSO** filter with the existing algorithms in terms of PSNR (dB), IEF and SSIM are given. The comparative restoration results on the color *Lena* image is given at the end of the section.

9.4.1 Results

The **Different length Particle Swarm Optimization(DPSO)** filter (section 6.2.1) executes for five standard digital images which are corrupted by random valued noises and salt and pepper noises with 10% to 100% densities and restoration results are generated. The restoration results in terms of PSNR (dB) for those images are presented in table 9.37. From the table 9.37((1) SPN), it is observed that the PSNR values for all the five images are consistent for the entire noise density range. From low to high noise density, the average PSNR values are also improving consistently. The visual representation of the graph is given in Fig. 9.42. From the table 9.37((2) RVN), it is observed that the PSNR values for all

the five images are consistent for the entire noise density range. From low to high noise density, the average PSNR values are also improving consistently. In both the cases, the Cameraman image suffers from low PSNRs, but for other images the restoration results in terms PSNR are very satisfactory. The visual representation of the graph is given in Fig. 9.43. The restoration results in terms of IEF by the five bench mark images corrupted by random valued noise and salt and pepper noise are given in table 9.38. Seeing the values given in table 9.38((1) SPN, (2) RVN) for the entire noise density, it can be stated that except the *Cameraman* image, the other images obtained good results in terms of IEF. The average IEF values for the noise densities can be noticed to understand the improvement of the result easily. The visual representation of the graph is given in Fig. 9.44 and Fig. 9.45. For Liftingbody image, the PSNRs obtained is maximum compared to other images. Lastly the restoration results in terms of SSIM by the **DPSO** method for those images are presented in table 9.37. From the SPN and RVN parts of the table, it is observed that the SSIM values for all the five images are consistent for the entire noise density range for RVN and SPN. Except Barbara and Cameraman images, for the other images the algorithm also performs well as the SSIM values are close to one. From high to low noise density, the average SSIM values are also improving continuously. The visual representations of the graphs are given in Fig. 9.46 and Fig. 9.47. For Liftingbody and Lena images the restoration results in terms of SSIM, the proposed algorithm obtained excellent restoration results.

9.4.2 Comparisons

The proposed algorithm is compared with the existing algorithms in terms of PSNR (dB), IEF and SSIM in this section. For both the noise models, the proposed algorithm **DPSO** is validated. In table 9.40, the results by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table, all results are generated on the images corrupted by SPN. The average PSNR values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it is seen that for the entire noise density, the proposed **DPSO** performs the best in terms of PSNR. The DPSO algorithm performs much better than the NED algorithm also in terms of quantitative restoration results. The comparative results of the **DPSO** and existing seven algorithms can also be seen in the Fig. 9.48.

In table 9.41, the restoration results given by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table, the other proposed algorithms which are compared in the previous sections are also kept in the table. In this table, all results are generated on the images corrupted by RVN. The *Average PSNR* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with

Table 9.37: PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DPSO** filter for varying SPNs and RVNs

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
(1) SPN										
Lena	40.01	38.15	37.54	36.43	35.16	34.23	32.60	30.87	28.65	26.25
Cameraman	34.20	33.19	31.76	29.62	28.80	27.82	26.80	24.48	23.69	22.70
Barbara	38.30	36.11	34.91	33.30	32.65	29.52	28.32	26.55	26.65	24.15
Liftingbody	42.85	40.61	39.81	39.13	37.97	35.10	34.26	32.46	31.43	28.63
Boat	37.25	34.50	33.25	31.56	30.56	29.00	28.61	27.32	26.50	26.25
Average	38.52	36.51	35.45	34.00	33.02	31.13	30.11	28.33	27.38	25.59
(2) RVN										
Lena	39.30	37.82	36.67	34.65	34.19	32.08	29.54	26.36	24.21	23.51
Cameraman	33.54	33.25	31.80	28.87	26.85	24.65	23.25	21.54	20.25	20.05
Barbara	37.45	35.60	34.76	32.81	31.33	28.50	27.23	24.76	24.30	23.76
Liftingbody	42.56	40.33	39.41	38.50	36.32	34.15	34.00	31.59	31.00	30.90
Boat	36.43	34.70	33.00	31.29	30.50	29.10	28.70	27.21	26.80	25.60
Average	37.85	36.34	35.12	33.22	31.83	29.69	28.54	26.29	25.31	24.76

different noise density from 10% to 100%. From this table, it is seen that for the entire noise density, the proposed algorithm **DPSO** performs the best in terms of PSNR. The comparative results of the **DPSO** and existing seven algorithms can also be seen in the Fig. 9.49. The FBDA filter performs better than ANDWP filter. From the figure, it can be cleared that the **DPSO** algorithm obtained much better restoration results than the existing algorithms for the entire noise density. The algorithm also obtained better PSNR than other proposed filters such as DWMD, ANDWP, VMM, EPR, SIR, GASIR and PSOSIR. In table 9.42, the restoration results obtained by the proposed algorithm are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table, all results are generated on the images corrupted by SPN. The *Average IEF* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it is seen that for the entire noise density, the proposed algorithm **DPSO** performs the best in terms of IEF. The SMF and AMF perform very poor. The comparative results of the **NED** and existing seven algorithms can also be seen in the Fig. 9.50.

In table 9.43, the results given by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table, the other proposed algorithms

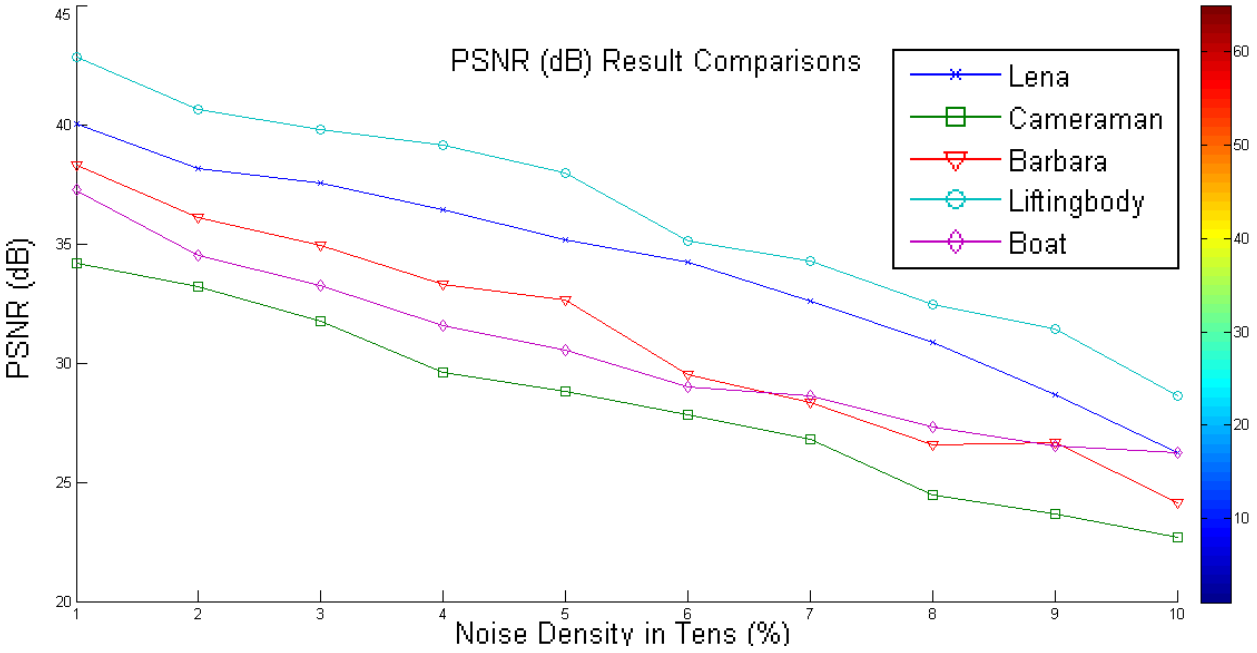


Figure 9.42: Graph for **DPSO** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs

which are compared in the previous sections are also kept in the table. In this table, all results are generated on the images corrupted by RVN. The *Average IEF* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it is clear that for the entire noise density, the proposed algorithm **DPSO** performs excellent in terms of IEF. The FBDA filter performs better than ANDWP filter. The comparative results of the **DPSO** and existing seven algorithms can also be seen in the Fig. 9.51. From the figure, it can be cleared that the **DPSO** algorithm obtained much better restoration results in terms of IEF than the existing algorithms for the entire noise density. The algorithm also obtained better IEF than other proposed filters such as DWMD, ANDWP, VMM, EPR, SIR, GASIR and PSOSIR. In table 9.44, the results generated by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table all results are generated on the images corrupted by SPN. The *Average SSIM* values for all the algorithms are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it is noticed that for the entire noise density, the proposed algorithm **DPSO** performs the best in terms of SSIM. The SMF and AMF perform very poor where as DWM, EDBA and IDBA perform similarly. The BDND and FBDA also obtained similar SSIM values. The comparative results of the **DPSO** and existing seven

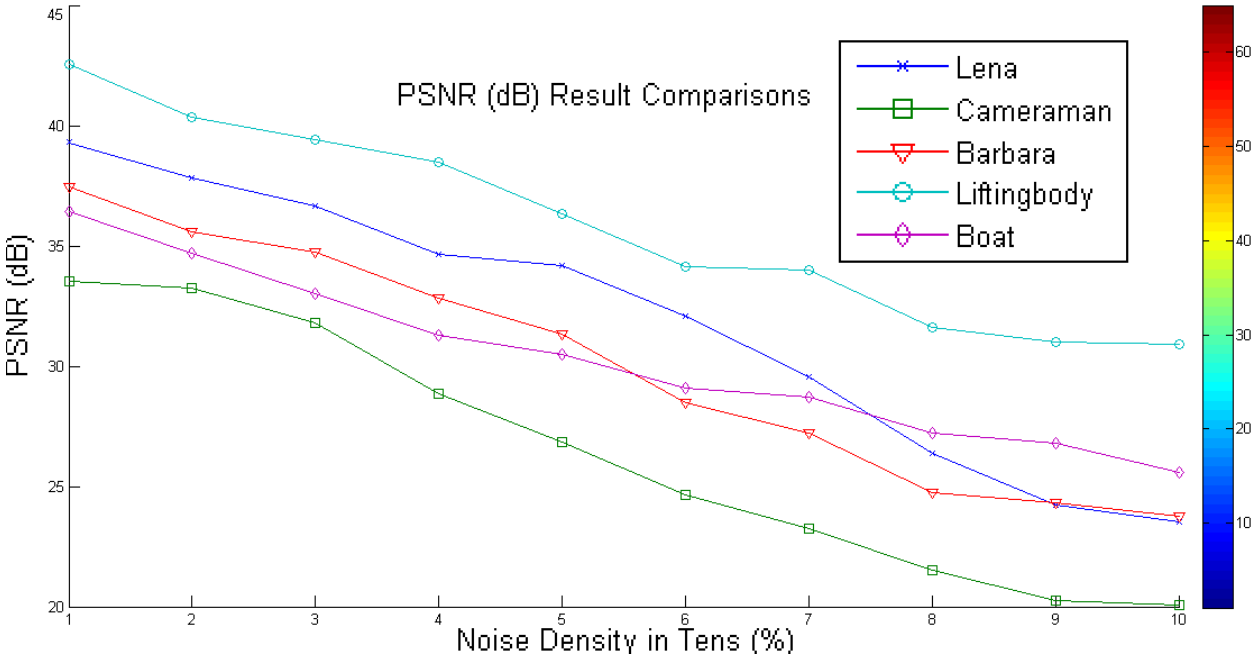


Figure 9.43: Graph for **DPSO** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

algorithms can also be seen in the Fig. 9.52.

In table 9.45, the results given by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA and FBDA algorithms. In this table, the other proposed algorithms which are compared in the previous sections are also kept in the table. In this table, all results are generated on the images corrupted by RVN. The *Average SSIM* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it is observed that for the entire noise density, the proposed **DPSO** performs excellent in terms of SSIM. The FBDA filter performs better than VMM filter. The comparative results of the **DPSO** and existing seven algorithms can also be seen in the Fig. 9.53. From the figure, it can be cleared that the **DPSO** algorithm obtained much better restoration results in terms of SSIM than the existing algorithms for the entire noise density. The algorithm also obtained better SSIM than other proposed filters such as DWMD, ANDWP, VMM, EPR, SIR, GASIR and PSOSIR.

Table 9.38: IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DPSO** filter for varying SPNs and RVNs

Image	*Noise Density/IEF*									
	10	20	30	40	50	60	70	80	90	100
(1) SPN										
Lena	471.23	443.29	411.22	378.05	344.12	332.22	342.62	311.09	313.62	239.22
Cameraman	349.50	331.74	291.60	273.53	264.72	250.25	227.25	222.18	182.00	182.30
Barbara	422.29	330.22	316.10	300.25	281.17	263.23	239.54	221.90	211.29	192.13
Liftingbody	573.50	550.60	554.47	530.45	511.00	485.50	413.25	431.44	390.17	367.29
Boat	364.72	355.17	322.20	292.81	274.21	265.60	231.31	220.25	217.10	191.17
Average	436.24	402.20	379.11	355.01	335.04	319.36	290.79	281.37	262.83	234.42
(2) RVN										
Lena	460.23	412.25	376.50	355.50	325.20	318.19	311.59	298.40	265.23	186.30
Cameraman	317.28	295.55	281.65	246.79	231.25	221.39	200.01	181.13	164.45	172.50
Barbara	378.13	323.17	289.556	271.17	251.34	245.29	224.00	201.65	187.16	176.90
Liftingbody	551.26	541.65	516.87	511.50	490.50	465.50	431.41	411.25	376.25	341.21
Boat	351.10	334.27	280.12	277.12	271.16	254.09	221.12	220.16	210.30	181.12
Average	411.60	381.37	348.93	332.41	313.89	300.89	277.62	262.51	240.67	211.60

Table 9.39: SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **DPSO** filter for varying SPNs and RVNs

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
(1) SPN										
Lena	0.9995	0.9992	0.9993	0.9982	0.9959	0.9938	0.9920	0.9878	0.9871	0.9850
Cameraman	0.9541	0.9522	0.9510	0.9487	0.9458	0.9450	0.9438	0.9420	0.9402	0.9356
Barbara	0.9571	0.9523	0.9476	0.9453	0.9462	0.9427	0.9423	0.9404	0.9369	0.9352
Liftingbody	0.9997	0.9990	0.9983	0.9966	0.9956	0.9935	0.9921	0.9904	0.9888	0.9865
Boat	0.9900	0.9887	0.9873	0.9847	0.9843	0.9807	0.9800	0.9767	0.9743	0.9723
Average	0.9801	0.9783	0.9767	0.9747	0.9736	0.9711	0.9700	0.9675	0.9655	0.9629
(2) RVN										
Lena	0.9994	0.9992	0.9988	0.9977	0.9956	0.9934	0.9921	0.9907	0.9902	0.9886
Cameraman	0.9406	0.9393	0.9386	0.9371	0.9350	0.9345	0.9334	0.9321	0.9315	0.9296
Barbara	0.9446	0.9431	0.9429	0.9427	0.9423	0.9412	0.9401	0.9391	0.9386	0.9380
Liftingbody	0.9997	0.9989	0.9980	0.9961	0.9956	0.9939	0.9937	0.9925	0.9917	0.9906
Boat	0.9856	0.9845	0.9838	0.9823	0.9817	0.9813	0.9803	0.9780	0.9770	0.9752
Average	0.9740	0.9730	0.9724	0.9712	0.9700	0.9689	0.9679	0.9665	0.9658	0.9644

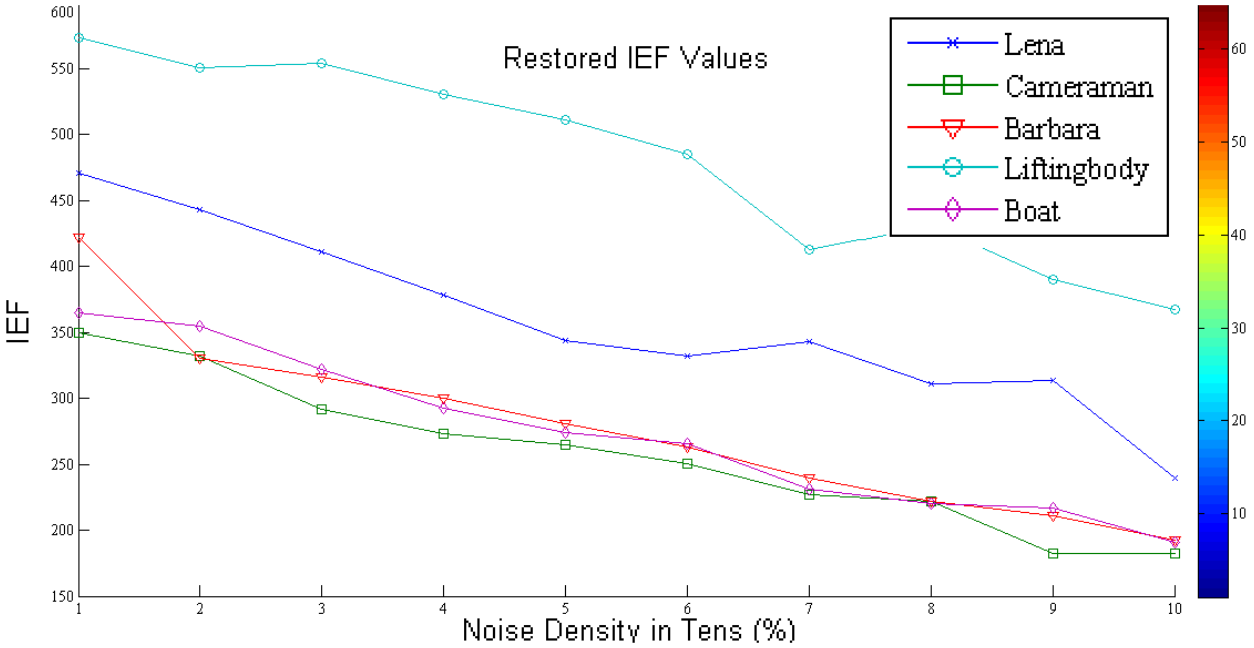


Figure 9.44: Graph for **DPSO** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs

Table 9.40: Comparison in *Average PSNR* of proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs

Filter	**Noise Density/Average PSNR(dB)**									
	10	20	30	40	50	60	70	80	90	100
SMF	32.22	30.56	28.51	26.76	24.59	22.76	20.40	18.89	17.57	16.87
AMF	32.87	30.79	28.76	28.45	26.89	24.81	22.47	19.19	17.88	16.90
DWM	33.65	31.35	30.87	29.66	27.63	25.93	24.09	22.11	19.45	17.63
EDBA	34.31	32.87	31.60	30.50	28.00	26.60	25.30	23.69	20.44	18.33
IDBA	35.29	33.70	31.79	31.10	29.50	27.11	25.54	24.25	20.32	19.06
BDND	35.41	33.69	31.50	31.32	30.79	30.00	26.32	25.21	21.25	20.62
FBDA	35.49	34.03	31.63	31.65	31.03	30.41	27.23	25.30	22.78	21.20
NED	36.68	34.74	33.62	32.10	31.15	29.26	28.16	26.43	25.39	23.69
DPSO	38.52	36.51	35.45	34.00	33.02	31.13	30.11	28.33	27.38	25.59

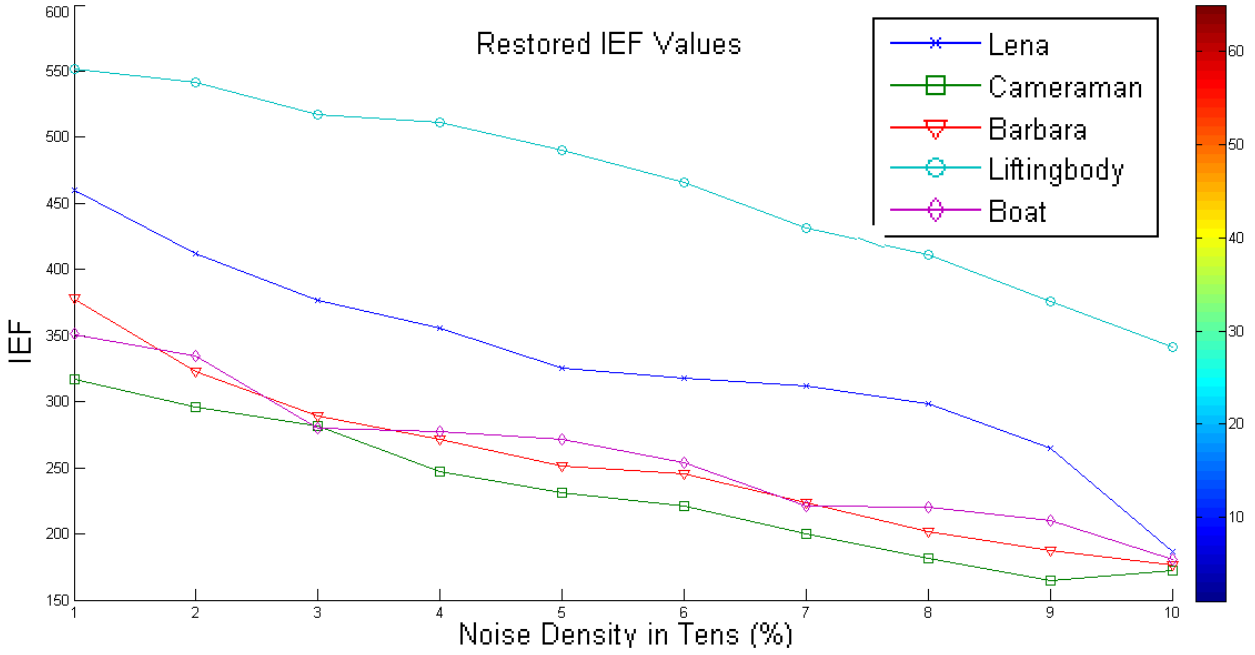


Figure 9.45: Graph for **DPSO** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

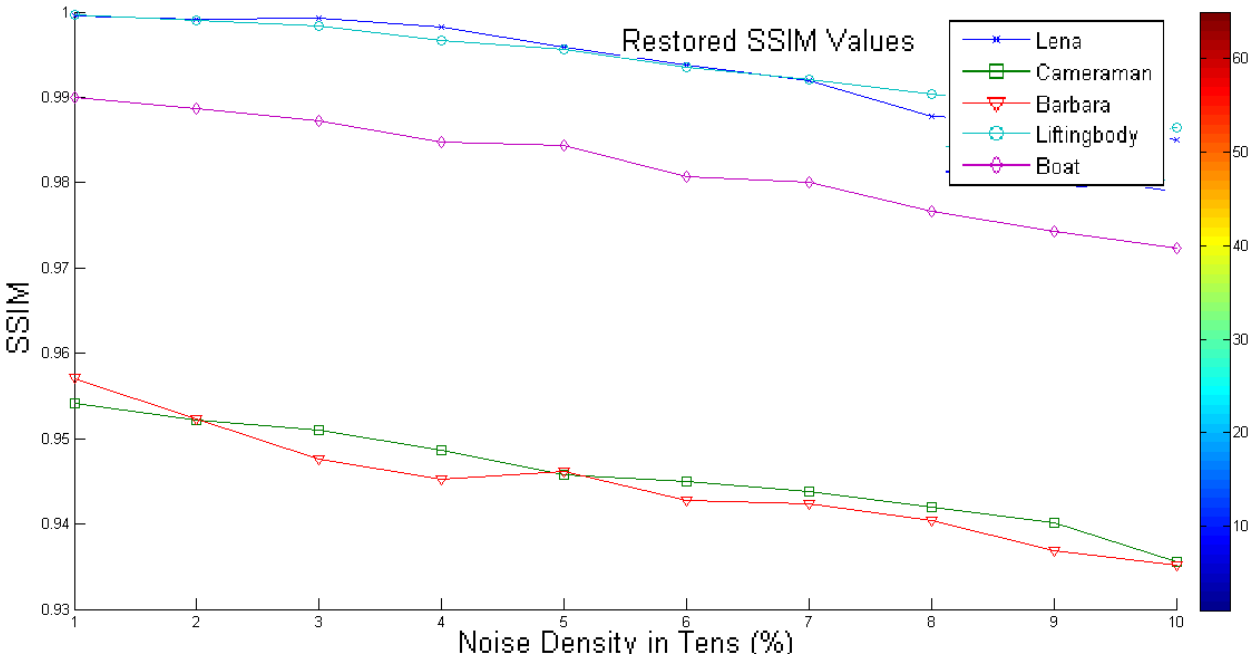


Figure 9.46: Graph for **DPSO** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs

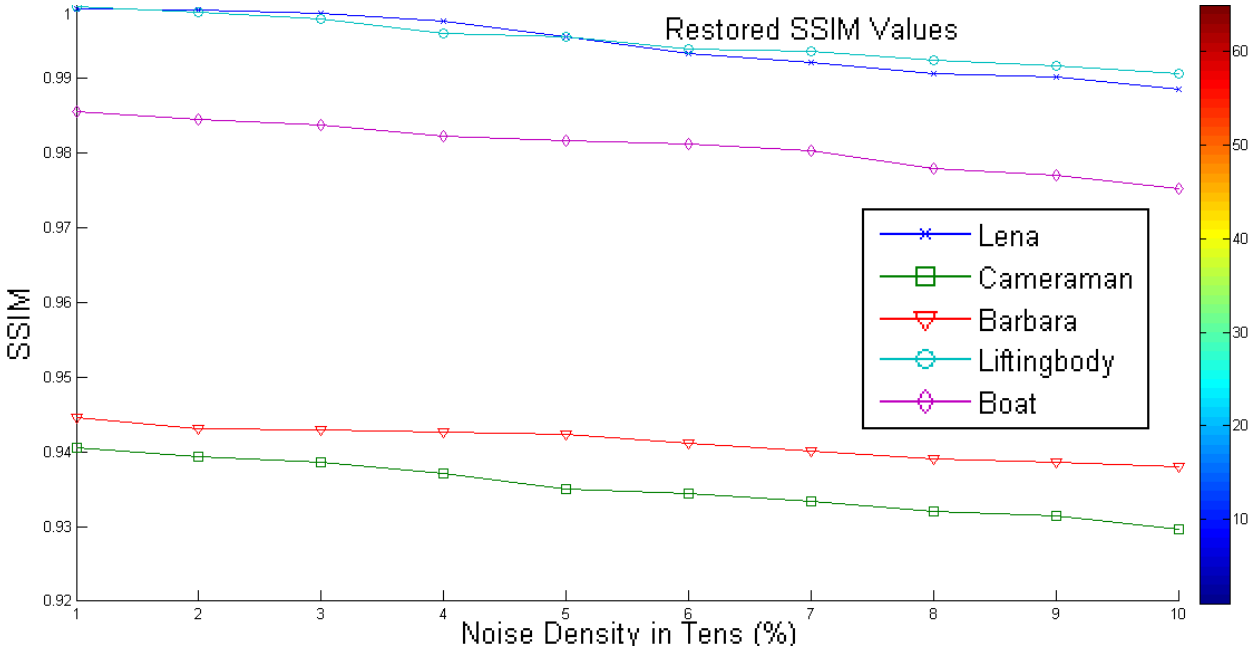


Figure 9.47: Graph for **DPSO** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

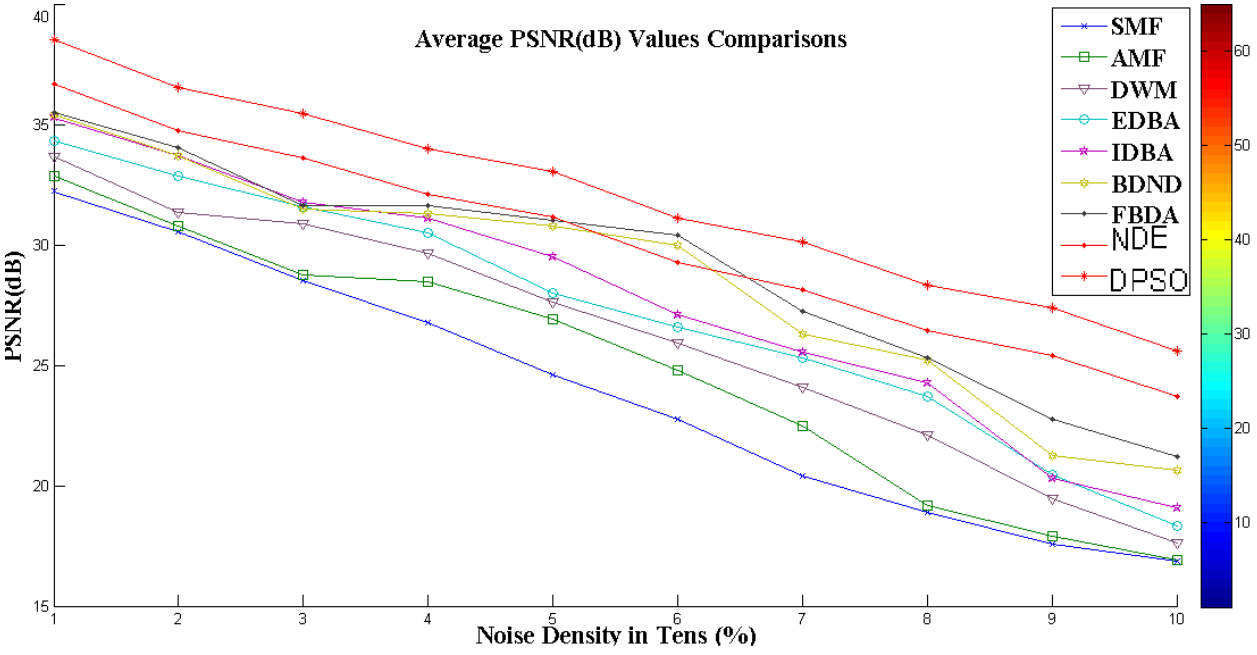


Figure 9.48: Graph of *Average PSNR* values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and DPSO filter

Table 9.41: Comparison in *Average PSNR* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs

Filter	**Noise Density/Average PSNR(dB)**									
	10	20	30	40	50	60	70	80	90	100
SMF	30.72	29.59	27.44	24.65	21.57	19.39	18.55	17.13	16.54	16.11
AMF	31.30	30.13	29.30	28.09	26.71	24.59	22.33	19.04	17.76	16.71
DWM	31.51	30.82	30.31	29.54	27.33	25.70	23.41	21.91	19.12	17.04
EDBA	32.09	30.54	30.10	29.12	27.01	24.66	23.81	21.71	19.32	17.23
IDBA	32.24	31.01	30.44	28.20	26.32	24.70	23.03	20.12	18.65	17.49
BDND	32.54	31.12	30.40	29.32	27.67	25.71	23.01	21.00	19.54	18.10
FBDA	33.62	31.90	31.22	30.40	28.76	26.79	24.92	23.55	21.79	19.70
DWMD	32.81	31.46	30.42	28.80	26.71	25.19	23.17	21.78	20.84	19.35
ANDWP	33.62	32.42	31.07	29.41	27.47	25.65	23.58	21.86	21.09	19.77
VMM	33.90	32.66	31.63	29.72	27.91	26.11	24.37	22.46	21.75	20.72
EPR	34.16	32.69	31.88	30.13	28.20	26.38	24.74	22.91	22.24	21.11
SIR	34.73	33.48	32.37	30.53	28.83	26.80	25.05	23.30	22.68	21.78
GASIR	35.39	33.98	32.68	30.98	29.23	27.33	25.93	24.27	23.06	22.25
PSOSIR	35.52	34.12	32.87	31.14	29.36	27.42	26.10	24.46	23.22	22.49
NED	35.95	34.49	33.24	31.41	29.90	27.81	26.57	24.40	23.53	22.74
DPSO	37.85	36.34	35.12	33.22	31.83	29.69	28.54	26.29	25.31	24.76

Table 9.42: Comparison of *Average IEF* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs

Filter	**Noise Density/Average IEF**									
	10	20	30	40	50	60	70	80	90	100
SMF	56.82	48.89	41.76	32.50	27.25	21.00	17.60	14.76	10.00	7.75
AMF	63.81	50.50	46.00	41.56	37.09	31.20	29.22	25.50	23.10	18.00
DWM	144.75	127.25	114.56	86.33	69.70	47.89	40.25	29.17	21.00	18.91
EDBA	197.50	156.50	144.25	127.60	92.64	72.90	67.08	53.75	44.72	40.00
IDBA	223.55	204.32	190.75	165.80	145.90	123.43	93.25	77.94	65.60	54.10
BDND	256.80	249.96	221.09	202.95	178.60	176.25	123.55	111.17	103.04	92.81
FBDA	295.10	263.72	266.50	242.55	237.01	210.50	176.69	143.50	130.56	115.95
NED	405.32	373.14	346.48	326.16	303.59	290.95	265.92	251.50	237.23	206.44
DPSO	436.24	402.20	379.11	355.01	335.04	319.36	290.79	281.37	262.83	234.42

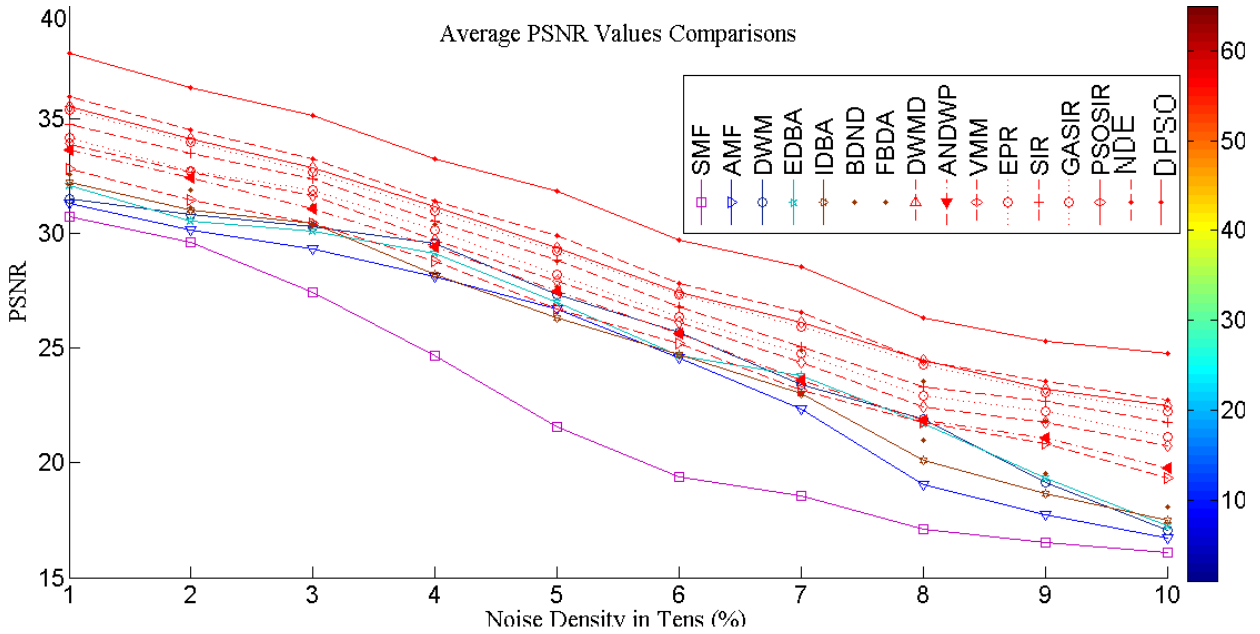


Figure 9.49: Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and DPSO filter

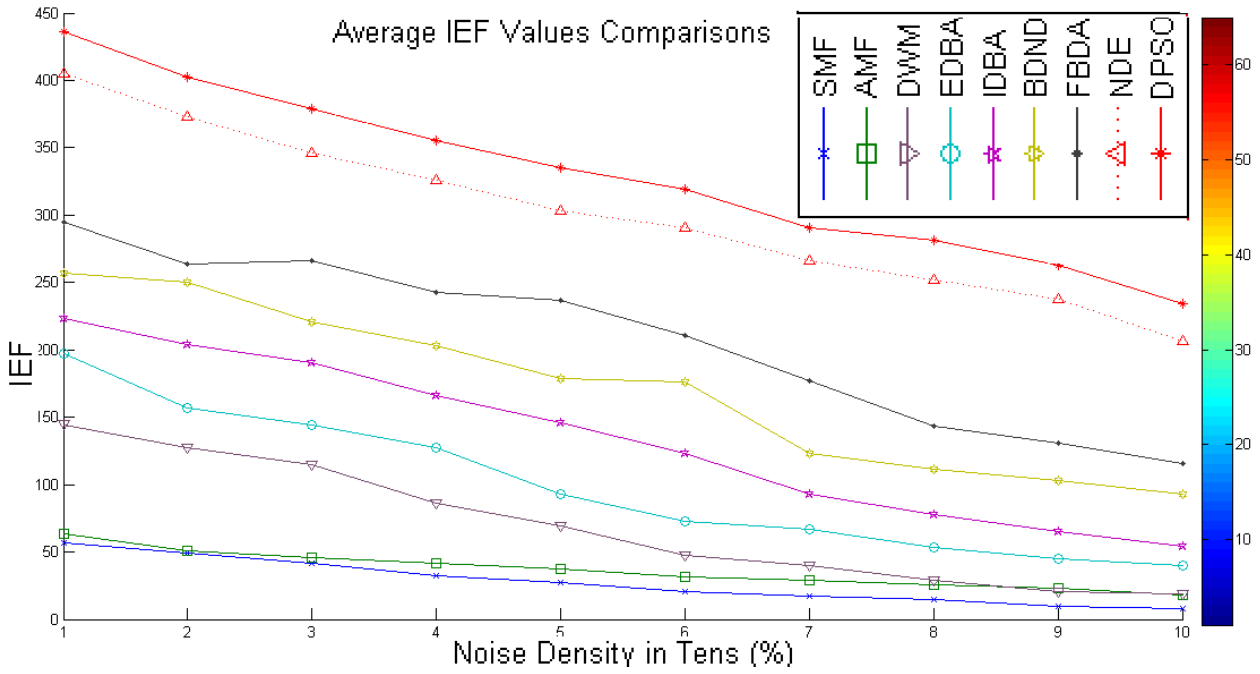


Figure 9.50: Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and DPSO filter

Table 9.43: Comparison of *Average IEF* for the proposed filter for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images RVNs

Filter	**Noise Density/Average IEF**									
	10	20	30	40	50	60	70	80	90	100
SMF	51.77	34.56	28.98	17.65	14.44	10.91	8.01	7.88	7.12	5.01
AMF	82.87	67.99	67.12	51.81	38.73	35.13	27.39	23.91	19.91	15.19
DWM	131.99	120.82	104.83	81.73	59.71	39.70	21.39	20.11	18.97	18.89
EDBA	169.61	139.01	135.92	115.10	79.18	69.99	62.99	47.22	40.63	32.91
IDBA	201.50	187.88	165.61	139.40	116.33	100.49	81.71	62.81	59.51	51.99
BDND	231.90	220.11	201.91	169.40	142.01	109.11	90.90	86.25	82.10	76.51
FBDA	296.55	312.89	278.25	223.50	187.25	160.50	129.75	105.25	94.50	90.88
DWMD	243.49	229.09	211.68	194.12	173.91	153.04	118.89	107.34	92.37	81.18
ANDWP	292.11	265.94	256.01	233.98	220.43	190.05	160.70	135.30	111.74	101.05
VMM	301.24	274.20	260.68	249.23	235.42	220.16	191.24	159.44	124.52	111.43
EPR	324.84	293.67	273.15	264.04	250.97	229.63	213.03	189.50	156.01	133.09
SIR	346.35	315.79	287.69	276.24	263.20	251.64	230.00	211.90	180.66	151.88
GASIR	372.37	344.05	306.62	287.24	274.48	259.06	240.25	225.00	202.85	171.09
PSOSIR	384.63	350.23	314.24	296.35	285.00	266.10	245.58	230.37	211.38	177.48
NED	390.93	354.68	325.78	306.28	289.91	276.47	254.19	242.15	217.95	189.48
DPSO	411.60	381.37	348.93	332.41	313.89	300.89	277.62	262.51	240.67	211.60

Table 9.44: Comparison of *Average SSIM* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs

Filter	**Noise Density/Average SSIM**									
	10	20	30	40	50	60	70	80	90	100
SMF	0.9423	0.9417	0.9384	0.9328	0.9319	0.9265	0.9201	0.9156	0.9127	0.9103
AMF	0.9476	0.9425	0.9369	0.9333	0.9297	0.9265	0.9240	0.9222	0.9204	0.9178
DWM	0.9507	0.9477	0.9451	0.9445	0.9410	0.9378	0.9256	0.9210	0.9176	0.9165
EDBA	0.9565	0.9541	0.9501	0.9467	0.9420	0.9370	0.9342	0.9311	0.9267	0.9229
IDBA	0.9583	0.9567	0.9540	0.9427	0.9420	0.9413	0.9391	0.9358	0.9342	0.9311
BDND	0.9622	0.9607	0.9585	0.9571	0.9535	0.9515	0.9500	0.9457	0.9406	0.9376
FBDA	0.9687	0.9620	0.9607	0.9589	0.9554	0.9527	0.9501	0.9478	0.9430	0.9403
NED	0.9709	0.9696	0.9685	0.9667	0.9642	0.9603	0.9574	0.9504	0.9443	0.9405
DPSO	0.9801	0.9783	0.9767	0.9747	0.9736	0.9711	0.9700	0.9675	0.9655	0.9629

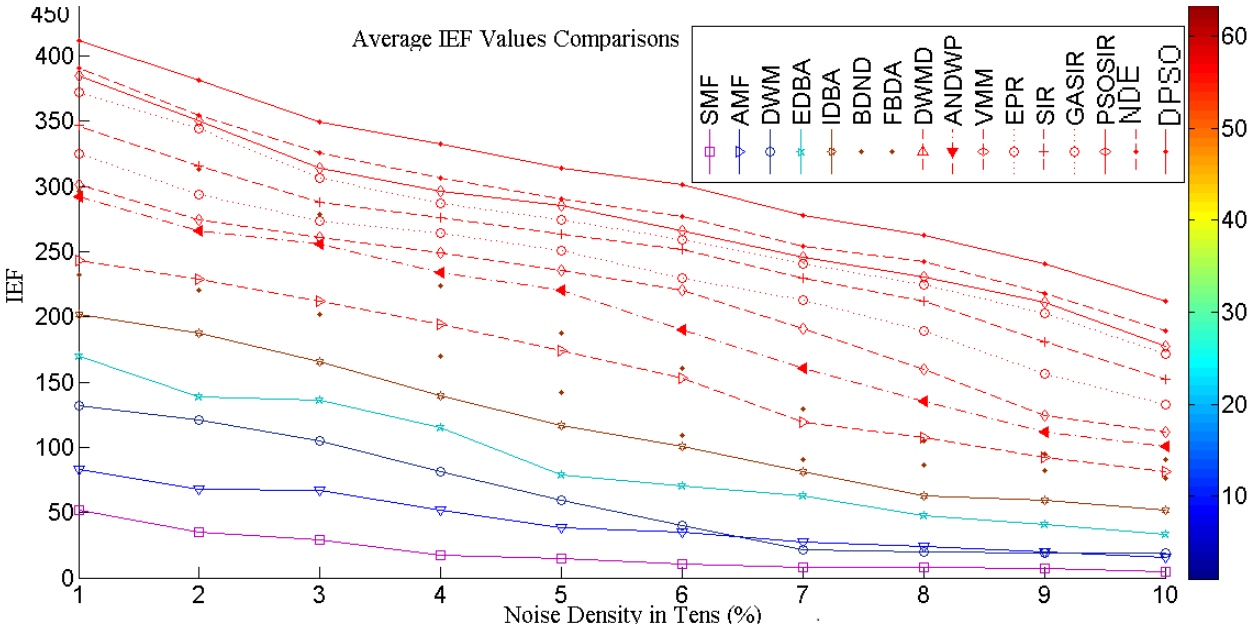


Figure 9.51: Graph for average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and DPSO filter

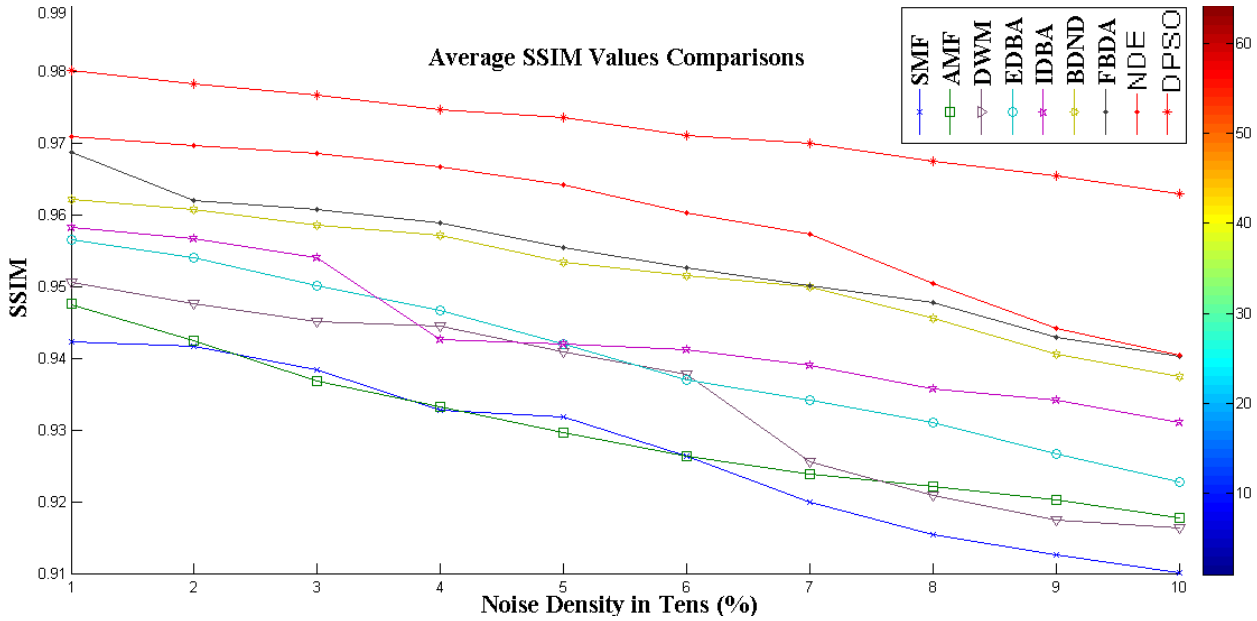


Figure 9.52: Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and DPSO filter

Table 9.45: Comparison of *Average SSIM* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs

Filter	**Noise Density/Average SSIM**									
	10	20	30	40	50	60	70	80	90	100
SMF	0.9362	0.9333	0.9319	0.9287	0.9259	0.9116	0.9087	0.9076	0.9012	0.8981
AMF	0.9412	0.9377	0.9329	0.9310	0.9254	0.9150	0.9076	0.9064	0.9059	0.9045
DWM	0.9420	0.9412	0.9343	0.9312	0.9240	0.9211	0.9150	0.9219	0.9136	0.9107
EDBA	0.9449	0.9440	0.9361	0.9322	0.9315	0.9243	0.9222	0.9217	0.9156	0.9130
IDBA	0.9510	0.9508	0.9455	0.9431	0.9319	0.9311	0.9276	0.9233	0.9176	0.9167
BDND	0.9512	0.9503	0.9430	0.9411	0.9355	0.9330	0.9310	0.9270	0.9235	0.9214
FBDA	0.9598	0.9565	0.9542	0.9507	0.9429	0.9403	0.9376	0.9333	0.9317	0.9310
DWMD	0.9489	0.9460	0.9387	0.9348	0.9327	0.9296	0.9257	0.9235	0.9172	0.9150
ANDWP	0.9527	0.9491	0.9444	0.9408	0.9377	0.9324	0.9282	0.9232	0.9179	0.9123
VMM	0.9577	0.9561	0.9517	0.9479	0.9426	0.9377	0.9351	0.9338	0.9280	0.9265
EPR	0.9626	0.9623	0.9579	0.9537	0.9501	0.9456	0.9418	0.9402	0.9343	0.9314
SIR	0.9646	0.9645	0.9603	0.9562	0.9539	0.9498	0.9440	0.9424	0.9375	0.9337
GASIR	0.9703	0.9687	0.9667	0.9632	0.9603	0.9560	0.9523	0.9477	0.9398	0.9363
PSOSIR	0.9709	0.9696	0.9685	0.9667	0.9642	0.9603	0.9574	0.9504	0.9443	0.9405
NED	0.9721	0.9718	0.9706	0.9686	0.9669	0.9642	0.9613	0.9581	0.9558	0.9542
DPSO	0.9740	0.9730	0.9724	0.9712	0.9700	0.9689	0.9679	0.9665	0.9658	0.9644

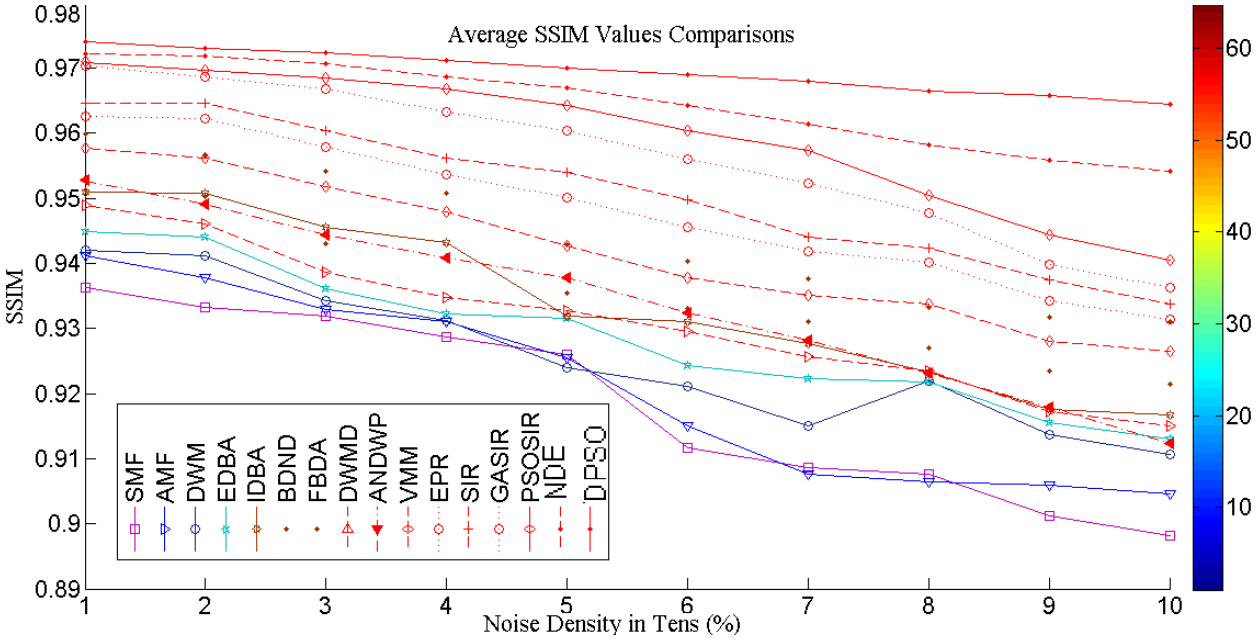


Figure 9.53: Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and DPSO filter

9.5 Fuzzy Switching Median Filter

In this section, a **fuzzy switching median filter (FSMF)** based restoration results are presented and compared with proposed and some recent existing algorithms. Two noise models are considered for making the noisy images. Those are *Random Valued Noise (RVN)* and *Salt and Pepper Noise (SPN)*. Results obtained from all the proposed filters on the spatial domain are compared and validated by extensive simulation results, in this section. The required execution time for all the proposed algorithms are studied and compared with the state of the art algorithms in the literature. Variable noise density from 10% to 100% have been applied to five bench mark images to evaluate the performance.

In section 9.5.1, restoration results on both the noise models are given with the standard bench mark images. In section 9.5.2, the comparison of the proposed filter with the existing algorithms in terms of PSNR (dB), IEF and SSIM are given. The comparative restoration results on the color *Lena* image is given at the end of this section. Discussion of experimental results and comparisons is given in section 9.7.

9.5.1 Results

The **fuzzy switching median filter (FSMF)** (chapter 7) executes for five standard digital images which are corrupted by random valued noises and salt-and-pepper noises with 10% to 100% densities and restoration results are generated. The restoration results in terms of PSNR (dB) for those images are presented in table 9.46. From the table 9.46((1) SPN), it is observed that the PSNR values for all the five images are consistent for the entire noise density range. The average PSNR values are improving consistently in accordance with the high to low noise density. The visual representation is given in Fig. 9.54. From the table 9.46((2) RVN), it is observed that the PSNR (dB) values for all the five images are consistent for the entire noise density range. From high to low noise density, the average PSNR values are also improving consistently. In both the cases, the Cameraman image suffers from low PSNRs, but for other images the restoration results in terms PSNR are very satisfactory. The visual representation is given in Fig. 9.55.

The restoration results in terms of IEF by the five bench mark images corrupted by random valued noise and salt and pepper noise are given in table 9.47. From the values given in table 9.47((1) SPN, (2) RVN), for the entire noise density, it is seen that except the *Cameraman* image, the other images obtained good results in terms of IEF. The average IEF values for the noise densities can be noticed to convince the improvement of the results easily. The visual representation of the graph is given in Fig. 9.56 and Fig. 9.57. For Liftingbody image, the PSNRs obtained are maximum compared to other images.

Lastly the restoration results in terms of SSIM by the **FSMF** method for those images

are presented in table 9.46. From the table, it is observed that the SSIM values for all the five images are consistent for the entire noise density range for RVN and SPN. Except Barbara and Cameraman images, for the other images the algorithm also performs well as the SSIM values are close to one. From high to low noise density, the average SSIM values are improving clearly. The visual representations of the graphs are given in Fig. 9.58 and Fig. 9.59. For Liftingbody and Lena images, the proposed algorithm has obtained excellent restoration results in terms of SSIM.

Table 9.46: PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **FSMF** filter for varying SPNs and RVNs

Image	*Noise Density/PSNR*									
	10	20	30	40	50	60	70	80	90	100
(1) SPN										
Lena	39.13	36.71	35.33	34.25	33.20	31.86	30.62	30.17	28.13	27.24
Cameraman	32.57	31.62	30.54	28.57	27.43	26.09	24.78	23.65	22.23	21.36
Barbara	36.75	35.21	33.25	32.10	31.43	28.05	26.75	25.55	24.67	23.50
Liftingbody	41.31	39.61	38.60	37.61	36.20	34.32	32.71	31.35	30.81	29.15
Boat	36.10	33.87	32.05	30.78	29.55	28.10	27.35	26.24	24.87	24.30
Average	37.17	35.40	33.95	32.66	31.56	29.68	28.44	27.39	26.14	25.11
(2) RVN										
Lena	38.15	36.90	35.13	33.78	32.23	30.80	28.25	25.82	23.57	22.15
Cameraman	32.41	31.62	30.20	28.32	26.05	23.23	21.43	20.29	19.43	18.32
Barbara	35.86	34.65	33.27	31.93	29.87	27.21	26.10	24.00	22.76	22.01
Liftingbody	41.04	39.79	38.20	36.82	35.21	33.46	32.64	30.98	29.59	28.23
Boat	35.33	33.52	32.16	30.54	29.67	28.32	27.10	26.32	25.43	24.16
Average	36.55	35.29	33.79	32.27	30.60	28.60	27.10	25.48	24.15	22.97

In tables 9.49 and 9.50, the bit error rates are shown. Two types of noise densities are considered on the Lena and Cameraman images which are corrupted with salt-and-pepper noise and random valued noise. The restoration results of the proposed filters are given in those tables. In those tables the number is a scalar that shows the number of bits that differ and the ratio is the value which is obtained by dividing by the total number of bits. From the tables it is seen that the ratios in both the noise models and noise densities are very less, which show the efficiency of the proposed algorithms.

In table 9.51, the restoration results in terms of PSNR are shown. The noise type considered is the quantization noise . Ten algorithms are considered for validation using the noise type on Lena and Cameraman images. From the table it is seen that the proposed

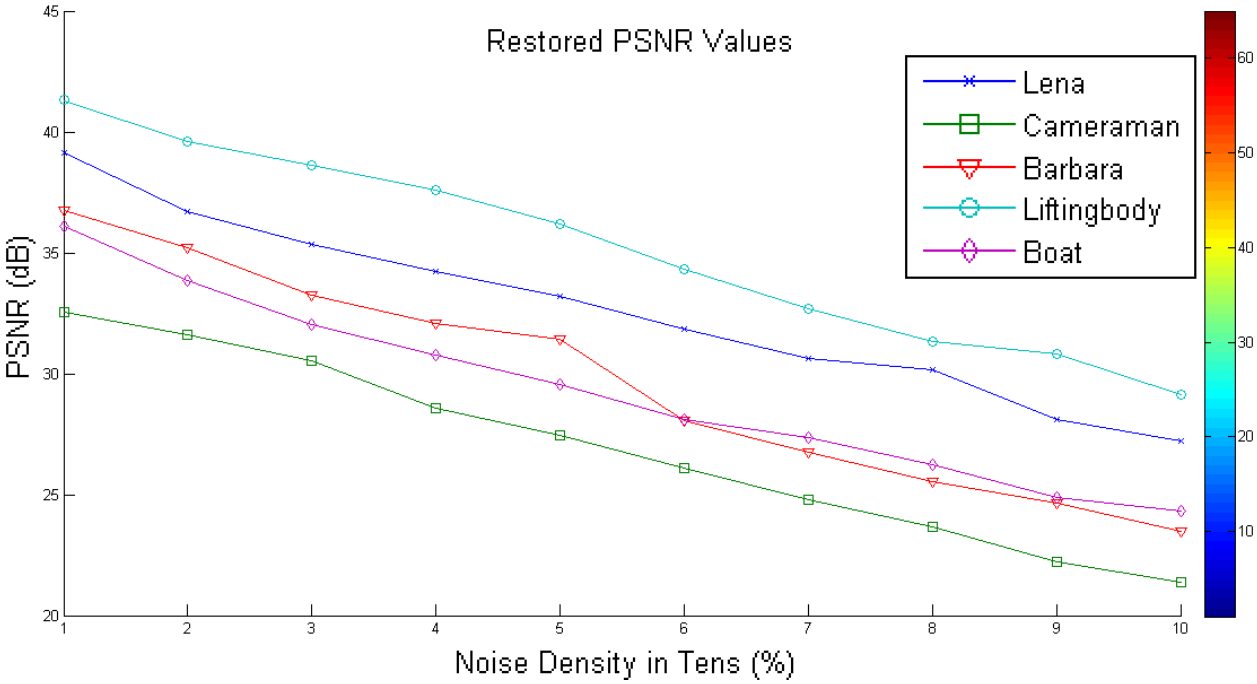


Figure 9.54: Graph for **FSMF** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs

filters starting from DWMD to DPSO, obtain an excellent restoration results.

In table 9.52, quantitative restoration results are generated in terms of PSNR (dB) for the anisotropic noise . The proposed filters obtain the restoration results given in the table. Randomly 50% rows and 50% columns are selected to make the Lena and Cameraman images corrupted. At the same time, double processing has also been done on the pixels during noise mixing. One fourth of the row positions and one fourth of the column positions are randomly selected to mix the noise. Lastly 50% row positions and 50% column positions are corrupted at the same time. The noise density is kept fixed on 50% on all the considerations of noise mixing. In the table, the first column shows the PSNR for the Lena and Cameraman images when only 50% rows are noisy. Similarly the second column shows the PSNR values after only 50% columns are corrupted. Third row gives the results after the 25% row positions and 25% column positions are mixed with anisotropic noises. Similar effect has also been done in the last column. From the results obtained it can be concluded that the filters perform excellent on the anisotropic noises on both the images regardless of noise mixing positions.

In table 9.53, the restoration results in terms of UIQI are shown. Both the noise models are considered to validate the results of the proposed filters on Lena and Cameraman images. The noise density is considered 50%. From the table it is obvious that the proposed filters

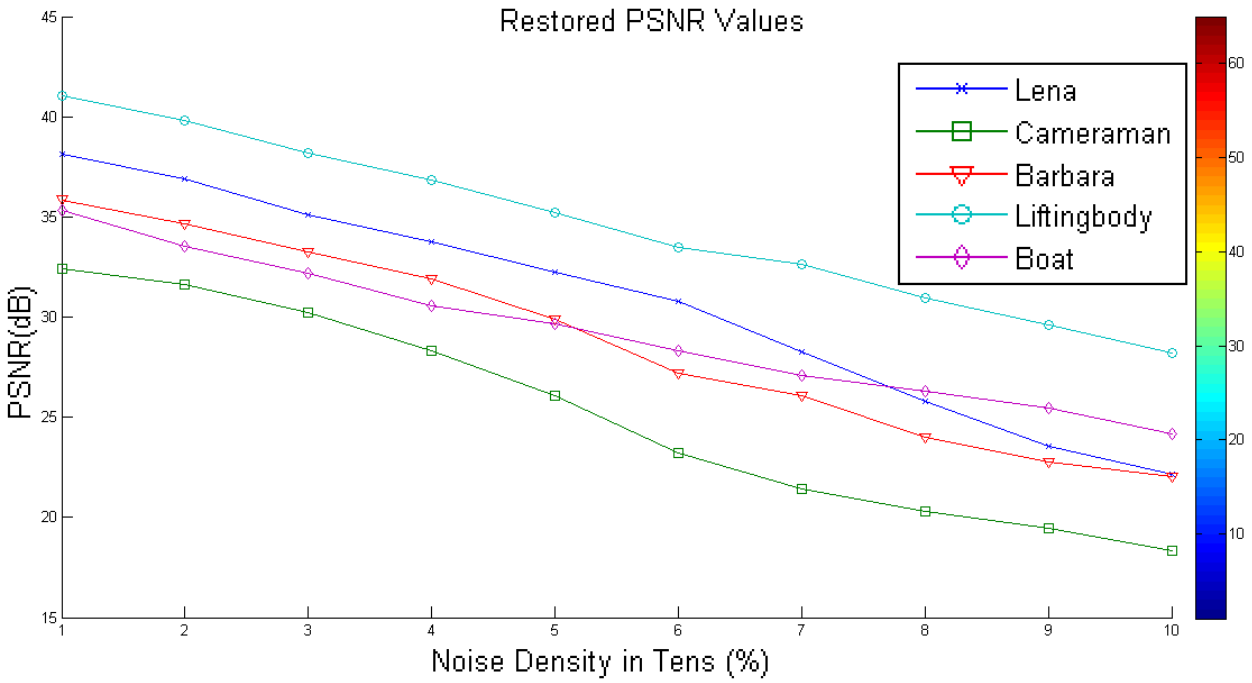


Figure 9.55: Graph for **FSMF** in PSNR for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

obtain excellent restoration results in terms of UIQI on both the noise models.

In figures 9.60, 9.61, 9.62 and 9.63 the regression analysis graphs are shown. Randomly a 3×3 window is selected as sub-image of Lena which is corrupted with salt and pepper noise and random valued noise with 30% noise density. All nine pixels of the 3×3 window are restored by the DWMD and DPSO filters. The pixels are plotted in the graph and then using the regression method one straight line is drawn. From the straight line in figures 9.60 and 9.61, the inference is that the line belongs to the restored pixels (Y-axis) more rather than to the original pixels (X-axis). But in figures 9.62 and 9.63, the regression line belongs to the original pixels (X-axis) more compared to the restored pixels (Y-axis). From the results it can be concluded that both of the proposed algorithms perform well but the DPSO preserves more originality in the restored Lena images.

In figures 9.64, 9.65 and 9.66, the color image components in terms of luminance/luma (Y) , and two chrominance/chroma (Cb and Cr) are shown. Color Lena image is corrupted with 50% salt-and-pepper noise and random valued noise in those figures. The results using the filters DWMD, FSMF and DPSO are shown. The color components of the noisy Lena image corrupted with SPN are shown in figure 9.64 d-f. The color components of the noisy Lena image corrupted with RVN are shown in figures 9.65 d-f. The original image color

Table 9.47: IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **FSMF** filter for varying SPNs and RVNs

Image	*Noise Density/IEF*									
	10	20	30	40	50	60	70	80	90	100
(1) SPN										
Lena	465.29	421.90	396.56	339.00	323.09	302.78	276.65	267.27	256.27	213.18
Cameraman	333.52	317.43	269.60	242.12	230.77	221.27	218.57	170.23	164.20	150.45
Barbara	407.25	319.50	303.23	287.00	261.59	256.27	212.73	191.82	177.16	161.50
Liftingbody	529.50	502.56	479.50	460.52	447.81	433.34	387.22	363.55	329.40	310.50
Boat	329.54	323.47	302.43	270.23	247.81	233.17	210.85	200.50	189.90	167.40
Average	413.02	376.97	350.26	319.77	302.21	289.36	261.20	238.67	223.38	200.60
(2) RVN										
Lena	431.17	391.50	340.90	321.80	307.75	288.71	258.91	240.35	215.55	171.80
Cameraman	301.22	278.17	251.72	222.69	201.18	172.83	167.23	156.50	137.50	120.74
Barbara	374.50	341.24	267.50	256.25	234.25	228.17	207.80	187.40	162.65	160.29
Liftingbody	534.54	509.50	479.44	449.78	431.25	417.76	379.27	350.80	336.76	303.75
Boat	317.70	281.50	251.67	220.90	202.78	189.82	176.87	163.56	134.63	112.00
Average	391.82	360.38	318.24	294.28	275.44	259.45	238.01	219.72	197.41	173.71

Table 9.48: SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images in **FSMF** filter for varying SPNs and RVNs

Image	*Noise Density/SSIM*									
	10	20	30	40	50	60	70	80	90	100
(1) SPN										
Lena	0.9991	0.9984	0.9978	0.9964	0.9932	0.9907	0.9900	0.9867	0.9835	0.9807
Cameraman	0.9512	0.9500	0.9471	0.9432	0.9407	0.9387	0.9368	0.9315	0.9278	0.9231
Barbara	0.9522	0.9502	0.9450	0.9421	0.9407	0.9378	0.9353	0.9316	0.9301	0.9280
Liftingbody	0.9983	0.9944	0.9912	0.9897	0.9883	0.9846	0.9825	0.9804	0.9763	0.9754
Boat	0.9857	0.9832	0.9830	0.9803	0.9780	0.9731	0.9716	0.9688	0.9632	0.9620
Average	0.9773	0.9752	0.9728	0.9703	0.9682	0.9650	0.9632	0.9598	0.9562	0.9538
(2) RVN										
Lena	0.9935	0.9930	0.9917	0.9880	0.9854	0.9830	0.9807	0.9788	0.9765	0.9723
Cameraman	0.9392	0.9368	0.9350	0.9313	0.9301	0.9289	0.9222	0.9200	0.9163	0.9150
Barbara	0.9422	0.9404	0.9388	0.9366	0.9332	0.9306	0.9277	0.9254	0.9227	0.9206
Liftingbody	0.9968	0.9950	0.9937	0.9923	0.9901	0.9864	0.9833	0.9800	0.9775	0.9744
Boat	0.9830	0.9817	0.9777	0.9751	0.9734	0.9720	0.9701	0.9670	0.9657	0.9650
Average	0.9709	0.9694	0.9674	0.9647	0.9624	0.9602	0.9568	0.9540	0.9517	0.9495

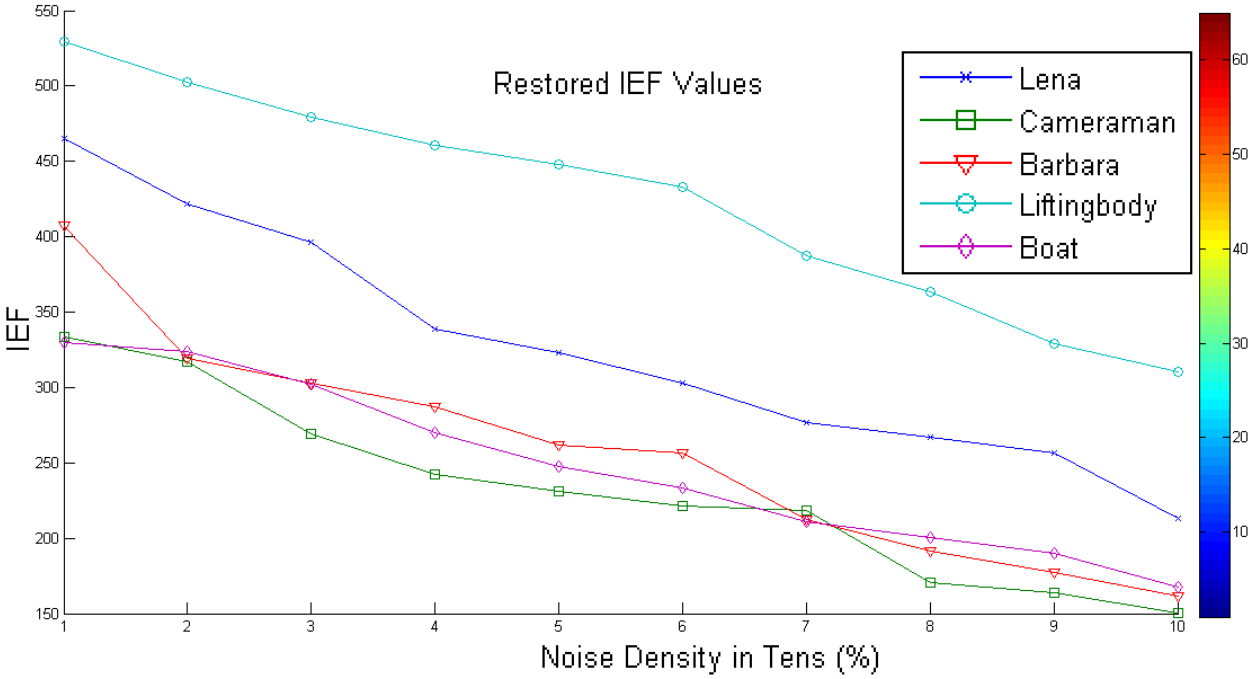


Figure 9.56: Graph for **FSMF** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs

components are shown in figure 9.64 a-c. By focusing on the color spaces in figures 9.64 and 9.65, it can be concluded that the DPSO filter can restore the noisy image well and obtains an excellent qualitative restoration results. The noisy pixels are almost eliminated from the image compared to the noisy image. The restored image color components are shown in figure 9.64 g-i for SPN, and in figure 9.65 g-i for RVN.

Similarly in figure 9.66, the color components are shown with the same experimental setup when the noisy images are restored by DWMD and FSMF filters. In case of DWMD filter, the restored images have some noisy pixels however the FSMF performs excellent in terms of noise elimination. From all these visual results, it is seen that the proposed algorithms can perform well in terms of noise separation by means of Y, Cb and Cr components.

9.5.2 Comparisons

The proposed algorithm has been compared with some recent existing algorithms in terms of PSNR (dB), IEF and SSIM. For both the noise models, the proposed algorithm **FSMF** has been validated. In table 9.54, the proposed algorithm is compared to SMF, AMF, DWM, EDBA, IDBA, FBDA and NASRI algorithms. In this table, all results are generated on

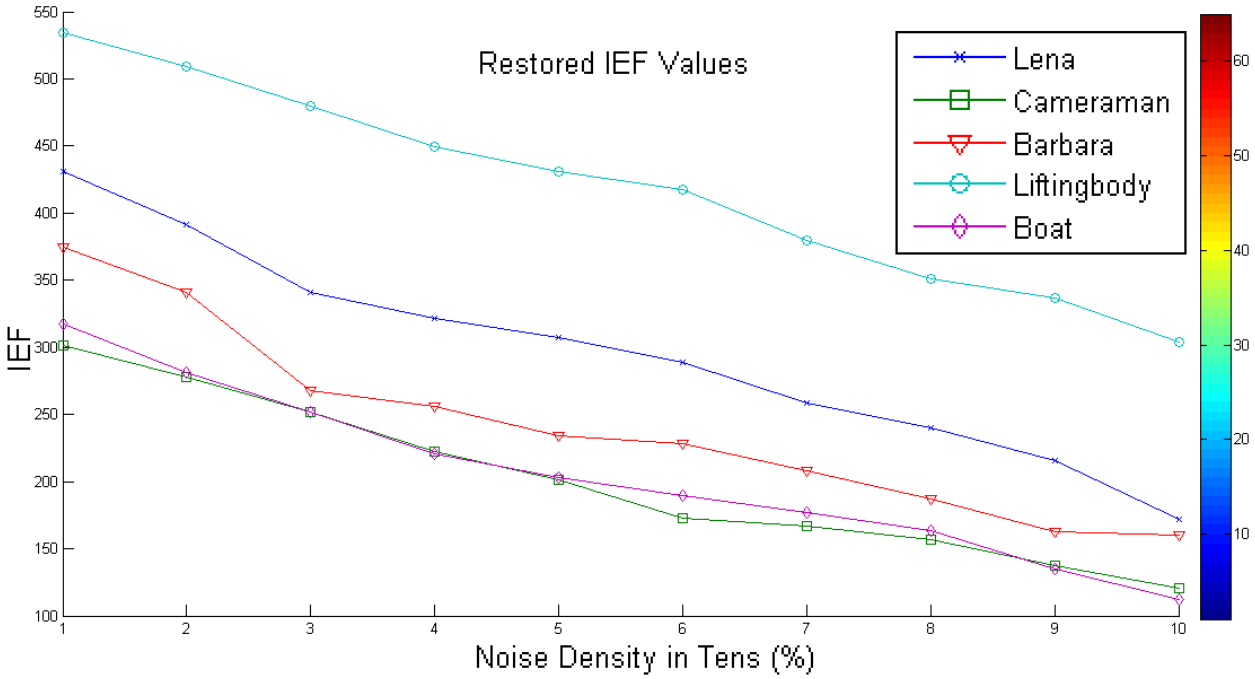


Figure 9.57: Graph for **FSMF** in IEF for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

the images corrupted by SPN. The *Average PSNR* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it can be seen that for the entire spectrum of noise density, the proposed algorithm **DPSO** performs the best in terms of PSNR. The DPSO algorithm performs much better than the NDE algorithm in terms of quantitative restoration results. The FSMF performs better than NASRI for the noise densities such as 10%, 30%, 60%, 80%, 90% and 100%. For other noise densities, the FSMF is defeated by the NASRI filter. But DPSO filter performs the best than any algorithm in the table in terms of PSNR. The NDE obtained much better PSNR than the FBDA filter but not NASRI. The FSMF filter has given slightly less PSNR than the DPSO filter. The comparative results of the **FSMF** along with the other proposed filters and existing eight algorithms can also be seen in the Fig. 9.67.

In table 9.55, the results obtained by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA, FBDA and NASRI algorithms. In the table, all results are generated on the images corrupted by RVN. The average PSNR values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it can be seen that for the entire noise density, the **DPSO**

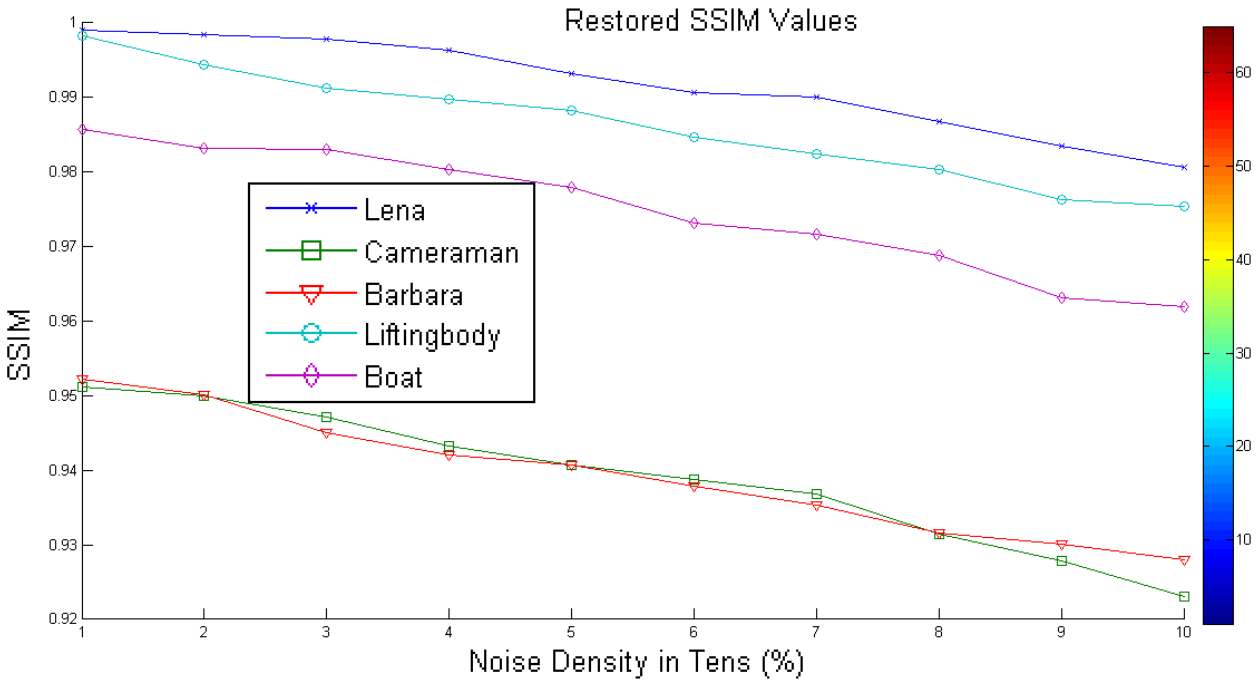


Figure 9.58: Graph for **FSMF** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% on SPNs

algorithm performs the best in terms of PSNR. The DPSO and FSMF filters obtained better PSNRs than any existing filters in the table. The NDE filter obtained better PSNRs than NASRI for the noise densities, 10%, 20%, 30%, 70%, 90% and 100%. The comparative results of the **FSMF** and existing eight algorithms can also be seen in the Fig. 9.68. The FBDA filter performs better than ANDWP filter. From the figure, it is clear that the **DPSO** algorithm has obtained much better restoration results than the existing algorithms for the entire noise density. The algorithm also obtained better PSNR than other proposed filters such as DWMD, ANDWP, VMM, EPR, SIR, GASIR, PSOSIR, NDE and DPSO.

In table 9.56, the restoration results by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA, FBDA and NASRI algorithms. In this table, all results are generated on the images corrupted by SPN. The average IEF values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it is seen that for the entire noise density, the **DPSO** algorithm performs the best in terms of IEF. The SMF and AMF perform very poor. The NASRI obtained better IEF than the NDE for only 10% noise density. For all other cases, the NDE performs better than NASRI. The FSMF filter performs much better restoration results in IEF than any existing algorithms. The DPSO performs much better than FSMF

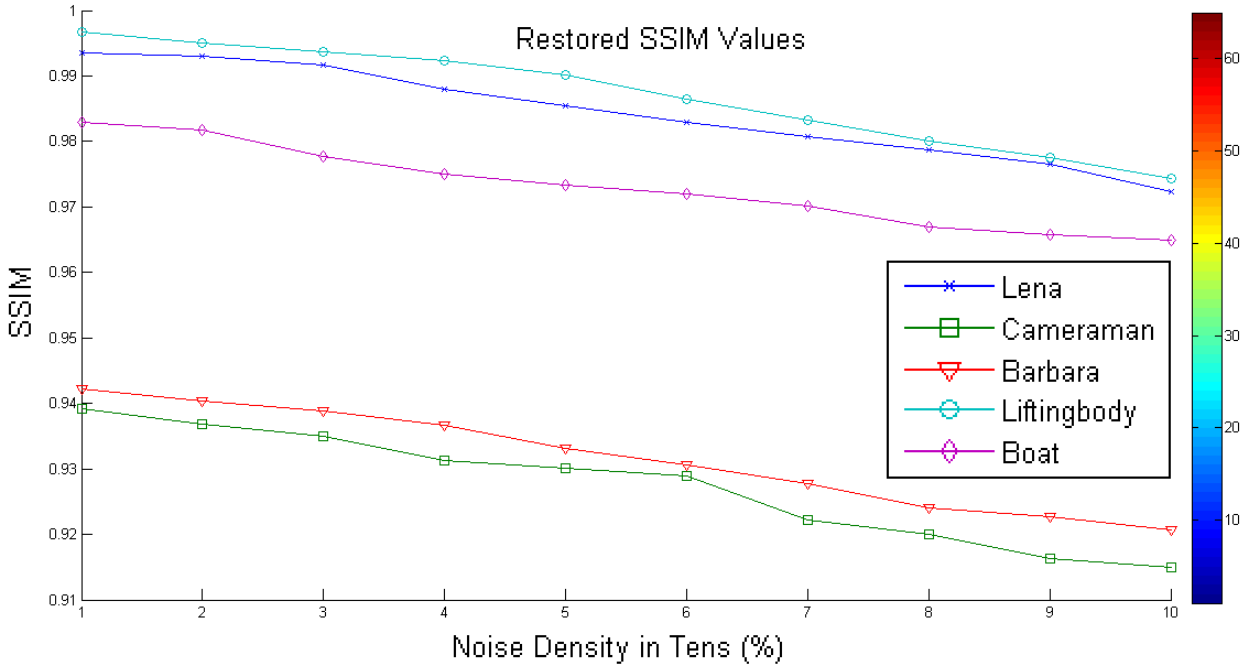


Figure 9.59: Graph for **FSMF** in SSIM for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs

filter. The comparative results of the **FSMF** and existing eight algorithms can be seen in the Fig. 9.69.

In table 9.57, the results by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA, FBDA and NASRI algorithms. In this table, the other proposed algorithms which are compared in the previous sections are also given in the table. In this table, all results are generated on the images corrupted by RVN. The *Average IEF* values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it can be shown that for the entire noise density, the algorithm **DPSO** performs excellent in terms of IEF compared to any algorithm in the table. The NDE filter performs better than the NASRI filter for the noise densities 10%, 20%, 30%, 50%, 60%, 70%, 80%, 90% and 100%. The FBDA filter performs better than ANDWP filter. The comparative results of the **FSMF** and existing eight algorithms can also be seen in the Fig. 9.70. From the figure, it can be cleared that the **DPSO** algorithm has given much better restoration results in terms of IEF than the existing algorithms for the entire noise density. The DPSO also defeats the FSMF filter. The algorithm also generated better IEF than other proposed filters such as DWMD, ANDWP, VMM, EPR, SIR, GASIR, PSOSIR and NDE.

Table 9.49: Results on Bit Error Rate on 30% and 90% noisy Lena image corrupted with SPN and RVN by the proposed filters

30% Noisy			90% Noisy	
Filter	Number	Ratio	Number	Ratio
(1) SPN				
DWMD	275774	0.1315	463760	0.2211
SIR	255249	0.1217	457934	0.2183
GASIR	254674	0.1214	372358	0.1776
PSOSIR	241580	0.1152	353478	0.1686
NDE	212613	0.1014	336461	0.1604
FSMF	144014	0.0686	296751	0.1415
DPSO	107582	0.0513	283035	0.1350
(2) RVN				
DWMD	285674	0.1362	487609	0.2325
SIR	267080	0.1273	436690	0.2082
GASIR	256164	0.1221	409800	0.1954
PSOSIR	232800	0.1110	368090	0.1755
NDE	194014	0.0925	326065	0.1554
FSMF	158075	0.0753	318595	0.1519
DPSO	132420	0.0631	307854	0.1468

In table 9.58, the restoration results by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA, FBDA and NASRI algorithms. In this table all results are generated on the images corrupted by SPN. The *Average SSIM* values for all the algorithms are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it is seen that for the entire noise density, the algorithm **DPSO** performs the best in terms of SSIM. The FSMF filter performs better than existing algorithms. The NASRI algorithm performs marginally better then the NDE filter. The SMF and AMF perform very poor where as DWM, EDBA and IDBA perform similarly. The BDND and FBDA have given similar SSIM values. The comparative visual results of the **FSMF** and existing eight algorithms can be seen in the Fig. 9.71.

In table 9.59, the restoration results by the proposed algorithms are compared to SMF, AMF, DWM, EDBA, IDBA, FBDA and NASRI algorithms. In this table, the other proposed algorithms which are compared in the previous sections are also considered in the table. In this table, all results are generated on the images corrupted by RVN. The *Average SSIM*

Table 9.50: Results on Bit Error Rate on 30% and 90% noisy Cameraman image corrupted with SPN and RVN by the proposed filters

30% Noisy			90% Noisy	
Filter	Number	Ratio	Number	Ratio
(1) SPN				
DWMD	87429	0.1668	112091	0.2138
SIR	78232	0.1492	104673	0.1996
GASIR	73991	0.1411	95787	0.1827
PSOSIR	68388	0.1304	87783	0.1674
NDE	64999	0.1240	83192	0.1587
FSMF	57176	0.1091	77865	0.1485
DPSO	45736	0.0872	62885	0.1199
(2) RVN				
DWMD	108705	0.2073	134115	0.2558
SIR	92019	0.1755	100511	0.1917
GASIR	77305	0.1474	99360	0.1895
PSOSIR	76028	0.1450	80245	0.1531
NDE	70486	0.1344	71012	0.1354
FSMF	59704	0.1139	67702	0.1291
DPSO	59521	0.1135	60104	0.1146

values are shown in this table for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with different noise density from 10% to 100%. From this table, it can be shown that for the entire noise density, the **DPSO** algorithm performs excellent in terms of SSIM. The FBDA filter performs better than VMM filter. The comparative results of the **DPSO** and existing eight algorithms can also be seen in the Fig. 9.72. From the figure, it can be cleared that the **DPSO** algorithm obtained much better restoration results in terms of SSIM than the existing algorithms for the entire noise density. The algorithm also has obtained better SSIM than other proposed filters such as DWMD, ANDWP, VMM, EPR, SIR, GASIR, PSOSIR, NDE and FSMF. Proposed filters are also compared with the existing filters for color image also. Color *Lena* image is applied to the eight existing filters and to the three proposed filters in this section. The noise percentage considered is 50% by salt and pepper noise. Fig. 9.73 represents the restored images to show the qualitative performances of the filters for the 50% noisy color *Lena* image. Figures c-j represent the corresponding restored color images by the filters SMF, AMF, DWM, EDBA, IDBA, BDND, FBDA and

Table 9.51: Restoration results in terms of PSNR on Quantization noise on Lena and Cameraman images by proposed filters

Filter/Image	Lena	Cameraman
DWMD	24.10	21.52
ANDWP	24.35	22.06
VMM	25.32	23.00
EPR	25.78	23.76
SIR	26.76	24.31
GASIR	28.11	25.87
PSOSIR	28.31	26.20
NDE	31.76	28.20
FSMF	33.44	29.69
DPSO	34.59	31.76

Table 9.52: Restoration results in PSNR (dB) on 50% anisotropic noise on Lena and Cameraman images by the proposed filters

Filter	50% Rows	50% Cols	25% Rows & 25% Cols	50% Rows & 50% Cols
(1) Lena				
DWMD	25.41	26.45	26.80	28.21
SIR	28.50	30.11	30.12	30.55
GASIR	30.10	31.23	31.50	32.60
PSOSIR	30.15	31.75	31.90	33.15
NDE	31.95	32.30	32.51	33.79
FSMF	33.32	32.75	32.80	35.65
DPSO	33.70	33.10	34.26	37.77
(2) Cameraman				
DWMD	17.55	19.65	22.80	21.94
SIR	21.20	20.80	25.20	23.70
GASIR	21.39	21.20	25.78	23.90
PSOSIR	21.55	21.40	26.54	24.80
NDE	23.94	22.32	27.55	26.67
FSMF	25.11	24.32	28.15	28.75
DPSO	25.86	25.77	29.00	29.03

Table 9.53: Restoration results in terms of UIQI on SPN and RVN on Lena and Cameraman images corrupted by 50% noise density by proposed filters

Filter/Image	Lena	Cameraman
(1) SPN		
DWMD	0.9390	0.7400
SIR	0.9562	0.7935
GASIR	0.9600	0.8290
PSOSIR	0.9740	0.8540
NDE	0.9890	0.9340
FSMF	0.9980	0.9400
DPSO	0.9990	0.9430
(2) RVN		
DWMD	0.9200	0.7220
SIR	0.9450	0.8090
GASIR	0.9780	0.8941
PSOSIR	0.9825	0.9080
NDE	0.9870	0.9176
FSMF	0.9910	0.9287
DPSO	0.9913	0.9357

NASRI. Figures k-m show the restored images by the proposed filters **NDE, DPSO** and **FSMF**. It is seen from these two figures that these filters can remove the noises very well and also preserves the image fine textures, compared to all other output images by the filters. Color *Lena* image is applied to the eight existing filters and to the ten proposed filters in this section. The noise percentage considered is 50% by random valued noise. Fig. 9.74 represents the restored images to show the qualitative performances of the filters for the 50% noisy color *Lena* image. Figures c-h represent the corresponding restored color images by the filters SMF, AMF, DWM, EDBA, IDBA, BDND, FBDA and NASRI. Figures i-r show the restored images by the proposed filters DWMD, ANDWP, VMM, EPR, SIR, GASIR, PSOSIR, NDE, DPSO and FSMF respectively. It is seen from these figures that proposed filters can remove the noises very well and also preserves the image fine textures, compared to all other output images by the filters.

The time required for execution of the proposed algorithms along with some existing algorithms for restoration of *Lena*, *Cameraman*, *Liftingbody Boat* and *Barbara* images having noise density of 50% using salt and pepper noise and random valued noise are shown in tables 9.60 and 9.61 respectively. These values are given in terms of seconds. In case of

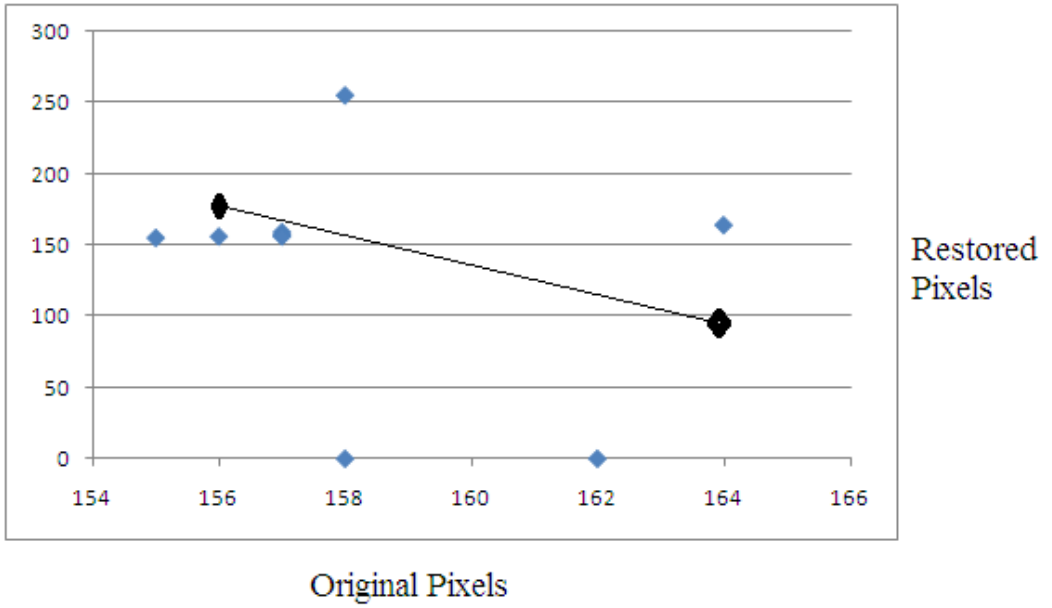


Figure 9.60: Graph representing the regression line of restored pixels by DWMD filter against original pixels for 3×3 sub-image of *Lena* image corrupted by SPN with 30% noise density

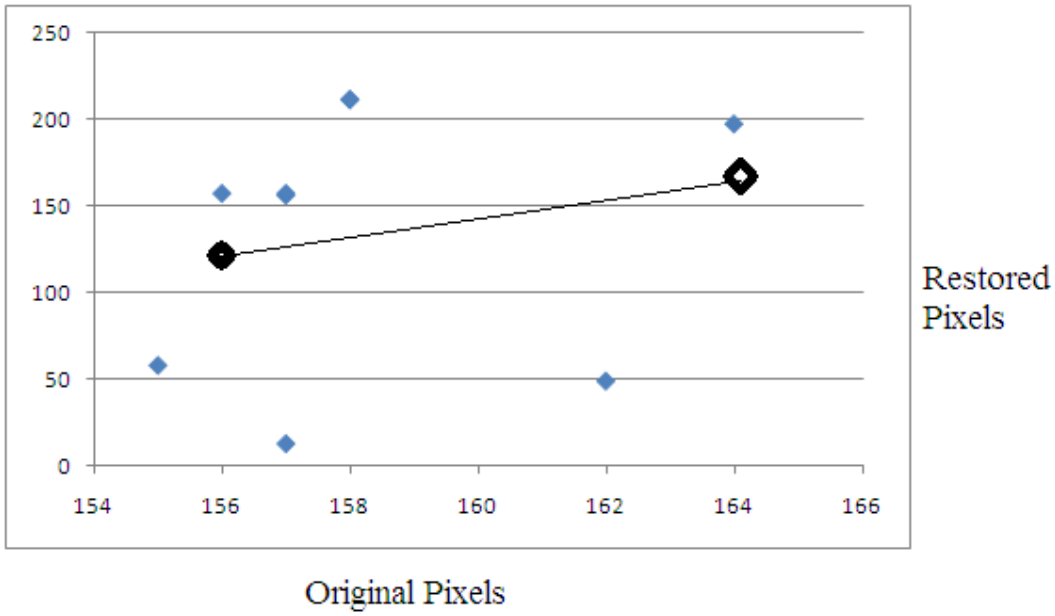


Figure 9.61: Graph representing the regression line of restored pixels by DWMD filter against original pixels for 3×3 sub-image of *Lena* image corrupted by RVN with 30% noise density

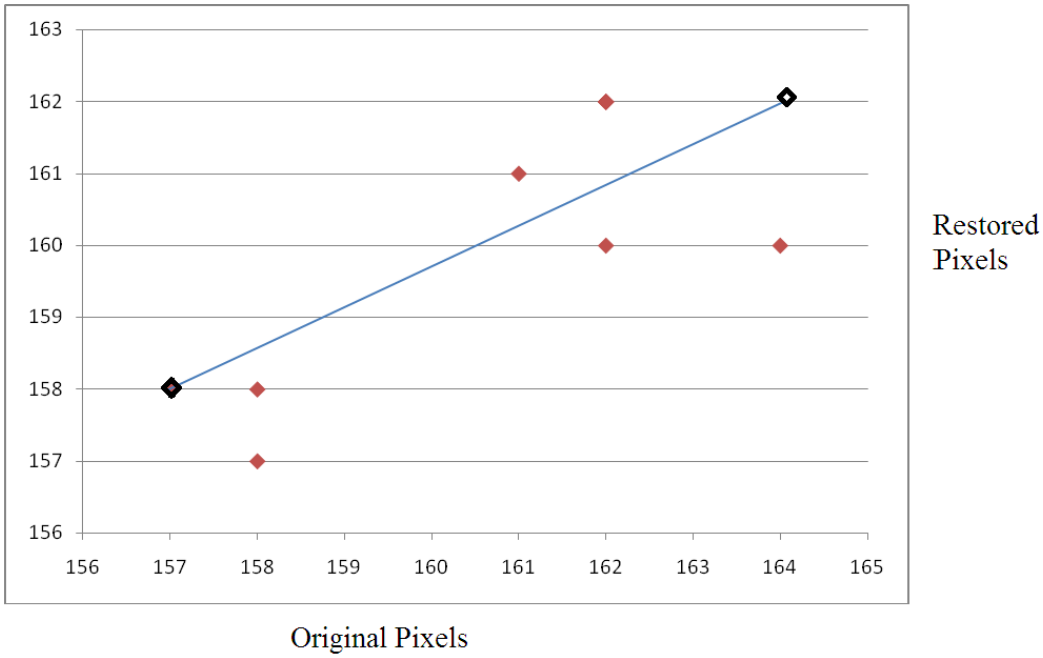


Figure 9.62: Graph representing the regression line of restored pixels by DPSO filter against original pixels for 3×3 sub-image of *Lena* image corrupted by SPN with 30% noise density

noise model 1 or salt and pepper noise (SPN), the required execution time for the standard median filter is very low for all the images used. The DWM filter performs excellent in terms of execution time. The EDBA takes more execution time than existing algorithms. Among the proposed algorithms, the FSMF filter needs less amount of time than the others. The NDE and DPSO require much larger times than all other algorithms. The proposed FSMF operator performs significantly better than the existing operators due to small window size and single time iteration used for the noise detection and filtering operations. The graphical representation in terms of execution time for the salt and pepper noise is given in Fig. 9.75. For the random valued noise the SMF and DWM filters perform excellent compared to other existing algorithms. The FBDA also requires less amount of time compared to other algorithms. Among the proposed algorithms the DWMD, ANDWP, VMM and EPR requires less amount of times. The PSOSIR defeats the GASIR algorithms, however they require more time than the SIR filter. The NDE requires maximum time for execution. The DPSO takes much more time than the others. The FSMF filter requires very less time compared to DPSO. The graphical representation in terms of execution time for the random valued noise is given in Fig. 9.76.

In case of Random Valued Noise, The execution time requirement of the proposed algo-

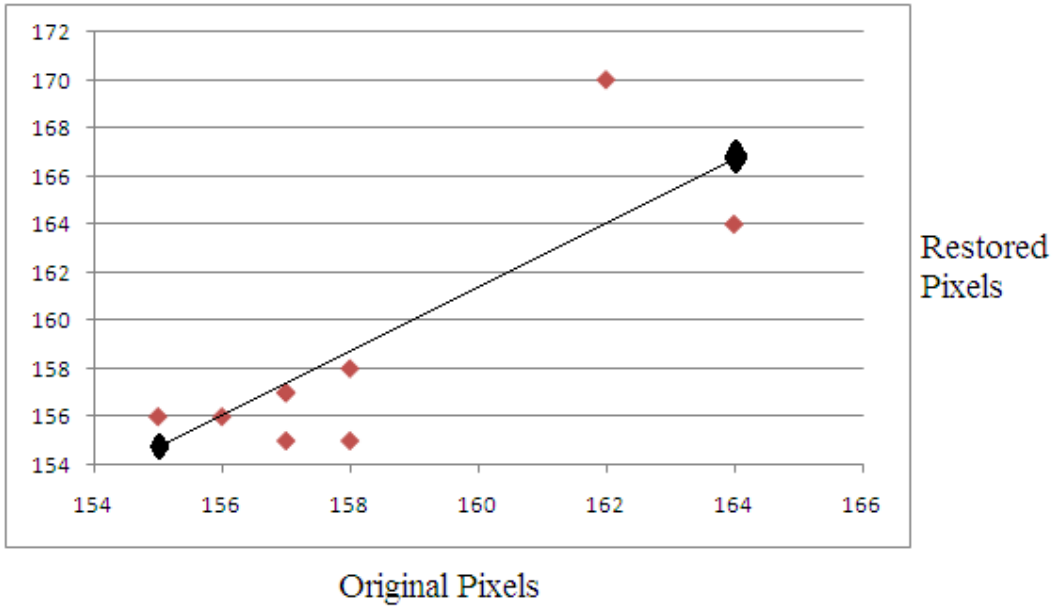
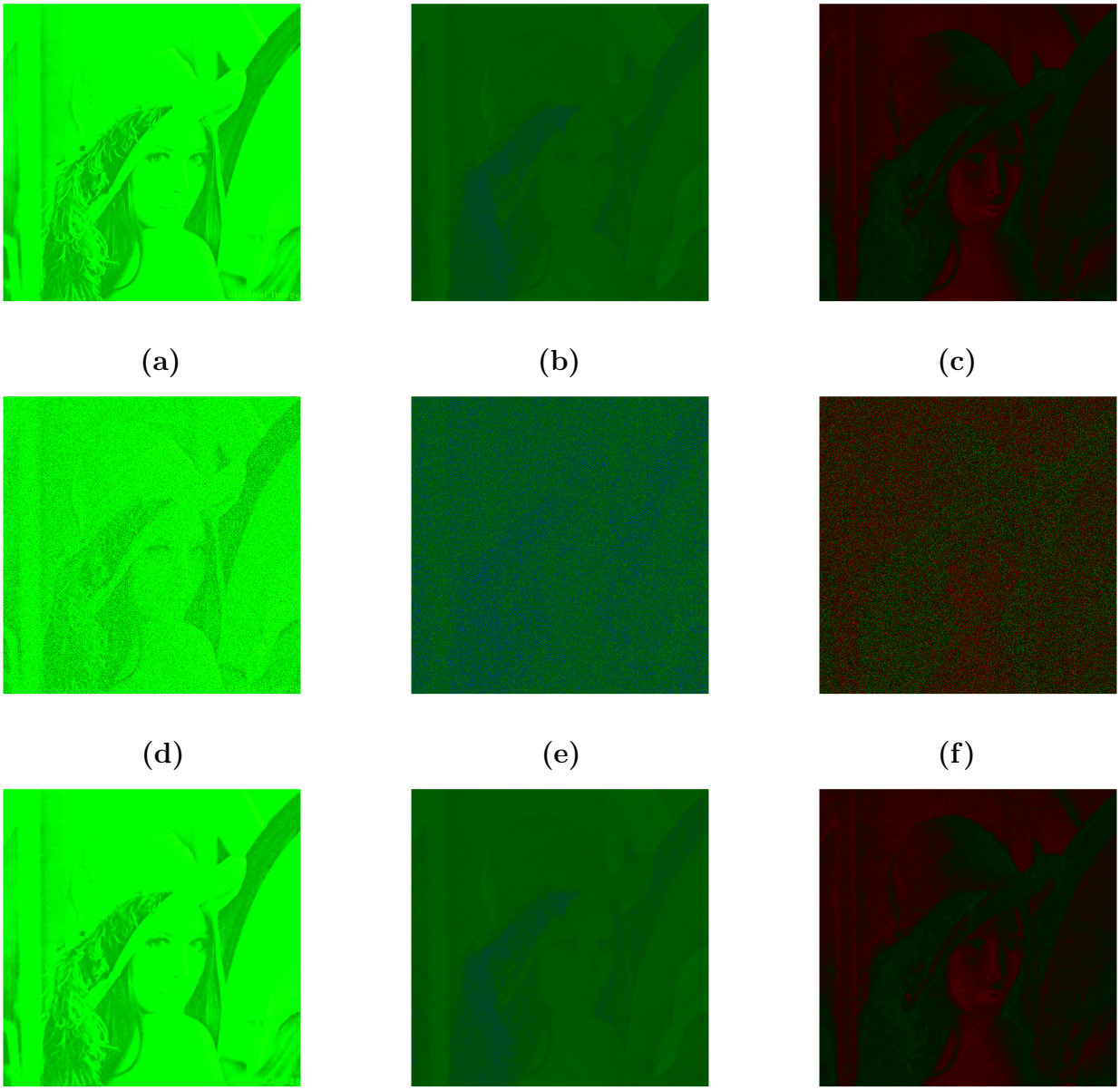


Figure 9.63: Graph representing the regression line of restored pixels by DPSO filter against original pixels for 3×3 sub-image of *Lena* image corrupted by RVN with 30% noise density

Table 9.54: Comparison of *Average PSNR* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs

Filter	**Noise Density/Average PSNR(dB)**									
	10	20	30	40	50	60	70	80	90	100
SMF	32.22	30.56	28.51	26.76	24.59	22.76	20.40	18.89	17.57	16.87
AMF	32.87	30.79	28.76	28.45	26.89	24.81	22.47	19.19	17.88	16.90
DWM	33.65	31.35	30.87	29.66	27.63	25.93	24.09	22.11	19.45	17.63
EDBA	34.31	32.87	31.60	30.50	28.00	26.60	25.30	23.69	20.44	18.33
IDBA	35.29	33.70	31.79	31.10	29.50	27.11	25.54	24.25	20.32	19.06
BDND	35.41	33.69	31.50	31.32	30.79	30.00	26.32	25.21	21.25	20.62
FBDA	35.49	34.03	31.63	31.65	31.03	30.41	27.23	25.30	22.78	21.20
NASRI	37.13	35.42	33.10	33.21	32.71	29.64	28.61	26.54	24.35	22.50
NDE	36.68	34.74	33.62	32.10	31.15	29.26	28.16	26.43	25.39	23.69
DPSO	38.52	36.51	35.45	34.00	33.02	31.13	30.11	28.33	27.38	25.59
FSMF	37.17	35.40	33.95	32.66	31.56	29.68	28.44	27.39	26.14	25.11

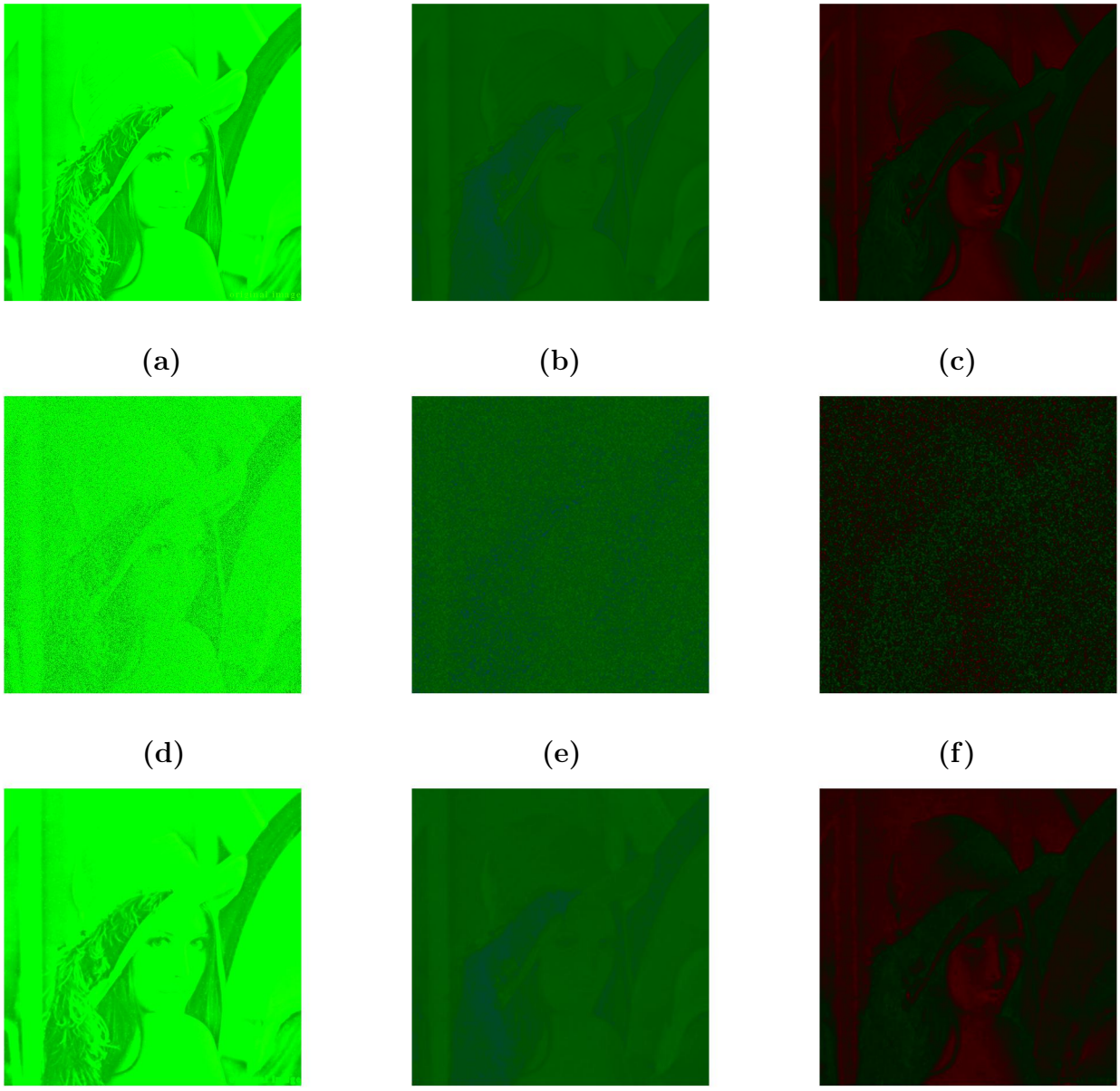


(a) **(b)** **(c)**

(d) **(e)** **(f)**

(g) **(h)** **(i)**

Figure 9.64: 50% noisy color Lena image corrupted with SPN formed by Y, Cb and Cr components from left to right. First row shows the components of original image. Similarly second and third rows show the components of noisy and restored images by DPSO filter respectively



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

(d) **(e)** **(f)**

(g) **(h)** **(i)**

Figure 9.65: 50% noisy color Lena image corrupted with RVN formed by Y, Cb and Cr components from left to right. First row shows the components of original image. Similarly second and third rows show the components of noisy and restored images by DPSO filter respectively

Table 9.55: Comparison of *Average PSNR* for proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs

Filter	**Noise Density/Average PSNR(dB)**									
	10	20	30	40	50	60	70	80	90	100
SMF	30.72	29.59	27.44	24.65	21.57	19.39	18.55	17.13	16.54	16.11
AMF	31.30	30.13	29.30	28.09	26.71	24.59	22.33	19.04	17.76	16.71
DWM	31.51	30.82	30.31	29.54	27.33	25.70	23.41	21.91	19.12	17.04
EDBA	32.09	30.54	30.10	29.12	27.01	24.66	23.81	21.71	19.32	17.23
IDBA	32.24	31.01	30.44	28.20	26.32	24.70	23.03	20.12	18.65	17.49
BDND	32.54	31.12	30.40	29.32	27.67	25.71	23.01	21.00	19.54	18.10
FBDA	33.62	31.90	31.22	30.40	28.76	26.79	24.92	23.55	21.79	19.70
NASRI	34.75	33.22	32.43	31.79	30.13	28.23	26.25	24.67	23.07	21.00
DWMD	32.81	31.46	30.42	28.80	26.71	25.19	23.17	21.78	20.84	19.35
ANDWP	33.62	32.42	31.07	29.41	27.47	25.65	23.58	21.86	21.09	19.77
VMM	33.90	32.66	31.63	29.72	27.91	26.11	24.37	22.46	21.75	20.72
EPR	34.16	32.69	31.88	30.13	28.20	26.38	24.74	22.91	22.24	21.11
SIR	34.73	33.48	32.37	30.53	28.83	26.80	25.05	23.30	22.68	21.78
GASIR	35.39	33.98	32.68	30.98	29.23	27.33	25.93	24.27	23.06	22.25
PSOSIR	35.52	34.12	32.87	31.14	29.36	27.42	26.10	24.46	23.22	22.49
NDE	35.95	34.49	33.24	31.41	29.90	27.81	26.57	24.40	23.53	22.74
DPSO	37.85	36.34	35.12	33.22	31.83	29.69	28.54	26.29	25.31	24.76
FSMF	36.55	35.29	33.79	32.27	30.60	28.60	27.10	25.48	24.15	22.97

rithms such as DWMD and ANDWP are less than all other existing algorithms except the SMF filter. So the performances of the DWMD and ANDWP are better than all algorithms except the SMF filter. The proposed VMM and EPR require comparable time with the existing algorithms such as BDND, FBDA and BDND. The proposed SIR filter require less time than AMF, EDBA and NASRI algorithms. The PSOSIR takes less time than AMF and EDBA filters. The GASIR require equivalent time to EDBA algorithm. The FSMF filter performs second best in the table, which takes marginal larger time than the SMF filter. Only two proposed algorithms such as NDE and DPSO take two seconds more time than the existing algorithms. In case of Salt-and-Pepper Noise, FSMF filter takes less time than any existing algorithm except the SMF and DWM filters. The proposed NDE and DPSO take marginally larger time than EDBA, IDBA, NASRI and AMF filters.

The execution times of most of the proposed algorithms are better or comparable to the

Table 9.56: Comparison of *Average IEF* for proposed filters in for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs

Filter	**Noise Density/Average IEF**									
	10	20	30	40	50	60	70	80	90	100
SMF	56.82	48.89	41.76	32.50	27.25	21.00	17.60	14.76	10.00	7.75
AMF	63.81	50.50	46.00	41.56	37.09	31.20	29.22	25.50	23.10	18.00
DWM	144.75	127.25	114.56	86.33	69.70	47.89	40.25	29.17	21.00	18.91
EDBA	197.50	156.50	144.25	127.60	92.64	72.90	67.08	53.75	44.72	40.00
IDBA	223.55	204.32	190.75	165.80	145.90	123.43	93.25	77.94	65.60	54.10
BDND	256.80	249.96	221.09	202.95	178.60	176.25	123.55	111.17	103.04	92.81
FBDA	295.10	263.72	266.50	242.55	237.01	210.50	176.69	143.50	130.56	115.95
NASRI	417.27	366.87	332.25	304.47	283.90	257.28	252.23	221.29	226.76	176.21
NDE	405.32	373.14	346.48	326.16	303.59	290.95	265.92	251.50	237.23	206.44
DPSO	436.24	402.20	379.11	355.01	335.04	319.36	290.79	281.37	262.83	234.42
FSMF	415.30	392.25	364.65	333.70	327.76	317.26	271.66	262.76	240.88	233.50

existing algorithms. Only two among them require larger execution time because they also obtain excellent restoration results compared to the state of the art methods. Among the ten proposed algorithms only NDE and DPSO require larger execution time. The NDE is not only a denoising algorithm but also it estimates the noise density in images and hence it needs to run for longer time for noise estimation. The DPSO is different length particle swarm optimization algorithm for image denoising which also perform image clustering in addition to denoising. The extra amount of time needed to partition the image prior to denoising.

Table 9.57: Comparison of *Average IEF* for the proposed filters in for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs

Filter	**Noise Density/Average IEF**									
	10	20	30	40	50	60	70	80	90	100
SMF	51.77	34.56	28.98	17.65	14.44	10.91	8.01	7.88	7.12	5.01
AMF	82.87	67.99	67.12	51.81	38.73	35.13	27.39	23.91	19.91	15.19
DWM	131.99	120.82	104.83	81.73	59.71	39.70	21.39	20.11	18.97	18.89
EDBA	169.61	139.01	135.92	115.10	79.18	69.99	62.99	47.22	40.63	32.91
IDBA	201.50	187.88	165.61	139.40	116.33	100.49	81.71	62.81	59.51	51.99
BDND	231.90	220.11	201.91	169.40	142.01	109.11	90.90	86.25	82.10	76.51
FBDA	296.55	312.89	278.25	223.50	187.25	160.50	129.75	105.25	94.50	90.88
NASRI	388.50	353.89	323.65	306.29	267.29	245.25	238.15	228.16	198.30	171.59
DWMD	243.49	229.09	211.68	194.12	173.91	153.04	118.89	107.34	92.37	81.18
ANDWP	292.11	265.94	256.01	233.98	220.43	190.05	160.70	135.30	111.74	101.05
VMM	301.24	274.20	260.68	249.23	235.42	220.16	191.24	159.44	124.52	111.43
EPR	324.84	293.67	273.15	264.04	250.97	229.63	213.03	189.50	156.01	133.09
SIR	346.35	315.79	287.69	276.24	263.20	251.64	230.00	211.90	180.66	151.88
GASIR	372.37	344.05	306.62	287.24	274.48	259.06	240.25	225.00	202.85	171.09
PSOSIR	384.63	350.23	314.24	296.35	285.00	266.10	245.58	230.37	211.38	177.48
NDE	390.93	354.68	325.78	306.28	289.91	276.47	254.19	242.15	217.95	189.48
DPSO	411.60	381.37	348.93	332.41	313.89	300.89	277.62	262.51	240.67	211.60
FSMF	391.82	360.38	318.24	294.28	275.44	259.45	238.01	219.72	197.41	173.71

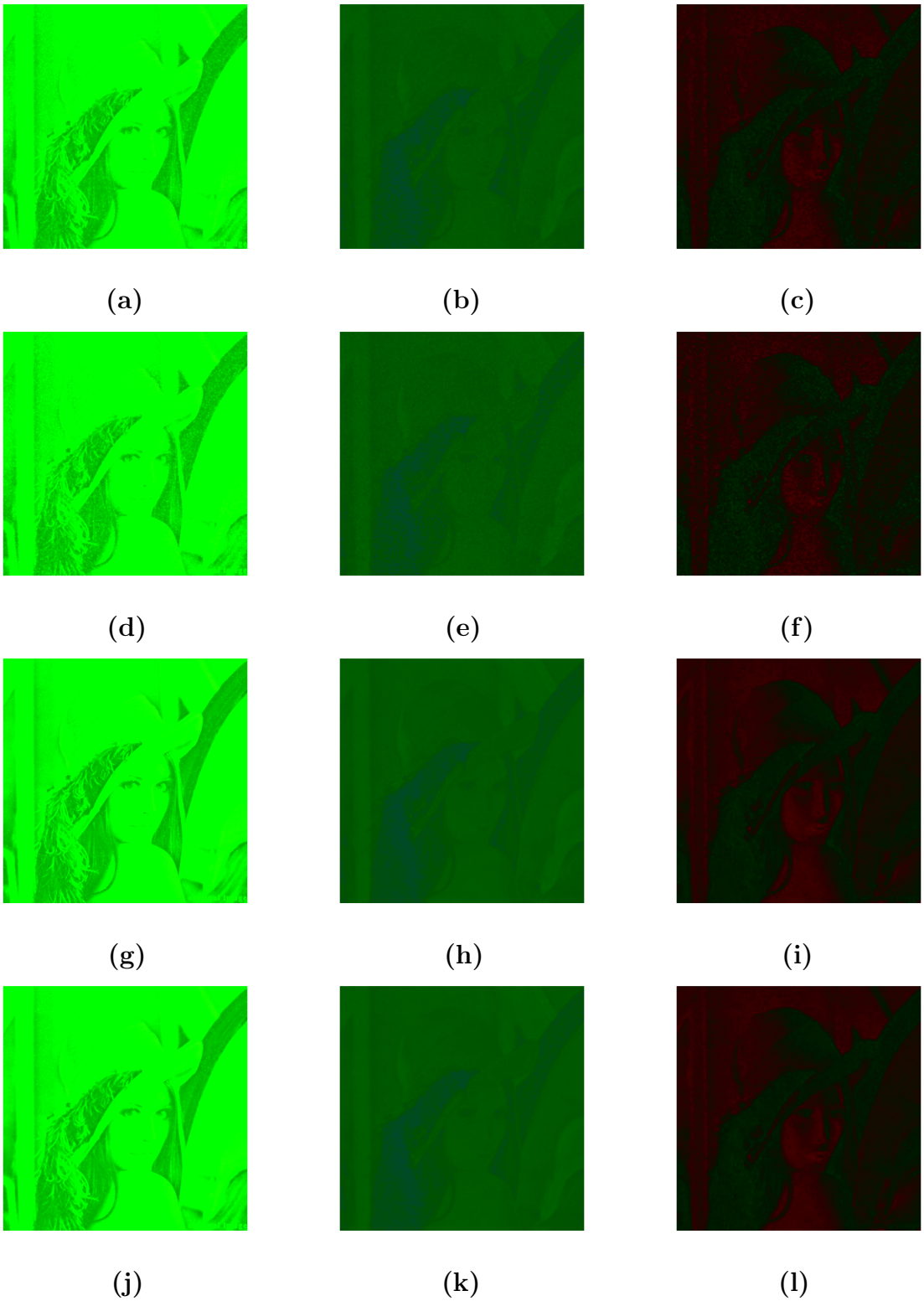


Figure 9.66: First and second rows show the components of restored images by DWMD filter with SPN and RVN respectively. Similarly third and fourth rows show the components of restored images by FSMF filter

Somnath Mukhopadhyay, University of Kalyani, India

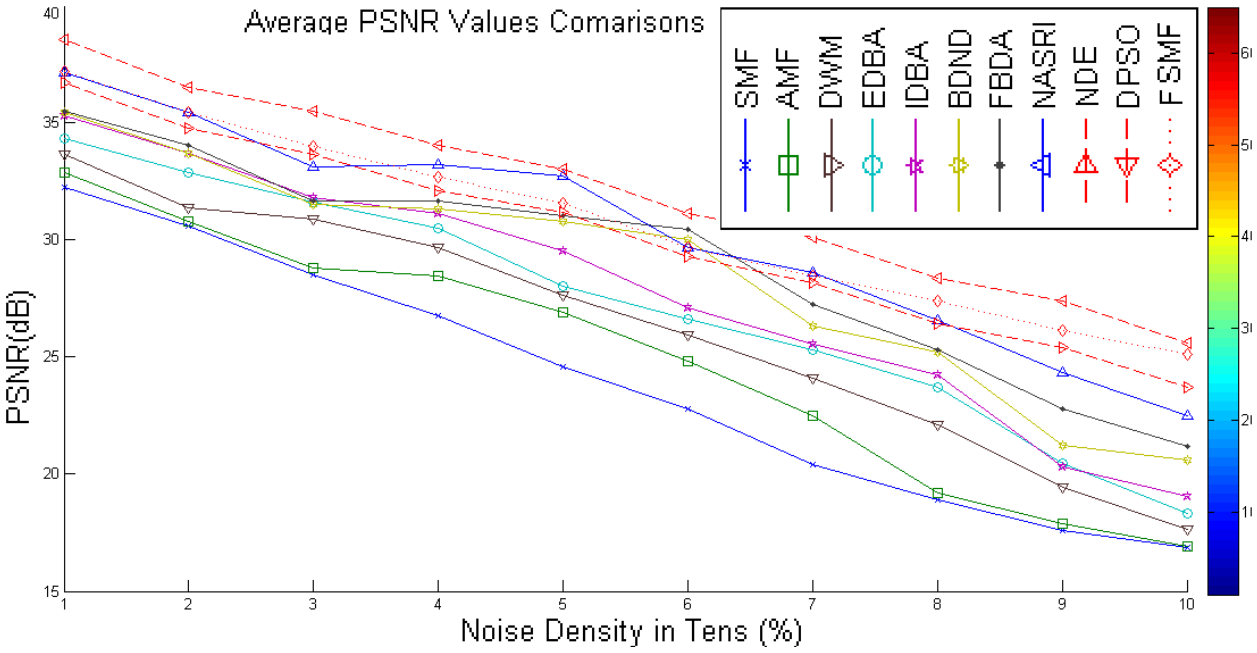


Figure 9.67: Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and **FSMF** filter

Table 9.58: Comparison of *Average SSIM* for the proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPNs

Filter	**Noise Density/Average SSIM**									
	10	20	30	40	50	60	70	80	90	100
SMF	0.9423	0.9417	0.9384	0.9328	0.9319	0.9265	0.9201	0.9156	0.9127	0.9103
AMF	0.9476	0.9425	0.9369	0.9333	0.9297	0.9265	0.9240	0.9222	0.9204	0.9178
DWM	0.9507	0.9477	0.9451	0.9445	0.9410	0.9378	0.9256	0.9210	0.9176	0.9165
EDBA	0.9565	0.9541	0.9501	0.9467	0.9420	0.9370	0.9342	0.9311	0.9267	0.9229
IDBA	0.9583	0.9567	0.9540	0.9427	0.9420	0.9413	0.9391	0.9358	0.9342	0.9311
BDND	0.9622	0.9607	0.9585	0.9571	0.9535	0.9515	0.9500	0.9457	0.9406	0.9376
FBDA	0.9687	0.9620	0.9607	0.9589	0.9554	0.9527	0.9501	0.9478	0.9430	0.9403
NASRI	0.9731	0.9703	0.9681	0.9654	0.9635	0.9620	0.9602	0.9588	0.9561	0.9513
NDE	0.9709	0.9696	0.9685	0.9667	0.9642	0.9603	0.9574	0.9504	0.9443	0.9405
DPSO	0.9801	0.9783	0.9767	0.9747	0.9736	0.9711	0.9700	0.9675	0.9655	0.9629
FSMF	0.9773	0.9752	0.9728	0.9703	0.9682	0.9650	0.9632	0.9598	0.9562	0.9538

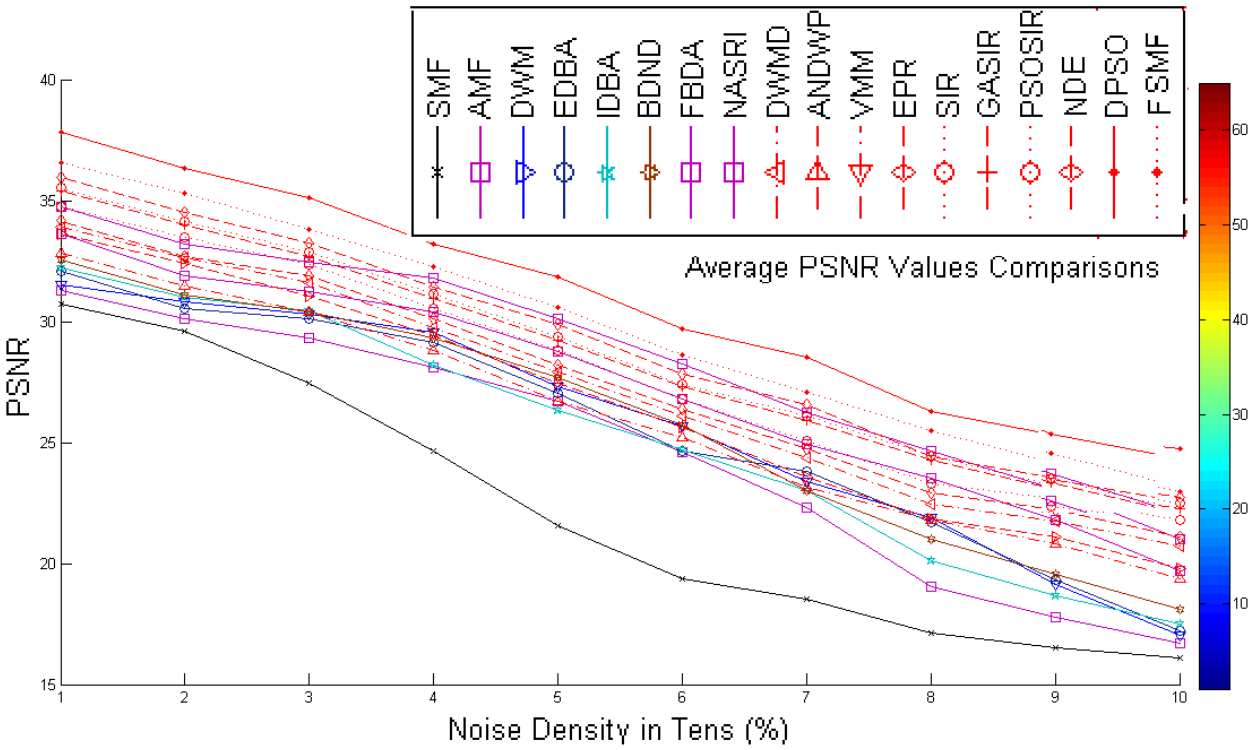


Figure 9.68: Graph of average PSNR values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and **FSMF** filter

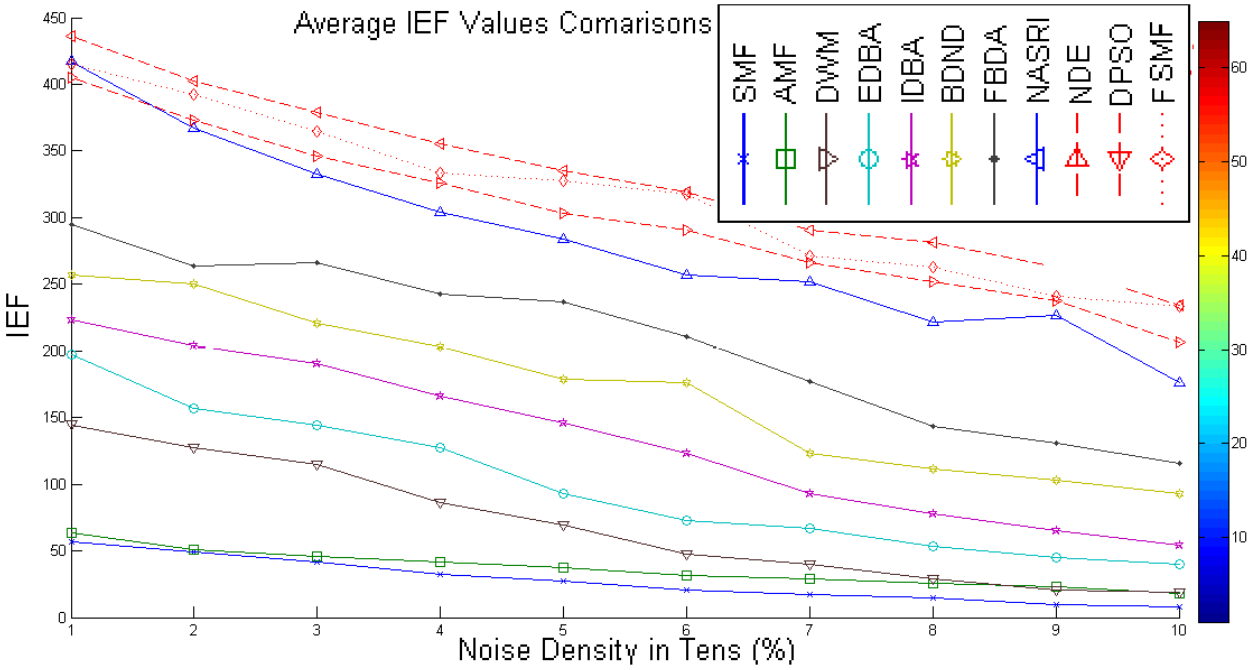


Figure 9.69: Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% SPNs by existing and **FSMF** filter

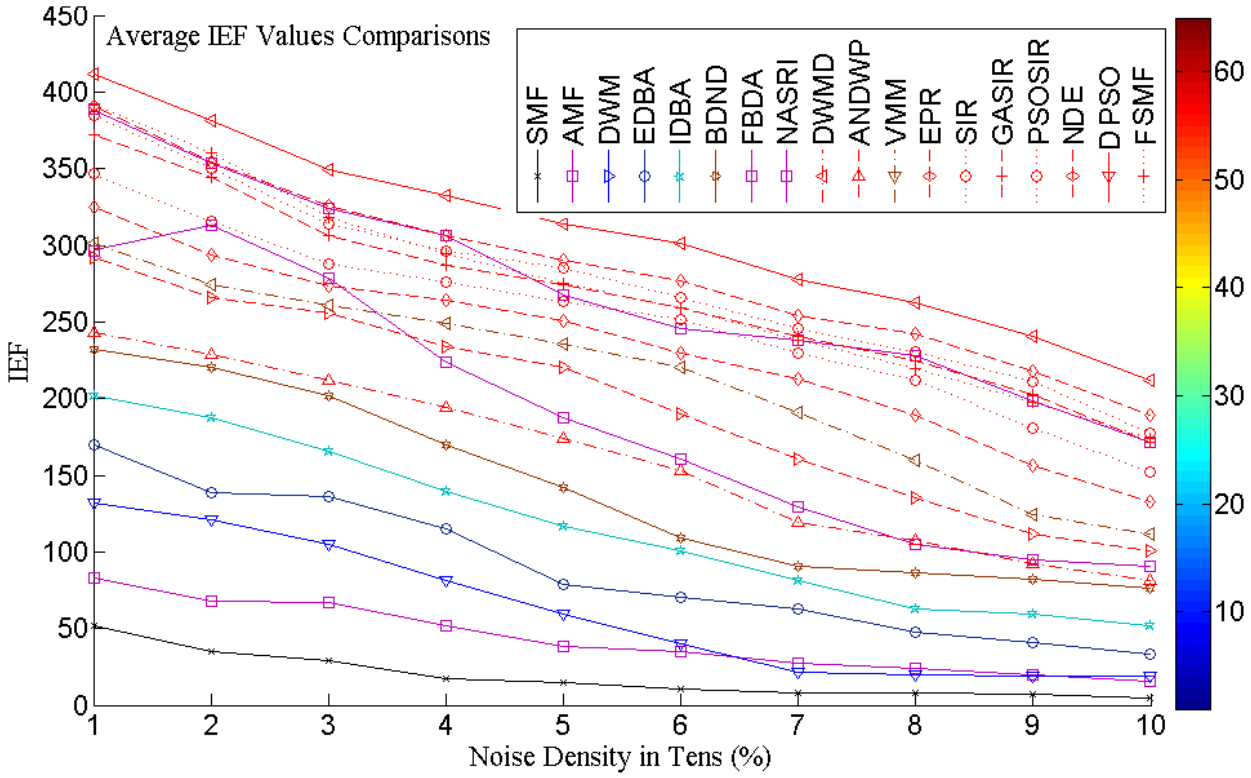


Figure 9.70: Graph of average IEF values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and **FSMF** filter

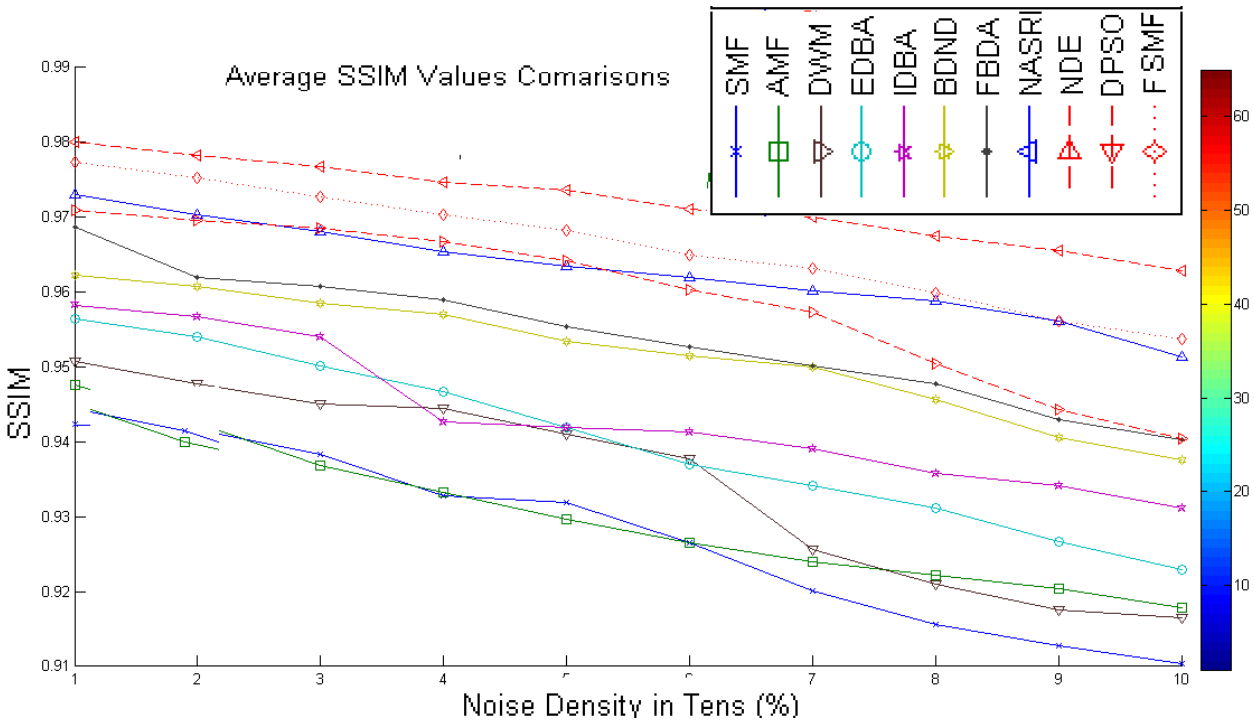


Figure 9.71: Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and **FSMF** filter

Table 9.59: Comparison of *Average SSIM* for the proposed filters for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVNs

Filter	**Noise Density/Average SSIM**									
	10	20	30	40	50	60	70	80	90	100
SMF	0.9362	0.9333	0.9319	0.9287	0.9259	0.9116	0.9087	0.9076	0.9012	0.8981
AMF	0.9412	0.9377	0.9329	0.9310	0.9254	0.9150	0.9076	0.9064	0.9059	0.9045
DWM	0.9420	0.9412	0.9343	0.9312	0.9240	0.9211	0.9150	0.9219	0.9136	0.9107
EDBA	0.9449	0.9440	0.9361	0.9322	0.9315	0.9243	0.9222	0.9217	0.9156	0.9130
IDBA	0.9510	0.9508	0.9455	0.9431	0.9319	0.9311	0.9276	0.9233	0.9176	0.9167
BDND	0.9512	0.9503	0.9430	0.9411	0.9355	0.9330	0.9310	0.9270	0.9235	0.9214
FBDA	0.9598	0.9565	0.9542	0.9507	0.9429	0.9403	0.9376	0.9333	0.9317	0.9310
NASRI	0.9598	0.9565	0.9542	0.9507	0.9429	0.9403	0.9376	0.9333	0.9317	0.9310
DWMD	0.9489	0.9460	0.9387	0.9348	0.9327	0.9296	0.9257	0.9235	0.9172	0.9150
ANDWP	0.9527	0.9491	0.9444	0.9408	0.9377	0.9324	0.9282	0.9232	0.9179	0.9123
VMM	0.9577	0.9561	0.9517	0.9479	0.9426	0.9377	0.9351	0.9338	0.9280	0.9265
EPR	0.9626	0.9623	0.9579	0.9537	0.9501	0.9456	0.9418	0.9402	0.9343	0.9314
SIR	0.9646	0.9645	0.9603	0.9562	0.9539	0.9498	0.9440	0.9424	0.9375	0.9337
GASIR	0.9703	0.9687	0.9667	0.9632	0.9603	0.9560	0.9523	0.9477	0.9398	0.9363
PSOSIR	0.9709	0.9696	0.9685	0.9667	0.9642	0.9603	0.9574	0.9504	0.9443	0.9405
NDE	0.9721	0.9718	0.9706	0.9686	0.9669	0.9642	0.9613	0.9581	0.9558	0.9542
DPSO	0.9740	0.9730	0.9724	0.9712	0.9700	0.9689	0.9679	0.9665	0.9658	0.9644
FSMF	0.9709	0.9694	0.9674	0.9647	0.9624	0.9602	0.9568	0.9540	0.9517	0.9495

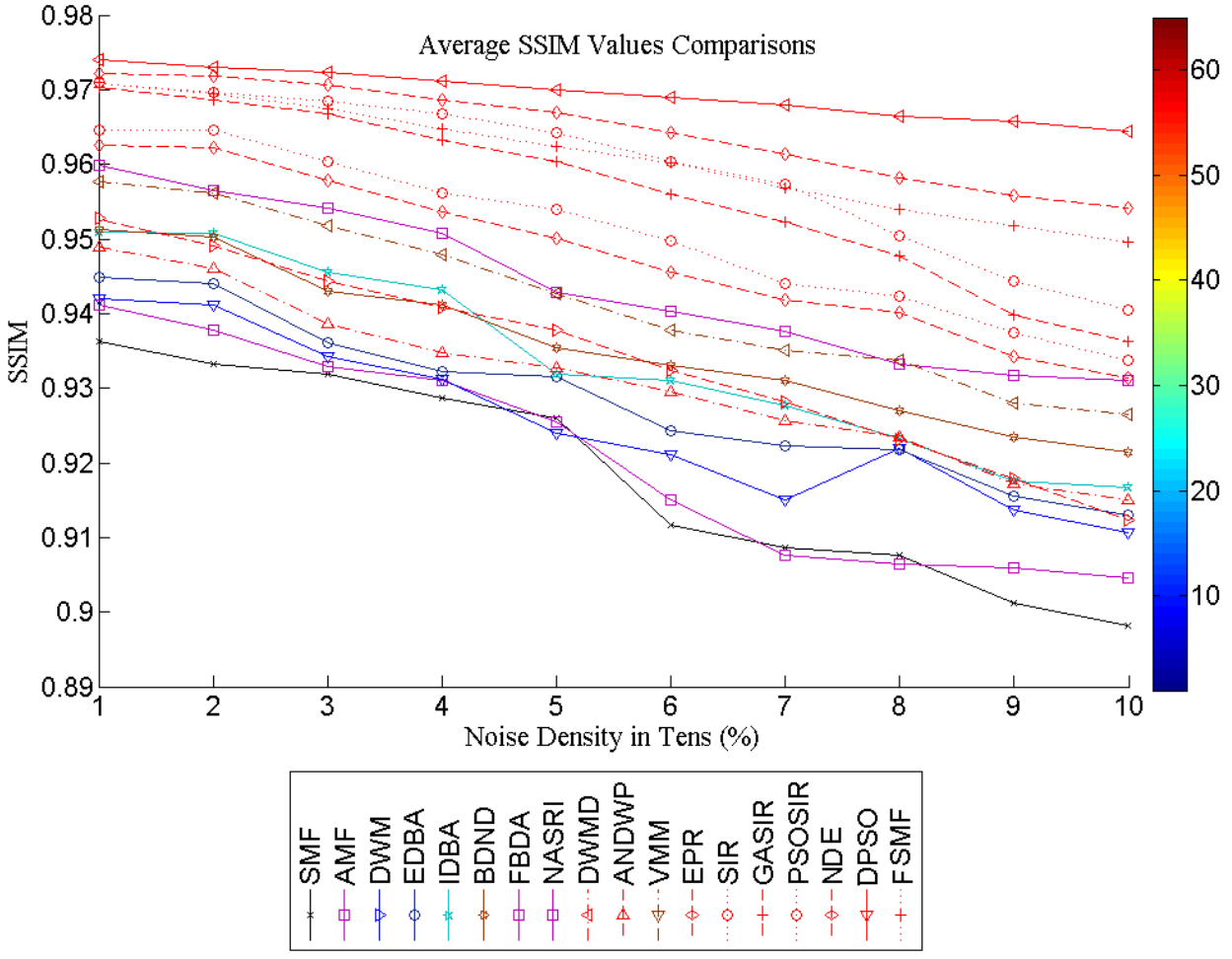


Figure 9.72: Graph of average SSIM values for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images for 10% to 100% RVNs by existing and **FSMF** filter

Figure 9.73: a and b represent the original and 50% noisy color *Lena* images respectively with SPN. Figures c-j represent the corresponding restored color images by the filters SMF, AMF, DWM, EDBA, IDBA, BDND, FBDA and NASRI. Figures k-m show the restored images by the proposed filters **NDE, DPSO** and **FSMF**

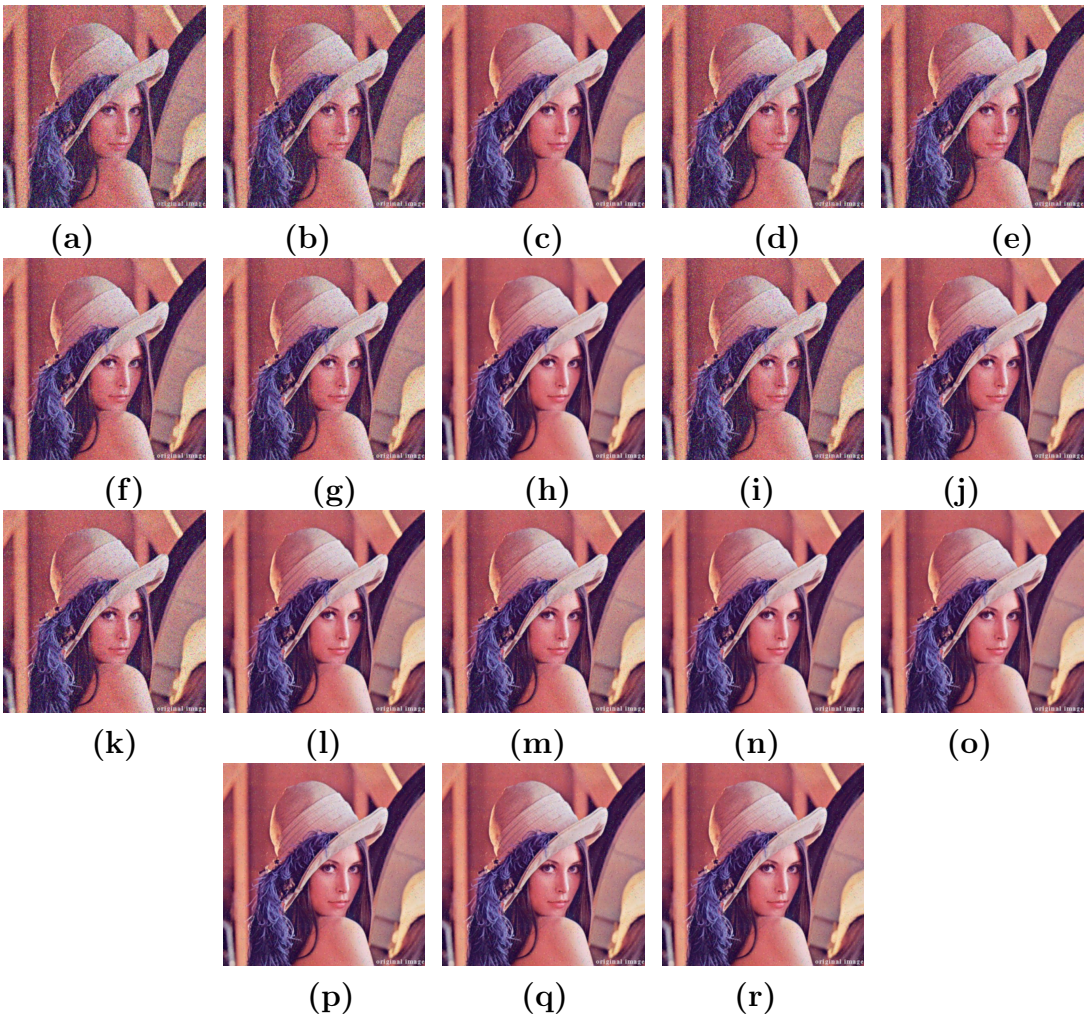


Figure 9.74: a-h represent restored images from 50% noisy color *Lena* image corrupted with RVN, by the filters SMF, AMF, DWM, EDBA, IDBA, BDND, FBDA and NASRI respectively. Figures i-r show the restored images by the proposed filters DWMD, ANDWP, VMM, EPR, SIR, GASIR, PSOSIR, NDE, DPSO and FSMF respectively

Table 9.60: Comparison of **Execution Time** in Seconds for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with SPN

Filter	*Lena*	*Cameraman*	*Liftingbody*	*Barbara*	*Boat*
SMF (3×3)	6.23	5.67	6.97	7.13	7.22
AMF	9.97	9.31	10.32	10.67	9.72
DWM	7.56	7.43	8.76	8.41	8.98
EDBA	10.45	9.34	10.35	9.06	10.32
IDBA	9.86	9.21	10.65	9.88	10.00
BDND	9.42	9.09	10.69	9.32	9.45
FBDA	8.54	8.43	10.33	9.87	9.69
NASRI	9.77	10.32	10.85	9.97	10.50
NDE	11.87	12.30	11.79	11.81	10.80
DPSO	12.78	12.70	12.20	12.40	12.56
FSMF	8.10	8.20	8.20	8.41	8.23

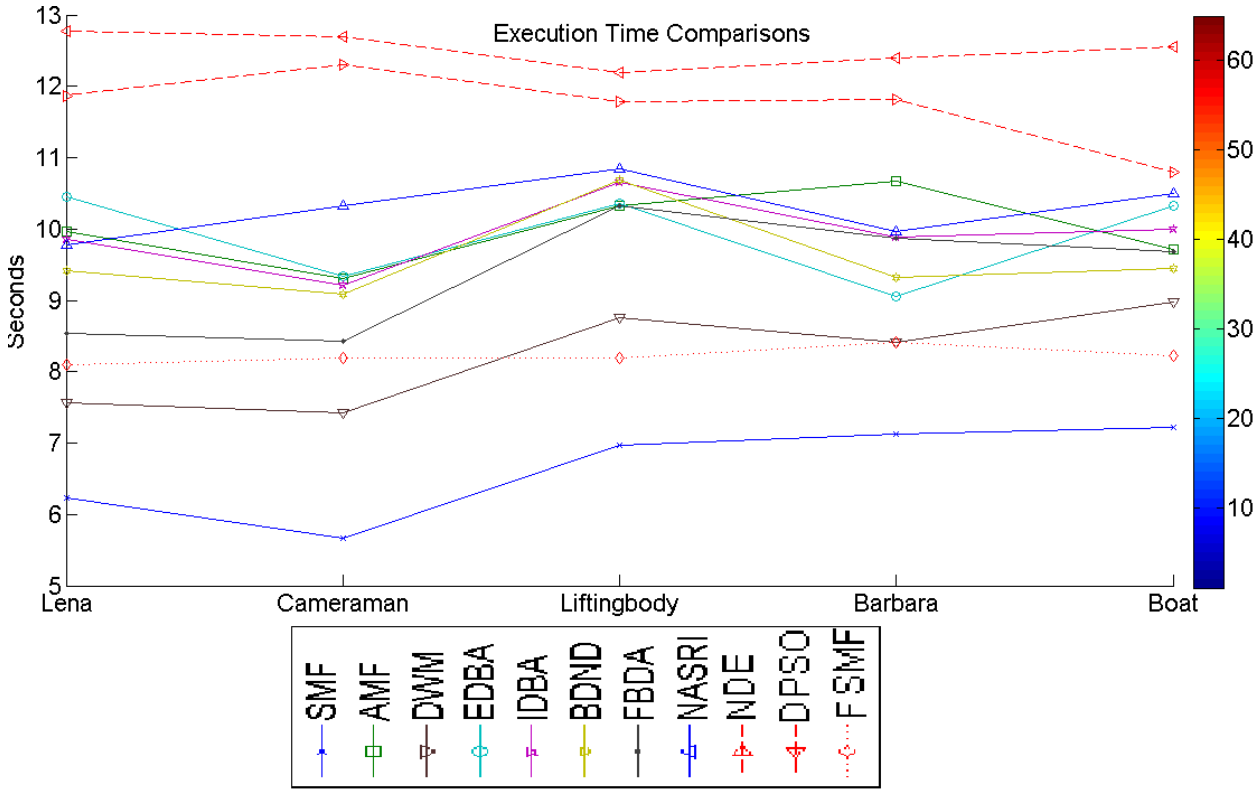


Figure 9.75: Comparisons of execution time (secs) for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images having 50% SPN

Table 9.61: Comparison of **Execution Time** in Seconds for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images with RVN

Filter	*Lena*	*Cameraman*	*Liftingbody*	*Barbara*	*Boat*
SMF (3×3)	7.01	5.52	7.43	7.82	7.91
AMF	10.23	9.72	10.44	10.54	10.00
DWM	8.48	8.42	9.38	8.48	8.87
EDBA	11.12	7.13	9.63	9.71	9.76
IDBA	9.39	8.06	9.86	10.32	9.87
BDND	9.12	7.31	9.37	9.73	10.10
FBDA	8.36	7.27	8.71	8.43	8.87
NASRI	9.90	10.56	10.50	10.20	10.78
DWMD	8.10	7.68	8.41	8.20	8.00
ANDWP	7.67	7.21	7.05	7.67	7.35
VMM	8.82	7.41	7.80	8.01	7.39
EPR	8.65	7.90	8.50	8.90	8.67
SIR	9.83	10.33	10.29	9.87	10.23
GASIR	11.32	9.39	11.26	10.32	10.21
PSOSIR	10.21	9.30	10.52	9.65	9.65
NDE	13.56	13.72	12.78	11.80	11.60
DPSO	13.12	12.51	12.19	11.40	11.22
FSMF	8.21	7.00	8.16	8.10	8.54

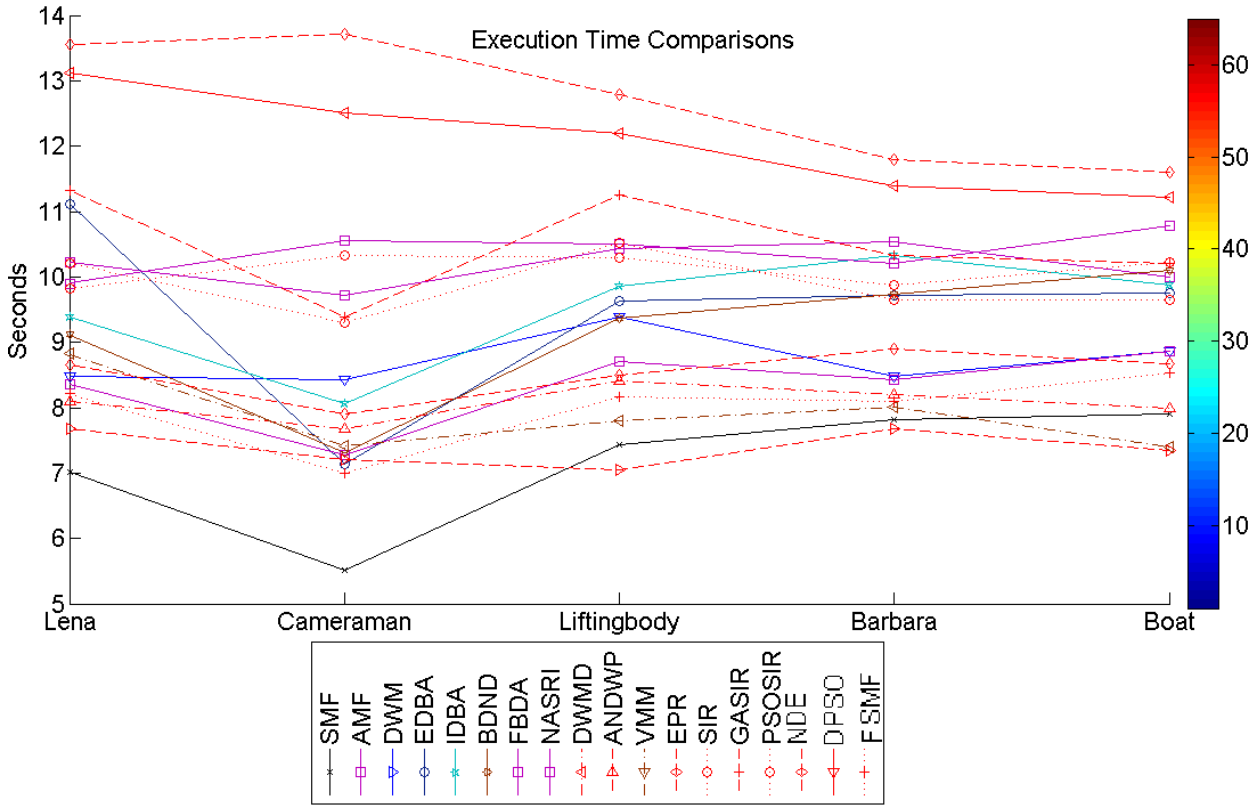


Figure 9.76: Comparisons of execution time (secs) for *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* images having 50% RVN

9.6 Denoising in Frequency Domain

Proposed frequency domain based filtering technique (section 8.1) is validated through experimental results and analysis. The frequency domain based technique has been proposed to denoise the mixed *Gaussian and Poisson* noises from the medical images. Some state of the art filters like VisuShrink [Donoho 1994], SureShrink [Donoho 1995b] and BayesShrink [H.Chipman 1997, S.Chang 2000] are implemented and compared with the proposed GA based BayesShrink method. One ultrasound image is being used for the comparison purpose with other methods. Various graphical representations of the results are given to substantiate the performance of the proposed technique. The ultrasound image is corrupted with low (σ=30), medium (σ=60) and high noise density (σ=90) using the mixed Gaussian and Poisson noises. The performance of the proposed operator is measured quantitatively using peak signal-to-noise ratio (PSNR). Some restoration results using the proposed method are given in section 9.6.1 and the comparisons with the existing operators are given in section 9.6.2. A brief discussion on experimental results and comparisons is given in section 9.7.

9.6.1 Results

The technique given in chapter 8 is a denoising method of medical images corrupted by *Gaussian and Poisson* noises have been thresholded and optimized using *Genetic Algorithm (GA)*. Firstly the noisy image is partitioned into fixed sized blocks and then transformed into wavelet domain. Some important parameters of the two dimensional discrete wavelet transformation such as the decomposition level and the threshold value have been searched and optimized in a wide range in the proposed technique. The Bayesian shrinkage method has been selected for thresholding because of its sub band dependence property. Some restoration results using the proposed method are given in this section. Fig. 9.77 has given the visual restoration results by applying the proposed algorithm on the *Ultrasound* image corrupted with low, medium and high densities of Gaussian and Poisson noises. Considering the three variations of noises the restored images still preserve the image fine details and textures very well.

9.6.2 Comparisons

Results obtained using the proposed GA based BayesShrink has been compared with three main existing filters in the wavelet domain. In table 9.62, some comparative restoration results have been given in terms of PSNR (dB) under the specified noise conditions. The table shows that the proposed filter performs significantly better than the existing filters. The corresponding graph for the table is given in Fig. 9.78.

The ultrasound image is corrupted with Gaussian noise with σ=50 and Poisson noises or Shot noise and then applied to the proposed algorithm. Figure 9.79 shows the visual restoration results by the VisuShrink, SureShrink, BayesShrink and the proposed operator. It is observed from the restored images that the proposed operator performs better than the existing operators.

Table 9.62: Comparisons of qualitative results in *PSNR (dB)* for *Ultrasound* image corrupted by Gaussian noise and Poisson noise or Shot noise

Filter	$\sigma = 30$	$\sigma = 60$	$\sigma = 90$
VisuShrink	27.31	23.54	21.69
SureShrink	28.00	24.78	23.11
BayesShrink	29.50	27.65	25.30
GA-BayesShrink	31.40	28.79	28.16

9.7 Discussions

The proposed techniques such as *Directional Weighted Minimum Deviation (DWMD)*, *All Neighbor Directional Weighted Pixels (ANDWP)*, *Variable Mask Median (VMM)*, *edge Preserving Restoration (EPR)*, *Sub-Image Restoration (SIR)*, *GA based Sub-Image Restoration (GASIR)*, *PSO based Sub-Image Restoration (PSOSIR)*, *Noise Estimation for Denoising (NDE)*, *Different length PSO (DPSO)* and *Fuzzy Switching Median Filter (FSMF)* are validated quantitatively and qualitatively in the spatial domain through exhaustive experimental results and comparisons. Salt and Pepper Noise (SPN) and Random Valued Noise (RVN) models are used to corrupt the images with noise density varies from minimum (10%) to maximum (100%). Standard images like *Lena*, *Cameraman*, *Liftingbody*, *Boat* and *Barbara* are used to test the performances of the proposed algorithms. The performance of the restoration process is quantified through peak signal to noise ratio (PSNR (dB)), structure similarity index measure (SSIM) and image enhancement factor (IEF). The proposed ten algorithms are compared with the state of the art operators such as SMF [Brownrigg 1984], AMF [Hwang 1995], BDND [Ng 2006], EDBA [Srinivasan 2007], IDBA [Nair 2008], FBDA [Nair 2012] and NASRI [Nasri 2013]. The execution time of all the algorithms considered for comparisons are studied and presented in the result section. Color *Lena* image of resolution 512×512 is also used to test the proposed algorithms.

The first proposed algorithm (DWMD) to the best algorithm (DPSO) of the thesis constitutes a consistent improvement in terms of the performance metrics. Among the existing algorithms, the FBDA and NASRI perform excellent, compared to the proposed algorithms.

Although the FSMF and DPSO perform the best compared to all existing algorithms. The execution time required for DPSO and NDE, are little high than the other algorithms, although they obtained best restoration results for all noise densities considered. The FSMF has obtained good PSNR values as well as it has affordable computation overhead. The first five proposed algorithms require three user parameters during the program run, and those parameters are searched in the GASIR and PSOSIR algorithms. Noise density estimation for denoising is performed for making an adaptive median filter. The FSMF operator has obtained excellent restoration results and less computational overhead. Although the DPSO algorithm outperforms all other algorithms and it has computational overhead larger the FSMF.

This section also validates the denoising algorithm for the medical images which are corrupted with Gaussian and Poisson noise or or Shot noise. Traditional BasyesShrink operator is modified and optimized in the proposed technique. In a specified range the Bayesian threshold is varied and the corresponding decomposition level of wavelet is searched using the Genetic algorithm. Genetic algorithm has been used very effectively to search the pair of wavelet parameters such as the optimal threshold and the value of decomposition level, since these two are the most important parameters of the wavelet denoising technique.

Considering the restoration results obtained from all algorithms considered for comparisons, the DPSO algorithm outperforms all other algorithms. The FSMF operator has obtained good PSNR, SSIM, UIQI and IEF values and it has very less computational overhead. Considering the color image outputs for both the noise models, the DPSO, NDE and FSMF outperform all the algorithms in terms of qualitative restoration results. The main objective of any image denoising algorithm depends on the performance of noise suppression capability and image fine detail preserving quality. Both the objectives are well established by the proposed algorithms as per the performance parameters such as PSNR, SSIM, UIQI and IEF values. Suppressing noises, optimizations of the algorithms and noise density estimation are done in the thesis quite efficiently.

Along with the proposed algorithm some other algorithms like VisuShrink, SureShrink, BayesShrink and the proposed GA based thresholding techniques have also been implemented. Results obtained demonstrate that the proposed method efficiently suppresses the Gaussian and Poisson noises with low, medium and high densities. Experimental results show that the proposed thresholding method based on the wavelet transform produces better restoration results in terms of PSNR (dB) and visual effects.

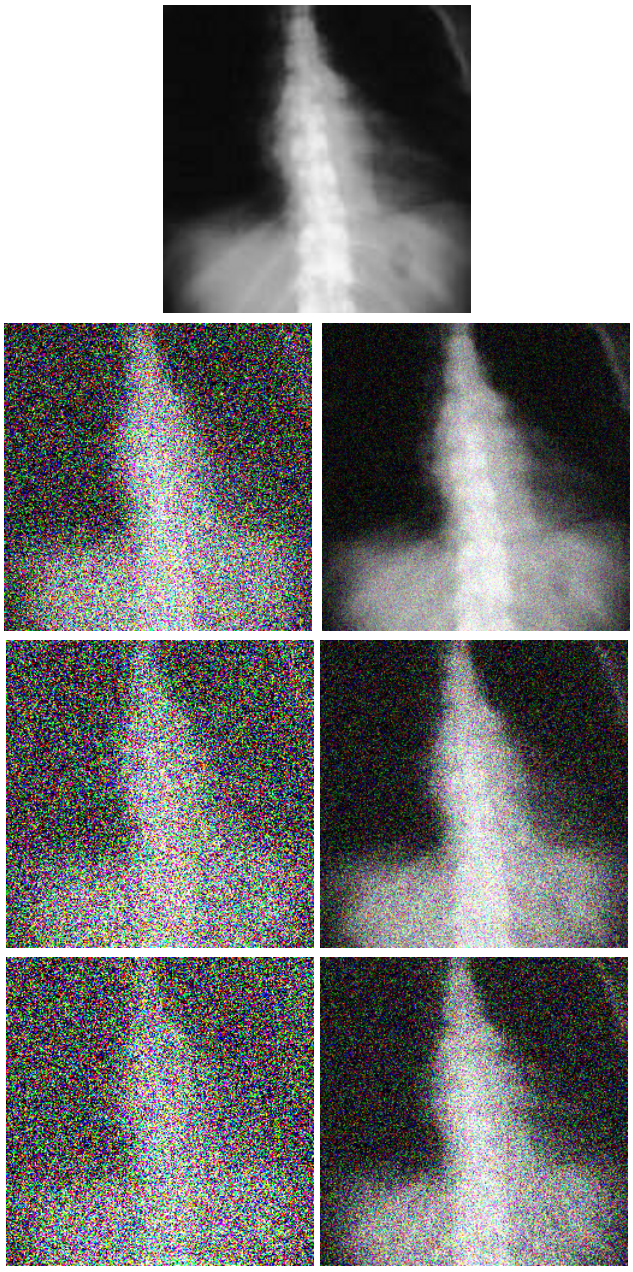


Figure 9.77: First row shows the original ultrasound image. Second row shows *noisy ultrasound image(σ=30)* and *restored image* from left to right, third row shows the *noisy ultrasound image(σ=60)* and *restored image* and fourth row show *noisy ultrasound image(σ=90)* and *restored image respectively.*

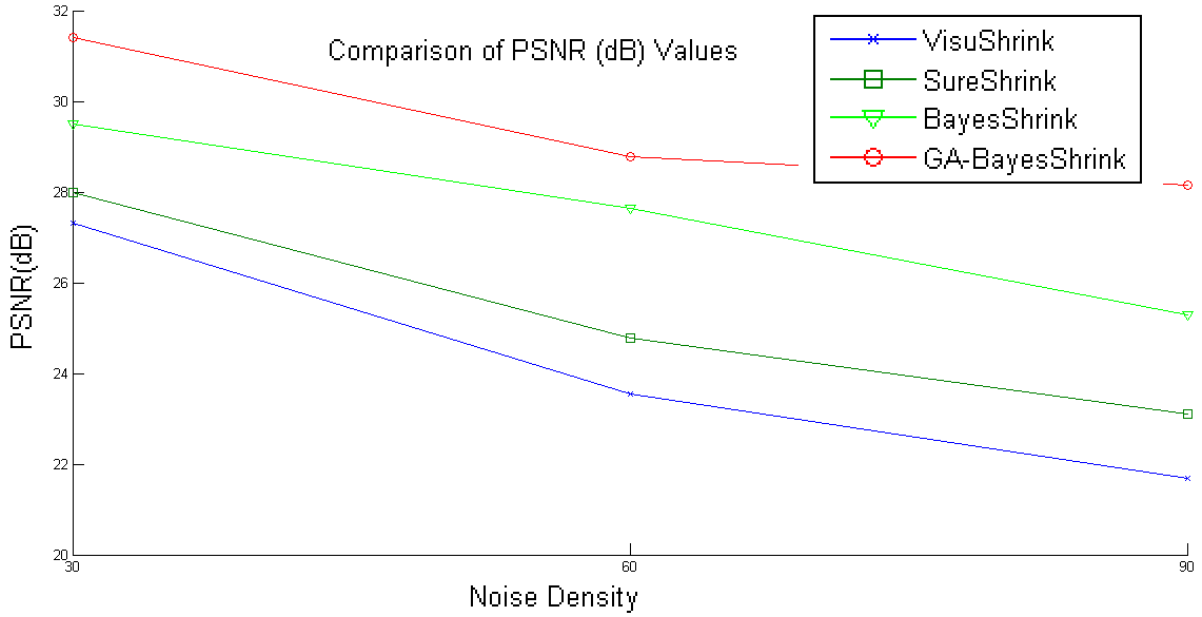
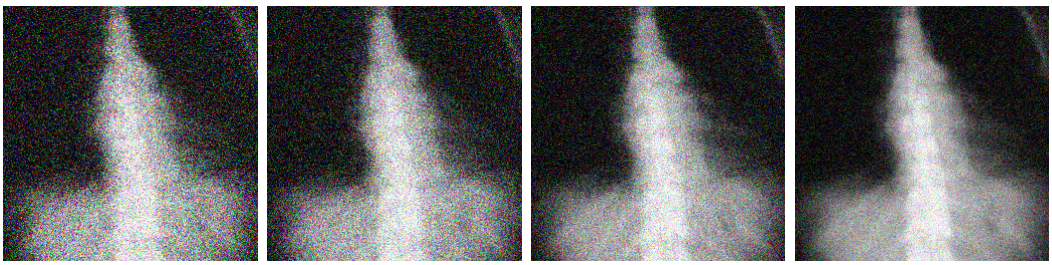


Figure 9.78: Graph representing comparison in *PSNR* for *Ultrasound* image corrupted by Gaussian noise and Poisson noise or Shot noise



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

Figure 9.79: a, b, c and d are restored images by VisuShrink, SureShrink, BayesShrink and the proposed shrinking method respectively while the noisy image has σ=50.

Part IV

Conclusive Discussions

Chapter 10

Conclusion

Contents

10.1 Limitations and Future Scope 252

The objective of the thesis was to design efficient techniques of digital and medical image denoising. The digital images are generally corrupted by salt-and-pepper noise (SPN) and random valued noises (RVN). Those two noise models were considered to train the standard digital images. The digital images have been filtered in spatial domain. In case of removal of Gaussian noise and Poisson noise, frequency domain filtering has been done.

Most of the recent methods proposed in the literature are suffering from several issues. The first one is selection of proper window size depending upon the noise density in the image during the noise detection and filtering operations. Second one is the poor performances of the existing methods for the images which are highly corrupted. Third one is the poor performances of the existing methods for the images which are corrupted by random valued noises. The existing methods also suffer from high computational cost of the algorithms when they provide good restoration results. There exist no such method which can deal with the SPN and RVN with good restoration results when the noise density varies from minimum (10%) to maximum (100%). Medical images are often corrupted due to the malfunctioning from of machineries in the clinical laboratories. For the medical images, most of the existing methods are based on Discrete Wavelet Transformation (DWT). In the wavelet domain, there should be a method which can deal with mixed Gaussian and Poisson noises in the same program run.

In this thesis, the first and foremost achievement is designing an efficient algorithm for *suppressing* the high density random valued noises in the digital images. A set of algorithms have been proposed in the thesis to deal with high density random valued noises in the digital images. The techniques outperform the existing methods in terms of noise suppression ability and computational overhead. Salient features of the techniques are that they can suppress salt and pepper noise in addition to random valued noise with variable noise densities in the digital images. Some algorithms are based on standard deviation operations

where as some are based on median operations of the neighboring pixels of the test pixel.

Optimizations of the filter is another achievement of the algorithm proposed to obtain the optimal restoration results. To make the noise suppressing algorithms automated, Genetic algorithm (GA) and Particle Swarm Optimization (PSO) based techniques are proposed to search the user parameters in a wide range to obtain the optimal restoration results.

A constant size window for all types of noise densities also degrades the restoration results in the denoising algorithms. From survey of algorithms for denoising it has been observed that most of the algorithms use hypothetical window to detect and restore the noisy pixel. The filtering window size must be compatible with the noise density of the image otherwise selection of pixels from the neighborhood would not be perfect and hence results will suffer. It may happen that a less noisy pixel is replaced by a noisier pixel by the restoration operation. The problem has been solved by an *estimation* of the noise density on the image and then depending upon the noise density, an empirical filtering window size is suggested.

Clustering algorithms on the image pixels have also been proposed for noise detection and filtering operations of the digital images. Fixed length PSO and Different length PSO based clustering algorithms have been proposed to classify the digital image by unsupervised way. Both the algorithms are able of suppressing salt-and pepper noise as well as random-valued-noise with good restoration results. General property of a noise-free image has been used to detect the noisy pixels which leads to a conclusion that the proposed technique may be treated as generalized noise removal approach for the digital images. The technique is also capable to deal with high density of impulses in the digital images. Three evaluation criteria are used to form the fitness function in the PSO-based clustering. The fitness function, and the *distance function* in PSO based clustering algorithm have been proposed in the thesis where the spatial distance as well as positional distance are computed to measure the similarity or dissimilarity of any pair of pixels during grouping of pixels. The DPSO algorithm performs the best in the thesis than any algorithm considered and proposed. Two objectives are achieved by this algorithm, novel image clustering algorithm and a novel denoising algorithm for SPN and RVN.

Fuzzy switching median filter has also been proposed for suppression of noise in the digital images. For noise detection purpose, instead of a single threshold, two threshold values are used in the proposed fuzzy membership (MF) function to measure the noise level and accordingly a filtering method has been applied to restore the corrupted pixel.

Although the spatial filters perform well on digital images but they have some constraints

regarding resolution degradation. These filters operate by smoothing over a fixed window and it produces artifacts around the object and sometimes causes over smoothing thus causes blurring of image. Wavelet transform is best suited for performance because of its properties like sparsity, multiresolution and multiscale nature. All frequency filters can also be implemented in the spatial domain and, if there exist a simple kernel for the desired filter effect, it is computationally less expensive to perform the filtering in the spatial domain. Frequency filtering is more appropriate if no straightforward kernel is found in the spatial domain, and may also be more efficient.

The technique for suppression of noises in the medical images corrupted by mixed Gaussian and Poisson noises is also proposed. The wavelet based algorithm for denoising medical images through thresholding and optimization using a recent stochastic and randomized search algorithm i.e., Genetic Algorithm (GA), has been proposed. A sub band adaptive thresholding algorithm has been proposed in the thesis to efficiently deal with the mixed Gaussian and Poisson noises.

To validate the proposed algorithms, extensive analyses of the results are done. A number of bench mark images have been corrupted to obtain the noisy images and then proposed algorithms are applied on these images for the purpose of testing of the algorithms. The result set is also made uniform that is, an homogeneous set of results have been made for each algorithm to compare them easily understandable. The proposed techniques are also applied into the color images to validate the proposed techniques of the thesis work. The performances of the proposed filters in the thesis work have been evaluated quantitatively and qualitatively through simulation results and analysis. The results obtained are compared with the existing filters against the noise models. Various graphical representations of the results using different attributes are also given to substantiate the performance of the proposed technique. Bench mark images are trained using low (10%) to extremely high noise density (100%) using all noise models. Various performance metrics are used to find the algorithmic performances of the proposed algorithms in the thesis work. Different statistical measures are used in the thesis work to validate the proposed algorithms by comparing it with most of the state of the art methods in the literature. Some measures of image quality such as *Peak Signal to Noise Ratio (PSNR (dB))*, *Image Enhancement Factor (IEF)*, Structure Similarity Index Measure (SSIM)., etc are used.

Digital image processing has very wide applications and almost all of the technical fields are impacted by that. Now a days there is no area of technical field where digital image processing is not used. One of the ways to develop the extension of image processing applications is according to their sources. The source for images in use today are the electromagnetic energy spectrum, acoustic, ultrasound and electronic. The types of images includes satellite

image, underwater image, robot vision image, medical images like medical resonance image (MRI), ultrasound image, Computed Tomography (CT) image etc,. Some of the major fields in which digital image processing is widely used are: image sharpening and restoration, medical field, remote sensing, geographical information system, geographical positioning system, transmission and encoding, machine or robot vision, color processing, pattern recognition, video processing, microscopic imaging and others. In all of these areas the captured images suffer from integration of noises. The noises need to be processed for better results in the subsequent stages of the image processing.

Image denoising is the primary preprocessing to all most all image analysts because the subsequent tasks such as object detection, shape digitization, edge detection, detecting invisible boundaries in the image are truly deviated by noises in the images. It is important because the human vision system is limited. Image denoising must be done to get clearer vision of the image. Image denoising may be used to make explicit information that is only implicit in the data. To discover new knowledge from the images, that should be filtered first.

Some of the utility of the proposed techniques are: to use information of the noise density in the noisy images, another is to use the image clustering information. The two techniques proposed in the thesis work can be applied to other tasks also. The estimation of noises in the images is properly done in the thesis. One can use that algorithm for other type of areas also. The novel image clustering algorithm proposed in the thesis work can be applied for image segmentations. The optimization algorithms developed in his thesis work can also be applied for other type of optimization problems.

10.1 Limitations and Future Scope

Noise suppression, optimization and noise density estimation on the digital images have been covered during the course of research. There is a scope of research to devise noise suppression algorithms in the medical images. Salt-and-Pepper noise and Random-Valued-noise in the digital images and Gaussian and Poisson noises in the medical images are suppressed successfully. The other type of noise such as *Speckle noise* can be suppressed in the images. The other type of images such as Satellite images, X-ray images, computed tomography (CT) images, magnetic resonance imaging (MRI) images, nuclear medicine imaging, brain images and SAR images can be denoised. The Electrocardiogram (ECG) signal or heart sound and Electroencephalography (EEG) signal can also be investigated for denoising for better clinical diagnosis by the doctors. The frequency domain can be more utilized in the medical image denoising process. Another perspective is that new statistical measures can be developed to test the algorithmic performances and quality of the restored images.

All these results and promising perspectives of research works confirm that statistical tools based denoising, noise estimation for denoising, denoising with optimization, image clustering for denoising, fuzzy switching median filtering and wavelet domain based denoising are suitable paradigms for noise estimation, suppression and optimization of impulses in digital and medical imagery that may be used for real applications integrating imaging knowledge for improving the image denoising process.

Part V

Bibliography

Bibliography

[Aizenberg 2005] I. Aizenberg, C. Butakoff and D. Paliy. *Impulsive noise removal using threshold Boolean filtering based on the impulse detecting functions.* IEEE Signal Processing Letters, vol. 12, no. 1, pages 63–66, Jan 2005. (Cited on page 29.)

[Alajlan 2004] Naif Alajlan, Mohamed Kamel and Ed Jernigan. *Detail preserving impulsive noise removal.* Signal Processing: Image Communication, vol. 19, no. 10, pages 993 – 1003, 2004. (Cited on pages 25 and 29.)

[Alajlan 2010] Naif Alajlan. *A Novel Recursive Algorithm For Detail-Preserving Impulse Noise Removal.* Journal of King Saud University - Computer and Information Sciences, vol. 22, no. 0, pages 37 – 44, 2010. (Cited on page 47.)

[Amudha 2012] J. Amudha, N. Pradeepa and R. Sudhakar. *A Survey on Digital Image Restoration.* Procedia Engineering, vol. 38, no. 0, pages 2378 – 2382, 2012. (Cited on page 53.)

[Anitha 2012] S. Anitha and V. Radha. *Enhanced Switching Median Filter for Denoising in 2D Patterned Textile Images.* Procedia Engineering, vol. 38, no. 0, pages 3362 – 3372, 2012. (Cited on page 51.)

[Antoniadis 2007] A Antoniadis. *Wavelet methods in statistics: some recent developments and their applications.* Stat Surv, vol. 1, pages 16–55, 2007. (Cited on page 9.)

[Aravind 2010] B. N. Aravind and K. V. Suresh. *Wavelet Based Image Denoising With Optimal Thresholding.* Image Processing (NCIMP 2010), pages 297 – 299, 2010. (Cited on page 47.)

[Awad 2011] A.S. Awad. *Standard Deviation for Obtaining the Optimal Direction in the Removal of Impulse Noise.* IEEE Signal Processing Letters, vol. 18, no. 7, pages 407–410, July 2011. (Cited on page 34.)

[Bacchelli 2006] Silvia Bacchelli and Serena Papi. *Image denoising using principal component analysis in the wavelet domain.* Journal of Computational and Applied Mathematics, vol. 189, no. 12, pages 606 – 621, 2006. Proceedings of The 11th International Congress on Computational and Applied Mathematics The 11th International Congress on Computational and Applied Mathematics. (Cited on page 44.)

[Bandyopadhyay 2001] S. Bandyopadhyay and S. K. Pal. *Pixel Classification Using Variable String Genetic Algorithms with Chromosome Differentiation*. IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 2, pages 303–308, 2001. (Cited on page 107.)

[Beaurepaire 1997] L. Beaurepaire, K. Chehdi and B. Vozel. *Identification of the nature of noise and estimation of its statistical parameters by analysis of local histograms*. In IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997, volume 4, pages 2805–2808 vol.4, Apr 1997. (Cited on page 24.)

[Besdok 2004] Erkan Besdok, P Civicioglu and Mustafa Alci. *Impulsive Noise Suppression from Highly Corrupted Images by Using Resilient Neural Networks*. In Artificial Intelligence and Soft Computing - ICAISC 2004, volume 3070 of *Lecture Notes in Computer Science*, pages 670–675. Springer Berlin Heidelberg, 2004. (Cited on page 37.)

[Besdok 2005] Erkan Besdok, P Civicioglu and Mustafa Alci. *Using an adaptive neuro-fuzzy inference system-based interpolant for impulsive noise suppression from highly distorted images*. Fuzzy Sets and Systems, vol. 150, no. 3, pages 525 – 543, 2005. (Cited on page 37.)

[Binh 2010] NguyenThanh Binh and Ashish Khare. *Adaptive Complex Wavelet Technique for Medical Image Denoising*. In Vo Toi and TruongQuangDang Khoa, editeurs, The Third International Conference on the Development of Biomedical Engineering in Vietnam, volume 27 of *IFMBE Proceedings*, pages 196–199. Springer Berlin Heidelberg, 2010. (Cited on pages 47 and 50.)

[Biswas 2012] Mantosh Biswas and Hari Om. *A New Soft-Thresholding Image Denoising Method*. Procedia Technology, vol. 6, no. 0, pages 10 – 15, 2012. 2nd International Conference on Communication, Computing and amp;amp; Security [ICCCS-2012]. (Cited on page 52.)

[Borisov 2008] B.D. Borisov. *Optimal filtration of a signal against flicker noises*. Optoelectronics, Instrumentation and Data Processing, vol. 44, no. 4, pages 317–324, 2008. (Cited on page 31.)

[Brownrigg 1984] D. R. K. Brownrigg. *The Weighted Median Filter*. Communications of the ACM, vol. 27, no. 8, pages 807–818, August 1984. (Cited on pages 13, 17, 22, 55, 79, 112, 136, 149 and 251.)

[Bruni 2008] Vittoria Bruni and Domenico Vitulano. *Image Denoising Using Similarities in the Time-Scale Plane*. In Proceedings of the 10th International Conference on

Advanced Concepts for Intelligent Vision Systems, ACIVS '08, pages 368 – 379, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 46.)

[Butakoff 2004] C. Butakoff and I. Aizenberg. *Effective Impulse Detector Based on Rank-Order Criteria*. IEEE Signal Processing Letters, vol. 11, no. 3, pages 363–366, March 2004. (Cited on page 28.)

[Cai 2001] T.T. Cai and B.W.Silverman. *Incorporating information on neighbouring coefficients into wavelet estimation*. Sankhya: The indian journal of statistics, vol. 63(b), no. 2, pages 127–148, 2001. (Cited on page 40.)

[Chan 2004] R.H. Chan, Chen Hu and M. Nikolova. *An iterative procedure for removing random-valued impulse noise*. IEEE Signal Processing Letters, vol. 11, no. 12, pages 921–924, Dec 2004. (Cited on page 28.)

[Chanda 2002] B. Chanda and D. Dutta Majumder. Digital image processing and analysis. Prentice-Hall of India, 1 édition, 2002. (Cited on page 3.)

[Chang 2000a] S. Chang, B Yu and M. Vetterli. *Spatially adaptive wavelet thresholding based on context modeling for image denoising*. IEEE Transactions on Image Processing, vol. 9, no. 9, pages 1522–1531, 2000. (Cited on page 6.)

[Chang 2000b] S.G. Chang, Bin Yu and M. Vetterli. *Spatially adaptive wavelet thresholding with context modeling for image denoising*. IEEE Transactions on Image Processing, vol. 9, no. 9, pages 1522–1531, Sep 2000. (Cited on page 43.)

[Chen 1999] T. Chen, K.K. Ma and L.h. Chen. *Tri- state median filter for image de noising*. IEEE Transaction Image Processing, vol. 8, no. 12, pages 1834–1838, December 1999. (Cited on page 24.)

[Chen 2001a] T. Chen and H.R. Wu. *Application of partition based median type filters for suppressing noise in images*. IEEE Transactions Image Processing, vol. 10, no. 6, pages 829–836, June 2001. (Cited on pages 24 and 25.)

[Chen 2001b] Tao Chen and Hong Ren Wu. *Adaptive Impulse Detection Using Center Weighted Median Filters*. IEEE Signal Processing Letters, vol. 8, no. 1, pages 1–3, January 2001. (Cited on page 25.)

[Chen 2001c] Tao Chen and Hong Ren Wu. *Application of partition-based median type filters for suppressing noise in images*. IEEE Transactions on Image Processing, vol. 10, no. 6, pages 829–836, Jun 2001. (Cited on page 26.)

[Chen 2001d] Tao Chen and Hong Ren Wu. *Space variant Median Filters for the Restoration of Impulse noise Corrupted images*. IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 48, no. 8, pages 784–789, August 2001. (Cited on page 25.)

[Chen 2003] G.Y. Chen and T.D. Bui. *Multiwavelet denoising using neighbouring coefficients*. IEEE Signal Processing Letters, vol. 10, pages 211–214, 2003. (Cited on page 41.)

[Chen 2005] G.Y. Chen, T.D. Bui and A.Krzyzak. *Image denoising using neighbouring wavelet coefficients*. Integrated Computer-Aided Engineering, vol. 12, pages 99–107, 2005. (Cited on page 43.)

[Chen 2008] Guangyi Chen and Wei-Ping Zhu. *Image Denoising Using Three Scales of Wavelet Coefficients*. In Fuchun Sun, Jianwei Zhang, Ying Tan, Jinde Cao and Wen Yu, editeurs, Advances in Neural Networks - ISNN 2008, volume 5264 of *Lecture Notes in Computer Science*, pages 376–383. Springer Berlin Heidelberg, 2008. (Cited on page 46.)

[Chen 2012] G. Chen, W.-P. Zhu and W. Xie. *Wavelet-based image denoising using three scales of dependency*. Image Processing, IET, vol. 6, no. 6, pages 756–760, August 2012. (Cited on page 52.)

[Cheng 2004] Hui Cheng, Qiuze Yu, Jinwen Tian and Jian Liu. *Image denoising using wavelet and support vector regression*. In IEEE First Symposium on Multi-Agent Security and Survivability, 2004, pages 43–46, Dec 2004. (Cited on page 42.)

[Cherkassky 2001] Vladimir Cherkassky and Steven Kilts. *Comparison of Wavelet Thresholding Methods for Denoising ECG Signals*. In Georg Dorffner, Horst Bischof and Kurt Hornik, editeurs, Artificial Neural Networks ICANN 2001, volume 2130 of *Lecture Notes in Computer Science*, pages 625–629. Springer Berlin Heidelberg, 2001. (Cited on page 41.)

[Cho 2006] Ung-Keun Cho, Jin-Hyuk Hong and Sung-Bae Cho. *Evolutionary Image Enhancement for Impulsive Noise Reduction*. In De-Shuang Huang, Kang Li and George William Irwin, editeurs, Intelligent Computing, chapitre Lecture Notes in Computer Science, pages 678–683. Springer Berlin Heidelberg, 2006. (Cited on page 30.)

[Choi 1997] YoungSik Choi and R. Krishnapuram. *A robust approach to image enhancement based on fuzzy logic*. IEEE Transactions on Image Processing, vol. 6, no. 6, pages 808–825, Jun 1997. (Cited on page 37.)

[Chou 2011] Chien-ming Chou. *A Threshold Based Wavelet Denoising Method for Hydrological Data Modelling*. Water Resources Management, vol. 25, no. 7, pages 1809–1830, 2011. (Cited on page 48.)

[Chui 2012] Mingwei Chui, Youqian Feng, Wei Wang, Zhengchao Li and Xiaodong Xu. *Image Denoising Method with Adaptive Bayes Threshold in Nonsubsampled Contourlet Domain*. AASRI Procedia, vol. 1, no. 0, pages 512 – 518, 2012. AASRI Conference on Computational Intelligence and Bioinformatics. (Cited on page 51.)

[Coifman 1994] R.R. Coifman and D.L Donoho. *Wavelet Invariant denoising in Wavelets and statistics*. Springer Lecture Notes in Statics, Springer, New York, vol. 103, 1994. (Cited on page 141.)

[Crank 1947] J. Crank and P. Nicolson. *A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type*. Mathematical Proceedings of the Cambridge Philosophical Society, vol. 43, pages 50 – 67, 1 1947. (Cited on page 50.)

[Crnojevic 2004] Vladimir Crnojevic, Vojin Senk and Zeljin Trpovski. *Advanced impulse Detection based on Pixel- Wise MAD*. IEEE Signal Processing Letters, vol. 11, no. 7, pages 589–592, July 2004. (Cited on pages 24, 27 and 28.)

[Crouse 1998] M.S. Crouse, R.D. Nowak and R.G. Baraniuk. *Wavelet based signal processing using Hidden Markov Models*. IEEE Transactions on Signal Processing, vol. 46, pages 886–902, 1998. (Cited on page 40.)

[de Oliveira Mota 2011] Hilton de Oliveira Mota, Leonardo Chaves Dutra da Rocha, Thiago Cunha de Moura Salles and Flvio Henrique Vasconcelos. *Partial discharge signal denoising with spatially adaptive wavelet thresholding and support vector machines*. Electric Power Systems Research, vol. 81, no. 2, pages 644 – 659, 2011. (Cited on page 49.)

[D.E. 1989] Goldberg D.E. *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison- Wesley, 1989. (Cited on pages 93, 97 and 143.)

[Deng 2009] Chao Deng and Ji Yu An. *An Impulse Noise Removal Based on a Wavelet Neural Network*. In Information and Computing Science, 2009. ICIC '09. Second International Conference on, volume 2, pages 71–74, May 2009. (Cited on page 46.)

[Deng 2010] Chao Deng, Hong-Min Liu and Zhi-Heng Wang. *Applying an improved neural network to impulse noise removal*. In Wavelet Analysis and Pattern Recognition

(ICWAPR), 2010 International Conference on, pages 207–210, July 2010. (Cited on page 33.)

[Dong 2007] Yiqiu Dong and Shufang XU. *A New Directional Weighted Median Filter for Removal of Random Valued Impulse Noise*. IEEE Signal Processing Letters, vol. 14, no. 3, pages 193–196, March 2007. (Cited on page 30.)

[Donoho 1994] David L. Donoho and Iain M. Johnstone. *Ideal spatial adaptation by wavelet shrinkage*. Biometrika, vol. 81, no. 3, pages 425 – 455, 1994. (Cited on pages 6, 7, 8, 9, 39, 43, 52 and 250.)

[Donoho 1995a] David L. Donoho. *De-noising by soft-thresholding*. IEEE Transactions on Information Theory,, vol. 41, no. 3, pages 613 – 627, May 1995. (Cited on pages 6, 9 and 39.)

[Donoho 1995b] David L. Donoho and Iain M. Johnstone. *Adapting to unknown smoothness via wavelet shrinkage*. J Am Stat Assoc, vol. 90, no. 432, pages 1200–1224, 1995. (Cited on pages 6, 7, 8, 9, 39 and 250.)

[Donoho 1995c] David L. Donoho and Iain M. Johnstone. *Minimax estimation via wavelet shrinkage*. Ann Stat, vol. 43, no. 3, pages 879–921, 1995. (Cited on pages 6, 9 and 39.)

[Donoho 1995d] David L. Donoho, Iain M. Johnstone, Gerard Kerkyacharian and Dominique Picard. *Wavelet shrinkage: asymptopia*. Journal of the Royal Statistical Society, Ser. B, pages 371 – 394, 1995. (Cited on page 39.)

[Donoho 2004] D. L Donoho and M.E Raimondo. *A fast wavelet algorithm for image deblurring*. IEEE Transactions on Image Processing, vol. 16, no. 2, 2004. (Cited on pages 42 and 48.)

[Downie 1998] T.R Downie and B.W.Silverman. *The discrete multiple wavelet transform and thresholding methods*. IEEE Transactions on Signal Processing, vol. 46, pages 2558–2561, 1998. (Cited on page 141.)

[Duits 2003] Remco Duits, Michael Felsberg, Luc Florack and Bram Platel. *Scale Spaces on a Bounded Domain*. In Proceedings of the 4th International Conference on Scale Space Methods in Computer Vision, Scale Space'03, pages 494 – 510, Berlin, Heidelberg, 2003. Springer-Verlag. (Cited on page 47.)

[E. Abreu 1996] S. K. Mitra E. Abreu M. Lightstone and K. Arakawa. *A new efficient approach for the removal of impulse noise from highly corrupted images*. IEEE Transactions on Image Processing, vol. 5, no. 6, pages 1012–1025, June 1996. (Cited on page 23.)

[Eberhart 1995] R. Eberhart and J. Kennedy. *A new optimizer using particle swarm theory.* In Micro Machine and Human Science, 1995. MHS, Proceedings of the Sixth International Symposium on, pages 39 – 43. IEEE, October 1995. (Cited on page 100.)

[Emin Yüksel 2004] M. Emin Yüksel and E. Besdok. *A simple neuro-fuzzy impulse detector for efficient blur reduction of impulse noise removal operators for digital images.* IEEE Transactions on Fuzzy Systems, vol. 12, no. 6, pages 854–865, Dec 2004. (Cited on page 24.)

[Eslami 2006] Ramin Eslami and Hayder Radha. *Translation-invariant contourlet transform and its application to image denoising.* IEEE Transactions on Image Processing, vol. 15, no. 11, pages 3362 – 3374, 2006. (Cited on page 44.)

[Esmin 2008] A. A. A. Esmin, D. L. Pereira and F. P. A. de Arajo. *Study of different approach to clustering data by using particle swarm optimization algorithm.* Proceedings of the IEEE World Congress on Evolutionary Computation (CEC 2008), Hong Kong, China, pages 1817–1822, June 2008. (Cited on page 122.)

[Everitt 1993] B. S. Everitt. *Cluster Analysis.* Third Edition, Halsted press, 1993. (Cited on page 107.)

[FDM] *Fuzzy Decision mapping Support System.* http://proceedings.esri.com/library /user-conf /proc99 /proceed /papers /pap480 /p480.htm (accessed on 1st July, 2012). (Cited on page 131.)

[Feng 2006] Deng-Chao Feng, Zhao-Xuan Yang and Xiao-Jun Qiao. *The Application of Wavelet Neural Network with Orthonormal Bases in Digital Image Denoising.* In Jun Wang, Zhang Yi, JacekM. Zurada, Bao-Liang Lu and Hujun Yin, editeurs, Advances in Neural Networks - ISNN 2006, volume 3972 of *Lecture Notes in Computer Science*, pages 539–544. Springer Berlin Heidelberg, 2006. (Cited on page 44.)

[Fodor 2001] Imola K. Fodor and Chandrika Kamath. *Denoising through wavlet shrinkage: An empirical study.* Rapport technique, Center for applied computing Lawrence Livermore National Laboratory, July 2001. (Cited on page 51.)

[Ghazel 2005] M. Ghazel, G.H. Freeman, E.R. Vrscay and R.K. Ward. *Wavelet Image Denoising Using Localized Thresholding Operators.* In Mohamed Kamel and Aurlio Campilho, editeurs, Image Analysis and Recognition, volume 3656 of *Lecture Notes in Computer Science*, pages 149–158. Springer Berlin Heidelberg, 2005. (Cited on page 42.)

[Gonzalez 1992] R. C. Gonzalez and R. E. Woods. Digital image processing. Addison Wesley, 2 édition, 1992. (Cited on pages 3, 19 and 22.)

[Gose 1996] Earl Gose, Richard Johnsonbaugh and Steve Jost. Pattern recognition and image analysis. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996. (Cited on page 107.)

[Guo 2009] Xuan Guo, Baoping Guo, Tao Hu and Ou Yang. *An Adaptive Hybrid Filtering for Removing Impulse Noise in Color Images*. In Wen Yu, Haibo He and Nian Zhang, editeurs, Advances in Neural Networks, volume 5553 of *Lecture Notes in Computer Science*, pages 494–501. Springer Berlin Heidelberg, 2009. (Cited on page 32.)

[Guo 2011] Dongfeng Guo and Jianbiao Chen. *The Application of Contourlet Transform to Image Denoising*. Procedia Engineering, vol. 15, no. 0, pages 2333 – 2338, 2011. CEIS 2011. (Cited on page 49.)

[Han 1997] W. Y. Han and J. C. Lin. *Minimum-Maximum Exclusive Mean (MMEM) Filter to Remove Impulse Noise from Highly Corrupted Images*. Electronics Letters, vol. 32, no. 2, pages 124–125, January 1997. (Cited on page 24.)

[Harish 2005] Er. Harish, Er. Monika Verma and Er. Aashiima. *Filter for removal of impulse noise by using fuzzy logic*. International Journal Of Image Processing (IJIP), vol. 3, no. 5, pages 195 – 202, 2005. (Cited on page 29.)

[H.Chipman 1997] H.Chipman, E. Kolaczyk and R. McCulloch. *Adaptive Bayesian wavelet shrinkage*. J Am Stat Assoc, vol. 440, no. 92, pages 1413–1421, 1997. (Cited on pages 7, 8, 43, 51 and 250.)

[He 2011] Ming He, Hao Wang, Hong fu Xie, Ke jun Yang and Qing-Wei Gao. *Anisotropic Wavelet-Based Image Denoising Using Multiscale Products*. Procedia Engineering, vol. 15, no. 0, pages 2793 – 2797, 2011. (Cited on page 49.)

[Holland 1992] John H. Holland. Adaptation in natural and artificial systems. MIT Press, Cambridge, MA, USA, 1992. (Cited on page 93.)

[Hua 1997] Zhong Hua and Man-Nang Chong. *A wavelet-denoising approach for removing background noise in medical images*. In Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on, pages 980–983 vol.2, Sep 1997. (Cited on page 40.)

[Hubber 1981] P. Hubber. Robust statistics. Wiley, New York, 1981. (Cited on page 28.)

[Hwang 1995] H. Hwang and R. Haddad. *Adaptive median filters: new algorithms and results*. IEEE Transactions on Image Processing, vol. 4, no. 4, pages 499–502, Apr 1995. (Cited on pages 149 and 251.)

[Ima] *Defining Fuzzy Sets*. http://www.calvin.edu/ pribeiro /othrlnks/Fuzzy/fuzzysets.htm (accessed on 1st July, 2012). (Cited on pages xxxi and 138.)

[J. 1995] Kennedy. J. and R. C Eberhart. *Particle swarm optimization*. IEEE International Conference on Neural Network, Perth, Australia, pages 1942–1948, 1995. (Cited on pages 100 and 103.)

[Jain 1988] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Enlewood Cliffs, NJ, 1988. (Cited on page 107.)

[Jain 1999] A. K. Jain, M. N. Murty and P. J. Flynn. *Data Clustering: a Review*. ACM Computer Survey, vol. 31, no. 3, pages 264–323, September 1999. (Cited on page 107.)

[Jaiswal 2014] Ayushi Jaiswal, Jayprakash Upadhyay and Ajay Somkuwar. *Image denoising and quality measurements by using filtering and wavelet based techniques*. AEU - International Journal of Electronics and Communications, no. 0, 2014. (Cited on page 54.)

[Jansen 2001] M Jansen. *Noise Reduction by wavelet thresholding*. Springer-Verlag, New York, USA, 2001. (Cited on page 9.)

[Javier 2010] Lapez Javier, Lanzarini Laura and Giusti Armando. *VarMOPSO: Multi-Objective Particle Swarm Optimization with Variable Population Size*. In Advances in Artificial Intelligence IBERAMIA 2010, volume 6433 of *Lecture Notes in Computer Science*, pages 60–69. Springer Berlin Heidelberg, 2010. (Cited on page 124.)

[J.Weaver 1991] J.Weaver, X.Yansun and DHL Cromwell. *Filtering noise from images with wavelet transforms*. Magn Reson Med, vol. 21, no. 2, pages 288–295, 1991. (Cited on page 9.)

[Katari 2007] Venkatesh Katari, Suresh Ch, Ra Satapathy, Member IEEE, Jvr Murthy and Pvgd Prasad Reddy. *Hybridized Improved Genetic Algorithm with Variable Length Chromosome for Image Clustering Abstract*. International Journal of Computer Science and Network Security, vol. 7, no. 11, pages 1121–131, November 2007. (Cited on page 124.)

[Kavitha 2010] A.R. Kavitha and C. Chellamuthu. *Implementation of gray-level clustering algorithm for image segmentation*. Procedia Computer Science, vol. 2, no. 0, pages

314 – 320, 2010. Proceedings of the International Conference and Exhibition on Biometrics Technology. (Cited on page 118.)

[Ke 2011] Cao Ke, Wang Feng and Shang Qing-wei. *A Novel Image Denoising Algorithm Based on Crank-Nicholson Semi-Implicit Difference Scheme*. Procedia Engineering, vol. 23, no. 0, pages 647 – 652, 2011. (Cited on page 50.)

[Khriji 1998] L. Khriji and M. Gabbouj. *Median-Rational Hybrid Filters*. In Proceedings of International Conference on Image Processing, vol. 2, pages 853–857, October 1998. (Cited on page 24.)

[Kim 2004] Min Sik Kim, Taekhyun Kim, Yong june Shin, Simon S. Lam and Edward J. Powers. *A wavelet-based approach to detect shared congestion*. In in Proc. ACM SIGCOMM, pages 293 – 305, 2004. (Cited on page 42.)

[Kingsbury 1998a] A. Kingsbury. *The dual-tree complex wavelet transform: a new efficient tool for image restoration and enhancement*. pages 319 – 322, 1998. In Proc. EUSIPCO 98. Rhodes: EURASIP. (Cited on pages 50 and 51.)

[Kingsbury 1998b] Nick Kingsbury. *The Dual-Tree Complex Wavelet Transform: A New Technique For Shift Invariance And Directional Filters*. pages 319 – 322, 1998. (Cited on pages 50 and 51.)

[Kingsbury 1999] N. Kingsbury. *Image processing with complex wavelets*. Phil. Trans.Roy. Soc. London, Ser. A, vol. 357, no. 1760, pages 2543–2560, September 1999. (Cited on page 40.)

[Kingsbury 2001] Nick Kingsbury. *Complex Wavelets for Shift Invariant Analysis and Filtering of Signals*. Applied and Computational Harmonic Analysis, vol. 10, no. 3, pages 234 – 253, 2001. (Cited on pages 50 and 51.)

[Kondo 2002] K. Kondo, M. Haseyama and H. Kitajima. *An Accurate Noise Detector for Image Restoration*. In Proceedings of International Conference on Image Processing, vol. 1, pages 321–324, September 2002. (Cited on page 27.)

[Kong 1996] H. Kong and L. Guan. *A neural network adaptive filter for the removal of impulse noise in digital images*. Neural Networks, vol. 9, no. 3, pages 373 – 378, 1996. (Cited on page 24.)

[Krommweh 2009] Jens Krommweh and Gerlind Plonka. *Directional Haar wavelet frames on triangles*. Applied and Computational Harmonic Analysis, vol. 27, no. 2, pages 215 – 234, 2009. (Cited on page 46.)

[L.B. 2009] Miranda Andre L.B., Garcia LuAsPauloF, Carvalho AndrAC.P.L.F. and AnaC. Lorena. *Use of Classification Algorithms in Noise Detection and Elimination.* In Hybrid Artificial Intelligence Systems, volume 5572 of *Lecture Notes in Computer Science*, pages 417–424. Springer Berlin Heidelberg, 2009. (Cited on page 118.)

[Li 2007] J. Li, S.S. Mohamed, M.M.A. Salama and G.H. Freeman. *Subband-Adaptive and Spatially-Adaptive Wavelet Thresholding for Denoising and Feature Preservation of Texture Images.* In Mohamed Kamel and Aurlio Campilho, editeurs, Image Analysis and Recognition, volume 4633 of *Lecture Notes in Computer Science*, pages 24–37. Springer Berlin Heidelberg, 2007. (Cited on page 45.)

[Li 2011a] Junfeng Li and Wenzhan Daia. *Adaptive Image Denoising Algorithm Based on Correlativity.* Procedia Engineering, vol. 23, no. 0, pages 345 – 350, 2011. (Cited on page 50.)

[Li 2011b] Minghui Li, Zhenhong Jia, Jie Yang, Yingjie Hu and Dianjun Li. *An Algorithm for Remote Sensing Image Denoising Based on the Combination of the Improved BiShrink and DTCWT.* Procedia Engineering, vol. 24, no. 0, pages 470 – 474, 2011. International Conference on Advances in Engineering 2011. (Cited on page 50.)

[Lin 2003] En-Bing Lin and Yi Ling. *Image compression and denoising via nonseparable wavelet approximation.* Journal of Computational and Applied Mathematics, vol. 155, no. 1, pages 131 – 152, 2003. Approximation Theory, Wavelets, and Numerical Analysis. (Cited on page 41.)

[Lin 2004] Tzu-Chao Lin and Pao-Ta Yu. *Partition fuzzy median filter based on fuzzy rules for image restoration.* Fuzzy Sets and Systems, vol. 147, no. 1, pages 75 – 97, 2004. Hybrid Methods for Adaptive Systems. (Cited on page 29.)

[Liu 2012] Feng Liu, Yutai Wang and Yanxiang Wang. *Research and Implementation of Heart Sound Denoising.* Physics Procedia, vol. 25, no. 0, pages 777 – 785, 2012. International Conference on Solid State Devices and Materials Science, April 1-2, 2012, Macao. (Cited on page 53.)

[Ma 2003] Zhonghua Ma and Hong Ren Wu. *A histogram based adaptive vector filter for color image restoration.* In Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on, volume 1, pages 81–85, Dec 2003. (Cited on page 24.)

[Ma 2006] Qi-ming Ma, Xuan-yin Wang and Shuan-ping Du. *Method and application of wavelet shrinkage denoising based on genetic algorithm.* Journal of Zhejiang University SCIENCE A, vol. 7, no. 3, pages 361–367, 2006. (Cited on page 43.)

[Malfati 1997] M. Malfati and D. Roose. *Wavelet-based image denoising using a Markov random field a priori model.* IEEE Transactions on Image Processing, vol. 6, no. 4, pages 549–565, 1997. (Cited on page 141.)

[Mallat 1989] S. G. Mallat. *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 11, no. 7, pages 674 – 693, July 1989. (Cited on page 39.)

[Mallat 2009] Stephane Mallat. A wavelet tour of signal processing: The sparse way. Academic Press, 3 édition, 2009. (Cited on page 39.)

[Mandal 2010] J. K. Mandal and Aparna Sarkar. *A novel modified directional weighted median based filter for removal of random valued impulse noise.* International Symposium on Electronic System Design, pages 230–234, December 2010. (Cited on page 33.)

[Mandal 2011] J. K. Mandal and Aparna Sarkar. *A modified weighted based filter for removal of random impulse noise.* Second International Conference on Emerging Applications of Information Technology, pages 173–176, February 2011. (Cited on page 34.)

[Maulik 2003] U. Maulik and S. Bandyopadhyay. *Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification.* IEEE Transactions on Geoscience and Remote Sensing, vol. 41, no. 5, pages 1075–1081, May 2003. (Cited on pages 107 and 124.)

[Mehmet 2012] Mehmet, Muammer, Abdulkadir and Fikret Ata. *Denoising of weak ECG signals by using wavelet analysis and fuzzy thresholding.* Network Modeling Analysis in Health Informatics and Bioinformatics, vol. 1, no. 4, pages 135–140, 2012. (Cited on page 53.)

[Min 2011] Tae-Hong Min, Rae-Hong Park and SoonKeun Chang. *Noise reduction in high dynamic range images.* Signal, Image and Video Processing, vol. 5, no. 3, pages 315–328, 2011. (Cited on page 34.)

[Moore 1999] Michael S. Moore, Moncef Gabbouj and Sanjit K. Mitra. *Vector SD-ROM filter for removal of impulse noise from color images.* In Eurasip Conference, ECMCS, 1999. (Cited on page 23.)

[Moulik 2002] U. Moulik and S. Bandyopadhyay. *Performance Evaluation of Some Clustering Algorithms and Validity Indices*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 12, pages 1650–1654, 2002. (Cited on page 107.)

[Munteanu 2001] Cristian Munteanu and Agostinho Rosa. *Evolutionary Image Enhancement with User Behavior Modeling*. SIGAPP Appl. Comput. Rev., vol. 9, no. 1, pages 8 – 14, April 2001. (Cited on page 26.)

[N 1998] Weyrich N and Warhola GT. *Wavelet shrinkage and generalized cross validation for image denoising*. IEEE Transactions on Image Processing, vol. 7, no. 1, pages 82–90, 1998. (Cited on page 51.)

[Nair 2008] Madhu S. Nair, K. Revathy and Rao Tatavarti. *An improved decision based algorithm for impulse noise removal*. Proceedings of 2008 International Congress on Image and Signal Processing, CISP, pages 426 – 431, May 2008. (Cited on pages 36, 149 and 251.)

[Nair 2010] Madhu S. Nair and G. Raju. *A new fuzzy-based decision algorithm for high-density impulse noise removal*. Signal, Image and Video Processing, Springer-Verlag London, September 2010. (Cited on page 11.)

[Nair 2012] MadhuS. Nair and G. Raju. *A new fuzzy-based decision algorithm for high-density impulse noise removal*. Signal, Image and Video Processing, vol. 6, no. 4, pages 579–595, 2012. (Cited on pages 13, 34, 36, 149 and 251.)

[Nash 1996] S. G. Nash and A. Sofer. Linear and nonlinear programming. MacGraw-Hill, Singapore, 1996. (Cited on page 93.)

[Nasimudeen 2010] A. Nasimudeen, Madhu S. Nair and Rao Tatavarti. *Directional switching median filter using boundary discriminative noise detection by elimination*. Signal, Image and Video Processing, pages 1–12, 2010. (Cited on pages 33 and 36.)

[Nasri 2013] M. Nasri, S. Saryazdi and H. Nezamabadi-pour. *SNLM: A switching non-local means filter for removal of high density salt and pepper noise*. Scientia Iranica, vol. 20, no. 3, pages 760 – 764, 2013. (Cited on pages 36, 149 and 251.)

[Neelamani 2004] R. Neelamani, H. Choi and R. Baraniuk. *Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems*. IEEE Transactions on Signal Processing, vol. 52, pages 418–433, 2004. (Cited on pages 42 and 48.)

[Ng 2006] Pei-Eng Ng and Kai-Kuang Ma. *A switching median filter with boundary discriminative noise detection for extremely corrupted images*. IEEE Transactions on Image

Processing, vol. 16, no. 6, pages 1506 – 1516, June 2006. (Cited on pages 30, 33, 36, 149 and 251.)

[Nikolova 2004] Mila Nikolova. *A Variational Approach to Remove Outliers and Impulse Noise*. Journal of Mathematical Imaging and Vision, vol. 20, no. 1-2, pages 99–120, 2004. (Cited on page 28.)

[Omran 2005] M. Omran, A. Enge;brecht and A Salman. *Particle Swarm Optimization method for image clustering*. International Journal of Pattern Recognition and Artificial Intelligence, vol. 19, pages 297–322, 2005. (Cited on page 121.)

[Osman 1996] IbrahimH. Osman and Gilbert Laporte. *Metaheuristics: A bibliography*. Annals of Operations Research, vol. 63, no. 5, pages 511–623, 1996. (Cited on page 93.)

[Pakhira 2004] Malay K. Pakhira, Sanghamitra Bandyopadhyay and Ujjwal Maulik. *Validity index for crisp and fuzzy clusters*. Pattern Recognition, vol. 37, no. 3, pages 487 – 501, 2004. (Cited on page 124.)

[Pakhira 2005] Malay K. Pakhira, Sanghamitra Bandyopadhyay and Ujjwal Maulik. *A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification*. Fuzzy Sets and Systems, vol. 155, no. 2, pages 191 – 214, 2005. (Cited on page 124.)

[Pok 2003] G. Pok, Y. Liu and A. S. Nair. *Selective removal of impulse noise based on homogeneity level information*. IEEE Signal Processing Letters, vol. 12, no. 1, pages 85–92, January 2003. (Cited on page 24.)

[Ponti-Jr 2013] MoacirP. Ponti-Jr, NelsonD.A. Mascarenhas, PauloJ.S.G. Ferreira and ClaudioA.T. Suazo. *Three-dimensional noisy image restoration using filtered extrapolation and deconvolution*. Signal, Image and Video Processing, vol. 7, no. 1, pages 1–10, 2013. (Cited on page 36.)

[Qiu 2009] Minmin Qiu, Liu Liu, Hongwei Ding, Jin Dong and Wei Wang. *A new hybrid variable-length GA and PSO algorithm in continuous facility location problem with capacity and service level constraints*. In IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, 2009. SOLI '09, pages 546–551, July 2009. (Cited on page 124.)

[Raj 2012] V. Naga Prudhvi Raj and T. Venkateswarlu. *Denoising of Medical Images Using Dual Tree Complex Wavelet Transform*. Procedia Technology, vol. 4, no. 0, pages 238 – 244, 2012. 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012) on February 25 - 26, 2012. (Cited on page 51.)

[Rudin 1992] Leonid I. Rudin, Stanley Osher and Emad Fatemi. *Nonlinear total variation based noise removal algorithms*. Physica D: Nonlinear Phenomena, vol. 60, no. 14, pages 259 – 268, 1992. (Cited on page 50.)

[Russo 1996] F. Russo and G. Ramponi. *A Fuzzy filter for images corrupted by impulse noise*. IEEE Signal Processing Letters, vol. 3, pages 168–170, June 1996. (Cited on page 23.)

[Russo 1999a] F. Russo. *Noise removal from image data using recursive neuro-fuzzy filters*. In Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE, volume 3, pages 1818–1822 vol.3, 1999. (Cited on page 36.)

[Russo 1999b] Fabrizio Russo. *FIRE operators for image processing*. Fuzzy Sets and Systems, vol. 103, no. 2, pages 265 – 275, 1999. Soft Computing for Pattern Recognition. (Cited on page 36.)

[Russo 2004] F. Russo. *Impulse Noise Cancellation in Image Data Using A Two-Output Nonlinear Filter*. Measurement, no. 36, pages 205–213, 2004. (Cited on page 28.)

[Sa 2009] Pankaj Kumar Sa, Ratnakar Dash and Banshidhar Majhi. *Second order difference based Detection and Directional weighted median filter for removal of Random valued impulsive noise*. IEEE International Conference on Industrial and Information Systems ICIIS, pages 362–364, December 2009. (Cited on page 32.)

[S.Chang 2000] S.Chang, B Yu and M.Vetterli. *Adaptive wavelet thresholding for image denoising and compression*. IEEE Transactions on Image Processing, vol. 9, no. 9, pages 1532–1546, 2000. (Cited on pages 6, 7, 8, 43 and 250.)

[Schulte 2006] Stefan Schulte, Bruno Huysmans, Aleksandra PiÅ¾urica, EtienneE. Kerre and Wilfried Philips. *A New Fuzzy-Based Wavelet Shrinkage Image Denoising Technique*. In Jacques Blanc-Talon, Wilfried Philips, Dan Popescu and Paul Scheunders, editeurs, Advanced Concepts for Intelligent Vision Systems, volume 4179 of *Lecture Notes in Computer Science*, pages 12–23. Springer Berlin Heidelberg, 2006. (Cited on page 44.)

[Sendur 2002] J. Sendur and I. Selesnick. *Bivariate shrinkage with local variance estimation*. IEEE Signal Processing Letter, vol. 9, no. 12, pages 438 – 441, 2002. (Cited on page 50.)

[Senel 2002] H.G. Senel, II Peters R.A. and B. Dawant. *Topological median filters*. IEEE Transactions on Image Processing, vol. 11, no. 2, pages 89–104, Feb 2002. (Cited on page 37.)

[Shi 2012] Baoli Shi, Lihong Huang and Zhi-Feng Pang. *Fast algorithm for multiplicative noise removal*. Journal of Visual Communication and Image Representation, vol. 23, no. 1, pages 126 – 133, 2012. (Cited on page 35.)

[Si] *Swarm Intelligence*. www.swarmintelligence.org/ (accessed on 25th August, 2011). (Cited on page 100.)

[Silva 2012] RicardoDutra Silva, Rodrigo Minetto, William Robson Schwartz and Helio Pedrini. *Adaptive edge-preserving image denoising using wavelet transforms*. Pattern Analysis and Applications, pages 1–14, 2012. (Cited on pages 50 and 52.)

[Smolka 2003] Bogdan Smolka. *On the new robust algorithm of noise reduction in color images*. Computers and Graphics, vol. 27, no. 4, pages 503 – 513, 2003. (Cited on page 27.)

[Srikanth 1995] R. Srikanth, R. George, N. Warsi, D. Prabhu, F.E. Petry and B.P. Buckles. *A variable-length genetic algorithm for clustering and classification*. Pattern Recognition Letters, vol. 16, no. 8, pages 789 – 800, 1995. Genetic Algorithms. (Cited on page 124.)

[Srinivasan 2007] K. S. Srinivasan and D. Ebenezer. *A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises*. IEEE Signal Processing Letters, vol. 14, no. 3, pages 189–192, 2007 2007. (Cited on pages 31, 36, 149 and 251.)

[Strela 1999] V. Strela, P.N. Heller, G. Strang, P. Topiwala and C. Heil. *The application of multiwavelet filter banks to image processing*. IEEE Transactions on Image Processing, vol. 8, pages 548–563, 1999. (Cited on page 141.)

[Sun 1994a] T. Sun and Y. Neuvo. *Detail preserving median based filters in image processing*. Pattern Recognition Letters, vol. 15, no. 4, pages 341–347, 1994. (Cited on page 23.)

[Sun 1994b] Tong Sun and Y Neuvo. *Detail-preserving median based filters in image processing*. Pattern Recognition Letters, vol. 15, no. 4, pages 341 – 347, 1994. (Cited on page 24.)

[Tan 2006] P. Tan, M. Steinbach and V.Kumar. *Introduction to Data Mining*. Pearson Education, 2006. (Cited on page 107.)

[T.D.Bui 1998] T.D.Bui and G.Y.Chen. *Translation invariant denoising using multivavelets.* IEEE Transactions on Signal Processing, vol. 46, pages 3414–3420, 1998. (Cited on page 42.)

[Teck 2008] Hasan S. M.; Talib Abdullah Zawawi; Lam Tan Kian Teck Low Khong; Al-Khaffaf. *Extended Study on Removing Gaussian Noise in Mechanical Engineering Drawing Images using Median Filters.* In Academic Journal, pages 326 – 328. Proceedings of World Academy of Science: Engineering and Technology, November 2008. (Cited on page 31.)

[Tikkanen 1999] P. E. Tikkanen. *Nonlinear wavelet and wavelet packet denoising of electrocardiogram signal.* Biological Cybernetics, vol. 80, no. 4, pages 259–267, 1999. (Cited on page 40.)

[Toprak 2006] Abdullah Toprak and Anan Ger. *Suppression of Impulse Noise in Medical Images with the Use of Fuzzy Adaptive Median Filter.* Journal of Medical Systems, vol. 30, no. 6, pages 465–471, 2006. (Cited on page 43.)

[Tou 1974] J. T. Tou and R. C. Gonzalez. *Pattren Recognition Principles.* Addison-Wesley Reading, 1974. (Cited on page 107.)

[Tsai 2002] S. Tsai. Wavelet transform and denoising. Master's thesis, 2002. (Cited on page 39.)

[Tyagi 2013] Bindiya Tyagi and H.A. Vidya. *Application of Wavelet Shrinkage Denoising Method Based on Genetic Algorithm to Partial Discharge Signals.* In Veena S. Chakravarthi, Yasha Jyothi M. Shirur and Rekha Prasad, editeurs, Proceedings of International Conference on VLSI, Communication, Advanced Devices, Signals and Systems and Networking (VCASAN-2013), volume 258 of *Lecture Notes in Electrical Engineering*, pages 273–281. Springer India, 2013. (Cited on page 54.)

[Van De Ville 2003] D. Van De Ville, M. Nachtegael, Dietrich Van der Weken, E.E. Kerre, W. Philips and I. Lemahieu. *Noise reduction by fuzzy image filtering.* IEEE Transactions on Fuzzy Systems, vol. 11, no. 4, pages 429–436, Aug 2003. (Cited on page 37.)

[Vapnik 1995] Vladimir N. Vapnik. The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA, 1995. (Cited on page 41.)

[Wang 1998] Jung-Hua Wang and Lian-Da Lin. *Image restoration using parametric adaptive fuzzy filter.* In Fuzzy Information Processing Society - NAFIPS, 1998 Conference of the North American, pages 198–202, Aug 1998. (Cited on page 24.)

[Wang 1999] Zhou Wang and David Zhang. *Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images.* IEEE Transactions on Circuits and Systems, vol. 46, no. 1, pages 78–80, January 1999. (Cited on pages 12 and 25.)

[Wang 2002] Zhou Wang and A.C. Bovik. *A universal image quality index.* Signal Processing Letters, IEEE, vol. 9, no. 3, pages 81–84, March 2002. (Cited on page 14.)

[Wang 2010] Xiang-Yang Wang and Zhong-Kai Fu. *A wavelet-based image denoising using least squares support vector machine.* Engineering Applications of Artificial Intelligence, vol. 23, no. 6, pages 862 – 871, 2010. (Cited on page 48.)

[Wang 2012a] Lei Wang, Nian de Jiang and Xing Ning. *Research on Medical Image Enhancement Algorithm Based on GSM Model for Wavelet Coefficients.* Physics Procedia, vol. 33, no. 0, pages 1298 – 1303, 2012. 2012 International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012). (Cited on page 54.)

[Wang 2012b] Yang Wang, Xiaoqian Che and Siliang Ma. *Nonlinear filtering based on 3D wavelet transform for MRI denoising.* EURASIP Journal on Advances in Signal Processing, vol. 2012, no. 1, pages 1–14, 2012. (Cited on page 52.)

[Williams 1998] EdwinA. Williams and WilliamA. Crossley. *Empirically-Derived Population Size and Mutation Rate Guidelines for a Genetic Algorithm with Uniform Crossover.* In P.K. Chawdhry, R. Roy and R.K. Pant, editeurs, Soft Computing in Engineering Design and Manufacturing, pages 163–172. Springer London, 1998. (Cited on pages 94 and 142.)

[Windyga 2001] P. S. Windyga. *Fast Impulsive Noise Removal.* IEEE Transactions on Image Processing, vol. 10, no. 1, pages 173–179, January 2001. (Cited on page 25.)

[Wong 2011] Man To Wong, Xiangjian He and Wei-Chang Yeh. *Image clustering using Particle Swarm Optimization.* In Evolutionary Computation (CEC), 2011 IEEE Congress on, pages 262–268, June 2011. (Cited on page 122.)

[Xiao 2011] Fei Xiao and Yungang Zhang. *A Comparative Study on Thresholding Methods in Wavelet-based Image Denoising.* Procedia Engineering, vol. 15, no. 0, pages 3998 – 4003, 2011. CEIS 2011. (Cited on page 50.)

[Xu 2002] X. Xu and E. L. Miller. *Adaptive Two-Pass Median Filter to Remove Impulsive Noise.* In Proceedings of International Conference on Image Processing, no. I, pages 808–811, September 2002. (Cited on page 26.)

[Xu 2009] Haixiang Xu, Guangxi Zhu, Fuyuan Peng and Desheng Wang. *Adaptive fuzzy switching filter for images corrupted by impulse noise*. In Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on, volume 2, pages 792–795 Vol.2, 2009. (Cited on page 32.)

[Yang 2010] Lei Yang, Richard Parton, Graeme Ball, Zhen Qiu, Alan H. Greenaway, Ilan Davis and Weiping Lu. *An adaptive non-local means filter for denoising live-cell images and improving particle detection*. Journal of Structural Biology, vol. 172, no. 3, pages 233 – 243, 2010. (Cited on page 48.)

[Yuksel 2003] M. Emin Yuksel and Alper Basturk. *Efficient Removal of Impulse Noise from Highly Corrupted Digital Images by a Simple Neuro-Fuzzy Operator*. AEU - International Journal of Electronics and Communications, vol. 57, no. 3, pages 214 – 219, 2003. (Cited on page 37.)

[Yüksel 2004] M. Emin Yüksel, Alper Bastürk and Erkan Besdok. *Detail-preserving Restoration of Impulse Noise Corrupted Images by a Switching Median Filter Guided by a Simple Neuro-fuzzy Network*. EURASIP J. Appl. Signal Process., vol. 2004, pages 2451 – 2461, jan 2004. (Cited on page 24.)

[Yuksel 2005] M. Emin Yuksel and Alper Basturk. *A simple generalized neuro-fuzzy operator for efficient removal of impulse noise from highly corrupted digital images*. AEU - International Journal of Electronics and Communications, vol. 59, no. 1, pages 1 – 7, 2005. (Cited on page 37.)

[Yuksel 2006] M.E. Yuksel. *A hybrid neuro-fuzzy filter for edge preserving restoration of images corrupted by impulse noise*. IEEE Transactions on Image Processing, vol. 15, no. 4, pages 928–936, April 2006. (Cited on page 30.)

[Z. 1996] Michalewicz Z. *Genetic Algorithms +Data Structures = Evolution Programms*. Springer, 1996. (Cited on page 98.)

[Zadeh 1965] L.A. Zadeh. *Fuzzy sets*. Information and Control, vol. 8, no. 3, pages 338 – 353, 1965. (Cited on page 132.)

[Zeng 2012] Hanglin Zeng, Yuan-Zhong Liu, Yu-Mei Fan and Xuefei Tang. *An Improved Algorithm for Impulse Noise by Median Filter*. AASRI Procedia, vol. 1, no. 0, pages 68 – 73, 2012. AASRI Conference on Computational Intelligence and Bioinformatics. (Cited on page 35.)

[Zhang 2001] Xiao-Ping Zhang. *Thresholding neural network for adaptive noise reduction.* IEEE Transactions on Neural Networks, vol. 12, no. 3, pages 567–584, May 2001. (Cited on page 41.)

[Zhang 2002] S. Zhang and M. A. Karim. *A new impulse detector for switching median filters.* IEEE Signal Processing Letters, vol. 9, no. 11, pages 360–33, November 2002. (Cited on pages 24 and 27.)

[Zhang 2007] Shaoming Zhang and Ying Chen. *Image Denoising Based on Wavelet Support Vector Machine.* In Yuping Wang, Yiu-ming Cheung and Hailin Liu, editeurs, Computational Intelligence and Security, volume 4456 of *Lecture Notes in Computer Science*, pages 963–971. Springer Berlin Heidelberg, 2007. (Cited on page 45.)

Index

Accurate Noise Detector, 27
Adaptive Tow-Pass Median Filter, 26
Anisotropic Noise, 217

Bit Error Rate (BER), 216

Chrominance/Chroma (Cb and Cr), 218
Clustering, 107
Color Image Components, 218

Denoising, 4
Differential Ranked Impulse Detector, 28
Digital Image, 3
Digital Image Processing, 3
DWT, 6

Enhanced Ranked Impulse Detector, 28
Euclidean distance function, 118

Filters
    Adaptive Center Weighted Median Filter, 25
    Averaging Filter, 22
    Directional Weighted Minimum Deviation Filter (DWMD), 65
    Filtering in Frequency Domain, 4
    Filtering in Spatial Domain, 4
    MAD Filter, 28
    Median Filter, 22
    Minimum-Maximum Exclusive Mean Filter, 24
    Peak and Valley Filters, 25
    Progressive Switching Median Filter, 25
    Rank Order Mean Filter, 23
    Single Step Filters, 22
    Switching Median Filter, 23
    Tri-State Median Filters, 24
    Two-Output Filters, 28
    Variable Mask Median Filter(VMM), 77

Gaussian Noise, 11
Genetic Algorithm (GA)
    Crossover, 97
    Elitism, 97
    Selection, 97
Genetic Algorithm(GA)
    Chromosome Encoding, 95
    Mutation, 99
    Selection
        Binary Tournament Selection (BTS), 97
Gray Scale Image, 3

IDWT, 6
Image Quality, 12
Inter-cluster distance, 122
Intra-cluster distance, 122

Laplacian operator, 108
Luminance/Luma (Y), 218

multi resolution analysis (MRA), 141

noise
    myopotential, 41
Noise Model, 10

Optimization
    Genetic Algorithm (GA), 95
    Particle Swarm Optimization(PSO), 100

Particle Swarm Optimization
Fixed Length Swarm, 119
Particle Swarm Optimization
Different length swarm, 124
Peak Signal to Noise Ratio (PSNR(dB), 13

Quantization Noise, 216

Random Valued Noise (RVN), 12
Regression Analysis, 218

Shot Noise, 10
Shrinking
BayesShrink, 8
SurShrink, 8
VisuShrink, 8
Structural Similarity (SSIM), 13
subband, 6

Thresholding
Hard, 9
Soft, 9

Universal Image Quality Index (UIQI), 14

Wavelet, 6
thresholding, 6