**COMPARATIVE ANALYSIS OF DIFFERENT IMAGE DENOISING METHODS
IN SPATIAL AND TRANSFORM DOMAIN**

by

Zeeshan Ahmad

MSc (Electrical) 2005,

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, PAKISTAN

BE (Electrical) 2001

NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY. PAKISTAN

A Project

presented to Ryerson University

in partial fulfillment of the

requirement of the degree of

Master of Engineering

in

Electrical and Computer Engineering

Toronto, Ontario, Canada 2017

©Zeeshan Ahmad 2017

I hereby declare that I am the sole author of this Project. This is a true copy of the project, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this project to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this project by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my project may be made electronically available to the public.

**COMPARATIVE ANALYSIS OF DIFFERENT IMAGE DENOISING METHODS
IN SPATIAL AND TRANSFORM DOMAIN**

MASTER OF ENGINEERING 2017

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Zeeshan Ahmad

RYERSON UNIVERSITY

Abstract

Digital Images are the best source for humans to see, visualize, think, extract information and make conclusions. However during the acquisition of images, noise superimposes on the images and reduces the information and detail of the images. In order to restore the details of the images, noise must be reduced from the images. This requirement places the image denoising amongst the fundamental and challenging fields of computer vision and image processing.

In this project six fundamental techniques / algorithms of image denoising in spatial and transform domain are presented and their comparative analysis is also carried out. The noise model used in this project is Additive Gaussian noise. The algorithms are simulated on Matlab and experimental results are shown at different noise levels.

The performance of each image denoising technique is measured in terms of Peak Signal to Noise Ratio (PSNR) , Mean Structural Similarity (SSIM) Metrics and visual quality.

It is observed that the transform domain techniques used in this project achieved better results as compared to spatial domain techniques.

Acknowledgements

I am thankful to Almighty , the most gracious and most merciful, Who gives me the strength and devotion to accomplish this project successfully.

I would like to express my deep gratitude to my project supervisor Dr. Vadim Geurkov for his endless support, wisdom, patience and encouragement.

I would also like to thank all the teachers of Electrical and Computer Engineering department who taught me the best and enable me to work on this project.

Finally I am thankful to my family members for their prayers and encouragement , without their support I would never have completed this project.

Contents

Declaration	ii
Abstract	iii
Acknowledgement	v
List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Image Processing system	2
1.2 Motivation	3
1.3 Objectives of the project	4
1.4 Figure or Merits of project	4
1.5 Structure of Project	6
2. Theory and Back ground	7
2.1 Introduction	7
2.2 Image denoising Problem.	7
2.3 Noise Models.. . . .	8
2.4 Sources of Noise	9
2.5 Image denoising methods.	10
2.6 Summary of the chapter.	14
3. Transform domain image denoising methods	15
3.1 Introduction.	15
3.2 Low pass Filters	16

3.3 Wavelet Transform	27
3.4 Wiener Filtering.	30
3.5 Conclusion about transform domain techniques. . .	35
4. Spatial domain image denoising methods.	36
4.1 Introduction.	36
4.2 Mean Filters	36
4.3 Adaptive median filter	41
4.4 Mid Point Filter	44
4.5 Conclusion about spatial domain techniques	46
5. Conclusions and Future Work	47
5.1 Conclusion	47
5.2 Future Work	48
Appendix	49
References	63

List of Tables

3.1	Performance of ideal low pass filter at different cutoff frequencies and noise levels.	19
3.2	Performance of 6th order butterworth low pass filter at different cutoff frequencies and noise levels.	22
3.3	Performance of 8th order butterworth low pass filter at different cutoff frequencies and noise levels.	23
3.4	Performance of 6th Gaussian low pass filter at different cutoff frequencies and noise levels.. . . .	26
3.5	Performance of Wavelet thresholding for image denoising.	29
3.6	Performance of Wiener Filter for image denoising.. . . .	34
4.1	Performance of arithmetic mean filter with different noise levels and sizes of mask.	38
4.2	Performance of geometric mean filter with different noise levels and sizes of mask.	40
4.3	Optimized size of structural element unit at different noise levels.	42
4.4	Performance of Adaptive median filter with different noise levels and sizes of Mask.	43
4.5	Performance of midpoint filter with different noise levels.. . . .	44

List of Figures

1.1 Stages of typical image processing system	2
2.1 First Level decomposition of the image	13
3.1 2D View of ideal low pass filter with cutoff frequency =50.	17
3.2 Image denoising by ideal low pass filter.	18
3.3 2D View of Butterworth low pass filter with cutoff frequency =50.	21
3.4 Image denoising by Butterworth low pass filter of order '6'.	21
3.5 2D View of Gaussian low pass filter	25
3.6 Image denoising by Gaussian low pass filter.	25
3.7 Plot of Hard and Soft thresholding functions.	28
3.8 Image denoising by Wavelet Transform.	30
3.9 Image denoising by Wiener Filter.	32
3.10 Image denoising by Wiener Filter.	33
4.1 Image denoising by Arithmetic mean filter.	39
4.2 Image denoising by Geometric mean filter.	41
4.3 Image denoising by Adaptive median filter.	43
4.4 Image denoising by Midpoint filter.	45

Chapter 1

Introduction

For more than three decades digital Images are the best source for humans to see, visualize, think, extract information and make conclusions, thus making the field of image processing amongst the fastest growing branches of applied research.

Despite the invention of state of the art technology acquisition devices like digital camera, noise superimposes on the images especially if the acquisition of images take place under poor conditions. The causes of noise ranges from the electronic circuitry of the camera to the communication channel to transmit the images. This noise reduces the information and detail of the images, thus making it nearly impossible for humans and machines to extract information. Extracting information from the noisy images makes the image denoising the most active and challenging research area in the field of image processing and computer vision.

The important property of a good image denoising method is that it should remove noise as much as possible and also preserve the details of the image [7]

In order to extract information and preserve the details from the noisy images, many techniques/methods were proposed to design filters in the spatial domain as well as transform domain. These methods are dependent on the types of noise corrupted the images. The most common forms of noise are additive Gaussian noise, Speckle noise and periodic noise [8]. The type of the noise present on the image can be estimated by inspecting the histogram of the image.

In this project the methods of removing additive Gaussian noise are presented in detail. The comparative analysis of these methods is also carried out and presented in

this project report. The methods presented in this report ranges from spatial to transform domain.

1.1 Image Processing system

A typical image processing system is shown below in figure 1.1. It has three stages

- i. Image Acquisition
- ii. Processing
- iii. Classification / display

Figure 1.1 : Stages of typical image processing system

The brief explanation of each stage is described below

i. Image Acquisition

This stage is responsible for capturing and digitizing images . Examples of general-purpose acquisition devices include scanners, cameras, and camcorders. Acquisition devices can be interfaced with the main computer in a number of ways, for example, USB, FireWire, Camera Link, or Ethernet.

ii. Processing

This is the most challenging and the versatile stage of image processing system because the large number of processes can be applied to images in this stages. these processes include image compression, image segmentation, de-blurring of images, denoising of images and adjusting the size of the image. this stage also responsible for running software for algorithms.

iii. Classification / display

This stage is responsible for showing the image contents (after processing) for human and machine viewing. This stage enables the humans or machines to extract information present in the image

1.2 Motivation

The second stage of the image processing system (described above in section 1.1) is considered as the most challenging stage in which many operations or processes can be applied on the images. Image denoising is amongst one of those challenging fields of image processing. The purpose of image denoising is to restore the original image by retaining maximum details of the image.

Thus the need to remove noise from images and to extract information and to interpret their contents is the driving force that motivates me to do the project on this challenging area of image processing.

1.3 Objectives of the Project

In this project, different techniques / algorithms of image denoising in transform and spatial domains are applied to the noisy images and then comparative analysis of image denoising techniques are explained with the help of experimental results .

The figures of merits used in this project are Peak Signal to Noise Ratio (PSNR) and Mean Structural Similarity Metrics (SSIM) [4]

The details of transform domain techniques and their comparative analysis is presented in chapter 3 of the report while the details and the comparative analysis of spatial domain techniques are presented in chapter 4 of the report.

The quality of each image denoising method in this report is measured against the figures of merit mentioned above.

All the experimental results of the methods that will be explained in the chapter 4 and chapter 5 of the project are implemented in the Matlab.

The Matlab programs are placed in the Appendix of the Report.

1.4 Figure of merits of Project

The figures of merits used in this project for comparison among the methods are Peak Signal to Noise Ratio (PSNR) and Mean Structural Similarity Metrics (SSIM).

i. Peak Signal to Noise Ratio

Peak signal-to-noise ratio(PSNR), is the ratio between the maximum possible power of image and the power of corrupting noise that affects the fidelity of its representation. Because images have a very wide dynamic range therefore PSNR is usually expressed in terms of the logarithmic decibel scale.

Mathematically PSNR is expressed as

$$PSNR(dB) = 10 \log_{10} \left(\frac{255^2}{MSE} \right)$$

where

255 is the maximum possible pixel value of the image when the pixels are represented using 8 bits per sample,

MSE is the mean square error and is defined as

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (X(i,j) - Y(i,j))^2$$

where

X = Original image

y = Restored image

In this project PSNR is expressed in dB in all the experimental results. Higher values of PSNR are required from image denoising algorithms. The values above 20 are acceptable, however values around 30 are considered excellent.

ii. Mean structural similarity metric

The mean structural similarity (SSIM) index is a method for measuring the similarity between two original image and denoised image. The SSIM index is a full reference metric in which the denoised image is compared with the original or reference image. It is the statistical quantity ranges from 0 to 1. the values above 0.6 are acceptable, however the values around 0.8 are considered excellent. Since it is a normalized quantity, it has no unit.

1.5 Structure of the Project Report

The structure of the project report is as follows.

In chapter 2, the background of the image denoising problem is described. In chapter 3 of the report, transform domain methods of image denoising are explained and experimental results are presented. In chapter 4 , spatial domain methods are elaborated and experimental results are presented. In chapter 5, conclusion of the project report is presented.

Chapter 2

Theory and Background

2.1 Introduction

In this chapter the image denoising problem will be explained in detail. In addition, this chapter will also cover the sources of noise from image acquisition to image processing. Different models of noise will also be explained in this chapter and then the brief introduction of image denoising methods in spatial and transform domain will be explained.

2.2 Image Denoising Problem

The formulation of image denoising problem is straight forward. Let ' f ' be the original or ideal image. ' g ' is the noisy image obtained by adding a noise on the original image. The purpose of image denoising is to obtain an estimate ' F ' of the original image ' f '. Mathematically the image denoising formulation can be written as

$$g = f + n \quad (2.1)$$

Where f = Original Image

g = Noisy Image

n = Noise

The selection of image denoising method / technique depends upon the type of noise corrupted the image.

2.3 Noise Models

Noise is the unwanted or undesired artifact that contaminates the images. The most common noise models are following.

1. Gaussian Noise
2. Impulse (Salt and Pepper)
3. Uniform Noise

2.3.1 Gaussian Noise

The PDF of the Gaussian Noise is given by [2]

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

where x represents the gray level, μ is its mean and σ is its standard deviation.

2.3.2 Impulse (Salt and Pepper) Noise

The PDF of the impulse noise is given by [2]

$$P(x) = \begin{cases} P_p & \text{for } x = p \\ P_s & \text{for } x = s \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

Where P_p and P_s are the probabilities of occurrence of pixels whose values are equal to p (Pepper) and s (salt) respectively.

2.3.3 Uniform Noise

The PDF of the impulse noise is given by [2]

$$P(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases} \quad (2.4)$$

The mean of the uniform noise is given by

$$\mu = \frac{a + b}{2}$$

whereas the variance is given by

$$\sigma^2 = \frac{(b - a)^2}{12}$$

2.4 Sources of Noise

Noise corrupts the images at any stage from acquisition to processing and transmitting through channels. Image noise may be caused by different intrinsic (i.e., sensor) and extrinsic (i.e., environment) conditions which are often not possible to avoid. Intrinsic noise is also caused due to motion of camera, lens position and thermal condition of electronic circuitry of camera while the extrinsic noise is caused due to poor environmental conditions.

Different models of noise are explained above in section 2.2, however in this project only the Additive Gaussian Noise model is considered which is the most common form of noise superimposed on the images.

2.5 Image Denoising methods

There are two basics or fundamental methods of image denoising [5]

1. Spatial domain methods or filtering
2. Transform domain methods or filtering

2.5.1 Spatial Domain methods

Spatial domain methods are the oldest or traditional methods of image denoising.

They are further subdivided into Linear and Non Linear Filters.

2.5.1.1 Linear Filters

The most commonly used linear filter is the mean filter and is also known as averaging neighborhood filter. It uses convolution with a (usually 3×3) mask whose coefficients have a value of 1 and divides the result by a scaling factor (the total number of elements in the mask). A neighborhood averaging filter in which all coefficients are equal is also referred to as a box filter.

The convolution mask for a 3×3 mean filter is given by

$$h(x, y) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

However larger values of neighborhood can also be selected. The selection of the size of the neighborhood depends upon the level of noise contaminates the image. the other variations of mean filters are Geometric mean and Harmonic mean filters.

2.5.1.2 Non Linear Filters

The nonlinear filters also work at a neighborhood level, but do not process the pixel values using the convolution operator. Instead, they usually apply a ranking (sorting) function to the pixel values within the neighborhood and select a value from the sorted list. For this reason, these are sometimes called Rank filters [2]. The most commonly used Non linear filter for image denoising is the median filter. It works by sorting the pixel values within a neighborhood, finding the median value, and replacing the original pixel value with the median of that neighborhood.

if $h(x, y) = \begin{bmatrix} 9 & 12 & 0 \\ 5 & 5 & 9 \\ 8 & 10 & 7 \end{bmatrix}$ then $[9 \ 12 \ 0 \ 5 \ 5 \ 9 \ 8 \ 10 \ 7]$ shows the sorting

of the elements of mask. The median is given by

$$[0 \ 5 \ 5 \ 7 \ 8 \ 9 \ 9 \ 10 \ 12]$$

where ' 8 ' is the median of elements of mask.

The median filter works very well and significantly better than an averaging filter with comparable neighborhood size.

The ' max and min filters ' and ' mode filters ' are also fundamental types of non linear filter in spatial domain.

2.5.2 Transform Domain methods

Most of the research on image denoising is carried out in the transform domain especially the frequency domain [5]. Initially low pass filters in the frequency domain were used to remove the noise from the images. The commonly used low pass filters in frequency domain were Butterworth filters and Gaussian filter.

The wavelet theory , after its development , has become the most successful field for the image denoising problem.

2.5.2.1 The Butterworth Low Pass Filter

The Butterworth filter is the widely used filter in one dimensional Signal processing. However its performance is very reasonable when used in two dimensional signal Processing . It provides an alternative filtering strategy whose behavior is a function of the cutoff frequency ' D_0 ' and the order of the filter ' n '. The shape of the filter's frequency response, particularly the steepness of the transition between pass band and stop band, is also controlled by the order of the filter.

The Butterworth filter can be expressed mathematically as [2]

$$H(u, v) = \frac{1}{1 + [\frac{D(u,v)}{D_0}]^{2n}} \quad (2.5)$$

Where ' D_0 ' is the cutoff frequency and ' n ' is the order of the filter

2.5.2.2 The Gaussian Low Pass Filter

A Gaussian low-pass filter attenuates high frequencies using a transfer function whose shape is based on a Gaussian curve. The width of the bell-shaped curve can be controlled by specifying the parameter σ , which is functionally equivalent to the cutoff frequency ' D_0 ' defined previously. Lower values of σ correspond to more strict filtering, resulting in blurrier results. The smooth transition between pass band and stop band of the Gaussian LPF guarantees that there will be no noticeable ringing artifacts in the output image.

The Gaussian LPF can be mathematically described as [2]

$$H(u, v) = e^{-(D(u,v)^2)/2\sigma^2} \quad (2.6)$$

2.5.2.3 The Wavelet Transform

Since image is a two dimension data therefore the two dimensional wavelet transform is used for image denoising. Wavelet transform decomposes the image into four subbands each of different frequencies, thus enable us to find the right thresholding value to suppress the noise from the image. These four subbands are shown below in the figure 2.1

Figure 2.1 : First Level Decomposition of the image

There are numerous variants of wavelet transform. However The most investigated domain in denoising using Wavelet Transform is the non-linear coefficient thresholding based methods. The procedure in which small coefficients are removed while others are left untouched is called Hard Thresholding . The method generates spurious blips, better known as artifacts, in the images as a result of unsuccessful attempts of removing moderately large noise coefficients[6]. To overcome the demerits of hard thresholding, wavelet transform using soft thresholding was also introduced . In this scheme, coefficients above the threshold are shrunk by the absolute value of the threshold itself.

2.6 Summary of the chapter

There are large number of image denoising methods both in spatial domain and in transform domain. The selection of a particular method depends upon the model of noise corrupted the image. Since in this project we are considering only Additive Gaussian Noise therefore only those methods were described in this chapter that are most suitable for removing Additive Gaussian Noise. However all the other methods have their own significance.

Chapter 3

Transform domain image denoising methods

3.1 Introduction

Transform domain image denoising methods / filtering are the most favorite area of researchers for last two decades. There are numerous methods of designing image denoising filters in transform domain. However in this project following three methods of designing denoising filters are presented.

1. Low Pass Filters.
2. Wavelet Transform
3. Wiener Filtering

In this project report , the above mentioned three methods are explained in details and their comparison analysis is presented on the basis of PSNR and SSIM. The performance of these filters is also explained at different noise levels and at different cutoff frequencies.

The image used in this project is the standard lena image and the size of the image is 300 x 300 pixels

3.2 Low Pass Filters

In this section the performance of three different low pass filters will be presented and then comparative analysis will be presented. These filters are

1. Ideal Low Pass Filter
2. Butterworth Low Pass Filter
3. Gaussian Low pass Filter

3.2.1 Ideal Low Pass Filter

An ideal low-pass filter enhances all frequency components within a specified radius (from the center of the FT), while attenuating all others. Its mathematical formulation is given as follows [2]

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases} \quad (3.1)$$

where $D(u, v) = \sqrt{u^2 + v^2}$ represents the distance between a point of coordinates (u, v) and the origin of the 2D frequency plan, and D_0 is a nonnegative value, referred to as the cutoff frequency and cutoff radius. The 2D cutoff frequency is expressed in pixels.

The 2D view of ideal low pass filter for cutoff frequency 50 Pixels is shown below in figure 3.1

Figure 3.1 : 2D view of Ideal Low Pass Filter with cutoff frequency = 50 pixels

In this project standard gray lena image is used for all experiments. Ideal low pass filter is applied to remove the noise from image corrupted by different noise levels. in this project the noise levels are $\sigma = 20, 30, 40$ and 50 . The noise levels are measured in the same units as that of gray levels of the image.

Experiment with ideal low pass filter is performed at different cutoff frequencies. The values of cutoff frequencies are $30, 40, 50$ and 60 pixels.

The performance of ideal low pass filter at cutoff frequency 40 pixels is shown below in figure 3.2. The noise level of the image is 20

The details of the performance of ideal low pass filter at different cutoff frequencies is observed in terms of PSNR and SSIM and is shown below in table 3.1

Figure 3.2 : Image Denoising by ideal low pass filter

**(a)Original image (b) Noisy image with $\sigma=20$
(c) Denoised image with cutoff frequency = 40 pixels**

Noise Level (σ)	Cutoff Frequency	PSNR	SSIM
20	30	27.8735	0.7745
	40	28.95	0.7758
	50	29.37	0.7515
	60	29.29	0.7168
30	30	27.1617	0.7446
	40	27.64	0.7141
	50	27.3964	0.6548
	60	26.7048	0.5868
40	30	26.2589	0.7038
	40	26.1267	0.6370
	50	25.71	0.5675
	60	24.7324	0.4885
50	30	25.1832	0.6627
	40	24.8281	0.5750
	50	24.0599	0.4896
	60	23.1039	0.4095

Table 3.1 : Performance of Ideal Low Pass Filter at different cutoff frequencies and Noise levels.

The experimental results shown in Figure 3.2 and in Table 3.1 are obtained by Matlab Program # 01 present in the appendix of the project.

The results shown in table 3.1 indicates that the ideal low pass filter achieving excellent results. The PSNR is also approaching around 30 dB while SSIM approaching around 0.8. Another observation seen from table 3.1 is that if the noise level is low then higher values of cutoff frequencies will give better results.

3.2.2 Butterworth Low Pass Filter

The Butterworth filter is the widely used filter in one dimensional Signal processing. However its performance is very reasonable when used in two dimensional signal Processing . It provides an alternative filtering strategy whose behavior is a function of the cutoff frequency ' D_0 ' and the order of the filter ' n '. The shape of the filter's frequency response, particularly the steepness of the transition between pass band and stop band, is also controlled by the order of the filter.

The Butterworth filter can be expressed mathematically as [2]

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}} \quad (3.2)$$

Where ' D_0 ' is the cutoff frequency in pixels and ' n ' is the order of the filter.

In this project Butterworth low pass filters of order 6 and 8 are applied to remove the noise from image corrupted by different noise levels.

Experiments with Butterworth low pass filter is performed at different cutoff frequencies. The values of cutoff frequencies are 30, 40, 50 and 60 pixels

The performance of Butterworth low pass filter of order 6 at cutoff frequency 50 pixels is shown below in figure 3.4. The noise level on the image is 20.

The details of the performance of Butterworth low pass filter of order 6 and 8 at different cutoff frequencies are observed in terms of PSNR and SSIM and are shown below in tables 3.2 and 3.3 respectively.

Figure 3.3 : 2D view of Butterworth Low Pass Filter with cutoff frequency = 50 pixels

Figure 3.4 : Image denoising by Butterworth Lowpass filter of order 6

(a) Original image (b) Noisy image with $\sigma=20$

(c) Denoised image with cutoff frequency = 50 pixels

Noise Level (σ)	Cutoff Frequency	PSNR	SSIM
20	30	28.6393	0.8272
	40	29.8549	0.8301
	50	30.2994	0.8030
	60	30.1459	0.7656
30	30	27.8778	0.7951
	40	28.4970	0.7677
	50	28.3051	0.7100
	60	27.5901	0.6397
40	30	26.9198	0.7574
	40	26.9307	0.6598
	50	26.2975	0.6119
	60	25.4784	0.5331
50	30	25.8986	0.7199
	40	25.6319	0.6399
	50	24.9197	0.5474
	60	23.8737	0.4592

Table 3.2 : Performance of 6th order Butterworth Low Pass Filter at different cutoff frequencies and Noise levels.

Noise Level (σ)	Cutoff Frequency	PSNR	SSIM
20	30	28.49	0.8199
	40	29.6562	0.8198
	50	30.1374	.7982
	60	30.0159	0.7582
30	30	27.761	0.7887
	40	28.3304	0.7614
	50	28.0938	0.7036
	60	27.5127	0.6385
40	30	26.8509	0.7520
	40	26.7521	0.6882
	50	26.3248	0.6181
	60	25.4784	0.5404
50	30	25.8052	0.7109
	40	25.3763	0.6271
	50	24.7158	0.5380
	60	23.6813	0.4530

Table 3.3 : Performance of 8th order Butterworth Low Pass Filter at different cutoff frequencies and Noise levels.

The experimental results shown in Figure 3.4 and in Table 3.2 and Table 3.3 are obtained by Matlab Program # 02 present in the appendix of the project.

The results shown in table 3.2 and table 3.3 indicates that the Butterworth low pass filter achieving state of the art property because the PSNR is also approaching over 30 dB while SSIM approaching over 0.8.

Butterworth low pass filter achieving this property because in this filter there are two parameters which can control the performance of the filter. These parameters are the cutoff frequency and the order of the filter while in ideal low pass filter there was only one parameter .

Moreover it is also observed that if the noise level is low then higher values of cutoff frequencies will give better results.

3.2.3 Gaussian Low Pass Filter

A Gaussian low-pass filter attenuates high frequencies using a transfer function whose shape is based on a Gaussian curve. The width of the bell-shaped curve can be controlled by specifying the parameter σ , which is functionally equivalent to the cutoff frequency ' D_0 ' defined previously. Lower values of σ correspond to more strict filtering, resulting in blurrier results. The smooth transition between pass band and stop band of the Gaussian LPF guarantees that there will be no noticeable ringing artifacts in the output image.

The Gaussian LPF can be mathematically described as [2]

$$H(u, v) = e^{-(D(u, v)^2)/2\sigma^2} \quad (3.3)$$

The performance of Gaussian low pass filter at cutoff frequency 40 pixels is shown below in figure 3.6. The noise level on the image is 20.

The details of the performance of Gaussian low pass at different cutoff frequencies are observed in terms of PSNR and SSIM and are shown below in tables 3.4

Figure 3.5 : 2D view of Gaussian Low Pass Filter with cutoff frequency = 50 pixels

Figure 3.6 : Image denoising by Gaussian Lowpass filter

(a) Original image (b) Noisy image with $\sigma=20$

(c) Denoised image with cutoff frequency = 40 pixels

Noise Level (σ)	Cutoff Frequency	PSNR	SSIM
20	30	29.2888	0.8374
	40	30.2484	0.8184
	50	30.2633	0.7771
	60	29.8069	0.7237
30	30	28.1470	0.7824
	40	28.4823	0.7310
	50	27.9309	0.6599
	60	27.0860	0.5893
40	30	27.0515	0.7312
	40	26.6785	0.6442
	50	25.9592	0.5585
	60	24.8633	0.4785
50	30	25.7571	0.6697
	40	25.2146	0.5670
	50	24.1860	0.4724
	60	23.1376	0.3920

Table 3.4 : Performance of Gaussian Low Pass Filter at different cutoff frequencies and Noise levels.

The experimental results shown in Figure 3.6 and in Table 3.4 are obtained by Matlab Program # 03 present in the appendix of the project.

The results shown in table 3.4 indicates that the Gaussian low pass filter achieving the state of the art property when use for image denoising because the PSNR and SSIM approaching the highest values which are over 30dB and 0.8 respectively.

3.2.4 Conclusion about Low Pass Filters

Three different low pass filters were used in this project and the experimental results were presented. The observations obtained from the experimental results show that Gaussian low pass filter performs better than its other counterparts when compared with PSNR and SSIM values.

3.3 Wavelet Transform

Wavelets give a superior performance in image denoising due to properties such as sparsity and multi resolution structure. With Wavelet Transform gaining popularity in the last two decades various algorithms for denoising in wavelet domain were introduced. The focus was shifted from the Spatial and Fourier domain to the Wavelet transform domain. The most investigated domain in denoising using Wavelet Transform is the non-linear coefficient thresholding based methods. The procedure exploits sparsity property of the wavelet transform. The procedure in which small coefficients are removed while others are left untouched is called Hard thresholding but the method generates spurious blips, better known as artifacts, in the images as a result of unsuccessful attempts of removing moderately large noise coefficients. To overcome the demerits of hard thresholding, wavelet transform using soft thresholding was also introduced. In this scheme, coefficients above the threshold are shrunk by the absolute value of the threshold itself [5].

The plots of soft and hard thresholding functions are shown below in figure 3.7

In this project image denoising is performed using soft thresholding because the soft method is much better and yields more visually pleasant images. This is because the hard method is discontinuous and yields abrupt artifacts in the images recovered [9].

The global threshold value used in the project is calculated by using the following relation

$$T = 1.42\sigma\sqrt{2\log N} \quad (3.4)$$

where σ = standard deviation of the Noise

N = size of the image

Figure 3.7 : Plot of soft and hard thresholding functions

In this project first three decomposition levels are used for image denoising. the performance of wavelet transform using soft thresholding is observed in terms in terms of image quality factors like PSNR and SSIM.

The performance of wavelet transform for image denoising application is shown below in figure 3.8

The details of performance with respect to decomposition levels, PSNR and SSIM are shown below in table 3.5

Noise Level (σ)	Decomposition Levels	PSNR	SSIM
20	1	27.9248	0.6008
	2	29.9517	0.7837
	3	28.2463	0.7811
30	1	24.6899	0.443
	2	28.0753	0.7031
	3	27.0943	0.7510
40	1	22.3655	0.3442
	2	26.2503	0.6281
	3	26.1678	0.6281
50	1	20.554	0.7227
	2	24.9653	0.5582
	3	25.2328	0.6954

Table 3.5 : Performance of Wavelet soft thresholding for image denoising at different decomposition levels and noise levels.

Figure 3.8 : Image denoising by wavelet transform

(a) Original image (b) Noisy image with $\sigma=20$

(c) Denoised image by third level decomposition

The experimental results shown in Figure 3.8 and in Table 3.5 are obtained by Matlab Program # 04 present in the appendix of the project.

The results shown in table 3.5 indicates that the wavelet transform achieving excellent results. the PSNR is also approaching around 30 dB while SSIM approaching around 0.8. The table shows the results for three different levels of image decomposition using wavelet transform. In addition to this it is also observed that larger the value of noise on the image , highest will be the decomposition level to reduce the noise.

3.4 Wiener Filtering

Wiener filter has the ability to de-blur the image. the method adapted in this particular part of the project is to pass the noisy image through the blurring filter to get

the degraded image and then applying Wiener Filter to get the estimate of the original filter.

A motion blurring filter is implemented according to the equation [1]

$$H(u, v) = \frac{T}{\pi(ua + vb)} \sin[\pi(ua + vb)] e^{-i\pi(ua+vb)} = T * \text{sinc}(\pi(ua + vb)) * e^{-i\pi(ua+vb)}$$

where a and b indicates planar motion in x and y directions respectively; T is the duration of moving. In order to blur the image in 45 degree direction, a and b should adopt the same value. Here we set a=b=0.1. T is set to be 1 in this project. The frequency spectrum of the blurring filter is centered at coordinates (M/2, N/2). M and N are the row size and column size of the image to be processed respectively. So the actual blur filter function implemented in the project is [1]

$$H(u, v) = \text{sinc}(\pi(0.1 \left| u - \frac{M}{2} \right| + 0.1 \left| v - \frac{N}{2} \right|)) * e^{-i\pi(0.1(u-\frac{M}{2})+0.1(v-\frac{N}{2}))}.$$

The noisy image is filtered by this motion blurring function to produce the degraded image. Finally, a Wiener filter is applied to the degraded image to restore the original image.

The transfer function of the Wiener filter is given by [1]

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + \frac{S_n(u, v)}{S_f(u, v)}} \right] G(u, v), \quad (3.5)$$

where

$\hat{F}(u, v)$ = the restored image;

$H(u, v)$ = the degradation function;

$G(u, v)$ = the corrupted image;

$S_n(u, v)$ = the power spectrum of the noise;

$S_f(u, v)$ = the power spectrum of the original image.

The above method is applied to two images. They are lena images and text images.

The performance of wiener filter for image denoising application is shown below in figures 3.9 and 3.10.

The quality of the filtering is measured by PSNR and SSIM. The details of observations are shown below in table 3.6

Figure 3.9 : Image denoising by Wiener filter

(a) Original image (b) Noisy image with $\sigma=20$

(c) Degraded image by adding blur to noisy image

(d) Restored image

a

b

c

d

Figure 3.10 : Image denoising by Wiener filter

(a) Original image (b) Noisy image with $\sigma=40$

(c) Degraded image by adding blur to noisy image

(d) Restored image

Noise Level (σ)	Image	PSNR	SSIM
20	Lena	24.0948	0.7164
	Text	20.5880	0.6430
30	Lena	23.2344	0.6591
	Text	19.5203	0.6183
40	Lena	22.4913	0.6757
	Text	18.6873	0.5881
50	Lena	21.9972	0.6653
	Text	17.882	0.5827

Table 3.6 : Performance of Wiener filter for image denoising at different noise levels.

The experimental results shown in Figure 3.9 and in Table 3.6 are obtained by Matlab Program # 05 present in the appendix of the project.

The results shown in table 3.6 indicates that wiener filter is not achieving better results. The PSNR and SSIM values are low. However an important result is observed by comparing figures 3.9 and 3.10. The images in figures 3.9 and 3.10 are filtered with same Wiener filter but the visual quality of restored image in figure 3.10 is better than the image in figure 3.9. This is due to the fact that original image of figure 3.10 has more

high intensity contents than figure 3.9. Hence Wiener filter performs better for bright images.

3.5 Conclusion about transform domain techniques

In this chapter three transform domain techniques were discussed. These were low pass filtering, Wavelet transform method of image denoising and Wiener Filtering.

It is observed that the wavelet transformation is the most powerful tool of representing the noise of the image at different decomposition levels which helps to select the optimize value of the thresholding

It is concluded that the performance of Wavelet transform and Gaussian low pass filter achieving state of the art property in terms of PSNR and SSIM. However the results obtained by wavelet decomposition are of better visual quality than the other algorithms used in this chapter.

Chapter 4

Spatial domain image denoising methods

4.1 Introduction

Spatial domain methods are the conventional or classical methods of removing the noise from the images [5] . Spatial domain is the actual or original plane of the image itself. The methods used in the spatial domain are based on the direct manipulations of the pixels of the image. The methods use to denoise the image in spatial domain are called spatial filtering.

In this project the following three methods of image denoising in spatial domain are explained in detail and then their comparative analysis is carried out. In all these three methods the idea is to use the neighborhood of the pixel and then do some mathematical manipulation to remove the noise.

1. Mean Filters
2. Adaptive Median Filter
3. Mid Point Filter

4.2 Mean Filters

The mean filter which is also called as a neighborhood averaging filter is the most widely used filtering method in the spatial domain. It uses convolution with a (usually 3×3) mask whose coefficients have a value of 1 and divides the result by a

scaling factor (the total number of elements in the mask) The combination of the mask and function is called a filter [3]. A neighborhood averaging filter in which all coefficients are equal is also referred to as a box filter. The convolution mask for a (3×3) mean filter is given by

$$h(x, y) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.1)$$

The larger masks or neighborhoods can also be used but they can cause proportional blurriness in the resulting image.

In this project two variants of the mean filter are used. These are

1. Arithmetic mean filter
2. Geometric mean filter

4.2.1 Arithmetic mean Filter

The arithmetic mean filter, also known as averaging filter, operates on an $m \times n$ sliding window by calculating the average of all pixel values within the window and replacing the center pixel value in the destination image with the result. Its mathematical formulation is given as follows [2]

$$f(x, y) = \frac{1}{mn} \sum_{(r,c) \in W} g(r, c) \quad (4.2)$$

Where g is the noisy image, f is the restored image and r and c are the coordinates of rows and columns respectively, within the window W of size $m \times n$ where the operation takes place.

The performance of the arithmetic filter is shown below in table 4.1 in which different noise levels are used with different sizes of the mask. the figure of merit in this project are PSNR and SSIM.

The experimental results obtained by denoising the standard lena image with arithmetic filter is shown below in figure 4.1

Noise Level (σ)	Size of the mask	PSNR	SSIM
20	3	29.3353	0.7332
	5	28.5753	0.8093
	7	26.9727	0.8016
30	3	26.9579	0.5995
	5	27.5104	0.7440
	7	26.4406	0.7666
40	3	25.0403	0.4943
	5	26.3358	0.6694
	7	25.7940	0.7272
50	3	23.3259	0.4095
	5	25.1047	0.6060
	7	24.999	0.6860

Table 4.1 : Performance of Arithmetic filter with different noise levels and different sizes of masks

Figure 4.1: Image denoising by Arithmetic mean filter
(a) Original image (b) Noisy image with $\sigma=20$
(c) Denoised image by (3 x 3)

The experimental results shown in Figure 4.1 and in Table 4.1 are obtained by Matlab Program # 06 present in the appendix of the project

4.2.2 Geometric mean Filter

The geometric mean filter is a variation of the arithmetic mean filter and is primarily used on images with Gaussian noise. This filter is known to retain image detail better than the arithmetic mean filter. Its mathematical formulation is as follows [2]

$$f(x, y) = [\sum_{(r,c) \in W} g(r, c)]^{1/mn} \quad (4.3)$$

The performance of the Geometric filter is shown below in table 4.2 in which different noise levels are used with different sizes of the mask. the figure of merit in this project are PSNR and SSIM.

The experimental results obtained by denoising the standard lena image with arithmetic filter is shown below in figure 4.2

Noise Level (σ)	Size of the mask	PSNR	SSIM
20	3	26.1135	0.6080
	5	23.0970	0.6152
	7	20.9007	0.5893
30	3	22.7317	0.4440
	5	20.0906	0.4811
	7	18.0562	0.4687
40	3	18.9749	0.2870
	5	16.3609	0.3005
	7	14.6557	0.2686
50	3	16.3543	0.1945
	5	13.6118	0.1778
	7	11.9521	0.1466

Table 4.2 : Performance of Geometric filter with different noise levels and different sizes of masks

Figure 4.2: Image denoising by Geometric mean filter

(a) Original image (b) Noisy image with $\sigma=20$

(c) Denoised image by (3 x 3)

The experimental results shown in Figure 4.2 and in Table 4.2 are obtained by Matlab Program # 07 present in the appendix of the project.

From tables 4.1 and 4.2 it is observed that the arithmetic mean filter performs better than geometric mean filter when compared on the basis of PSNR and SSIM. It is also observed that if the noise on the images is of higher value then the filter of large size will be used to obtain better result.

4.3 Adaptive Median Filter

Median filter is a nonlinear filter and is widely used in removing the noise from the images. In median filtering the size of the structural element unit is very important and play a vital role in restoring the image of better quality.

In this project the adaptive median filter is used rather than the conventional median filter where the size of the mask or filter is fixed.

In adaptive median filtering the correct selection of the size of the mask is the optimization problem. in this method the optimization algorithm is used in such a way that the size of the mask depends upon the intensity of noise corrupted the image [6]. Greater the intensity of noise, higher will be the order or size of the mask.

Any optimization algorithm can be used to find the optimize size of the mask. However the the performance of Particle swarm optimization algorithm is better. The fitness function used for optimization is the PSNR of the images.

The results obtained by using PSO algorithm for optimizing the size of the mask or filter for different levels of noise is summarized below in the table 4.3 [6]

Noise Level	Size of mask or filter
20	3
30	5
40	5
50	7

Table 4.3 : Optimized size of mask or filter at different Noise levels

The performance of the Adaptive median filter is shown below in table 4.4 in which different noise levels are used with different sizes of the mask. The figure of merit in this project are PSNR and SSIM.

The experimental results obtained by denoising the standard lena image with adaptive median filter are shown below in figure 4.3.

Noise Level	Size of Mask	PSNR	SSIM
20	3	28.8124	0.6537
30	5	28.004	0.6817
40	5	26.2000	0.6001
50	7	25.2311	0.6223

Table 4.4 : Performance of Adaptive median filter with different noise levels and different sizes of Mask

From Table 4.4 it is observed that the adaptive median filter reduce the computational complexities as compare to the conventional median filter where the size of the mask is always find out by hit and trial method. Thus the adaprive median filter suggests the actual size of mask according the noise level present on the image.

Figure 4.3: Image denoising by adaptive median filter

(a) Original image (b) Noisy image with $\sigma=20$

(c) Denoised image by (3 x 3) Mask

The experimental results shown in Figure 4.3 and in Table 4.4 are obtained by Matlab Program # 08 present in the appendix of the project

4.4 Midpoint Filter

The midpoint filter calculates the average of the highest and lowest pixel values within a window, thereby combining order statistics and averaging into one filter. It is used to reduce Gaussian and uniform noise in images.

Mathematically [2],

$$f(x, y) = \frac{1}{2} [\max\{g(r, c) | (r, c) \in W\} + \min\{g(r, c) | (r, c) \in W\}] \quad (4.4)$$

The performance of the Midpoint filter is shown below in table 4.5 in which different noise levels are used.

The experimental results obtained by denoising the standard lena image with the Midpoint filter are shown below in figure 4.4

Noise Level	PSNR	SSIM
20	20.6486	0.6262
30	20.0606	0.5164
40	19.3173	0.4300
50	18.5479	0.3767

Table 4.5 : Performance of Midpoint filter with different noise levels

Figure 4.4: Image denoising by Midpoint filter

(a) Original image (b) Noisy image with $\sigma=20$

(c) Denoised image

The experimental results shown in Figure 4.4 and in Table 4.5 are obtained by Matlab Program # 09 present in the appendix of the project.

The results shown in table 4.5 indicates that Midpoint filter is not achieving better results. The PSNR and SSIM values are very low as compared to the mean and median filter. It is also observed that the performance of midpoint filter becomes poorer as the level of noise increases on the image .

4.5 Conclusion about Spatial domain Techniques

In this chapter three image denoising techniques were presented. It is observed that the performance of mean filters and adaptive median filter is better than midpoint filter when compared on the basis of PSNR and SSIM

However the adaptive median filter is better than the other filters because the selection of the size of mask or filter is optimized according to the noise level. Table 4.4 shows that the performance of adaptive median filter is consistent at all noise levels as compare to the performance or other filters.

Chapter 5

Conclusion and Future work

5.1 Conclusion

In this chapter six different techniques of image denoising were presented in detail. three techniques / Methods belong to transform domain and three belong to spatial domain. The comparative analysis of these methods were also presented. The figure of merit in this project were PSNR and SSIM. the quality and performance of each method is measured against these figure of merits.

The noise model used in this project was an additive Gaussian noise model. In reality it is not necessary to have the prior knowledge of the noise model. However the prior knowledge of noise model plays a significant role in selecting the suitable image denoising algorithm.

In this projected the performance of the image denoising algorithm was tested by adding an additive Gaussian noise of different noise levels to the images. These noise levels vary from low to high levels of variance. Since six different algorithms were tested in the project, the performance of these algorithms vary with different noise levels.

Noise is best described in frequency domain where all the spectral components of the noise are shown. Also the variations in the amplitudes of noise is more visible in frequency domain which helps to design the suitable denoising algorithm with optimum thresholding and coefficients.

Hence it is concluded that the performance of frequency domain techniques provide better results for image denoising applications. The performance of Gaussian

low pass filter and Wavelet transform is much more better than the other algorithms tested in this project. The denoised / restored image obtained by wavelet filtering is of better visual quality than the other filters even at the high level of noise.

Thus the transform domain techniques Provide better solution to the image denoising problem.

Furthermore it is acknowledged that all the objectives of the project stated at the beginning of the report are achieved.

5.2 Future work

In this project six fundamental image denoising methods / Algorithms were presented in detail and their comparative analysis is also carried out. Half of the methods were in spatial domain while half were in transform domain. These algorithms provide good fundamental knowledge of image denoising problem for the learners. After getting the understanding of the methods presented in this project the future work could be to explore the more advanced methods of image denoising.

Few of the advanced methods are the following

1. General Adaptive Monte Carlo Bayesian Image Denoising.
2. Stochastic image denoising based on Markov-chain Monte Carlo sampling.
3. Gaussian Scale mixture denoising.

Appendix

Program # 01

The following program is used to denoise image using ideal low pass filter at noise level=20 and cutoff frequency of 50

```
clc;
clear all;
close all;
g=imread('C:\Users\owner\Desktop\lena.jpg');
[M,N]=size(g)
m=0;
v=400;
f1=uint8(double(g)+m+randn(M,N)*sqrt(v));

%Taking the 2D FFT and shifting of noisy image

[M,N]=size(f1);
F=fft2(double(f1));
ft_shift = fftshift(F);

%Generating a transfer function for ideal low pass filter

u=0:(M-1);
v=0:(N-1);
idx=find(u>M/2);
u(idx)=u(idx)-M;
idy=find(v>N/2);
v(idy)=v(idy)-N;
[V,U]=meshgrid(v,u);
D=sqrt(U.^2+V.^2);
D0=50; %Cutoff frequency
H=double(D<=D0);
G=H.*F;

%Computing the inverse FFT and taking the real part

g1=uint8(real(ifft2(double(G))));

%Computing the Spectrum of the filtered image.

g2=fft2(double(g1));
ft_shift1 = fftshift(g2);
```

```

%Displaying the images

subplot(321);imshow(g);title('Original Image');
subplot(322);imshow(f1);title('Noisy Image with noise level 20');
subplot(323);imshow(log(1 + abs(ft_shift)), []);title('Spectrum of
noisy Image')
subplot(324);imshow(fftshift(H));title('Ideal lowpass Filter with
Cutoff frequency = 50')
subplot(325);imshow(g1);title('Filtered Image')
subplot(326);imshow(log(1 + abs(ft_shift1)), []); title('Spectrum of
Filtered Image');

A = double(g); B = double(g1);
d = sum((A(:)-B(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/d)
f=g;
f_res=g1;
mssim = SSIM(f,f_res)

```

Program # 02

The following program is used to denoise image using Butterworth low pass filter at noise level=20 and cutoff frequency of 50

```
clc;
clear all;
close all;
g=imread('C:\Users\owner\desktop\lena.jpg');
[M,N]=size(g)
m=0;
v=400;
f1=uint8(double(g)+m+randn(M,N)*sqrt(v));

%Taking the 2D FFT and shifting of noisy image

[M,N]=size(f1);
F=fft2(double(f1));
ft_shift = fftshift(F);

%Generating a real valued function and multiplied it with image

u=0:(M-1);
v=0:(N-1);
idx=find(u>M/2);
u(idx)=u(idx)-M;
idy=find(v>N/2);
v(idy)=v(idy)-N;
[V,U]=meshgrid(v,u);
D=sqrt(U.^2+V.^2);
D0=50;%Cutoff frequency
n=4;
H_but = 1 ./ (1 + (D./ D0) .^ (2 * n)); %Butterworth Filter

G=H_but.*F; %Applying Butterworth Filter to image

%Computing the inverse FFT and taking the real part

g1=uint8(real(ifft2(double(G))));

%Computing the Spectrum of the filtered image.

g2=fft2(double(g1));
```

```

ft_shift1 = fftshift(g2);

%Display all Results

subplot(321);imshow(g);title('Original Image');
subplot(322);imshow(f1);title('Noisy Image');
subplot(323);imshow(log(1 + abs(ft_shift)), []);title('Spectrum
of noisy Image')
subplot(324);imshow(fftshift(H_but));title('Butterworth lowpass
Filter')
subplot(325);imshow(g1);title('Filtered Image')
subplot(326);imshow(log(1 + abs(ft_shift1)), []);
title('Spectrum of Filtered Image');

A = double(g); B = double(g1);
d = sum((A(:)-B(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/d)
f=g;
f_res=g1;
mssim = SSIM(f,f_res)

```

Program # 03

The following program is used to denoise image using Gaussian low pass filter at noise level=20 and cutoff frequency of 40

```
clc;
clear all;
close all;
g=imread('C:\Users\owner\desktop\lena.jpg');
[M,N]=size(g)
m=0;
v=400;
f1=uint8(double(g)+m+randn(M,N)*sqrt(v));

%Taking the 2D FFT and shifting of noisy image

[M,N]=size(f1);
F=fft2(double(f1));
ft_shift = fftshift(F);

%Generating a real valued function and multiplied it with image

u=0:(M-1);
v=0:(N-1);
idx=find(u>M/2);
u(idx)=u(idx)-M;
idy=find(v>N/2);
v(idy)=v(idy)-N;
[V,U]=meshgrid(v,u);
D=sqrt(U.^2+V.^2);
sigma=40; %Standard Deviation
x=(D.^2)/(2*(sigma)^2);
H_gau = exp(-(x)); %Gaussian Filter
G=H_gau.*F; %Applying Butterworth Filter to image

%Computing the inverse FFT and taking the real part

g1=uint8(real(ifft2(double(G))));

%Computing the Spectrum of the filtered image.

g2=fft2(double(g1));
ft_shift1 = fftshift(g2);
```

```

%Display all Results

subplot(321);imshow(g);title('Original Image');
subplot(322);imshow(f1);title('Noisy Image');
subplot(323);imshow(log(1 + abs(ft_shift)), []);title('Spectrum
of noisy Image')
subplot(324);imshow(fftshift(H_gau));title('Gaussian lowpass
Filter')
subplot(325);imshow(g1);title('Filtered Image')
subplot(326);imshow(log(1 + abs(ft_shift1)), []);
title('Spectrum of Filtered Image');

A = double(g); B = double(g1);
d = sum((A(:)-B(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/d)
f=g;
f_res=g1;
mssim = SSIM(f,f_res)

```

Program # 04

The following program is used to denoise image using Wavelet decomposition at noise level=20.

```
close all;
clear all;
clc;
f=imread('C:\Users\owner\desktop\lena.jpg');
[M,N]=size(f)
m=0;
v=400;
g=uint8(double(f)+m+randn(M,N)*sqrt(v));

[c,s]=wavedec2(g,3,'db1');
[H1,V1,D1] = detcoef2('all',c,s,1);
A1 = appcoef2(c,s,'db1',1);
V1img = wcodemat(V1,255,'mat',1);
H1img = wcodemat(H1,255,'mat',1);
D1img = wcodemat(D1,255,'mat',1);
A1img = wcodemat(A1,255,'mat',1);
[H2,V2,D2] = detcoef2('all',c,s,2);
A2 = appcoef2(c,s,'db1',2);
V2img = wcodemat(V2,255,'mat',1);
H2img = wcodemat(H2,255,'mat',1);
D2img = wcodemat(D2,255,'mat',1);
A2img = wcodemat(A2,255,'mat',1);
[H3,V3,D3] = detcoef2('all',c,s,3);
A3 = appcoef2(c,s,'db1',2);
V3img = wcodemat(V3,255,'mat',1);
H3img = wcodemat(H3,255,'mat',1);
D3img = wcodemat(D3,255,'mat',1);
A3img = wcodemat(A3,255,'mat',1);
figure;
subplot(2,2,1);
imagesc(A1img);
colormap gray(255);
title('Approximation Coef. of Level 1');

subplot(2,2,2);
imagesc(H1img);
title('Horizontal detail Coef. of Level 1');

subplot(2,2,3);
imagesc(V1img);
title('Vertical detail Coef. of Level 1');
```



```

subplot(2,2,4);
imagesc(D1img);
title('Diagonal detail Coef. of Level 1');
figure;
subplot(2,2,1);
imagesc(A2img);
colormap gray(255);
title('Approximation Coef. of Level 2');
subplot(2,2,2);
imagesc(H2img);
title('Horizontal detail Coef. of Level 2');
subplot(2,2,3);
imagesc(V2img);
title('Vertical detail Coef. of Level 2');

subplot(2,2,4);
imagesc(D2img);
title('Diagonal detail Coef. of Level 2');
figure;
subplot(2,2,1);
imagesc(A3img);
colormap gray(255);
title('Approximation Coef. of Level 3');

subplot(2,2,2);
imagesc(H3img);
title('Horizontal detail Coef. of Level 3');

subplot(2,2,3);
imagesc(V3img);
title('Vertical detail Coef. of Level 3');

subplot(2,2,4);
imagesc(D3img);
title('Diagonal detail Coef. of Level 3');
[thr,soft] = ddencmp('den','wv',g)% thr is roughly equal to
sigma*sqrt(log(prod(size(g))))
F = wdencomp('gbl',g,'sym4',3,thr,soft,1);
figure;subplot(131);imshow(f);title('Original Image')
subplot(132);imshow(g);title('Noisy Image')
subplot(133);F=uint8(F);imshow(F);title('Filtered Image')
A = double(f); B = double(F);
d = sum((A(:)-B(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/d)
f_res=F;
mssim = SSIM(f,f_res)

```

Program # 05

The following program is used to denoise image using Wiener Filter at noise level=20.

```
clear all;
close all;
clc;
f=imread('C:\Users\owner\desktop\lena.jpg'); % input image
[M,N]=size(f);
m=0;var=400; % mean and variance of Gaussian noise
noise_gau=m+randn(M,N)*sqrt(var); % generate Gaussian noise
g=uint8(double(f)+noise_gau); % Noisy

% implement freq. domain blurring filter (the transform is
centered) with T=1,a=b=0.1
[M,N]=size(g);
fil_blur=zeros(M,N);
T=1;a=0.1;b=0.1; % parameters of the blurring filter
for u=1:M % row
for v=1:N % column
fil_blur(u,v)= T*sinc(a*abs(u-M/2)+b*abs(v-N/2))*exp(-
1i*pi*(a*(u-M/2)+b*(v-N/2)));
end
end

% Shifting the original image and takeing 2D-DFT

f_shift=double(g);
for i=1:M % row
for j=1:N % column
f_shift(i,j)=f_shift(i,j)*(-1)^(i+j);
end
end
F=fft2(f_shift); % transform of the original image
G=F.*(fil_blur); % blur the original image in frequency domain
%Doing the inverse FFT, shifting and taking the real part to get
the
%blurred image
g1=ifft2(G);
for i=1:M
for j=1:N
g1(i,j)=g1(i,j)*(-1)^(i+j);
end
end
```

```

% computing the transform of the degraded image
g_shift=double(g1);
for i=1:M
for j=1:N
g_shift(i,j)=g_shift(i,j)*(-1)^(i+j); %Shifting degraded image
end
end
G=fft2(g_shift); % transform of the degraded image
% computing the transform of the Gaussian noise
noise_shift=zeros(M,N);
for i=1:M
for j=1:N
noise_shift(i,j)=noise_gau(i,j)*(-1)^(i+j);%Shifting noise
end
end
N_gau=fft2(noise_shift); % transform of the Gaussian noise
% restore the image using the Wiener filter
% fil_blur is the transform of degradation function
% F is the transform of the undegraded image
% G is the transform of the degraded image
% N_gau is the transform of the Gaussian noise
F_res=G.*(abs(fil_blur).^2)./(fil_blur.*(abs(fil_blur).^2+(abs(N_gau).^2)./(abs(F).^2)));
%Doing IFFT, shifting and taking the real part to get the
restored image
f_res=ifft2(F_res);
for i=1:M
for j=1:N
f_res(i,j)=f_res(i,j)*(-1)^(i+j);
end
end
f_res=uint8(real(f_res)); % restored image

subplot(321);imshow(f);title('Original Image');
subplot(322);imshow(g);title('Noisy Image');
subplot(323);imshow(fil_blur);title('Blurring Filter')
subplot(324);imshow(uint8(g1));title('The Degraded Image by
Motion Blur and Additive Gaussian Noise');
subplot(325);imshow(f_res);title('Restored Image')

A = double(f); B = double(f_res);
d = sum((A(:)-B(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/d)
mssim = SSIM(f,f_res)

```

Program # 06

The following program is used to denoise image using Arithmetic Mean Filter at noise level=20 and mask size=3.

```
close all;
clear all;
clc;
f=imread('C:\Users\owner\desktop\lena.jpg');
%f=rgb2gray(f);
[M,N]=size(f);
m=0;
v=400;
g=uint8(double(f)+m+randn(M,N)*sqrt(v));
    h = fspecial('average',3);
    img =imfilter(g,h);
    subplot(131)
    imshow(f)
    title('Original Image');
    subplot(132)
    imshow(g)
    title('Gaussian Noise');
    subplot(133)
    imshow(img)
    title('Filtered Image');
    A = double(f); B = double(img);
d = sum((A(:)-B(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/d)
f_res=img;
mssim = SSIM(f,f_res)
```

Program # 07

The following program is used to denoise image using Geometric Mean Filter at noise level=20 and mask size=3.

```
close all;
clear all;
clc;
f=imread('C:\Users\owner\desktop\lena.jpg');
%f=rgb2gray(f);
[M,N]=size(f);
m=0;
v=400;
g=uint8(double(f)+m+randn(M,N)*sqrt(v));
m=3;
n=3;
img = exp(imfilter(log(double(g)), ones(m, n), 'replicate')) .^
(1/(m*n));
img=uint8(img);
subplot(131)
    imshow(f)
    title('Original Image');
subplot(132)
    imshow(g)
    title('Gaussian');
subplot(133)
    imshow(img)
    title('Filtered Image');
A = double(f); B = double(img);
d = sum((A(:)-B(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/d)
```

Program # 08

The following program is used to denoise image using Median Filter at noise level=20 and mask size=3.

```
close all;
clear all;
clc;

f=imread('C:\Users\owner\desktop\lena.jpg');
%f=rgb2gray(f);
[M,N]=size(f);
m=0;
v=400;
g=uint8(double(f)+m+randn(M,N)*sqrt(v));

    img=medfilt2(g, [3 3]);

    subplot(131);imshow(f);xlabel('a');

    subplot(132);imshow(g);xlabel('b');
    subplot(133);imshow(img);xlabel('c');
    A = double(f); B = double(img);
d = sum((A(:)-B(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/d)
f_res=img;
mssim = SSIM(f,f_res)
```

Program # 09

The following program is used to denoise image using Mid Point Filter at noise level=20 and mask size=3.

```
close all;
clear all;
clc;
f=imread('C:\Users\owner\desktop\lena.jpg');
%f=rgb2gray(f);
[M,N]=size(f);
m=0;
v=400;
g=uint8(double(f)+m+randn(M,N)*sqrt(v));
midpoint = inline('0.5 * (max(x(:)) + min(x(:)))');
img = nlfilter(g,[3 3],midpoint);

subplot(131);imshow(f);xlabel('a');

subplot(132);imshow(g);xlabel('b');
subplot(133);imshow(uint8(img));xlabel('c');
A = double(f); B =double(img);
d = sum((A(:)-B(:)).^2) / prod(size(A));
psnr = 10*log10(255*255/d)
f_res=img;
mssim = SSIM(f,f_res)
```

References

- [1] Rafael. C .Gonzalez and Richard E Woods, " Digital Image Processing ", 3rd Edition, Pearson International Edition prepared by pearson education.
- [2] Oge Marques, " Practical Image and Video Processing using Matlab ",published by John Wiley and sons in 2011, ISBN: 9781118093481.
- [3] Alasdair McAndrew, "An introduction to digital image processing with Matlab", Notes for SCM2511 Image Processing 1, published in 2004
- [4] Alexander Wong, Akshaya Mishra, Wen Zhang, Paul Fieguth, David A. Clausi "Stochastic image denoising based on Markov-chain Monte Carlo sampling", Elsevier Journal of signal processing 91 (2011) 2112-2120, published in 2011.
- [5] Mukesh C Motwani, Mukeash C Gadiya, Federick C Haris, " Survey of Image Denoising Techniques ".
- [6] Zhu youlian, Huang cheng, "Image Denoising Algorithm Based on PSO Optimizing Structuring Element ", 24th Chinese Control and Decision Conference (CCDC) 2012
- [7] Rajni and Anutam , "Image denoising techniques-An overview " International Journal of Computer Applications (0975 – 8887) Volume 86 – No 16, January 2014.
- [8] Manonmani S, Lalitha V.P, Dr. Shanta Rangaswamy, " Survey On Image Denoising Techniques ", International Journal of Science,Engineering and Technology Research (IJSETR) Volume 5, Issue 9, September 2016
- [9] Prerna Chimania and Prof. Dr.Vineet Richariya, " A Literature Survey on Image Denoising Techniques , International Journal of Emerging Technology and Advanced Engineering, Volume 5, Issue 4, April 2015.