

Electronic Thesis and Dissertation Repository

12-1-2014 12:00 AM

BM3D Image Denoising using SSIM Optimized Wiener Filter

Mahmud Hasan

The University of Western Ontario

Supervisor

Dr. Mahmoud R. El-Sakka

The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

© Mahmud Hasan 2014

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Other Computer Sciences Commons](#)

Recommended Citation

Hasan, Mahmud, "BM3D Image Denoising using SSIM Optimized Wiener Filter" (2014). *Electronic Thesis and Dissertation Repository*. 2547.

<https://ir.lib.uwo.ca/etd/2547>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.

BM3D IMAGE DENOISING USING SSIM OPTIMIZED WIENER FILTER
(Thesis format: Monograph)

by

Mahmud Hasan

Graduate Program in Department of Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

© Mahmud Hasan 2014

Abstract

Image denoising is considered as a salient pre-processing in sophisticated imaging applications. Over decades, numerous studies have been conducted in denoising. Recently proposed *Block Matching and 3D (BM3D) Filtering* added a new dimension to the study of denoising. *BM3D* is the current state-of-the-art of denoising and is capable of achieving better denoising as compared to any other existing method. However, the performance is not yet on the bound for image denoising. Therefore, there is scope to improve *BM3D* to achieve high quality denoising. In this thesis, to improve *BM3D*, we first attempted to improve Wiener filter (the core of *BM3D*) by maximizing the Structural Similarity (SSIM) between the true and the estimated image, instead of minimizing the Mean Square Error (MSE) between them. Moreover, for the *DC-Only BM3D* profile, we introduced a 3D zigzag thresholding. Experimental results demonstrate that regardless of the type of the image, our proposed method achieves better denoising than that of *BM3D*.

Keywords: Image Denoising, Image Restoration, *BM3D*, Wiener Filter, Structural Similarity, Collaborative Filtering, Hard Thresholding, Mean Square Error

Dedication

To my beloved wife

Maksuda Khatun

Who sacrificed her career to build up mine

Acknowledgments

At this happy moment, I would like to express my heartiest gratitude to the Almighty for the strength, patience, intelligence and endless kindness he provided me with to finalize this thesis.

I am grateful to my honorable supervisor Dr. Mahmoud R. El-Sakka for his valuable direction, guidance, comments and encouragement throughout this work. It was an absolute honor and privilege to work with such a modest and wise person like him. His wisdom and notable thoughts have helped this thesis become an ultimate success. He redirected my view of thinking to a progressive path every time I discussed my research problems with him. This dissertation under his supervision will always be a remarkable experience in my life.

On the eve of my graduation, I would like to remember and broaden my delicate respect to all the professors of The University of Western Ontario who helped me through many courses to build my background for this dissertation. Without their sincere care, the understanding of Image Processing fundamentals would have been impossible for me.

I acknowledge the support of my friends, family members, my loving mother and research group members throughout this long tiring period. Their inspiration and encouragement strengthened me in my tough time to keep focused.

Finally, I would like to thank The Department of Computer Science at The University of Western Ontario to fund my graduate study. I would also like to thank The Government of People's Republic of Bangladesh and The Chittagong University of Engineering & Technology that provided me almost cost free education up to undergraduate and paved me a way for higher studies.

Contents

Abstract	ii
Dedication	iii
Acknowledgments	iv
List of Figures	viii
List of Tables	xii
List of Appendices	xiv
1 Introduction	1
1.1 Motivations	1
1.2 Problem Statement	2
1.3 Objectives	2
1.4 Thesis Contribution	3
1.5 Thesis Outline	3
2 Background Study	4
2.1 Noise and its Variants on Digital Images	4
2.1.1 Additive Noise	4
Additive White Gaussian Noise (AWGN)	5
Rayleigh Noise Distribution	5
Gamma Distribution	6
Exponential Noise Distribution	7
2.1.2 Multiplicative Noise	8
Salt and Pepper Noise	8
Speckle Noise	9
Periodic Noise	9
2.2 Image Denoising	9
2.2.1 Spatial Domain Denoising	11
Mean Filter	11
Median Filter	11
Min and Max Filter	11
Midpoint Filter	12
Gaussian Smoothing	12

	Conservative Smoothing	12
	Unsharp Filter	12
	Laplacian of Gaussian	13
2.2.2	Frequency Domain Denoising	13
	Low Pass Filter	13
	High Pass Filter	14
	Band Pass Filter	14
2.2.3	Edge Preserving Denoising	14
	Anisotropic Diffusion	14
2.2.4	Non Edge Preserving Denoising	15
2.3	Image Restoration	15
2.3.1	Degradation Model	15
2.3.2	Estimation of Degradation Function	16
2.3.3	Effect of Degradation Function on Image Restoration	16
2.3.4	Techniques of Image Restoration	17
2.3.5	Inverse Filter	18
2.3.6	Wiener Filter	19
2.4	Block Matching and 3D (<i>BM3D</i>) Filtering based Image Denoising	20
2.4.1	Algorithm of <i>BM3D</i>	20
	<i>BM3D</i> First Step	20
	<i>BM3D</i> Second Step	21
2.4.2	Improvements over <i>BM3D</i>	22
2.4.3	Persisting Limitations	22
	Poor Performance of DC-Only Profile	22
	Choice of Wiener Filter	23
	Parameterized Setup	23
	Poor Performance for Higher Noise Levels	23
	Poor Performance for Non Textured Images	23
2.4.4	Possible Improvements	23
3	Methodology	24
3.1	Improvement Over Wiener Filter	24
3.1.1	MSE Optimized Wiener Filter	24
3.1.2	Structural Similarity (SSIM)	25
3.1.3	SSIM Optimized Wiener Filter	27
3.2	Improvement Over <i>BM3D</i>	27
3.2.1	DC-Only Profile	27
	3D Zigzag Based Thresholding	28
3.2.2	Wavelet Profile	29
	Normal Profile	30
	Fast Profile	30
3.2.3	Color Profile	30
4	Experimental Results and Analysis	32
4.1	Data Set	32

4.2	Performance Measurement Metrics	32
4.2.1	Objective Fidelity Criteria	32
	Peak Signal to Noise Ratio (PSNR)	33
	Structural Similarity (SSIM)	33
4.2.2	Subjective Fidelity Criteria	33
4.3	Improved Wiener Filter for Image Denoising	34
4.4	Improved Wiener Filter for Image Restoration	34
4.5	Improved <i>BM3D</i> for DC-Only Profile	36
4.5.1	Parameterized Setup for DC-Only Profile	36
4.5.2	With Improved Wiener Filter Only	38
	Non-Overlapping Similar Patches	38
	Moderate (50%) Overlapping Similar Patches	38
	Maximum (75%) Overlapping Similar Patches	40
4.5.3	With 3D Zigzag Thresholding Only	44
	Non-Overlapping Similar Patches	44
	Moderate (50%) Overlapping Similar Blocks	44
	Maximum (75%) Overlapping Similar Blocks	44
4.5.4	With Improved Wiener Filter and 3D Zigzag Thresholding	47
	Non-Overlapping Similar Patches	49
	Moderate (50%) Overlapping Similar Blocks	49
	Maximum (75%) Overlapping Similar Blocks	49
4.5.5	Effect of Gamma (γ) in 3D Zigzag Thresholding	54
4.5.6	Performance Bound for 3D Zigzag Thresholding	55
4.6	Improved <i>BM3D</i> for Wavelet Profile	56
4.6.1	Parameterized Setup for Wavelet Profile	56
4.6.2	Normal Profile	56
4.6.3	Fast Profile	58
4.7	Extension of Wavelet Profile for Color Image Denoising	59
4.8	Summary of Results	62
4.9	Intensity Profile	66
5	Conclusion and Future Work	68
5.1	Conclusion	68
5.2	Future Work	68
	Bibliography	70
	A Detail Objective Experimental Results	73
	B Detail Subjective Experimental Results	86
	Curriculum Vitae	123

List of Figures

2.1	Distribution of AWGN	5
2.2	Example of Additive White Gaussian Noise. Left: Original Lena Image. Right: AWGN Added with Zero Mean and 0.05 Variance.	6
2.3	Rayleigh Distribution	6
2.4	Example of Rayleigh Noise. Left: Original Lena Image. Right: Rayleigh Noise Added with $a = 0$ and $b = 0.10$	7
2.5	Gamma Distribution	7
2.6	Example of Gamma Noise. Left: Original Lena Image. Right: Gamma Noise Added with $a = 10$ and $b = 2$	8
2.7	Example of Salt and Pepper Noise. Left: Original Lena Image. Right: Salt and Pepper Added to Lena Image.	9
2.8	Example of Speckle Noise. Left: Original Lena Image. Right: Speckle Noise with $Variance = 0.05$	10
2.9	Example of Periodic Noise. Left: Original Lena Image. Right: Periodic Noise Added to Lena Image.	10
2.10	Example images. Left: Cameraman. Right: House	17
2.11	Spectrum of The Images of Fig. 2.10.	17
2.12	Restored Cameraman Image using Power Spectrum of Cameraman	18
2.13	Restored Cameraman Image using Power Spectrum of House	18
2.14	BM3D Block Diagram	20
3.1	Effect of MSE on Brightness Increase	25
3.2	Effect of MSE on Brightness Change	26
3.3	Hard Thresholding in DC-Only Profile	28
3.4	ZigZag Thresholding in DC-Only Profile	29
3.5	ZigZag Ordering	30
4.1	Test Image Set	33
4.2	Average PSNR Comparison for Denoising	34
4.3	Average SSIM Comparison for Denoising	35
4.4	Average PSNR Comparison for Restoration	35
4.5	Average SSIM Comparison for Restoration	35
4.6	Average PSNR Comparison with No Overlapping for DC-Only Profile	39
4.7	Average SSIM Comparison with No Overlapping for DC-Only Profile	39
4.8	Subjective Comparison of DC-Only Profile with No Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.	39
4.9	Average PSNR Comparison with 50% Overlapping for DC-Only Profile	40

4.10	Average SSIM Comparison with 50% Overlapping for DC-Only Profile	41
4.11	Subjective Comparison of DC-Only Profile with 50% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.	41
4.12	Average PSNR Comparison with 75% Overlapping for DC-Only Profile	42
4.13	Average SSIM Comparison with 75% Overlapping for DC-Only Profile	43
4.14	Subjective Comparison of DC-Only Profile with 50% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.	43
4.15	Average PSNR Comparison of DC-Only Profile with No Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)	45
4.16	Average SSIM Comparison of DC-Only Profile with No Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)	45
4.17	Subjective Comparison of DC-Only Profile with No Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.	45
4.18	Average PSNR Comparison of DC-Only Profile with 50% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)	46
4.19	Average SSIM Comparison of DC-Only Profile with 50% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)	46
4.20	Subjective Comparison of DC-Only Profile with 50% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.	46
4.21	Average PSNR Comparison of DC-Only Profile with 75% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)	48
4.22	Average SSIM Comparison of DC-Only Profile with 75% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)	48
4.23	Subjective Comparison of DC-Only Profile with 75% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.	48
4.24	Average PSNR Comparison of DC-Only Profile with Non Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)	50
4.25	Average SSIM Comparison of DC-Only Profile with Non Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)	50
4.26	Subjective Comparison of DC-Only Profile with Non Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.	51
4.27	Average PSNR Comparison of DC-Only Profile with 50% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)	51
4.28	Average SSIM Comparison of DC-Only Profile with 50% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)	52
4.29	Subjective Comparison of DC-Only Profile with 50% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.	52
4.30	Average PSNR Comparison of DC-Only Profile with 75% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)	53
4.31	Average SSIM Comparison of DC-Only Profile with 75% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)	53
4.32	Subjective Comparison of DC-Only Profile with 75% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.	55
4.33	Average PSNR Comparison of Normal Profile of BM3D with Proposed Method	57
4.34	Average SSIM Comparison of Normal Profile of BM3D with Proposed Method	57

4.35	Subjective Measure for Normal Profile of BM3D with Proposed Method	58
4.36	Average PSNR Comparison of Fast Profile of BM3D with Proposed Method	60
4.37	Average SSIM Comparison of Fast Profile of BM3D with Proposed Method	60
4.38	Subjective Measure for Fast Profile of BM3D with Proposed Method	61
4.39	24 – <i>bit</i> True Color Images used for Color Image Denoising Experiments	61
4.40	Average PSNR Comparison of Color Profile (Normal) of BM3D with Proposed Method	63
4.41	Average SSIM Comparison of Color Profile (Normal) of BM3D with Proposed Method	63
4.42	Subjective Measure for Color Profile (Normal) of BM3D with Proposed Method	63
4.43	Average PSNR Comparison of Color Profile (Fast) of BM3D with Proposed Method	64
4.44	Average SSIM Comparison of Color Profile (Fast) of BM3D with Proposed Method	64
4.45	Subjective Measure for Color Profile (Fast) of BM3D with Proposed Method	65
4.46	Lena Image for Intensity Profile Calculation. Red Line shows the Scan Line taken as input to Intensity Profile	66
4.47	Intensity Profile for the Scan Line shown in Fig. 4.46	66
4.48	Intensity Profiles for <i>BM3D</i> and Proposed Method with respect to the True Image	67
B.1	Wavelet (Normal) Profile Output of Lena Image for $\sigma = 40$	87
B.2	Wavelet (Normal) Profile Output of Lena Image for $\sigma = 70$	88
B.3	Wavelet (Normal) Profile Output of Lena Image for $\sigma = 100$	89
B.4	Wavelet (Normal) Profile Output of Barbara Image for $\sigma = 40$	90
B.5	Wavelet (Normal) Profile Output of Barbara Image for $\sigma = 70$	91
B.6	Wavelet (Normal) Profile Output of Barbara Image for $\sigma = 100$	92
B.7	Wavelet (Normal) Profile Output of Peppers Image for $\sigma = 40$	93
B.8	Wavelet (Normal) Profile Output of Peppers Image for $\sigma = 70$	94
B.9	Wavelet (Normal) Profile Output of Peppers Image for $\sigma = 100$	95
B.10	Wavelet (Fast) Profile Output of Lena Image for $\sigma = 40$	96
B.11	Wavelet (Fast) Profile Output of Lena Image for $\sigma = 70$	97
B.12	Wavelet (Fast) Profile Output of Lena Image for $\sigma = 100$	98
B.13	Wavelet (Fast) Profile Output of Barbara Image for $\sigma = 40$	99
B.14	Wavelet (Fast) Profile Output of Barbara Image for $\sigma = 70$	100
B.15	Wavelet (Fast) Profile Output of Barbara Image for $\sigma = 100$	101
B.16	Wavelet (Fast) Profile Output of Peppers Image for $\sigma = 40$	102
B.17	Wavelet (Fast) Profile Output of Peppers Image for $\sigma = 70$	103
B.18	Wavelet (Fast) Profile Output of Peppers Image for $\sigma = 100$	104
B.19	Color (Normal) Profile Output of Lena Image for $\sigma = 40$	105
B.20	Color (Normal) Profile Output of Lena Image for $\sigma = 70$	106
B.21	Color (Normal) Profile Output of Lena Image for $\sigma = 100$	107
B.22	Color (Normal) Profile Output of Peppers Image for $\sigma = 40$	108
B.23	Color (Normal) Profile Output of Peppers Image for $\sigma = 70$	109
B.24	Color (Normal) Profile Output of Peppers Image for $\sigma = 100$	110
B.25	Color (Normal) Profile Output of Barbara Image for $\sigma = 40$	111

B.26 Color (Normal) Profile Output of Barbara Image for $\sigma = 70$	112
B.27 Color (Normal) Profile Output of Barbara Image for $\sigma = 100$	113
B.28 Color (Fast) Profile Output of Lena Image for $\sigma = 40$	114
B.29 Color (Fast) Profile Output of Lena Image for $\sigma = 70$	115
B.30 Color (Fast) Profile Output of Lena Image for $\sigma = 100$	116
B.31 Color (Fast) Profile Output of Peppers Image for $\sigma = 40$	117
B.32 Color (Fast) Profile Output of Peppers Image for $\sigma = 70$	118
B.33 Color (Fast) Profile Output of Peppers Image for $\sigma = 100$	119
B.34 Color (Fast) Profile Output of Barbara Image for $\sigma = 40$	120
B.35 Color (Fast) Profile Output of Barbara Image for $\sigma = 70$	121
B.36 Color (Fast) Profile Output of Barbara Image for $\sigma = 100$	122

List of Tables

4.1	Average PSNR-SSIM Comparison of Wiener Filter for Denoising	36
4.2	Average PSNR-SSIM Comparison of Wiener Filter for Restoration	37
4.3	Parameterized Setup for <i>BM3D</i> DC-Only Profile	37
4.4	Performance Comparison of DC-Only Profile with No Overlapping using Improved Wiener Filter	38
4.5	Performance Comparison of DC-Only Profile with 50% Overlapping using Improved Wiener Filter	40
4.6	Performance Comparison of DC-Only Profile with 75% Overlapping using Improved Wiener Filter	42
4.7	Performance Comparison of DC-Only Profile with No Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)	44
4.8	Performance Comparison of DC-Only Profile with 50% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)	47
4.9	Performance Comparison of DC-Only Profile with 75% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)	47
4.10	Performance Comparison of DC-Only Profile with No Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)	49
4.11	Performance Comparison of DC-Only Profile with 50% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)	54
4.12	Performance Comparison of DC-Only Profile with 75% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)	54
4.13	Parameterized Setup for Wavelet Profile of <i>BM3D</i>	58
4.14	Performance Comparison of Normal Profile with Proposed Method	59
4.15	Performance Comparison of Fast Profile with Proposed Method	59
4.16	Performance Comparison of Color Profile (Normal) with Proposed Method	62
4.17	Performance Comparison of Color Profile (Fast) with Proposed Method	65
4.18	Summary of Experimental Results	65
A.1	Wavelet (Normal) Profile for $\sigma = 10$	73
A.2	Wavelet (Normal) Profile for $\sigma = 20$	74
A.3	Wavelet (Normal) Profile for $\sigma = 30$	74
A.4	Wavelet (Normal) Profile for $\sigma = 40$	74
A.5	Wavelet (Normal) Profile for $\sigma = 50$	75
A.6	Wavelet (Normal) Profile for $\sigma = 60$	75
A.7	Wavelet (Normal) Profile for $\sigma = 70$	75
A.8	Wavelet (Normal) Profile for $\sigma = 80$	76

A.9 Wavelet (Normal) Profile for $\sigma = 90$	76
A.10 Wavelet (Normal) Profile for $\sigma = 100$	76
A.11 Wavelet (Fast) Profile for $\sigma = 10$	77
A.12 Wavelet (Fast) Profile for $\sigma = 20$	77
A.13 Wavelet (Fast) Profile for $\sigma = 30$	77
A.14 Wavelet (Fast) Profile for $\sigma = 40$	78
A.15 Wavelet (Fast) Profile for $\sigma = 50$	78
A.16 Wavelet (Fast) Profile for $\sigma = 60$	78
A.17 Wavelet (Fast) Profile for $\sigma = 70$	79
A.18 Wavelet (Fast) Profile for $\sigma = 80$	79
A.19 Wavelet (Fast) Profile for $\sigma = 90$	79
A.20 Wavelet (Fast) Profile for $\sigma = 100$	80
A.21 Color (Normal) Profile for $\sigma = 10$	80
A.22 Color (Normal) Profile for $\sigma = 20$	80
A.23 Color (Normal) Profile for $\sigma = 30$	80
A.24 Color (Normal) Profile for $\sigma = 40$	81
A.25 Color (Normal) Profile for $\sigma = 50$	81
A.26 Color (Normal) Profile for $\sigma = 60$	81
A.27 Color (Normal) Profile for $\sigma = 70$	81
A.28 Color (Normal) Profile for $\sigma = 80$	82
A.29 Color (Normal) Profile for $\sigma = 90$	82
A.30 Color (Normal) Profile for $\sigma = 100$	82
A.31 Color (Fast) Profile for $\sigma = 10$	82
A.32 Color (Fast) Profile for $\sigma = 20$	83
A.33 Color (Fast) Profile for $\sigma = 30$	83
A.34 Color (Fast) Profile for $\sigma = 40$	83
A.35 Color (Fast) Profile for $\sigma = 50$	83
A.36 Color (Fast) Profile for $\sigma = 60$	84
A.37 Color (Fast) Profile for $\sigma = 70$	84
A.38 Color (Fast) Profile for $\sigma = 80$	84
A.39 Color (Fast) Profile for $\sigma = 90$	85
A.40 Color (Fast) Profile for $\sigma = 100$	85

List of Appendices

Appendix A Detail Objective Experimental Results	73
Appendix B Detail Subjective Experimental Results	86

Chapter 1

Introduction

A digital image is defined as a two dimensional discrete function $f(i, j)$, where i and j are spatial co-ordinates of the image f . (i, j) denotes a location of the image. The value of i and j vary depending on the dimension of the image, however, the value of $f(i, j)$ is limited between 0 – 255 for gray-scale images. The value of $f(i, j)$ for any given i and j is called the *intensity* of the image at location (i, j) . The intensity is also known as *brightness* or *pixel value* of the image.

There are a number of mechanical and electronic interference introduced during the image acquisition process. Such interferences generate some unexpected or random brightness information known as *noise*. For general purpose images, the noise may be negligible and often ignored. However, there are sophisticated imaging applications where even the small amount of noise makes further processing impossible. In such cases, reducing noise from the image without much degrading the image quality is a challenging task. Image denoising is often considered as an important pre-processing step for applications like medical and satellite imaging, image segmentation, object recognition and image classifications. Image denoising is a vast and well studied area, yet there is scope and necessity for improvement.

In this chapter, we will address the extreme necessity of high quality image denoising, briefly state our problem description and mention the objectives of this thesis. We will also state the major contributions and outline of this thesis.

1.1 Motivations

There are different types of noise that can contaminate a digital image. Depending on the noise type, there are various algorithms present in the literature for denoising the image. *Block Matching and 3D Filtering (BM3D)* [1] is one such popular algorithm that reduces *Additive White Gaussian Noise (AWGN)* [2] from digital images.

In terms of denoising performance, *BM3D* is considered as the best denoising filter so far. It exhibits remarkable results as compared to other existing methods. *BM3D* works in two identical steps. In first step, it generates a basic estimate of the noisy image using hard thresholding. Then in the second step, it uses Wiener filter to actually denoise the noisy image. To do so, *BM3D* uses the basic estimate generated from the first step as an oracle in the Wiener filter.

Wiener filter is an age-old benchmark for image denoising and restoration [3]. This filter needs a degradation function for denoising or restoration. The better the degradation function is, the more denoising is achievable by Wiener filter. *BM3D* uses the basic estimated image from the first step as the degradation function of Wiener filter. Thus the ultimate performance of *BM3D* largely depends on how good the basic estimate is.

Although *BM3D* achieves good denoising performance, it is not sufficient to denoise highly noisy images specially generated in sensitive applications like satellite imaging. In other words, the performance of *BM3D* decreases with the increase of noise level. Again, among the different profiles of *BM3D* (a profile is a specific set of parameters), the *DC-Only* profile (meaning that the 3D transform used is the 3D-DCT) generally performs poorer than the others. Therefore, there is scope to either propose better denoising technique than *BM3D* or to make *BM3D* perform better than what it can currently do.

1.2 Problem Statement

Wiener filter [3] for degraded signal restoration was proposed more than 50 years ago, though the use of Wiener filter in digital image processing was introduced later. Different researches attempted to improve the performance of Wiener filter, however, they did not directly address one persisting problem of Wiener filter that it uses an objective function MSE which is often a misleading measure. In other words, we can use better measure than MSE as the objective function of Wiener filter. Also, if Wiener filter can be improved, we can also improve the performance of *BM3D*, since it uses Wiener filter as one of its fundamental components.

In this thesis, we will primarily focus on the improvement of Wiener filter. Then we will use this improved Wiener filter in *BM3D* to improve its response as well. We will understand throughout this thesis that our improved Wiener filter will eventually improve the performance of all profiles of *BM3D*. In addition, we will also design some other components to improve the performance of *BM3D*, specially the performance of *DC-Only* profile.

1.3 Objectives

This thesis has several objectives. Although improving the performance of *BM3D* is our major objective, we have some other objectives as well in order to improve *BM3D* performance. We will discuss a lot of details in this thesis, starting from the denoising background to high level sparse collaborative filtering. However, we will keep the reader focused on the following objectives that we will meet at the end of this thesis.

1. Improve the performance of Wiener filter for image denoising.
2. Improve the performance of Wiener filter for image restoration.
3. Improve the performance of *DC-Only* profile of *BM3D*.
4. Improve the performance of *Wavelet (Normal and Fast)* profile of *BM3D*.
5. Improve the performance of *Color (Normal and Fast)* profile of *BM3D*.

1.4 Thesis Contribution

The major contribution of this thesis is the improvement over the different profiles of the current state of the art denoising algorithm *BM3D*. In achieving this, we also contribute to the denoising/restoration benchmark Wiener filter. We propose an improved Wiener filter that essentially works better in terms of denoising/restoration performance than the existing one. We use this improved Wiener filter in *BM3D*. Experimental results demonstrate that our proposed method works considerably better than original *BM3D*, thus taking a further step towards the performance bound of image denoising.

Moreover, there is no open-source *BM3D* code developed using Matlab. The only *executable* Matlab code available online is provided by the authors of *BM3D*. Although there is another C source code provided by Marc Lebrun [4], users and researchers of image denoising from all over the world are still looking for an open-source Matlab code of *BM3D* which is easier to use, edit, modify and execute. Therefore, in this thesis, we develop a Matlab source code of *BM3D*. We made sure that the performance given by our developed code is a *true copy* of the original *BM3D* before we performed the experiments. Development of this Matlab code is another contribution of this thesis.

1.5 Thesis Outline

This thesis is outlined as follows.

In **Chapter 2**, we will discuss the image denoising in details. We will start from the scratch of denoising and cover up to very recent studies of denoising. This chapter will provide the reader a solid background with fundamental knowledge of image denoising and an overall idea of how the current patch-based image denoising systems work.

In **Chapter 3**, we will present our proposed methods in detail. This chapter will also discuss the effect of each of our proposals on the denoising performance.

We will discuss different experimental setup and their corresponding results in **Chapter 4**. This chapter will prove that the objectives mentioned in **Chapter 1** has been achieved. This chapter covers only an average performance measure, the in-depth output and comparative results are given in Appendix A and B.

Finally, we will extend our present idea for possible future works and draw conclusion in **Chapter 5**.

Chapter 2

Background Study

Image denoising refers to reducing noise from digital images. Image denoising is an important application since the captured digital images are often contaminated by different types of noise due to a number of factors such as the performance of image sensors present at the capturing devices and the environment condition while capturing the image. Almost all the image processing applications require the image to be as noiseless as possible. Therefore, denoising stands as a significant area of study in digital image processing.

In this chapter, we will review the types of noise present in digital images, existing denoising techniques, degradation models and restoration techniques. We will also study the current state of the art of image denoising techniques and its related literature.

2.1 Noise and its Variants on Digital Images

Generally, noise is a random change of spatial information such as brightness or color. There are many variants of this randomness; however, they can roughly be classified into two groups- *additive* and *multiplicative*.

2.1.1 Additive Noise

If the noise is independent of image signal and is superimposed upon the image, the noise is considered as additive noise. Equation 2.1 gives the mathematical expression for this noise.

$$z(i, j) = y(i, j) + \eta(i, j) \quad (2.1)$$

Here, the i and j are spatial coordinates for the image location, $y(i, j)$ is the true image signal and $\eta(i, j)$ is the noise superimposed upon the image signal $y(i, j)$. $z(i, j)$ is the image contaminated by additive noise.

There are many variants of additive noise. We will provide a brief introduction to them in the following subsections.

Additive White Gaussian Noise (AWGN)

When the Fourier spectrum of the noise is constant, it is referred to as white. This noise follows the Gaussian (normal) distribution [5] and its power spectral density is constant. Therefore this noise is named AWGN [6]. The probability density function of a Gaussian random variable, z , is given by Equation 2.2.

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (2.2)$$

In this equation, z represents gray level, μ is the *mean* of gray values and σ is the *standard deviation* of the distribution. Generally, the standard deviation is not used for calculation of white Gaussian noise, rather, a squared standard deviation (σ^2), called *variance*, is used. The white Gaussian noise is identically distributed and statistically independent of the image information. The distribution of AWGN is shown in Fig. 2.1. If we apply this noise on the test image Lena, we will have the output as shown in Fig. 2.2.

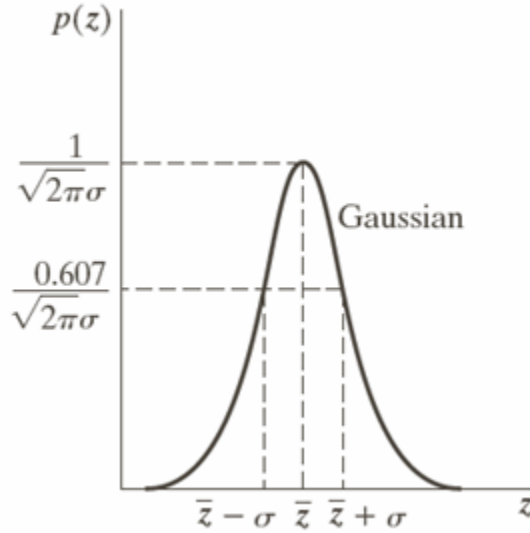


Figure 2.1: Distribution of AWGN

Since we will work with only AWGN throughout this study, for the rest of the part of this thesis, the term *noise* will refer to *AWGN* unless specified otherwise.

Rayleigh Noise Distribution

The probability density function of Rayleigh distribution is given by Equation 2.3. Unlike AWGN, its density is right skewed as shown in Fig. 2.3. The mean and variance of Rayleigh distribution is given by $\mu = a + \sqrt{\pi b}/4$ and $\sigma^2 = \frac{b(4-\pi)}{4}$, respectively. An example of Rayleigh noise on Lena image is shown in Fig. 2.4.

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \quad (2.3)$$



Figure 2.2: Example of Additive White Gaussian Noise. Left: Original Lena Image. Right: AWGN Added with Zero Mean and 0.05 Variance.

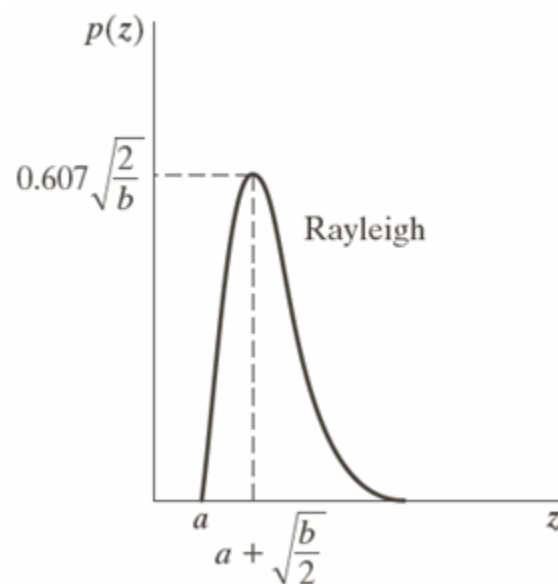


Figure 2.3: Rayleigh Distribution

Gamma Distribution

The Gamma noise is also known as Erlang noise. Like Rayleigh, this distribution is also right skewed as shown in Fig. 2.5. The distribution is given by Equation 2.4. Gamma distribution depends on two parameters a and b where, a must be greater than 0 and b is a positive integer. The mean and variance of this distribution is given by $\mu = \frac{b}{a}$ and $\sigma^2 = \frac{b}{a^2}$, respectively. An example of gamma noise added to Lena image is shown in Fig. 2.6



Figure 2.4: Example of Rayleigh Noise. Left: Original Lena Image. Right: Rayleigh Noise Added with $a = 0$ and $b = 0.10$.

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \quad (2.4)$$

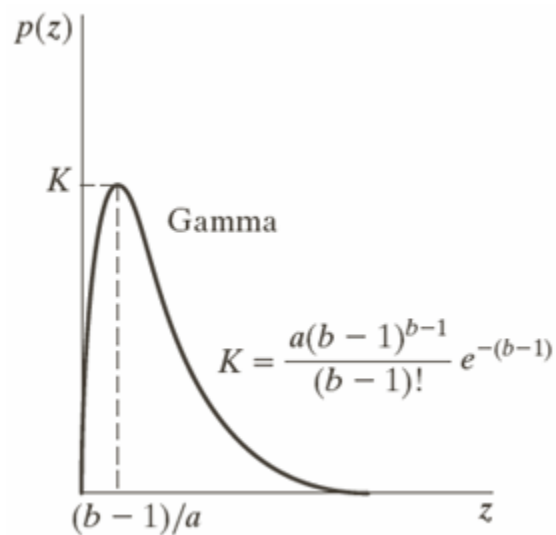


Figure 2.5: Gamma Distribution

Exponential Noise Distribution

As indicated by the name, exponential noise is an exponential distribution or an exponential function for all gray levels above or equal 0. This is basically a special case of Erlang(Gamma)



Figure 2.6: Example of Gamma Noise. Left: Original Lena Image. Right: Gamma Noise Added with $a = 10$ and $b = 2$.

distribution with $b = 1$. For instance, let us consider Fig. 2.5 with $b = 1$. If $b = 1$, the dotted line overlaps the $p(z)$ axis thereby shifting the curve its left. Consequently, we get a downward exponential distribution.

2.1.2 Multiplicative Noise

Multiplicative noise is the second major group of noise that may also contaminate a digital image. Unlike additive noise, multiplicative noise is signal (i.e., image) dependent and multiplied to the image [7]. In the following subsections, we will discuss a few common multiplicative noise.

Salt and Pepper Noise

The salt and pepper noise is a kind of impulse noise, spreaded over the image intensity. It is bipolar in nature, meaning that the noise is realized as two defined values- light dot or dark dot. This noise is also known as shot or spike noise. Fig. 2.7. shows an example of salt and pepper noise over the popular image Lena. The probability density function of salt and pepper noise is given by Equation 2.5.

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$



Figure 2.7: Example of Salt and Pepper Noise. Left: Original Lena Image. Right: Salt and Pepper Added to Lena Image.

Speckle Noise

Speckle noise is a grainy salt and pepper noise usually generated by Synthetic Aperture Radar (SAR) or medical images. It is generally caused by the interaction of out of phase waves with a target [8, 9]. The mathematical formulation of speckle noise is given by Equation 2.6 where a_R and a_I are independent Gaussian with zero mean. An example of speckle noise is shown in Fig. 2.8.

$$p(z) = a_R + ja_I \quad (2.6)$$

Periodic Noise

Periodic noise is very common during image acquisition phase when there is electrical or electromechanical interference. In periodic noise, the image is seen to have certain repeated noise pattern as shown in Fig. 2.9. This noise follows simple periodic patterns such as *Sine* or *Cosine* as given in Equation 2.7, where x and y are discrete image dimensions and a and b are two constants.

$$p(z) = \sin\left(\frac{x}{a} + \frac{y}{b}\right) \quad (2.7)$$

2.2 Image Denoising

Numerous denoising techniques have been proposed over years. We can classify them in many ways. One way to classify the denoising methods is to separate them into *spatial denoising*



Figure 2.8: Example of Speckle Noise. Left: Original Lena Image. Right: Speckle Noise with $Variance = 0.05$.

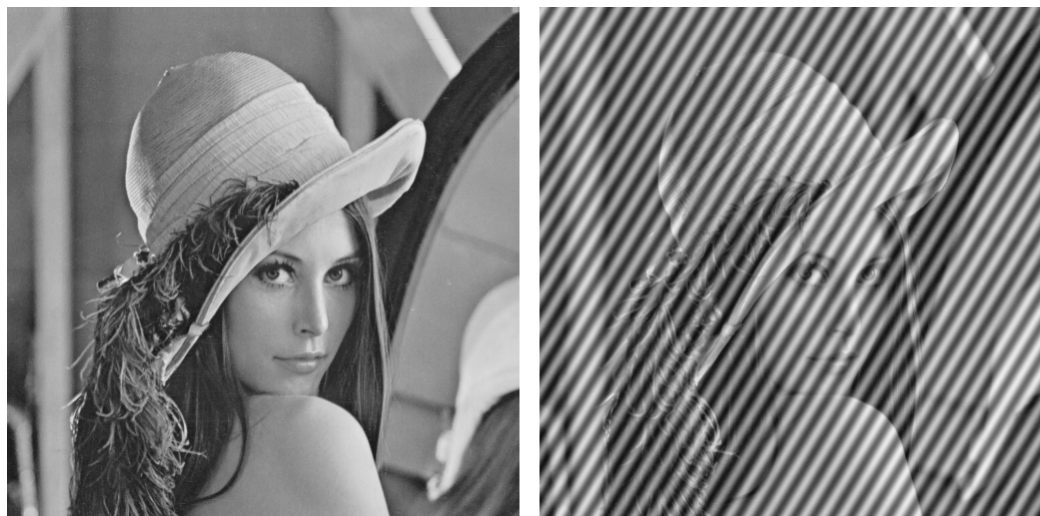


Figure 2.9: Example of Periodic Noise. Left: Original Lena Image. Right: Periodic Noise Added to Lena Image.

and *spectral (frequency) denoising*. Another way to classify them is to think as *additive noise filters* and *multiplicative noise filters*. A final approach to separate them is to classify as *edge preserving* and *non-edge preserving denoising*. We will start discussing the filters based on their working domain, i.e., whether they work in spatial domain or frequency domain. Then we will summarize them according to whether they preserve edges or not.

2.2.1 Spatial Domain Denoising

In this class of denoising, an image is denoised in spatial domain, based on the statistics of spatial information. Spatial domain denoising is better in the sense that it reduces the cost of domain transformation. However, since it is based on only spatial statistics, there is no other information considered. Therefore, there is always possibility to treat one intensity value in a wrong way.

Mean Filter

This is one of the simplest and oldest method of spatial denoising. The idea, as suggested by the name, is to replace a particular intensity value by the mean (arithmetic average) of all the intensity values within a given $n \times n$ kernel (window) centered at the pixel being denoised, where the typical value of n can be 3, 5 or 7. Mean filter smooths the image, i.e., reduces the variation of image intensities. This filter is very effective in reducing impulse noise.

There are many variations of mean filter present in the literature such as *Geometric Mean Filter*, *Harmonic Mean Filter* and *Contra-harmonic Mean Filter*. We will not cover them here, for further reading about these filters, we refer the reader to [6, 10].

Median Filter

Median filter is similar to Mean Filter, except that the pixel being denoised is replaced by the median value of the window pixels. Although similar, median filter is considered often better as it can preserve more image detail than mean filter. This is because, since it is not taking the arithmetic average of the kernel pixels, there is less probability that it will create unrealistic pixel values on the edges. It is better than Mean filter also in the sense that it will never be affected by a single unrepresentative pixel value.

Median filter belongs to a class of filters known as *Order Statistics Filters*. This is because, like Median, these filters attempt to reduce noise from spatial information based on some kind of ordering. Other filters of this class include *Min Filter*, *Max Filter* and *Midpoint Filter* [6].

Min and Max Filter

Min and Max filters are used to filter the *darkest* and *brightest* intensities of an image, thus reducing salt and pepper noise. Min and Max filters are given by Equations 2.8 and 2.9 respectively. In both equations, $g(s, t)$ is the window taken from noisy image with s and t being the window (local) co-ordinates.

$$\hat{f}(x, y) = \min\{g(s, t)\} \quad (2.8)$$

$$\hat{f}(x, y) = \max\{g(s, t)\} \quad (2.9)$$

Midpoint Filter

The Midpoint filter is best for reducing randomly distributed noise such as Gaussian noise. The midpoint filter finds the midpoint between the maximum and minimum intensity value within the given kernel or window. This filter is defined in Equation 2.10.

$$\hat{f}(x, y) = \frac{1}{2}[\max\{g(s, t)\} + \min\{g(s, t)\}] \quad (2.10)$$

Gaussian Smoothing

Gaussian smoothing is another popular spatial domain filter that is generally used to *blur* the image, thus removing the noise as well as the image details. Like Mean and Median filter, it also uses a kernel, however, it is different in the sense that the kernel is generated to represent a Gaussian bell shaped hump. Then kernel is convolved with the same size image block in order to perform the smoothing. The Gaussian smoothing function that is used to generate the kernel is given by Equation 2.11.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.11)$$

Conservative Smoothing

This technique is used to remove spikes or shot noise such as salt and pepper noise. The concept behind conservative smoothing is very simple, it considers that the noise should be of very high frequency as compared to the neighboring intensity values. Hence, it tries to provide a minimum and maximum bound for the pixels within a given window and then suppress any intensity value that goes beyond the upper and lower bound. Since it does not perform averaging or block-wise convolution, it preserves image details and edges really good.

Unsharp Filter

Unsharp filter is a sharpening operator in spatial domain. The main purpose of using an unsharp filter is to enhance edges and details. This filter achieves sharpening through the following operations.

First, this filter produces an edge image $g(x, y)$ from the input image $f(x, y)$ according to Equation 2.12. Prior doing so, it must also produce a smooth image using any smoothing operator.

$$g(x, y) = f(x, y) - f_{smooth}(x, y) \quad (2.12)$$

Finally, the sharp version f_{sharp} is generated using Equation 2.13. k is a scaling factor used in Equation 2.13, where a higher value indicates more sharpness. Typical values of k varies between 0.2 to 0.7.

$$f_{sharp}(x, y) = f(x, y) + k \times g(x, y) \quad (2.13)$$

Laplacian of Gaussian

Like unsharp filter, Laplacian operator (also known as Laplacian of Gaussian) is used to enhance edges rather than enhancing the whole image. It is often used for edge detection, however, this can also be studied in image denoising since it enhances edges and details.

Laplacian is defined as the 2nd derivative of the image intensity. It is defined as Equation 2.14. If we omit the y portion of this equation, it will give the edge information only along x axis and vice-versa.

$$L(x, y) = \frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2} \quad (2.14)$$

Because the digital image is a 2D discrete data, we should find a discrete approximation of the Laplacian operator in order to find the edges from an image or enhance the edges.

2.2.2 Frequency Domain Denoising

Spatial domain denoising tries to remove noise based on the spatial statistics. However, it is often easier to detect and remove noise if the intensity values are transferred to a different domain such as frequency domain. If a suitable kernel can be obtained, all frequency domain denoising can also be performed in spatial domain with less computational cost.

In frequency domain, both edge preserving and non edge preserving ideas can be implemented in the form of *low frequency attenuation (low pass filter)* and *high frequency attenuation (high pass filter)* respectively. In general, the frequency domain denoising follows Equation 2.15. $F(k, l)$ is the input image $f(x, y)$ transformed to frequency domain. $H(k, l)$ is the denoising filter which is multiplied with the transformed image $F(k, l)$. Finally, the operation produces the denoised image $G(k, l)$.

$$G(k, l) = F(k, l) \times H(k, l) \quad (2.15)$$

Low Pass Filter

Low pass filter is one of the basic frequency domain denoising filter. The idea is to let the low frequency values pass through the filter while attenuating the high frequency values. As a result, the final denoised image is smooth. It is equivalent to smoothing in spatial domain, since the high frequency means intensity change in spatial domain. Therefore, it is not edge preserving technique.

The simplest low pass filter is called *ideal low pass filter*. It hinders all high frequencies above a threshold called *cut-off frequency*. The ideal low pass filter can be thought as Equation 2.16.

$$H(k, l) = \begin{cases} 1 & \text{if } \sqrt{k^2 + l^2} < f_{cut-off} \\ 0 & \text{if } \sqrt{k^2 + l^2} > f_{cut-off} \end{cases} \quad (2.16)$$

High Pass Filter

High pass filter is similar to low pass filter, except that it attenuates low frequencies while passing high frequencies. Since flat region in the spatial domain corresponds to low frequencies in frequency domain, this filter preserves edges. Like low pass filter, this filter also has an ideal state which can also be described by the Equation 2.16 with the relational operators interchanged.

Band Pass Filter

A combination of low pass and high pass filter produce a band pass filter. Here, all high and low frequencies are attenuated and only a range of frequencies are allowed to pass through. Thus, it provides a band range for the image frequencies.

2.2.3 Edge Preserving Denoising

From the discussion of Section 2.2.1 and 2.2.2, we know that there are some denoising techniques in both spatial and frequency domain that preserve edges while the others are not sensitive to edges. For example- Median Filter, Conservative Filter, Unsharp Filter, Laplacian Filter and High Pass Filter are all edge preserving denoising techniques. In this section, we will discuss a more robust edge preserving denoising technique.

Anisotropic Diffusion

This idea was first suggested by Perona and Malik in [11]. We know that, in general, a noisy image is blurred using Gaussian Kernel to reduce noise from it. Typical techniques for noise reduction do so by isotropic diffusion, meaning that it does not preserve edges while blurring the noisy image. It just tries to average each pixel by its surrounding pixel values. As a result, the noises are removed but some edges are removed as well. Consequently, it becomes hard, if not impossible, to say where the true edges were actually located. For example, when we try to detect edges from a noisy image, many false edges (which are noise) are detected. If we blur (denoise) the image first and then try to detect edges, false edges are removed; however, some true edges are also removed. This is due to the generalized blurring applied to the whole image.

To resolve this issue, Perona-Malik suggests that any smoothing/blurring algorithm should have the following criteria-

1. Causality
2. Immediate Localization
3. Piecewise Smoothing

They propose an algorithm for denoising an image where the edges are not removed as in Gaussian as a result of blurring. They came up with two key ideas to propose such a smoothing function:

1. Using non linear smoothing instead of using linear smoothing/blurring
2. Using anisotropic diffusion instead of using isotropic diffusion.

Unlike linear smoothing, Non Linear Smoothing does not just treat each pixel with the same convolution. Rather, it tries to convolve each pixel with varying intensity. In other words, if a pixel is an edge-pixel, it applies little or no blurring and if it is not an edge-pixel, it applies more blurring. This idea removes the noises of intra-region i.e. removes noises from the image except edges. Simply stated, it blurs the whole image except the edges. However, since this process works only intra-region, not inter-region, edges stay rough as it was.

Later, to make the edges smooth, we have to apply Anisotropic Diffusion. In this diffusion technique, we are not directly taking the Laplacian of an image (divergence of gradient of the image). Instead, we are introducing a new function $g()$ multiplied by the gradient of the image and take the divergence of the result. The function $g()$ calculates the amount of diffusion to be applied on the edges. If it is a strong edge, it doesn't diffuse much. If it is a weak edge (or noise), it diffuses the edge.

To conclude, the idea proposed by Perona-Malik is ideal for noise reduction from images and better than the typical noise reduction methods that make use of blurring in order to reduce noises. In other words, this idea is ideal for detecting edges from noisy images.

2.2.4 Non Edge Preserving Denoising

There are many denoising methods that do not preserve edges. These methods are also referred to as *isotropic*, meaning that it treats all pixel values equally regardless of whether they belong to edge or area. As discussed earlier, *Gaussian Smoothing* and *Low Pass Filter* are two popular non edge preserving denoising methods.

2.3 Image Restoration

Image restoration is another area of image processing which is closely related to denoising. Image restoration and image denoising are similar in the sense that both try to produce better quality image from some degraded images. However, the difference is visible only when we understand how enhancement/denoising and restoration are performed in practice.

2.3.1 Degradation Model

The point that makes the big difference between image enhancement and restoration is the usage of an extra degradation function. In Sections 2.2.1 and 2.2.2 we have seen a number of spatial and frequency domain enhancement techniques. All of them could be computed without the help of outside knowledge. For example, to calculate mean, median or bandpass filter all we need to have is the noisy image only.

What if we try to denoise an image based on a mathematical model that gives us more information about how the image was degraded? Studies proved that having a mathematical model of how the image was degraded, we can design filters that can approximate an image which is more acceptable and less erroneous. This idea is what we define as *Image Restoration*.

2.3.2 Estimation of Degradation Function

We understand that a degradation function (often denoted by H in denoising/restoration literature) helps the process of quality enhancement improve significantly. But how can we estimate the degradation model? There are three possibilities available [6].

1. Estimation by Observation
2. Estimation by Experimentation
3. Estimation by Modeling

Estimation by observation assumes that no prior knowledge about degradation model is provided. Once the degraded image is received, it tries to estimate the degradation function from the image itself.

If we have access to the equipments that are used to acquire the degraded function, we can estimate an accurate replica of the degraded image. If we can acquire a close replica, then we can estimate a model of the degradation by comparing these two images. Estimation of degradation function through this process is called estimation by experimentation.

The most common practice in estimating degradation function is estimation by modeling. This is completely mathematical formulation that is derived assuming to solve some basic image acquisition problems such as motion blur created due to movement of camera between shutter open and closed. This method is most robust and reliable in the sense that the background behind it is purely mathematical.

2.3.3 Effect of Degradation Function on Image Restoration

Since there are different ways to estimate the degradation function, it is obvious that the degradation functions obtained through each of the discussed techniques for the same image will not provide same quality restoration.

In this subsection, we will see how different degradation function responds to image restoration process. The images present in this subsection are reproduced from the experiments conducted in [12].

Let us consider two images *cameraman* and *house* in Fig. 2.10 for our discussion. Let us also consider their corresponding power spectrum in Fig. 2.11. Our target is to restore the Cameraman image using these power spectrum. To do so, we will first *blur* the Cameraman image and then try to restore the image.

For restoration, we now have two estimates available- one is the power spectrum of cameraman itself and the other is the power spectrum of house image. Fig. 2.12 shows the cameraman image restored using the power spectrum of itself and Fig. 2.13 shows the restored cameraman using the power spectrum of house image.

Although there can be found no visual distinction between these two images with bare human eyes, the PSNR of the 2.12 is higher than the PSNR of 2.13. That is, we can say that- better the degradation function, better the restoration output.

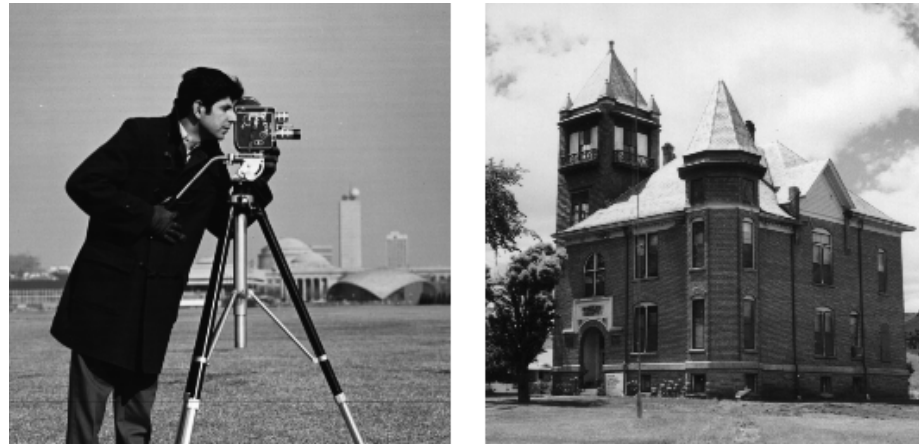
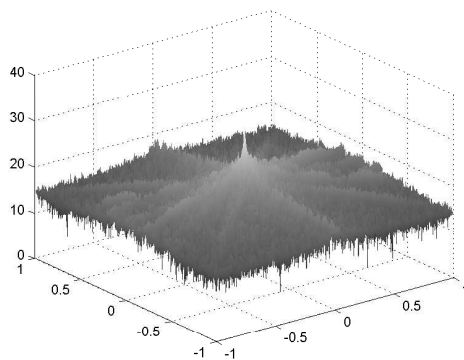
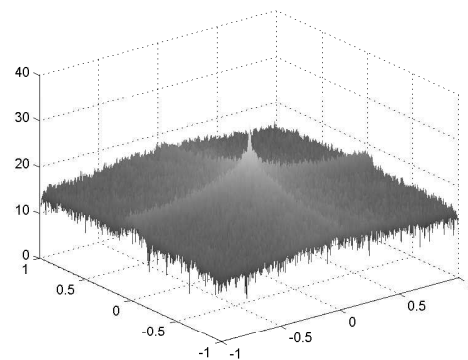


Figure 2.10: Example images. Left: Cameraman. Right: House



(a) Spectrum of Camera Image



(b) Spectrum of House Image

Figure 2.11: Spectrum of The Images of Fig. 2.10.

2.3.4 Techniques of Image Restoration

Generally, estimation by modeling can be used to restore a degraded image without any prior knowledge about the degradation function since it is based on some mathematical assumption. Also, it is not practical to estimate different degradation function for each image. Therefore, we will only focus on the degradation process that are obtained through mathematical modeling.

In this subsection, we will discuss two frequency domain filters that are commonly used for image restoration. The filters assume that the mathematical degradation model is available while applying them to restore images.



Figure 2.12: Restored Cameraman Image using Power Spectrum of Cameraman



Figure 2.13: Restored Cameraman Image using Power Spectrum of House

2.3.5 Inverse Filter

The inverse filter [13] computes an estimate of the original image in the transformed domain by dividing the transformed of the degraded image by the degradation function. Equation

2.17 shows the relation where $\hat{F}(u, v)$ is the Fourier transform of the input image $f(x, y)$ and $G(u, v)$ is the Fourier transform of the degraded image $g(x, y)$. As noted earlier, $H(u, v)$ is the degradation function. Examining the Equation 2.17, we realize that the inverse filter does not have provision to deal with noise. Rather, it only works with restoration.

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} \quad (2.17)$$

2.3.6 Wiener Filter

Wiener filter provides an opportunity to deal with both noises and degradation. This feature makes the Wiener filter unique in both image denoising and restoration.

Wiener filter is also called *Minimum Mean Square Error*. This is because the core idea behind Wiener filter is to satisfy an objective function which is the Mean Square Error (MSE). In other words, Wiener filter guarantees that the image restored by Wiener Filter \hat{f} will have minimum MSE with respect to original uncorrupted image f . Equation 2.18 shows that the expectation of Wiener filter is to have minimum MSE between f and \hat{f} .

$$e^2 = E(f - \hat{f})^2 \quad (2.18)$$

The Wiener filter is defined by Equation 2.19. Note that all terms are given in transformed domain. Here, $H(u, v)$ is the degradation function, $H^*(u, v)$ is conjugate complex of $H(u, v)$, S_n is the power spectrum of noise and is defined as $S_n = |N(u, v)|^2$, S_f is the power spectrum of undegraded image and is defined as $S_f = |F(u, v)|^2$. $G(u, v)$ is the transform of the degraded image and $\hat{F}(u, v)$ is the final estimate for the restored image. Once we make inverse transform of $\hat{F}(u, v)$, we will have $\hat{f}(x, y)$ which is the denoised/restored approximation for original image $f(x, y)$.

$$\hat{F}(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \frac{S_n}{S_f}} G(u, v) \quad (2.19)$$

From Equation 2.19, we can solve for $G(u, v)$ as in Equation 2.20 and rewrite Equation 2.20 as in equation 2.21. Now, if the noise is zero, the term inside square brackets in 2.21 becomes 1, which means the Wiener filter is reduced to Inverse filter and works for only restoration. However, if there is noise, Wiener filter incorporates itself for removal of noise along with restoration. This is what makes the Wiener filter unique.

$$G(u, v) = \frac{H^*(u, v) S_f(u, v)}{|H(u, v)|^2 S_f(u, v) + S_n(u, v)} \quad (2.20)$$

$$G(u, v) = \frac{1}{H(u, v)} \left[\frac{|H(u, v)|^2}{|H(u, v)|^2 + \frac{S_n(u, v)}{S_f(u, v)}} \right] \quad (2.21)$$

2.4 Block Matching and 3D (*BM3D*) Filtering based Image Denoising

In recent years, probably the most discussed denoising technique is *Block Matching and 3D Filtering (BM3D)* denoising technique [1]. It was first suggested by K. Dabov et al. in 2007. Later, it was rigorously reviewed in [4]. The idea has become extensively popular in denoising over last few years. *BM3D* achieves excellent performance for reducing AWGN. In this section, we will study *BM3D* and its possible variants. We will also present the persisting problems of *BM3D* and try to find if there is any solution to those problems.

2.4.1 Algorithm of *BM3D*

The discussion of *BM3D* should be better started with a step by step discussion of its algorithm. Let's start with Fig. 2.14 that shows the flow chart of *BM3D* [1, 14].

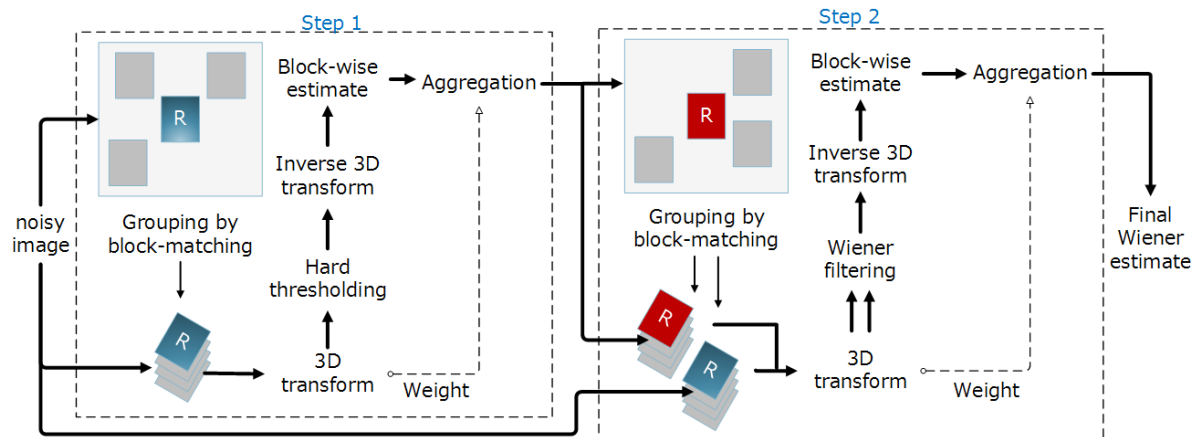


Figure 2.14: *BM3D* Block Diagram

BM3D takes the concept of Non-Local Means (NLM) published in 2005 [15] in the sense that it also attempts denoising based on finding similar patches within a given window. *BM3D* has two identical steps namely step 1 and step 2. They are identical in the sense that they have no operational difference, the difference lies in the component that are used for two steps. For example, first step uses hard thresholding while second step uses Wiener filtering. Other than that, both steps are same. *BM3D* basically tries to denoise the noisy image in the first step to generate a basic estimate. This basic estimate is used in Wiener filtering of the second step as an oracle (i.e. degradation model) [1].

BM3D First Step

In step one, *BM3D* starts its operation by dividing the noisy image into a number of blocks or patches. For each patch, it then generates a window centering the block being processed. *BM3D* defines this center patch as *reference patch*. Then within this window, *BM3D* looks for the patches similar to the reference patch. Usually, a good number of similar patches are found. *BM3D* defines a threshold that decides whether two patches are similar or not. Once

the similar patches are found, *BM3D* stacks all the similar patches together thus building a 3D block, where the first patch is the reference patch and others patches are sorted according to their distance to the reference patch. *BM3D* allows a number of 3D transform techniques to transform the domain from spatial to frequency (as indicated by *3D Transform* in Fig. 2.14). After the 3D transform is performed, the most important part of the first step, known as *hard thresholding*, is executed. Hard thresholding is nothing but a filter that allows all coefficients above a certain threshold and converts the remaining to zero. This is the only operation is first step that performs denoising, the rest are only to make a platform for hard thresholding. After hard thresholding, *BM3D* tries to generate a basic estimate. For this, it needs to transform the block coefficients to intensity values in spatial domain. This is known as inverse 3D transform. After performing inverse 3D transform what we get is the 3D block that we started working with. But this time it is denoised. Now, for each patch in the 3D block, we have to make an aggregation to have the estimation for the reference patch the block is containing. The aggregation part is simply taking different weights and making estimate for each pixel. Once the aggregation is done, we have our basic estimate to start the second step.

In theory, it is obvious that the more patches are present in the 3D block, the more estimates we will have for one single pixel and the more denoised basic estimate we will have. However, in practice, it is seen that after a certain number of similar patches, *BM3D* does not seem to perform better.

BM3D Second Step

The second step is similar to first step with two small differences. First, the 3D grouping is now performed on *basic estimate* that we obtained from first step, not on the *noisy image* as in step 1. Second, we are not using hard thresholding any more after 3D transform. Instead, we will now use Wiener filter. From the discussion of Section 2.3.4, we know that Wiener filter needs a degradation function $H()$ to work. In *BM3D*, the 3D group built on the basic estimate is considered as the degradation function for *BM3D* while the corresponding 3D group of noisy image is the degraded image function $G()$.

Equation 2.22 shows how the Wiener filter works in *BM3D*. Here, $\mathbb{P}^{basic}(P)(\xi)$ is the 3D block from the basic image and $\mathbb{P}(P)$ is the corresponding 3D block from the noisy image. τ_{3D}^{wien} denotes the 3D transformation for Wiener filter phase. Once we compute the inverse 3D transform of 2.22, we get $\mathbb{P}^{wien}(P)$ which is the final estimate for one block. Once we get the estimate, we perform a weighted aggregation operation like step one to start building the final denoised image.

$$\omega_p(\xi) = \frac{|\tau_{3D}^{wien}\mathbb{P}^{basic}(P)(\xi)|^2}{|\tau_{3D}^{wien}\mathbb{P}^{basic}(P)(\xi)|^2 + \sigma^2} \cdot \tau_{3D}^{wien}\mathbb{P}(P) \quad (2.22)$$

Comparing Equation 2.22 with Equation 2.19, we can see that both equations are exactly same except that Equation 2.22 works with 3D data.

2.4.2 Improvements over *BM3D*

BM3D achieves better Peak Signal to Noise Ratio (PSNR) (this measure is defined in Section 4.2.1) as compared to any existing image denoising methods. There is no denoising algorithm present in the literature that can beat *BM3D*. So, it is natural to think if there is any improvement possible over *BM3D*. In 2010, Chatterjee et al. [16] attempted to find the answer where they defined the lower bound for MSE and ultimately reported that still, improvements over *BM3D* is possible. In other words, there are denoising methods yet to be invented that will reach the lower bound and hence make denoising dead.

Harold et al. [17] came with the concept of learning based denoising and compared their results with *BM3D*. For some images, they achieved better PSNR as compared to *BM3D*, however, for many images they could not beat *BM3D*. They reported that if the image contains repetitive pattern *BM3D* gives extra ordinary performance. This is because, images containing texture or repetitive patterns should produce more similar patches, therefore, *BM3D* can produce more estimates and weights for each pixel resulting a clear improvement on the final estimate.

Zhang et al. [18] tried to select the similar patches based on the concept that initially they will choose all similar blocks based on MSE, then from those selected blocks they will only take those that have higher SSIM [19] with respect to the reference block. They could achieve some PSNR for higher noise levels only.

Dabov et al. [1], who proposed *BM3D* first, further extended their methods beyond only gray scale image denoising. They provided extension for color images [20], for gray and color video denoising [21]. They also proposed a new denoising concept named *BM4D* [22] where a 4D group is formed from 3D block of volumetric elements or voxels. They performed numerous experiments of their methods to compare with other existing denoising standards such as NLN.

Some studies as in [23] proposed completely new ideas those are patch based as in *BM3D* or Non-Local Means. However, they finally end up with lower performance as compared to *BM3D*. One came with the almost same idea as we are using, however, with the difference that they proposed a SSIM based Non-Local Means filter, not *BM3D* [24].

2.4.3 Persisting Limitations

We studied the *BM3D* image denoising in detail and discovered that there still persist a few limitations that can be taken into consideration. In this subsection, we will report those limitations and suggest possible solutions to them.

Poor Performance of DC-Only Profile

BM3D has several profiles based on the parameters chosen. One of them is *DC-Only* meaning that a 3D Discrete Cosine Transform (DCT) is used for the 3D transform part of *BM3D*. Since the DCT results in DC and AC coefficients in transform domain and the DC coefficient preserve the block information, the hard thresholding cannot be performed on DC coefficient. Again, beside preserving the block information, DC coefficient also preserves noise information from

the block, therefore the final outcome of DC-only profile is poorer than other (wavelet and color) profiles where a combination of 2D DCT and 1D Wavelet or 3D wavelet is used.

Choice of Wiener Filter

BM3D uses the benchmark denoising/restoration Wiener filter in second step. The time Wiener filter was introduced in literature, there was no better similarity measures available, therefore MSE was used as the objective function in Wiener filter. However, lately there are better similarity measures such as SSIM [19].

Parameterized Setup

Regardless the type of transform used, *BM3D* keeps fixed set of parameter. However, every transform has its own energy compaction technique. Therefore, using same parameterized setup for all types of transform should be taken into consideration of rigorous review. In the subsequent chapters, we will discuss the parameterized setup in more details.

Poor Performance for Higher Noise Levels

BM3D has poor response to images corrupted by higher level of noise as compared to its response to images corrupted by lower level of noise.

Poor Performance for Non Textured Images

For non textured images, *BM3D* has poor response as compared to its response to textured image. This is probably because if a noise is textured, there are a lot of repetitive patterns that help the *BM3D* find more similar blocks. However, for non textured images, there are less similar blocks, sometimes even not sufficient blocks. Therefore, *BM3D* responses poor for this case.

2.4.4 Possible Improvements

Based on the discussion of Section 2.4.3 and the review of [16], we can say that, still there is scope to improve *BM3D*. In this study, we will primarily focus on the poor performance of DC-Only profile and try to find a solution for this problem. We will also focus on the improvement of Wiener filter in order to eventually improve *BM3D* performance, regardless of the type of the transformation and profile. We will present the solutions to these problems and examine their potentials in the next chapter.

Chapter 3

Methodology

Having built a solid background of the noise types and the denoising methods in Chapter 2, we are now ready to explain our methodology in details. We start this chapter by discussing the persisting problems of Wiener filter. We then theoretically propose an improved Wiener filter. Afterwards, we explain the improvements we propose for *BM3D*. We suggest an improved Wiener filter and a 3D zigzag thresholding to do so.

3.1 Improvement Over Wiener Filter

From Section 2.3.6, we have come to know that Wiener filter is a nearly optimal frequency domain filter that is capable of achieving both image denoising and degradation. The only problem with this filter is that it requires a degradation function which is often unknown in the real world. In this section, we review Wiener filter rigorously, report the cause for its performance limitations and finally propose an improved Wiener filter.

3.1.1 MSE Optimized Wiener Filter

As stated in Section 2.3.6, Wiener filter tries to minimize its objective function shown in Equation 2.18 while denoising/restoring a degraded image. This function is also known as Mean Square Error (MSE) as defined in Equation 3.1.

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - \hat{I}(i, j)]^2 \quad (3.1)$$

In this equation, I and \hat{I} are considered as the true noise free image and the reconstructed (or denoised) image, respectively. Now, the less the difference between these two is, the closer the images are. Also, more closeness indicates more accurately denoised image. With this fundamental property, MSE has been being used as an image quality measurement metric since long [25].

Wiener filter (Equation 2.19) guarantees that the denoised/restored image is the closest image possible to the true undegraded image, since this filter is optimized for MSE. We might conclude at this point, because, in ideal case Wiener filter may reconstruct an image whose

MSE with respect to the true image is zero. That is, both images are exactly same! However, there are other sides of MSE, too that prevent Wiener filter from achieving more accurate and perfect results.

3.1.2 Structural Similarity (SSIM)

Wang et al. [19] showed that although MSE is a good quality measure, it is sometimes seriously misleading. This is because MSE does not consider anything other than the point-to-point distance [26]. For example, if we add a constant value to all the pixel value of an image just to increase its brightness, the images are *visually* exactly same, however, MSE still generates a huge error because of point-to-point distance measurement. Fig. 3.1 shows such an example of misleading of MSE measure.



Figure 3.1: Effect of MSE on Brightness Increase

In this figure, we used the well-known Lena image on the left and added a constant 30 to all of its pixel values in order to increase its brightness. The brightened image is shown in right side of the figure. Although there is no visual distortion in the image, the calculated MSE between them is $899.8470 \approx 900$. This value is not negligible, neither we can consider MSE as a true error measure.

Now let us also consider Fig. 3.2. This is similar to Fig. 3.1 except that now we have not just added a positive constant to all brightness level, instead, if the brightness value is below 150, we added -30 to it and $+30$ otherwise. Due to the square term present in Equation 3.1, MSE still generates the same error value (900) as in Fig. 3.1, however, visually the right side image of Fig. 3.2 is much distorted as compared to that of Fig. 3.1.

To avoid such misleading error measure, Wang et al. [19] proposed that modern image quality measurement metrics should not depend only on point-to-point distance, it should also consider the geometric or structural similarity between the images. Otherwise, there is possibility to have same error results for different images as discussed now. Thus they proposed a new



Figure 3.2: Effect of MSE on Brightness Change

error measurement metric called *Structural Similarity (SSIM)*. Their proposed error measure SSIM is defined in Equation 3.2.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3.2)$$

In this equation, x and y denote two blocks of the same position from the true image and the reconstructed image. μ_x and μ_y denote the arithmetic mean or average of x and y blocks, respectively. σ_x^2 and σ_y^2 indicate the variance of x and y blocks, respectively while σ_{xy} is the co-variance between x and y . c_1 and c_2 are two variables to stabilize the division with weak denominator and their values are calculated using Equations 3.3 and 3.4, respectively. Here, L is the dynamic range of the image and k_1 and k_2 are two constants whose values are $k_1 = 0.01$ and $k_2 = 0.03$, respectively [19, 26].

$$c_1 = (k_1L)^2 \quad (3.3)$$

$$c_2 = (k_2L)^2 \quad (3.4)$$

This measure is calculated block by block. Therefore, a mean of all blocks is used to represent the final SSIM index value for the whole image. The SSIM index generally varies between -1 to $+1$ which is often taken as absolute to avoid the negativity issue. Thus, the SSIM index is a fractional number between 0 and 1 (inclusive), where a 0 indicates *no similarity* and a 1 indicates *exact similarity* between two images. Using this measure, we have 0.9650 and 0.7850 for Fig. 3.1 and Fig. 3.2, respectively, which indicates that the right side image in Fig. 3.2 is much more distorted than that of Fig. 3.1. This distortion was overlooked by MSE.

There are other misleading measures of MSE, however, we will not cover them in this thesis. For a detail list of misleading characteristics of MSE, we refer the reader to [19], [26],

[27] and [28].

3.1.3 SSIM Optimized Wiener Filter

From the discussion of Section 3.1.2, it has been clear that the MSE is not good for assessing the closeness between two images, it is rather good for assessing the distance between them. Instead, the SSIM is a more acceptable alternative. This is because, MSE deals with image data while SSIM deals with image information. Now, if we have a better closeness measurement metric, why shouldn't we optimize the Wiener filter using this metric, instead of MSE?

The optimization of one denoising filter by SSIM instead of MSE is not very recent. Chanappayya et al. in [29, 30] showed that any linear filter can be optimized for SSIM. They compared their proposed SSIM optimized filter's result with MSE optimized Wiener filter. Their reported results showed that they were able to achieve higher SSIM value with respect to MSE optimized Wiener filter, however, their PSNR gain was still poor than MSE optimized Wiener filter. Therefore, there is scope to propose an SSIM optimized Wiener filter.

Our proposed SSIM optimized Wiener filter considers the structural similarity between the reconstructed image and the true image. Since higher SSIM index indicates more similar images, our goal is to estimate an image which has maximum possible SSIM. In this case, the expectation function of Wiener filter becomes Equation 3.5 which we now need to maximize. In this equation, E is the expectation and f and \hat{f} are the true and reconstructed images, respectively.

$$e = E\{sim(f, \hat{f})\} \quad (3.5)$$

Having defined the expectation function, our target is to ensure that our designed Wiener filter maximizes our expectation. Taking a careful look at Equation 2.19, we realize that replacing the term $\frac{S_n}{S_f}$ by a variable K is reasonable and finding a suitable value of K is possible [6]. Therefore, for our proposed case, we can start with the lowest possible value of K and loop through the highest possible value of K . For each K , we record the SSIM error measure in a vector and then restores the image using that K for which the error has been recorded maximum. Thus, for a given range of noise level, it is guaranteed that our proposed Wiener filter should be SSIM optimized. Likewise, it should also provide better denoising and restoration. As explained in Chapter 4, we will see that this idea in effect improves the Wiener filter's performance.

3.2 Improvement Over *BM3D*

From the discussion of Section 2.4.1, we know that *BM3D* works in two identical steps. Both the steps are similar except that in second step, *BM3D* uses Wiener filter instead of hard thresholding.

3.2.1 DC-Only Profile

The performance of *BM3D* varies depending on the type of transformation used in both steps. There are two major profiles of *BM3D*- *DC-Only profile* and *Wavelet profile*. In DC-Only

profile, *BM3D* uses three dimensional Discrete Cosine Transform as 3D transform. For the Wavelet profile, on the other hand, *BM3D* uses a combination of 2D Bi-orthogonal transform and 1D Haar or Walsh-Hadamard Transform. DC-Only profile generally produces poorer result as compared to its Wavelet counterpart [1, 31].

Discrete Cosine Transform preserves the average information in its DC Coefficient [32]. If the image is noisy, beside preserving the average image information, it also preserves the noise information in the DC Coefficient. Also, when the hard thresholding is performed, it is done on only AC Coefficients, not on the DC Coefficient in order to avoid average image information loss. Thus, DC Coefficient preserves some noise information along with the image information even after hard thresholding. This is why DC-Only profile performs poorer than the Wavelet profile.

We propose two different improvements over *BM3D*. First, we propose to replace the Wiener filter of *BM3D* by our already improved Wiener filter. This improvements will be used regardless of the type of profile used for *BM3D*. Second, we propose a 3D zigzag thresholding only for the DC-Only profile.

3D Zigzag Based Thresholding

Before starting our discussion of 3D zigzag thresholding, we would like to review the hard thresholding step of *BM3D*. This will help the reader understand the impact and technique of 3D zigzag thresholding.

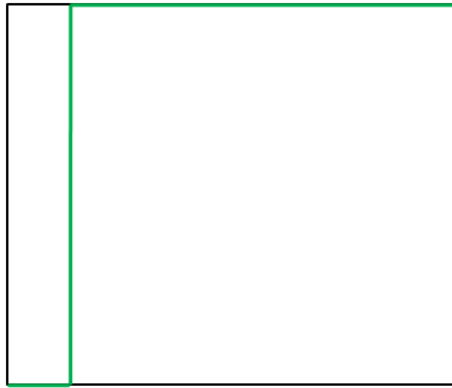


Figure 3.3: Hard Thresholding in DC-Only Profile

It is obvious that the surprisingly better result produced by *BM3D* is completely depends on a couple of things, including- (1) how accurately you can group the similar blocks, (2) how better the basic estimate is and (3) how you treat the Wiener filter. Also, the better estimate is generated in step one, the better oracle we have in step 2 for Wiener filter. The grouping technique of *BM3D* is good enough for achieving the collaborative filtering. Once *BM3D* calculates the similar patches with respect to a reference patch, it stacks them together such that they represent a 3D block. Now, these coefficients are collaboratively hard thresholded, meaning that coefficient values greater than λ_{3D} will be kept unchanged while coefficients less than λ_{3D} will be thresholded to 0 (zero). Here, λ_{3D} is the 3D hard thresholding operator. The value for this operator is empirically found by the authors of *BM3D* [1].

In Discrete Cosine Transform, the first coefficient is basically the average of all the pixel values within a given block [33]. Therefore, for an accurate inverse transform, the unchanged DC Coefficient is important. Therefore, BM3D does not apply hard thresholding on DC Coefficient. Instead, it applies hard thresholding on all AC-Coefficients. This *DC-skipping* hard thresholding is shown in Fig. 3.3. The first coefficient is unchanged (shown below in green color), then a sudden transition takes place and all other coefficients (AC) are treated in the same way.

Discrete Cosine Transform does not preserve its average block information in DC Coefficient only, it also preserves some of its information in the first few AC coefficients [6, 32, 33]. Therefore, thresholding all the AC Coefficients in the same manner means losing some 3D block information. In order to keep all the image information and just remove the noise, we should use a zigzag thresholding. Zigzag thresholding is realized by applying little or no thresholding on the DC and first few AC coefficients and then applying more thresholding on rest of the AC coefficients. Fig. 3.4 which is essentially a *gamma* (γ) curve will clear the idea.

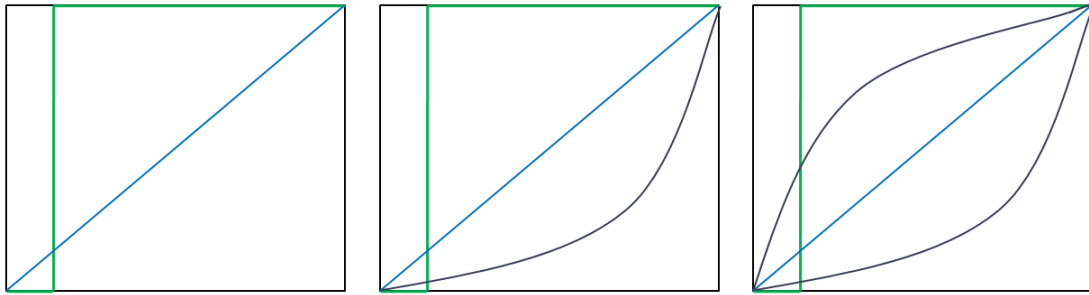


Figure 3.4: ZigZag Thresholding in DC-Only Profile

Here, like Fig. 3.3, we apply no thresholding on the DC (first) coefficient. Unlike Fig. 3.3 we do not perform a sharp transition. Instead, we linearly or exponentially increase the thresholding value. The left most curve in Fig. 3.4 shows a completely linear increase of thresholding value while the other two curves show exponential increases depending on the value of γ . A gamma curve is as simple as $y = x^\gamma$, if $\gamma = 1$, the y is linear and so on. The overall idea is to treating all the coefficients of the 3D block in such a way so that the first few coefficients of the block is less affected by the threshold while the latter coefficients are more affected. In Chapter 4, we will see how the value of γ affects the PSNR improvement.

We are defining this idea as *3D zigzag thresholding* because in DCT, the optimized way to traverse the coefficients is zigzag ordering as shown in Fig. 3.5. In our case, we follow the same order, just in 3D instead of 2D.

This idea of 3D zigzag thresholding will emphasize more on preserving image information and less sharp transition between the coefficients. In Chapter 4, we will see how this idea leads to an improved denoising performance in the DC-Only profile of *BM3D*.

3.2.2 Wavelet Profile

As stated earlier, Wavelet profile is a combination of 2D Bi-orthogonal transform and a 1D Haar or Walsh-Hadamard transform. This profile performs better than the DC-Only profile

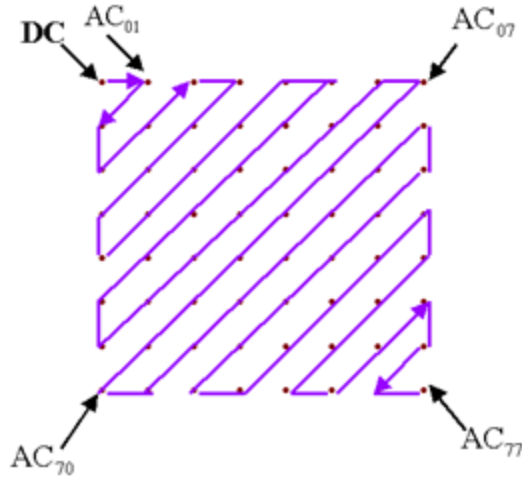


Figure 3.5: ZigZag Ordering

in terms of PSNR achievement. This profile has two sub-streams- *Normal Profile* and *Fast Profile*.

Normal Profile

The normal profile is realized by using a 2D Bi-orthogonal transform for the noise level $\sigma \leq 40$ with σ being the standard deviation of AWGN. For $\sigma > 40$, this profile uses a 2D-DCT. In both level of noise, it uses a 1D Haar transform in common. For this profile, we just propose to use our improved Wiener filter instead of the one being used by *BM3D*.

Fast Profile

This profile is named so because of its faster execution as compared to Normal Profile. The faster execution is achieved by using larger block size as compared to Normal Profile while finding similar blocks. Another difference between Fast Profile and Normal Profile is that fast profile does not use Discrete Cosine Transform if $\sigma > 40$. The fast profile compromises the PSNR gain with time. It also uses predictive searching instead of exhaustive searching as in normal profile. Again, we propose only our improved Wiener filter for this profile, no zigzag thresholding.

3.2.3 Color Profile

As the name suggests, it is used for color image denoising. We can easily extend our idea of gray-scale image denoising for color image denoising. Given an 24-bit true color *RGB* image, *BM3D* first converts it to *YUV* color space. The *YUV* color space is special color domain that separates the luminance (*Y*) and the chrominance (*U,V*) information [34]. The *Y* channel contains the valuable image information such as details, edges, shades, objects and texture patterns. On the other hand, the chrominance channels mostly contain low frequency image information such as color. Since *Y* contains the most valuable information, *BM3D* performs a

3D-grouping on Y channel and use the same grouping for U and V channels. Then it applies the gray-scale denoising algorithm separately on each channel. Finally, $BM3D$ computes an inverse color space transform from YUV to RGB .

For this profile, we only propose using our improved Wiener filter. This is because, this profile is exactly same as the Wavelet profile except that it first transforms and separates the color channels.

Chapter 4

Experimental Results and Analysis

In this chapter, we will discuss the performance and output we obtained from different experiments for our proposed methods. All the experiments were performed so that it truly reflects the methods and proposals mentioned in Chapter 3. We start with the performance measurement metrics we will use to judge our outputs and then discuss the effect of our proposed Wiener filter and proposed *BM3D*.

4.1 Data Set

For all the experiments performed in this thesis, we used the eight standard images shown in Fig. 4.1. The advantage of using these images is that they have some characteristics that we need for our experiments. For instance, *BM3D* exhibits slightly different behavior to the textured and non-textured images. For textured images like Baboon, Barbara and Peppers in Fig. 4.1, *BM3D* easily finds a lot of similar blocks which makes collaborative filtering easier for *BM3D*. On the other hand, when *BM3D* works with images that have sudden variation of gray levels or considerably less similar patterns or textures, it does not perform as good as it does for textured images. Therefore a combination of both textured and non-textured images is good for calculating the average results as the average result will indicate an overall usability of the propose methods.

4.2 Performance Measurement Metrics

The performance of our experiments are measured using both subjective and objective measures in order to avoid the confusion between these fidelity criteria [35]. Also, an objective measure gives us completely blind results which is good for understanding the differences between the outputs. However, at the end, human eyes are the ultimate users of denoised images, so we considered subjective measures as well.

4.2.1 Objective Fidelity Criteria

The objective fidelity criteria is defined as *blind quality assessment* since it just takes into account the mathematical formula and the numeric values of the image to be assessed. Therefore,



Figure 4.1: Test Image Set

it provides a strict measure regardless the type of noise contaminated the image and the type of image being denoised. There are some standard measurement metrics in this criteria.

Peak Signal to Noise Ratio (PSNR)

Our first objective measure Peak Signal to Noise Ratio (PSNR) is given by Equation 4.1 and is based on Equation 3.1. In this measure, a higher value indicates a better restored/denoised image. Here, MAX_I indicates the maximum intensity of the image. Note that, since the MSE is the core of the PSNR quality measure, we do not separately report the responses of MSE measures in this thesis.

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (4.1)$$

Structural Similarity (SSIM)

As discussed in Section 3.1.2, MSE is sometimes misleading so is the PSNR. This is because PSNR depends on the value of MSE. That is why, we also calculated SSIM for measuring the quality of our denoised images. The SSIM is given in Equation 3.2. As stated in Section 3.1.2, a higher value in SSIM indicates better denoising.

4.2.2 Subjective Fidelity Criteria

We also used subjective fidelity criteria, i.e., we relied not only on the mathematical measures, but also on how human eyes perceiving the denoising performance. For this measure, we will present an example for each type of denoising we propose for the reader's convenience. If the

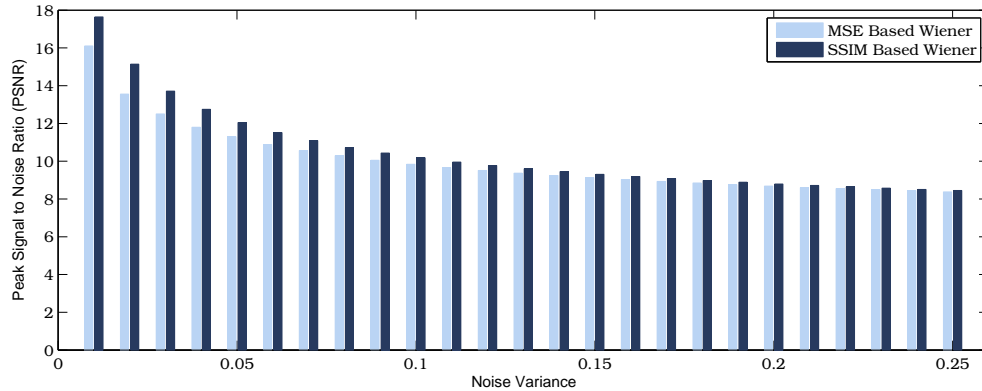


Figure 4.2: Average PSNR Comparison for Denoising

reader wishes to check the visual distinction of our denoising as compared to the existing methods, we refer the reader to Appendix A and Appendix B, in which we provide all experimental results and output images from our experiments.

4.3 Improved Wiener Filter for Image Denoising

To check the response of our proposed Wiener filter, we start with the noise free images. We add noise using Matlab function `imnoise` to those images. The `imnoise` function takes the noise variance as input. This function rescales the noise variance within a scale from 0.0 to 1.0. We used noise variance from 0.0 to 0.25 to generate the results of Section 4.3 and Section 4.4.

In this case, our obtained result is promising. For all the objective performance measurement metrics, we obtained better results. Fig. 4.2 and Fig. 4.3 respectively show the average PSNR and SSIM comparison of our proposed SSIM optimized Wiener filter with the MSE optimized Wiener filter. Clearly, the proposed method achieves consistent improvement over MSE optimized basic Wiener filter. All the obtained average results are given in Table 4.1 for reader's convenience. Note that, Wiener filter also takes a degradation function along with the noise signal. We used a Gaussian blurred function as standard degradation function in this experiment.

4.4 Improved Wiener Filter for Image Restoration

To observe the Wiener response for restoration, we take into consideration the images that are noisy as well as Gaussian blurred (degraded). The Gaussian blur is achieved by using Equation 2.11. We present the average PSNR and SSIM comparison for them in Fig. 4.4 and in Fig. 4.5, respectively. The data obtained from experiments to generate Fig. 4.4 and Fig. 4.5 are presented in Table 4.2.

It is clear from table 4.2 that our proposed Wiener filter is also capable of achieving better image restoration, not only image denoising. Now we can use this improved filter inside *BM3D* to observe its effect.

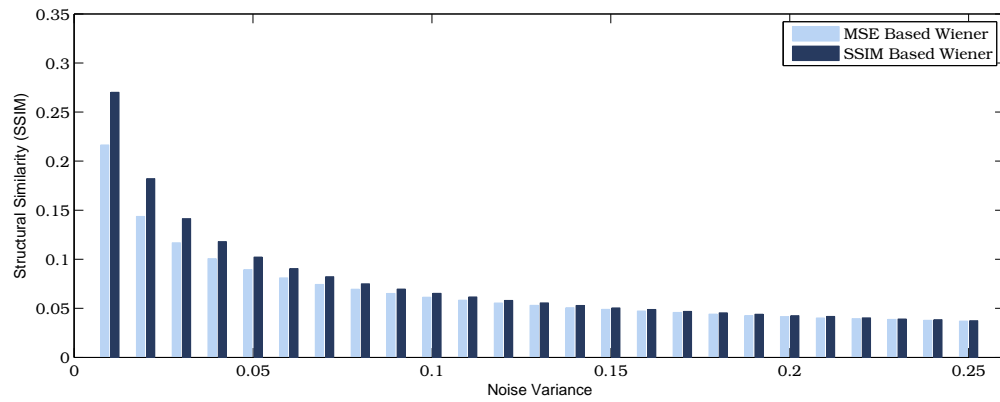


Figure 4.3: Average SSIM Comparison for Denoising

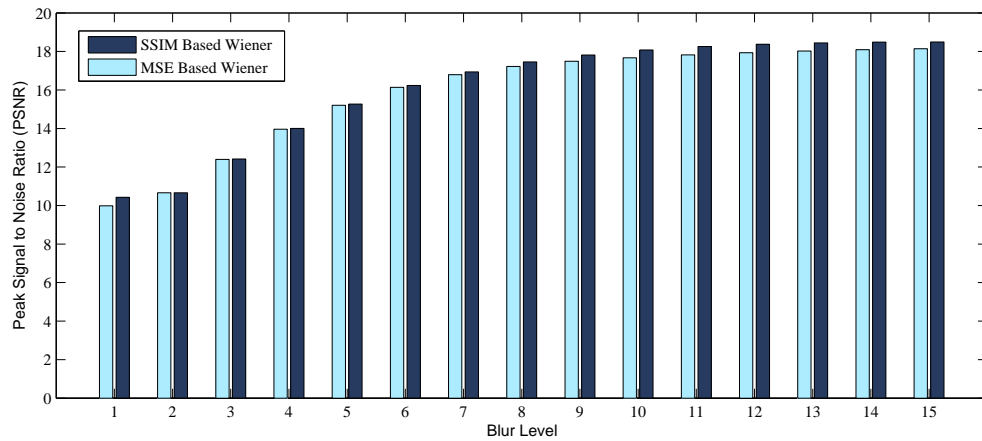


Figure 4.4: Average PSNR Comparison for Restoration

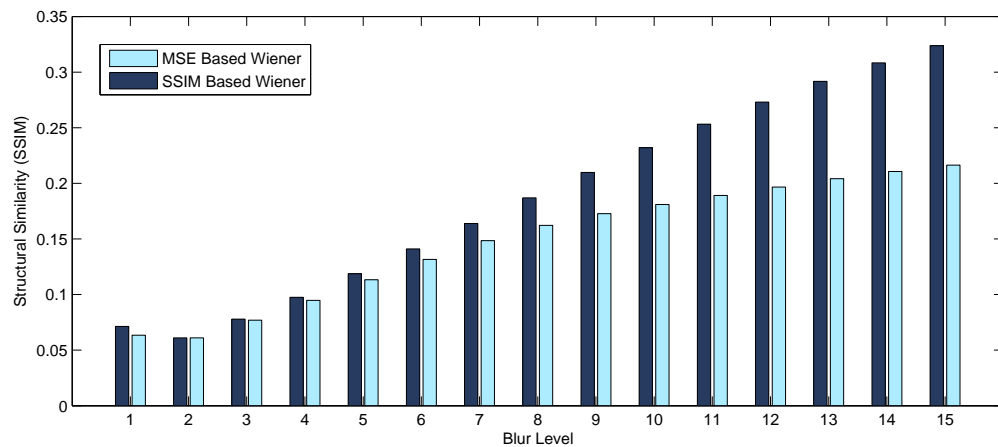


Figure 4.5: Average SSIM Comparison for Restoration

Noise Variance	MSE Based Wiener Filter		Proposed Wiener Filter		Gain	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.01	16.102	0.21644	17.639	0.27011	1.536	0.05367
0.02	13.552	0.14361	15.141	0.18221	1.589	0.03859
0.03	12.504	0.11680	13.712	0.14142	1.208	0.02462
0.04	11.799	0.10068	12.750	0.11803	0.951	0.01735
0.05	11.293	0.08943	12.047	0.10228	0.754	0.01284
0.06	10.886	0.08110	11.508	0.09057	0.622	0.00946
0.07	10.558	0.07429	11.085	0.08228	0.527	0.00799
0.08	10.288	0.06946	10.720	0.07504	0.432	0.00558
0.09	10.048	0.06514	10.430	0.06959	0.381	0.00445
0.10	9.840	0.06136	10.180	0.06524	0.340	0.00388
0.11	9.661	0.05828	9.955	0.06163	0.293	0.00335
0.12	9.508	0.05542	9.766	0.05802	0.257	0.00260
0.13	9.365	0.05304	9.599	0.05560	0.234	0.00256
0.14	9.240	0.05062	9.445	0.05294	0.204	0.00232
0.15	9.131	0.04895	9.304	0.05047	0.172	0.00151
0.16	9.028	0.04730	9.188	0.04878	0.160	0.00148
0.17	8.929	0.04581	9.078	0.04683	0.148	0.00102
0.18	8.849	0.04407	8.976	0.04543	0.126	0.00135
0.19	8.749	0.04266	8.887	0.04398	0.137	0.00131
0.20	8.682	0.04160	8.792	0.04258	0.110	0.00097
0.21	8.598	0.04013	8.715	0.04171	0.116	0.00158
0.22	8.544	0.03944	8.640	0.04027	0.096	0.00082
0.23	8.486	0.03871	8.571	0.03917	0.084	0.00046
0.24	8.428	0.03777	8.505	0.03841	0.076	0.00063
0.25	8.371	0.03694	8.447	0.03744	0.075	0.00049
Average	10.018	0.06796	10.443	0.07600	0.425	0.00804

Table 4.1: Average PSNR-SSIM Comparison of Wiener Filter for Denoising

4.5 Improved *BM3D* for DC-Only Profile

As discussed in Section 3.2.1, we know that the DC-Only profile performs poorer than the Wavelet profile because of the energy compaction technique of the Discrete Cosine Transform. In Chapter 3, we proposed two different solutions to solve the problem of this profile of *BM3D*. In this section, we will see the effects of those solutions.

4.5.1 Parameterized Setup for DC-Only Profile

To solve the problem of DC-Only profile, we considered the parameterized setup given in 4.3. The notations used are directly taken (except S_{step}^{ht}) from the original article [1]. For reader's convenience, we noted their meanings as well. The other parameters that are not present in table

Blur Level	MSE Based Wiener Filter		SSIM Based Wiener Filter	
	PSNR	SSIM	PSNR	SSIM
1	9.984	0.06348	10.425	0.07130
2	10.661	0.06104	10.665	0.06105
3	12.397	0.07690	12.417	0.07789
4	13.963	0.09472	14.002	0.09747
5	15.207	0.11335	15.269	0.11873
6	16.137	0.13162	16.232	0.14104
7	16.792	0.14841	16.941	0.16387
8	17.223	0.16225	17.455	0.18698
9	17.490	0.17274	17.817	0.20979
10	17.672	0.18095	18.076	0.23208
11	17.822	0.18907	18.255	0.25319
12	17.929	0.19660	18.372	0.27304
13	18.021	0.20417	18.443	0.29173
14	18.090	0.21064	18.484	0.30833
15	18.143	0.21640	18.495	0.32377
Average	15.835	0.148	16.090	0.187

Table 4.2: Average PSNR-SSIM Comparison of Wiener Filter for Restoration

4.3 are used with their default values as in [1]. In table 4.3, the values of S_{step}^{ht} indicate that we used *No Overlapping*, *50% Overlapping (Moderate Overlapping)* and *75% Overlapping (Maximum Overlapping)* respectively for our experiments.

Also, we didn't use the Matlab function `imnoise` to generate the noise for these experiments. Since we have to compare our results with the original *BM3D*, we also tried to follow the convention that *BM3D* followed. Therefore, we used the standard deviation $\sigma = 0$ to $\sigma = 100$ in Equation 2.2 to generate the noise. For simplicity, we discretized σ in ten different levels, starting from $\sigma = 10$ to $\sigma = 100$.

Notations	Meaning	Value
τ_{2D}^{ht}	2D Transform	2D-DCT
τ_{1D}	1D Transform	1D-DCT
N_1^{ht}	Patch Size	8
N_{step}^{ht}	Step Size for Moving Reference Patch	3
N_S^{ht}	Search Window Size	40
S_{step}^{ht}	Step Size for Moving Similar Patches	0,4,6

Table 4.3: Parameterized Setup for *BM3D* DC-Only Profile

4.5.2 With Improved Wiener Filter Only

First, we will present the DC-Only profile modified by our proposed Wiener filter only. That is, we just replace the Wiener filter of *BM3D* with our proposed Wiener filter. We compare our results with the DC-Only profile of *BM3D*. We used the same parameterized setup for both methods.

Non-Overlapping Similar Patches

The term *Non-Overlapping Similar Patches* means when we find the similar patches with respect to the reference patch, we do not overlap the similar patches. In other words, when we consider one patch as *similar* to reference patch, we make sure that no pixel value of this patch has even been considered for computation.

The use of our proposed Wiener filter was satisfactory in this case. The recorded responses show that we keep improving the *BM3D* performances as the noise level (σ) goes higher. Table 4.4 shows the detailed results obtained for this experiment.

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	24.62	24.62	0.999810	0.999964
20	23.38	24.62	0.999700	0.999947
30	22.67	24.59	0.999610	0.999913
40	22.16	24.52	0.999533	0.999878
50	21.77	24.37	0.999464	0.999844
60	21.46	24.10	0.999403	0.999808
70	21.21	23.75	0.999350	0.999777
80	20.97	23.31	0.999296	0.999743
90	20.80	22.81	0.999253	0.999713
100	20.61	22.35	0.999206	0.999681

Table 4.4: Performance Comparison of DC-Only Profile with No Overlapping using Improved Wiener Filter

It is clear from the table 4.4 that except for $\sigma = 10$ (where we have same PSNR as in *BM3D*), our proposed method improved the performance of *BM3D* considerably. For the easy visualization of the improvement, we have shown the PSNR and SSIM comparison of our proposed method with *BM3D* in Fig. 4.6 and Fig. 4.7, respectively.

Fig. 4.8 shows a subjective measurement for the test image Lena. Here, the noise is $\sigma = 50$. It is seen that *BM3D* output is more blurry whereas the proposed method is much better detailed preserving.

Moderate (50%) Overlapping Similar Patches

In this case, when we move the patch in a sliding window manner in order to find the similar patch with respect to the reference patch, we take half of the patch into consideration that has already been taken as similar patch. The important observation in this case is that our proposed

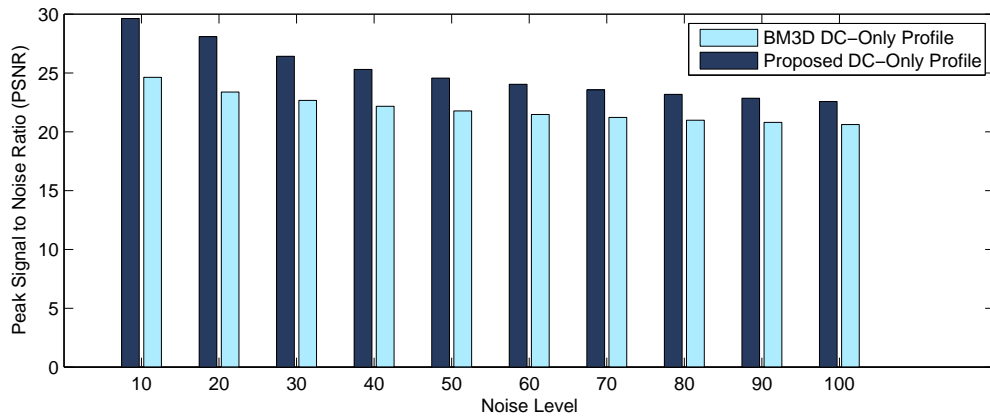


Figure 4.6: Average PSNR Comparison with No Overlapping for DC-Only Profile

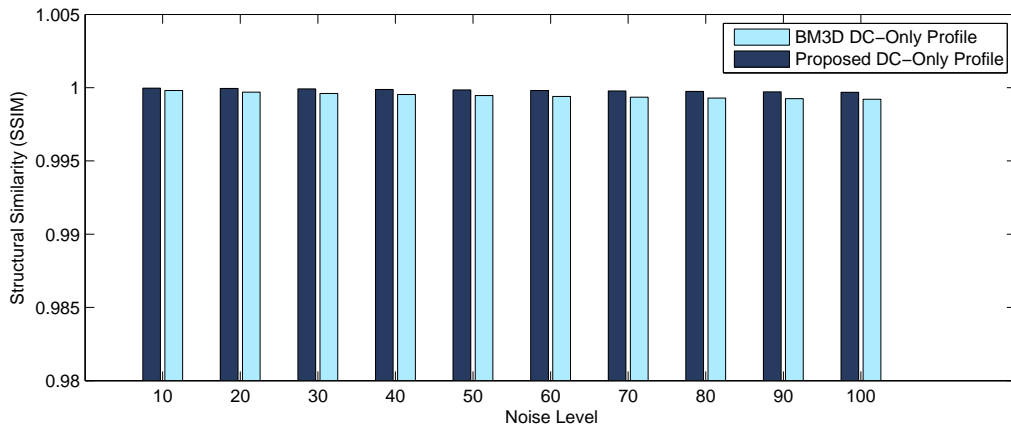


Figure 4.7: Average SSIM Comparison with No Overlapping for DC-Only Profile

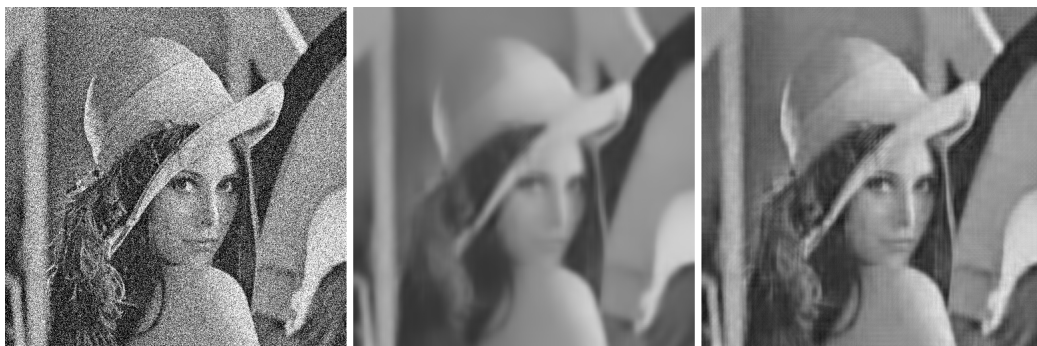


Figure 4.8: Subjective Comparison of DC-Only Profile with No Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method’s Output.

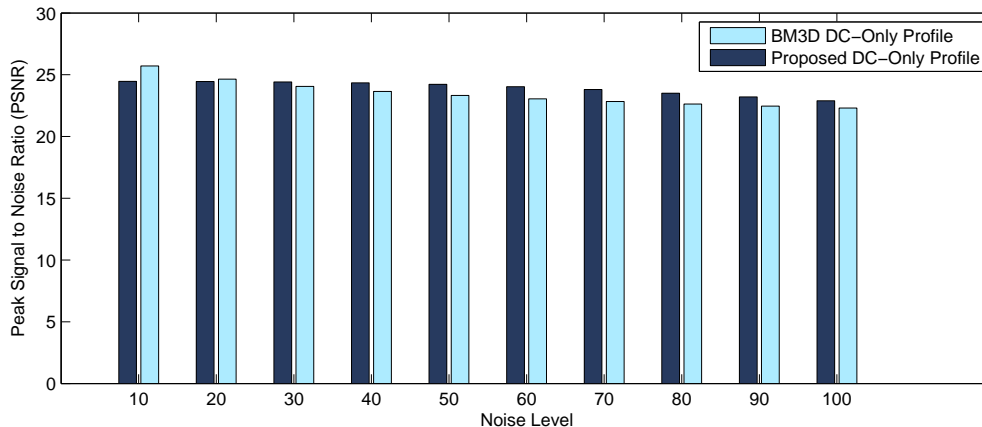


Figure 4.9: Average PSNR Comparison with 50% Overlapping for DC-Only Profile

method works better only for higher noise levels. The detailed experimental results are shown in table 4.5. The comparison curves for this case are shown in Fig. 4.9 and Fig. 4.10. We also present a subjective measure in Fig. 4.11.

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	25.71	24.47	0.999879	0.999810
20	24.65	24.45	0.999821	0.999808
30	24.06	24.42	0.999776	0.999806
40	23.65	24.34	0.999737	0.999800
50	23.33	24.22	0.999701	0.999791
60	23.05	24.03	0.999669	0.999778
70	22.83	23.80	0.999639	0.999763
80	22.63	23.51	0.999612	0.999743
90	22.46	23.20	0.999587	0.999719
100	22.31	22.89	0.999563	0.999694

Table 4.5: Performance Comparison of DC-Only Profile with 50% Overlapping using Improved Wiener Filter

Apparently, from the observation of Fig. 4.8 and Fig. 4.11, we can say that *BM3D* performance improves as we take overlapping similar blocks. However, at the same time, our proposed method gives poorer improvement over *BM3D*. We can see from tables 4.5 and 4.4 that our proposed method gives little or no improvement when overlapping is in effect. Otherwise, our proposed method outperforms *BM3D*.

Maximum (75%) Overlapping Similar Patches

In this case, we overlap the similar patch finding by 75% i.e., only two pixels are left when we take the second patch to find the similarity with respect to the reference patch. Interestingly, we could not outperform *BM3D* in this case. This experiment indicates that the DC-Only profile

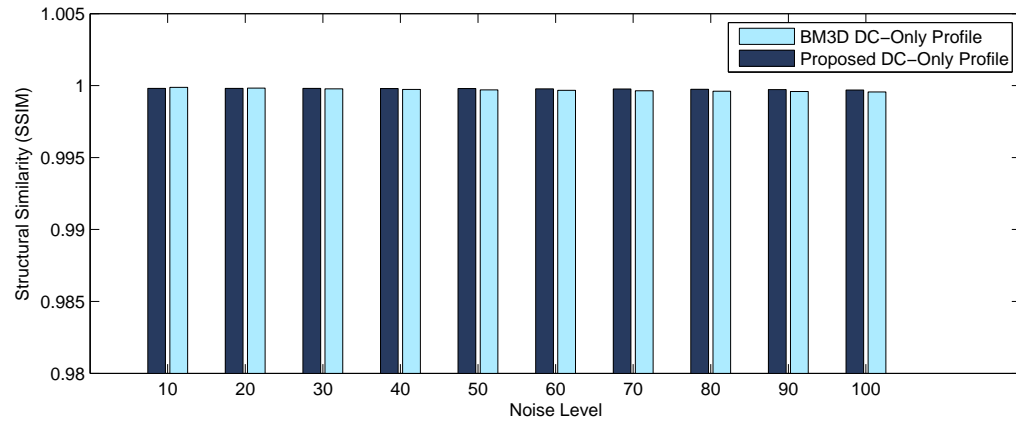


Figure 4.10: Average SSIM Comparison with 50% Overlapping for DC-Only Profile



Figure 4.11: Subjective Comparison of DC-Only Profile with 50% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.

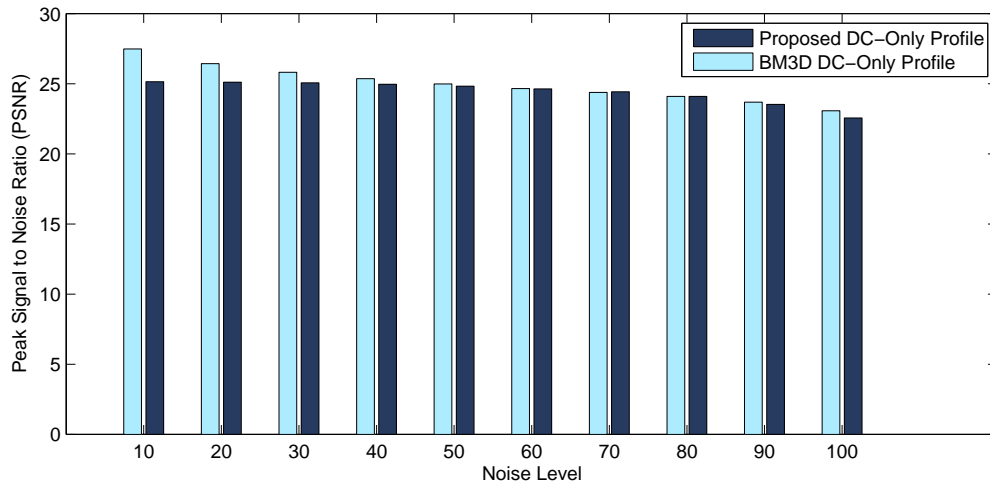


Figure 4.12: Average PSNR Comparison with 75% Overlapping for DC-Only Profile

of *BM3D* works best when there are maximum overlapping. Table 4.6 shows the data obtained for this experiment.

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	27.48	25.14	0.999939	0.999851
20	26.43	25.11	0.999910	0.999849
30	25.81	25.06	0.999884	0.999845
40	25.36	24.96	0.999860	0.999837
50	24.98	24.82	0.999836	0.999826
60	24.65	24.63	0.999813	0.999812
70	24.38	24.42	0.999792	0.999795
80	24.09	24.09	0.999769	0.999772
90	23.68	23.53	0.999742	0.999741
100	23.07	22.56	0.999708	0.999694

Table 4.6: Performance Comparison of DC-Only Profile with 75% Overlapping using Improved Wiener Filter

Fig. 4.12, Fig. 4.13 and Fig. 4.14 show the comparison curves and the subjective measures for this experiment.

To conclude this subsection, we will note that although increasing the overlapping amount improves the performance of *BM3D*, for real time execution, we can not take maximum overlapping for this profile. In terms of time, no overlapping is the best. In this case, our proposed method outperforms *BM3D*. For the average case, we can take 50% overlapping. Still, our proposed method outperforms *BM3D*.

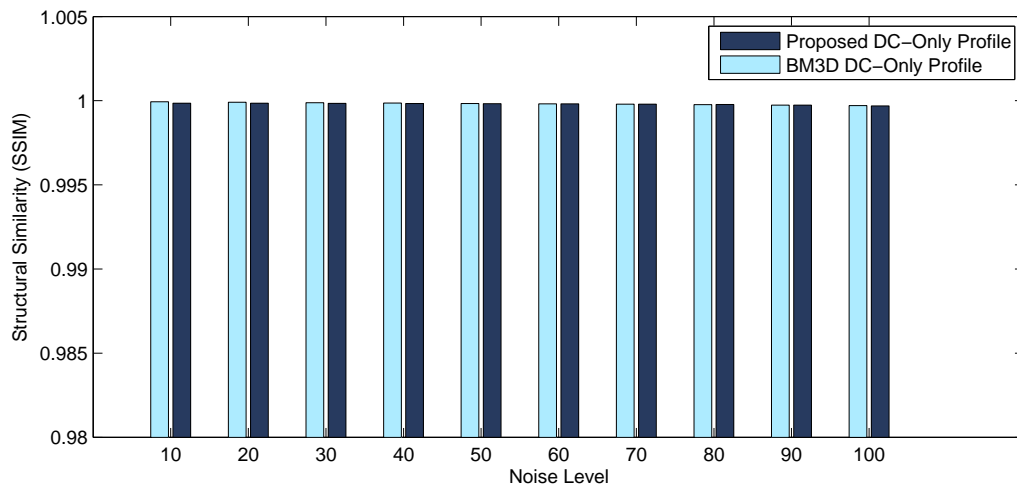


Figure 4.13: Average SSIM Comparison with 75% Overlapping for DC-Only Profile



Figure 4.14: Subjective Comparison of DC-Only Profile with 50% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.

4.5.3 With 3D Zigzag Thresholding Only

As discussed in Section 3.2.1, we are also interested about applying a 3D zigzag thresholding instead of applying the same threshold for all AC coefficients. We have already discussed that when applying 3D zigzag thresholding, we are basically dealing with a gamma curve ($y = x^\gamma$), where the value of γ will affect the output. In this subsection, we will present results only for $\gamma = 1$. Later, in Section 4.5.5, we will discuss the effect of changing γ .

Non-Overlapping Similar Patches

For this experiment, our obtained results are summarized in table 4.7. It is apparent from the results that the idea of using 3D zigzag thresholding does not improve the performance. This is because, when we take less overlapping blocks, we are eventually having less number of estimates for each pixel. In addition, using a linear zigzag thresholding means applying lower threshold value to the DC and first few AC coefficients that inherently hold the noise along with the image information. However, we will notice later that using more overlapping while finding similar blocks will resolve this issue. The comparison curves for this case are given in Fig. 4.15 and Fig. 4.16, respectively. For the subjective measure, we present Fig. 4.17.

Noise Level	BM3D PSNR	3D-Zigzag PSNR	BM3D SSIM	3D-ZigZag SSIM
10	24.63	24.19	0.999811	0.999770
20	23.38	23.04	0.999700	0.999654
30	22.69	22.38	0.999610	0.999563
40	22.16	21.91	0.999532	0.999486
50	21.77	21.56	0.999463	0.999419
60	21.47	21.28	0.999404	0.999362
70	21.20	21.04	0.999347	0.999307
80	20.99	20.85	0.999299	0.999263
90	20.81	20.69	0.999255	0.999223
100	20.63	20.53	0.999210	0.999181

Table 4.7: Performance Comparison of DC-Only Profile with No Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)

Moderate (50%) Overlapping Similar Blocks

The results obtained for this section are summarized in table 4.8. Like other subsections of this chapter, comparison curves and subjective measures are given in Fig. 4.18, Fig. 4.19 and Fig. 4.20. We can observe that in this case our proposed method is achieving better performance as compared to *BM3D*.

Maximum (75%) Overlapping Similar Blocks

The results of this section (presented in table 4.9) will lead to an important decision- should we always use the 3D zigzag thresholding regardless of the noise level? The answer is *NO*.

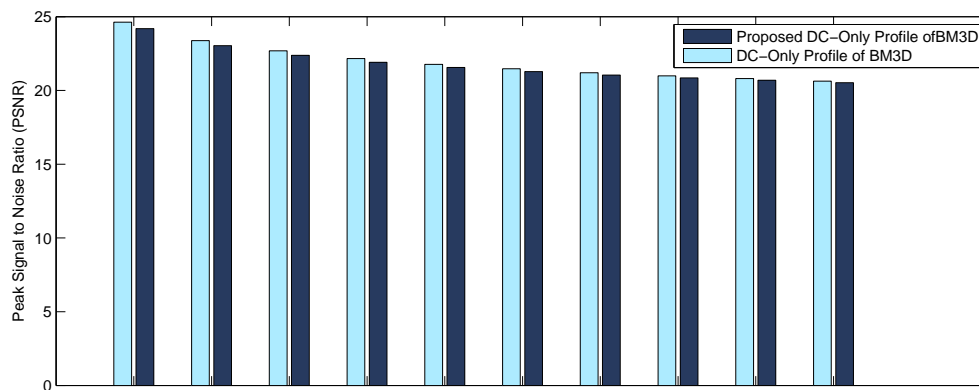


Figure 4.15: Average PSNR Comparison of DC-Only Profile with No Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)

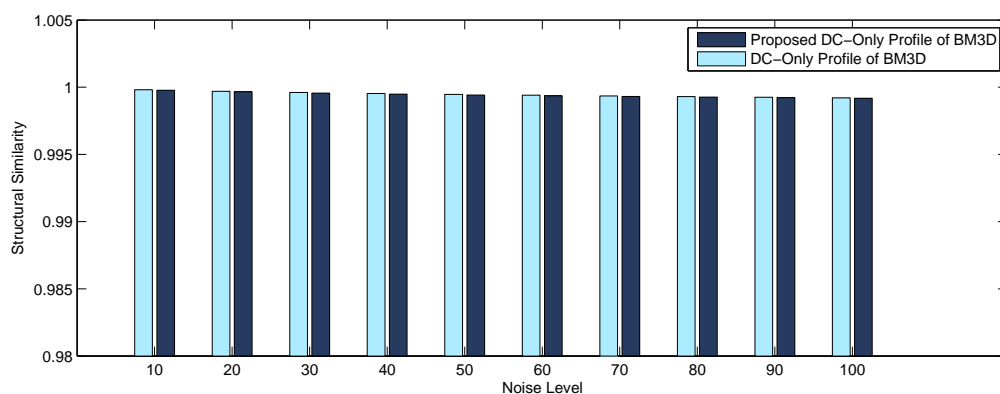


Figure 4.16: Average SSIM Comparison of DC-Only Profile with No Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)

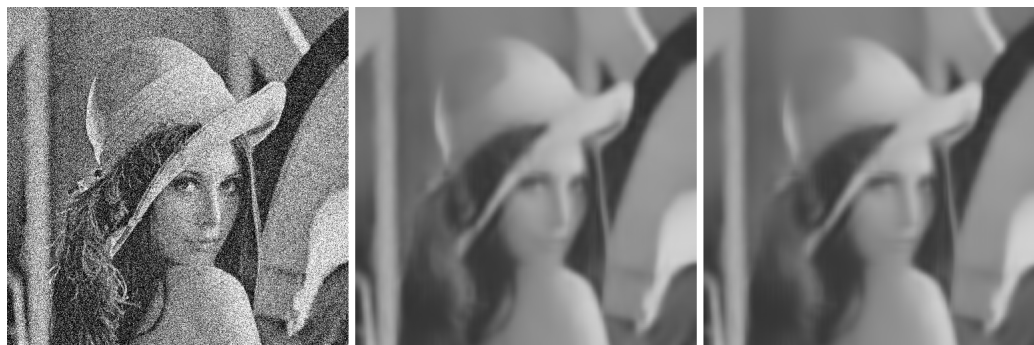


Figure 4.17: Subjective Comparison of DC-Only Profile with No Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.

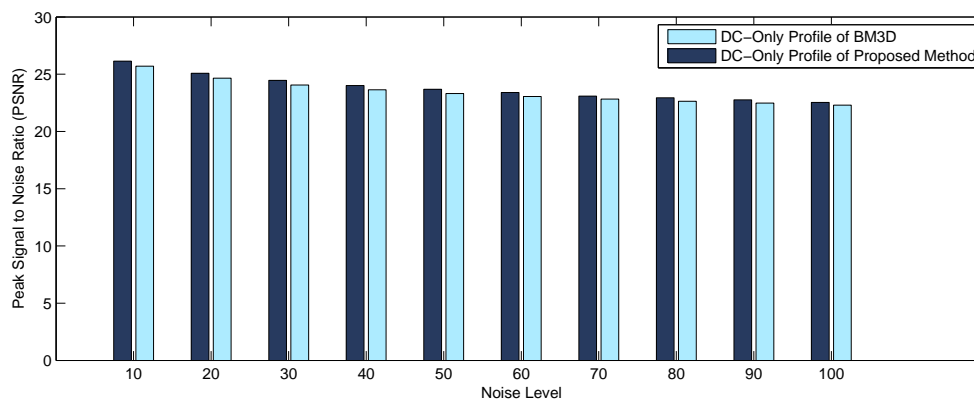


Figure 4.18: Average PSNR Comparison of DC-Only Profile with 50% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)

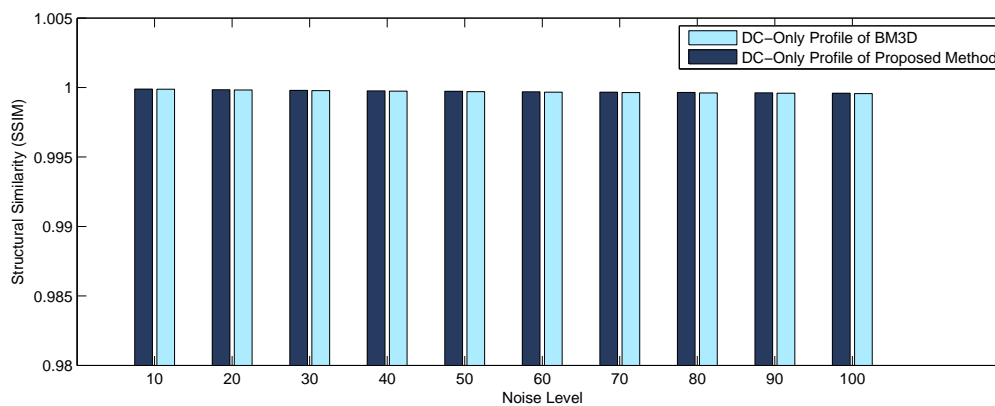


Figure 4.19: Average SSIM Comparison of DC-Only Profile with 50% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)

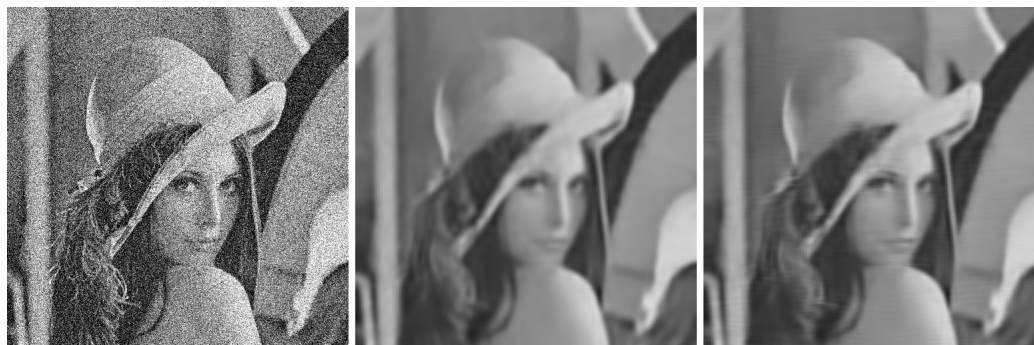


Figure 4.20: Subjective Comparison of DC-Only Profile with 50% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.

Noise Level	BM3D PSNR	3D-Zigzag PSNR	BM3D SSIM	3D-ZigZag SSIM
10	25.70	26.14	0.999879	0.999891
20	24.65	25.08	0.999821	0.999839
30	24.06	24.46	0.999775	0.999796
40	23.64	24.01	0.999735	0.999760
50	23.31	23.68	0.999700	0.999727
60	23.05	23.40	0.999669	0.999697
70	22.83	23.09	0.999639	0.999669
80	22.63	22.94	0.999611	0.999643
90	22.48	22.76	0.999589	0.999622
100	22.30	22.53	0.999562	0.999594

Table 4.8: Performance Comparison of DC-Only Profile with 50% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)

This is because, with the higher level of noise, the DC coefficient and first few AC coefficients will keep preserving more noise information along with the original image information. So, using lower threshold values for them means keeping the noises as well. Therefore, when the noise level is higher, we should not use the 3D zigzag thresholding. Instead, we should use our proposed improved Wiener filter only to achieve better performance. The experimental results shown in Fig. 4.21 and Fig. 4.22 show that after $\sigma = 50$, we should stop using 3D zigzag thresholding and use the improved Wiener filter only. The subjective measure for this portion of experiment is given in Fig. 4.23.

Noise Level	BM3D PSNR	3D-Zigzag PSNR	BM3D SSIM	3D-ZigZag SSIM
10	28.78	30.19	0.999954	0.999972
20	27.68	28.93	0.999928	0.999944
30	27.03	27.99	0.999905	0.999930
40	26.53	27.07	0.999883	0.999904
50	26.04	26.06	0.999858	0.999871
60	25.69	24.99	0.999837	0.999834
70	25.32	23.79	0.999813	0.999785
80	24.93	22.55	0.999788	0.999726
90	24.46	21.32	0.999760	0.999654
100	23.75	20.06	0.999723	0.999562

Table 4.9: Performance Comparison of DC-Only Profile with 75% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)

4.5.4 With Improved Wiener Filter and 3D Zigzag Thresholding

Once we observed the effect of our proposed improved Wiener filter and 3D zigzag thresholding separately in *BM3D*, we are now ready to put them together. Note that, the proposed

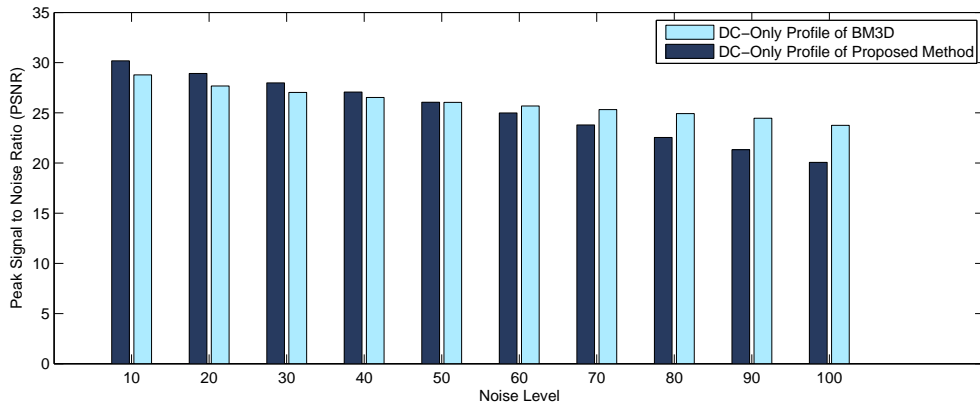


Figure 4.21: Average PSNR Comparison of DC-Only Profile with 75% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)

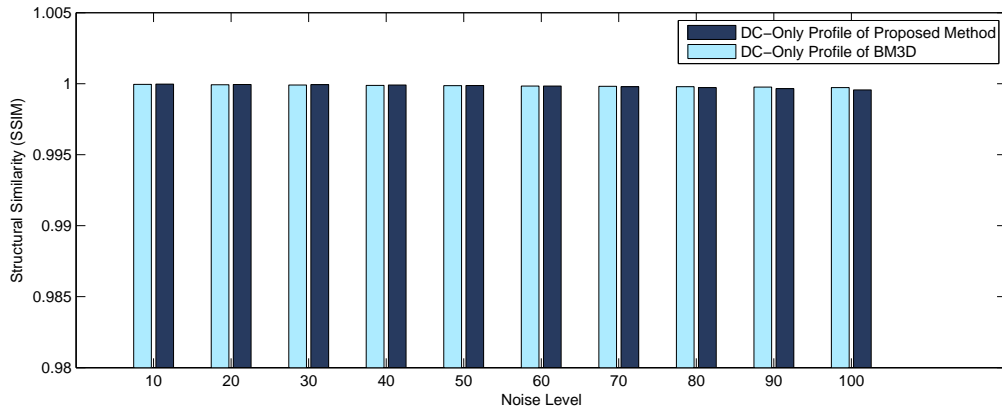


Figure 4.22: Average SSIM Comparison of DC-Only Profile with 75% Overlapping using 3D Zigzag Thresholding ($\gamma = 1$)



Figure 4.23: Subjective Comparison of DC-Only Profile with 75% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.

Wiener filter should affect the performance of second step of *BM3D* only while the 3D zigzag thresholding should affect the first step. Also, we have already observed that due to its noise preserving nature, the proposed 3D zigzag thresholding should not be used for higher noise levels. If used, it will eventually deteriorate the performance of second step too since the output of first step is used as an oracle to the Wiener filter in second step.

Non-Overlapping Similar Patches

The results obtained for this section are given in table 4.10. Clearly we have improved the performance of DC-Only profile of *BM3D*. The performance comparison curves present in Fig. 4.24 and Fig. 4.25 visually show the improvement achieved in our proposed case. The subjective measure given in Fig. 4.26 shows that the visual improvement is notable.

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	24.33	23.22	0.999803	0.999694
20	23.14	23.21	0.999691	0.999693
30	22.47	23.22	0.999602	0.999693
40	22.00	23.26	0.999528	0.999696
50	21.64	23.33	0.999461	0.999701
60	21.35	23.36	0.999401	0.999702
70	21.11	23.35	0.999351	0.999703
80	20.91	23.29	0.999302	0.999700
90	20.73	23.04	0.999258	0.999687
100	20.58	22.52	0.999218	0.999661

Table 4.10: Performance Comparison of DC-Only Profile with No Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)

Moderate (50%) Overlapping Similar Blocks

The results obtained are summarized in table 4.11. The comparison curves are given in Fig. 4.27 and Fig. 4.28. For visual inspection of how we improved *BM3D*, we present the subjective measure in Fig. 4.29.

Maximum (75%) Overlapping Similar Blocks

The final experiments we made for DC-Only profile of *BM3D* is by making 75% overlapping while searching similar patches. The results are given in table 4.12. Since we didn't stop using the 3D zigzag thresholding after $\sigma = 50$, a sudden performance drop after $\sigma = 50$ is noticeable.

We present the performance comparison curves for PSNR and SSIM quality measures in Fig. 4.30 and Fig. 4.31 respectively. The visual inspection on the proposed method can be made from Fig. 4.32.

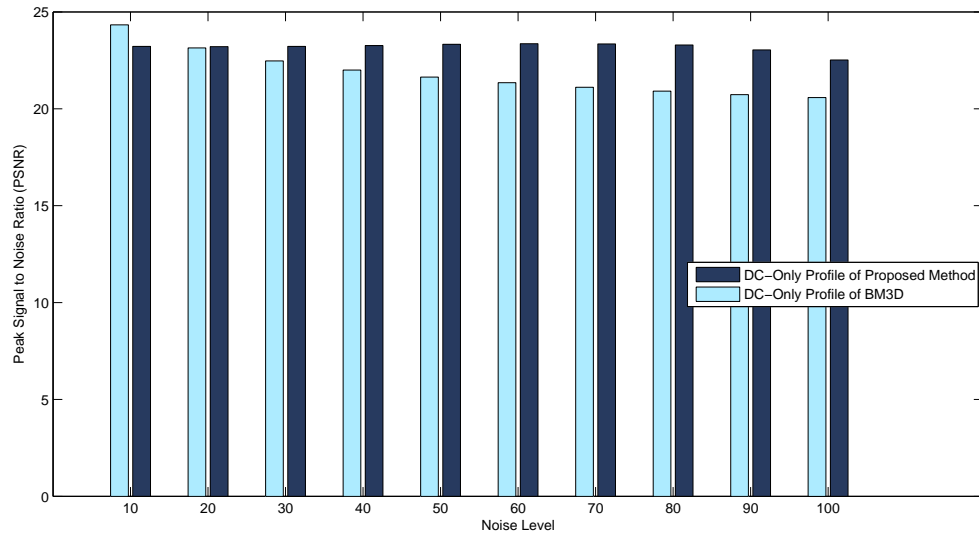


Figure 4.24: Average PSNR Comparison of DC-Only Profile with Non Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)

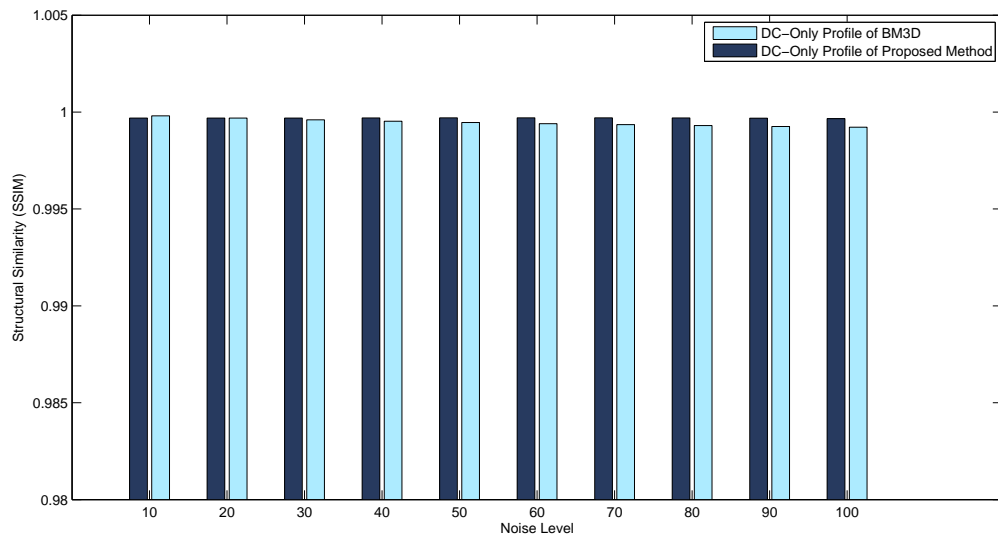


Figure 4.25: Average SSIM Comparison of DC-Only Profile with Non Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)

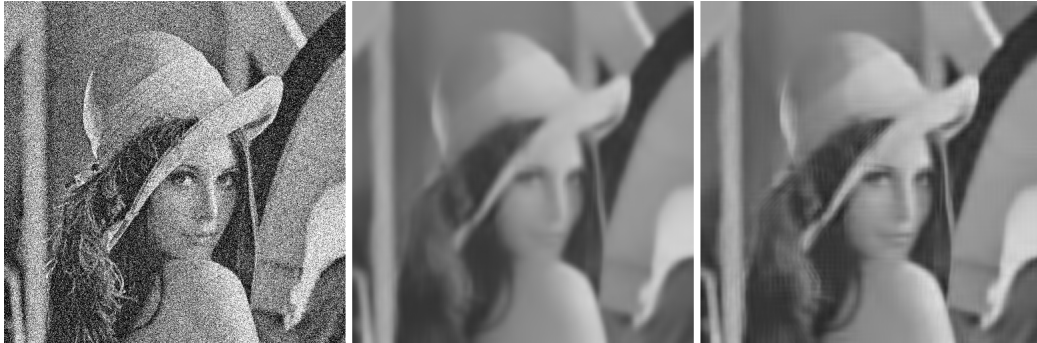


Figure 4.26: Subjective Comparison of DC-Only Profile with Non Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.

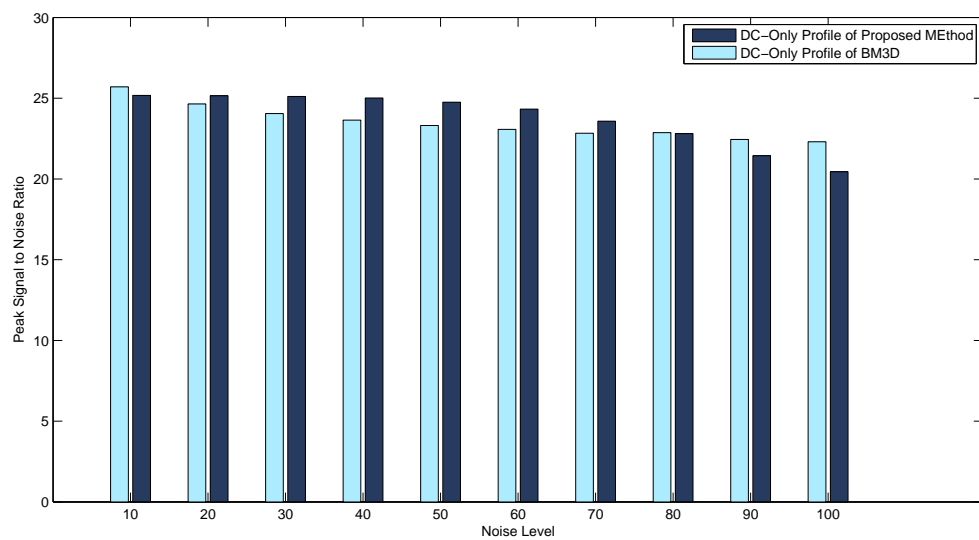


Figure 4.27: Average PSNR Comparison of DC-Only Profile with 50% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)

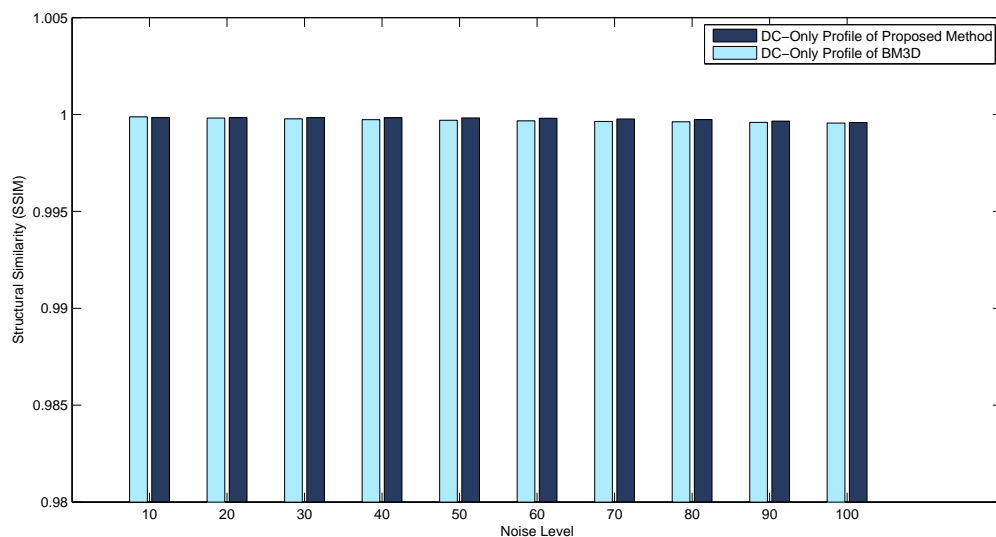


Figure 4.28: Average SSIM Comparison of DC-Only Profile with 50% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)

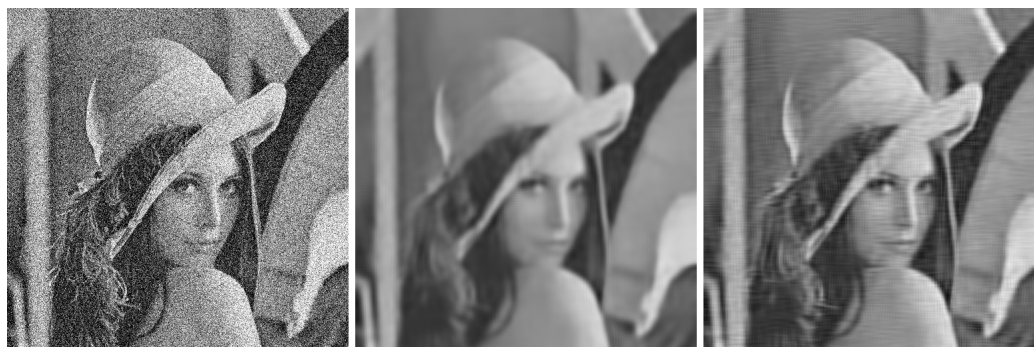


Figure 4.29: Subjective Comparison of DC-Only Profile with 50% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method's Output.

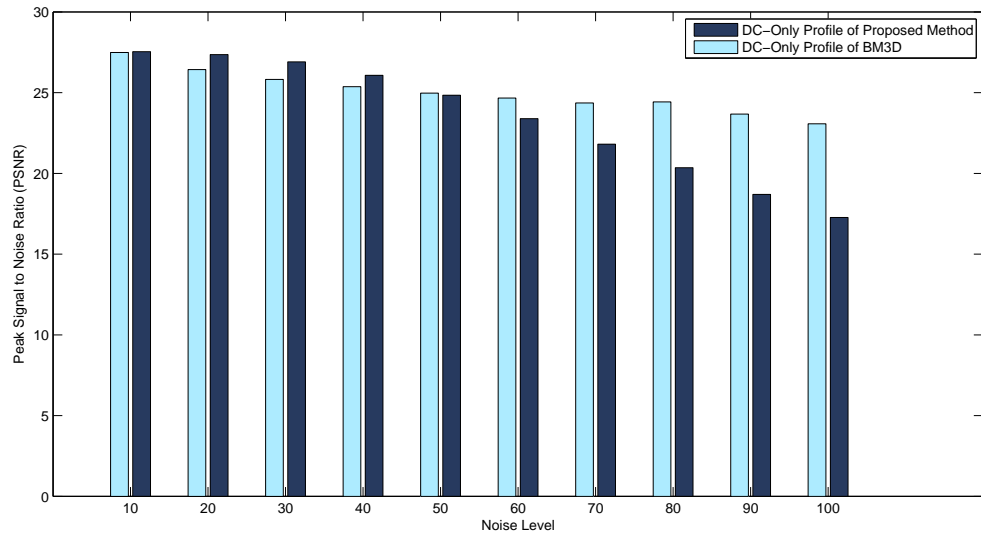


Figure 4.30: Average PSNR Comparison of DC-Only Profile with 75% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)

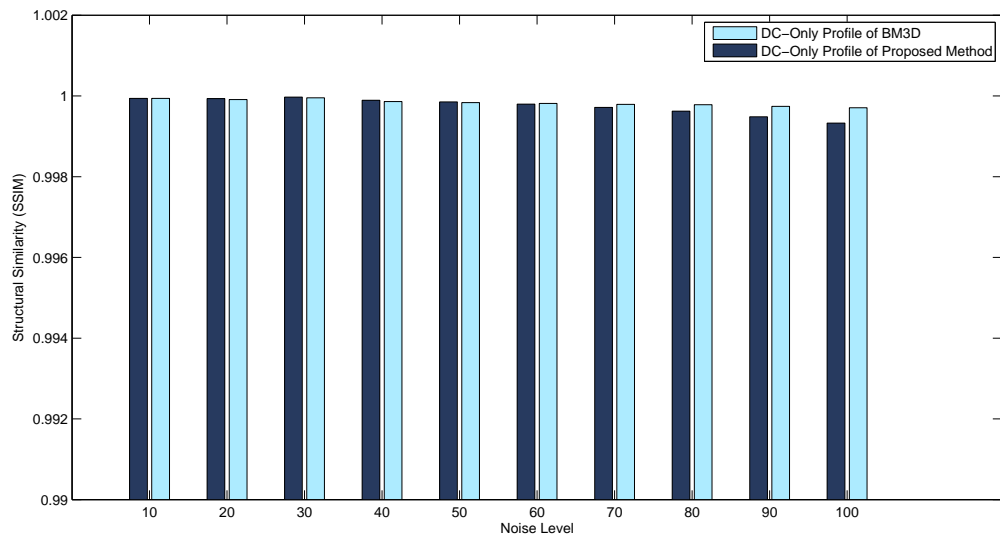


Figure 4.31: Average SSIM Comparison of DC-Only Profile with 75% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	25.70	25.17	0.999879	0.999846
20	24.64	25.15	0.999820	0.999844
30	24.05	25.11	0.999774	0.999840
40	23.64	25.01	0.999735	0.999834
50	23.31	24.75	0.999700	0.999822
60	23.07	24.32	0.999671	0.999803
70	22.83	23.58	0.999640	0.999771
80	22.87	22.81	0.999622	0.999736
90	22.45	21.44	0.999596	0.999664
100	22.30	20.45	0.999561	0.999588

Table 4.11: Performance Comparison of DC-Only Profile with 50% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	27.49	27.54	0.999940	0.999939
20	26.43	27.36	0.999909	0.999932
30	25.82	26.91	0.999966	0.999937
40	25.37	26.07	0.999860	0.999893
50	24.97	24.84	0.999836	0.999853
60	24.67	23.39	0.999814	0.999797
70	24.36	21.81	0.999791	0.999717
80	24.43	20.35	0.999783	0.999624
90	23.67	18.70	0.999741	0.999484
100	23.07	17.27	0.999707	0.999326

Table 4.12: Performance Comparison of DC-Only Profile with 75% Overlapping using Improved Wiener Filter and 3D Zigzag Thresholding ($\gamma = 1$)

4.5.5 Effect of Gamma (γ) in 3D Zigzag Thresholding

The 3D zigzag thresholding follows Equation 4.2. We have seen that the 3D zigzag thresholding cannot be used regardless of the type of images or noise level if the zigzag thresholding is linear, meaning that it increases by the same amount each time it iterates.

$$y = x^\gamma \quad (4.2)$$

What if our γ curve is quadratic or exponential? We executed some easy experiments to see the effect of γ . We changed the value $\gamma = 1$ to $\gamma = 1.5$ i.e., made it *exponential*. Then we compared our result with the output of $\gamma = 1$. In this case, we got better denoising for *Non Overlapping Blocks*, almost same denoising for *50% Overlapping Blocks* and worst denoising for *75% Overlapping Blocks* as compared to $\gamma = 1$.

$\gamma = 0.5$ should produce different results and so do the other values of γ . We leave this



Figure 4.32: Subjective Comparison of DC-Only Profile with 75% Overlapping. Left: Noisy Image, Middle: BM3D Output, Right: Proposed Method’s Output.

direction of study as a future work. We note that a suitable value of γ can lead to better denoising of the first step of the DC-Only profile of *BM3D*.

4.5.6 Performance Bound for 3D Zigzag Thresholding

Our proposed improvement over Wiener filter leads to improvement regardless of the type of the images or noise level as long as we don’t go for the maximum overlapping of a patch while searching for similar patches. In practice, of course, we never go for the maximum overlapping. The idea of *BM3D* is to look for the similar patches within a given window and to keep only 16 most similar patches for building the 3D-block- indicated by the variable N_2^{ht} in [1]. *BM3D* authors mentioned in [1] that it is not possible to improve the denoising performance much using more than 16 similar patches. For a general image, it is possible to find more than 16 similar patches even without overlapping the patches of the search window. For instance, consider a search window consist of 40×40 pixel and we are working with patches with 8×8 . Hence, we have 25 patches of 8×8 dimension. Since the most similar patch with respect to the reference is the reference patch itself, we need to find more 15 similar patches out of 24 patches. For a image with wide smooth region or repetitive pattern, it is not impossible to find so. We need to go for 50% overlapping only if we don’t find 15 similar patches out 24 patches. In this thesis, all DC-Only profile experiments were performed based on *all* similar patches, not based on only 16 patches. Our results in every case show that the more we take similar patches, the more denoising is achievable. This is one finding of this thesis. As long as we follow the parameters set for original *BM3D*, we don’t need overlapping blocks and we can say that our proposed improvement of Wiener filter is capable of achieving more denoising for DC-Only profile of *BM3D* regardless of the type of images or noise level.

However, when we use our proposed 3D zigzag thresholding, we need to be careful! As discussed earlier, we cannot use this idea for higher level of noise. If we use for higher noise, it will deteriorate the image instead of improving it. This is because the idea of 3D zigzag thresholding is to apply more thresholding for less important image coefficients and vice versa. Also, this idea tries to avoid the sudden change of threshold level in case of *BM3D*. To do

so, it skips the DC coefficient and first few AC coefficients. Unfortunately, it also skips a good amount of noise along with the DC and first few AC coefficients. When the noise level increases, DC and first few AC coefficients preserve more noise information.

Therefore, from our experiments, we suggest to use 3D zigzag thresholding (linear) upto the noise level $\sigma = 50$ for DC-Only profile.

4.6 Improved BM3D for Wavelet Profile

This profile may be defined as *Mainstream BM3D* since the authors of *BM3D* [1] recommended to use Wavelet transform in their proposed denoising method. Although we improved the performance of DC-Only profile, unless we achieve improvement in the main stream *BM3D*, we cannot claim that we improved the performance of *BM3D*. This is because, the main stream *BM3D* has PSNR gain way better than DC-Only profile. Hence, the major concern of this thesis is to improve the performance of Wavelet profile of *BM3D*.

In this section, we will explore that how the idea of our proposed Wiener filter achieves better denoising performance of *BM3D*. Unlike DC-Only profile, since there is no DC and AC coefficients, we have only proposed improvement in this case- to use our improved Wiener filter in the second step of *BM3D*.

4.6.1 Parameterized Setup for Wavelet Profile

For the wavelet profile, we used exactly same parameterized setup as in *BM3D* [1]. We present the basic parameterized setup from the original article [1] in table 4.13 for reader's convenience. We use exactly the same parameters to ensure the same environment for the experiment. The wavelet profile uses two sub-profiles called *Normal Profile* and *Fast Profile*. The only difference between them is that we compromise the denoising performance to reduce computational complexity in fast profile. Another difference between these two profiles is the fast profile uses predictive searching in order to decrease its searching time while the normal profile uses only exhaustive searching.

4.6.2 Normal Profile

In normal profile, we executed the same experiments as in DC-Only profiles. Since the wavelet profile itself exploits higher performance as compared to DC-Only profile, even a small increase in PSNR/SSIM indicates a reasonable improvement. The experimental results for this profile are given in table 4.14.

The performance comparison curves show that our proposed method is capable of achieving a maximum of 1dB PSNR for higher noise levels, while for the lower noise level the achieved PSNR is less. Fig. 4.33 and Fig. 4.34 respective show the PSNR and SSIM comparison of normal profile of *BM3D* with our proposed method. Fig. 4.35 presents a subjective measure for Lena image with $\sigma = 50$.

It is clearly visible from Fig. 4.33 and Fig. 4.34 that there is a sharp PSNR drop at $\sigma = 40$. This is because *BM3D* treats all noise levels below $\sigma = 40$ different than the noise levels above

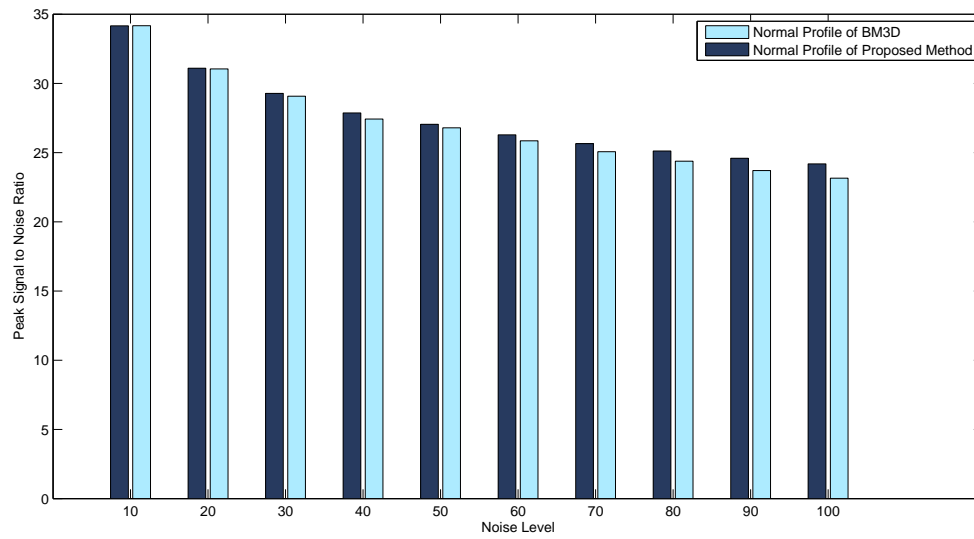


Figure 4.33: Average PSNR Comparison of Normal Profile of BM3D with Proposed Method

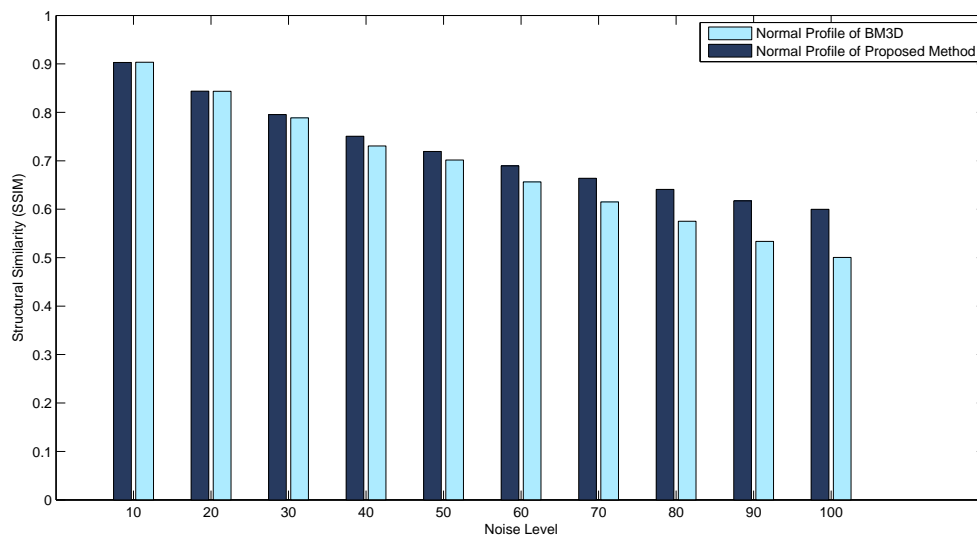


Figure 4.34: Average SSIM Comparison of Normal Profile of BM3D with Proposed Method

	Notations	Meaning	Fast Profile	Normal Profile		
				$\sigma \leq 40$	$\sigma > 40$	
Parameters for Step 1 (ht)	τ_{2D}^{ht}	2D Transform Used	2D-Bior1.5	2D-Bior1.5	2D-DCT	
	N_1^{ht}	Patch Size	8	8	12	
	N_2^{ht}	Maximum Number of Similar Patches Retained	16	16	16	
	N_{step}^{ht}	Step of Reference Patch	6	3	4	
	N_S^{ht}	Size of Search Window	25	39	39	
	N_{FS}^{ht}	Exhaustive Search Window Size	6	1	1	
	N_{PR}^{ht}	Predictive Search Window Size	3	-	-	
	β^{ht}	Parameters for Kaiser Window	2.0	2.0	2.0	
	λ_{2D}	Pre-processing Threshold	0	0	2.0	
	λ_{3D}	Hard Threshold	2.7	2.7	2.8	
	τ_{match}^{ht}	Similarity Threshold for Patches	2500	2500	5000	
Parameters for Step 2 (wie)	τ_{2D}^{wie}	2D Transform Used	2D-DCT	2D-DCT	2D-DCT	
	N_1^{wie}	Patch Size	8	8	11	
	N_2^{wie}	Maximum Number of Similar Patches Retained	16	32	32	
	N_{step}^{wie}	Step of Reference Patch	5	3	6	
	N_S^{wie}	Size of Search Window	25	39	39	
	N_{FS}^{wie}	Exhaustive Search Window Size	5	1	1	
	N_{PR}^{wie}	Predictive Search Window Size	2	-	-	
		τ_{match}^{wie}	Similarity Threshold for Patches	400	400	3500
		β^{wie}	Parameters for Kaiser Window	2.0	2.0	2.0
Common			1D-Haar	1D-Haar	1D-Haar	

Table 4.13: Parameterized Setup for Wavelet Profile of BM3D

$\sigma = 40$. Since we are following the same algorithm of *BM3D*, except that we are using our proposed Wiener filter, we are having the same sharp PSNR drop at $\sigma = 40$.

4.6.3 Fast Profile

The fast profile is similar to normal profile, except that this profile is faster and the searching is predictive and the performance is slightly lower than normal profile. Still, this profile is comparable with normal profile unlike DC-Only profile where we have huge PSNR drop as compared to normal profile. The experimental results obtained for fast profile is presented in table 4.15.



Figure 4.35: Subjective Measure for Normal Profile of BM3D with Proposed Method

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	34.17	34.16	0.903426	0.902944
20	31.04	31.10	0.843516	0.843756
30	29.08	29.28	0.788699	0.795423
40	27.42	27.87	0.730636	0.750625
50	26.79	27.05	0.701603	0.719319
60	25.85	26.28	0.656344	0.689510
70	25.06	25.65	0.615045	0.663826
80	24.37	25.11	0.575113	0.640845
90	23.70	24.59	0.533586	0.617141
100	23.15	24.18	0.500376	0.599786

Table 4.14: Performance Comparison of Normal Profile with Proposed Method

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	34.18	34.17	0.903621	0.903149
20	31.04	31.09	0.843739	0.843953
30	29.07	29.27	0.788761	0.795254
40	27.45	27.91	0.731555	0.752113
50	26.81	27.06	0.702944	0.720458
60	25.89	26.30	0.657783	0.690009
70	25.07	25.63	0.614373	0.662845
80	24.38	25.11	0.575239	0.641276
90	23.76	24.64	0.538769	0.620653
100	23.14	24.17	0.499743	0.599324

Table 4.15: Performance Comparison of Fast Profile with Proposed Method

We plot the table 4.15 in Fig. 4.36 and Fig. 4.36. One can examine that these two figures are almost similar to Fig. 4.33 and Fig. 4.34. If we compare these with any of DC-Only profile, we realize why the usage of DC-Only profile is discouraged. However, since we improve the DC-Only profile of *BM3D*, they are now comparable to any of the wavelet profiles. Fig. 4.38 shows a subjective measure for the image Lena with $\sigma = 50$.

4.7 Extension of Wavelet Profile for Color Image Denoising

In color image denoising experiments of this section, we used four images as shown in Fig. 4.39. These four images are all 24-bit true color 512×512 images.

One approach to employ Color *BM3D* (*CBM3D*) to 24-bit true color images is to apply *BM3D* separately on each of its channels. However, correct grouping is one of the key properties of *BM3D* and it largely depends on the noise level. Again, grouping is a time consuming operation, doing it thrice makes the algorithm considerably slower. Moreover, three

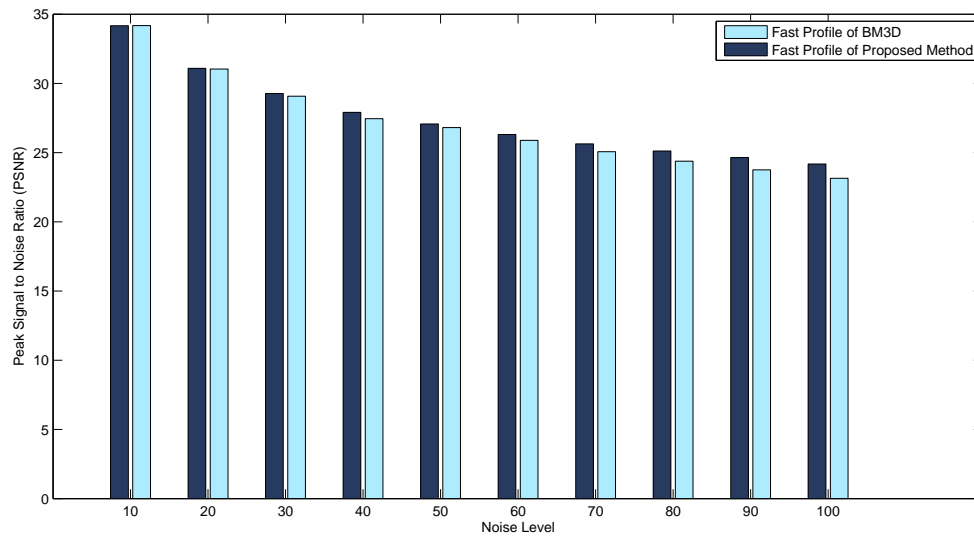


Figure 4.36: Average PSNR Comparison of Fast Profile of BM3D with Proposed Method

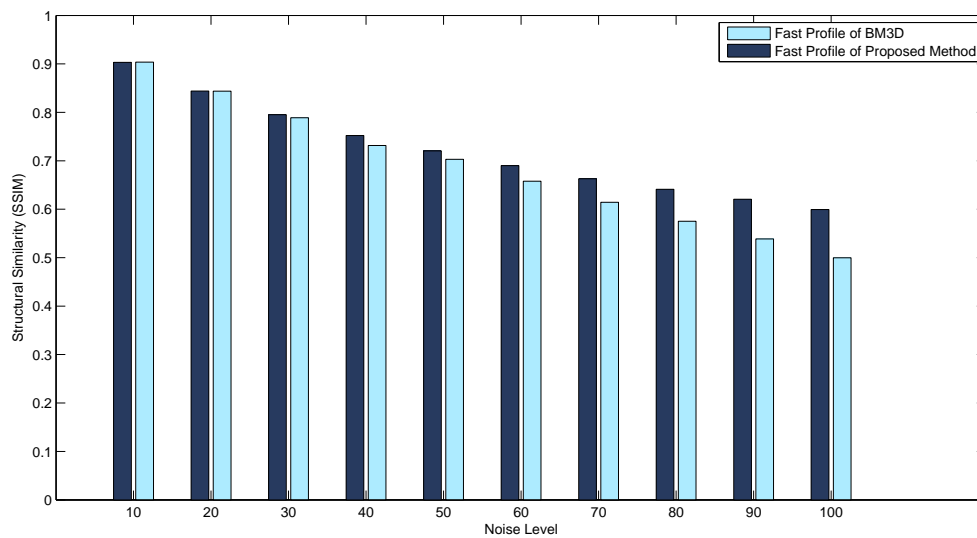


Figure 4.37: Average SSIM Comparison of Fast Profile of BM3D with Proposed Method



Figure 4.38: Subjective Measure for Fast Profile of BM3D with Proposed Method



Figure 4.39: 24 – bit True Color Images used for Color Image Denoising Experiments

channels should generate three different grouping with sparsity of image information which lead to erroneous hard thresholding.

BM3D extension to color image denoising is realized by converting a noisy RGB image with AWGN on each of its channel to a luminance and chrominance transformed space. In the transformed space, the luminance signal contains much of the image information while the chrominance signals contain low frequency information. Therefore, *CMB3D* performs grouping on luminance channel only and use exactly the same grouping for chrominance channels. The idea behind this form of grouping is the concept that if the luminance of two blocks are mutually similar, then the chrominance of these blocks are also mutually similar [1].

We used the same concept for color image denoising as in *CBM3D* except that in second step, we used our improved Wiener filter instead of the existing Wiener filter. The experiments are performed for both normal and fast profiles. The experimental results for normal color profile are presented in table 4.16. The performance comparison charts in Fig. 4.40 and Fig. 4.41 show that our proposed method achieve much better denoising performance than original *BM3D*. For visual inspection of denoised color images of normal profile, we refer the reader to Fig. 4.42.

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	34.11	34.11	0.937238	0.937338
20	31.29	31.35	0.896293	0.897323
30	29.56	29.70	0.860060	0.863810
40	27.92	28.14	0.818205	0.824023
50	27.61	27.80	0.797585	0.806443
60	26.81	27.06	0.768438	0.781308
70	26.07	26.43	0.736980	0.757270
80	25.47	25.93	0.709575	0.737285
90	24.89	25.42	0.681548	0.717313
100	24.23	24.85	0.648800	0.692223

Table 4.16: Performance Comparison of Color Profile (Normal) with Proposed Method

For the fast color profile, the experimental results are shown in table 4.17. The data are plotted in Fig. 4.43 and Fig. 4.44 respectively for PSNR and SSIM measurement. The output is shown in Fig. 4.45 for $\sigma = 50$ for a visual inspection of the reader. Since the fast profile compromises the performance in order to reduce time, the output is poorer than normal profile. Therefore, for the image presented for $\sigma = 50$ in Fig. 4.45, the difference may not be that much visible. However, for a higher level of noise, the difference is clearly visible to human eyes. We present all the outputs of our experiments in Appendix B.

4.8 Summary of Results

In this chapter, we presented a number of experimental results that we extensively performed on *BM3D* and our proposed method. For Wavelet and Color profiles of *BM3D* (a.k.a. main stream *BM3D*), we achieved better PSNR and SSIM for our method. The DC-Only profile

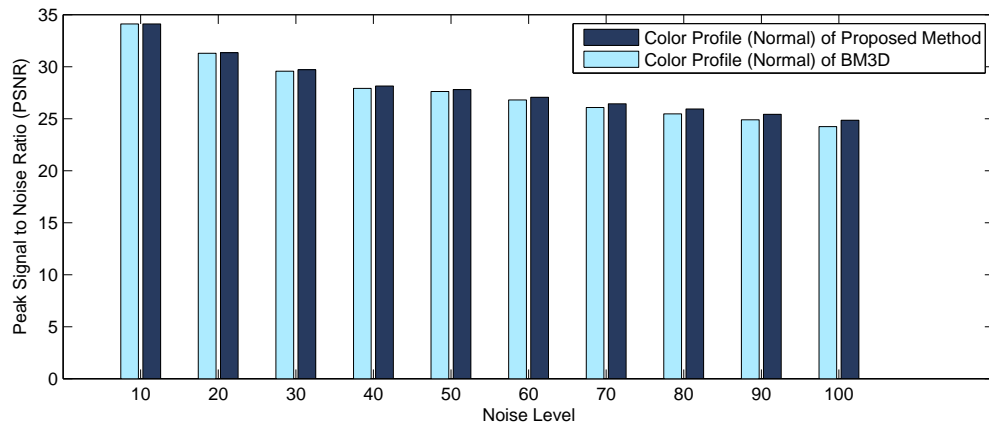


Figure 4.40: Average PSNR Comparison of Color Profile (Normal) of BM3D with Proposed Method

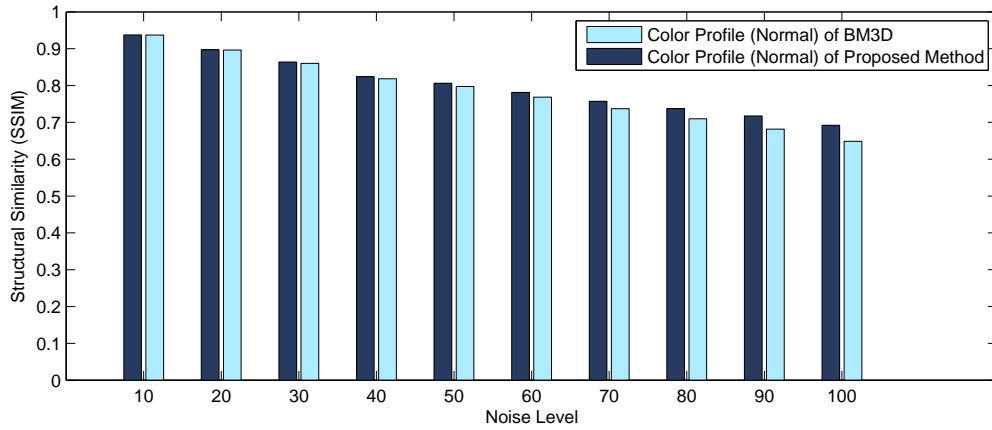


Figure 4.41: Average SSIM Comparison of Color Profile (Normal) of BM3D with Proposed Method



Figure 4.42: Subjective Measure for Color Profile (Normal) of BM3D with Proposed Method

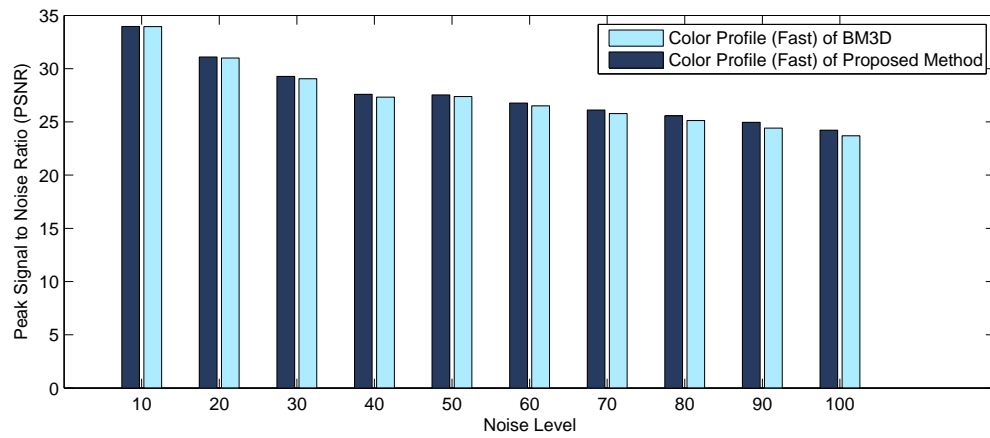


Figure 4.43: Average PSNR Comparison of Color Profile (Fast) of BM3D with Proposed Method

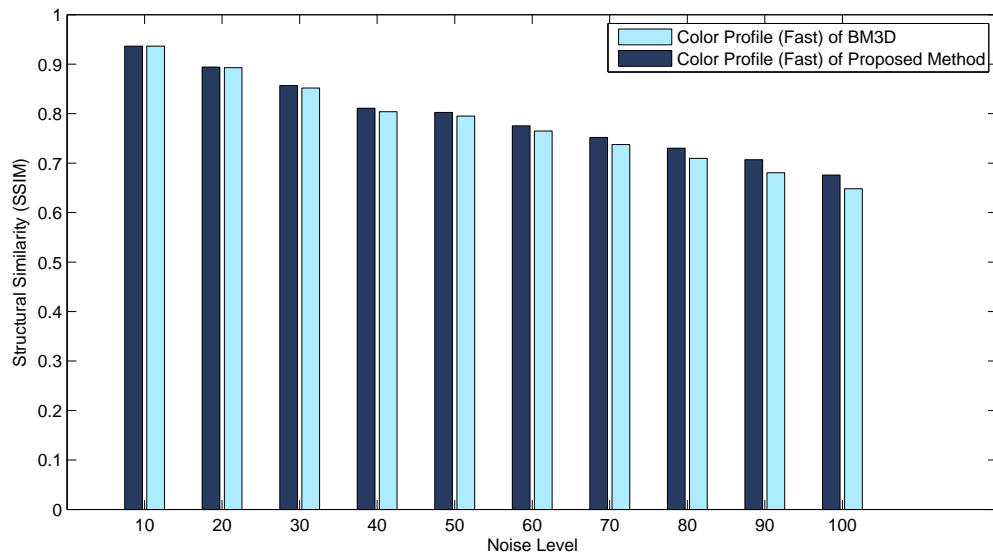


Figure 4.44: Average SSIM Comparison of Color Profile (Fast) of BM3D with Proposed Method

Noise Level	BM3D PSNR	Proposed PSNR	BM3D SSIM	Proposed SSIM
10	33.95	33.96	0.936408	0.936363
20	31.00	31.09	0.892963	0.894203
30	29.05	29.28	0.851715	0.856908
40	27.32	27.59	0.803835	0.810890
50	27.38	27.54	0.795188	0.802460
60	26.50	26.77	0.764728	0.775403
70	25.78	26.12	0.737650	0.751798
80	25.13	25.57	0.709728	0.730303
90	24.42	24.96	0.680508	0.706988
100	23.69	24.22	0.648338	0.675823

Table 4.17: Performance Comparison of Color Profile (Fast) with Proposed Method



Figure 4.45: Subjective Measure for Color Profile (Fast) of BM3D with Proposed Method

is usually poorer performance given, we also showed that for most of the cases, our proposed method works better than DC-Only profile of *BM3D*. In this section, we summarize the results presented in the chapter for reader's convenience. The summary is given in table 4.18.

		Proposed Method	
DC-Only Profile	Using Improved Wiener Filter Only	No Overlapping	Better
		Moderate Overlapping	Better
		Maximum Overlapping	Close and Comparable to BM3D
	Using 3D Zigzag Thresholding Only	No Overlapping	Better
		Moderate Overlapping	Better
		Maximum Overlapping	Better up to $\sigma = 50$
Using Improved Wiener Filter and 3D Zigzag Thresholding	No Overlapping	Better	
	Moderate Overlapping	Better up to $\sigma = 70$	
	Maximum Overlapping	Better up to $\sigma = 50$	
Wavelet Profile	Normal Profile	Better	
	Fast Profile	Better	
Color Profile	Normal Profile	Better	
	Fast Profile	Better	

Table 4.18: Summary of Experimental Results

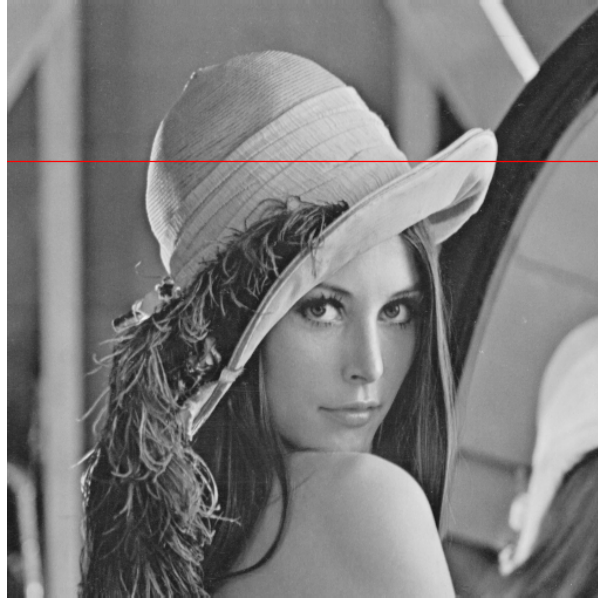


Figure 4.46: Lena Image for Intensity Profile Calculation. Red Line shows the Scan Line taken as input to Intensity Profile

4.9 Intensity Profile

Intensity profile is one measure for inspecting how sharp the edges are after denoising [36]. In intensity profile, we choose one scan line on the true image, noisy image and denoised image and plot them together to see how close the denoised profile is to the original profile. Also, it shows us how sharp the edges are after achieving denoising. For instance, let us consider the Lena image given in Fig. 4.46. Here, we consider the 140th row (indicated by red line) for our intensity profile calculation. If we plot this scan line, we end up with the Fig. 4.47.

Now, to check how close our proposed method (as well as *BM3D*) is to the true noise free image, we perform the same task. That is, we consider the 140th row from our denoised image and *BM3D*'s denoised image and plot them together. The denoised images are taken for

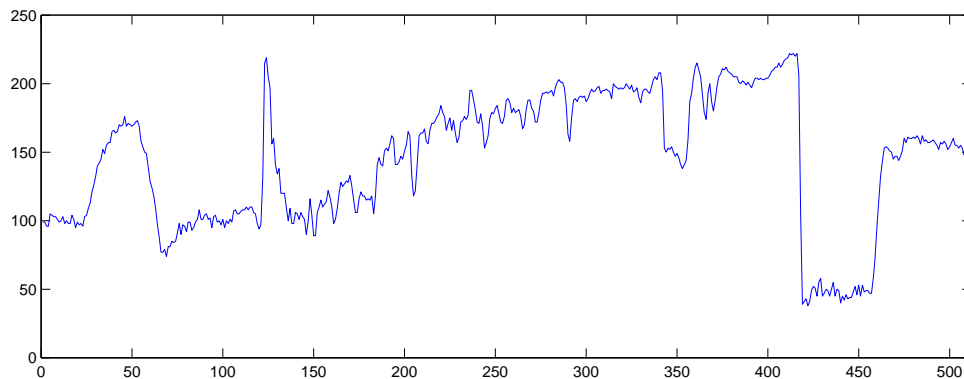


Figure 4.47: Intensity Profile for the Scan Line shown in Fig. 4.46

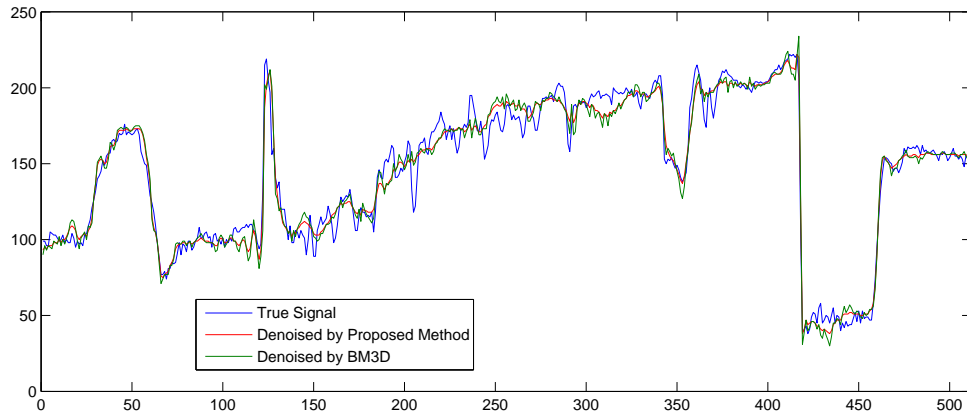


Figure 4.48: Intensity Profiles for *BM3D* and Proposed Method with respect to the True Image

$\sigma = 50$. This is shown in Fig. 4.48.

It is visible from this intensity profile that our proposed method is more closer to the true signal as compared to its *BM3D* counterpart. Having a careful look at all the transitions of Fig. 4.48, we notice that the red line is more close to the blue line as compared to the green line. It proves that the proposed method is capable of achieving more sharpness than that of *BM3D*.

To conclude this chapter, we can say that we have been able to improve the performance of *BM3D* in the sense that our proposed method achieves clearly better result than the *BM3D* for its main stream profiles. Also, for the DC-Only profile, since *BM3D* does not accept more than 16 similar patches to be kept in the 3D group, it is possible to achieve 16 similar patches using no overlapping of the similar patches except in the case of extremely non-textured images.

Chapter 5

Conclusion and Future Work

We conclude this thesis by discussing the possible extension of our works, its applications and the summary of our proposed method.

5.1 Conclusion

In this thesis, we rigorously reviewed the noise for digital images, its variants and existing methods of image denoising. We presented a number of ideas that have essentially improved the performance of the current state of the art of image denoising- *BM3D*. Our ideas work for both monochrome (gray-scale) and color images.

We discussed with appropriate example how the MSE can be misleading and how this affects the performance of Wiener filter. We then proposed an improved Wiener filter which is optimized for SSIM. Our performance analysis proved that our proposed Wiener filter works better than the existing MSE optimized Wiener filter.

We also proposed several improvements for different profiles of *BM3D*. We considered using our improved Wiener filter as a core component of *BM3D* in its second step. This SSIM optimized Wiener filter improved the PSNR gain of *BM3D* up to a more than $1dB$. For the state of the art like *BM3D*, improving $1dB$ PSNR is considered huge.

We particularly contributed in the DC-Only profile of *BM3D*. We proposed a 3D zigzag thresholding to get rid of the problem of data loss from the DC and the first few AC coefficients of the 3D blocks. Our performance analysis for DC-only profile showed that 3D zigzag thresholding idea has been able to make the output of DC-only profile less blurry, thereby improving the denoising performance of this profile.

Last but not the least, we developed a Matlab software for *BM3D* image denoising. This should help a lot of users around the world to carry on research on image denoising, particularly on *BM3D*.

5.2 Future Work

Wiener filter is widely used in many areas of image processing including image compression, denoising, restoration, transmission, registration, projection and video rendering. Wiener filter is also used for computer graphics, statistical signal processing and audio signal processing. As

long as the Wiener filter involves a 2D discrete signal, our proposed improvement guarantees to perform better since all other Wiener filters are optimized for MSE. Hence, one possible future work for our study can be a rigorous review of where else our proposed method can contribute.

There are other variants of SSIM recently proposed called Multi Scale Structural Similarity (MS-SSIM) [37]. One good future work is to observe the affect of our proposed improvement over Wiener filter as well as over *BM3D* using the MS-SSIM instead of SSIM. Also, there are other image quality measurement metrics such as Image Quality Index (IQI) [38], Normalized Correlation [39], Sum of Absolute Differences and many others [40]. A possible future work should test our proposed method by using all of these measures, instead of only using SSIM and present the differences.

BM3D is specially designed for reducing AWGN. However, there are other additive noise like exponential noise, gamma noise and Rayleigh noise which are very similar in nature to AWGN. One possible future work is play with all these noises and observe if *BM3D* is good for other noise types, too.

In recent years, *BM3D* has been extended for video denoising. Also there is *BM4D* already present [22]. Since our idea is to change the core of *BM_xD* in general, a study of assessing our method in all *BM_xD* versions will be a good possible future work.

BM3D has different response to non-textured images as it does not find more similar blocks needed for collaborative filtering [14]. In order to get more number of similar blocks, one can check if taking variable size blocks may give us so. That is, we can rescale blocks of different size that are structurally similar and see if that improves the performance of *BM3D* for the non textured images where it is difficult to find similar blocks.

Bibliography

- [1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image Denoising by Sparse 3-D Transform-domain Collaborative Filtering,” *Image Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [2] S. V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons, 2008.
- [3] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. MIT press Cambridge, MA, 1949, vol. 2.
- [4] M. Lebrun, “An Analysis and Implementation of the BM3D Image Denoising Method,” *Image Processing On Line*, no. 2012, 2012.
- [5] J. Rao, “A Characterization of the Normal Distribution,” *The Annals of Mathematical Statistics*, pp. 914–919, 1958.
- [6] R. Gonzales and R. Woods, *Digital Image Processing*. Addison-Wesley Publishing Company, 2007.
- [7] M. Petrou and C. Petrou, *Image processing: the fundamentals*. John Wiley & Sons, 2010.
- [8] F. Qiu, J. Berglund, J. R. Jensen, P. Thakkar, and D. Ren, “Speckle Noise Reduction in SAR Imagery using a Local Adaptive Median Filter,” *GIScience & Remote Sensing*, vol. 41, no. 3, pp. 244–266, 2004.
- [9] A. Babu and M. R. El-Sakka, “SRAD with Weighted Diffusion Function,” in *Image Analysis and Recognition*. Springer, 2013, pp. 264–273.
- [10] B. K. Gunturk and X. Li, *Image Restoration: Fundamentals and Advances*. CRC Press, 2012.
- [11] P. Perona and J. Malik, “Scale-space and Edge Detection using Anisotropic Diffusion,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 7, pp. 629–639, 1990.
- [12] S. Eddins, “Matlab Central Steve on Image Processing,” 2007.

- [13] M. Rothenberg, "A New Inverse-filtering Technique for Deriving the Glottal Air Flow Waveform During Voicing," *The Journal of the Acoustical Society of America*, vol. 53, no. 6, pp. 1632–1645, 1973.
- [14] M. M. Hasan, "Adaptive Edge-Guided Block-Matching and 3D Filtering (BM3D) Image Denoising Algorithm," Master's thesis, The University of Western Ontario, 2014.
- [15] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 60–65.
- [16] P. Chatterjee and P. Milanfar, "Is Denoising Dead?" *Image Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 895–911, 2010.
- [17] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image Denoising: Can Plain Neural Networks Compete with BM3D?" in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2392–2399.
- [18] Y. S. Zhang, S. J. Zhu, and Y. J. Li, "BM3D Denoising Algorithm with Adaptive Block-match Thresholds," *Applied Mechanics and Materials*, vol. 229, pp. 1715–1720, 2012.
- [19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.
- [20] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color Image Denoising via Sparse 3D Collaborative Filtering with Grouping Constraint in Luminance-Chrominance Space," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1. IEEE, 2007, pp. I–313.
- [21] K. Dabov, A. Foi, and K. Egiazarian, "Video Denoising by Sparse 3D Transform-domain Collaborative Filtering," in *Proc. 15th European Signal Processing Conference*, vol. 1, no. 2, 2007, p. 7.
- [22] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction," *Image Processing, IEEE Transactions on*, vol. 22, no. 1, pp. 119–133, 2013.
- [23] P. Chatterjee and P. Milanfar, "Patch-based Near-optimal Image Denoising," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1635–1649, 2012.
- [24] A. Rehman and Z. Wang, "SSIM-based Non-local Means Image Denoising," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 217–220.
- [25] E. L. Lehmann and G. Casella, *Theory of Point Estimation*. Springer, 1998, vol. 31.
- [26] Z. Wang and A. C. Bovik, "Modern Image Quality Assessment," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 2, no. 1, pp. 1–156, 2006.

- [27] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 2366–2369.
- [28] Z. Wang and A. C. Bovik, "Mean Squared Error: Love it or Leave it? A New Look at Signal Fidelity Measures," *Signal Processing Magazine, IEEE*, vol. 26, no. 1, pp. 98–117, 2009.
- [29] S. S. Channappayya, A. C. Bovik, and R. Heath, "A Linear Estimator Optimized for the Structural Similarity Index and its Application to Image Denoising," in *Image Processing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 2637–2640.
- [30] S. S. Channappayya, A. C. Bovik, C. Caramanis, and R. W. Heath, "Design of Linear Equalizers Optimized for the Structural Similarity Index," *Image Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 857–872, 2008.
- [31] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian *et al.*, "BM3D Image Denoising with Shape-adaptive Principal Component Analysis," in *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.
- [32] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *Computers, IEEE Transactions on*, vol. 100, no. 1, pp. 90–93, 1974.
- [33] G. K. Wallace, "The JPEG Still Picture Compression Standard," *Consumer Electronics, IEEE Transactions on*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [34] G. M. R. B. Black, "YUV Color space," *Communications Engineering Desk Reference*, p. 469, 2009.
- [35] J. Mannos and D. J. Sakrison, "The Effects of a Visual Fidelity Criterion of the Encoding of Images," *Information Theory, IEEE Transactions on*, vol. 20, no. 4, pp. 525–536, 1974.
- [36] A. Babu, "Ratio-based Edge Detection Inspired Speckle Reducing Anisotropic Diffusion," Ph.D. dissertation, The University of Western Ontario, 2013.
- [37] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale Structural Similarity for Image Quality Assessment," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, vol. 2. Ieee, 2003, pp. 1398–1402.
- [38] Z. Wang and A. C. Bovik, "A Universal Image Quality Index," *Signal Processing Letters, IEEE*, vol. 9, no. 3, pp. 81–84, 2002.
- [39] V. Laparra, J. Muñoz-Marí, and J. Malo, "Divisive Normalization Image Quality Metric Revisited," *JOSA A*, vol. 27, no. 4, pp. 852–864, 2010.
- [40] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference Image Quality Assessment in the Spatial Domain," *Image Processing, IEEE Transactions on*, vol. 21, no. 12, pp. 4695–4708, 2012.

Appendix A

Detail Objective Experimental Results

In this chapter, we will present detail objective experimental results of some selective cases of our experiments. We consider that the Wavelet Profile and Color Profile of BM3D are more significant than the DC-Only profile and main stream focus in the study of image denoising. Therefore, we present all the experimental results of Wavelet and Color profiles in this chapter.

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
10	Baboon	33.15	33.13	0.931550	0.932690
	Barbara	34.92	34.92	0.941430	0.941430
	Boat	33.87	33.87	0.887070	0.887070
	Couple	34.00	34.00	0.907880	0.907880
	Hill	33.58	33.58	0.883120	0.883120
	Lena	35.86	35.87	0.915690	0.915070
	Man	33.92	33.92	0.906480	0.906480
	Peppers	34.05	33.99	0.854190	0.849810
	Average	34.17	34.16	0.903430	0.902940

Table A.1: Wavelet (Normal) Profile for $\sigma = 10$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
20	Baboon	29.02	29.05	0.848660	0.850620
	Barbara	31.62	31.73	0.903390	0.904950
	Boat	30.86	30.84	0.827120	0.824540
	Couple	30.78	30.77	0.846980	0.845820
	Hill	30.71	30.66	0.805920	0.802080
	Lena	32.87	33.07	0.872560	0.877150
	Man	30.61	30.56	0.835330	0.832630
	Peppers	31.88	32.09	0.808170	0.812260
	Average	31.04	31.10	0.843516	0.843756

Table A.2: Wavelet (Normal) Profile for $\sigma = 20$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
30	Baboon	26.86	26.82	0.776360	0.772650
	Barbara	29.45	29.71	0.857860	0.865290
	Boat	28.91	29.06	0.774270	0.777990
	Couple	28.78	28.84	0.791530	0.792490
	Hill	28.95	29.01	0.748810	0.747080
	Lena	30.71	31.18	0.821120	0.841970
	Man	28.80	28.85	0.777730	0.779060
	Peppers	30.15	30.75	0.761910	0.786850
	Average	29.08	29.28	0.788699	0.795423

Table A.3: Wavelet (Normal) Profile for $\sigma = 30$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
40	Baboon	25.24	25.21	0.710480	0.698040
	Barbara	27.46	28.00	0.801790	0.821250
	Boat	27.25	27.59	0.718020	0.733090
	Couple	27.03	27.33	0.728720	0.740180
	Hill	27.55	27.88	0.688940	0.701360
	Lena	28.90	29.78	0.765250	0.814680
	Man	27.19	27.50	0.715510	0.732000
	Peppers	28.73	29.65	0.716380	0.764400
	Average	27.42	27.87	0.730636	0.750625

Table A.4: Wavelet (Normal) Profile for $\sigma = 40$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
50	Baboon	24.52	24.35	0.657970	0.638940
	Barbara	27.02	27.23	0.779400	0.792810
	Boat	26.57	26.67	0.686130	0.698350
	Couple	26.36	26.43	0.700950	0.706620
	Hill	26.83	27.05	0.660690	0.670120
	Lena	28.29	28.97	0.745130	0.795930
	Man	26.59	26.75	0.684520	0.700400
	Peppers	28.15	28.92	0.698030	0.751380
	Average	26.79	27.05	0.701603	0.719319

Table A.5: Wavelet (Normal) Profile for $\sigma = 50$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
60	Baboon	23.78	23.63	0.618860	0.594590
	Barbara	25.90	26.30	0.730790	0.760850
	Boat	25.71	25.97	0.643800	0.670080
	Couple	25.37	25.63	0.646710	0.667810
	Hill	25.97	26.43	0.614980	0.643080
	Lena	27.20	28.05	0.697690	0.773000
	Man	25.68	26.08	0.639080	0.674720
	Peppers	27.16	28.16	0.658840	0.731950
	Average	25.85	26.28	0.656344	0.689510

Table A.6: Wavelet (Normal) Profile for $\sigma = 60$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
70	Baboon	23.03	22.99	0.573580	0.552350
	Barbara	25.00	25.47	0.682850	0.725100
	Boat	24.89	25.33	0.601890	0.647450
	Couple	24.62	24.95	0.608490	0.639370
	Hill	25.23	25.85	0.574340	0.617100
	Lena	26.37	27.49	0.655380	0.758650
	Man	25.08	25.61	0.605760	0.657400
	Peppers	26.26	27.47	0.618070	0.713190
	Average	25.06	25.65	0.615045	0.663826

Table A.7: Wavelet (Normal) Profile for $\sigma = 70$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
80	Baboon	22.48	22.55	0.527810	0.513720
	Barbara	24.27	24.80	0.642830	0.697260
	Boat	24.19	24.77	0.563100	0.626100
	Couple	23.87	24.40	0.564140	0.614420
	Hill	24.55	25.36	0.533760	0.596140
	Lena	25.62	26.94	0.615430	0.741450
	Man	24.41	25.12	0.562440	0.635770
	Peppers	25.62	26.96	0.591390	0.701900
	Average	24.37	25.11	0.575113	0.640845

Table A.8: Wavelet (Normal) Profile for $\sigma = 80$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
90	Baboon	22.02	22.19	0.497240	0.487390
	Barbara	23.37	24.06	0.586400	0.658840
	Boat	23.57	24.32	0.527020	0.606510
	Couple	23.25	23.90	0.525320	0.586880
	Hill	23.92	24.89	0.502380	0.579110
	Lena	24.90	26.39	0.563140	0.721340
	Man	23.73	24.56	0.520240	0.610960
	Peppers	24.85	26.39	0.546950	0.686100
	Average	23.70	24.59	0.533586	0.617141

Table A.9: Wavelet (Normal) Profile for $\sigma = 90$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
100	Baboon	21.55	21.87	0.462840	0.465420
	Barbara	22.84	23.59	0.556010	0.639640
	Boat	23.00	23.95	0.488360	0.591060
	Couple	22.74	23.53	0.490430	0.567970
	Hill	23.39	24.54	0.466490	0.560120
	Lena	24.25	25.85	0.531180	0.702630
	Man	23.14	24.21	0.488750	0.597690
	Peppers	24.28	25.94	0.518950	0.673760
	Average	23.15	24.18	0.500376	0.599786

Table A.10: Wavelet (Normal) Profile for $\sigma = 100$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
10	Baboon	33.16	33.13	0.931270	0.932390
	Barbara	34.93	34.93	0.941350	0.941350
	Boat	33.87	33.87	0.887670	0.887670
	Couple	33.99	33.99	0.907610	0.907610
	Hill	33.60	33.60	0.883600	0.883600
	Lena	35.89	35.91	0.916760	0.916310
	Man	33.92	33.92	0.907000	0.907000
	Peppers	34.05	33.98	0.853710	0.849260
	Average	34.18	34.17	0.903621	0.903149

Table A.11: Wavelet (Fast) Profile for $\sigma = 10$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
20	Baboon	29.04	29.07	0.849380	0.851240
	Barbara	31.59	31.71	0.902770	0.904420
	Boat	30.84	30.82	0.827640	0.824730
	Couple	30.74	30.72	0.846240	0.845060
	Hill	30.72	30.67	0.807060	0.803010
	Lena	32.88	33.09	0.872300	0.877400
	Man	30.63	30.57	0.836150	0.833180
	Peppers	31.85	32.07	0.808370	0.812580
	Average	31.04	31.09	0.843739	0.843953

Table A.12: Wavelet (Fast) Profile for $\sigma = 20$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
30	Baboon	26.86	26.81	0.777110	0.773470
	Barbara	29.46	29.73	0.858710	0.865560
	Boat	28.90	29.02	0.774490	0.777430
	Couple	28.75	28.81	0.791140	0.791940
	Hill	29.03	29.07	0.748180	0.746190
	Lena	30.63	31.16	0.819620	0.842590
	Man	28.78	28.81	0.778150	0.778250
	Peppers	30.17	30.75	0.762690	0.786600
	Average	29.07	29.27	0.788761	0.795254

Table A.13: Wavelet (Fast) Profile for $\sigma = 30$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
40	Baboon	25.22	25.19	0.711180	0.700260
	Barbara	27.40	27.93	0.800450	0.819870
	Boat	27.30	27.67	0.716920	0.733740
	Couple	27.14	27.43	0.730820	0.742630
	Hill	27.63	27.92	0.695310	0.705710
	Lena	28.94	29.84	0.765740	0.814980
	Man	27.28	27.59	0.717220	0.734890
	Peppers	28.70	29.68	0.714800	0.764820
	Average	27.45	27.91	0.731555	0.752113

Table A.14: Wavelet (Fast) Profile for $\sigma = 40$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
50	Baboon	24.49	24.33	0.658580	0.639940
	Barbara	27.01	27.28	0.780420	0.796500
	Boat	26.62	26.73	0.691200	0.703540
	Couple	26.35	26.40	0.698540	0.703520
	Hill	26.91	27.12	0.661850	0.670750
	Lena	28.28	28.93	0.743470	0.795380
	Man	26.59	26.77	0.688190	0.703670
	Peppers	28.19	28.95	0.701300	0.750360
	Average	26.81	27.06	0.702944	0.720458

Table A.15: Wavelet (Fast) Profile for $\sigma = 50$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
60	Baboon	23.68	23.51	0.612070	0.588070
	Barbara	26.07	26.44	0.737710	0.765210
	Boat	25.64	25.88	0.644410	0.669640
	Couple	25.49	25.68	0.652200	0.670660
	Hill	26.03	26.47	0.615600	0.642310
	Lena	27.19	28.10	0.696380	0.774610
	Man	25.74	26.11	0.639820	0.675620
	Peppers	27.25	28.25	0.664070	0.733950
	Average	25.89	26.30	0.657783	0.690009

Table A.16: Wavelet (Fast) Profile for $\sigma = 60$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
70	Baboon	23.06	22.97	0.571170	0.547940
	Barbara	24.90	25.39	0.682100	0.723670
	Boat	24.94	25.38	0.605550	0.651390
	Couple	24.59	24.97	0.602640	0.636160
	Hill	25.28	25.83	0.575470	0.615920
	Lena	26.36	27.49	0.654900	0.758500
	Man	25.05	25.57	0.600520	0.653470
	Peppers	26.34	27.45	0.622630	0.715710
	Average	25.07	25.63	0.614373	0.662845

Table A.17: Wavelet (Fast) Profile for $\sigma = 70$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
80	Baboon	22.47	22.54	0.529530	0.515860
	Barbara	24.26	24.79	0.640930	0.695970
	Boat	24.20	24.79	0.563420	0.627060
	Couple	23.88	24.43	0.561090	0.611460
	Hill	24.49	25.30	0.538900	0.597960
	Lena	25.64	26.90	0.615660	0.741220
	Man	24.39	25.08	0.566210	0.634450
	Peppers	25.70	27.08	0.586170	0.706230
	Average	24.38	25.11	0.575239	0.641276

Table A.18: Wavelet (Fast) Profile for $\sigma = 80$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
90	Baboon	22.05	22.23	0.504750	0.495060
	Barbara	23.59	24.28	0.596710	0.668320
	Boat	23.54	24.27	0.525470	0.605780
	Couple	23.37	24.00	0.526530	0.589610
	Hill	23.93	24.92	0.505360	0.580120
	Lena	24.91	26.37	0.576250	0.724210
	Man	23.70	24.60	0.519100	0.613200
	Peppers	24.94	26.47	0.555980	0.688920
	Average	23.76	24.64	0.538769	0.620653

Table A.19: Wavelet (Fast) Profile for $\sigma = 90$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
100	Baboon	21.53	21.87	0.462160	0.467680
	Barbara	22.79	23.62	0.551840	0.638910
	Boat	23.07	23.97	0.490560	0.589510
	Couple	22.76	23.53	0.486420	0.563620
	Hill	23.45	24.60	0.471760	0.564500
	Lena	24.17	25.80	0.528330	0.702180
	Man	23.16	24.21	0.489100	0.596650
	Peppers	24.22	25.81	0.517770	0.671540
	Average	23.14	24.17	0.499743	0.599324

Table A.20: Wavelet (Fast) Profile for $\sigma = 100$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
10	Lenna	35.24	35.23	0.935030	0.934250
	Peppers	33.83	33.79	0.904260	0.902250
	Baboon	30.49	30.62	0.937620	0.940970
	Barbara	36.86	36.81	0.972040	0.971880
		Average	34.11	34.11	0.937238

Table A.21: Color (Normal) Profile for $\sigma = 10$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
20	Lenna	32.92	33.00	0.904030	0.903970
	Peppers	31.75	31.83	0.867290	0.867650
	Baboon	26.90	26.96	0.866510	0.870060
	Barbara	33.58	33.61	0.947340	0.947610
		Average	31.29	31.35	0.896293

Table A.22: Color (Normal) Profile for $\sigma = 20$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
30	Lenna	31.38	31.61	0.876160	0.880430
	Peppers	30.42	30.65	0.843170	0.849110
	Baboon	25.09	25.13	0.800900	0.804800
	Barbara	31.35	31.43	0.920010	0.920900
		Average	29.56	29.70	0.860060

Table A.23: Color (Normal) Profile for $\sigma = 30$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
40	Lenna	29.73	30.13	0.840850	0.851660
	Peppers	28.96	29.33	0.812880	0.826620
	Baboon	23.88	23.86	0.742130	0.739960
	Barbara	29.09	29.22	0.876960	0.877850
	Average	27.92	28.14	0.818205	0.824023

Table A.24: Color (Normal) Profile for $\sigma = 40$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
50	Lenna	29.56	29.90	0.832130	0.845410
	Peppers	28.63	28.94	0.801940	0.819060
	Baboon	23.11	23.14	0.683650	0.687760
	Barbara	29.14	29.21	0.872620	0.873540
	Average	27.61	27.80	0.797585	0.806443

Table A.25: Color (Normal) Profile for $\sigma = 50$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
60	Lenna	28.76	29.19	0.806080	0.828030
	Peppers	27.83	28.28	0.777150	0.807110
	Baboon	22.53	22.49	0.646220	0.640570
	Barbara	28.10	28.30	0.844300	0.849520
	Average	26.81	27.06	0.768438	0.781308

Table A.26: Color (Normal) Profile for $\sigma = 60$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
70	Lenna	27.93	28.58	0.772070	0.812010
	Peppers	27.10	27.70	0.751830	0.795980
	Baboon	22.05	21.95	0.610580	0.595720
	Barbara	27.20	27.51	0.813440	0.825370
	Average	26.07	26.43	0.736980	0.757270

Table A.27: Color (Normal) Profile for $\sigma = 70$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
80	Lenna	27.29	28.10	0.747080	0.800860
	Peppers	26.47	27.16	0.728130	0.785180
	Baboon	21.66	21.54	0.580320	0.558810
	Barbara	26.45	26.91	0.782770	0.804290
	Average	25.47	25.93	0.709575	0.737285

Table A.28: Color (Normal) Profile for $\sigma = 80$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
90	Lenna	26.70	27.65	0.715120	0.786420
	Peppers	25.88	26.61	0.704160	0.774070
	Baboon	21.28	21.16	0.555180	0.527250
	Barbara	25.70	26.27	0.751730	0.781510
	Average	24.89	25.42	0.681548	0.717313

Table A.29: Color (Normal) Profile for $\sigma = 90$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
100	Lenna	26.02	27.06	0.688520	0.771150
	Peppers	25.17	26.05	0.672260	0.758670
	Baboon	20.86	20.73	0.520520	0.485160
	Barbara	24.88	25.56	0.713900	0.753910
	Average	24.23	24.85	0.648800	0.692223

Table A.30: Color (Normal) Profile for $\sigma = 100$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
10	Lenna	35.12	35.12	0.933100	0.933100
	Peppers	33.75	33.73	0.903720	0.902220
	Baboon	30.42	30.47	0.938690	0.940010
	Barbara	36.52	36.52	0.970120	0.970120
	Average	33.95	33.96	0.936408	0.936363

Table A.31: Color (Fast) Profile for $\sigma = 10$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
20	Lenna	32.60	32.78	0.899840	0.901480
	Peppers	31.56	31.68	0.863260	0.864550
	Baboon	26.75	26.77	0.866120	0.867670
	Barbara	33.07	33.13	0.942630	0.943110
	Average	31.00	31.09	0.892963	0.894203

Table A.32: Color (Fast) Profile for $\sigma = 20$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
30	Lenna	30.82	31.19	0.866390	0.873410
	Peppers	29.93	30.31	0.833470	0.842960
	Baboon	24.89	24.90	0.797860	0.799560
	Barbara	30.57	30.74	0.909140	0.911700
	Average	29.05	29.28	0.851715	0.856908

Table A.33: Color (Fast) Profile for $\sigma = 30$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
40	Lenna	28.98	29.47	0.824800	0.839020
	Peppers	28.32	28.73	0.797520	0.813570
	Baboon	23.63	23.61	0.735600	0.730090
	Barbara	28.34	28.55	0.857420	0.860880
	Average	27.32	27.59	0.803835	0.810890

Table A.34: Color (Fast) Profile for $\sigma = 40$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
50	Lenna	29.29	29.57	0.834980	0.842280
	Peppers	28.48	28.75	0.805850	0.816570
	Baboon	23.00	23.08	0.675960	0.687070
	Barbara	28.76	28.77	0.863960	0.863920
	Average	27.38	27.54	0.795188	0.802460

Table A.35: Color (Fast) Profile for $\sigma = 50$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
60	Lenna	28.34	28.83	0.808630	0.825300
	Peppers	27.57	28.02	0.780250	0.802030
	Baboon	22.38	22.41	0.633920	0.637390
	Barbara	27.72	27.80	0.836110	0.836890
	Average	26.50	26.77	0.764728	0.775403

Table A.36: Color (Fast) Profile for $\sigma = 60$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
70	Lenna	27.54	28.13	0.779980	0.805430
	Peppers	26.83	27.36	0.759110	0.789420
	Baboon	21.90	21.88	0.600930	0.596450
	Barbara	26.86	27.10	0.810580	0.815890
	Average	25.78	26.12	0.737650	0.751798

Table A.37: Color (Fast) Profile for $\sigma = 70$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
80	Lenna	26.88	27.64	0.756980	0.795180
	Peppers	26.19	26.85	0.736940	0.778840
	Baboon	21.48	21.44	0.568940	0.559870
	Barbara	25.99	26.34	0.776050	0.787320
	Average	25.13	25.57	0.709728	0.730303

Table A.38: Color (Fast) Profile for $\sigma = 80$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
90	Lenna	26.05	26.99	0.727660	0.778340
	Peppers	25.39	26.18	0.709910	0.765200
	Baboon	21.02	20.96	0.535580	0.516550
	Barbara	25.21	25.70	0.748880	0.767860
	Average	24.42	24.96	0.680508	0.706988

Table A.39: Color (Fast) Profile for $\sigma = 90$

Noise Level	Image Name	PSNR		SSIM	
		BM3D	Proposed Method	BM3D	Proposed Method
100	Lenna	25.24	26.22	0.695060	0.755510
	Peppers	24.70	25.51	0.687930	0.751310
	Baboon	20.60	20.51	0.501250	0.468520
	Barbara	24.24	24.66	0.709110	0.727950
	Average	23.69	24.22	0.648338	0.675823

Table A.40: Color (Fast) Profile for $\sigma = 100$

Appendix B

Detail Subjective Experimental Results

In this chapter we present some selective output images from our proposed method as well as from original BM3D for visual inspection. We choose Barbara, Lena and Peppers from Wavelet and Color Profiles and present their outputs for noise standard deviation $\sigma = 40$, $\sigma = 70$ and $\sigma = 100$. The reason for choosing these three values of σ is that $\sigma = 40$ is considered to be maximum value of σ for lower level of noise. We consider $\sigma = 70$ as moderate level of noise and $\sigma = 100$ as extreme level of noise.



(a) True Image

(b) Noise Added with $\sigma = 40$ 

(c) Denoised by BM3D. PSNR=28.90dB, SSIM=0.765.

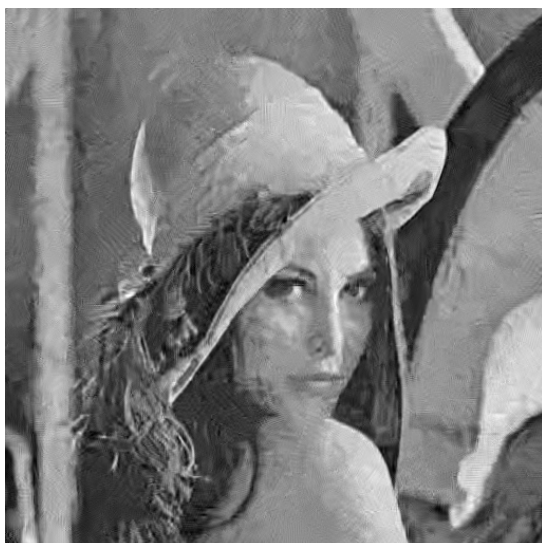


(d) Denoised by Proposed Method. PSNR=29.78dB, SSIM=0.815.

Figure B.1: Wavelet (Normal) Profile Output of Lena Image for $\sigma = 40$



(a) True Image

(b) Noise Added with $\sigma = 70$ (c) Denoised by BM3D. PSNR= $26.37dB$, SSIM= 0.655 .(d) Denoised by Proposed Method. PSNR= $27.49dB$, SSIM= 0.759 .Figure B.2: Wavelet (Normal) Profile Output of Lena Image for $\sigma = 70$



(a) True Image

(b) Noise Added with $\sigma = 100$ (c) Denoised by BM3D. PSNR= $24.25dB$, SSIM= 0.531 .(d) Denoised by Proposed Method. PSNR= $25.85dB$, SSIM= 0.703 .Figure B.3: Wavelet (Normal) Profile Output of Lena Image for $\sigma = 100$



(a) True Image

(b) Noise Added with $\sigma = 40$ (c) Denoised by BM3D. PSNR= $27.46dB$, SSIM= 0.802 .(d) Denoised by Proposed Method. PSNR= $28.00dB$, SSIM= 0.821 .Figure B.4: Wavelet (Normal) Profile Output of Barbara Image for $\sigma = 40$



(a) True Image

(b) Noise Added with $\sigma = 70$ 

(c) Denoised by BM3D. PSNR=25.00dB, SSIM=0.683.



(d) Denoised by Proposed Method. PSNR=25.47dB, SSIM=0.725.

Figure B.5: Wavelet (Normal) Profile Output of Barbara Image for $\sigma = 70$



(a) True Image

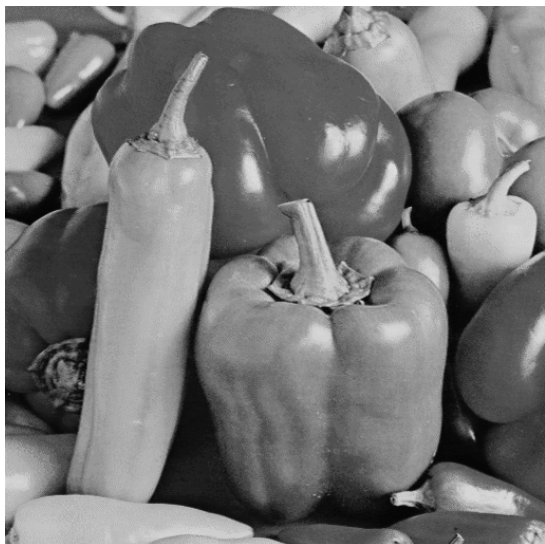
(b) Noise Added with $\sigma = 100$ 

(c) Denoised by BM3D. PSNR=22.84dB, SSIM=0.556.

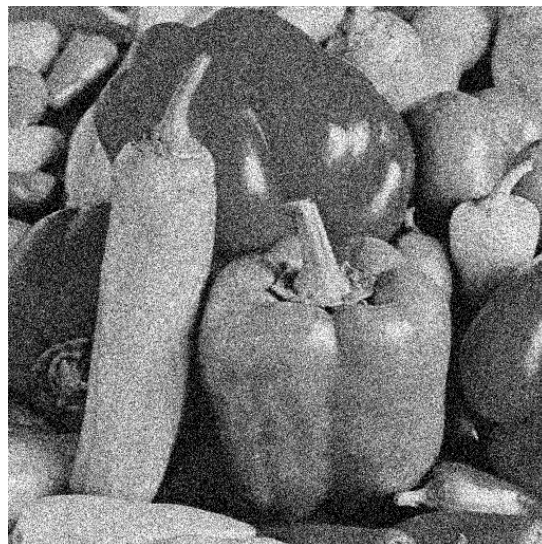


(d) Denoised by Proposed Method. PSNR=23.59dB, SSIM=0.640.

Figure B.6: Wavelet (Normal) Profile Output of Barbara Image for $\sigma = 100$

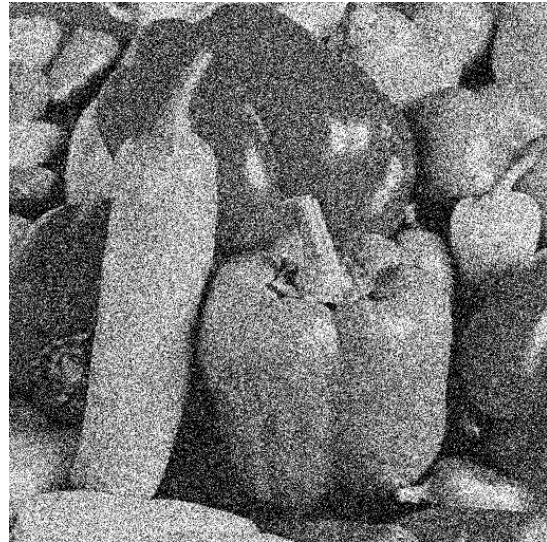
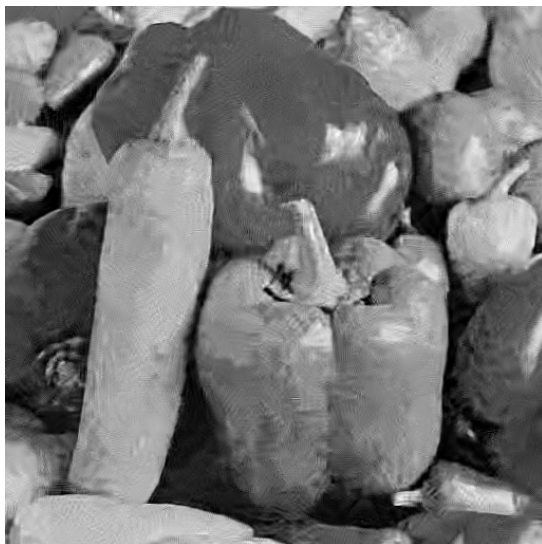


(a) True Image

(b) Noise Added with $\sigma = 40$ (c) Denoised by BM3D. PSNR= $28.73dB$, SSIM= 0.716 .(d) Denoised by Proposed Method. PSNR= $29.65dB$, SSIM= 0.764 .Figure B.7: Wavelet (Normal) Profile Output of Peppers Image for $\sigma = 40$

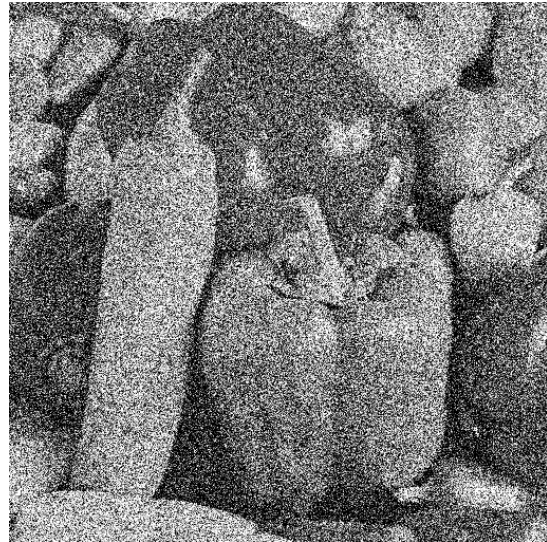
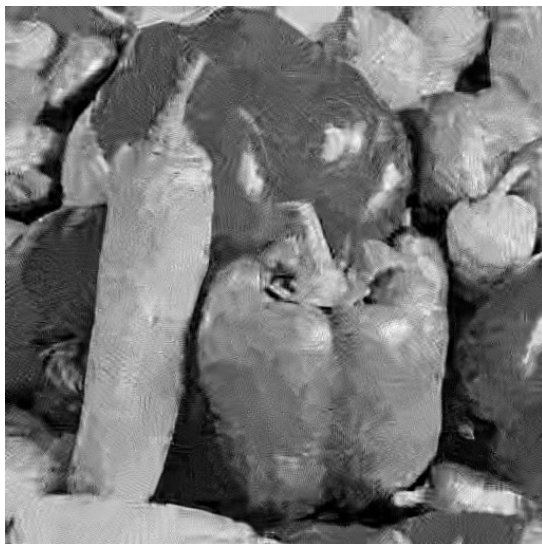


(a) True Image

(b) Noise Added with $\sigma = 70$ (c) Denoised by BM3D. PSNR= $26.26dB$, SSIM= 0.618 .(d) Denoised by Proposed Method. PSNR= $27.47dB$, SSIM= 0.713 .Figure B.8: Wavelet (Normal) Profile Output of Peppers Image for $\sigma = 70$



(a) True Image

(b) Noise Added with $\sigma = 100$ 

(c) Denoised by BM3D. PSNR=24.28dB, SSIM=0.519.



(d) Denoised by Proposed Method. PSNR=25.94dB, SSIM=0.674.

Figure B.9: Wavelet (Normal) Profile Output of Peppers Image for $\sigma = 100$



(a) True Image

(b) Noise Added with $\sigma = 40$ (c) Denoised by BM3D. PSNR= $28.94dB$, SSIM= 0.766 .(d) Denoised by Proposed Method. PSNR= $29.84dB$, SSIM= 0.815 .Figure B.10: Wavelet (Fast) Profile Output of Lena Image for $\sigma = 40$



(a) True Image

(b) Noise Added with $\sigma = 70$ (c) Denoised by BM3D. PSNR= $26.36dB$, SSIM= 0.655 .(d) Denoised by Proposed Method. PSNR= $27.49dB$, SSIM= 0.756 .Figure B.11: Wavelet (Fast) Profile Output of Lena Image for $\sigma = 70$



(a) True Image

(b) Noise Added with $\sigma = 100$ (c) Denoised by BM3D. PSNR= $24.17dB$, SSIM= 0.528 .(d) Denoised by Proposed Method. PSNR= $25.80dB$, SSIM= 0.702 .Figure B.12: Wavelet (Fast) Profile Output of Lena Image for $\sigma = 100$



(a) True Image

(b) Noise Added with $\sigma = 40$ (c) Denoised by BM3D. PSNR= $27.40dB$, SSIM= 0.800 .(d) Denoised by Proposed Method. PSNR= $27.93dB$, SSIM= 0.820 .Figure B.13: Wavelet (Fast) Profile Output of Barbara Image for $\sigma = 40$



(a) True Image

(b) Noise Added with $\sigma = 70$ 

(c) Denoised by BM3D. PSNR=24.90dB, SSIM=0.682.



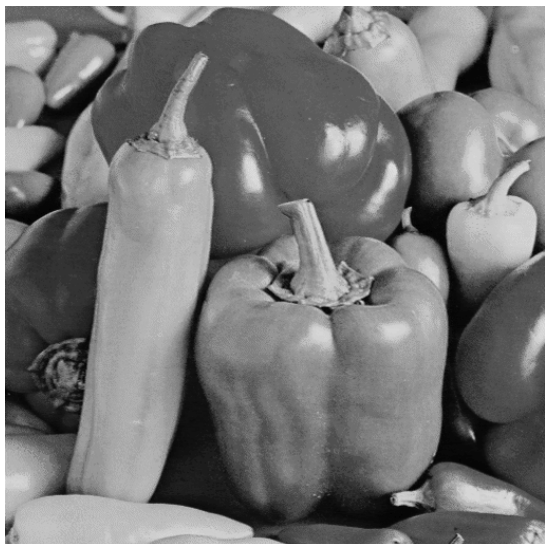
(d) Denoised by Proposed Method. PSNR=25.39dB, SSIM=0.724.

Figure B.14: Wavelet (Fast) Profile Output of Barbara Image for $\sigma = 70$

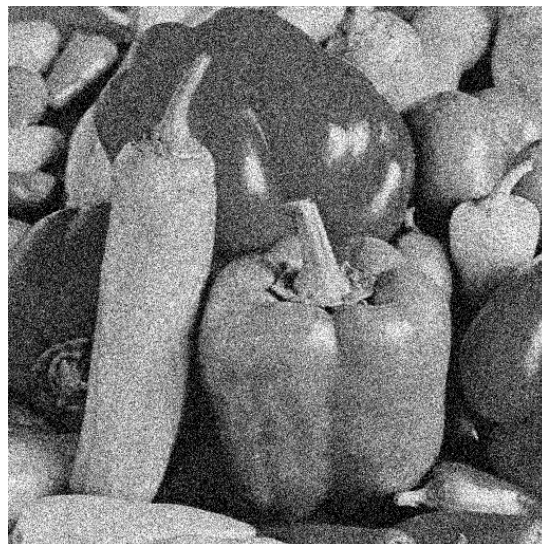
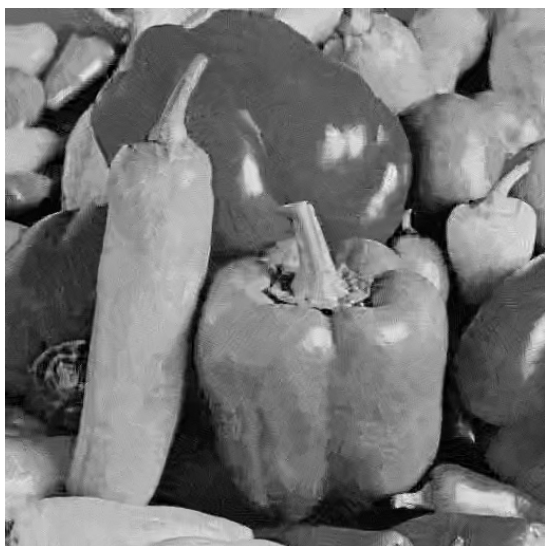


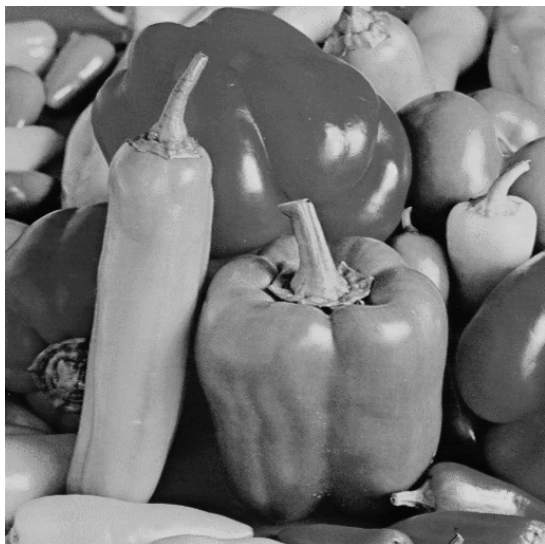
(a) True Image

(b) Noise Added with $\sigma = 100$ (c) Denoised by BM3D. PSNR= $22.79dB$, SSIM= 0.552 .(d) Denoised by Proposed Method. PSNR= $23.62dB$, SSIM= 0.639 .Figure B.15: Wavelet (Fast) Profile Output of Barbara Image for $\sigma = 100$

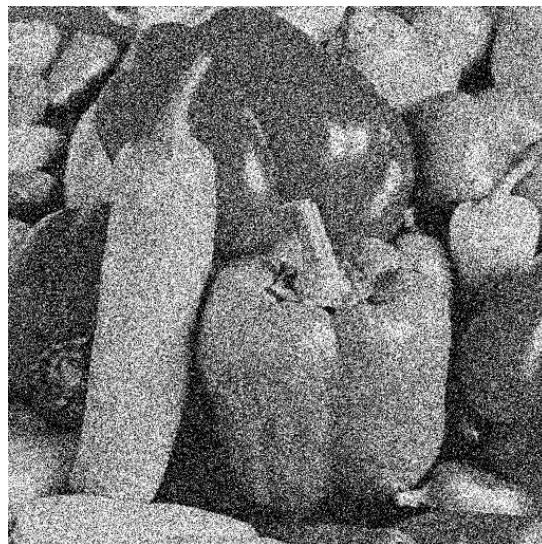
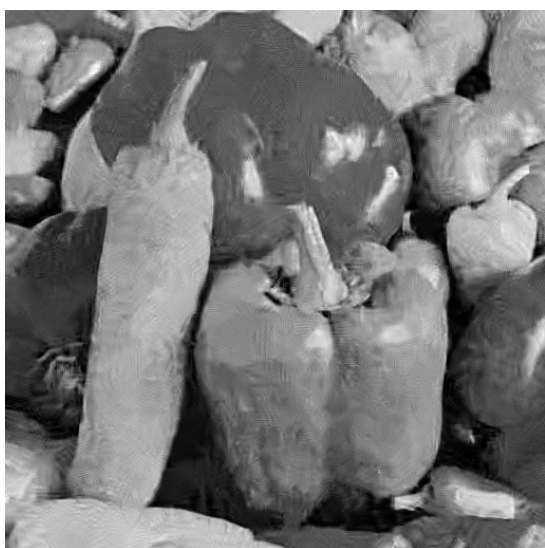


(a) True Image

(b) Noise Added with $\sigma = 40$ (c) Denoised by BM3D. PSNR= $28.73dB$, SSIM= 0.716 .(d) Denoised by Proposed Method. PSNR= $29.65dB$, SSIM= 0.764 .Figure B.16: Wavelet (Fast) Profile Output of Peppers Image for $\sigma = 40$



(a) True Image

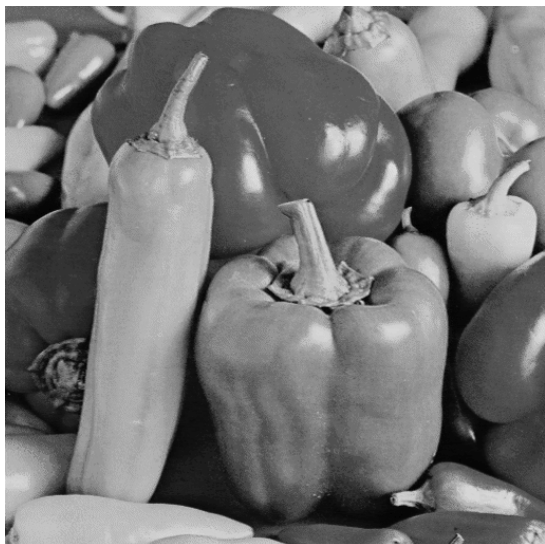
(b) Noise Added with $\sigma = 70$ 

(c) Denoised by BM3D. PSNR=26.26dB, SSIM=0.618.

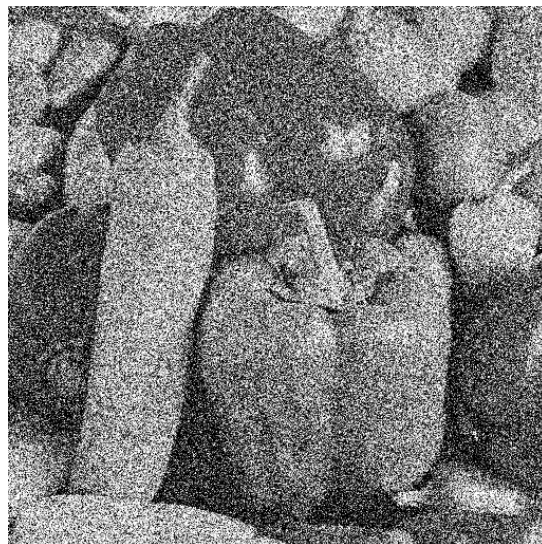


(d) Denoised by Proposed Method. PSNR=27.47dB, SSIM=0.713.

Figure B.17: Wavelet (Fast) Profile Output of Peppers Image for $\sigma = 70$



(a) True Image

(b) Noise Added with $\sigma = 100$ 

(c) Denoised by BM3D. PSNR=24.28dB, SSIM=0.519.

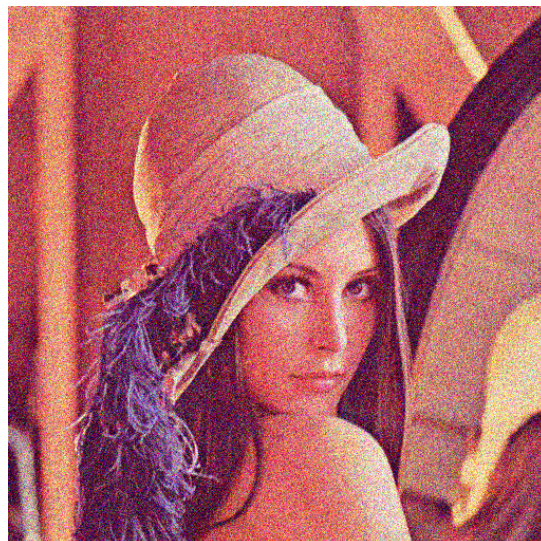


(d) Denoised by Proposed Method. PSNR=25.94dB, SSIM=0.674.

Figure B.18: Wavelet (Fast) Profile Output of Peppers Image for $\sigma = 100$

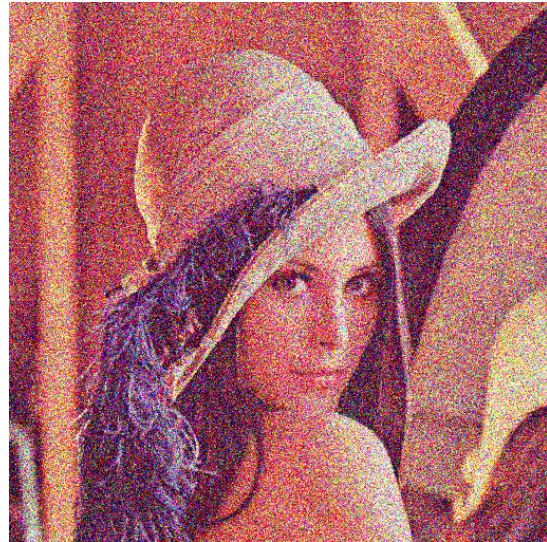
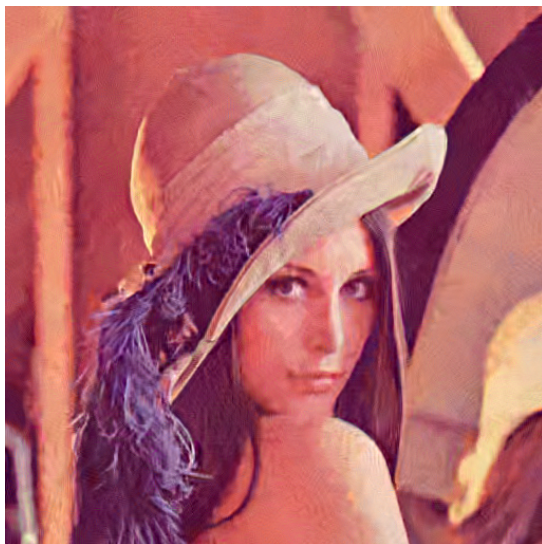
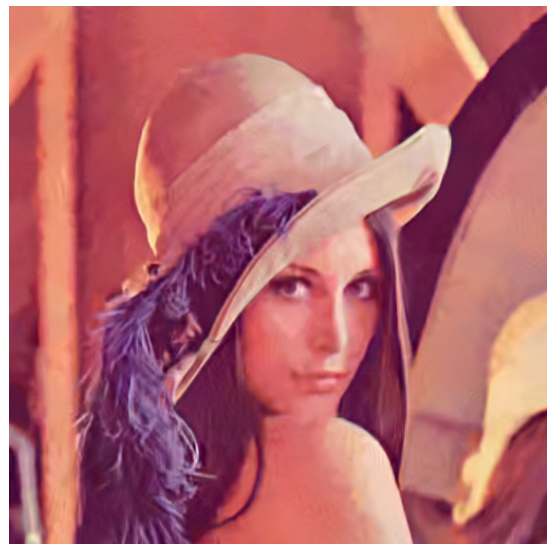


(a) True Image

(b) Noise Added with $\sigma = 40$ (c) Denoised by BM3D. PSNR= $29.73dB$, SSIM= 0.841 .(d) Denoised by Proposed Method. PSNR= $30.13dB$, SSIM= 0.852 .Figure B.19: Color (Normal) Profile Output of Lena Image for $\sigma = 40$

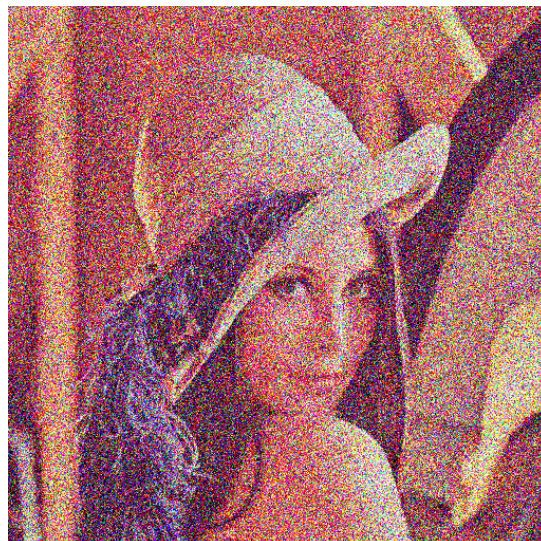
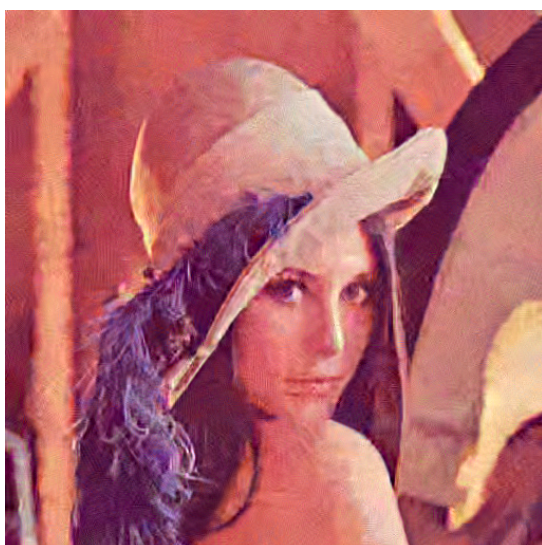


(a) True Image

(b) Noise Added with $\sigma = 70$ (c) Denoised by BM3D. PSNR= $27.93dB$, SSIM= 0.772 .(d) Denoised by Proposed Method. PSNR= $28.58dB$, SSIM= 0.812 .Figure B.20: Color (Normal) Profile Output of Lena Image for $\sigma = 70$



(a) True Image

(b) Noise Added with $\sigma = 100$ (c) Denoised by BM3D. PSNR= $26.02dB$, SSIM= 0.689 .(d) Denoised by Proposed Method. PSNR= $27.06dB$, SSIM= 0.771 .Figure B.21: Color (Normal) Profile Output of Lena Image for $\sigma = 100$



(a) True Image

(b) Noise Added with $\sigma = 40$ 

(c) Denoised by BM3D. PSNR=28.96dB, SSIM=0.813.

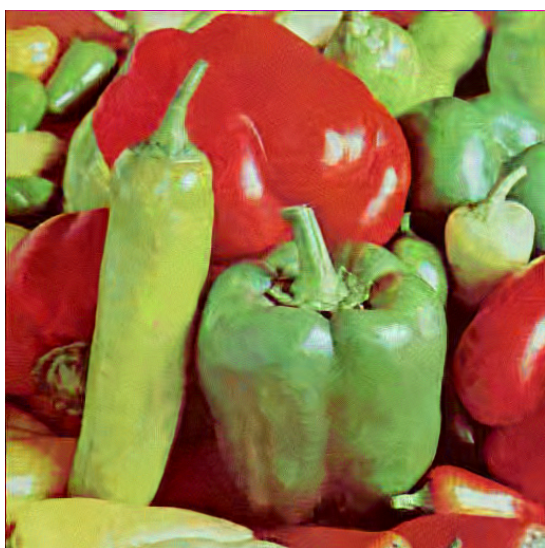


(d) Denoised by Proposed Method. PSNR=29.33dB, SSIM=0.827.

Figure B.22: Color (Normal) Profile Output of Peppers Image for $\sigma = 40$



(a) True Image

(b) Noise Added with $\sigma = 70$ (c) Denoised by BM3D. PSNR= $27.10dB$, SSIM= 0.752 .(d) Denoised by Proposed Method. PSNR= $27.70dB$, SSIM= 0.796 .Figure B.23: Color (Normal) Profile Output of Peppers Image for $\sigma = 70$

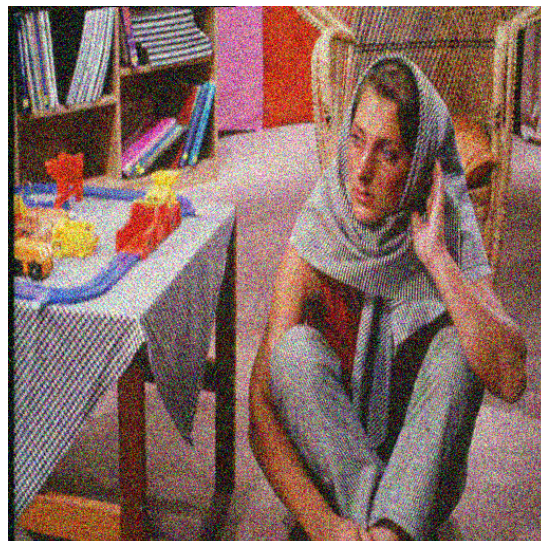


(a) True Image

(b) Noise Added with $\sigma = 100$ (c) Denoised by BM3D. PSNR= $25.17dB$, SSIM= 0.672 .(d) Denoised by Proposed Method. PSNR= $26.05dB$, SSIM= 0.759 .Figure B.24: Color (Normal) Profile Output of Peppers Image for $\sigma = 100$

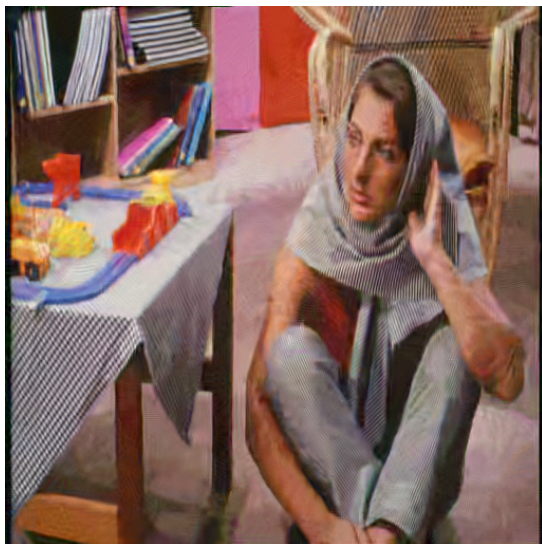


(a) True Image

(b) Noise Added with $\sigma = 40$ (c) Denoised by BM3D. PSNR= $29.09dB$, SSIM= 0.877 .(d) Denoised by Proposed Method. PSNR= $29.22dB$, SSIM= 0.878 .Figure B.25: Color (Normal) Profile Output of Barbara Image for $\sigma = 40$

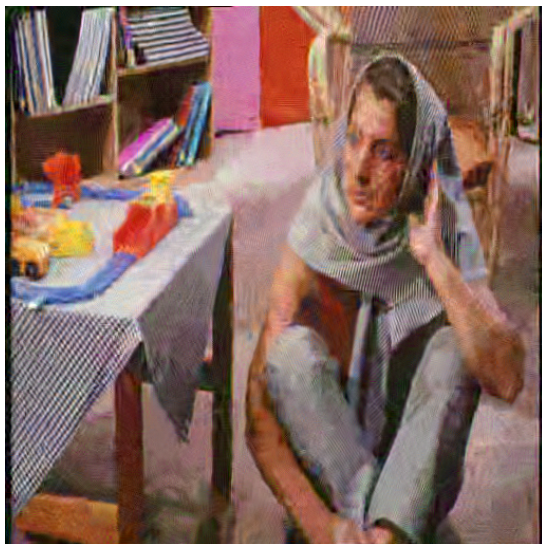


(a) True Image

(b) Noise Added with $\sigma = 70$ (c) Denoised by BM3D. PSNR= $27.20dB$, SSIM= 0.813 .(d) Denoised by Proposed Method. PSNR= $27.51dB$, SSIM= 0.825 .Figure B.26: Color (Normal) Profile Output of Barbara Image for $\sigma = 70$



(a) True Image

(b) Noise Added with $\sigma = 100$ 

(c) Denoised by BM3D. PSNR=24.88dB, SSIM=0.714.

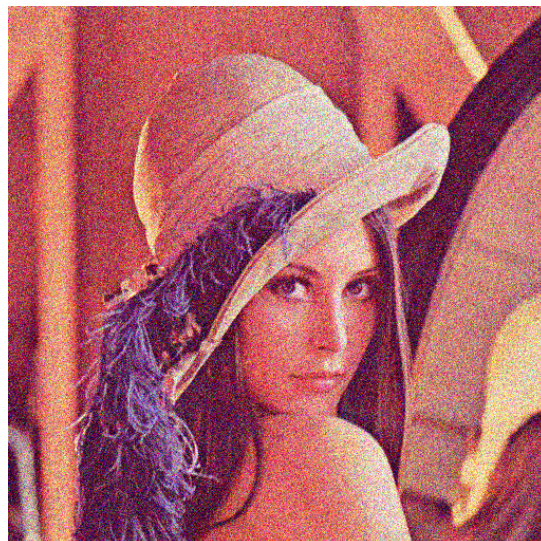


(d) Denoised by Proposed Method. PSNR=25.56dB, SSIM=0.754.

Figure B.27: Color (Normal) Profile Output of Barbara Image for $\sigma = 100$



(a) True Image

(b) Noise Added with $\sigma = 40$ 

(c) Denoised by BM3D. PSNR=28.98dB, SSIM=0.825.

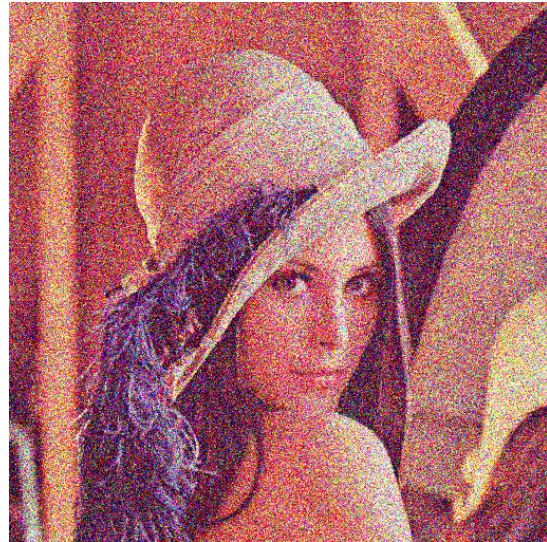


(d) Denoised by Proposed Method. PSNR=29.47dB, SSIM=0.839.

Figure B.28: Color (Fast) Profile Output of Lena Image for $\sigma = 40$

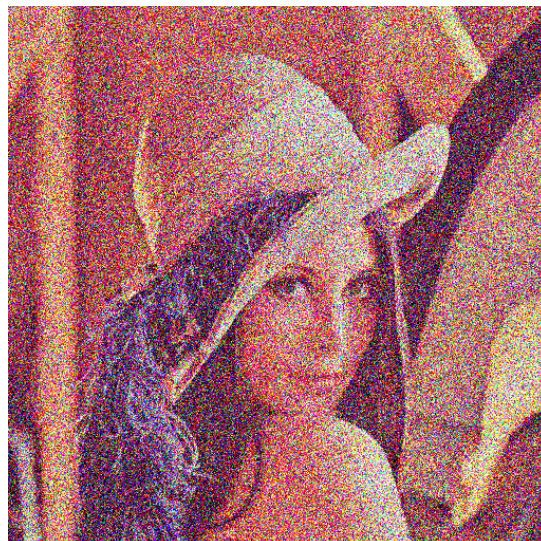
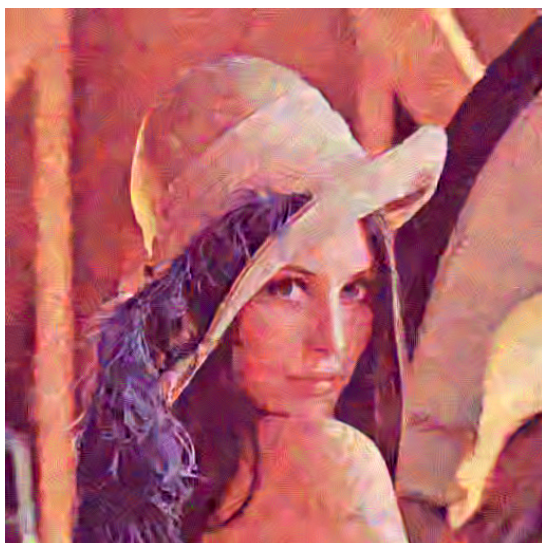


(a) True Image

(b) Noise Added with $\sigma = 70$ (c) Denoised by BM3D. PSNR= $27.54dB$, SSIM= 0.780 .(d) Denoised by Proposed Method. PSNR= $28.13dB$, SSIM= 0.805 .Figure B.29: Color (Fast) Profile Output of Lena Image for $\sigma = 70$



(a) True Image

(b) Noise Added with $\sigma = 100$ 

(c) Denoised by BM3D. PSNR=25.24dB, SSIM=0.695.

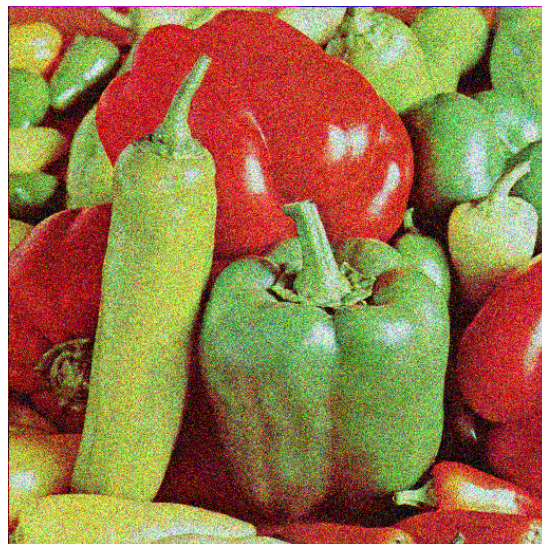


(d) Denoised by Proposed Method. PSNR=26.22dB, SSIM=0.756.

Figure B.30: Color (Fast) Profile Output of Lena Image for $\sigma = 100$



(a) True Image

(b) Noise Added with $\sigma = 40$ (c) Denoised by BM3D. PSNR= $28.32dB$, SSIM= 0.798 .(d) Denoised by Proposed Method. PSNR= $28.72dB$, SSIM= 0.814 .Figure B.31: Color (Fast) Profile Output of Peppers Image for $\sigma = 40$



(a) True Image

(b) Noise Added with $\sigma = 70$ 

(c) Denoised by BM3D. PSNR=26.83dB, SSIM=0.759.

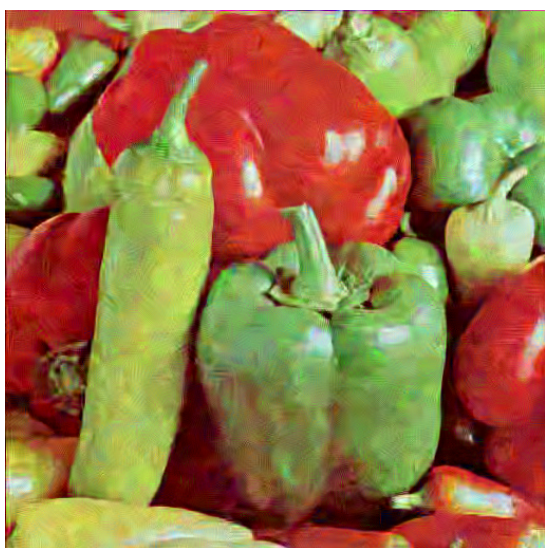


(d) Denoised by Proposed Method. PSNR=27.36dB, SSIM=0.789.

Figure B.32: Color (Fast) Profile Output of Peppers Image for $\sigma = 70$



(a) True Image

(b) Noise Added with $\sigma = 100$ 

(c) Denoised by BM3D. PSNR=24.70dB, SSIM=0.688.

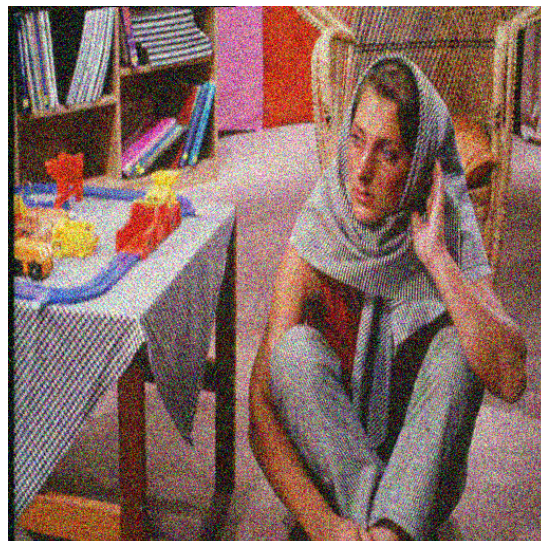
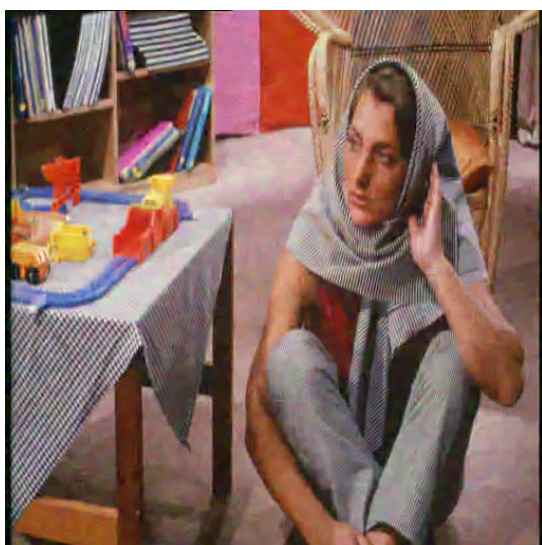


(d) Denoised by Proposed Method. PSNR=25.51dB, SSIM=0.751.

Figure B.33: Color (Fast) Profile Output of Peppers Image for $\sigma = 100$

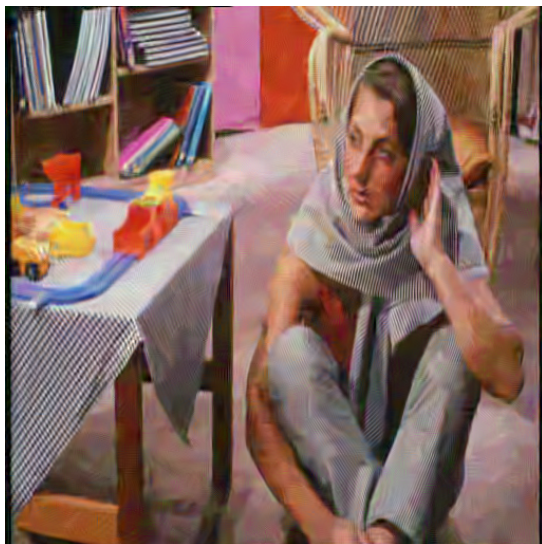


(a) True Image

(b) Noise Added with $\sigma = 40$ (c) Denoised by BM3D. PSNR= $28.34dB$, SSIM= 0.857 .(d) Denoised by Proposed Method. PSNR= $28.55dB$, SSIM= 0.861 .Figure B.34: Color (Fast) Profile Output of Barbara Image for $\sigma = 40$

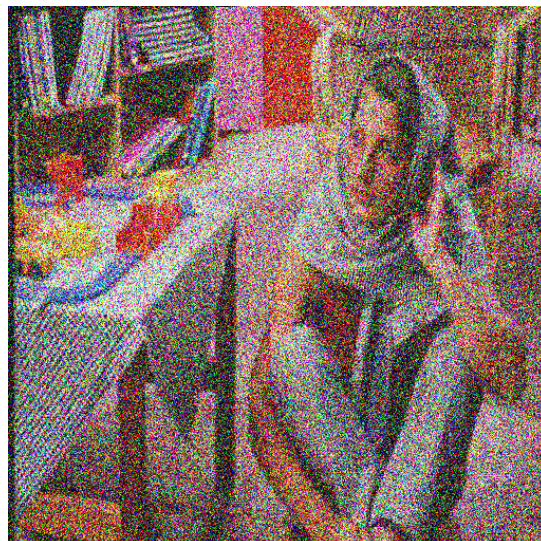
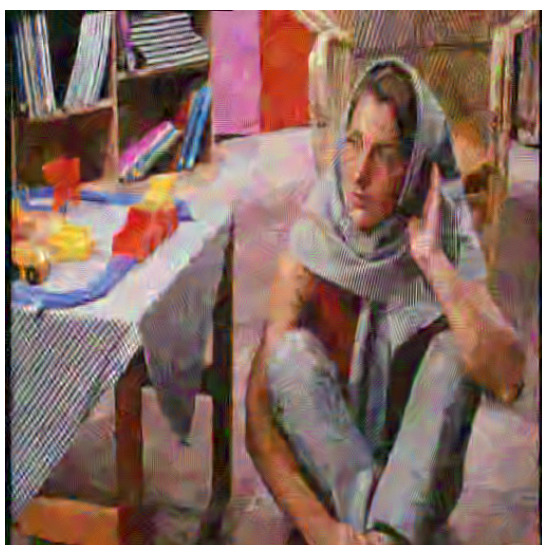


(a) True Image

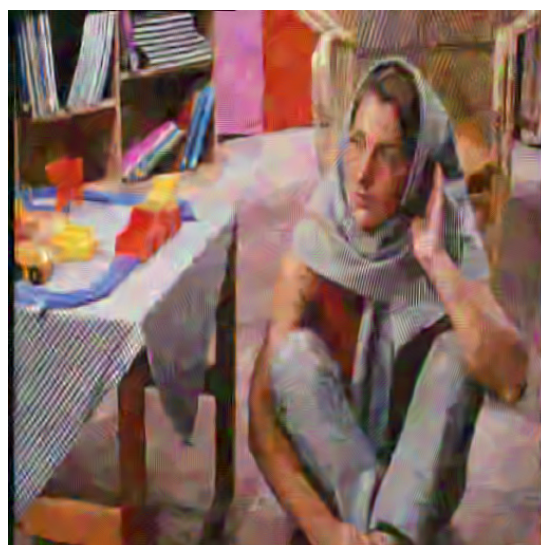
(b) Noise Added with $\sigma = 70$ (c) Denoised by BM3D. PSNR= $26.86dB$, SSIM= 0.811 .(d) Denoised by Proposed Method. PSNR= $27.10dB$, SSIM= 0.816 .Figure B.35: Color (Fast) Profile Output of Barbara Image for $\sigma = 70$



(a) True Image

(b) Noise Added with $\sigma = 100$ 

(c) Denoised by BM3D. PSNR=24.24dB, SSIM=0.709.



(d) Denoised by Proposed Method. PSNR=24.66dB, SSIM=0.728.

Figure B.36: Color (Fast) Profile Output of Barbara Image for $\sigma = 100$

Curriculum Vitae

Name: Mahmud Hasan

Post-Secondary Education and Degrees: Bachelor of Science in Computer Science & Engineering
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)
Chittagong, Bangladesh.
2007 - 2011

Honours and Awards: Dean's Honor List for being Highest CGPA Scorer
CUET, 2011

Related Work Experience: Teaching Assistant & Research Assistant
The University of Western Ontario
2013 - 2014

Lecturer
Department of Allied Engineering
Bangladesh University of Textiles
2012-2013

Lecturer
Department of Computer Science
Stamford University Bangladesh
2011-2012

Publications:

Under Review