

Electronic Thesis and Dissertation Repository

8-24-2020 11:45 AM

Framework For Kernel Based BM3D Algorithm

Mena Abdelrahman Massoud, *The University of Western Ontario*

Supervisor: professor\Mahmoud El-Sakka, *The University of Western Ontario*

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

© Mena Abdelrahman Massoud 2020

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



Part of the [Artificial Intelligence and Robotics Commons](#), and the [Other Computer Sciences Commons](#)

Recommended Citation

Massoud, Mena Abdelrahman, "Framework For Kernel Based BM3D Algorithm" (2020). *Electronic Thesis and Dissertation Repository*. 7314.

<https://ir.lib.uwo.ca/etd/7314>

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.

Abstract

Patch-based approaches such as Block Matching and 3D collaborative Filtering (BM3D) algorithm represent the current state-of-the-art in image denoising. However, BM3D still suffers from degradation in performance in smooth areas as well as loss of image details, specifically in the presence of high noise levels. Integrating shape adaptive methods with BM3D improves the denoising outcome including the visual quality of the denoised image; and also maintains image details. In this study, we proposed a framework that produces multiple images using various shapes. These images were aggregated at the pixel or patch levels for both stages in BM3D, and when appropriately aggregated, resulted in better denoising performance than BM3D by 1.15 dB, on average.

Keywords: Digital Image Processing, Image Restoration, Additive White Gaussian Noise, Image Denoising, Block Matching and 3D Filtering Algorithm (BM3D), Flat kernel, Adaptive Kernel, Adaptive Shape Algorithms.

Lay Abstract

Noise in images usually occurs during image acquisition and/or transmission, when image information can be lost, and because of this, research in denoising digital images focuses on improving image information. BM3D the most prominent image denoising algorithm for the last decade, it utilizes a process of searching for matching patches to improve image quality. Similar to BM3D, Our proposed framework uses different shapes next to the square shape to improve patch matching. However, Instead of obtaining only one output image, our framework combines various obtained images. The combination of these images improves the numerical and visual quality of the denoised imageto a greater extent than BM3D.

Dedication

To my loving parents Abdelrahman Massoud and Wafaa Abdel Wahed
My aunt Mahassen Elgwaily and her loving family
without whom none of my success would have been possible

Acknowledgements

I would like to thank God the almighty for granting me the patience, passion and resilience to complete my thesis successfully.

I also would like to express gratitude to my supervisor, Dr. Mahmoud El-Sakka, for his patience and guidance for almost two years throughout this program. It was my privilege to work under his supervision. In addition to encouraging and supporting, he inspired me to pursue my dreams and goals in life.

Contents

List of Figures	V
List of Tables	X
1 Introduction	1
1.1 Problem Definition	1
1.2 Motivation	3
1.3 Thesis Contribution	3
1.4 Thesis Outline	3
2 literature Review	5
2.1 Additive White Gaussian Noise	5
2.2 Measurements of Image Quality	7
2.2.1 Mean Square Error (MSE)	7
2.2.2 Peak Signal to Noise (PSNR)	7
2.2.3 Structure Similarity Index Method (SSIM)	8
2.3 Dataset	9
2.4 Image Denoising Methods	10
2.5 Spatial Domain Image Denoising	10
2.5.1 Bilateral Filter (BF)	10
2.5.2 A three stage integrated denoising approach for grey scale images	13
2.5.3 Non-local Means (NLM)	13
2.5.4 Statistical Nearest Neighbors for Image Denoising	16
2.5.5 Non-local Methods with Shape-Adaptive Patches (NLM-SAP)	19
2.6 Hybrid Domain Image Denoising	25
2.6.1 Block-matching and 3D filtering (BM3D)	25

2.6.2	Image denoising with morphology-and size-adaptive block-matching transform domain filtering	30
2.6.3	Image denoising based on non-local means filter and its method noise thresholding	33
2.7	Summary	35
3	Methodology	37
3.1	Introduction	37
3.2	Framework Analysis	40
3.3	Proposed Framework	45
4	Results	55
4.1	Numerical Results	55
4.1.1	Discussion on Numerical Results	65
4.2	Visual Results	65
4.2.1	Discussion on Visual Results	76
5	Conclusion and Future Work	77
5.1	Conclusion	77
5.2	Future Work	77
	Bibliography	79

List of Figures

1.1	Digital Image produced in 1921 [1]	1
2.1	Noise levels with sigma equals to 20, 40, 60, and 80, respectively, and the corresponding pdf curves. (a) Noise sigma = 20. (b) AWGN pdf curve for noise with sigma = 20. (c) Noise sigma = 40. (d) AWGN pdf curve for noise with sigma = 40. (e) Noise sigma = 60. (f) AWGN pdf curve for noise with sigma = 60. (g) Noise sigma = 80. (h) AWGN pdf curve for noise with sigma = 80.	6
2.2	Images used in the comparison of performance measures. (a) Baboon image. (b) Cameraman image. (c) Couple image. (d) Barbara image. (e) Lake image. (f) Lena image. (g) Boat image. (h) Goldhill image.	9
2.3	NLM Strategy	15
2.4	Arbitrary patch shapes used in NLM image denoising	20
2.5	Pie patch shape with two angles θ_1 , θ_2 and a radius R to determine the shape structure	21
2.6	Rectangle patch shape with l_0 height, l_r width and θ orientation	21
2.7	Lena Denoising using non-local means (nlm), statistical nearest neighbour (SNN), and fast shape adaptive non-local means (SA-Fast-NLM) approaches at sigma = 20. (a) Original image. (b)Noisy image (PSNR = 22.10 db). (c) Denoised using NLM (PSNR = 31.18 db). (d) Denoised image using SNN (PSNR = 31.22 db). (e) Denoised using SA-Fast-NLM (PSNR = 31.53 db)	24
2.8	BM3D filtering flowchart	25
2.9	BM3D Parameters [24]	29
2.10	Lena image denoised by BM3D at sigma = 20. (a) Original image. (b) Noisy image (PSNR = 22.25). (c) Denoised image (PSNR = 30.23)	30

2.11	Explanation of the algorithm flow of image denoising with morphology and size adaptive block-matching transform domain filtering (SA-BM1-3D) [25]	31
2.12	The stages used in the combination of wavelet thresholding and NLM algorithm	33
3.1	Lena image at sigma = 40, denoised using BM3D algorithm showing disturbance in smooth areas and loss in fine details. (a) Original Image. (b) Noisy image sigma = 40. (c) Denoised after stage one, (d) Denoised after stage two.	38
3.2	Different shaped Gaussian Kernels:(a) Square flat kernel.(b) Numerical representation of square flat kernel. (c) Circular flat kernel. (d) Numerical representation of circular flat kernel. (e) Horizontal flat kernel. (f) Numerical representation of horizontal flat kernel.(g) Vertical flat kernel. (h) Numerical representation of vertical flat kernel. (i) D45 flat kernel. (j) Numerical representation of D45 flat kernel. (k)D135 flat kernel. (l) Numerical representation of D135 flat kernel. (m) Kaiser Kernel. (n) Numerical representation of kaiser kernel.	39
3.3	Lena image denoised by BM3D using different shaped kernels for stage one: (a) Original image. (b) Noisy image with sigma = 20 (PSNR = 22.10). (c) Original BM3D stage one denoised image (PSNR = 32.30). (d) square kernel (PSNR = 32.19). (e) Circular kernel (PSNR = 32.12). (f) Horizontal kernel (PSNR = 32.16). (g) Vertical kernel (psnr = 32.15). (h) Diagonal with angle 135 kernel (PSNR = 32.20). (i) Diagonal with angle 45 kernel (PSNR = 32.16).	42
3.4	Lena image denoised by BM3D using different shaped kernels for stage two: (a) Original image. (b) Noisy image with sigma = 20 (PSNR = 22.10). (c) Original BM3D stage two denoised image (PSNR = 32.83). (d) Square kernel (PSNR = 33.43). (e) Circular kernel (PSNR = 33.44). (f) Horizontal kernel (PSNR = 33.44). (g) Vertical kernel (PSNR = 33.42). (h) Diagonal with angle 135 kernel (PSNR = 33.44). (i) Diagonal with angle 45 kernel (PSNR = 33.45).	44
3.5	Stage one for the proposed framework	46
3.6	Stage two for the proposed framework	47
3.7	Lena image, after stage one, comprising six images that represent the locations of the pixels that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.	49

3.8	Lena image, after stage two, comprising six images that represent the locations of the pixels that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.	50
3.9	Lena image, after stage one, comprising six images that represent the locations of the patches that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.	51
3.10	Lena image, after stage two, comprising six images that represent the locations of the patches that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.	52
3.11	Denoised Lena image results using the proposed framework. (a) Original image. (b) Noisy image with $\sigma = 20$ (PSNR = 22.14). (c) Original BM3D after stage one. (PSNR = 32.23) (d) Original BM3D after stage two (PSNR = 33.77). Pixel based results, (e) Proposed framework after stage one (PSNR = 33.69). (f) Proposed framework after stage two (PSNR = 33.99). Patch based results, (g) Proposed framework after stage one (PSNR = 32.74). (h) Proposed framework after stage two (PSNR = 33.26)	54
4.1	Barbara image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.	56
4.2	Baboon image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.	57
4.3	Boat image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.	58
4.4	Cameraman image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.	59
4.5	Goldhill image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.	60
4.6	Lena image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.	61

4.7	Couple image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.	62
4.8	Lake image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.	63
4.9	Average PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.	64
4.10	Denoising Couple image with sigma = 10 using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	66
4.11	Denoising Couple image with sigma = 20 using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	67
4.12	Denoising Couple image with sigma = 30 using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	68
4.13	Denoising Couple image with sigma = 40 using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	69
4.14	Denoising Couple image with sigma = 50 using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	70
4.15	Denoising Couple image with sigma = 60 using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	71

4.16	Denoising Couple image with $\sigma = 70$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	72
4.17	Denoising Couple image with $\sigma = 80$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	73
4.18	Denoising Couple image with $\sigma = 90$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	74
4.19	Denoising Couple image with $\sigma = 100$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.	75

List of Tables

2.1	PSNR comparison between the noisy images and NLM denoising performance at different noise levels ($\sigma = 20, 40, 60, 80$	16
2.2	PSNR comparison between noisy, non-local means (Original NLM), nearest neighbour (NN) and statistical nearest neighbour (SNN) methods for different noise (σ) values.	18
2.3	Images PSNR comparison between noisy, original nlm (Orig NLM), original shape adaptive patches (Orig-SAP), fast shape adaptive patches (fastSAP) and 8 Gaussian shapes (8 Gaus.dir.) for different σ noise levels 20, 40, 60 and 80	23
2.4	PSNR comparison between noisy image and BM3D denoised results for 20, 40, 60, 80 σ noise levels	29
2.5	Compared PSNR of noisy, BM3D and shape adaptive BM3D with various noise σ 5, 15, 25 and 35.	32
2.6	Comparison of PSNR for noisy, non-local means (NLM), non local means nearest neighbour (nlmNN), non-local means statistical nearest neighbour (nlmSNN), shape adaptive non-local means (Original-SAP) and Block Matching and 3D Collaborative Filtering (BM3D) for multiple noise levels 20, 40, 60, and 80. . . .	35
4.1	Barbara image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 σ noise levels.	56
4.2	Barbara image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 σ noise levels.	56
4.3	Baboon image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 σ noise levels.	57
4.4	Baboon image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 σ noise levels.	57

4.5	Boat image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.	58
4.6	Boat image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.	58
4.7	Cameraman image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.	59
4.8	Cameraman image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.	59
4.9	Goldhill image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.	60
4.10	Goldhill image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.	60
4.11	Lena image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.	61
4.12	Lena image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.	61
4.13	Couple image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.	62
4.14	Couple image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.	62
4.15	Lake image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.	63
4.16	Lake image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.	63
4.17	Average PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.	64
4.18	Average PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.	64

1 Introduction

1.1 Problem Definition

The field of digital image processing started evolving early in the year 1920s when associated with the newspaper industry. One of the biggest problems in this field is the quality of the digital images. Fig. (1.1) shows the first picture sent through a submerged cable in the Atlantic ocean between New York and London, UK. There is a disturbance in all the grey intensity levels of the image. Even though digital image processing evolved greatly since this picture, it still faces many challenges nowadays.



Figure 1.1: Digital Image produced in 1921 [1]

The primary source of noise in digital images emerges either during image acquisition or

transmission or both. Image acquisition rely on imaging sensors that are sensitive to bright light. Same as the role of the light receptors that are located in the retina membrane inside the human eye.

Imaging devices, for example, Charge-coupled device (CCD), which is a type of camera sensors, contains a sensor presented as a 2-D array of several million tiny solar cells. Each cell in this grid corresponds to a pixel in the digital image. The light from the object is reflected in the sensor grid allowing the solar cells to estimate the number of photons projected in each cell. Finally, the analog-to-digital converter (ADC) turns each of these estimated charges into a digital pixel value. In more detail, the cell in the sensor subjected to more light produces a high photon estimated value, which is converted to a high-intensity value. In greyscale images, the pixel will have a value close to 255 (white) and vice versa. Logically, increasing the number of cells in the sensor increases the resolution and accuracy of the details in the image to perfectly match the scene. However, this is not a hundred percent fact due to several reasons.

On one hand, the image acquisition environment involves a lot of variables such as light levels and sensor quality. Those factors affect the estimation of the final pixel values in the images causing noise and artifacts to appear in the picture. On the other hand, image transmission through wired or wireless networks causes variations in the real pixel values due to atmospheric conditions or in general flows in the medium of transformation.

Noise is divided into two types: dependent and independent noise. Dependent noise depends on the pixel value, like multiplicative noise. While, independent noise is uncorrelated with the pixel intensity like white Gaussian noise, rayleigh noise, erlang (gamma) noise, exponential noise, and salt-and-pepper noise. Different noise models are simulated by generating an array that is similar in size to the given image. The intensity values are stochastic numbers with specific probability density function except for salt and pepper noise.

In real-world applications, dealing with noisy images is a much harder problem. First, images captured usually contain multiple noise models. For instance, Images obtained using

satellite imaging include speckle noise, Gaussian noise, and impulse noise. Secondly, choosing the appropriate denoising approach requires prior knowledge of the noise model(s) present in the image. Finally, assuming that the noise model is known, many of the current denoising methods cause either loss in some of the image details, blurring, disturbance in smooth areas or ringing artifacts around the edges.

1.2 Motivation

Additive white Gaussian noise (AWGN) is one of the most common types of noise in images. AWGN is randomly distributed over images.

Block matching and 3D collaborative filtering (BM3D) is a state-of-art patch-based algorithm. BM3D is a two-stage algorithm. In both stages, block matching is applied between neighbouring patches. Because of the existing noise, the matching process does not always give good results. Consequently, BM3D still suffers from degradation in the denoising performance at high noise levels. The degradation appears as a disturbance in smooth areas and data loss in edge and texture areas.

1.3 Thesis Contribution

This research aims to develop a framework for BM3D that modifying the patch matching step, where multiple geometric shape kernels are utilized to generate multiple denoised BM3D images. These images are aggregated, either at the pixel or patch level. We are aiming in the future to aggregate these images automatically.

1.4 Thesis Outline

Our thesis is divided into five chapters, as follows:

- Chapter One: Discussing some of the image denoising problems and applications.
- Chapter Two: Explaining in detail some image denoising algorithms.
- Chapter Three: Discussing the proposed methodology.
- Chapter Four: Presenting the experimental results of our proposed framework and comparing it to current state-of-art methods.
- Chapter Five: Offering thesis conclusions and future work.

2 literature Review

In this chapter, additive white Gaussian noise is explained. Then a brief explanation for image quality assessment methods. Finally, some of the methods proposed by researches are discussed and compared to show the improvement in the quality of image denoising.

2.1 Additive White Gaussian Noise

Gaussian noise occurs in images due to changes in the values of the original signal. It is called white noise. The naming came from the fact that the Fourier spectrum of this noise model is constant. Meaning that it carries the physical properties of the white light, which is composed of equal proportions of the color spectrum. AWGN is simulated in grey images as a disturbance in the grey values of image pixels, caused by adding random values to the pixel values, as shown in Equ. (2.1),

$$I_{nse} = I + \sigma \times rand(m \times n) \quad (2.1)$$

I_{nse} is the AWGN image, σ is the standard deviation, $m \times n$ represents the size of the image. $rand$ is a function that generates normal (Gaussian) distributed variables that can be represented by the following probability density function (pdf) function. The noise and its corresponding PDF curve is shown in Fig. (2.1),

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (2.2)$$

Where x refers to a pixel in the image, μ is the mean of all pixels, finally, σ^2 is the variance. In the case of Gaussian noise, the mean is equal to zero. Images with AWGN are presented in Fig. (2.1). The x-axis shows the increase in the range due to the increase of standard deviation.

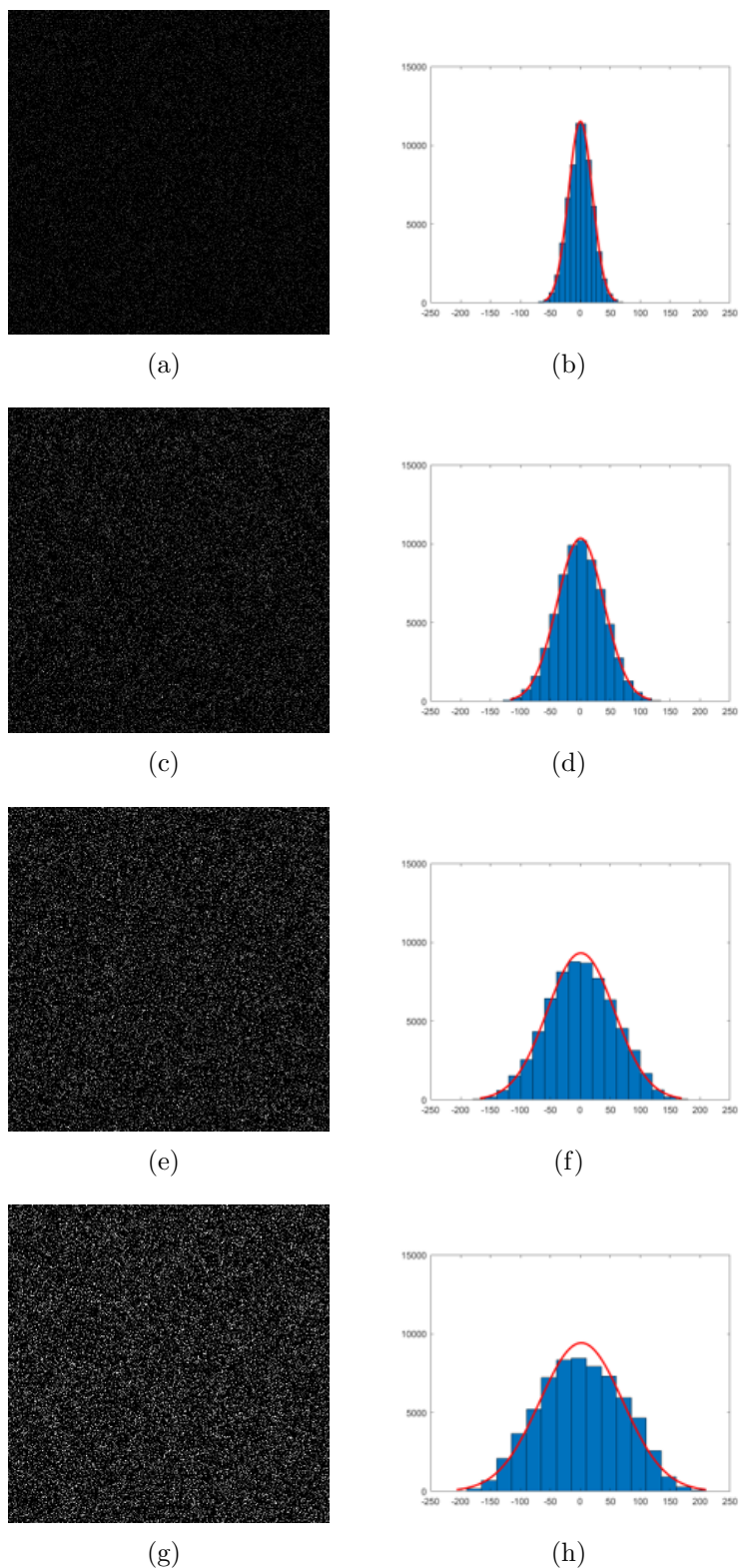


Figure 2.1: Noise levels with sigma equals to 20, 40, 60, and 80, respectively, and the corresponding pdf curves. (a) Noise sigma = 20. (b) AWGN pdf curve for noise with sigma = 20. (c) Noise sigma = 40. (d) AWGN pdf curve for noise with sigma = 40. (e) Noise sigma = 60. (f) AWGN pdf curve for noise with sigma = 60. (g) Noise sigma = 80. (h) AWGN pdf curve for noise with sigma = 80.

2.2 Measurements of Image Quality

There are many image quality assessment metrics that are used to evaluate and measure the quality of the image after processing. The image quality is measured by methods like Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structured Similarity Index Method (SSIM) . Despite all these metrics Human Vision System (HVS) cannot be underestimated in evaluating image quality. In this paper, The approaches are evaluated using PSNR .

2.2.1 Mean Square Error (MSE)

MSE is one of the most common measurements of image quality. The MSE is the variance in the estimated value of the processed image from the original clean image; meaning MSE is the difference between the estimated value and the expected value. The closer the value of MSE to zero the higher the image quality. The MSE function is defined as Equ. (2.3).

$$MSE = \frac{1}{MN} \sum_{n=1}^M \sum_{m=1}^M [\hat{g}(n, m) - g(n, m)]^2 \quad (2.3)$$

2.2.2 Peak Signal to Noise (PSNR)

PSNR is used to calculate logarithm of the ratio between the maximum signal power, estimated by 255, and the power of the noise alternations with affects the quality of the information. PSNR is the commonly used to estimate the quality of the reconstructed image quality as close as possible to the human vision estimation. The PSNR relation is inverse to MSE, so the higher the PSNR value the better the image quality and vice versa. PSNR is defined as the following Equ. (2.4).

$$PSNR = 10 \log_{10} \frac{(peakval)^2}{MSE} \quad (2.4)$$

2.2.3 Structure Similarity Index Method (SSIM)

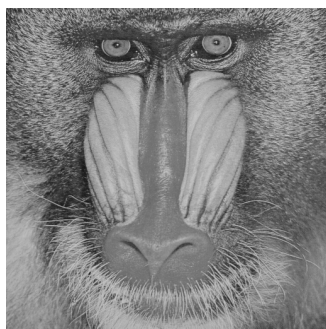
In this method, image degradation is dealt with as the change in the structural data in the image. The structural data refer to spatial closed pixels, contrast and luminance information. SSIM is used to measure the structure similarity not just the intensity level similarity, which can be some times miss leading. SSIM gives a value between 0 and 1, if the SSIM value is closer to 1 this indicates a higher similarity when compared to the original image, hence a better quality and vice versa. SSIM is calculated by Equ. (2.5).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.5)$$

μ_x and μ_y refer to the means. σ_x and σ_y refer to the standard deviation and σ_{xy} is the cross-covariance for given x, y images.

2.3 Dataset

We use eight images to explain the performance of the discussed algorithms. The used images are composed of both smooth and texture area. Images are shown in Fig. (2.2).



(a)



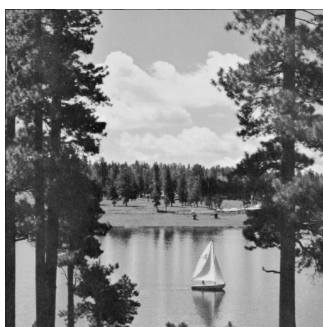
(b)



(c)



(d)



(e)



(f)



(g)



(h)

Figure 2.2: Images used in the comparison of performance measures. (a) Baboon image. (b) Cameraman image. (c) Couple image. (d) Barbara image. (e) Lake image. (f) Lena image. (g) Boat image. (h) Goldhill image.

2.4 Image Denoising Methods

Image denoising have applications with huge significance. For instance, medical image denoising [2], [3], [4] and remote sensing image denoising [5]. In addition to denoising is used as a pre-processing for classification, segmentation, edge detection,...,etc.

In this chapter, image denoising techniques are divided into two categories: spatial domain, and methods that combine spatial and transform (hybrid) domain denoising. Later, some of the proposed methods to improve the denoising performance are discussed. In this study low level noise refers to noisy images with $\sigma = 10, 20, \text{ and } 30$. While high level noise refers to noisy images with $\sigma = 40, 50, 60, 70, 80, 90, \text{ and } 100$.

2.5 Spatial Domain Image Denoising

Denoising images in the spatial domain are divided into two categories: Linear and non-linear. An averaging filter is an example of a linear filter while the median filter is a non-linear filter. State-of-art spatial denoising methods mainly classified as linear methods. The linear method can be sub-divided into two categories: pixel-based, such as Bilateral Filter(BL) and patch-based, such as Non-Local Means (NLM). The idea of most of these algorithms is about replacing the noise pixel with the average value of the neighboring pixels. Spatial Domain algorithms are good in preserving most of the high contrast image features but does not preserve low contrast details.

2.5.1 Bilateral Filter (BF)

Bilateral filter firstly introduced [6] by Tomasi and Manduchi. BF is a non-linear smoothing edge-preserving filter used in image denoising. The idea of BF involves both the spatial domain and range domain. BF takes advantage of having close similar pixels in the image. First, the

spatial domain filter applied to the image f is represented by Equ. (2.6)

$$h(x) = \frac{1}{K_d(x)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) d(\xi) \quad (2.6)$$

$$K_d(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) d(\xi)$$

Where $k_d(x)$ is the normalization factor, $c(\xi, x)$ is the Gaussian shift-invariant function of the Euclidean distance between the center pixel x and the nearby pixel ξ to represent the closeness between those pixels. As shown in Equ. (2.7)

$$c(\xi, x) = e^{-\frac{1}{2} \left(\frac{d(\xi, x)}{\sigma_d} \right)^2}, \quad d(\xi, x) = d(x, \xi) = \|\xi - x\| \quad (2.7)$$

Second, range domain filter is shown as Equ. (2.8)

$$h(x) = \frac{1}{K_r(x)} \int_{-\infty}^{\infty} f(\xi) s(f(\xi), f(x)) d(\xi) \quad (2.8)$$

$$K_r(x) = \int_{-\infty}^{\infty} f(\xi) s(f(\xi), f(x)) d(\xi)$$

Where $s(f(\xi), f(x)) d(\xi)$ is the photometric(intensity) similarity between the center pixel and another nearby pixel. As shown in Equ. (2.9). In which $\delta(\phi, f)$ is the distance between similar intensity values ϕ, f .

$$s(\xi, x) = e^{-\frac{1}{2} \left(\frac{\delta(f(\xi), f(x))}{\sigma_r} \right)^2}, \quad \delta(\phi, f) = \delta(\phi - f) = \|\phi - f\| \quad (2.9)$$

Both the spatial and range domain filters are integrated to present the bilateral filter in Equ. (2.10), in which the pixel is replaced by the average of the nearby pixels that have almost the same brightness.

$$h(x) = \frac{1}{K(x)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d(\xi) \quad (2.10)$$

$$K(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d(\xi)$$

BF is considered a good edge-preserving denoising method. However, it is not robust and highly sensitive to noise, especially high noise levels. Due to the previous reason artifacts like staircase effect and false edges appears in the image. Besides, it is computationally expensive to scan

and apply the filter to each pixel in the image. The range kernel is the reason behind the high computational burden as the averaging process of the intensities is non-linear.

2.5.2 A three stage integrated denoising approach for grey scale images

Goyal et al. [7] introduce a new method by integrating BF and bit slicing to denoise images. The technique is composed of the following three steps:

- Step one: An estimate of the denoised image is generated by applying the following Equ. (2.11)

$$\hat{I}_{\gamma 1} = \theta_1 \hat{I}_{\gamma a} + \theta_2 \hat{I}_{\gamma b} \quad (2.11)$$

$\hat{I}_{\gamma a}$ is the denoised input image using weighted bilateral filter (WBF), $\hat{I}_{\gamma b}$ is the input image denoised by applying the standard bilateral filter (SBF) followed by applying the robust bilateral filter (RBF). and θ_1, θ_2 refer to the weights adjusted using SURE unbiased risk estimator to combine those two results into a final single denoised image.

- Step two: Anisotropic diffusion filter (ADF) is applied using a partial differential equation (PDE) on the residual noise image for two main reasons. First, ADF is used to remove artifacts. Second, reduce the blurriness caused by averaging Gaussian filter.
- Step Three: During the inspection of the 8-bit planes (0-7), it is noticed that the noise is mainly concentrated in the lower bit planes. While the high bit planes contain most of the image features. After testing, denoising the three lowest order bit planes using bitonic filter while leaving 3,4,5,6 and 7-bit planes unchanged gives the best denoising results while preserving most of the image small features.

2.5.3 Non-local Means (NLM)

Non-local Means is an image de-noising filter that relies on patch matching between close regions in the same image [8],[9],[10] and [11]. Given the noisy image v , such that $v = v(i)|i \in I$. The weighted average multiplied by noisy pixel at spatial location j is shown by Equ. (2.12).

$$NLu[v](i) = \sum_{j \in I} w(i, j) v(j) \quad (2.12)$$

The weighted average relies on the similarity between two patches one centered by the pixel of interest i the other is centered by a neighbouring pixel j . These weights are defined as,

$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{\|v(N_i) - v(N_j)\|_{2,\alpha}^2}{h^2}} \quad (2.13)$$

$$Z(i) = \sum_j e^{-\frac{\|v(N_i) - v(N_j)\|_{2,\alpha}^2}{h^2}}$$

Where h is the degree of filtering used to control the amount of exponential decay. Practically, h takes the value of the variance in the image. The weight w outputs values within the range $[0,1]$. $v(N_i)$ and $v(N_j)$ represent square patch of certain size centered by pixel i and j respectively, contained in a search window of a pre-determined size.

The w function that assigns weights to numerically represent the similarity between patches those weights relies on the Euclidean distance as an input. The Euclidean distance is evaluated as the difference of intensity between corresponding values in those two patches. If the distance is small then the Gaussian function output weight is large and vice versa, mathematically the distance is represented by $\|v(N_i) - v(N_j)\|_{2,\alpha}^2$, where $a > 0$ is the standard deviation of the Gaussian kernel. For instance, in Fig. (2.3) the patch centered by pixel p which is the pixel of interest and another patch centered by pixel q_i the distance between them is measured. Then the Gaussian weight function which is the line between patches is the descriptor distance. In other words, the weight of the patch given the distance value. Finally, All those patches get multiplied by there weights then added to get the final average and multiply by the value of the pixel of interest. NLM shows more robustness towards noise than BF and produces improved results.

NLM is considered to be an important reference in image denoising methods due to its simplicity, good numerical and visual results shown in table (2.1)¹ and Fig. (2.7). NLM as a data-adaptive algorithm faces huge problems in estimating the parameters. First, NLM does not preserve low contrast details. Second, the algorithm is computationally expensive when running for each pixel in the image. Consequently, one solution is searching for similar patches

¹This results are obtained from the code written by Jose Vicente Manjon-Herrera <https://www.mathworks.com/matlabcentral/fileexchange/13176-non-local-means-filter>

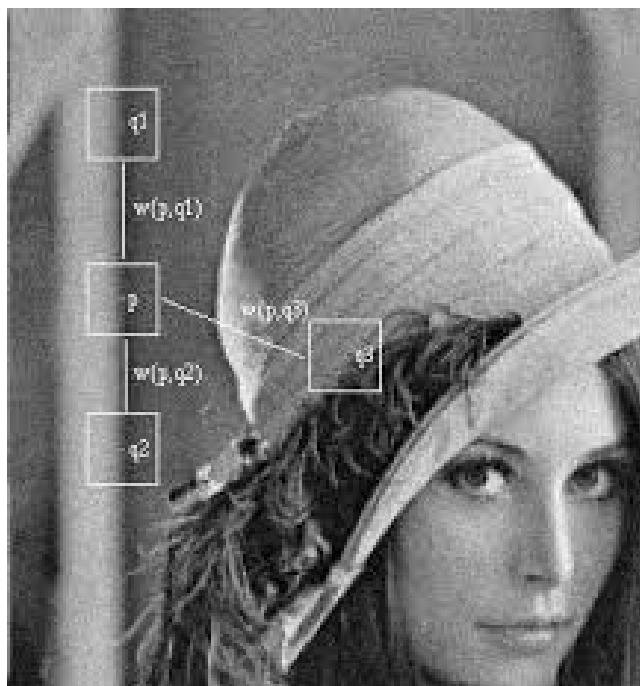


Figure 2.3: NLM Strategy

that occur within a search window boundaries. By default the search window of size 21×21 and patch size of size 7×7 . These parameters are used to look for similarities in the whole image causing the reduction of the computational burden from $O(N^2M^2W^2)$ to $O(R^2NMW^2)$, where the image of size $N \times M$, W^2 is the number of elements in the patch and R is the search window. Thirdly, when no matching patches found during search then the pixel keeps its noisy value causing problems like ringing artifacts around the edges, and the images still contain low noise levels.

NLM modifications are mainly concentrated on four aspects:

1. Approaches that address the high time consumption problem in NLM, this problem is tackled by many methods. For instance, methods that avoid unnecessary calculations for the distances and weights of unmatching patches, another is speeding up by using Fourier transform (FFT) [12]. Finally, approaches that use dictionaries and tree structures to speed the search for similar patches[13].
2. Methods that improve the similarity metric. For example, shape adaptive, rotation-invariant approaches [14] and approaches that consider the permutations of texture in patches[15].

3. Methods that improve the weight of similar patches by using other kernels like flat kernel [16], Euclidean median [17], Hausdorff distance [18], or adjusted Gaussian weight coefficients with Laplace operator[19].
4. Methods to optimize NLM parameters: patch size, search window, and smoothing parameter using Stein’s unbiased risk estimator (SURE) [20].

Table 2.1: PSNR comparison between the noisy images and NLM denoising performance at different noise levels (sigma) = 20, 40, 60, 80

Sigma	Method	baboon	barbara	boats	camera	couple	goldhill	lake	lena
20	Noisy	22.25	22.10	22.25	22.45	22.23	22.14	22.10	22.12
	NLM	29.15	27.59	30.28	29.40	29.37	28.42	31.18	28.75
40	Noisy	16.41	16.13	16.38	16.67	16.57	16.64	16.26	16.28
	NLM	24.50	24.92	25.95	25.60	25.63	24.91	27.25	25.03
60	Noisy	13.34	13.00	13.24	13.54	13.46	13.61	13.15	13.15
	NLM	21.92	23.60	23.46	22.68	23.30	22.64	24.79	23.10
80	Noisy	11.46	11.14	11.36	11.56	11.53	11.74	11.29	11.31
	NLM	20.21	22.68	21.97	20.70	21.66	20.76	23.32	21.82

2.5.4 Statistical Nearest Neighbors for Image Denoising

A new approach [21] is introduced to improve the quality of the patches extracted during patch matching. Hence improve the quality of the output. The reduction in the number of matching patches can help in reducing the loss in details and time complexity, but it can also cause low-frequency artifacts. The matching patches are selected using the Nearest Neighbour (NN) approach. NN introduces bias to the search results due to the present noise. The reference patch is referred to as μ_r . While γ_k refer to the matching patches selected and the prediction error is defined by $\hat{\mu}(\mu_r)$. In NN search, Equ. (2.14), bias is introduced to the search results due to the noise present in the reference patch.

$$d^2(\mu_r, \gamma_k) = (2\sigma^2 P) \sum_{i=0}^{P-1} G(0, 1)^2 \quad (2.14)$$

Where $G(\mu, \sigma^2)$ refers to the Gaussian function with mean 0 and variance σ^2 . The sum of the total pixel values P has χ_P^2 distribution and the estimated distance is shown by Equ. (2.15).

$$E[d^2(\mu_r, \gamma_k)] = 2\sigma^2 \quad (2.15)$$

The proposed SNN estimation for the noise free patch $\hat{\mu}(\mu_r)$ γ_k , is defined as the following Equ. (2.16), the image contains multiple patches $k = 1, \dots, N_n$

$$E[\hat{\mu}(\mu_r)] = \frac{\Delta\delta}{N_n} \sum_{\delta} \sum_{k=1..N_n} E[\gamma_k|\delta].p(\delta) \quad (2.16)$$

Where $\Delta\delta$ is a small interval value. Meaning that the neighbouring patches γ_k lies within the range $[\mu_r - \gamma, \mu_r + \gamma]$.

$$\|d^2(\mu_r, \gamma_k) - o.2\sigma^2\| \quad (2.17)$$

Where o refers to the offset, in case of NN $o = 0$, while in SNN $o = 1$. Equ. (2.14) is applied for the central pixel in patch generating Equ. (2.18),

$$d(\mu.\gamma) \cong \sigma.\sqrt{2P-1} + G(0, \sigma^2) = \sigma + G(0, \sigma^2) \quad (2.18)$$

Meaning that the distance between the noisy patches an the reference patch is almost equivalent to σ . SNN consider all the patches within the range of $\mu \pm o.\sigma$.

The variance in the SNN approach is more than the variance in NN. Even though, the bias estimate is removed, causing a decrease in noise-to-noise matching and better noise removal. Problems appear in SNN when the noise level is low SNN tends to gather more patches that are not closely correlated to the current reference patch causing low-contrast blur details. Also, as the number of patches increases more in SNN than NN, so the variance increase and finally leading to a smaller expected error. A comparison between NLM, NN, and SNN methods are shown in table (2.2)² and visual results are shown in Fig. (2.7).

²SNN results are obtained from the following code, <https://github.com/NVlabs/SNN>

Table 2.2: PSNR comparison between noisy, non-local means (Original NLM), nearest neighbour (NN) and statistical nearest neighbour (SNN) methods for different noise (sigma) values.

Sigma	Method	barbara	baboon	boat	camera	goldhill	lake	lena	couple
20	Noisy	22.25	22.10	22.25	22.45	22.23	22.14	22.10	22.12
	Original NLM	29.15	27.59	30.28	29.40	29.37	28.42	31.18	28.75
	nlm-NN	28.52	27.48	29.42	27.03	29.00	26.82	30.53	28.32
	nlm-SNN	28.54	27.42	29.79	27.19	29.11	26.93	31.22	28.42
40	Noisy	16.41	16.13	16.38	16.67	16.57	16.64	16.26	16.28
	Original NLM	24.50	24.92	25.95	25.60	25.63	24.91	27.25	25.03
	nlm-NN	24.71	24.28	25.28	24.34	25.02	24.19	25.90	24.63
	nlm-SNN	24.96	24.88	25.89	24.54	25.20	24.59	27.63	25.19
60	Noisy	13.34	13.00	13.24	13.54	13.46	13.61	13.15	13.15
	Original NLM	21.92	23.60	23.46	22.68	23.30	22.64	24.79	23.10
	nlm-NN	21.82	21.92	22.32	21.78	22.14	21.82	22.78	21.90
	nlm-SNN	21.89	23.53	22.76	21.68	22.16	22.05	25.07	22.81
80	Noisy	11.46	11.14	11.36	11.56	11.53	11.74	11.29	11.31
	Original NLM	20.21	22.68	21.97	20.70	21.66	20.76	23.32	21.82
	nlm-NN	19.69	20.16	20.33	19.76	20.10	19.88	20.72	20.03
	nlmSNN	18.45	22.37	20.04	18.84	19.35	18.64	22.24	20.41

2.5.5 Non-local Methods with Shape-Adaptive Patches (NLM-SAP)

A proposed approach [22] suggested replacing the common square patch used in the NLM algorithm by other arbitrary shapes, i.e. Disks, Half-pies, Quarter-pies, and Bands. The shapes of the patches used are determined using Stein's Unbiased Risk Estimate (SURE). Fast Fourier transform is used to apply the arbitrary patches to NLMs.

To determine the shapes of the arbitrary patches shown in Fig. (2.4), the Euclidean distance between two pixels is generalized using the following:

$$d_S^2(x, x') = \sum S_\tau (Y(x + \tau) - Y(x' - \tau))^2 \quad (2.19)$$

Where S refers to the shape parameter. For instance, in the original NLM square shape of the patches is represented by:

$$S(\tau) = \begin{cases} \exp(-(\tau_1^2 + \tau_2^2)/2a^2) & , \text{if } \|\tau\|_\infty \leq \frac{p-1}{2} \\ 0 & , \text{otherwise} \end{cases} \quad (2.20)$$

While the flat kernel patch represented by:

$$S(\tau) = \begin{cases} 1 & , \text{if } \|\tau\|_2 \leq \frac{p-1}{2} \\ 0 & , \text{otherwise} \end{cases} \quad (2.21)$$

The general pie form used in Fig. (2.5) next to some alternation resulting from variations in the values of two angles θ_1 , θ_2 , and radius R to determine the shape structure. Finally, patches with a simple rectangle shape with different sizes are also considered as a patch choice. Fig. (2.6).

The adaptive patch approach decreases the "halo noise" caused due to the problem of not extracting enough patches to remove the noise from the neighboring area while preserving the image features. The patches considered in this approach can have different sizes and orientations.

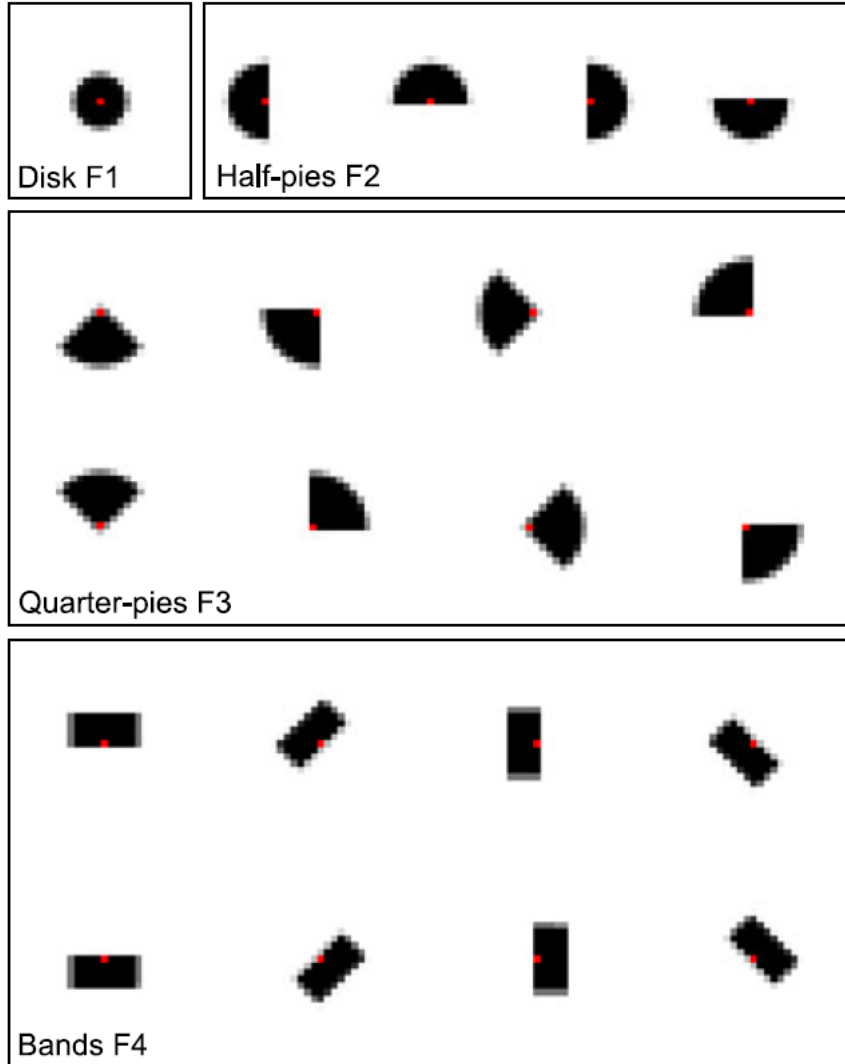


Figure 2.4: Arbitrary patch shapes used in NLM image denoising

To compute the weights of the patches 2D-FFT is applied using Equ. (2.22).

$$d_S^2(x, x + \delta) = \sum_{\tau \in \Omega} S(\tau) (Y(x + \tau) - Y(x + \delta + \tau))^2 = (\check{S} \star \Delta_\delta)(x) \quad (2.22)$$

This distance measured between the patches of shape S is the difference between reference patch and the δ translated patch. $\check{S}(\tau) = S(-\tau)$, $\Delta_\delta(x) = (Y(x) - Y(x + \delta))^2$ and \star refer to the convolution operator. The convolution of two functions \check{S} and Δ_δ can be computed in the transform domain as the multiplication of the two functions in the Fourier transform, shown in

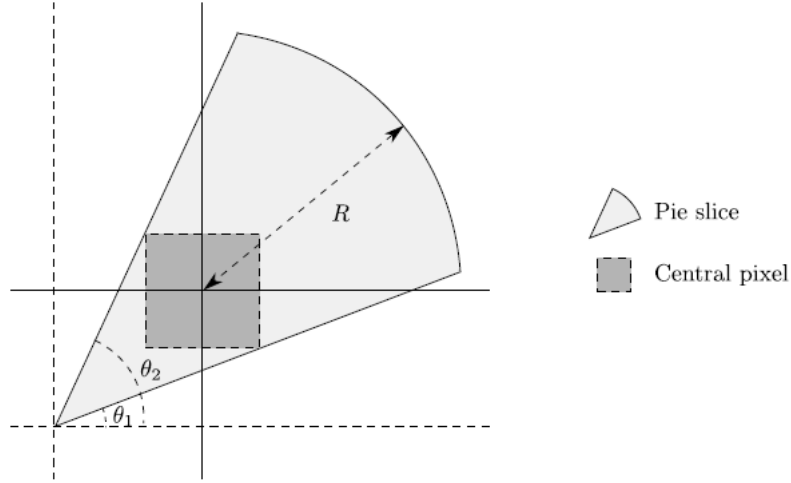


Figure 2.5: Pie patch shape with two angles θ_1 , θ_2 and a radius R to determine the shape structure

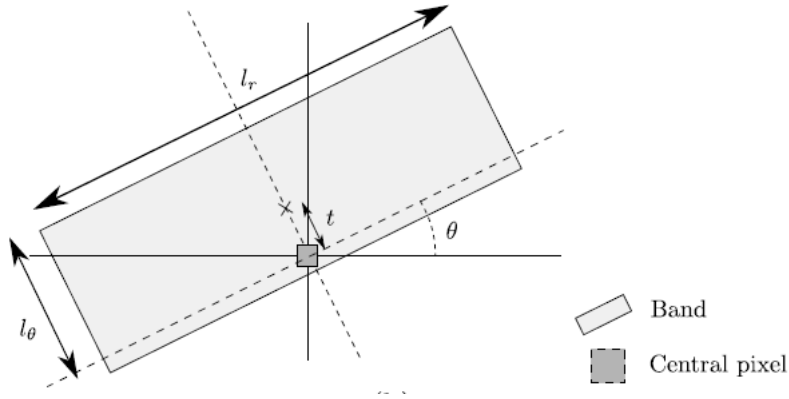


Figure 2.6: Rectangle patch shape with l_0 height, l_r width and θ orientation

Equ. (2.23), causes a huge decrease in the required computational efforts.

$$\check{S} \star \Delta_\delta = \mathcal{F}^{-1}(\mathcal{F}(\check{S})\mathcal{F}(\Delta_\delta)) = \mathcal{F}^{-1}(\mathcal{F}(\bar{S})\mathcal{F}(\Delta_\delta)) \quad (2.23)$$

\mathcal{F} refers to 2D-discrete Fourier transform (2D-FFT), \mathcal{F}^{-1} is inverse 2D-FFT transform. Assume that noisy pixel x in the image has $\hat{f}_1(x), \dots, \hat{f}_K(x)$ pixel estimates obtained from different patch shapes. Then the estimated values are aggregated to obtain the pixel value. The first proposed estimate is a uniformly weighted aggregation (UWA) to estimate x :

$$\hat{f}_{UWA}(x) = \frac{1}{K} \sum_{k=1}^K \hat{f}_k(x) \quad (2.24)$$

The second proposed estimate to limit the halo of noise weighted average variance (WAV) is

used to assign weights for each estimated value for the pixel. However, having large variance estimates cause over-smooth edges and details. A third method used to take into consideration the bias of the estimated values instead of the variance is Stein's Unbiased Estimator of the Risk (SURE) which is used to determine the parameters.i.e: bandwidth h and the patch size p to derive the weights to produce an unbiased estimate of the risk of NLM-SAP denoiser. And finally, since combining several estimates produces better results Exponentially Weighted Aggregation (EWA) is also considered as an estimator that gives better results than other estimators in most images.

The numerical results shows a good deal of improvement shown in the following table (2.3)

3.

³This results are obtained from the following code http://josephsalmon.eu/code/index_codes.php?page=NLMSAP

Table 2.3: Images PSNR comparison between noisy, original nlm (Orig NLM), original shape adaptive patches (Orig-SAP), fast shape adaptive patches (fastSAP) and 8 Gaussian shapes (8 Gaus.dir.) for different sigma noise levels 20, 40, 60 and 80

Sigma	Method	barbara	baboon	boat	camera	goldhill	lake	lena	couple
20	Noisy	22.25	22.10	22.25	22.45	22.23	22.14	22.10	22.12
	Orig NLM	29.15	27.59	30.28	29.40	29.37	28.42	31.18	28.75
	Orig SAP	29.25	27.22	30.21	29.57	29.11	28.12	31.47	28.73
	fastSAP	29.24	27.31	30.34	29.56	29.22	28.39	31.53	28.69
	8 Gau.dir	27.34	26.11	28.69	27.95	27.67	26.28	29.85	27.21
40	Noisy	16.41	16.13	16.38	16.67	16.57	16.64	16.26	16.28
	Orig NLM	24.50	24.92	25.95	25.60	25.63	24.91	27.25	25.03
	Orig SAP	23.67	24.54	25.40	24.94	24.79	23.80	27.19	24.35
	fastSAP	24.03	24.67	25.79	25.53	25.19	24.57	27.45	24.65
	8 Gau.dir	22.48	24.32	24.60	23.56	24.13	22.59	26.34	23.68
60	Noisy	13.34	13.00	13.24	13.54	13.46	13.61	13.15	13.15
	Orig NLM	21.92	23.60	23.46	22.68	23.30	22.64	24.79	23.10
	Orig SAP	20.43	23.51	22.69	21.29	22.28	21.03	24.28	22.28
	fastSAP	20.78	23.59	22.96	21.84	22.54	21.65	24.59	22.49
	8 Gau.dir	20.51	23.60	22.78	21.09	22.24	20.79	24.30	22.34
80	Noisy	11.46	11.14	11.36	11.56	11.53	11.74	11.29	11.31
	Orig NLM	20.21	22.68	21.97	20.70	21.66	20.76	23.32	21.82
	Orig SAP	18.85	22.81	21.31	19.25	20.59	19.16	22.76	21.16
	fastSAP	18.99	22.84	21.41	19.53	20.73	19.42	22.88	21.24
	8 Gau.dir	19.30	22.95	21.66	19.63	20.93	19.50	23.12	21.42



Figure 2.7: Lena Denoising using non-local means (nlm), statistical nearest neighbour (SNN), and fast shape adaptive non-local means (SA-Fast-NLM) approaches at $\sigma = 20$. (a) Original image. (b) Noisy image (PSNR = 22.10 db). (c) Denoised using NLM (PSNR = 31.18 db). (d) Denoised image using SNN (PSNR = 31.22 db). (e) Denoised using SA-Fast-NLM (PSNR = 31.53 db)

2.6 Hybrid Domain Image Denoising

Image denoising methods that contribute both spatial domains and transform domain denoising as complements, proved to be powerful denoising tools, such as the current state-of-art BM3D. The spatial domain methods produce good quality images, but many of the small details are smoothed during the denoising process. While, the transform domain method might not produce the same good quality. It produces ringing artifact around the edges. They also preserve tiny details in the texture of the image.

2.6.1 Block-matching and 3D filtering (BM3D)

BM3D [23] is a denoising filter that depends on self similarities that the images have alongside collaborative filtering. BM3D is divided into two main stages: Stage one in which similar patches are stacked and wavelet shrinkage is applied. Stage two uses the estimated output from stage one and apply the Wiener filter on 3D grouped patches.

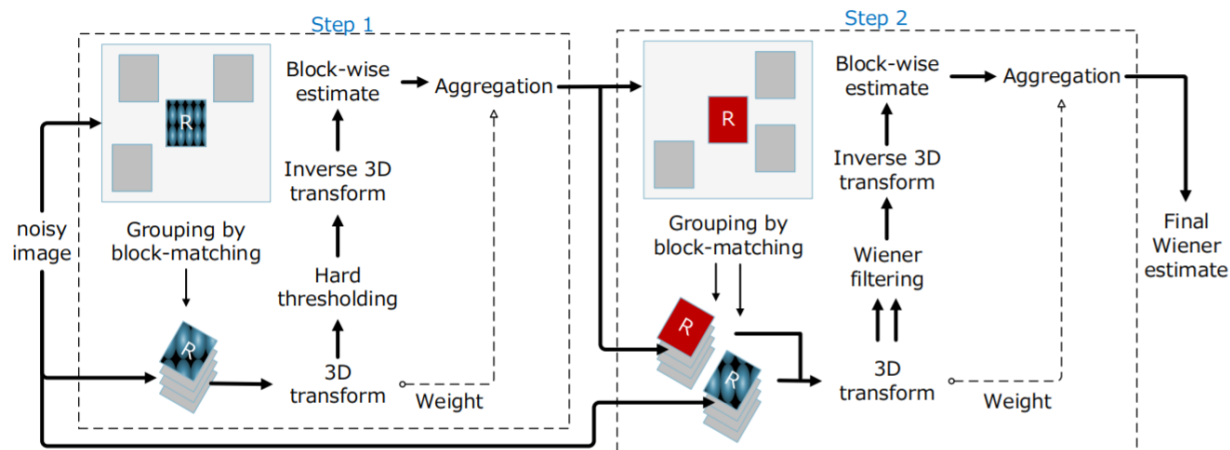


Figure 2.8: BM3D filtering flowchart

Stage One: Basic Estimation

The first stage in BM3D is to stack matching or neighboring blocks into a 3D array so that when the transform domain is applied the data sparsity is improved. Sparsity is defined as the matrix or the vector where most of its values are zero. The corresponding image processing is

a wavelet transform.

Patch Grouping

Assuming that a noisy reference patch is denoted by Z_{xR} , x is the spatial coordinates of the pixel. $Z(x)$ as a central patch pixel and the patch size is N_1^2 . In Fig. (2.8) the first step in stage one of BM3D algorithms starts by block matching using normalized Euclidean distance measure between Z_x blocks which are defined by Equ. (2.25).

$$d(Z_{xR}, Z_x) = \frac{\|\hat{\Upsilon}(\tau_{hard}^{2D}(Z_{xR})) - \Upsilon(\tau_{hard}^{2D}(Z_x))\|_2^2}{(N_1^{hard})} \quad (2.25)$$

To reduce the noise, hard thresholding (thr) is applied after 2D transform as referred to by the unitary transform operator τ_{2D} is applied on the patches similar to Z_{xR} , then hard threshold operator Υ with $\lambda_{2D\sigma}$ is applied. Patch grouping is done by including the patches with distances less than the threshold value to exclude the less similar patches. As shown in the following expression(2.26):

$$\Upsilon(\lambda, \lambda_{thr}) = \begin{cases} \lambda & \text{if } |\lambda| > \lambda_{thr} \\ 0 & \text{otherwise} \end{cases} \quad (2.26)$$

λ_{thr} is the maximum distance between two patches that allow the patch to join the set of similar patches called $Z_{S_{xR}}$.

Collaborative Hard Threshold

Given a 3D array $Z_{S_{xR}}$ with $N_1 \times N_1 \times |S_{xR}|$ dimension. 3D transform consists of 2D transform over the patches and 1D Walsh-Hadamard transform over the third dimension. The Walsh-Hadamard transform requires an even number of similar patches such that N^{hard} is always a power of two. If an even number of patches is not accessible then we decrease the number of extracted patches until a power of two can be obtained. The noise is removed using hard thresholding. τ_{3D}^{-1} . Finally, the denoised block is estimated as the inverse 3D transform value, as defined in Equ. (2.27)

$$\hat{Y}_{S_{x_R}} = \tau_{3D}^{-1}(\Upsilon(\tau_{3D}(Z_{S_{x_R}}), \lambda_{thr3D\sigma} \sqrt{2\log(N_1^2)})) \quad (2.27)$$

The projecting from the wavelet transform domain to the image spatial domain is not perfect because a single patch can be a correct solution if placed in multiple places in the original image.

Patch Aggregation

The weighting method gives more weight to homogeneous patches unlike areas that combine multiple different textures, causing the preservation of edges and the avoiding of ringing artifact around the edges. The weights of the patches in original BM3D is calculated as the product of the estimated patch after inverse 3D transform and the Kaiser window. Currently, each pixel has more than one estimated average weight due to patch overlap, that is why aggregation is used to estimate the proper weight that contributes to the final pixel value. In original BM3D the weight of the pixel is inversely proportional to the total block variance estimate. Weight is defined for each block in the final set of patches using the following Equ. (2.28).

$$w_{x_R} = \begin{cases} \frac{1}{N_{har}} & \text{if } N_{har} \geq 1 \\ 1 & \text{otherwise} \end{cases} \quad (2.28)$$

N_{har} is the number of transform coefficients with values other than zero after applying a threshold. Meaning that the more different the 3D block is the less this block contributes to the weight. The final estimate of the pixel \hat{y} is given by Equ. (2.29)

$$\hat{y} = \frac{\sum_{x_R \in X} \sum_{x_m \in S_{x_R}} w_{x_R} \hat{Y}_{x_m}^{x_R}(x)}{\sum_{x_R \in X} \sum_{x_m \in S_{x_R}} w_{x_R} \chi_{x_m}^{x_R}(x)}, \forall x \in X \quad (2.29)$$

Stage Two: Wiener Filter

In this stage, the basic estimate of stage one and the noisy image is used to obtain a better-denoised estimate of the image. The final denoised image is obtained by applying Wiener filter shrinkage is applied in the 3D transformed domain stack extracted from the basic image.

Patch Grouping

The second stage in BM3D involves applying the Wiener filter as shown in Fig. (??). The first step is to find the matching blocks within the estimated image resulting from stage one using a hard threshold on d-distance measure defined by Equ. (2.30).

$$S_{x_R} = x \in X \mid \frac{\|(E_{x_r} - \bar{E}_{x_R}) - (E_x - \bar{E}_x)\|_2^2}{N_1^{thr}} < \tau_{match} \quad (2.30)$$

E_{x_R} and E_x are two matching with \bar{E}_{x_R} and \bar{E}_x referring to the mean values respectively. The mean is subtracted to prevent the patch similarity searching to be biased.

Wiener Filter Shrinkage

The Wiener filter is evaluated in 3D transform by Equ. (2.31).

$$W_{S_{x_R}} = \frac{|\tau_{3D}(E_{S_{x_R}})|^2}{|\tau_{3D}(E_{S_{x_R}})|^2 + \sigma^2} \quad (2.31)$$

$E_{S_{x_R}}$ is a 3D array of stacked matching patches. The 3D array of noisy observation $Z_{S_{x_R}}$ have the 3D transform and Wiener filter $W_{S_{x_R}}$ corresponding elements multiplied and then the inverse 3D transform is applied giving Equ. (2.32).

$$\hat{Y}_{S_{x_R}} = \tau_{3D}^{-1}(W_{S_{x_R}} \tau_{3D}(Z_{S_{x_R}})) \quad (2.32)$$

Patch Aggregation

Patch aggregation is the same as stage one. The final weight assigned to the patches is inversely proportional of $\hat{Y}_{S_{x_R}}$ defined as:

$$W_{x_R} = \left(\sum_{i=1}^{N_1} \sum_{j=1}^{N_1} \sum_{y=1}^{|S_{x_R}|} |W_{S_{x_R}}(i, j, t)|^2 \right)^{-1} \quad (2.33)$$

Figure 2.9: BM3D Parameters [24]

The results of BM3D, table (2.4)⁴, show a high noticeable improvement in both PSNR and the image quality. Although, BM3D have many drawback points. First, implementation is considered to be more complex than other denoising algorithms, Second BM3D performance during patch matching is both time consuming and does not always obtain the best matching results. Third, BM3D cannot preserve details in high textural images in addition to the disturbance in smooth areas in images with high standard deviation noise.

Table 2.4: PSNR comparison between noisy image and BM3D denoised results for 20, 40, 60, 80 sigma noise levels

Sigma	Method	barbara	baboon	boat	camera	goldhill	lake	lena	couple
20	Noisy	22.25	22.10	22.25	22.45	22.23	22.14	22.10	22.12
	BM3D	30.23	28.24	31.50	30.29	30.56	29.40	32.70	30.30
40	Noisy	16.41	16.13	16.38	16.67	16.57	16.64	16.26	16.28
	BM3D	25.96	25.60	27.86	26.79	27.06	25.85	29.30	26.67
60	Noisy	13.34	13.00	13.24	13.54	13.46	13.61	13.15	13.15
	BM3D	23.52	24.54	25.35	23.84	24.54	23.42	27.18	24.55
80	Noisy	11.46	11.14	11.36	11.56	11.53	11.74	11.29	11.31
	BM3D	21.65	23.63	23.70	21.82	22.50	21.55	25.51	22.97

⁴The BM3D results are generated from the code in <http://www.cs.tut.fi/~foi/GCF-BM3D/>



(a)



(b)



(c)

Figure 2.10: Lena image denoised by BM3D at $\sigma = 20$. (a) Original image. (b) Noisy image (PSNR = 22.25). (c) Denoised image (PSNR = 30.23)

2.6.2 Image denoising with morphology-and size-adaptive block-matching transform domain filtering

An approach is proposed in order to improve BM3D search performance in the smooth areas and low contrast details called Shape Adaptive Block Matching and 1D - 3D filtering (SA-BM1-3D) [25]. The flow of the proposed algorithm is shown in Fig. (2.11)

Step(1) : Apply patch matching, same as BM3D, but instead of applying 3D transform, we perform 1D Haar transform on the third dimension. To avoid artifacts caused by thresholding small noise values from highly noisy coefficients allowing the preservation of details.

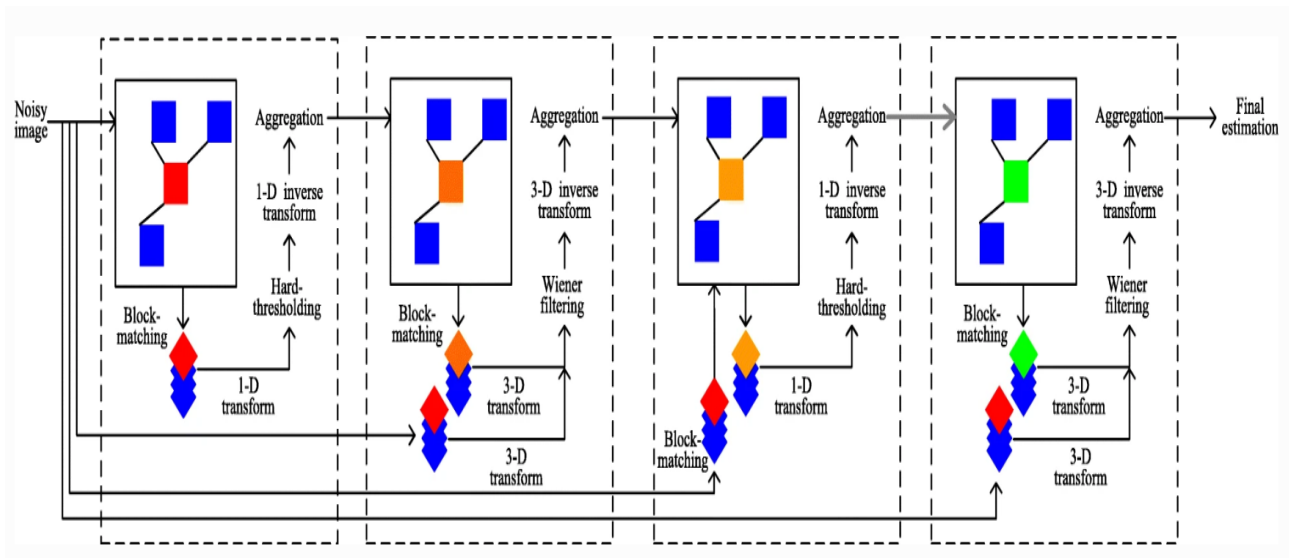


Figure 2.11: Explanation of the algorithm flow of image denoising with morphology and size adaptive block-matching transform domain filtering (SA-BM1-3D) [25]

Step(2) : The resulting image from step 1 is used as a reference and Wiener filter is applied on 3D transform coefficients of the original noisy image, even though 3D transform threshold is not good in preserving image edges.

Step(3) : DCT is applied on the reference patches then Alternating Current (AC) energy of the DCT coefficients is calculated by Equ. (2.34). AC is used to classify the image into three areas contour, texture, and smooth components. According to the regional energy the block size is adaptively determined. Specifically, small patches are used for contour areas, medium patches are used for texture areas and large patches are used for smooth areas.

$$E_{AC} = \sum_{i=1}^N \sum_{j=1}^N |\hat{B}_R(i, j)| - |\hat{B}_R(1, 1)| \quad (2.34)$$

\hat{B} refers to the resulting image blocks of inverse transform of the shrank values of the transform values. The morphological components of an image patch are determined as follows.

$$\begin{cases} C_{contour} & , \text{if } E_{AC-c\sigma} \geq K_1 \\ C_{texture} & , \text{if } K_2 \leq E_{AC-c\sigma} < K_1 \\ C_{smooth} & , \text{if } E_{AC-c\sigma} < K_2 \end{cases} \quad (2.35)$$

K_1 , K_2 values are empirically determined, $c = 0.18$. Then block-matching on the original noisy image is applied and we use these blocks to extract patches at the same location to implement 1D Haar transformation followed by thresholding on the blocks.

Step(4) : Apply the same steps applied in Step 3 to obtained adaptive similar patches by determining the type of the reference patch and chose the appropriate patch size. Then 3D Wiener filter is applied.

The results of this approach, according to the paper and the table (2.5)⁵, give better results and fewer artifacts than BM3D and SAPCA-BM3D. However, the time complexity of SA-BM1-3D is more than BM3D since BM3D is only two stages while SA-BM1-3D is applied in four stages.

Table 2.5: Compared PSNR of noisy, BM3D and shape adaptive BM3D with various noise sigma 5, 15, 25 and 35.

Image Name	Method	sigma = 5	sigma = 15	sigma = 25	sigma = 35
Barbara	Noisy	34.16	24.59	20.18	17.25
	BM3D	38.01	31.56	28.76	26.89
	SAP-BM3D	38.34	33.30	30.99	29.35
boat	Noisy	34.13	24.56	20.17	17.26
	BM3D	39.39	32.98	30.43	28.76
	SAP-BM3D	37.47	32.29	30.03	28.51
camera	Noisy	34.18	24.61	20.21	17.27
	BM3D	38.20	31.72	29.25	27.76
	SAP-BM3D	38.53	32.36	29.81	28.17
goldhill	Noisy	34.14	24.56	20.13	17.23
	BM3D	37.94	31.97	29.37	27.83
	SAP-BM3D	37.30	32.05	29.96	28.62
lena	Noisy	34.15	24.63	20.14	17.21
	BM3D	40.26	34.29	31.63	29.94
	SAP-BM3D	38.82	34.42	32.22	30.72

⁵SAP-BM3D results are taken from [25], BM3D code results are from the code present in <http://www.cs.tut.fi/~foi/GCF-BM3D/>

2.6.3 Image denoising based on non-local means filter and its method noise thresholding

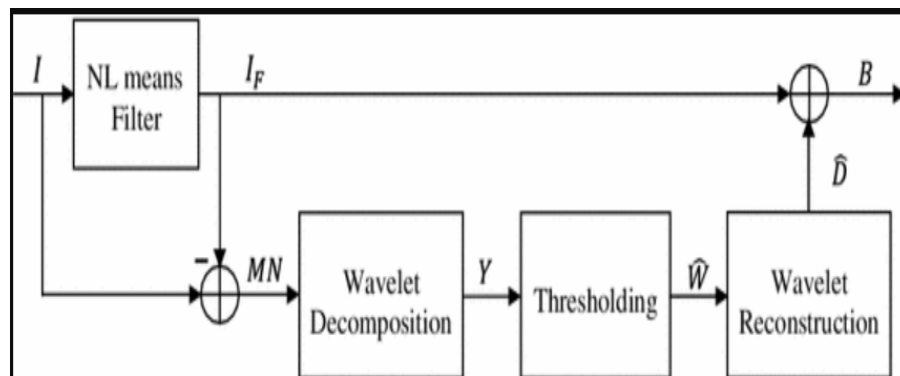


Figure 2.12: The stages used in the combination of wavelet thresholding and NLM algorithm

Denoising method based on the NLM filter and wavelet threshold by Kumar in [26]. This approach tries to minimize the problem of losing too many details, especially at high noise levels as NLM performance deteriorates. The methods shown in Fig. (2.12). The NLM denoising is applied on noisy image I then the difference between the original image and the NLM result image is called method noise. Method noise contains the noise and all the details lost during NLM de-noising, defined in Equ. (2.36)

$$MN = I - I_F \quad (2.36)$$

Where I is the given original noisy image. I_F is the output of the de-noised operator.

$$MN = D + N \quad (2.37)$$

Where D stand for image details and N us the Gaussian noise. Now the problem is to determine the details, like features and edges in the MN image, overlooked by NLM denoising using the wavelet domain. Equ. (2.38) represented,

$$Y = W + N_\omega \quad (2.38)$$

Where Y is noisy wavelet coefficients (method noise), W is coefficients of the detail image and N_ω is the Gaussian noise. Estimating W by thresholding Y with adaptive BayesShrink. BayesShrink is adaptive, soft thresholding in which the wavelet coefficients are assumed to have

Gaussian distribution. The threshold value is determined such that Bayesian risk is minimized in each sub-band, the threshold value is defined by Equ. (2.39),

$$T = \frac{\sigma_n^2}{\sigma_\omega} \quad (2.39)$$

Where σ_n^2 is the variance of the noise for a sub-band HH_1 . The noise variance is determined by the median estimator σ_ω is the standard deviation of the wavelet coefficients.

2.7 Summary

Table 2.6: Comparison of PSNR for noisy, non-local means (NLM), non local means nearest neighbour (nlmNN), non-local means statistical nearest neighbour (nlmSNN), shape adaptive non-local means (Original-SAP) and Block Matching and 3D Collaborative Filtering (BM3D) for multiple noise levels 20, 40, 60, and 80.

Method	Sigma	barbara	baboon	boat	camera	goldhill	lake	lena	couple
20	Noisy	22.25	22.10	22.25	22.45	22.23	22.14	22.10	22.12
	Orig NLM	29.15	27.59	30.28	29.40	29.37	28.42	31.18	28.75
	nlmNN	28.52	27.48	29.42	27.03	29.00	26.82	30.53	28.32
	nlmSNN	28.54	27.42	29.79	27.19	29.11	26.93	31.22	28.42
	fastSAP	29.24	27.31	30.34	29.56	29.22	28.39	31.53	28.69
	BM3D	30.23	28.24	31.50	30.29	30.56	29.40	32.70	30.30
40	Noisy	16.41	16.13	16.38	16.67	16.57	16.64	16.26	16.28
	Orig NLM	24.50	24.92	25.95	25.60	25.63	24.91	27.25	25.03
	nlmNN	24.71	24.28	25.28	24.34	25.02	24.19	25.90	24.63
	nlmSNN	24.96	24.88	25.89	24.54	25.20	24.59	27.63	25.19
	fastSAP	24.03	24.67	25.79	25.53	25.19	24.57	27.45	24.65
	BM3D	25.96	25.60	27.86	26.79	27.06	25.85	29.30	26.67
60	Noisy	13.34	13.00	13.24	13.54	13.46	13.61	13.15	13.15
	Orig NLM	21.92	23.60	23.46	22.68	23.30	22.64	24.79	23.10
	nlmNN	21.82	21.92	22.32	21.78	22.14	21.82	22.78	21.90
	nlmSNN	21.89	23.53	22.76	21.68	22.16	22.05	25.07	22.81
	fastSAP	20.78	23.59	22.96	21.84	22.54	21.65	24.59	22.49
	BM3D	23.52	24.54	25.35	23.84	24.54	23.42	27.18	24.55
80	Noisy	11.46	11.14	11.36	11.56	11.53	11.74	11.29	11.31
	Orig NLM	20.21	22.68	21.97	20.70	21.66	20.76	23.32	21.82
	nlmNN	19.69	20.16	20.33	19.76	20.10	19.88	20.72	20.03
	nlmSNN	18.45	22.37	20.04	18.84	19.35	18.64	22.24	20.41
	fastSAP	18.99	22.84	21.41	19.53	20.73	19.42	22.88	21.24
	BM3D	21.65	23.63	23.70	21.82	22.50	21.55	25.51	22.97

In this chapter, results in table 2.6 show that the BM3D algorithm is the best so far. We discussed various types of image denoising algorithms. Algorithms with significant improve-

ments are presented visual results. The shape and size adaptive algorithms explained with respect to both NLM and BM3D shows an improvement in numerical and visual results. Our suggested framework highlights this idea to produce better image denoising results and also gives high significance to all shape image structures.

3 Methodology

3.1 Introduction

Block-matching and 3D filtering (BM3D) is considered as a prominent state-of-art denoising algorithm. In Fig. (3.1) smooth areas in BM3D suffer from distortion. While edge areas loss details (Lena's hat).

BM3D still has room for improvement. In this chapter, a framework is proposed to add different kernel shapes to the original BM3D. BM3D originally uses the Kaiser (Gaussian) kernel to determine the denoised pixel value. Instead, the suggested kernel shapes shown in Fig. (3.2) are used.

The reason for choosing these shapes is that the edge areas in images can be quantized into four angles 0, 45, 90, and 135 degrees. Besides, using these shapes allows BM3D to preserve image details while maintaining smooth areas [25]. This framework can be used with more shapes that can improve the denoising performance. However, for this study this shapes are sufficient.

The multiple kernels are used to produce various images with different performances. The combination of these images is the most crucial part of the framework.



Figure 3.1: Lena image at $\sigma = 40$, denoised using BM3D algorithm showing disturbance in smooth areas and loss in fine details. (a) Original Image. (b) Noisy image $\sigma = 40$. (c) Denoised after stage one, (d) Denoised after stage two.

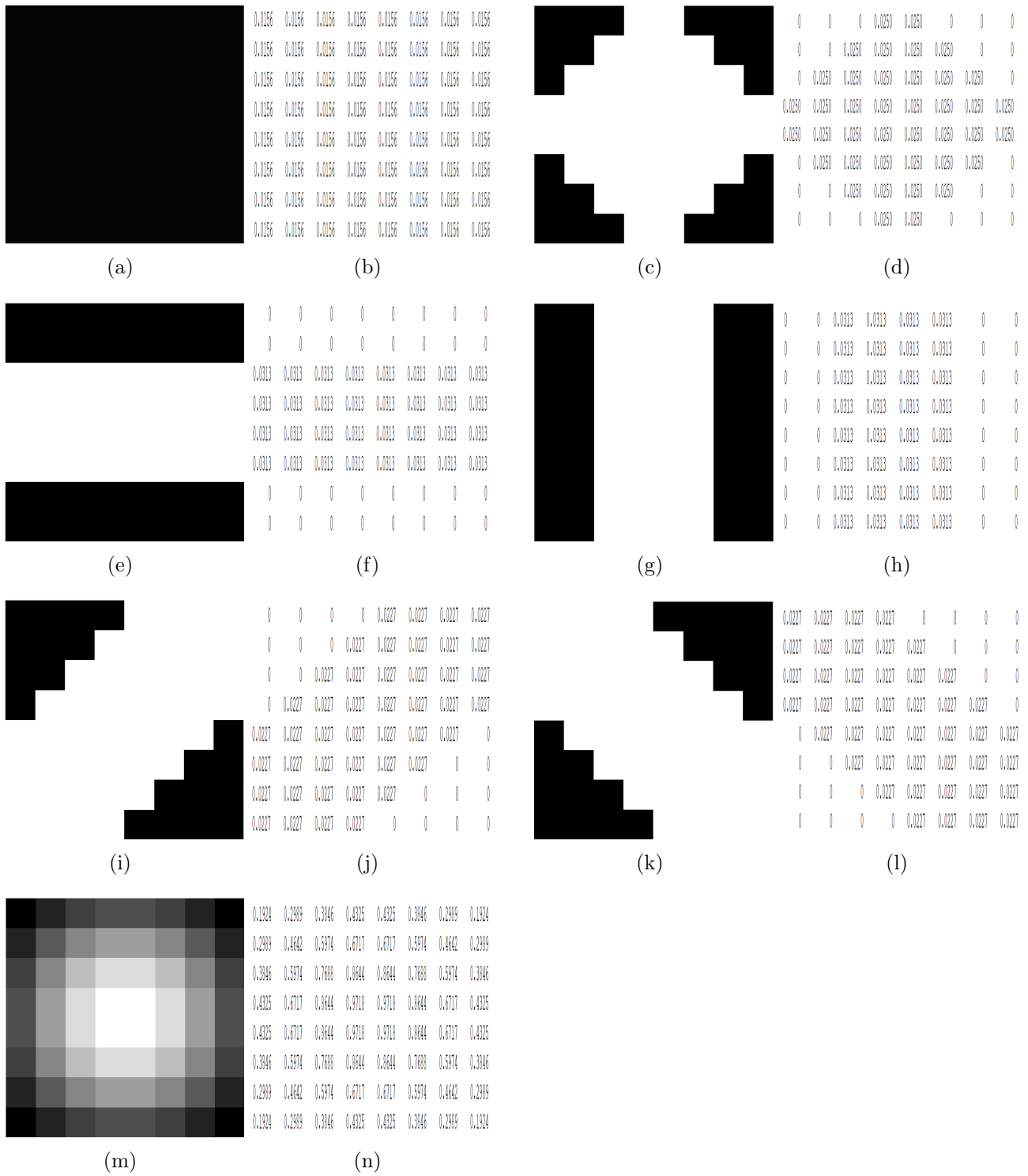


Figure 3.2: Different shaped Gaussian Kernels:(a) Square flat kernel.(b) Numerical representation of square flat kernel. (c) Circular flat kernel. (d) Numerical representation of circular flat kernel. (e) Horizontal flat kernel. (f) Numerical representation of horizontal flat kernel.(g) Vertical flat kernel. (h) Numerical representation of vertical flat kernel. (i) D45 flat kernel. (j) Numerical representation of D45 flat kernel. (k)D135 flat kernel. (l) Numerical representation of D135 flat kernel. (m) Kaiser Kernel. (n) Numerical representation of kaiser kernel.

3.2 Framework Analysis

An experiment is conducted to show the benefit of using various kernel shapes. The kernels are used to estimate the final weights of the pixels from multiple patches. The weights are based on the distances from the reference patch. The multiple kernels are applied in the aggregation step after stages one and two of the original BM3D. The output after utilizing the six shapes are images, as shown in Fig. (3.3) after BM3D stage one and Fig. (3.4) after stage two. The images show an improvement in the PSNR between the noisy image and the denoised images. However, the improvement is less than the original BM3D results in most cases (except for square kernel in some cases when the noise is low). The combination of the resulting images using either pixel-based or patch-based methods is a significant improvement in PSNR values and consequently, on the quality of the image.



(a)



(b)



(c)



Figure 3.3: Lena image denoised by BM3D using different shaped kernels for stage one: (a) Original image. (b) Noisy image with $\sigma = 20$ (PSNR = 22.10). (c) Original BM3D stage one denoised image (PSNR = 32.30). (d) square kernel (PSNR = 32.19). (e) Circular kernel (PSNR = 32.12). (f) Horizontal kernel (PSNR = 32.16). (g) Vertical kernel (psnr = 32.15). (h) Diagonal with angle 135 kernel (PSNR = 32.20). (i) Diagonal with angle 45 kernel (PSNR = 32.16).



(a)



(b)



(c)



Figure 3.4: Lena image denoised by BM3D using different shaped kernels for stage two: (a) Original image. (b) Noisy image with $\sigma = 20$ (PSNR = 22.10). (c) Original BM3D stage two denoised image (PSNR = 32.83). (d) Square kernel (PSNR = 33.43). (e) Circular kernel (PSNR = 33.44). (f) Horizontal kernel (PSNR = 33.44). (g) Vertical kernel (PSNR = 33.42). (h) Diagonal with angle 135 kernel (PSNR = 33.44). (i) Diagonal with angle 45 kernel (PSNR = 33.45).

3.3 Proposed Framework

It is assumed that the given image is corrupted by AWGN, i.e., the noise and image are uncorrelated. Our proposed framework replaced the square Gaussian kernel used in BM3D aggregation step by six various flat kernels. Shown in Fig. (3.5), Fig. (3.6). BM3D originally uses a window of size 8×8 for noise levels less than 40. Otherwise, it uses a window of size 12×12 .

The input of the aggregation step is 3D stacked patches. These patches should be placed back in their original locations in the image. However, due to overlapping and similarities between image patches, each pixel has multiple evaluations. In original BM3D, the evaluation of a normalized sum of the pixel values (in each patch) multiplied by the kernel values is used to give final pixel estimation. as shown in Equ. (3.1),

$$\hat{y} = \frac{\sum_{x_R \in X} \sum_{x_m \in S_{x_R}} w_{x_R} \hat{Y}_{x_m}^{x_R}(x)}{\sum_{x_R \in X} \sum_{x_m \in S_{x_R}} w_{x_R} \chi_{x_m}^{x_R}(x)}, \forall x \in X \quad (3.1)$$

Where \hat{Y} refers to the pixel value in patch x . w_{x_R} refers to the weight of the pixel obtained from evaluating its location from the center pixel using the kernel in the original BM3D. The proposed framework applies to Equ. (3.1) six times, each time with a different kernel.

The flat kernel gives similar weights to all the pixels in the given patch area, thus reducing the computations. Instead of multiplying each pixel by a different value in the kernel depending on its distance from the pixel at the patch center, all the pixels in the patch are multiplied by the same value. Another reason for using a flat kernel is its ability to fit the underlying structure of the image, i.e., instead of relying on the distance from the central pixel, we rely on the shape of the extracted patch and how much it is similar to the structures surrounding it.

The most crucial step in the framework is aggregating the output images into one final denoised image. Using the original noiseless image, we combine the results. The framework does not rely on the original image or consider it as part of the algorithm, it is used for feasibility only. The original image allow use to estimate the optimal results for the framework and prove that there is still a room for improvement by using adaptive shapes. Even though, in our future

work, we are looking for other approaches that can be used to aggregate the image automatically.

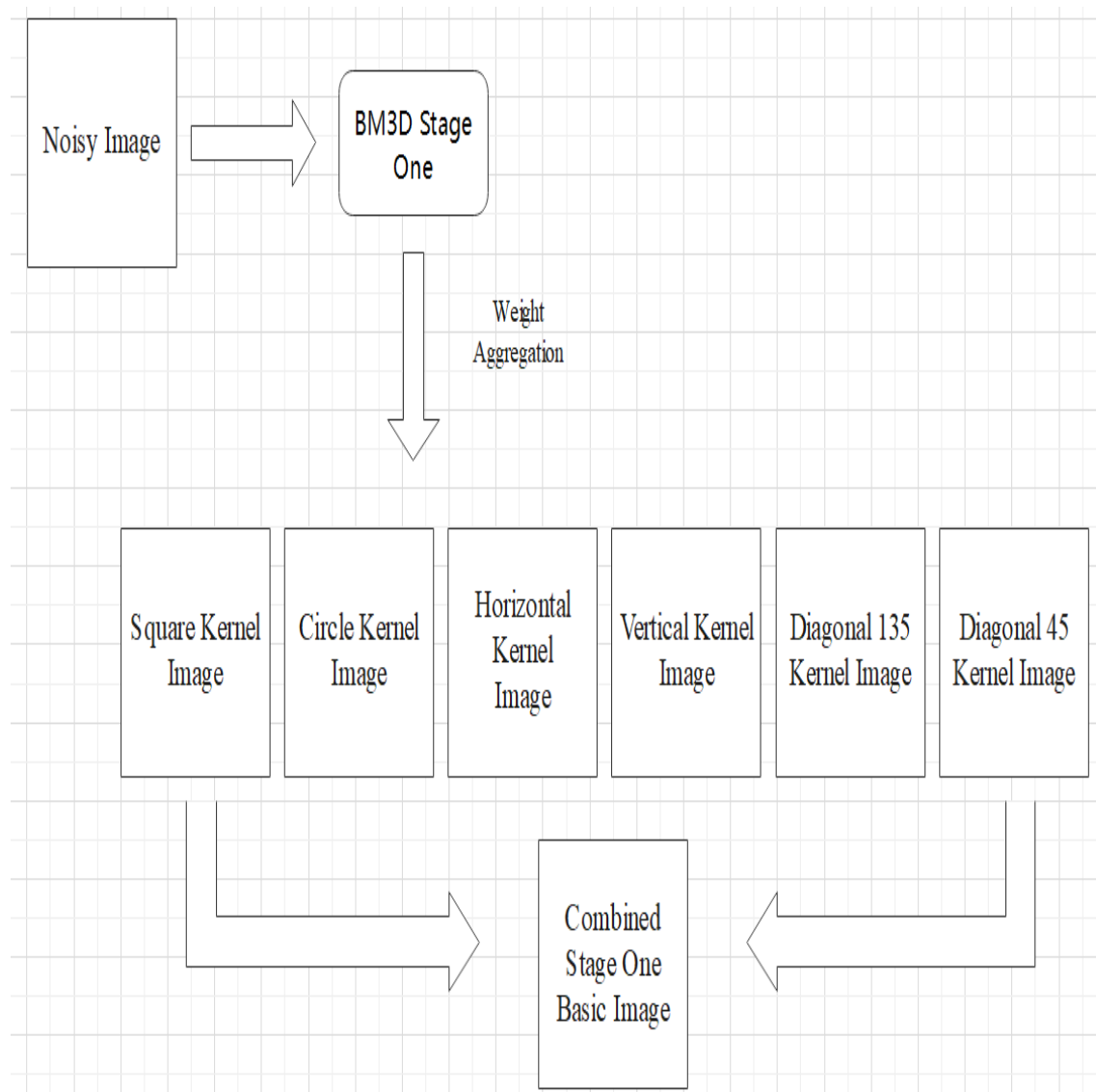


Figure 3.5: Stage one for the proposed framework

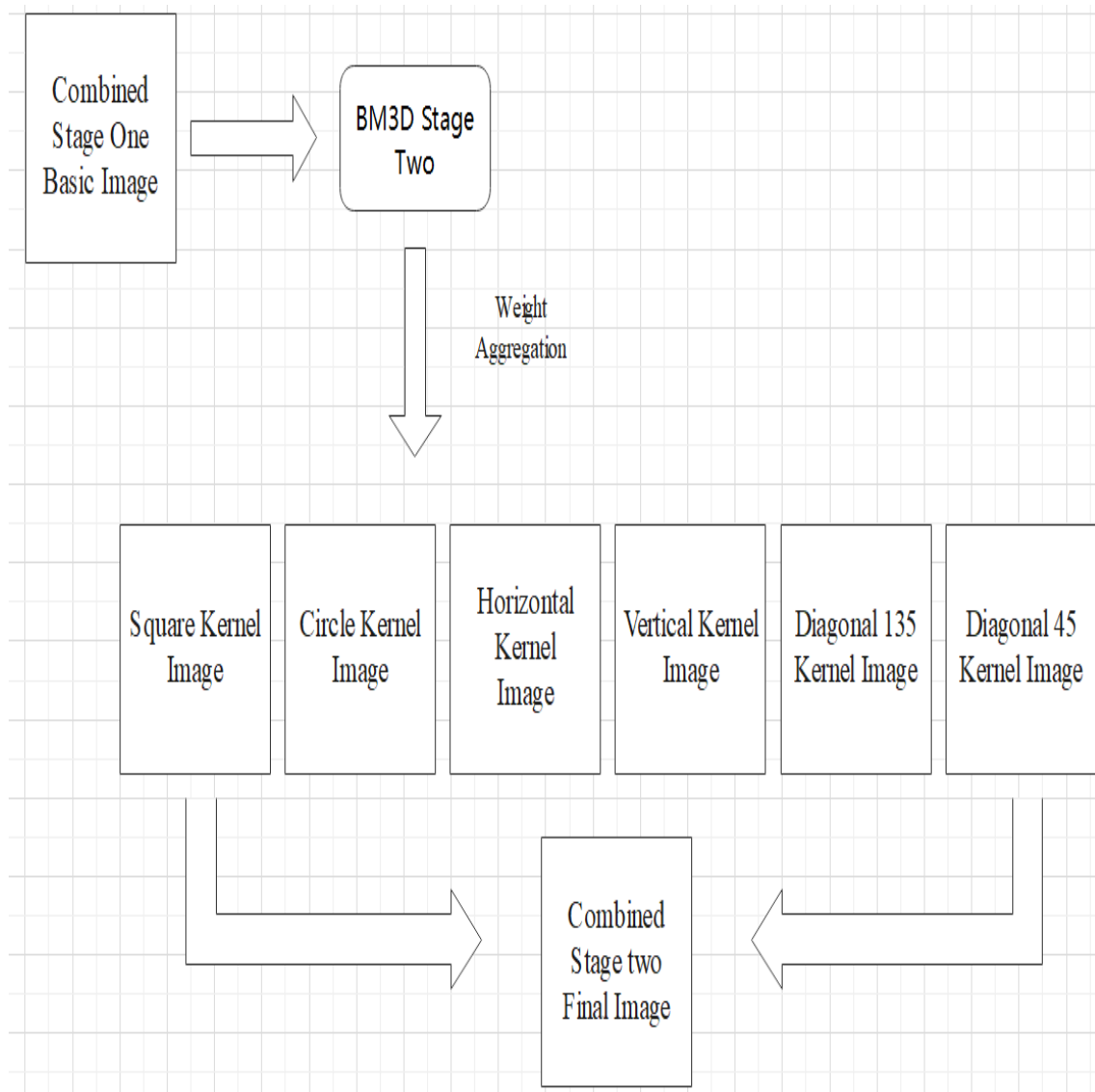


Figure 3.6: Stage two for the proposed framework

Given the six images that represent the solution space for each pixel or patch and the original image, the optimal solution is the one with the least pixel or patch error. The least error choices are shown in the following Fig. (3.7) for stage one and Fig. (3.8) for stage two.

Another approach to determine the relationship between the six images is to used patches comparison to the original image patches. Extract the corresponding patches from the six images obtained from different kernel weights, then compare it to the corresponding original image patch and choosing the absolute minimum difference patch results, as in Fig. (3.9) for stage one and Fig. (3.10) for stage two.

The patch size chosen is 3×3 , as when the size of the patch increases the improvement declines due to the increase of the mismatching between the estimated value and the kernel. The kernel used to calculate the final pixel value is not necessarily the same kernel used for all its neighbouring pixels.

Finally, the results of aggregated pixels are shown in Fig. (3.11) for stage one and stage two.

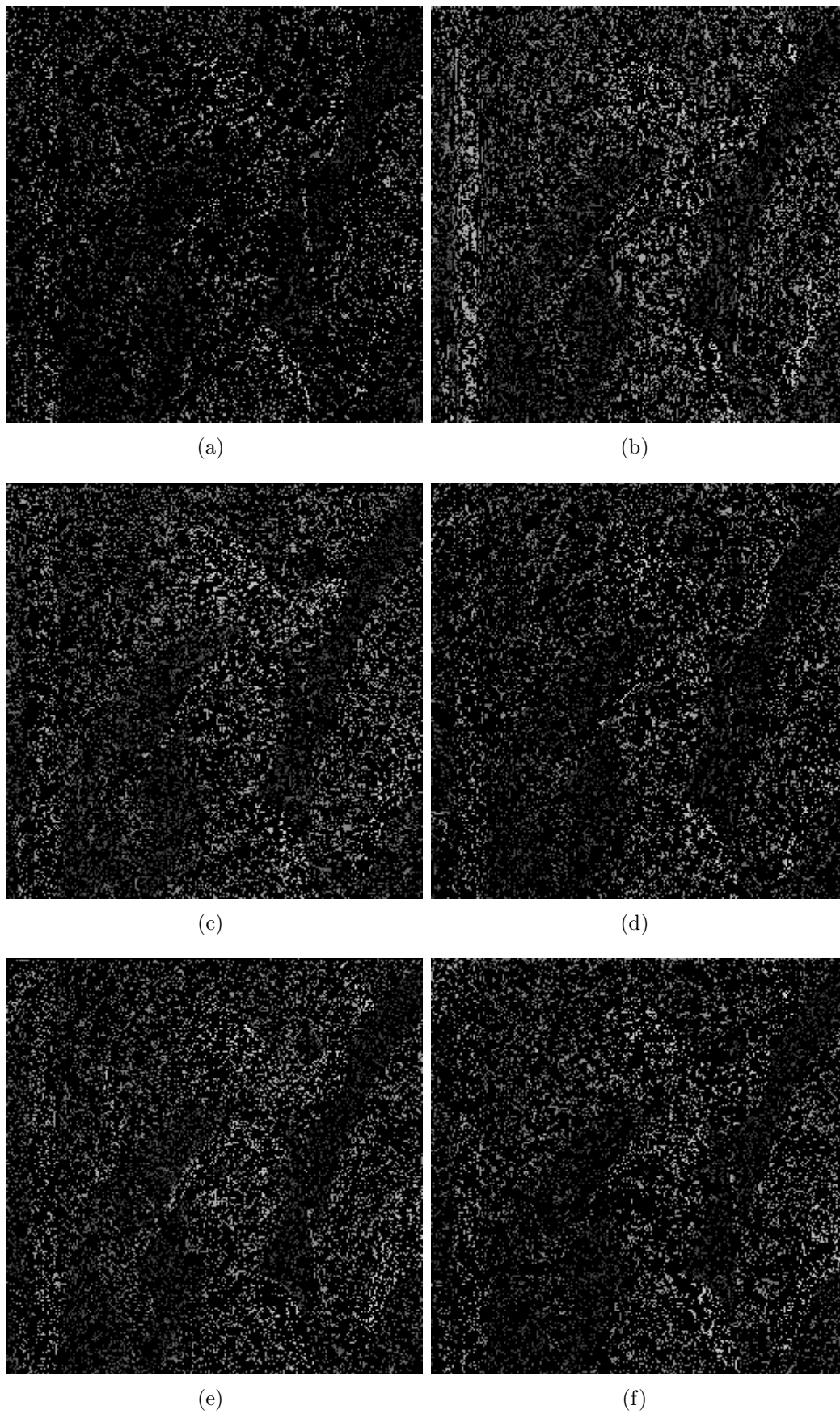


Figure 3.7: Lena image, after stage one, comprising six images that represent the locations of the pixels that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.

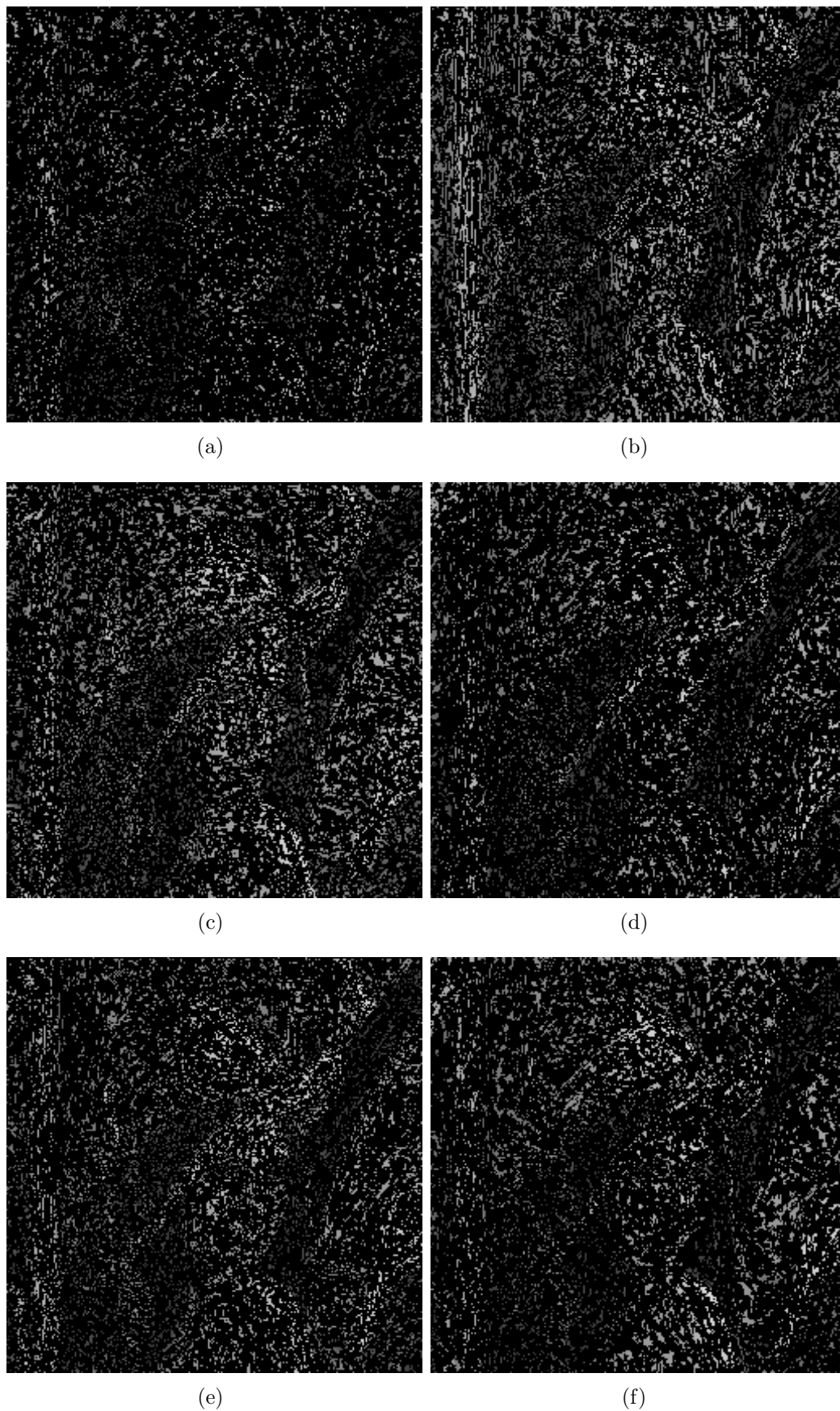


Figure 3.8: Lena image, after stage two, comprising six images that represent the locations of the pixels that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.

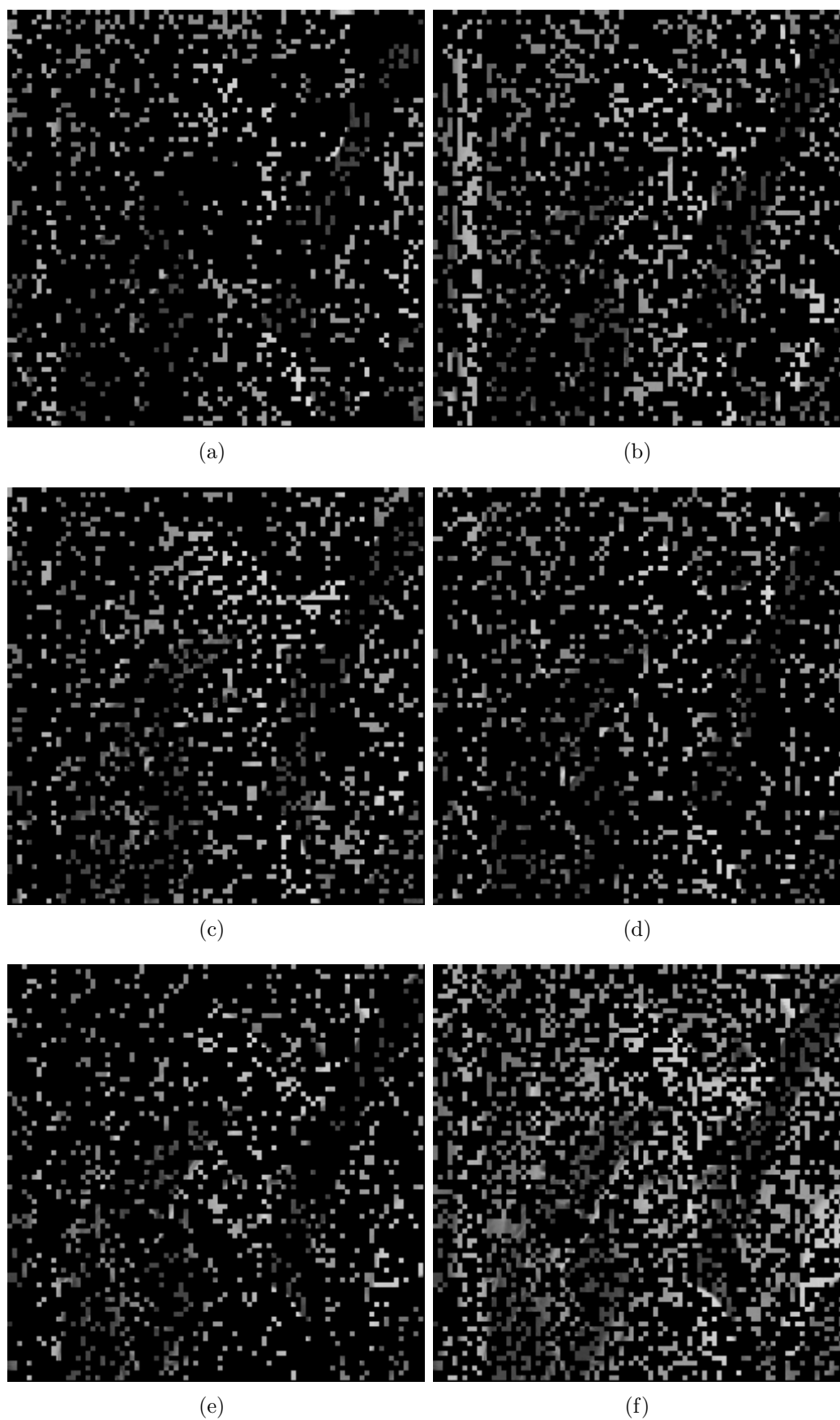


Figure 3.9: Lena image, after stage one, comprising six images that represent the locations of the patches that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.

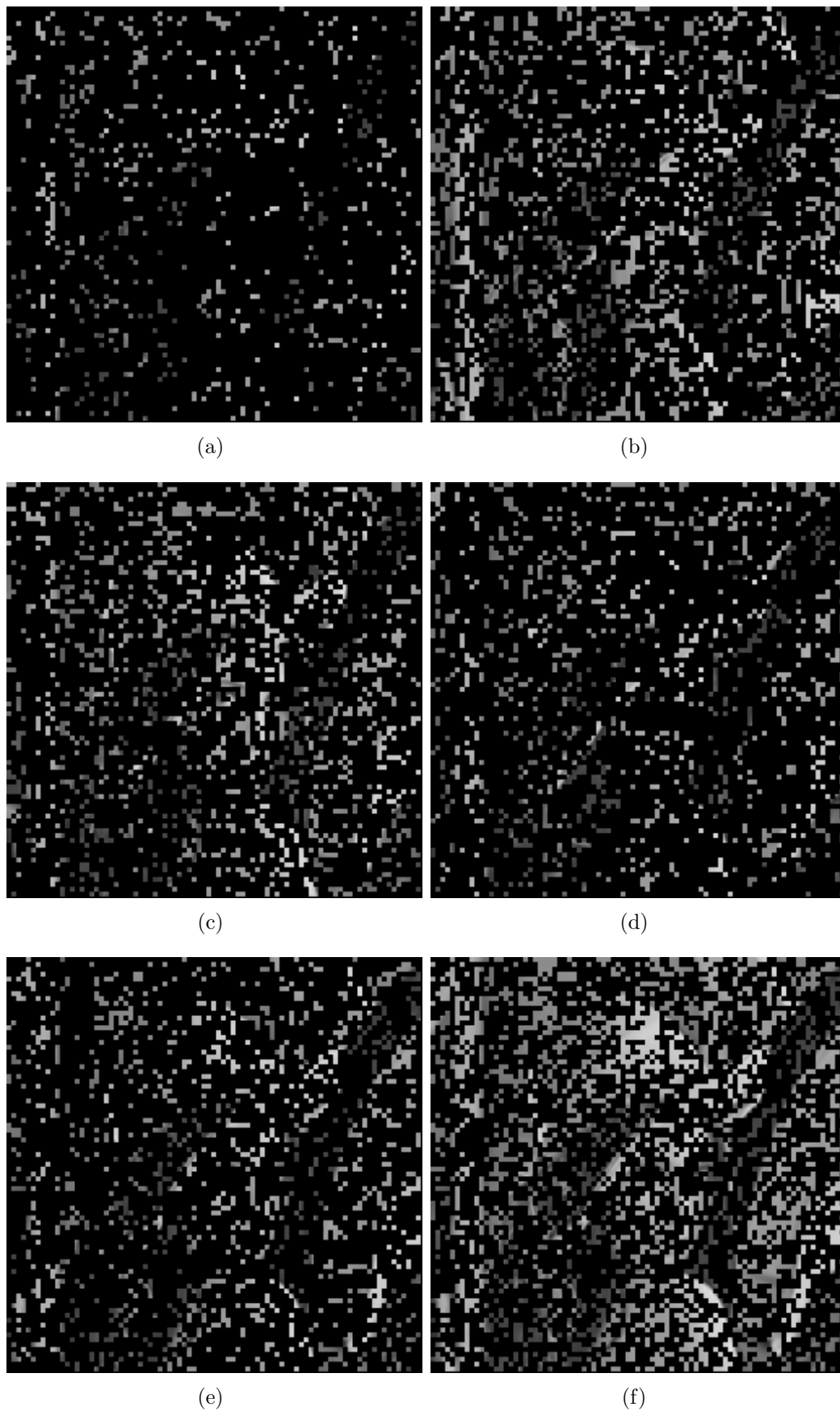


Figure 3.10: Lena image, after stage two, comprising six images that represent the locations of the patches that should be taken from each image to form the optimal denoised image. (a) Square kernel image. (b) Circular kernel image. (c) Horizontal kernel image. (d) Vertical kernel image. (e) D45 kernel image. (f) D135 kernel image.



(a)



(b)



(c)



(d)



(e)



(f)



Figure 3.11: Denoised Lena image results using the proposed framework. (a) Original image. (b) Noisy image with $\sigma = 20$ (PSNR = 22.14). (c) Original BM3D after stage one. (PSNR = 32.23) (d) Original BM3D after stage two (PSNR = 33.77). Pixel based results, (e) Proposed framework after stage one (PSNR = 33.69). (f) Proposed framework after stage two (PSNR = 33.99). Patch based results, (g) Proposed framework after stage one (PSNR = 32.74). (h) Proposed framework after stage two (PSNR = 33.26)

4 Results

In this chapter, we will discuss the results of the proposed framework. The performance of the framework is compared to the original BM3D after stage one and two. The dataset of images used in the evaluation is shown in Fig. (2.2). Finally, we will conclude the overall numerical and visual results.

4.1 Numerical Results

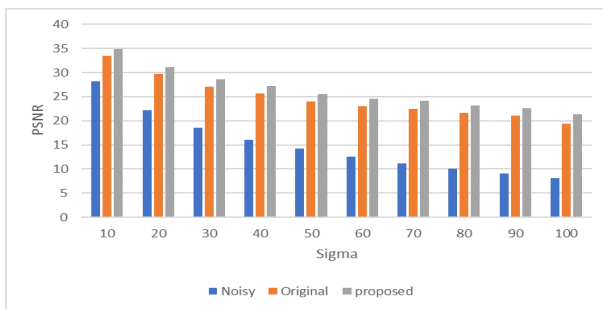
We have applied the proposed framework after both stages one and two of the BM3D scheme on multiple images for various noise levels 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. The results include the PSNR of the noisy image, the original BM3D image after stage one and stage two, the images obtained from using the different kernels (square, circle, horizontal, vertical, diagonal 134 and diagonal 45) and the image generated from the proposed framework. For Stage one results, see tables 4.1, 4.3, 4.5, 4.7, 4.9, 4.11, 4.13, 4.15. For Stage two results, see tables 4.2, 4.4, 4.6, 4.8, 4.10, 4.12, 4.14, 4.16. The results show that the performance of the proposed framework surpasses the performance of the original BM3D scheme. This is true, although the denoising performance of each kernel is slightly less than the original BM3D and in some cases.

Table 4.1: Barbara image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.

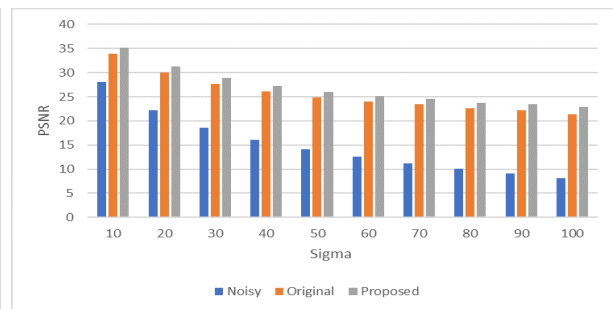
Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal	Diagonal		Proposed	Improv.
								135	45		
10	28.12	33.54	33.55	33.15	33.30	33.43	33.47	33.17	34.89	1.35	
20	22.14	29.73	29.66	29.18	29.53	29.59	29.63	29.44	31.14	1.41	
30	18.58	27.13	27.04	26.67	27.00	27.03	27.00	26.90	28.57	1.44	
40	16.11	25.65	25.54	25.15	25.48	25.51	25.50	25.40	27.18	1.53	
50	14.18	24.01	23.90	23.57	23.86	23.85	23.89	23.76	25.57	1.56	
60	12.52	23.06	22.98	22.61	22.89	22.91	22.93	22.80	24.53	1.47	
70	11.24	22.53	22.44	22.06	22.35	22.39	22.37	22.21	24.11	1.58	
80	10.10	21.69	21.63	21.35	21.56	21.51	21.64	21.42	23.15	1.46	
90	9.05	21.09	21.02	20.80	20.88	20.96	20.98	20.80	22.65	1.56	
100	8.15	19.45	19.50	18.85	18.98	19.29	19.29	18.65	21.40	1.95	

Table 4.2: Barbara image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal	Diagonal		Proposed	Improv.
								135	45		
10	28.12	33.97	34.38	34.38	34.38	34.39	34.40	34.44	35.21	1.24	
20	22.14	30.01	30.52	30.49	30.49	30.51	30.51	30.52	31.32	1.31	
30	18.58	27.62	28.23	28.17	28.17	28.21	28.20	28.17	28.93	1.31	
40	16.11	26.09	26.65	26.63	26.64	26.65	26.65	26.64	27.16	1.07	
50	14.18	24.87	25.49	25.46	25.49	25.49	25.47	25.46	25.98	1.11	
60	12.52	24.00	24.63	24.60	24.62	24.63	24.62	24.60	25.11	1.11	
70	11.24	23.48	24.16	24.09	24.15	24.14	24.15	24.09	24.64	1.16	
80	10.10	22.61	23.25	23.20	23.24	23.22	23.26	23.19	23.74	1.13	
90	9.05	22.22	22.92	22.88	22.91	22.91	22.90	22.86	23.42	1.20	
100	8.15	21.32	22.38	22.31	22.36	22.38	22.34	22.33	22.96	1.64	



(a)



(b)

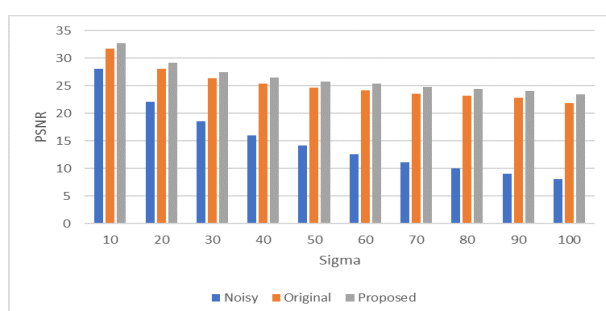
Figure 4.1: Barbara image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 4.3: Baboon image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.

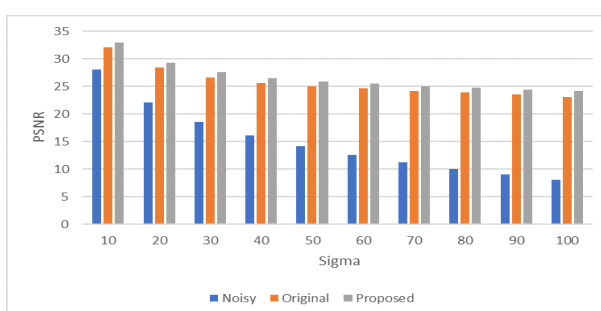
Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.10	31.72	31.69	31.58	31.57	31.67	31.65	31.55	32.80	1.08
20	22.15	28.13	28.08	28.03	27.99	28.06	28.06	28.00	29.18	1.05
30	18.56	26.42	26.38	26.32	26.29	26.37	26.36	26.28	27.49	1.07
40	16.06	25.39	25.36	25.26	25.23	25.32	25.33	25.21	26.51	1.12
50	14.18	24.66	24.63	24.51	24.52	24.58	24.59	24.47	25.82	1.16
60	12.57	24.19	24.17	24.04	24.02	24.12	24.11	23.94	25.45	1.26
70	11.20	23.58	23.57	23.42	23.38	23.49	23.51	23.30	24.81	1.23
80	10.05	23.19	23.18	23.01	22.97	23.08	23.11	22.86	24.49	1.30
90	9.05	22.80	22.80	22.61	22.57	22.73	22.71	22.47	24.10	1.30
100	8.13	21.84	21.94	21.37	21.41	21.75	21.71	21.07	23.49	1.65

Table 4.4: Baboon image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.10	32.06	32.53	32.52	32.51	32.53	32.53	32.50	32.98	0.92
20	22.15	28.39	28.94	28.91	28.89	28.92	28.92	28.86	29.30	0.91
30	18.56	26.65	27.20	27.16	27.16	27.19	27.18	27.13	27.55	0.90
40	16.06	25.69	26.23	26.21	26.20	26.22	26.22	26.17	26.53	0.84
50	14.18	25.04	25.55	25.53	25.53	25.54	25.54	25.50	25.86	0.82
60	12.57	24.65	25.20	25.17	25.17	25.19	25.18	25.14	25.54	0.89
70	11.20	24.16	24.71	24.68	24.67	24.68	24.71	24.65	25.06	0.90
80	10.05	23.87	24.42	24.40	24.38	24.40	24.41	24.37	24.79	0.92
90	9.05	23.51	24.06	24.03	24.04	24.05	24.05	24.03	24.45	0.94
100	8.13	23.13	23.79	23.74	23.77	23.78	23.77	23.75	24.22	1.09



(a)



(b)

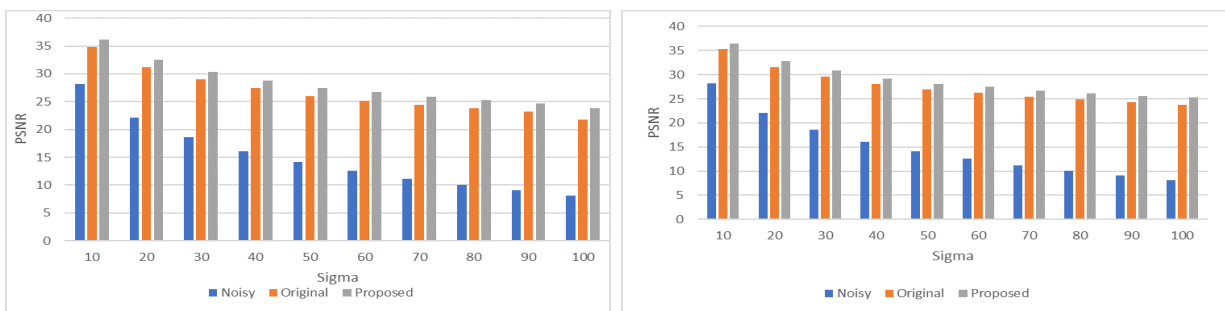
Figure 4.2: Baboon image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 4.5: Boat image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.16	34.79	34.76	34.56	34.56	34.73	34.69	34.54	36.10	1.31
20	22.10	31.16	31.08	30.99	31.00	31.07	31.07	31.01	32.50	1.34
30	18.60	29.03	28.93	28.85	28.88	28.90	28.95	28.86	30.39	1.36
40	16.10	27.42	27.35	27.16	27.25	27.28	27.34	27.16	28.83	1.41
50	14.14	26.05	25.99	25.75	25.84	25.94	25.94	25.71	27.48	1.43
60	12.59	25.21	25.15	24.89	25.02	25.09	25.09	24.84	26.68	1.47
70	11.20	24.46	24.42	24.16	24.25	24.34	24.37	24.11	25.93	1.47
80	10.10	23.82	23.77	23.58	23.63	23.72	23.70	23.52	25.27	1.45
90	9.06	23.18	23.14	22.97	22.91	23.07	23.08	22.84	24.64	1.46
100	8.16	21.83	21.92	21.32	21.25	21.67	21.69	20.92	23.78	1.95

Table 4.6: Boat image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.16	35.26	35.74	35.71	35.75	35.74	35.74	35.75	36.42	1.16
20	22.10	31.57	32.14	32.10	32.11	32.13	32.13	32.11	32.79	1.22
30	18.60	29.56	30.19	30.16	30.15	30.17	30.18	30.15	30.82	1.26
40	16.10	28.09	28.70	28.67	28.68	28.68	28.68	28.66	29.11	1.02
50	14.14	26.97	27.64	27.61	27.60	27.61	27.63	27.60	28.05	1.08
60	12.59	26.28	27.00	26.95	26.96	26.98	26.98	26.95	27.44	1.16
70	11.20	25.41	26.17	26.14	26.13	26.16	26.16	26.13	26.62	1.21
80	10.10	24.87	25.62	25.57	25.57	25.59	25.60	25.56	26.05	1.18
90	9.06	24.29	25.06	25.01	25.01	25.03	25.04	25.00	25.50	1.21
100	8.16	23.70	24.71	24.64	24.65	24.66	24.70	24.65	25.21	1.51



(a)

(b)

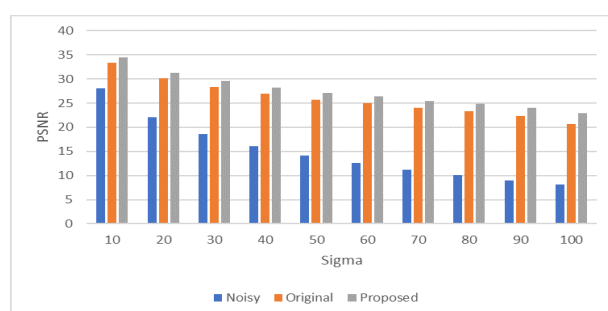
Figure 4.3: Boat image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 4.7: Cameraman image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.

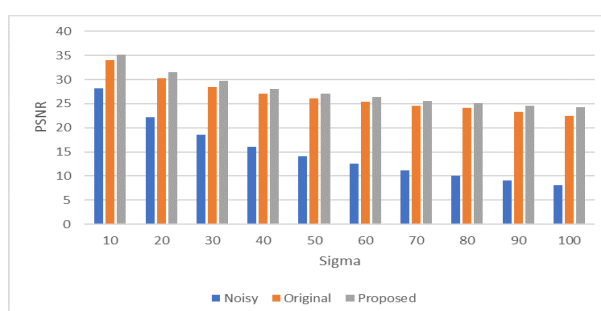
Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.14	33.33	33.40	32.98	33.03	33.29	33.26	32.87	34.56	1.23
20	22.13	30.10	30.09	29.85	29.87	30.02	30.01	29.79	31.34	1.24
30	18.60	28.30	28.25	28.09	28.09	28.21	28.19	28.04	29.59	1.29
40	16.09	26.93	26.85	26.74	26.79	26.84	26.81	26.73	28.28	1.35
50	14.14	25.75	25.65	25.55	25.59	25.64	25.62	25.55	27.12	1.37
60	12.57	24.96	24.86	24.84	24.74	24.83	24.86	24.75	26.39	1.43
70	11.23	24.03	23.96	23.80	23.84	23.92	23.91	23.76	25.43	1.40
80	10.06	23.32	23.24	23.13	23.12	23.20	23.21	23.07	24.81	1.49
90	9.02	22.34	22.31	22.02	22.07	22.23	22.21	21.89	23.97	1.63
100	8.10	20.62	20.71	20.00	20.05	20.49	20.53	19.63	22.90	2.28

Table 4.8: Cameraman image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.14	34.00	34.47	34.40	34.43	34.45	34.44	34.41	35.11	1.11
20	22.13	30.25	30.76	30.68	30.70	30.74	30.72	30.68	31.47	1.22
30	18.60	28.51	29.04	28.95	28.96	29.02	29.01	28.96	29.78	1.27
40	16.09	27.07	27.55	27.56	27.56	27.56	27.55	27.57	28.02	0.95
50	14.14	26.11	26.61	26.62	26.62	26.62	26.61	26.63	27.07	0.96
60	12.57	25.38	25.91	25.90	25.90	25.90	25.91	25.90	26.36	0.98
70	11.23	24.56	25.12	25.10	25.12	25.12	25.11	25.11	25.57	1.01
80	10.06	24.08	24.73	24.72	24.70	24.74	24.70	24.71	25.18	1.10
90	9.02	23.33	24.04	24.02	24.02	24.02	24.04	24.01	24.51	1.18
100	8.10	22.51	23.68	23.60	23.62	23.65	23.65	23.60	24.24	1.73



(a)



(b)

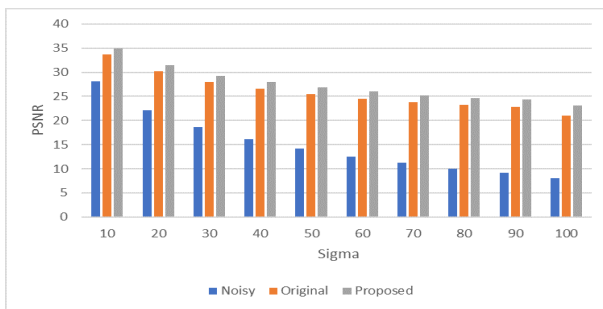
Figure 4.4: Cameraman image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 4.9: Goldhill image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.

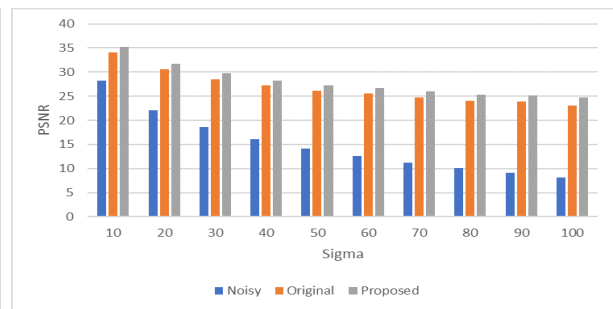
Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal	Diagonal		Proposed	Improv.
								135	45		
10	28.16	33.71	33.69	33.43	33.49	33.65	33.63	33.44	34.97	1.26	
20	22.08	30.17	30.11	29.94	29.99	30.08	30.07	29.93	31.49	1.32	
30	18.57	27.93	27.86	27.66	27.79	27.85	27.83	27.73	29.25	1.32	
40	16.07	26.54	26.45	26.33	26.38	26.43	26.44	26.38	27.97	1.43	
50	14.18	25.47	25.40	25.25	25.30	25.38	25.34	25.25	26.88	1.41	
60	12.57	24.56	24.50	24.30	24.41	24.41	24.46	24.27	25.99	1.43	
70	11.26	23.75	23.70	23.49	23.55	23.66	23.61	23.46	25.20	1.45	
80	10.04	23.24	23.20	22.97	23.06	23.15	23.11	22.92	24.70	1.46	
90	9.09	22.81	22.78	22.58	22.55	22.66	22.73	22.51	24.31	1.50	
100	8.08	21.00	21.06	20.48	20.43	20.85	20.80	20.15	23.09	2.09	

Table 4.10: Goldhill image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal	Diagonal		Proposed	Improv.
								135	45		
10	28.16	34.11	34.59	34.57	34.57	34.60	34.57	34.56	35.18	1.07	
20	22.08	30.56	31.11	31.06	31.09	31.11	31.09	31.06	31.70	1.14	
30	18.57	28.50	29.12	29.06	29.10	29.13	29.09	29.08	29.69	1.19	
40	16.07	27.28	27.84	27.80	27.82	27.83	27.83	27.81	28.26	0.98	
50	14.18	26.20	26.81	26.75	26.79	26.78	26.79	26.75	27.23	1.03	
60	12.57	25.63	26.27	26.23	26.24	26.23	26.28	26.24	26.72	1.09	
70	11.26	24.77	25.47	25.40	25.46	25.43	25.47	25.44	25.93	1.16	
80	10.04	24.10	24.83	24.79	24.81	24.82	24.81	24.80	25.31	1.21	
90	9.09	23.93	24.70	24.65	24.64	24.65	24.70	24.63	25.17	1.24	
100	8.08	23.02	24.12	24.06	24.08	24.11	24.10	24.08	24.70	1.68	



(a)



(b)

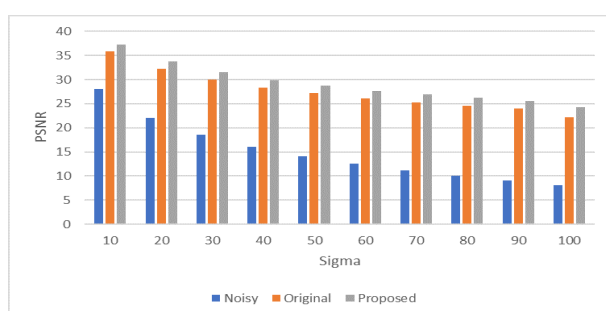
Figure 4.5: Goldhill image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 4.11: Lena image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.

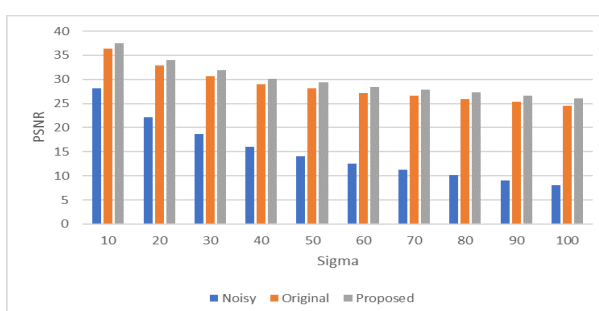
Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.11	35.87	35.83	35.61	35.71	35.75	35.81	35.61	37.23	1.36
20	22.11	32.30	32.19	32.12	32.16	32.15	32.20	32.16	33.77	1.47
30	18.61	30.04	29.94	29.86	29.87	29.90	29.94	29.85	31.52	1.48
40	16.05	28.31	28.20	28.13	28.15	28.14	28.23	28.13	29.84	1.53
50	14.14	27.23	27.14	27.02	27.03	27.06	27.12	26.96	28.79	1.56
60	12.56	26.05	25.97	25.85	25.80	25.89	25.94	25.76	27.63	1.58
70	11.24	25.25	25.15	25.09	25.00	25.05	25.17	24.98	26.90	1.65
80	10.10	24.53	24.43	24.40	24.23	24.36	24.40	24.26	26.20	1.67
90	9.09	23.95	23.91	23.74	23.63	23.79	23.84	23.54	25.59	1.64
100	8.14	22.22	22.30	21.78	21.62	22.01	22.08	21.31	24.28	2.06

Table 4.12: Lena image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.11	36.34	36.82	36.83	36.86	36.82	36.84	36.88	37.47	1.13
20	22.11	32.83	33.43	33.44	33.44	33.42	33.44	33.45	34.06	1.23
30	18.61	30.67	31.34	31.32	31.32	31.33	31.34	31.32	31.91	1.24
40	16.05	28.99	29.61	29.62	29.61	29.61	29.61	29.63	30.06	1.07
50	14.14	28.22	28.90	28.89	28.90	28.89	28.91	28.91	29.36	1.14
60	12.56	27.21	27.94	27.93	27.93	27.93	27.95	27.94	28.40	1.19
70	11.24	26.60	27.38	27.35	27.37	27.35	27.39	27.36	27.86	1.26
80	10.10	25.93	26.75	26.73	26.73	26.73	26.75	26.74	27.25	1.32
90	9.09	25.41	26.19	26.17	26.17	26.17	26.19	26.18	26.68	1.27
100	8.14	24.46	25.46	25.43	25.47	25.45	25.46	25.47	26.01	1.55



(a)



(b)

Figure 4.6: Lena image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 4.13: Couple image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal	Diagonal		Proposed	Improv.
								135	45		
10	28.15	33.65	33.63	33.36	33.42	33.58	33.57	33.34	34.97	1.32	
20	22.07	29.76	29.69	29.56	29.60	29.67	29.68	29.58	31.15	1.39	
30	18.61	27.49	27.40	27.32	27.36	27.43	27.38	27.35	28.83	1.34	
40	16.04	26.12	26.05	25.98	25.96	26.01	26.05	25.95	27.46	1.34	
50	14.15	24.87	24.79	24.70	24.73	24.78	24.77	24.71	26.22	1.35	
60	12.56	24.04	23.98	23.85	23.90	23.95	23.94	23.85	25.37	1.33	
70	11.22	23.29	23.24	23.12	23.13	23.20	23.20	23.09	24.62	1.33	
80	10.10	22.82	22.78	22.64	22.64	22.74	22.72	22.60	24.15	1.33	
90	9.05	22.29	22.26	22.08	22.09	22.20	22.19	21.97	23.67	1.38	
100	8.14	21.08	21.17	20.62	20.60	20.98	20.96	20.28	22.91	1.83	

Table 4.14: Couple image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal	Diagonal		Proposed	Improv.
								135	45		
10	28.15	34.05	34.53	34.53	34.51	34.54	34.53	34.53	35.19	1.14	
20	22.07	30.28	30.87	30.84	30.83	30.86	30.85	30.82	31.51	1.23	
30	18.61	28.10	28.75	28.70	28.71	28.74	28.72	28.69	29.34	1.24	
40	16.04	26.75	27.36	27.33	27.33	27.35	27.35	27.32	27.74	0.99	
50	14.15	25.72	26.36	26.33	26.32	26.34	26.35	26.32	26.74	1.02	
60	12.56	24.96	25.62	25.58	25.59	25.60	25.61	25.58	26.01	1.05	
70	11.22	24.28	24.96	24.92	24.93	24.95	24.95	24.92	25.37	1.09	
80	10.10	23.71	24.39	24.34	24.35	24.37	24.37	24.34	24.77	1.06	
90	9.05	23.24	23.96	23.87	23.93	23.93	23.94	23.89	24.36	1.12	
100	8.14	22.67	23.51	23.44	23.46	23.47	23.49	23.44	23.97	1.30	

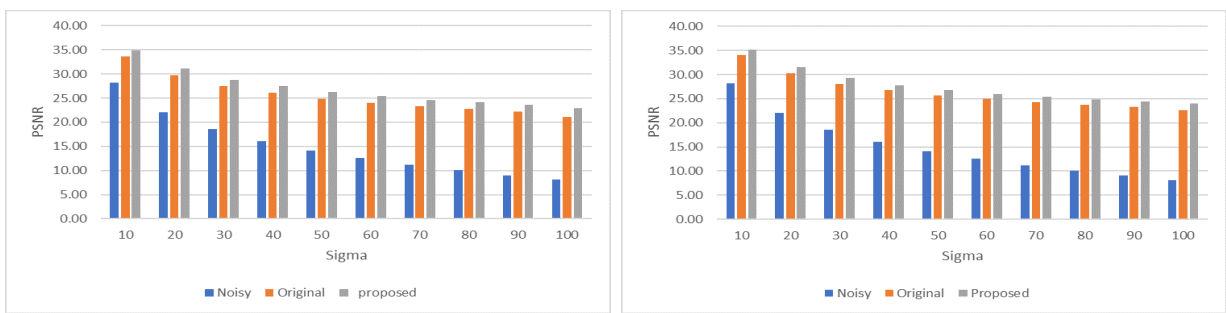


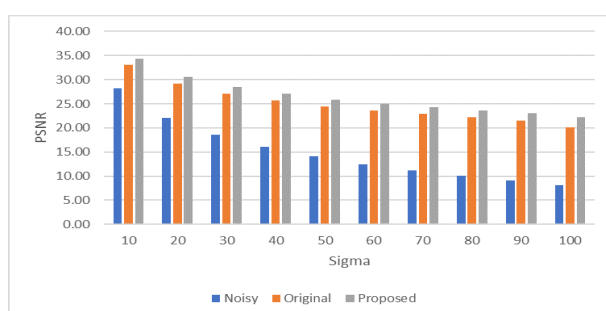
Figure 4.7: Couple image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 4.15: Lake image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.

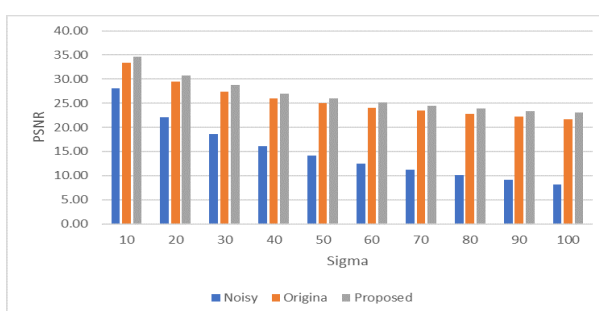
Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.15	33.02	33.02	32.76	32.73	32.95	32.95	32.69	34.30	1.28
20	22.07	29.21	29.14	28.98	29.04	29.11	29.13	29.03	30.54	1.33
30	18.55	27.09	26.99	26.91	26.94	26.97	26.99	26.96	28.46	1.37
40	16.12	25.65	25.54	25.47	25.53	25.54	25.54	25.52	27.05	1.40
50	14.14	24.51	24.42	24.31	24.35	24.41	24.38	24.32	25.90	1.39
60	12.52	23.55	23.45	23.41	23.37	23.47	23.42	23.40	24.98	1.43
70	11.23	22.85	22.77	22.68	22.69	22.75	22.75	22.68	24.25	1.40
80	10.09	22.15	22.07	21.99	21.99	22.04	22.06	21.96	23.56	1.41
90	9.08	21.48	21.43	21.27	21.23	21.40	21.33	21.13	22.98	1.50
100	8.14	20.16	20.22	19.62	19.64	20.01	20.01	19.27	22.16	2.00

Table 4.16: Lake image PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.15	33.43	33.85	33.83	33.83	33.85	33.85	33.85	34.65	1.22
20	22.07	29.43	29.95	29.90	29.93	29.93	29.94	29.92	30.76	1.33
30	18.55	27.43	28.00	27.92	27.97	27.99	27.99	27.97	28.79	1.36
40	16.12	26.04	26.58	26.55	26.57	26.58	26.57	26.55	26.98	0.94
50	14.14	24.99	25.58	25.55	25.57	25.57	25.57	25.55	25.98	0.99
60	12.52	24.09	24.69	24.67	24.68	24.68	24.69	24.67	25.11	1.02
70	11.23	23.47	24.10	24.07	24.06	24.08	24.09	24.06	24.50	1.03
80	10.09	22.84	23.47	23.44	23.44	23.45	23.46	23.44	23.88	1.04
90	9.08	22.30	22.97	22.95	22.95	22.97	22.95	22.95	23.39	1.09
100	8.14	21.66	22.57	22.51	22.52	22.55	22.55	22.50	23.05	1.39



(a)



(b)

Figure 4.8: Lake image PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

Table 4.17: Average PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage one form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.14	33.70	33.70	33.43	33.48	33.63	33.63	33.40	34.98	1.27
20	22.11	30.07	30.01	29.83	29.90	29.97	29.98	29.87	31.39	1.32
30	18.59	27.93	27.85	27.71	27.78	27.83	27.83	27.75	29.26	1.33
40	16.08	26.50	26.42	26.28	26.35	26.38	26.41	26.31	27.89	1.39
50	14.16	25.32	25.24	25.08	25.15	25.21	25.21	25.09	26.72	1.40
60	12.56	24.45	24.38	24.22	24.27	24.33	24.34	24.20	25.88	1.43
70	11.23	23.72	23.66	23.48	23.52	23.60	23.61	23.45	25.16	1.44
80	10.08	23.10	23.04	22.88	22.90	22.98	22.99	22.83	24.54	1.45
90	9.06	22.49	22.46	22.26	22.24	22.38	22.38	22.14	23.99	1.50
100	8.13	21.03	21.10	20.51	20.50	20.88	20.88	20.16	23.00	1.98

Table 4.18: Average PSNR for noisy, original BM3D, six kernel images, the pixel proposed framework and improvement after stage two form 10 to 100 sigma noise levels.

Sigma	Noisy	Original	Square	Circle	Horizontal	Vertical	Diagonal		Proposed	Improv.
							135	45		
10	28.14	34.15	34.61	34.60	34.61	34.62	34.61	34.62	35.28	1.12
20	22.11	30.42	30.97	30.93	30.94	30.95	30.95	30.93	31.61	1.20
30	18.59	28.38	28.98	28.93	28.94	28.97	28.96	28.93	29.60	1.22
40	16.08	27.00	27.57	27.55	27.55	27.56	27.56	27.54	27.98	0.98
50	14.16	26.02	26.62	26.59	26.60	26.61	26.61	26.59	27.03	1.02
60	12.56	25.28	25.91	25.88	25.89	25.89	25.90	25.88	26.34	1.06
70	11.23	24.59	25.26	25.22	25.24	25.24	25.25	25.22	25.69	1.10
80	10.08	24.00	24.68	24.65	24.65	24.67	24.67	24.64	25.12	1.12
90	9.06	23.53	24.24	24.20	24.21	24.22	24.23	24.19	24.69	1.16
100	8.13	22.81	23.78	23.72	23.74	23.76	23.76	23.73	24.30	1.49

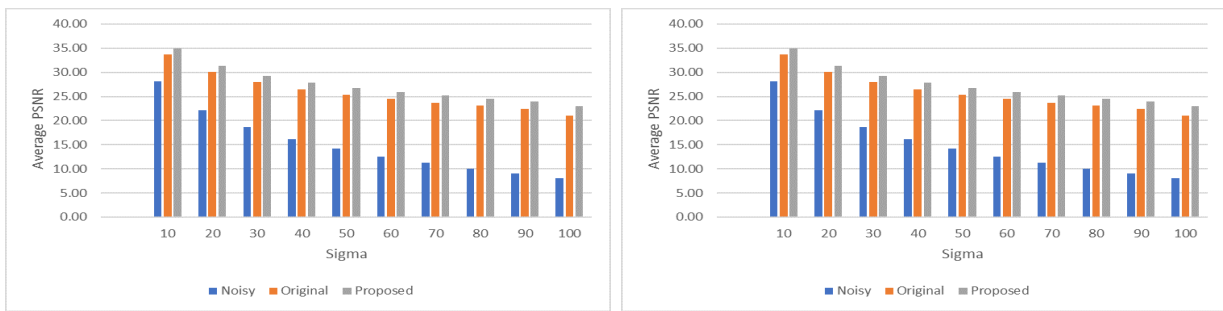


Figure 4.9: Average PSNR comparison between noisy, original BM3D and the proposed framework at various noise levels (Sigma). (a) After stage one. (b) After stage two.

4.1.1 Discussion on Numerical Results

The previously discussed results show an improvement in the PSNR between the original BM3D and the final result of the proposed framework. Despite the fact that the six different kernel images have lower PSNR in stage one. For instance the average table (4.17) the square kernel, circular kernel, horizontal kernel, vertical kernel, diagonal 135, and diagonal 45, for all sigma values, are less than original BM3D stage one result by 0.04, 0.26, 0.22, 0.11, 0.11, and 0.31 respectively. Also, the square kernel in some cases gives the same or better results than original BM3D such as Cameraman, Lake images.

The framework stage one output image is the input of stage two. Consequently, stage two six kernels give higher results than stage two original BM3D. As shown in average table (4.18), the square kernel, circular kernel, horizontal kernel, vertical kernel, diagonal 135, and diagonal 45, for all sigma values, are more than original BM3D stage two result by 0.64, 0.61, 0.62, 0.63, 0.63, and 0.61 respectively. Finally, the combined image produced after stage two gives a significant increase with respect to the PSNR. The average improvement between the original BM3D and the proposed work for the given dataset in Fig. (2.2) is approximately 1.15 dB.

4.2 Visual Results

In this section, we show the improvement of the framework on a sample image (Couple image). The results is viewed at different noise levels 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 as shown in Fig. (4.10), Fig. (4.11), Fig. (4.12), Fig. (4.13), Fig. (4.14), Fig. (4.15), Fig. (4.16), Fig. (4.17), Fig. (4.18) and Fig. (4.19) respectively. We compare the original image, noisy image with the corresponding stage one, two of the original BM3D in addition to, stage one, two of the proposed framework.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 4.10: Denoising Couple image with $\sigma = 10$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.



(a)



(b)



(c)



(d)



(e)

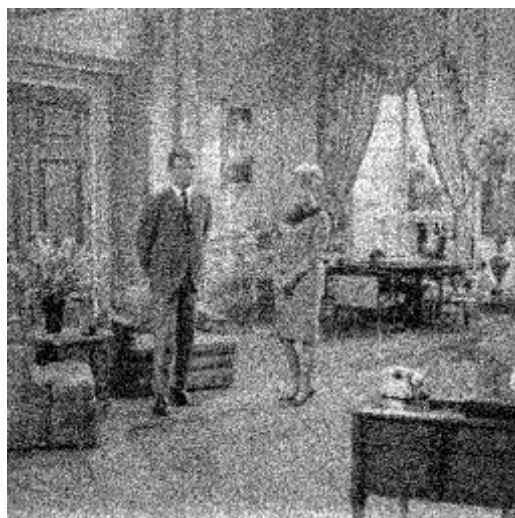


(f)

Figure 4.11: Denoising Couple image with $\sigma = 20$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.



(a)



(b)



(c)



(d)



(e)

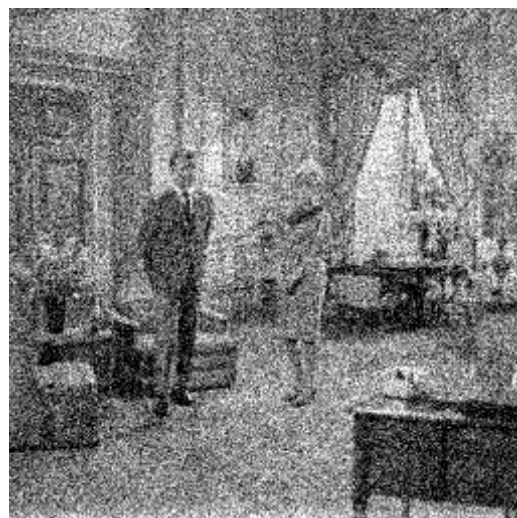


(f)

Figure 4.12: Denoising Couple image with $\sigma = 30$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 4.13: Denoising Couple image with $\sigma = 40$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 4.14: Denoising Couple image with $\sigma = 50$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.



(a)



(b)



(c)



(d)



(e)

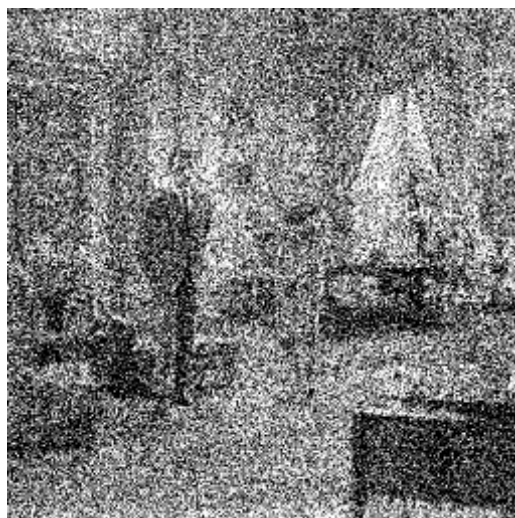


(f)

Figure 4.15: Denoising Couple image with $\sigma = 60$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.



(a)



(b)



(c)



(d)



(e)

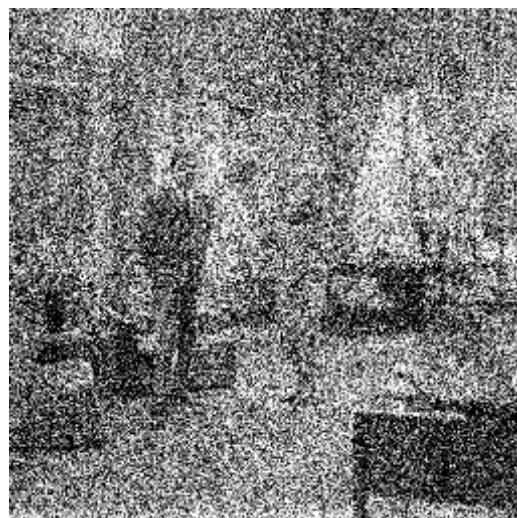


(f)

Figure 4.16: Denoising Couple image with $\sigma = 70$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.



(a)



(b)



(c)



(d)



(e)

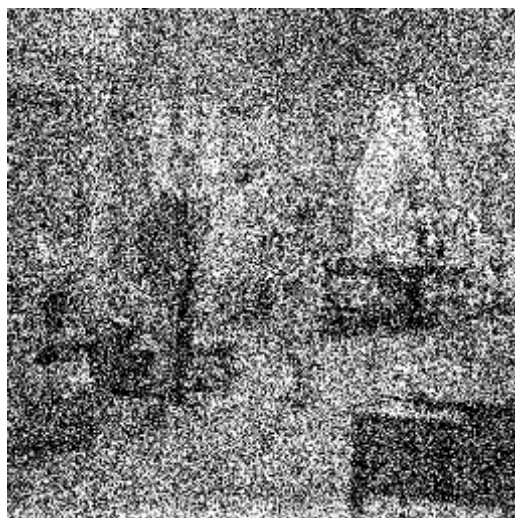


(f)

Figure 4.17: Denoising Couple image with $\sigma = 80$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.



(a)



(b)



(c)



(d)



(e)

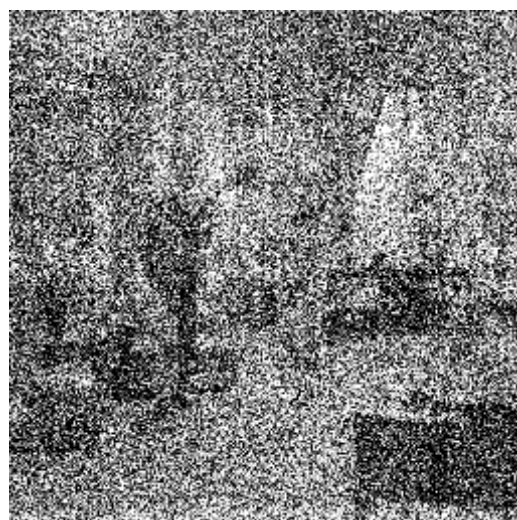


(f)

Figure 4.18: Denoising Couple image with $\sigma = 90$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 4.19: Denoising Couple image with $\sigma = 100$ using original BM3D and the proposed framework. (a) Original image. (b) Noisy image. (c) Original BM3D stage one. (d) Original BM3D stage two. (e) Proposed BM3D stage one. (f) Proposed framework stage two.

4.2.1 Discussion on Visual Results

The proposed framework visual results at low noise levels cannot be spotted visually. However, for high noise levels, areas like the floor in Couples images appear smoother and details like the door area in the images appear more clear than original BM3D results in both stages one and two.

5 Conclusion and Future Work

5.1 Conclusion

In this chapter, we will summarize the conclusion of our framework and the possible approaches that can be considered to apply the combination of various kernel images automatically.

In this framework, we conduct an experiment to improve denoising performance using shape adaptive shapes. First, The results show, for low noise images ($\sigma = 10, 20, \text{ and } 30$), the average improvement of the pixel-based over the eight images is 1.18 approximately. As for high noise images ($\sigma = 40, 50, 60, 70, 80, 90, \text{ and } 100$), the average improvement of the pixel-based over the same images is 1.13 approximately. Second, the pixel-based combination approach is better than the patch-based approach. However, the patch approach has less computational requirements than the pixel approach.

In the proposed approach, an expansion to the BM3D algorithm is introduced. This method outperforms the original BM3D scheme, where the minor details are preserved while maintaining the smoothness of texture-less areas. The proposed geometric kernels enable shape adaptivity to the inner image structure details.

5.2 Future Work

The proposed approach can be improved in many ways:

- The pixels or patches angles might be a good representation of the relationship between

the kernel used and the pixel location.

- Supervised learning can be used to learn how to choose the pixels or patches with the least error value from the output images. Networks used can be as simple as multilayer perceptron (MLP) or more sophisticated as a deep learning network as Deep convolution neural network (DCNN).
- Mean square error estimators as [27] can be used to give the option of discarding the reference image from the calculations.
- Parallelisation in the aggregation step can be applied to generate various images at the same time instead of running BM3D multiple times.

Bibliography

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006, ISBN: 013168728X.
- [2] Y. E. Gökdağ, F. Şansal, and Y. D. Gökdel, “Image denoising using 2-d wavelet algorithm for gaussian-corrupted confocal microscopy images,” *Biomedical Signal Processing and Control*, vol. 54, p. 101594, 2019, ISSN: 1746-8094. DOI: <https://doi.org/10.1016/j.bspc.2019.101594>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1746809419301740>.
- [3] H. Koyuncu and R. Ceylan, “Elimination of white gaussian noise in arterial phase ct images to bring adrenal tumours into the forefront,” *Computerized Medical Imaging and Graphics*, vol. 65, pp. 46–57, 2018, Advances in Biomedical Image Processing, ISSN: 0895-6111. DOI: <https://doi.org/10.1016/j.compmedimag.2017.05.004>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0895611117300472>.
- [4] H. V. Bhujle and B. H. Vadavadagi, “Nlm based magnetic resonance image denoising—a review,” *Biomedical Signal Processing and Control*, vol. 47, pp. 252–261, 2019.
- [5] X. J. He, Y. Q. Wang, Y. Li, and A. G. Xu, “A novel local variance-based filtering method for denoising remote sensing images,” *Remote Sensing Letters*, vol. 10, no. 8, pp. 736–745, 2019. DOI: 10.1080/2150704X.2019.1606469. eprint: <https://doi.org/10.1080/2150704X.2019.1606469>. [Online]. Available: <https://doi.org/10.1080/2150704X.2019.1606469>.
- [6] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Jan. 1998, pp. 839–846. DOI: 10.1109/ICCV.1998.710815.

- [7] B. Goyal, A. Dogra, S. Agrawal, and B. S. Sohi, “A three stage integrated denoising approach for grey scale images,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–16, 2018. DOI: 10.1007/s12652-018-1019-5. [Online]. Available: <https://app.dimensions.ai/details/publication/pub.1106706026>.
- [8] A. Buades, B. Coll, and J. M. Morel, “On image denoising methods,” *CMLA Preprint*, vol. 5, 2004.
- [9] A. Buades, B. Coll, and J.-M. Morel, “Non-local means denoising,” *Image Processing On Line*, vol. 1, pp. 208–212, 2011.
- [10] ———, “Self-similarity-based image denoising,” *Commun. ACM*, vol. 54, no. 5, pp. 109–117, May 2011, ISSN: 0001-0782. DOI: 10.1145/1941487.1941513. [Online]. Available: <http://doi.acm.org/10.1145/1941487.1941513>.
- [11] M. H. Alkinani and M. R. El-Sakka, “Patch-based models and algorithms for image denoising: A comparative review between patch-based images denoising methods for additive noise reduction,” *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, pp. 1–27, 2017.
- [12] C.-A. Deledalle, V. Duval, and J. Salmon, “Non-local methods with shape-adaptive patches (nlm-sap),” *J. Math. Imaging Vis.*, vol. 43, no. 2, pp. 103–120, Jun. 2012, ISSN: 0924-9907. DOI: 10.1007/s10851-011-0294-y. [Online]. Available: <http://dx.doi.org/10.1007/s10851-011-0294-y>.
- [13] H. Bhujle and S. Chaudhuri, “Accelerating non-local denoising with a patch based dictionary,” in *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, ser. ICVGIP ’12, Mumbai, India: ACM, 2012, 16:1–16:8, ISBN: 978-1-4503-1660-6. DOI: 10.1145/2425333.2425349. [Online]. Available: <http://doi.acm.org/10.1145/2425333.2425349>.
- [14] S. Sreehari, S. Venkatakrishnan, L. Drummy, J. Simmons, and C. A. Bouman, “Rotationally-invariant non-local means for image denoising and tomography,” in *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 542–546.
- [15] I. Ram, M. Elad, and I. Cohen, “Image denoising using nl-means via smooth patch ordering,” May 2013, pp. 1350–1354. DOI: 10.1109/ICASSP.2013.6637871.

- [16] J. Salmon and Y. Strozecski, "Patch reprojections for non-local methods," *Signal Processing*, vol. 92, no. 2, pp. 477–489, 2012, ISSN: 0165-1684. DOI: <https://doi.org/10.1016/j.sigpro.2011.08.011>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168411002878>.
- [17] K. N. Chaudhury and A. Singer, "Non-local euclidean medians," *IEEE Signal Processing Letters*, vol. 19, pp. 745–748, 2012.
- [18] K. Leng, "An improved non-local means algorithm for image denoising," in *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, Aug. 2017, pp. 149–153. DOI: 10.1109/SIPROCESS.2017.8124523.
- [19] B. Chen, Y. Yan, L. Wang, M. Chen, C. Wang, and H. Ma, "One dimension nlm denoising method based on hasudorff distance and its application in otlc," in *2019 IEEE Asia Power and Energy Engineering Conference (APEEC)*, Mar. 2019, pp. 75–79. DOI: 10.1109/APEEC.2019.8720666.
- [20] D. Van De Ville and M. Kocher, "Sure-based non-local means," *IEEE Signal Processing Letters*, vol. 16, no. 11, pp. 973–976, Nov. 2009, ISSN: 1558-2361. DOI: 10.1109/LSP.2009.2027669.
- [21] I. Frosio and J. Kautz, "Statistical nearest neighbors for image denoising," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 723–738, 2018.
- [22] C.-A. Deledalle, V. Duval, and J. Salmon, "Non-local methods with shape-adaptive patches (nlm-sap)," *Journal of Mathematical Imaging and Vision*, vol. 43, no. 2, pp. 103–120, Jun. 2012, ISSN: 1573-7683. DOI: 10.1007/s10851-011-0294-y. [Online]. Available: <https://doi.org/10.1007/s10851-011-0294-y>.
- [23] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3d filtering," in *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, International Society for Optics and Photonics, vol. 6064, 2006, p. 606 414.
- [24] M. Alkinani and M. El-Sakka, "Patch-based models and algorithms for image denoising: A comparative review between patch-based images denoising methods for additive noise reduction," *EURASIP Journal on Image and Video Processing*, vol. 2017, p. 58, Aug. 2017. DOI: 10.1186/s13640-017-0203-4.

- [25] Y. Hou and D. Shen, “Image denoising with morphology- and size-adaptive block-matching transform domain filtering,” *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, p. 59, Jul. 2018, ISSN: 1687-5281. DOI: 10.1186/s13640-018-0301-y. [Online]. Available: <https://doi.org/10.1186/s13640-018-0301-y>.
- [26] B. S. Kumar, “Image denoising based on non-local means filter and its method noise thresholding,” *Signal, image and video processing*, vol. 7, no. 6, pp. 1211–1227, 2013.
- [27] D. Brunet, E. R. Vrscay, and Z. Wang, “The use of residuals in image denoising,” in *Proceedings of the 6th International Conference on Image Analysis and Recognition*, ser. ICIAR '09, Halifax, Nova Scotia, Canada: Springer-Verlag, 2009, pp. 1–12, ISBN: 9783642026102. DOI: 10.1007/978-3-642-02611-9_1. [Online]. Available: https://doi.org/10.1007/978-3-642-02611-9_1.

Curriculum Vitae

Name:

Mematallah Massoud.

Post-secondary Education:

Bachelor of Computer Science, the University of Alexandria, Egypt, 2013-2017.

Master of Science, University of Western Ontario, London, ON, 2018-2020.

Honors and Awards:

Western University Graduate Research Scholarship 2018-2020.

Related Work:

Teaching Assistant, Western University 2018-2019.