Designing a Framework for Noise dependent Filter Selection Algorithm

A Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

In

Electronic Instrumentation and Control

Submitted by

SAPNA BAKSHI

Regn. No.-801051019

Under the Guidance of

Ms. Ruchika Lamba
Lecturer
EIED

Mr. M.D Singh
Assistant Professor
EIED

Department of Electrical and Instrumentation Engineering

Thapar University

(Established under the section 3 of UGC act, 1956)

Patiala, 147004, Punjab, India

July 2012

CERTIFICATE

This is to certify that my work presented in this thesis entitled **"Designing a Framework for Noise dependent Filter Selection Algorithm"** in partial fulfillment of the requirements for the award of degree of master of engineering in **Electronic Instrumentation and control Engineering** at Thapar University, Patiala is an original record under supervision and guidance of **Mrs. Ruchika Lamba** and **Mr. M D Singh**. The matter embodied in this report has not been submitted anywhere for the award of any other degree of this or any other university.

Sapna

Sapna Bakshi

Roll No. 801051019

Date 13.7.12

It is certified that the above statement made by the student is true to best of our knowledge.

Ruchika

Mrs. Ruchika Lamba

Lecturer, EIED

Supervisor

Thapar University, Patiala

Mr. M D Singh

Assistant Professor, EIED

Supervisor

Thapar University, Patiala

S. Ghosh

Dr Smarajit Ghosh

Professor and Head, EIED

Thapar University, Patiala

Dr. S. K. Mohapatra

Dean of Academic Affairs

Thapar University, Patiala

ACKNOWLEDGEMENT

I am highly grateful to **Dr. Smarajit Ghosh**, Head, Department of Electrical & Instrumentation Engineering, Thapar University, Patiala (Formerly known as Thapar Institute of Engineering and Technology, Patiala), for providing this opportunity to carry out the present work.

I would like to express a deep sense of gratitude and thanks profusely to my supervisors, **Ms. Ruchika Lamba,** lecturer and **Mr. M.D. Singh,** Assistant Professor, Electrical and Instrumentation Engineering Department, Thapar University, Patiala. Without their wise counsel and able guidance, it would have been impossible to complete the present work.

I also express my gratitude to other faculty members of the department for their intellectual support throughout the course of this work.

The copious help received from the technical staff of the department for the excellent laboratory support is also acknowledged.

Finally, I am indebted to all whosoever have contributed to provide help to carry out the present work.

Sapna Bakshi

ABSTRACT

This work provides a framework to design an automatic filtering algorithm, which will work according to the noise present in the image. For this purpose, can be used about the working of the non linear geometric, diffusion, enhanced Frost, Lee, Homogeneous mass area, Wiener and Median filter for different of images. This comparative study is based on the output of the noisy and the filtered images using these filters. Three types of Noises: Gaussian, Poisson and Speckle are used to produces Noisy images from noise free image for the purpose of evaluation. Parameters are characterized on the basis of some standard image quality matrices like SNR, correlation coefficient, mean square error and peak signal to noise ratio. The performance of these filters is evaluated by comparative study of values of these parameters of noisy, filtered and original images. Based upon this analysis, results are compiled. These statistical metrics are also displayed graphically and they are compared for both the noisy and the filtered images.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Overview

Most of the imaging systems have a common problem of 'Noise'. This noise may be inherent in the system like ultrasound or it may due to external sources, artifacts etc. Unwanted data which may reduce the contrast deteriorating the shape or size of objects in the image and blurring of edges or dilution of fine details in the image may be termed as noise. It may be due to one or more of the following reasons

- Physical nature of the system
- Shortcomings of image acquisition devices
- Image developing mechanism
- Due to environment.

Mathematically there are two basic models of Noise; additive and multiplicative. Additive noise is systematic in nature and can be easily modeled and hence removed or reduced easily. Most additive noise filtering approaches utilize the fast Fourier transform, convolution, or recursive algorithms. In the transform and convolution methods, the autocorrelation between pixels is either assumed or approximated. Whereas multiplicative noise is image dependent, complex to model and hence difficult to reduce.

Due to these reasons there is a requirement of filters. In this thesis work filters such as non linear geometric, diffusion, enhanced frost, enhanced lee, homogeneous mass area, wiener and median filters have been used to remove the speckle, Gaussian and Poisson noise at different noise density. Some standard images are used here different noises are added and then filter is applied and evaluation is done by quality metrics. Parameters are characterized on the basis of some standard image quality matrices like SNR, correlation coefficient, mean square error and peak signal to noise ratio. The performance of these filter is evaluated by comparative study of values of these parameters of noisy, filtered and

original images. On basis of these parameters values best filter out of these used filters is analysed. This methodology suggest us the automatic filtering technique.

1.2 Organization of Thesis

This thesis is organized as per the following format.

Chapter-1 **"Introduction"** it includes the overview of this thesis report.

Chapter 2 **"Review of Literature"** involves the description of the entire literature survey.

Chapter 3 **"Background"** gives a complete idea of the background if image and different types and parameters of image.

Chapter 4 **"Methodology"** contains the proposed filters which are being used to remove the noise in image.

Chapter 5 **"Results and Discussions"** contains the results obtained (in the form of both tables and graphs) for both the noised and the filtered images.

Chapter 6 **"Conclusion"** concludes this thesis work. This chapter gives brief outline of the work.

CHAPTER 2
REVIEW OF LITERATURE

Yongjian Yu *et.al*.[1] provides the derivation of speckle reducing anisotropic diffusion by showing that Lee and Frost filters can be cast as partial differential equations, and then SRAD is derived by allowing edge-sensitive anisotropic diffusion. As the conventional anisotropic diffusion is the edge-sensitive diffusion for images corrupted with additive noise, SRAD is edge sensitive diffusion for speckled noise. Lee and Frost filters utilize the coefficient of variation in adaptive filtering, SRAD exploits the instantaneous coefficient of variation. SRAD is a diffusion method tailored to ultrasonic and radar imaging applications.

Kiyo Tomiyasu [2] proposed a computer program to simulate radar return from a single pixel containing numerous scatters. The phase and amplitude of the signal reflected from the pixel generally varies from pulse to pulse due to intra pixel interference. After integration, the pixel response usually differs with each "azimuth look," and this speckle variation between azimuth looks may be as much as 20 db or more. Speckle variation can be significantly reduced by averaging or median filtering multiple azimuth looks per pixel.

Armand Lopes *et.al*.[3] developed many adaptive filters for speckle reduction. Filters analyzed were based on a test related to the local coefficient of variation of the observed image, which describes the scene heterogeneity. In this paper, the most well-known adaptive filters for SAR images are analyzed. It is shown that they are only reliable in a bounded field. Their behavior is then modified by introducing two thresholds on the coefficient of variation. The lower one is easily established as a function of the image parameters. The second upper threshold is fixed approximately, but a study should be done to determine it more precisely. When tested on real or simulated SAR images, the proposed modified filters adequately average homogeneous areas and better preserve, at the same time, edges and textural information.

Mustafa Karaman *et.al.[*4] proposed an adaptive smoothing technique for speckle suppression in medical B-scan ultrasonic imaging. The technique is based on filtering with appropriately shaped and sized local kernels. For each image pixel, a filtering kernel, which fits to the local homogeneous region containing the processed pixel, is obtained through a local statistics based region growing technique. Performance of the proposed filter has been tested on the phantom and tissue images. The results show that the filter effectively reduces the speckle while preserving the resolvable details.
An adaptive smoothening speckle reduction technique involves 3 steps:

- Computation of local statistics
- Region growing procedure
- Application of smoothening operator

Richard N. Czerwinska *et.al*.[5] presented a novel adaptation of the median filter to the problem of boundary-preserving speckle reduction in ultrasonic imaging. The technique involves applying a bank of oriented one-dimensional median filters to the image, and retaining at each point the largest value among all the filter bank outputs. The result is an operator which suppresses speckle noise while retaining the structure of the image, particularly the thin bright streaks, which tend to occur along boundaries between tissue layers. The technique is compared to a block median filter and an algorithm discussed by Loupas *et al.,* and is shown to be far superior to the median filter, and noticeably better than the Loupas filter at enhancing thin lines.

Chedsada Chinrungrueng *et.al.*[6] developed an edge preserving noise reduction filter called as Savitzky-Golay filter. This paper describes a novel filter which is a two-dimensional extension of the one-dimensional Savitzky-Golay filter. This filter is based on the least squares polynomial surface fitting to image intensities. The performance of the proposed filter has been compared with that of the commonly used median filter in reducing speckle noise in ultrasound images. Experimental results indicate that the new filter can achieve, at least, the same level of noise reduction and edge preservation as that of the median filter, but with far less computation time. Since its complexity scales linearly with the problem size, the new filter is suitable for filtering problems with large windows.

In addition, it also proved to be less sensitive to the size of the filtering window compared to the median filter.

Yongjian Yu *et. al.* [7] generalized the 2D SRAD algorithm developed for despeckling ultrasound images to obtain a SRAD algorithm capable of enhancing volumetric ultrasound data. First, an instantaneous coefficient of variation edge detector appropriate for three dimensional set is presented. Then, the 3D SRAD partial differentiation equation is formulated by replacing the gradient operator in the traditional diffusion mechanism with the derived 3D ICOV operator. Performance comparison between the 3D SRAD and 2D SRAD has been made using synthetic 3D data.

Khalifa Djemal [8] proposed speckle reduction in ultrasound images by minimization of total variation. To limit the noise in an image, some techniques are based on the calculation of an average intensity in each pixel of the image by considering a some neighbor. However, these techniques tend to attenuate contours present in the image. This affects edge and particularly penalize for the segmentation algorithms whose finality is to find contours. The restoration method by minimization of total variation consists of minimizing under constraints of the great variations present in the image while preserving contours. Results of the work show that speckle is removed by this method, discontinuities are practically preserved and the regions of restored image are more homogeneous.

G. Deng *et.al*.[9] proposed an adaptive Gaussian filter for noise reduction and edge detection. The author proposed an adaptive Gaussian filtering algorithm in which the filter variance is adapted to both the noise characteristics and the local variance of the signal. The use of Gaussian filter as preprocessing for edge detection also gave rise to edge position displacement, edges vanishing, and phantom edges. Using Gaussian filter for noise suppression, the noise is smoothed out, at the same time the signal is also distorted.
Results show that

- the window size of (5x5) is suitable for calculating the local signal variance,
- the minimum filter variance is always located at the edge point, thus the adaptive algorithm causes less distortion to the edges,

- the image processed by adaptive Gaussian filter always has smaller mean square error (MSE) than the non-adaptive Gaussian filter, and
- The edges extracted from the image processed by the adaptive algorithm are better than that from the non-adaptive algorithm, especially at locations where two edges cross.

James C. Brailean *et.al*.[10] presented a thorough review of noise reduction filters for digital image sequences. Detailed descriptions of several spatiotemporal and temporal noise reduction algorithms were provided.

Four major categories of filters were identified:

- Non motion compensated spatiotemporal
- motion compensated spatiotemporal
- Non motion compensated temporal
- Motion compensated temporal filtering.

For each category several algorithms have been compared, pointing out the similarities and differences in each approach. It is clear from results that although many image sequence filtering approaches exist, there is no filter that can be singled out as the solution for all applications. That is, for applications constrained by computational complexity and memory availability, such as in real time high SNR applications, a non motion compensated temporal filter would be the most attractive. Depending on the amount of memory available to the filter, a non motion compensated spatiotemporal filter may also be appropriate. However, for a low SNR application that does not have these constraints, a motion compensated adaptable filter would provide the highest quality (in an MSE sense) filtered sequence.

Eero P. Simoncelli *et.al*.[11] proposed noise removal via Bayesian wavelet coring. The classical solution to the noise removal problem is the Wiener filter, which utilizes the second-order statistics of the Fourier decomposition. In this, the author developed a Bayesian estimator that is a natural extension of the Wiener solution, and that exploits these higher-order statistics. The estimator is based on two factors - a sub band representation and a statistical model - both of which can be generalized. The resulting

nonlinear estimator performs a “coring” operation. The authors provided a simple model for the sub band statistics, and use it to develop a semi-blind noise-removal algorithm based on a steerable wavelet pyramid. This approach also generalized to other types of distortion, including blurring and corruption with non-additive noise.

Dimitri Van De Ville *et.al*.[12] proposed a new fuzzy filter for the noise reduction of images corrupted with additive noise. The filter consists of two stages. The first stage computes a fuzzy derivative for eight different directions. The second stage uses these fuzzy derivatives to perform fuzzy smoothing by weighting the contributions of neighboring pixel values. Both stages are based on fuzzy rules which make use of membership functions. The filter can be applied iteratively to effectively reduce heavy noise. In particular, the shape of the membership functions is adapted according to the remaining noise level after each iteration, making use of the distribution of the homogeneity in the image. A statistical model for the noise distribution can be incorporated to relate the homogeneity to the adaptation scheme of the membership functions. l Results obtained by the author shows the feasibility of this proposed approach.

M. Nachtegael *et.al*.[13] proposed fuzzy filters for noise reduction in the case of Gaussian noise. The reduction of noise in an image sometimes is as a goal itself, and sometimes is considered as a pre-processing step. In this paper, various fuzzy logic based filters have been used which can reduce the MSE values with factors between 5 and 12. This is quite good, but especially for higher levels of gaussian noises not enough to produce really good images.

Stefan Schulte *et al*.[14] described a new algorithm that is especially developed for reducing all kinds of impulse noise fuzzy impulse noise detection and reduction method (FIDRM). It can also be applied to images having a mixture of impulse noise and other types of noise. The result is an image quasi without impulse noise so that other filters can be used afterwards. This nonlinear filtering technique contains two separated steps: an impulse noise detection step and a reduction

step that preserves edge sharpness. Based on the concept of fuzzy gradient values, this detection method constructs a fuzzy set impulse noise. Fuzzy set is represented by a membership function that will be used by the filtering method, which is a fuzzy averaging of neighboring pixels. FIDRM provides a significant improvement on other existing filters. FIDRM is not only very fast, but also very effective for reducing little as well as very high impulse noise.

Triet Le *et al*.[15] proposed a variation model to de-noise an image corrupted by Poisson noise. This new model used total-variation regularization, which preserves edges. This approach uses a data-fidelity term that is suitable for Poisson noise. The result is that the strength of the regularization is signal dependent, precisely like Poisson noise. Noise of varying scales can be removed by this model, while preserving low-contrast features in regions of low intensity. If the
image also contains higher intensity regions, the regularization will be stronger there and still remove the noise.

Michael R. Keenan *et.al.[*16] presented a simple method for weighting the data to account for Poisson noise. Using a simple time-of-flight secondary ion mass spectrometry, it was demonstrated that PCA, when applied to the weighted data, leads to results that are more interpretable, provide greater noise rejection and are more robust than standard PCA. The weighting presented is also shown to be an optimal approach to scaling data as a pretreatment prior to multivariate statistical analysis.

Piotr S. Windyga [17] proposed a filter that removes fast impulsive noise. This recursive nonlinear filter is composed of two conditional rules, which are applied independently, in any order, one after the other. It identifies noisy items by inspection of their surrounding neighborhood, and afterwards it replaces their values with the most "conservative" ones out of their neighbors values. In this way, no new values are introduced and the histogram distribution range is conserved. A new filter was designed to be much faster than the median filter while performing comparably in terms of both image information conservation and noise reduction, which suggests that it could replace the median filter for

the preliminary processing included in state-of-the-art noise removal filters. This new filter may eliminate or attenuate most noisy pixels in synthetic and natural images not excessively contaminated.

Krisana Chinnasarn *et.al*.[18] proposed a novel method based on the kFill algorithm that can be accomplished in single-pass scan over the image. The algorithm is capable to remove simultaneously both salt and pepper noise of any sizes that are smaller than the size of document objects. Document containing text and graphics components are usually acquired as binary images for computer processing purposes. Salt-and-pepper noise is a prevalent artifact in such images. Removing this noise usually requires iterative or multiple-pass processing, some techniques even cause distortions in document components. This algorithm is fast and effective. It can remove noise of different size and shape while maintaining the sharpness of the text and graphics components.

Kenny Kal Vin Toh *et.al*.[19] presented a new fuzzy switching median (FSM) filter employing fuzzy techniques in image processing. The proposed filter is able to remove salt-and pepper noise in digital images while preserving image details and textures very well. This is a recursive fuzzy switching median filter which is an extension to the classical switching median filter by employing fuzzy inference mechanism. By incorporating fuzzy reasoning in correcting the detected noisy pixel, the low complexity FSM filter is able to outperform some well known existing salt-and pepper noise fuzzy and classical filters. Unlike some filtering mechanisms which require iterations, and thus consumed lengthy processing time, the FSM filter only need to be applied once and is very efficient with its computational time.

Kenny Kal Vin Toh. *et.al*[20] proposed a novel two-stage noise adaptive fuzzy switching median (NAFSM) filter for salt-and-pepper noise detection and removal. Initially, the detection stage utilizes the histogram of the corrupted image to identify noise pixels. These detected “noise pixels” will then be subjected to the second stage of the filtering action, while “noise-free pixels” are retained and left unprocessed. Then, the NAFSM filtering mechanism employs fuzzy reasoning to handle uncertainty present in the extracted local

information that is introduced by noise. The proposed filter is able to suppress high-density of salt-and-pepper noise, at the same time preserving fine image details, edges and textures well. In addition, it does not require any further tuning or training of parameters once optimized. By carefully considering the tradeoff between the complexity of the filtering algorithm and the performance of the filter, the proposed NAFSM filter is able to yield good filtering results with efficient processing time.

Changhong Wu *et. al*.[21] proposed a new approach which is based on integration of anisotropic filter and directional median filter . In this approach, the image is first convolved with anisotropic filter and then is filtered by DMF. This model can effectively reduce Gaussian distributed noises and impulse noises along the direction of ridge flow. Advantage of using DMF is it can join broken fingerprint ridges, fill out the holes and smooth irregular ridges.

Lingfeng Meng *et. al*.[22] presented a new method of decimation-free directional filter banks (DFB)-based analysis for fingerprint enhancement. Its analysis outputs are shift-invariant directional images but not directional sub-band images in previous DFBs. The algorithm first decomposes a fingerprint image into eight directional images in the analysis stage, then obtains the orientation image based on directional energy distributions for each block and processes the directional images based on the orientation image in the processing stage, at last reconstructs them to the enhanced image in the synthesis stage. The influence of noise is attenuated and the injured ridges are recovered because the component corresponding to ridge direction in a block was accentuated, while the others were weakened by a lower weighted value. So far, speckle noise reduction [1-8] methods have been used and their application is found in ultrasonic and radar imaging applications. Many adaptive smoothing techniques have been used for speckle suppression. For Gaussian [9-16] noise reduction many adaptive filtering algorithm in which the filter variance is adapted to both the noise characteristics and the local variance of the signal. Many fuzzy filters have also been used and they can be applied to images having a mixture of impulse noise and other types of noise. For Poisson noise, model such as total- variation

regularization have been used. Still some scope is there to filter noise by using other adaptive algorithm.

CHAPTER 3
BACKGROUND

3.1 IMAGE BASICS

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows which is shown in fig 3.1. In a (8-bit) grayscale image, each picture element has an assigned intensity that ranges from 0 to 255 as shown in fig 3.2. A grayscale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey. A normal grayscale image has 8 bit color depth equal to 256 grayscales. A "true color" image has 24 bit color depth which is equal to 8 x 8 x 8 bits equal to 256 x 256 x 256 colors approximately equal to 16 million colors. Some grayscale images have more grayscales, for instance 16 bit is 65536 grayscales [23].

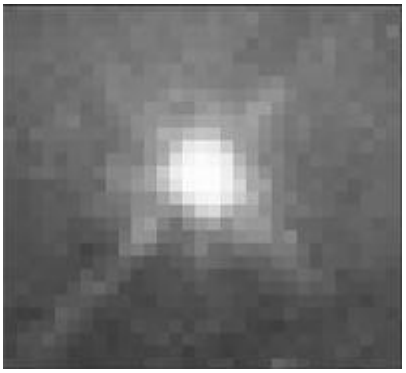


Fig 3.1 An image is an array or a matrix of pixels arranged in columns and rows.

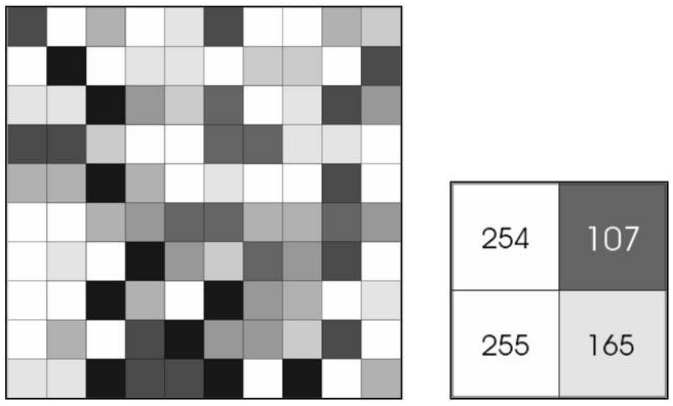


Fig 3.2 Each pixel has a value from 0 (black) to 255 (white).

3.2 DIGITAL IMAGE PROCESSING

An image may be defined as a two-dimensional function f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of are all finite, discrete quantities, we call the image a digital image as shown in fig 3.3(b). The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels, and pixels. Pixel is the term most widely used to denote the elements of a digital image [24].

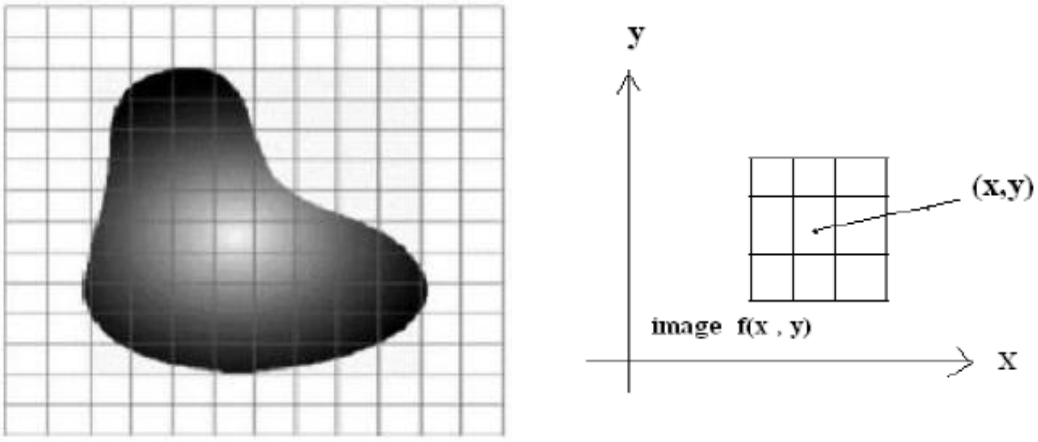


Fig 3.3 (a) Original Image (b) Digital Image

Image processing generally involves following steps:-

1. Import an image with an optical scanner or directly through digital photography.
2. Manipulate or analyze the image in some way. This stage can include:-
    - Image segmentation
    - Image data compression
    - Image restoration

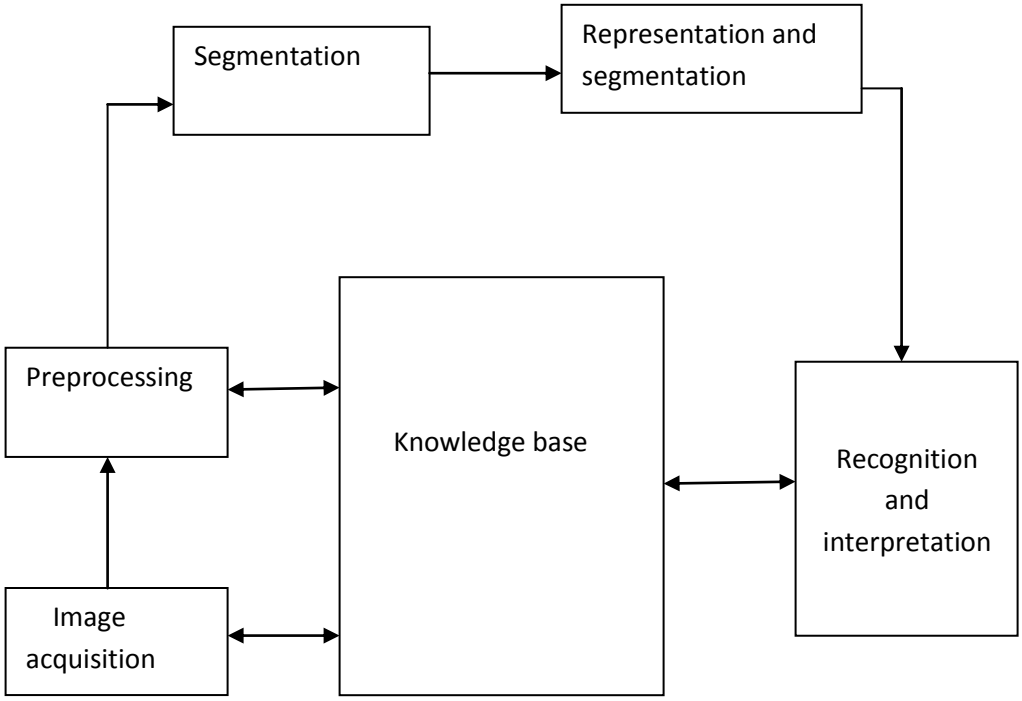


Fig 3.4 Fundamental steps in Digital Image

Image Segmentation:

The process of partitioning a digital image into multiple regions (set of pixel) is called image segmentation. Segmentation of an image entails the division or separation of the image into regions of similar attribute.

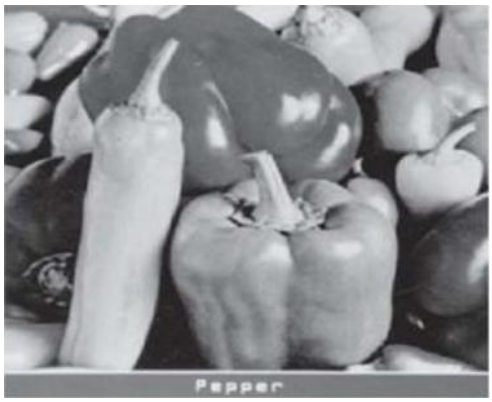
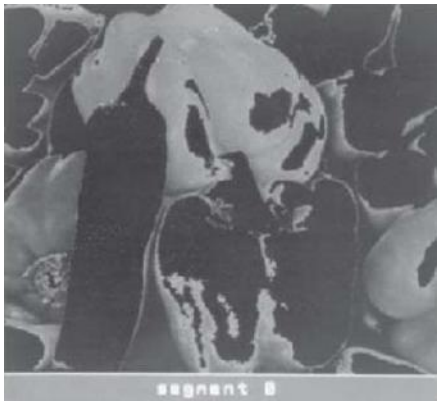


Fig 3.5 (a) Original Image



(b) Segmentation image

Image Data Compression:

Image data compression techniques are concerned with reduction of the number of bits required to store or transmit images without any appreciable loss of information.

Fig 3.6 (a) Original Image (b) Compressed image

Image Restoration:

Image restoration refers to removal or minimization of degradations in an image. Image restoration differs from image enhancement. Restoration techniques often depend only on the class or ensemble properties of a data set where as image enhancement techniques are much more image dependent.

3. Output the result. The result might be the image altered in some way or it might be a report based on analysis of the image.

3.3 CLASSIFICATION OF IMAGES

An image can be classified into following categories.

3.3.1 Intensity Images

An intensity image [25] is a data matrix whose values have been scaled to represent intensities. When the elements of an intensity image are of class unit 8, or class unit 16, they have integer values in the range [0, 255] and [0, 65535].respectively. If the image is of

class double, the values are floating-point numbers. Values of scaled, class double intensity images are in the range [0, 1].

3.3.2 Indexed Images

Array of class logical, unit 8, Unit 16, single, or double whose pixel values are directed indices into a color map. The color map is an m-by-3 array of class double. For single or double arrays, integer values range from [1, p]. For logical, unit8, or unit 16 arrays, values range from [0, p-1]. An indexed image consists of an array and a color map matrix. The pixel values in the array are directed indices into a color map.

3.3.3 Binary Image

Binary images have a very specific meaning in MATLAB. In a binary image, each pixel assumes one of only two discrete values: 1 or 0, interpreted as black and white, respectively. A binary image is stored as a logical array. Fig 3.7 shows the binary image.

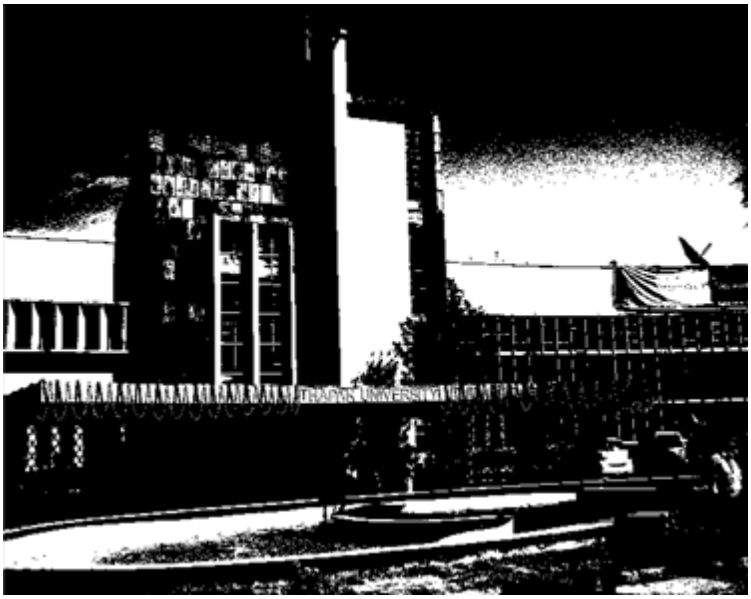


Fig 3.7 Binary image

3.3.4 Grayscale Images

A grayscale image (also called gray-scale, gray scale, or gray-level) is a data matrix whose values represent intensities within some range. MATLAB stores a grayscale image as an individual matrix, with each element of the matrix corresponding to one image pixel. By

convention, this documentation uses the variable name I to refer to grayscale images. Array of class unit8, unit16, int16, single, or double whose pixel values. For single or double arrays, values range from [0, 1]. For unit8, values range from [0, 255]. For unit16, values range from [0, 65535]. For int16, values range from [-32768, 32767]. Fig 3.8 shows the grayscale image.

Fig 3.8 Grayscale image

3.3.5 True Color Images

A true color image is an image in which each pixel is specified by three values one each for the red, blue, and green components of the pixel‘s color. MATLAB store true color images as an m-by-n-by-3 data array that defines red, green, and blue color components for each individual pixel. True color images do not use a color map. The color of each pixel is determined by the combination of the red, green, and blue intensities stored in each color plane at the pixel‘s location. Graphics file formats store true color images as 24-bit images, where the red, green, and blue components are 8 bits each. This yields a potential of 16 million colors. The precision with which a real-life image can be replicated has led to the commonly used term true color image. Fig 3.9 shows the true color image.

Fig 3.9 Color image

3.4 IMAGE NOISE

Image noise [26] is the random variation of brightness or color information in images produced by the sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is generally regarded as an undesirable by-product of image capture. Although these unwanted fluctuations are referred to as "noise" by analogy with unwanted sound, they are inaudible and actually beneficial in some applications, such as dithering**.**

Image noise is the term applied to pictures, a counterpart to the white noise, we would hear in an audio or video file. Analogue cameras will show image noise through grainy specs on the picture whereas digital cameras will show image noise through random speckles throughout the picture. While most of the time image noise should be avoided, sometimes it can create the illusion of an older picture. Things such as exposure, temperature and different camera modes can affect image noise. No matter what we do to prevent image noise, some will always be present. Any electronic unit that sends or receives a signal will be susceptible to it. For digital cameras, light that enters the lens and misaligns with the sensors will cause image noise. Even if we cannot see the noise when you look at a picture, there is some form of image noise in any image we take. The same thing can be said for audio and video productions. Every type of electronic device receives some form of noise and sends it on to what it is creating.

3.5 TYPES OF NOISES IN IMAGE

The different types of noises are discussed in following section.

3.5.1 Amplifier noise (Gaussian noise)

In communications, a random interference is generated by the movement of electricity in the line. It is similar to white noise, but confined to a narrower range of frequencies. We can actually see and hear Gaussian noise when we tune our TV to a channel that is not operating. A random distribution of artifacts in analog video images that makes everything look soft and slightly blurry. On close inspection, one can see tiny specks in random patterns. Found on films shot with older cameras as well as films and videotapes that have been archived for a long time, dynamic noise reduction (DNR) circuits can eliminate much of the Gaussian noise [27] [9] when the analog material is converted to digital. Gaussian noise is noise that has a probability density function (abbreviated pdf) of the normal distribution (also known as Gaussian distribution). In other words, the values that the noise can take on are Gaussian distributed. It is most commonly used as additive white noise to yield additive white Gaussian noise (AWGN). Gaussian noise is properly defined as noise with a Gaussian amplitude distribution. This says nothing of the correlation of the noise in time or of the spectral density of the noise. Labeling Gaussian noise as 'white' describes the correlation of the noise. Fig 3.10 shows the grayscale images.

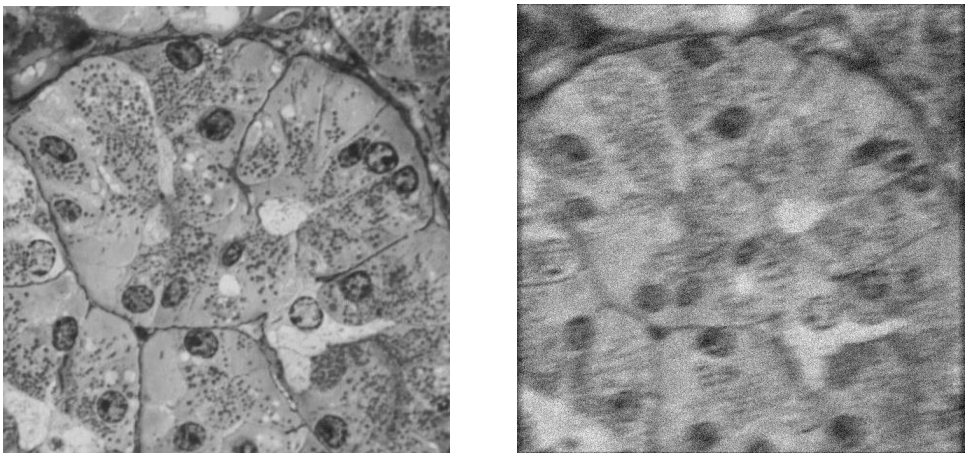


Fig 3.10 Grayscale image and the degraded image with Gaussian noise

3.5.2 Salt-and-Pepper Noise

Salt and pepper noise [19] is a form of noise typically seen on images. It represents itself as randomly occurring white and black pixels. Usual and effective noise reduction method for this type of noise involves the usage of median filter. Fig 3.11 shows image deteriorated by salt and pepper noise.

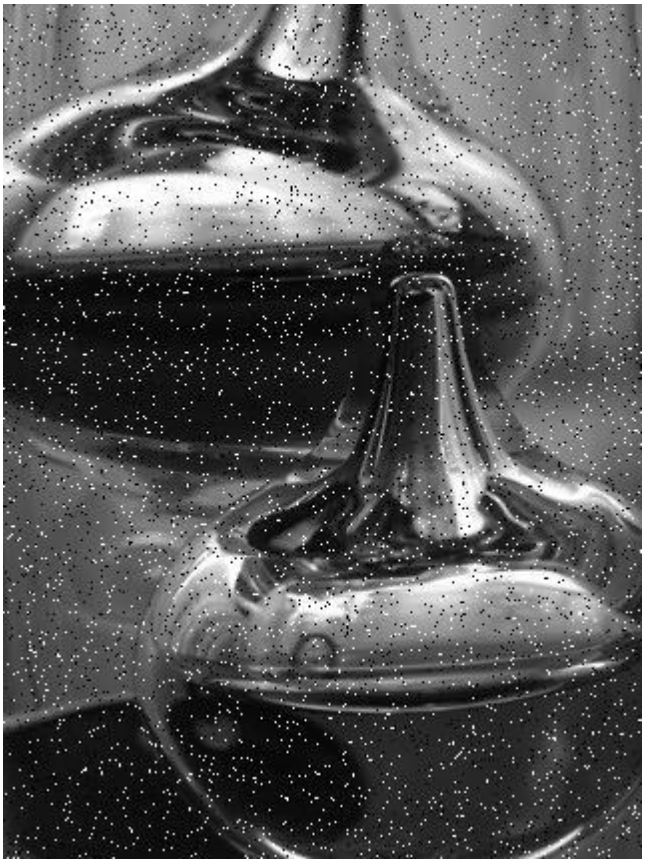


Fig 3.11 Salt and pepper noise

3.5.3 Poisson noise

The dominant noise in the lighter parts of an image from an image sensor is typically that caused by statistical quantum fluctuations, that is, variation in the number of photons sensed at a given exposure level; this noise is known as photon shot noise. Shot noise [16] has a root mean- square value proportional to the square root of the image intensity, and the noises at different pixels are independent of one another. Shot noise follows a Poisson

distribution, which is usually not very different from Gaussian. In addition to photon shot noise, there can be additional shot noise from the dark leakage current in the image sensor; this noise is sometimes known as "dark shot noise" or "dark current shot noise". Dark current is greatest at "hot pixels" within the image sensor; the variable dark charge of normal and hot pixels can be subtracted off (using "dark frame subtraction"), leaving only the shot noise, or random component, of the leakage; if dark frame subtraction is not done, or if the exposure time is long enough that the hot pixel charge exceeds the linear charge capacity, the noise will be more than just shot noise, and hot pixels appear as salt-and-pepper noise. Fig 3.12 shows the image degraded by poison noise

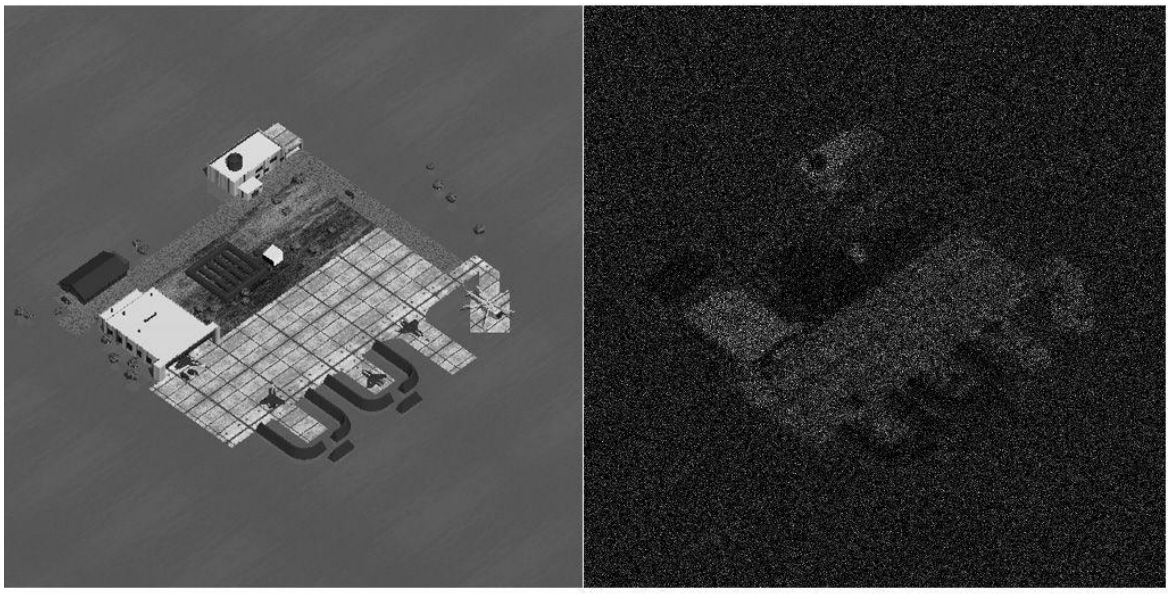


Fig 3.12 (a) Original image (b) image degraded by poisson noise

3.4.5 Speckle Noise

Speckle noise [28] is a granular noise that inherently exists in and degrades the quality of the active radar and synthetic aperture radar (SAR) images. Speckle noise in conventional radar results from random fluctuations in the return signal from an object that is no bigger than a single image-processing element. It increases the mean grey level of a local area. Speckle noise [2] in SAR is generally more serious, causing difficulties for image interpretation. It is caused by coherent processing of backscattered signals from multiple distributed targets. In SAR oceanography, for example, speckle noise [3] is caused by

signals from elementary scatterers, the gravity-capillary ripples, and manifests as a pedestal image, beneath the image of the sea waves. Speckle noise in SAR is a multiplicative noise, i.e. it is in direct proportion to the local grey level in any area. Speckle noise is also known as modal noise which is the noise generated in an optical fiber system by the combination of mode-dependent optical losses and fluctuation in the distribution of optical energy among the guided modes or in the relative phases of the guided modes.

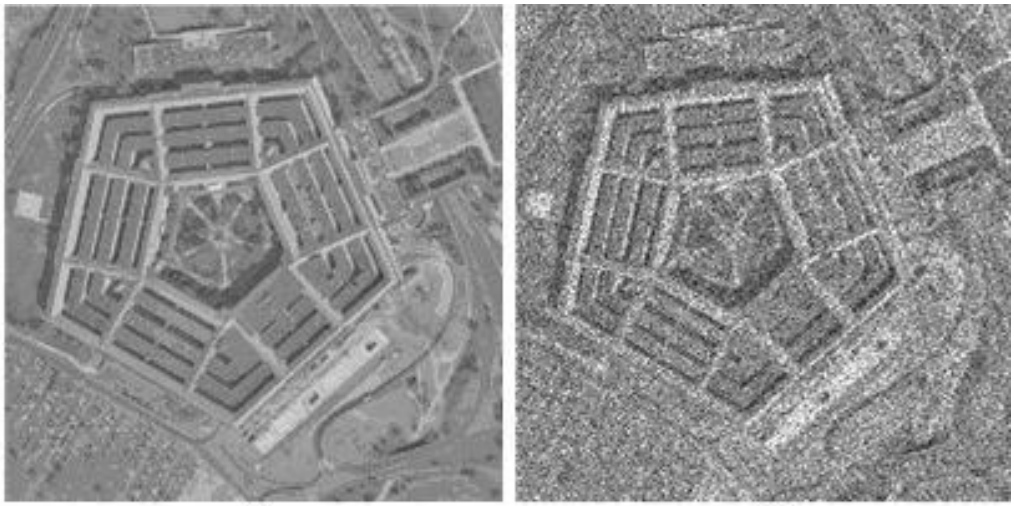


Fig 3.13 (a) Original image (b) image degraded by speckle noise

3.5.5 Non-Isotropic Noise

Some noise sources show up with a significant orientation in images. For example, image sensors are sometimes subject to row noise or column noise. In film, scratches are an example of non-isotropic noise.

CHAPTER 4
METHODOLOGY

4.1 QUALITATIVE ATTRIBUTES OF AN IMAGE

1) **PSNR** (peak signal-to-noise ratio):- Ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation [26].

- Measure of quality of reconstruction
- Signal in this case is the original data, and the noise is the error.

2) **MSE (Mean Square Error)**: The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error.

$$\text{PSNR} = 20\log_{10}\left(\frac{255}{\sqrt{\text{mse}}}\right) \quad (4.1)$$

A lower value for MSE means lesser error, and as seen from the inverse relation between the MSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction. So, having a lower MSE (and a high PSNR), it is a better one**.**

3) **RMSE (Root-Mean-Square Error):** To get a measure of how similar two images are, you can calculate the root-mean-square (RMS) value of the difference between the images. If the images are exactly identical, this value is zero. The following function uses the difference function, and then calculates the RMS value from the histogram of the resulting image.

4) **LMSE:** Producing the least mean squares of the error signal (difference between the desired and the actual signal). "Least mean square" means that we perform the following steps:-

- Calculate the difference between the data value and the model prediction at several different places (this is called the error).
- Square the error to make all values positive (square).

- Calculate the average (mean square).
- find the model alternative that gives the smallest error (least mean square)

5) **Normalized Cross Correlation: -** The Cross-Correlation function can be described as

$$cross\ corr(s,t) = \sum_x \sum_y R(x,y) I(x-s, y-t) \quad (4.2)$$

where, I-input image intensity

R-reference image intensity

And the summation is taken over the region (s,t) where R and I overlap. For any value of (s,t) inside R(x, y) the cross-correlation function yields one value of Cross Corr. The maximum value of Cross Corr (s, t) indicates the position where I(x, y) best matches R(x, y).

$$NCC = \frac{1}{var(Y)} \sum_b \frac{numY(b)}{num\ T\ of\ Y} var\ Y(b) \quad (4.3)$$

The correlation ratio ranges from 0 for very good image to 1 for very bad image. Based Upon these parameters, image can be judged either it's a good or bad image.

4.2 Need for filtering

Image filtering is useful for many applications, including smoothing, sharpening, removing noise, and edge detection. A filter is defined by a kernel, which is a small array applied to each pixel and its neighbors within an image. In most applications, the center of the kernel is aligned with the current pixel, and is a square with an odd number (3, 5, 7, etc.) of elements in each dimension. The process used to apply filters to an image is known as convolution, and may be applied in either the spatial or frequency domain.

Within the spatial domain, the first part of the convolution process multiplies the elements of the kernel by the matching pixel values when the kernel is centered over a pixel. The elements of the resulting array (which is the same size as the kernel) are averaged, and the original pixel value is replaced with this result. The CONVOL function performs this

convolution process for an entire image. Within the frequency domain, convolution can be performed by multiplying the FFT (Fast Fourier Transform) of the image by the FFT of the kernel, and then transforming back into the spatial domain. The kernel is padded with zero values to enlarge it to the same size as the image before the forward FFT is applied. These types of filters are usually specified within the frequency domain and do not need to be transformed.

4.3 Filters Used

Following filters are used for the removal of speckle, Gaussian and poison noise.

4.3.1 Non Linear Geometric Filter

Geometric filtering [29] is done on speckle which appears in the image as narrow walls and valleys. The geometric filter, through iterative repetition, gradually tears down the narrow walls (bright edges) and fills up the narrow valleys (dark edges), thus smearing the weak edges that need to be preserved. The geometric filter investigated in this study uses a nonlinear noise reduction technique. It compares the intensity of the central pixel in a 3 × 3 neighborhood with those of its eight neighbors and, based upon the neighborhood pixel intensities, it increments or decrements the intensity of the central pixel such that it becomes more representative of its surroundings. The operation of the geometric filter may be described with Fig. 4.1 and has the following form:

1. Select direction and assign pixel values

Select the direction be NS and the corresponding three consecutive pixels be *a*, *b*, *c* (see Fig. 4.1a, b).

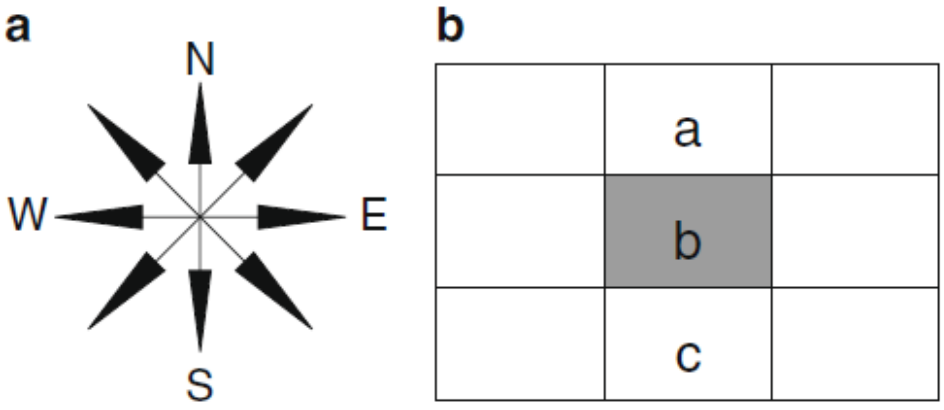


Fig. 4.1 (a) Directions of implementation of the geometric filter, (b) pixels selected for the NS direction (intensity of central pixel b is adjusted based on the values of intensities of pixels a, b, and c).

2. Carry out central pixel adjustments

Do the following intensity adjustments (see Fig. 4.1b)

if $a \geq b + 2$ then $b = b + 1$,

if $a > b$ and $b \leq c$ then $b = b + 1$,

if $c > b$ and $b \leq a$ then $b = b + 1$,

if $c \leq b + 2$ then $b = b + 1$,

if $a \leq b - 2$ then $b = b - 1$,

if $a < b$ and $b \geq c$ then $b = b - 1$,

if $c < b$ and $b \geq a$ then $b = b - 1$

if $c \leq b - 2$ then $b = b - 1$.

3. Repeat steps 1 and 2 for directions west-east (WE), west-north to south-east (WN-SE), and north-east to west-south (NE to WS) (see Fig. 4.1a).

4.3.2 Diffusion Filter

Diffusion filter [30] removes the noise from an image by modifying the image via solving a partial differential equation (PDE). Smoothing is carried out depending on the image edges and their directions. Anisotropic diffusion is an efficient nonlinear technique for simultaneously performing contrast enhancement and noise reduction. It smoothes homogeneous image regions, but retains image edges without requiring any information

from the image power spectrum. It may, thus, directly be applied to images. Anisotropic diffusion can be used to remove noise from digital images without blurring edges. With a constant diffusion coefficient, the anisotropic diffusion equations reduce to the heat equation which is equivalent to Gaussian blurring. This is ideal for removing noise but also indiscriminately blurs edges too. Along the same lines as noise removal, anisotropic diffusion can be used in edge detection algorithms. By running the diffusion with an edge seeking diffusion coefficient for a certain number of iterations, the image can be evolved towards a piecewise constant image with the boundaries between the constant components being detected as edges.

4.3.3 Frost Filter

An adaptive filter that corrects for multiplicative noise. It estimates the unspeckled pixel value using a sub window of the processing window. The frost filter is a local statistic filter. It attempts to model the SAR imaging process. The frost filter [29] parameters are adjusted according to local (quiet small) area statistic about the target pixel. When uniform regions are filtered the frost filter acts as a mean filter. When high contrast regions are filtered, the filter acts as a high pass filter with rapid decay of elements away from the filter center. Thus, large uniform areas will tend to smooth out and speckle noise is removed, whilst high contrast edges and other objects will retain their signal values and not be smoothed. The replacing value can be calculated for the frost filter.

$$x = \frac{y_1 m_1 + y_2 m_2 + \cdots y_n m_n}{m_1 + m_2 + \cdots m_n} \tag{4.4}$$

$$m = e^{-A*T} \tag{4.5}$$

$$A = damp * c_i^2 \tag{4.6}$$

Where damp is the factor that determines the extent of exponential damping for the image
T= absolute value of the pixel distance from the centre pixel to its neighbors
y = original pixel value
m= exponential weighing factor

Frost filter uses an adaptive filtering algorithm which is an exponentially damped convolution kernel that adapts itself to features by using local statistics. The adaptive filter computes a set of weight values for each pixel within the filter window surrounding each pixel. The frost filter differs from the lee filter by the fact that the image reflectivity is estimated by convolving the observed image by the impulse response of the SAR system. The impulse response of the SAR system is obtained by minimizing the mean square error between the observed image and the image reflectivity model, which is assumed to be an autoregressive process. The MASK parameter specifies the area within the input channel which filtered, the rest of image is unchanged.

4.3.4 Enhanced Frost Filter

The Enhanced frost filter [29] is used primarily to filter speckled radar data. It is designed to smooth out noise while retaining edges or shape features in the image. The filter size can be specified through the filter size parameters. Different filter sizes will greatly affect the quality of processed images. if the filter is too small, the noise filter algorithm is not effective. If the filter is too large, subtle details of the image will be lost in the filtering process. A 7 x7 filter usually gives the best results. The number looks parameter is used to estimate noise variance and it effectively controls the amount of smoothing applied to the image by the filter. Theoretically, the correct value for number looks should be the effective number of looks of the radar image. It should be closed to actual number of looks, but may be different if the image has undergone resampling. The user may experimentally adjust the number looks value so as to control the effect of the filter. A small number looks value leads to more smoothing. A larger number looks value preserves more image features. A damping factor (DAMP) is required by the enhanced frost filter. The value of DAMP defines the extent of exponential damping (smaller the value, the smaller the damping effect). It depends on non filtered image and may require trial and error experiments to determine the best value. The default value for DAMP is 1. Enhanced frost filter performs spatial filtering on each individual pixel in an image using the gray values in a square window surrounding each pixel. The dimension of the filter must be odd, and must be at least 3x3. All pixels are filtered. In order to filter pixels located near the edges of the image, edge pixel values are replicated to give sufficient data. A bit map specifies

the area within the layer which will be produced. Only this area will be affected filtered and rest of the image will be unchanged. If no bit map is connected, the entire database is processed. The enhanced frost filter model requires that the signal represents power. If the input image is an amplitude format, each gray level will be squared to derive power and finally square root will be applied to the filtered result.

$$x = \frac{y_1 m_1 + y_2 m_2 + \cdots y_n m_n}{m_1 + m_2 + \cdots m_n} \quad (4.7)$$

$$m = e^{-damp\left(\frac{c_i - c_u}{c_{max} - c_i}\right) * T} \quad (4.8)$$

where L= number of looks $c_u = (1 \div L)^{1/2}$

$$c_{i=} \frac{std\ deviation}{mean} \quad (4.9)$$

$$c_{max} = \sqrt{\left(1 + \frac{2}{l}\right)} \quad (4.10)$$

4.3.4.1 Algorithm

1. Centre the filter window on the target pixel.
2. Calculate x using formula

$$x = \frac{y_1 m_1 + y_2 m_2 + \cdots y_n m_n}{m_1 + m_2 + \cdots m_n} \quad (4.11)$$

$$m = e^{-damp\left(\frac{c_i - c_u}{c_{max} - c_i}\right) * T} \quad (4.12)$$

where L= number of looks $c_u = (1 \div L)^{1/2}$

$$c_{i=} \frac{std\ deviation}{mean} \quad (4.13)$$

$$c_{max} = \sqrt{\left(1 + \frac{2}{l}\right)} \quad (4.14)$$

3. Replace the value of the target pixel with the value of x.
4. Go to the first step until the whole image is processed.

4.3.5 Enhanced Lee Filter

The Enhanced lee filter [29] is used primarily to filter speckled radar data. It is designed to smooth out noise while retaining edges or shape features in the image. The filter size can be specified through the filter size parameters. Different filter sizes will greatly affect the quality of processed images. If the filter is too small, the noise filter algorithm is not effective. If the filter is too large, subtle details of the image will be lost in the filtering process. A 7 x7 filter usually gives the best results. The number looks parameter is used to estimate noise variance and it effectively controls the amount of smoothing applied to the image by the filter. Theoretically, the correct value for number looks should be the effective number of looks of the radar image. It should be closed to actual number of looks, but may be different if the image has undergone resampling. The user may experimentally adjust the number looks value so as to control the effect of the filter. A small number looks value leads to more smoothing. A larger number looks value preserves more image features. A damping factor (DAMP) is required by the enhanced lee filter. The value of DAMP defines the extent of exponential damping (smaller the value, the smaller the damping effect). It depends on non filtered image and may require trial and error experiments to determine the best value. The default value for DAMP is 1. Enhanced lee filter performs spatial filtering on each individual pixel in an image using the gray values in a square window surrounding each pixel. The dimension of the filter must be odd, and must be atleast 3x3. The resulting value of the x is as follows:

For

$$c_i \leq c_{max} \tag{4.15}$$

$$x = y \tag{4.16}$$

$$c_u < c_i < c_{\max} \tag{4.17}$$

$$x = y * w + y * (1 - w) \tag{4.18}$$

$$c_i \geq c_{\max} \tag{4.19}$$

$$x = y \tag{4.20}$$

Where

$$w = e^{-damp\ factor(\frac{c_i - c_u}{c_{max} - c_t})} \tag{4.21}$$

$$c_u = (1 \div L)^{1/2} \quad (4.22)$$

$$c_i = \frac{std\ deviation}{mean} \quad (4.23)$$

$$c_{max} = \sqrt{\left(1 + \frac{2}{l}\right)} \quad (4.24)$$

L= number of looks

4.3.5.1 Algorithm

1. Read image 'I' and append it by one row and column on each size.

2. Take 3x3 kernel (mask) of the appended image and go to step 3.

3. Calculate the following for centre pixel locally:

$$c_{max} = \sqrt{\left(1 + \frac{2}{l}\right)} \quad (4.25)$$

$$c_u = (1 \div L)^{1/2} \quad (4.26)$$

L= number of looks

4. The resulting gray value of x:

$$c_i \leq c_{max} \quad (4.27)$$

$$x = y \quad (4.28)$$

$$c_u < c_i < c_{max} \quad (4.29)$$

$$x = y * w + y * (1 - w) \quad (4.30)$$

$$c_i \geq c_{\max} \quad (4.31)$$

$$x = y$$

5. After calculation of each pixel value go to step 2.

4.3.6 Homogeneous Mass Area Filter

The homogeneous mass area filter is a 2D filter operating in a [5x5] pixel neighborhood by searching for the most homogeneous neighborhood area around each pixel using [3x3] subset window. The middle pixel of the [5x5] neighborhood is substituted with the average gray level of the [3x3] mask with the smallest speckle index C, which for log compressed images is given by

$$C=\sigma^2 / g$$

Where σ^2 and g represent the variance and the mean of the [3x3] window, respectively. The window with the smallest c is the most homogeneous semi window, which presumably does not contain any edge. The filter is iteratively applied until the gray levels of almost all pixels in the image do not change.

4.3.6.1 Algorithm

1. Load the image of filtering.
2. Specify the region of interest to be filtered, moving window size, the number of iterations (n), and the edge detector to be used.
3. Starting from the left upper corner of the image, rotate a mask around the middle pixel of the window for each moving window.
4. Detect the position of the mask for which c is minimum.
5. Assign the average gray level of the mask at the selected position to the middle pixel in the 5x5 window.
6. Repeat steps 4-5 for all pixels in the image by sliding the moving window from left to right.
7. Repeat steps 3- 6 for a second iteration of despeckle filtering.
8. Compute the image quality evaluation metrics and the texture features foe the original and despeckled images.
9. Display the original and despeckled images, the image quality and evaluation metrics, and the texture features.

4.3.7 Wiener Filter

- Wiener filters [11] are a class of optimum linear filters which involve linear estimation of a desired signal sequence from another related sequence.
- In the statistical approach to the solution of the linear filtering problem, we assume the availability of certain statistical parameters (e.g. mean and correlation functions) of the useful signal and unwanted additive noise. The problem is to design a linear filter with the noisy data as input and the requirement of minimizing the effect of the noise at the filter output according to some statistical criterion.

• A useful approach to this filter-optimization problem is to minimize the mean-square value of the error signal that is defined as the difference between some desired response and the actual filter output. For stationary inputs, the resulting solution is commonly known as the Weiner filter.

The input to the Wiener filter is assumed to be a signal, *s(t),* corrupted by additive noise,*n(t).* The output, *ŝ (t),* is calculated by means of a filter, *g(t),* using the following convolution:

$$\hat{s}(t) = g(t) * [\hat{s}(t) + n(t)] \tag{4.32}$$

where

- *ŝ (t)*is the original signal (not exactly known; to be estimated)
- *n(t)*is the noise
- *s(t)*is the estimated signal (the intention is to equal *s(t+α)*)
- *g(t)*is the Wiener filter's impulse response

The error is defined as

$$e(t) = \hat{s}(t+\alpha) - \hat{s}(t) \tag{4.33}$$

where α is the delay of the Wiener filter (since it is causal)

In other words, the error is the difference between the estimated signal and the true signal shifted by α.

The squared error is

$$e^2(t) = s^2(t+\alpha) - 2s(t+\alpha)\hat{s}(t) + \hat{s}^2(t) \tag{4.34}$$

where

- *s(t+α)* is the desired output of the filter
- *e(t)* is the error

Depending on the value α, the problem can be described as follows:

- If $\alpha>0$ then the problem is that of prediction (error is reduced when $\hat{s}(t)$ is similar to a later value of s)
- If $\alpha=0$ then the problem is that of filtering (error is reduced when $\hat{s}(t)$is similar to $s(t)$)
- If $\alpha<0$ then the problem is that of smoothing (error is reduced when $\hat{s}(t)$ is similar to an earlier value of s)

Writing $\hat{s}(t)$as a convolution integral:

$$\hat{s}(t)=\int_{-\infty}^{\infty} g(\tau)[s(t-\tau)+n(t-\tau)]\,d\tau \qquad (4.35)$$

Taking

$$E(e^2)=R_s(0) - 2\int_{-\infty}^{\infty} g(\tau)R_{xs}(\tau+\alpha)\,d\tau + \iint_{-\infty,-\infty}^{\infty,\infty} g(\tau)g(\theta)R_x(\tau-\theta)\,d\tau d\theta \quad (4.36)$$

the expected value of the squared error results in

where

- $x(t)=s(t)+n(t)$ is the observed signal
- R_s is the autocorrelation function of $s(t)$
- R_x is the autocorrelation function of $x(t)$
- R_{xs} is the cross-correlation function of $x(t)$ and $s(t)$

If the signal $s(t)$ and the noise $n(t)$ are uncorrelated (i.e., the cross-correlation R_{sn} is zero), then this means that

- $R_{xs}=R_s$
- $R_x=R_s+R_n$

For many applications, the assumption of uncorrelated signal and noise is reasonable.

The goal is to minimize $E(e^2)$, the expected value of the squared error, by finding the optimal $g(\tau)$, the Wiener filter impulse response function. The minimum may be found by calculating the first order incremental change in the least square resulting from an incremental change in g(.) for positive time. This is

$$\delta E(e^2) = -2 \int_{-\infty}^{\infty} \delta g(\tau)(R_{xs}(\tau+\alpha) - \int_0^{\infty} g(\theta)\, R_x(\tau-\theta)d\theta)\, d\tau \qquad (4.37)$$

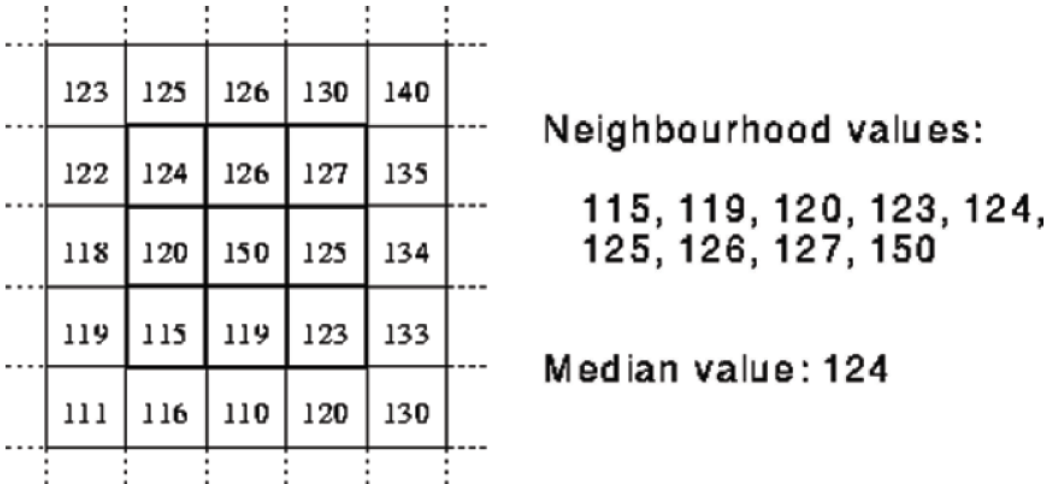
For a minimum, this must vanish identically for all $\delta g(\tau), \tau > 0$ which leads to the Wiener-Hopf equation

$$R_{xs}(\tau+\alpha) = \int_0^{\infty} g(\theta)\; R_x(\tau-\theta)d\theta \qquad (4.38)$$

This is the fundamental equation of the Wiener theory.

4.3.8 Median Filter

- In median filtering [31], the neighboring pixels are ranked according to brightness (intensity) and the median value becomes the new value for the central pixel.
- Median filters can do an excellent job of rejecting noise in which some individual pixels have extreme values.
- In the median filtering operation, the pixel values in the neighborhood window are ranked according to intensity, and the middle value (the median) becomes the output value for the pixel under evaluation.



- In particular, compared to the smoothing filters examined thus far, median filters offer three advantages:

- No reduction in contrast across steps, since output values available consist only of those present in the neighborhood (no averages).

- Median filtering does not shift boundaries, as can happen with conventional smoothing filters (a contrast dependent problem).

- Since the median is less sensitive than the mean to extreme values (outliers), those extreme values are more effectively removed.

- The median is, in a sense, a more robust "average" than the mean, as it is not affected by outliers (extreme values).

- Since the output pixel value is one of the neighboring values, new "unrealistic" values are not created near edges.

- Since edges are minimally degraded, median filters can be applied repeatedly, if necessary.

This filter first sorts the surrounding pixels values in the window to an orderly set and replaces the center pixel within the define window with the middle value in the set. Median filtering is a non-linear filtering technique that works best with impulse noise whilst retaining sharp edges in the image. The main disadvantage of the Median filter is the extra computation time needed to sort the intensity value of each set.

CHAPTER 5
RESULTS AND CONCLUSIONS

Various filters are analyzed to find their performance with different types of noise and noise levels. Results for standard images when speckle, Gaussian and Poisson noise is there, are presented in this chapter. Different filters such as non linear geometric filter (gmf), diffusion filter, enhanced frost filter, enhanced lee filter, homogeneous mass filter (hmf), wiener filter and median filter are used to remove these noises. The performance of these filters can be seen through different parameters such as correlation coefficient, laplacian, mean square error, peak signal to noise ratio. Value of noise density is also changed. On the basis of these values we came to know one best filter for the removal of particular type of noise.

5.1 Speckle noise

(1) For image moon.tiff

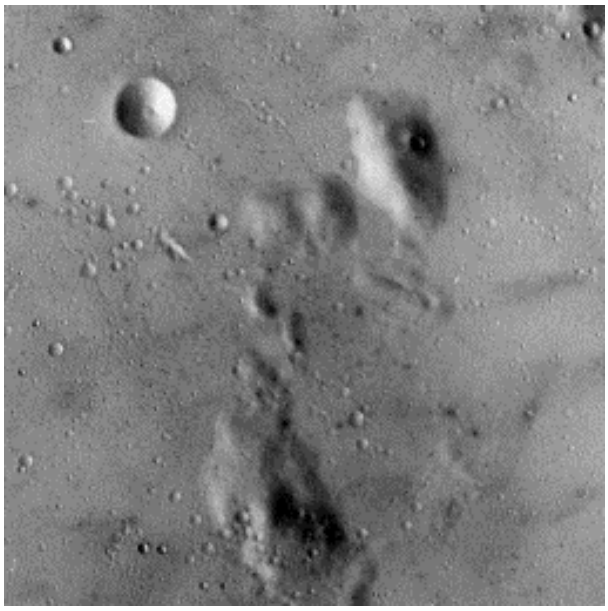


Fig 5.1 moon.tiff

1(a) For Noise Density = 0.5

Filtering Results

Table 5.1 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr1	snr_mse
noisy	0.771556	0.270221	98.05338	4.137499	28.21618	4.137499
gmf	0.894624	0.265536	3.917236	18.12233	42.20101	18.12233
diffusion	0.780188	0.283536	96.67891	4.198807	28.27749	4.198807
frost	1.016294	-0.36724	3.968811	18.06552	42.1442	18.06552
enlee	0.866426	0.168968	30.72217	9.177606	33.25629	9.177606
hmf	0.887497	0.341489	73.41098	5.394514	29.47319	5.394514
wiener	0.875684	0.303824	72.20877	5.466225	29.5449	5.466225
median	0.846561	0.258397	175.687	1.604728	25.68341	1.604728

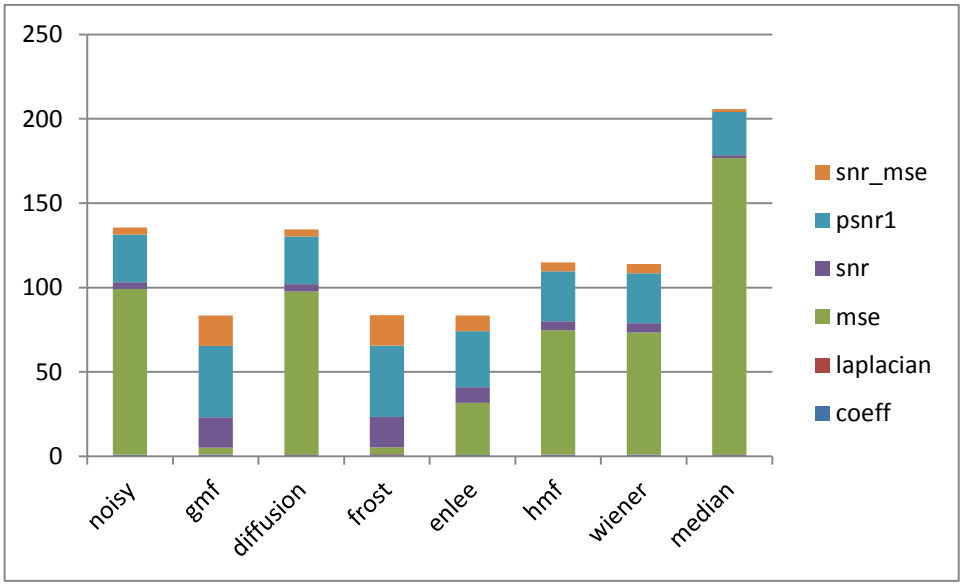


Fig 5.2 Performance of various filters on moon.tiff at noise density 0.5

As shown in fig 5.2 frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

1(b) for noise density = 0.7

Filtering Results

Table 5.2 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr1	snr_mse
noisy	0.738659	0.23096	148.5962	2.332047	26.41073	2.332047
gmf	0.882105	0.234668	17.6806	11.57715	35.65583	11.57715
diffusion	0.748614	0.237646	149.9021	2.294048	26.37273	2.294048
frost	1.016294	-0.36724	3.968811	18.06552	42.1442	18.06552
enlee	0.842099	0.142739	126.6459	3.026214	27.10489	3.026214
hmf	0.862402	0.30214	188.7067	1.294251	25.37293	1.294251
wiener	0.924247	0.346685	199.7698	1.046826	25.12551	1.046826
median	0.840691	0.257515	178.9804	1.52407	25.60275	1.52407

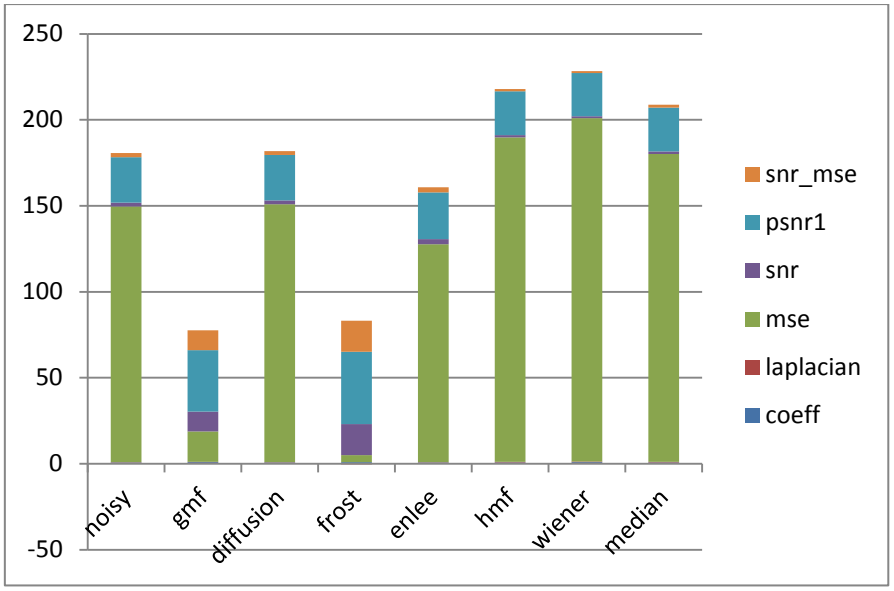


Fig 5.3 Performance of various filters on moon.tiff at noise density 0.7

As shown in fig 5.3 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

1(c) for noise density = 0.85

Filtering Results

Table 5.3 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.720084	0.198822	186.9619	1.334594	25.41327	1.334594
gmf	0.868732	0.200243	53.59312	6.761034	30.83971	6.761034
diffusion	0.724933	0.20463	188.7428	1.293421	25.3721	1.293421
frost	1.016326	-0.36593	3.968872	18.06545	42.14413	18.06545
enlee	0.811983	0.124916	185.8589	1.360291	25.43897	1.360291
hmf	0.839341	0.271384	236.6134	0.311732	24.39041	0.311732
wiener	0.905998	0.291786	244.0238	0.177803	24.25648	0.177803
median	0.80832	0.228766	230.6744	0.42213	24.50081	0.42213

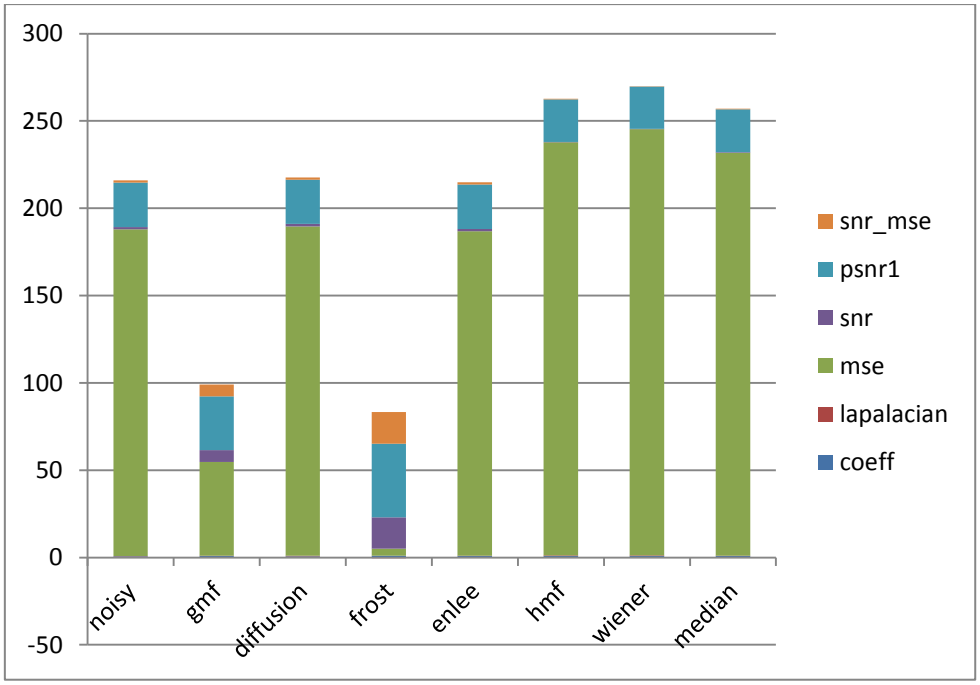


Fig 5.4 Performance of various filters on moon.tiff at noise density 0.85

As shown in fig 5.4 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

(2) For image2.tiff

Fig 5.5 image2.tiff

2(a) For Noise Density = 0.5

Filtering Results

Table 5.4 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.821393	0.190301	92.08558	4.296873	28.48889	4.296873
gmf	0.924167	0.192069	2.762875	19.52518	43.71719	19.52518
diffusion	0.825029	0.19719	91.27067	4.335477	28.52749	4.335477
frost	1.197087	-0.16817	1.488819	22.21037	46.40238	22.21037
enlee	0.92994	0.170422	25.1687	9.930182	34.1222	9.930182
hmf	0.94157	0.301972	69.87742	5.495421	29.68744	5.495421
wiener	0.975063	0.281173	38.27335	8.109825	32.30184	8.109825
median	0.926072	0.212757	64.73363	5.827489	30.0195	5.827489

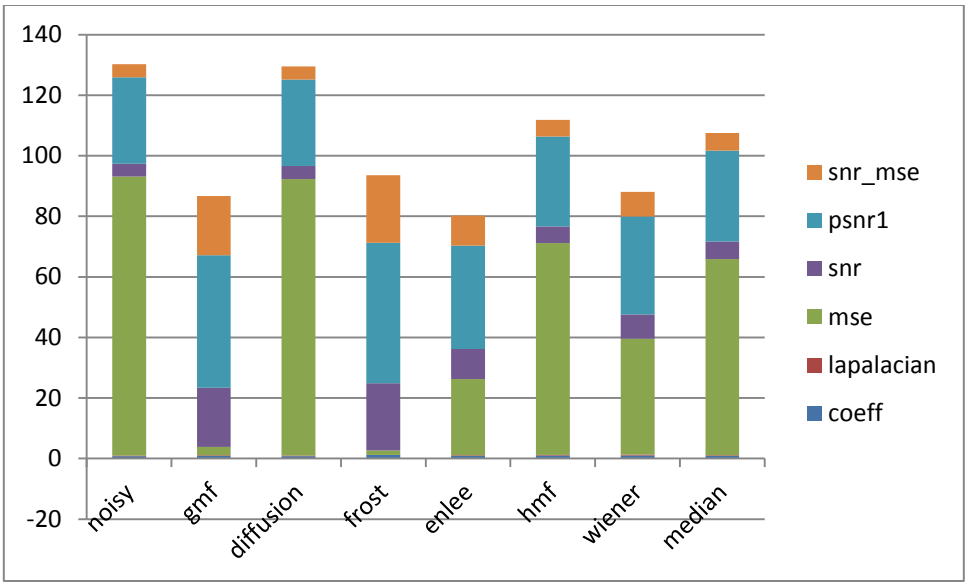


Fig 5.6 Performance of various filters on image.tiff at noise density 0.5

As shown in fig 5.6 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

2(b) For Noise Density = 0.7

Filtering Results

Table 5.5 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.790052	0.163965	141.49	2.431531	26.62354	2.431531
gmf	0.915929	0.166078	15.46839	12.04434	36.23635	12.04434
diffusion	0.796202	0.164246	142.5955	2.397732	26.58975	2.397732
frost	1.197235	-0.1643	1.489063	22.20966	46.40167	22.20966
enlee	0.908896	0.146829	118.5157	3.201032	27.39305	3.201032
hmf	0.920537	0.255742	179.3653	1.401405	25.59342	1.401405
wiener	0.972649	0.223779	184.1041	1.288154	25.48017	1.288154
median	0.902377	0.181513	165.7212	1.745009	25.93702	1.745009

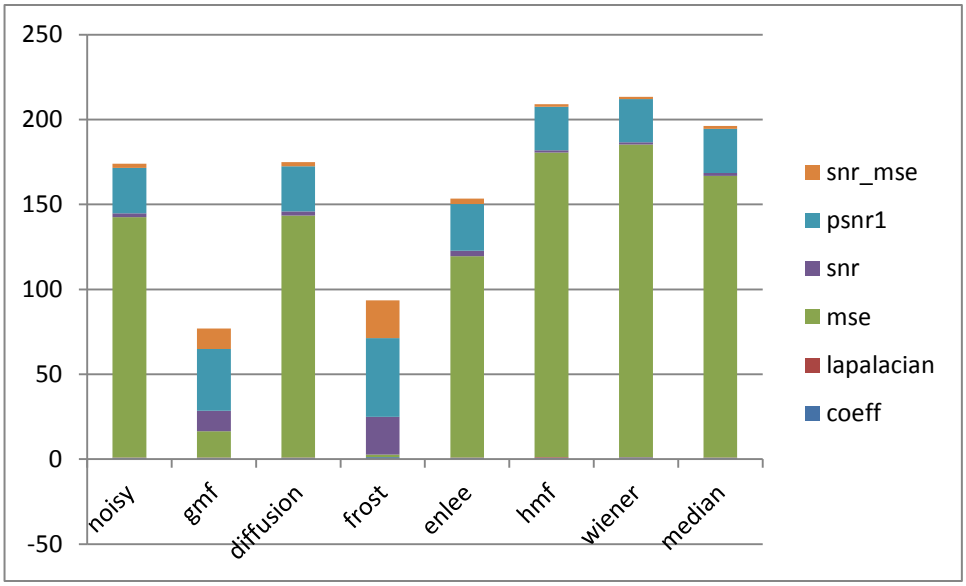


Fig 5.7 Performance of various filters on image.tiff at noise density 0.7

As shown in fig 5.7 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

2(c) For Noise Density = 0.85

Filtering Results

Table 5.6 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.768543	0.143616	178.883	1.4131	25.60511	1.4131
gmf	0.908068	0.149299	49.08486	7.029314	31.22133	7.029314
diffusion	0.771369	0.1492	181.1419	1.358601	25.55061	1.358601
frost	1.19763	-0.15467	1.489716	22.20776	46.39977	22.20776
enlee	0.886966	0.131057	176.6612	1.467378	25.65939	1.467378
hmf	0.909112	0.230861	228.871	0.342882	24.5349	0.342882
wiener	0.959736	0.179492	233.8508	0.249401	24.44142	0.249401
median	0.884111	0.154202	221.9737	0.475774	24.66779	0.475774

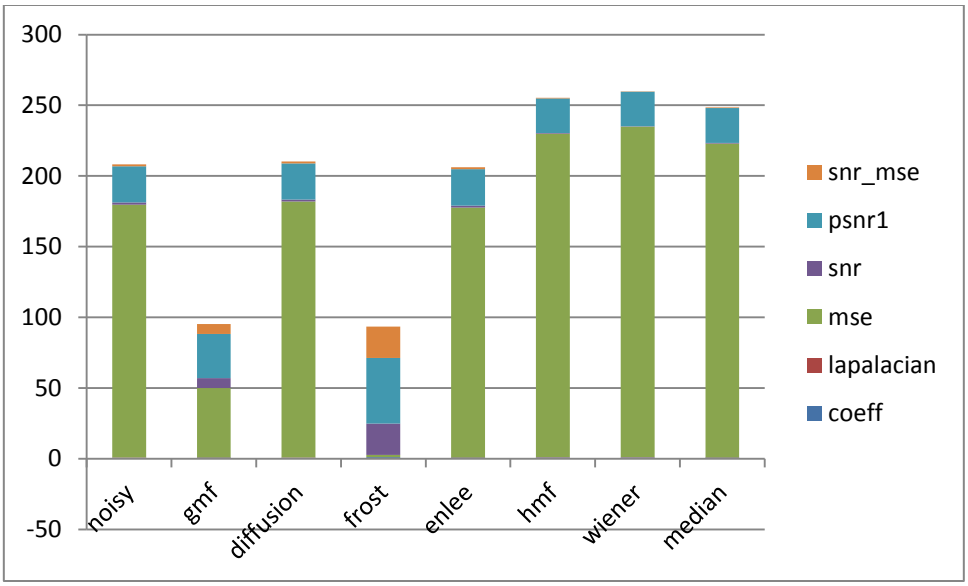


Fig 5.8 Performance of various filters on image.tiff at noise density 0.85

As shown in fig 5.8 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

(3) for synth.tiff

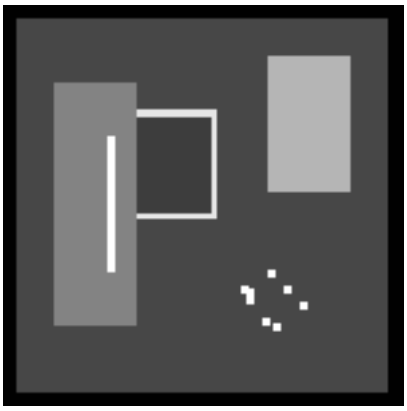


Fig 5.9 synth.tiff

3(a) For Noise Density = 0.5

Filtering Results

Table 5.7 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	mse_snr
noisy	0.83324	0.429387	73.27867	4.787889	29.48103	4.787889
gmf	0.932716	0.563506	2.224	19.96632	44.65946	19.96632
diffusion	0.843841	0.43783	70.23778	4.971957	29.6651	4.971957
frost	1.056044	-0.00304	0.026489	39.20703	63.90017	39.20703
enlee	0.932523	0.506806	22.4276	9.929837	34.62298	9.929837
hmf	0.932523	0.506806	22.4276	9.929837	34.62298	9.929837
wiener	0.972541	0.636365	28.39249	8.90563	33.59877	8.90563
median	0.950461	0.57991	43.01613	7.101351	31.79449	7.101351

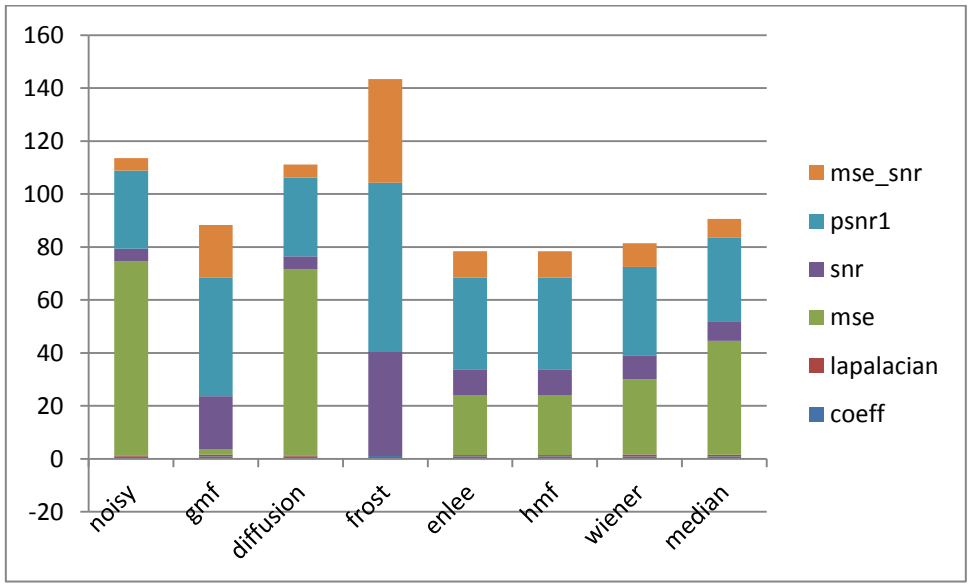


Fig 5.10 Performance of various filters on synth.tiff at noise density 0.5

As shown in fig 5.10 for frost filter the value of mse and laplacian is small and the value of psnr, snr, coefficient and snr_mse is large as required.

3(b) For Noise Density = 0.7

Filtering Results

Table 5.8 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	mse_snr
noisy	0.794341	0.390883	116.7686	2.764403	27.45754	2.764403
gmf	0.934751	0.555585	8.080889	14.36307	39.05621	14.36307
diffusion	0.810537	0.396027	117.663	2.731265	27.4244	2.731265
frost	0.926686	0.011193	0.032044	38.38014	63.07328	38.38014
enlee	0.909436	0.392256	109.6776	3.036487	27.72963	3.036487
hmf	0.948135	0.56214	128.1463	2.360605	27.05374	2.360605
wiener	0.97314	0.549827	128.2602	2.356746	27.04989	2.356746
median	0.942245	0.490399	122.4238	2.559007	27.25215	2.559007

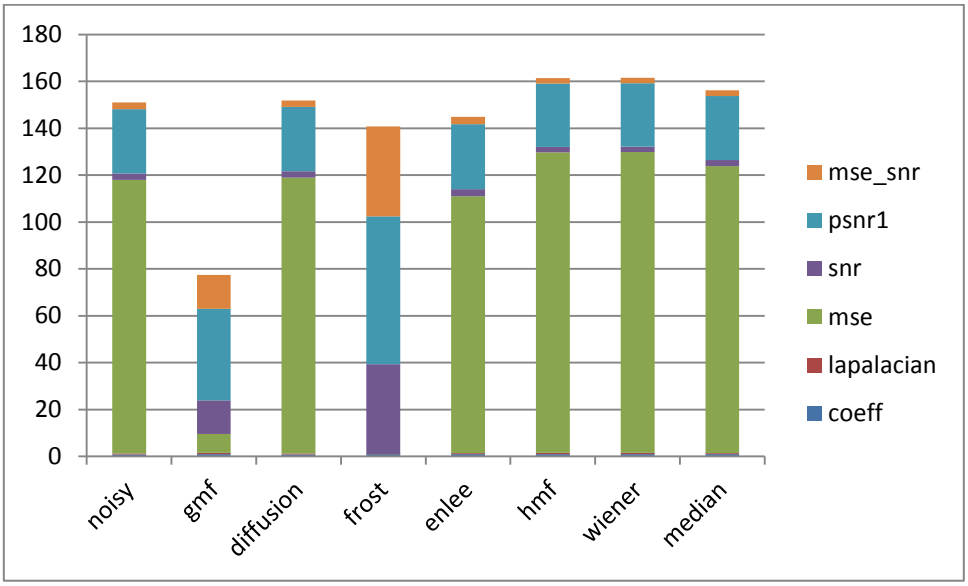


Fig 5.11 Performance of various filters on synth.tiff at noise density 0.7

As shown in fig 5.11 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

3(c) For Noise Density = 0.85

Filtering Results

Table 5.9 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	mse_snr
noisy	0.76715	0.337798	150.2941	1.668244	26.36138	1.668244
gmf	0.934204	0.509056	27.53973	9.038067	33.73121	9.038067
diffusion	0.775729	0.355841	150.8014	1.653612	26.34675	1.653612
frost	0.623165	0.040299	0.048667	36.56535	61.25849	36.56535
enlee	0.909346	0.347121	166.4881	1.223833	25.91697	1.223833
hmf	0.937801	0.394076	191.524	0.615432	25.30857	0.615432
wiener	0.975047	0.436053	205.0448	0.319178	25.01232	0.319178
median	0.936048	0.384349	182.1735	0.832814	25.52595	0.832814

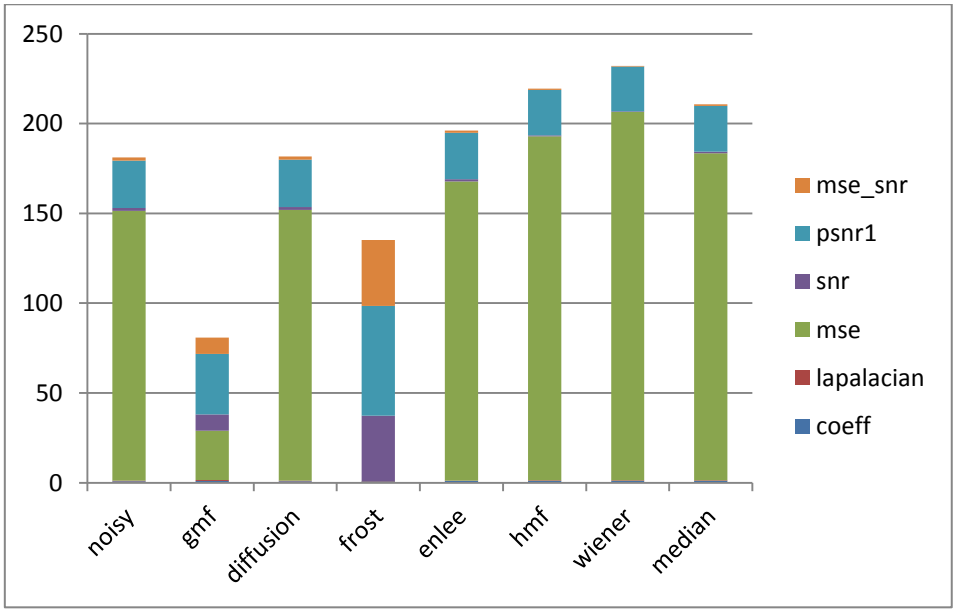


Fig 5.12 Performance of various filters on synth.tiff at noise density 0.85

As shown in fig 5.12 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

(4) For image3.TIF

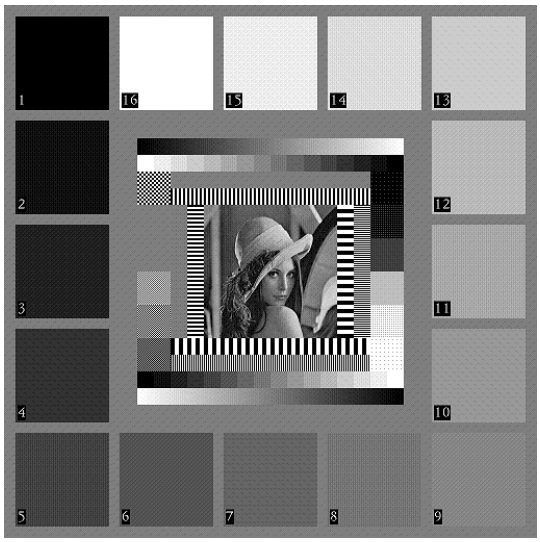


Fig 5.13 image3.TIF

4(a) For Noise Density = 0.5

Filtering Results

Table 5.10 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr	snr_mse
noisy	0.886678	0.753191	83.86029	4.386752	28.89524	4.386752
gmf	0.842453	0.459944	4.241341	17.34728	41.85577	17.34728
diffusion	0.890879	0.753348	83.4695	4.407037	28.91553	4.407037
frost	0.7717	0.67963	1.988297	20.6375	45.14599	20.6375
enlee	0.880174	0.632733	60.4362	5.809344	30.31783	5.809344
hmf	0.848905	0.290142	78.07142	4.697394	29.20588	4.697394
wiener	0.895208	0.898622	72.39263	5.025372	29.53386	5.025372
median	0.912398	0.851772	71.11468	5.102723	29.61121	5.102723

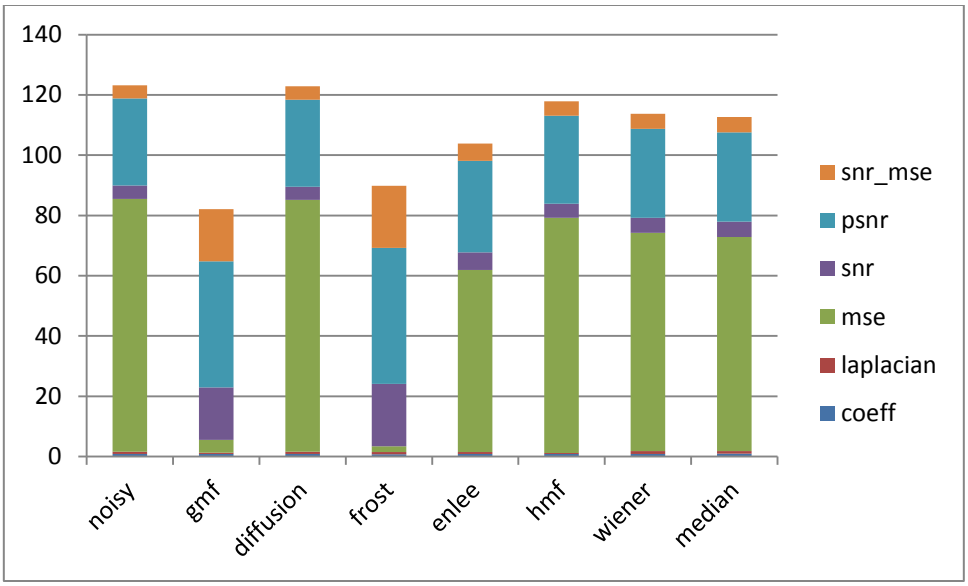


Fig 5.14 Performance of various filters on image3.TIF at noise density 0.5

As shown in fig 5.14 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

4(b) For Noise Density = 0.7

Filtering Results

Table 5.11 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr	snr_mse
noisy	0.866175	0.682349	128.8569	2.521238	27.02973	2.521238
gmf	0.831611	0.454837	17.40044	11.21671	35.7252	11.21671
diffusion	0.871664	0.681535	129.5541	2.497805	27.00629	2.497805
frost	0.7717	0.67963	1.988297	20.6375	45.14599	20.6375
enlee	0.877402	0.596085	129.4895	2.499969	27.00846	2.499969
hmf	0.842799	0.271205	165.0077	1.447273	25.95576	1.447273
wiener	0.890426	0.856289	168.6127	1.353414	25.8619	1.353414
median	0.90352	0.799224	154.8085	1.724367	26.23286	1.724367

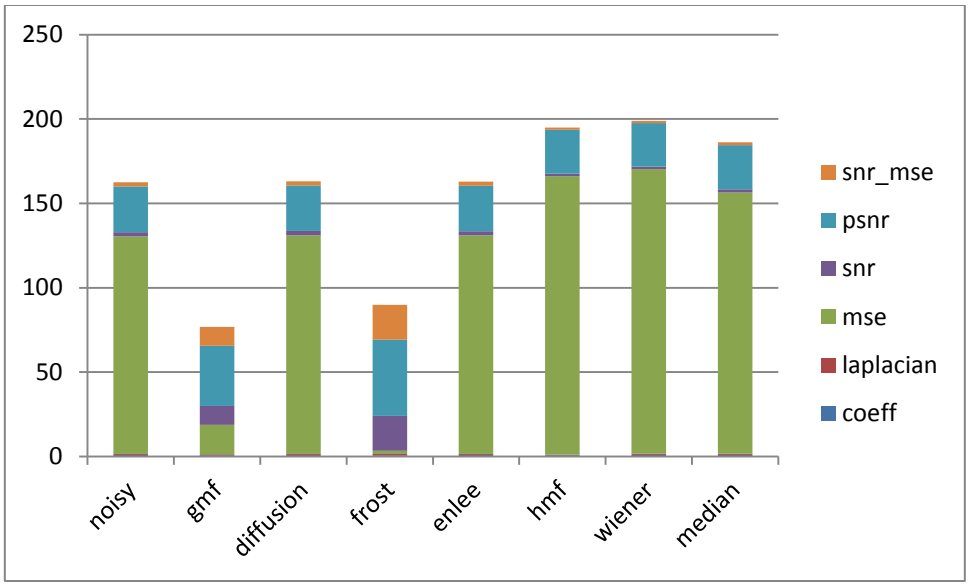


Fig 5.15 Performance of various filters on image3.TIF at noise density 0.7

As shown in fig 5.15 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

4(c) For Noise Density = 0.85

Filtering Results

Table 5.13 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr	snr_mse
noisy	0.848006	0.618477	162.6432	1.509956	26.01844	1.509956
gmf	0.824835	0.442324	48.10795	6.800147	31.30863	6.800147
diffusion	0.852689	0.618918	164.7873	1.453078	25.96157	1.453078
frost	0.7717	0.67963	1.988297	20.6375	45.14599	20.6375
enlee	0.871855	0.551578	165.7385	1.428081	25.93657	1.428081
hmf	0.836096	0.243445	202.9573	0.54827	25.05676	0.54827
wiener	0.889916	0.787893	203.97	0.526653	25.03514	0.526653
median	0.897133	0.752819	198.6373	0.641706	25.15019	0.641706

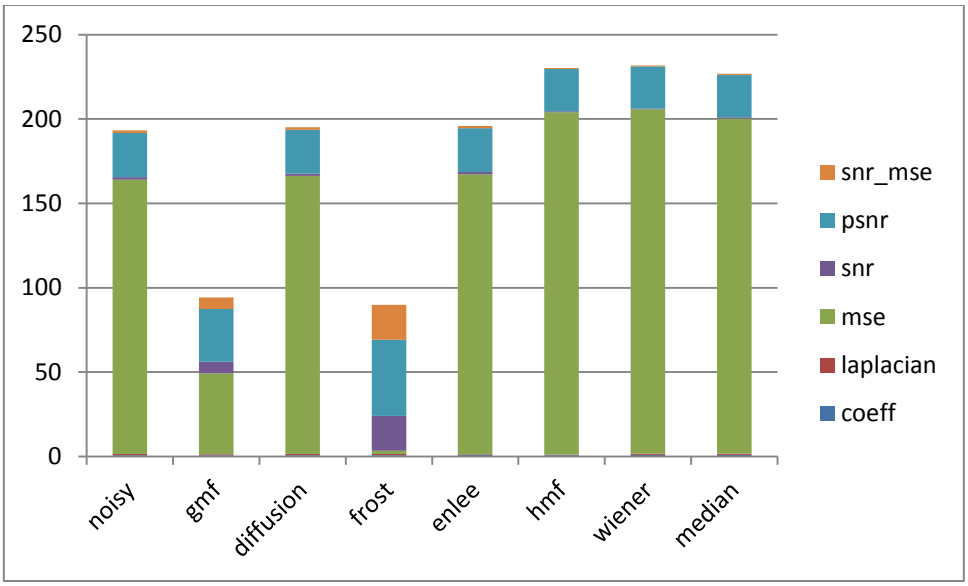


Fig 5.16 Performance of various filters on image3.TIF at noise density 0.85

As shown in fig 5.16 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

From above concluded that enhanced frost filter is best filter for the removal of speckle noise. The application of these filters is in ultrasound images. as ultrasound images contain speckle noise so these filters can be used for removal of speckle noise.

5.2 Gaussian Noise

(1) For image moon.tiff

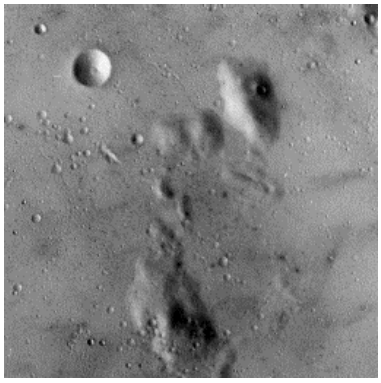


Fig 5.17 moon.tiff

1(a) For Noise Density = 0.03

Filtering Results

Table 5.14 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr1	snr_mse
noisy	0.749266	0.188076	102.8239	3.931186	28.00986	3.931186
gmf	0.770791	0.134452	3.54776	18.55258	42.63126	18.55258
diffusion	0.753406	0.185142	102.3942	3.949371	28.02805	3.949371
frost	0.970826	-0.15843	6.340042	16.0312	40.10988	16.0312
enlee	0.807971	0.090609	40.23285	8.006317	32.085	8.006317
hmf	0.868845	0.274702	68.50545	5.694873	29.77355	5.694873
wiener	0.906209	0.307536	55.38368	6.618306	30.69699	6.618306
median	0.856689	0.249482	76.37827	5.222427	29.30111	5.222427

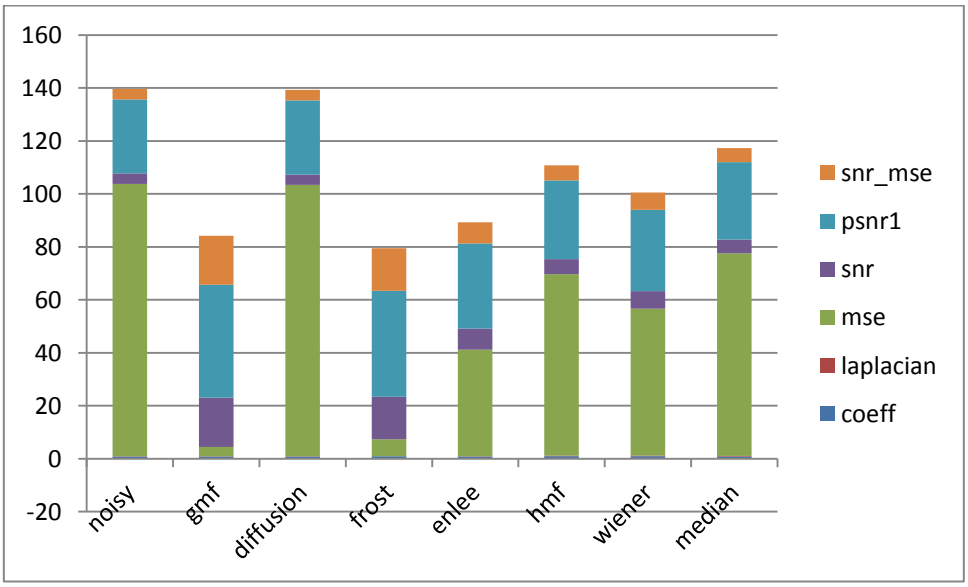


Fig 5.18 Performance of various filters on moon.tiff at noise density 0.03

As shown in fig 5.18 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

1(b) For Noise Density = 0.05

Filtering Results

Table 5.15 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr1	snr_mse
noisy	0.72231	0.144034	107.5959	3.734168	27.81285	3.734168
gmf	0.744111	0.106478	3.657623	18.42013	42.49881	18.42013
diffusion	0.729745	0.154481	106.8582	3.764046	27.84273	3.764046
frost	0.878831	-0.09327	9.893311	14.09871	38.17739	14.09871
enlee	0.776171	0.075166	47.94293	7.244879	31.32356	7.244879
hmf	0.841591	0.240999	79.19591	5.065097	29.14378	5.065097
wiener	0.875922	0.24534	64.784	5.937447	30.01613	5.937447
median	0.815084	0.201697	85.58255	4.728272	28.80695	4.728272

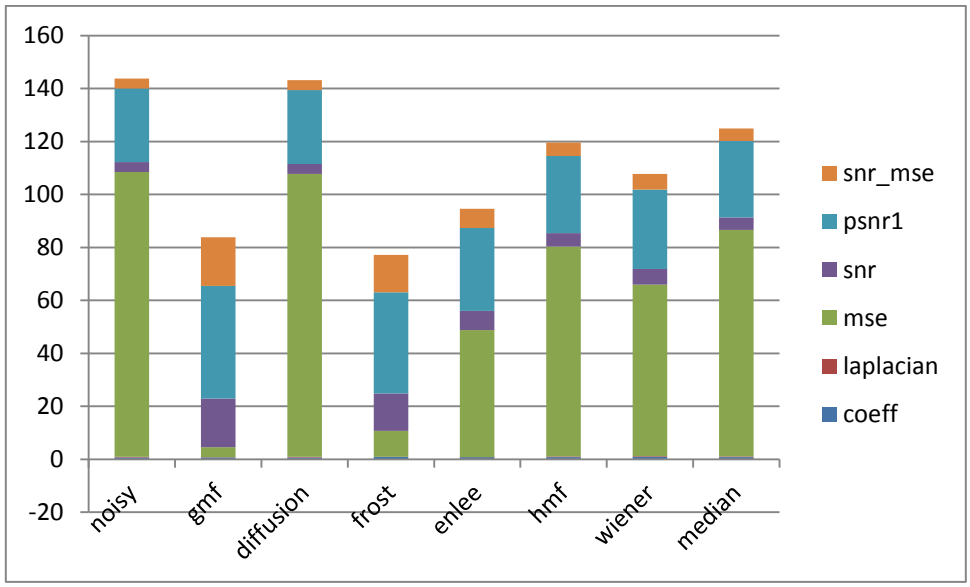


Fig 5.19 Performance of various filters on moon.tiff at noise density 0.05

As shown in fig 5.19 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

1(c) For Noise Density = 0.07

Filtering Results

Table 5.16 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr1	snr_mse
noisy	0.707349	0.117796	110.8301	3.605548	27.68423	3.605548
gmf	0.73888	0.087219	3.933456	18.10438	42.18306	18.10438
diffusion	0.708577	0.127888	111.1069	3.594714	27.67339	3.594714
frost	0.736347	-0.06189	13.9939	12.59274	36.67142	12.59274
enlee	0.751568	0.063827	51.10164	6.967776	31.04646	6.967776
hmf	0.814568	0.209699	84.8876	4.763682	28.84236	4.763682
wiener	0.857197	0.218029	71.80952	5.490304	29.56898	5.490304
median	0.785225	0.172438	94.39999	4.302405	28.38108	4.302405

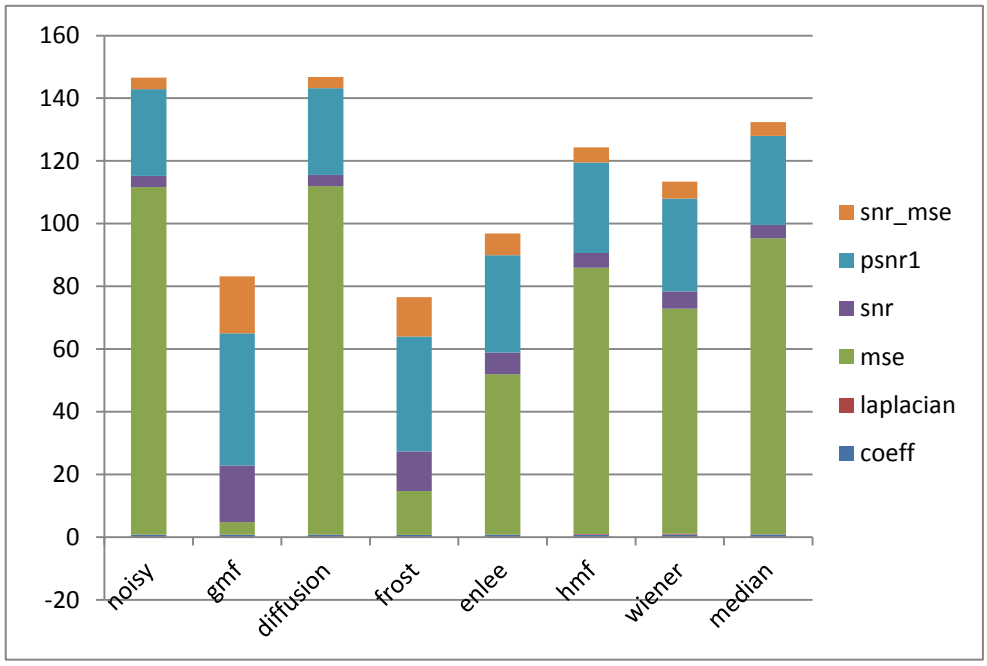


Fig 5.20 Performance of various filters on moon.tiff at noise density 0.07

As shown in fig 5.20 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

(2) For image2.tiff

Fig 5.21 image2.tiff

2(a) For Noise Density = 0.3

Filtering Results

Table 5.17 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.784669	0.117734	100.3125	3.925241	28.11725	3.925241
gmf	0.826115	0.08608	2.195965	20.52254	44.71455	20.52254
diffusion	0.790462	0.121521	99.14217	3.976205	28.16822	3.976205
frost	1.077778	-0.03682	5.525951	16.51472	40.70673	16.51472
enlee	0.856895	0.080513	36.62729	8.300741	32.49276	8.300741
hmf	0.921041	0.228486	67.5017	5.645642	29.83766	5.645642
wiener	0.945829	0.194052	49.47374	6.995042	31.18706	6.995042
median	0.902636	0.159707	73.59826	5.270114	29.46213	5.270114

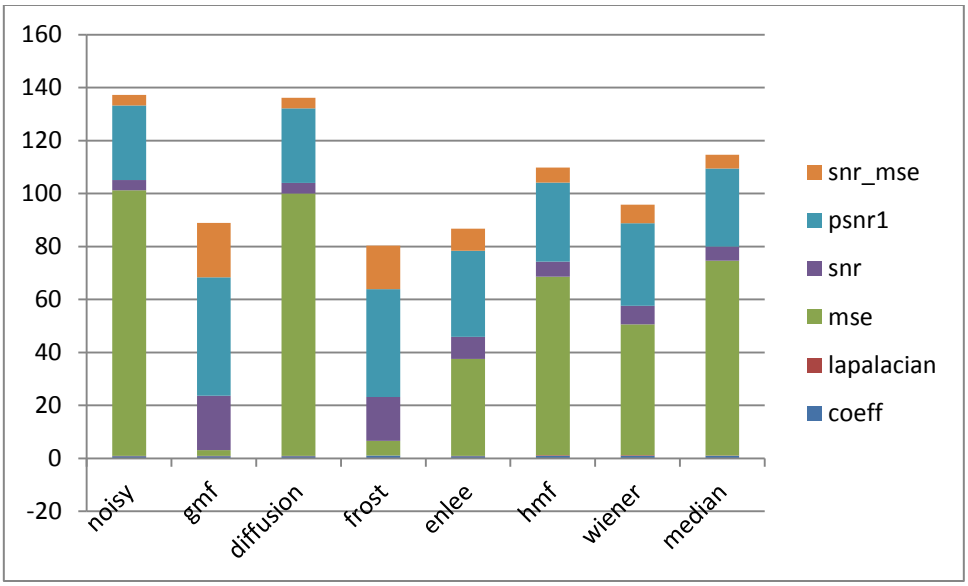


Fig 5.22 Performance of various filters on image2.tiff at noise density 0.03

As shown in fig 5.22 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

2(b) For Noise Density = 0.05

Filtering Results

Table 5.18 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.749896	0.094708	105.7367	3.696533	27.88855	3.696533
gmf	0.790733	0.068075	2.429092	20.08435	44.27636	20.08435
diffusion	0.755675	0.096977	104.5231	3.746667	27.93868	3.746667
frost	0.825324	-0.02048	10.20567	13.85038	38.04239	13.85038
enlee	0.8156	0.062923	43.18223	7.585739	31.77775	7.585739
hmf	0.898537	0.174857	75.29485	5.171137	29.36315	5.171137
wiener	0.924312	0.142541	58.13115	6.2947	30.48671	6.2947
median	0.867788	0.125721	82.56974	4.77058	28.96259	4.77058

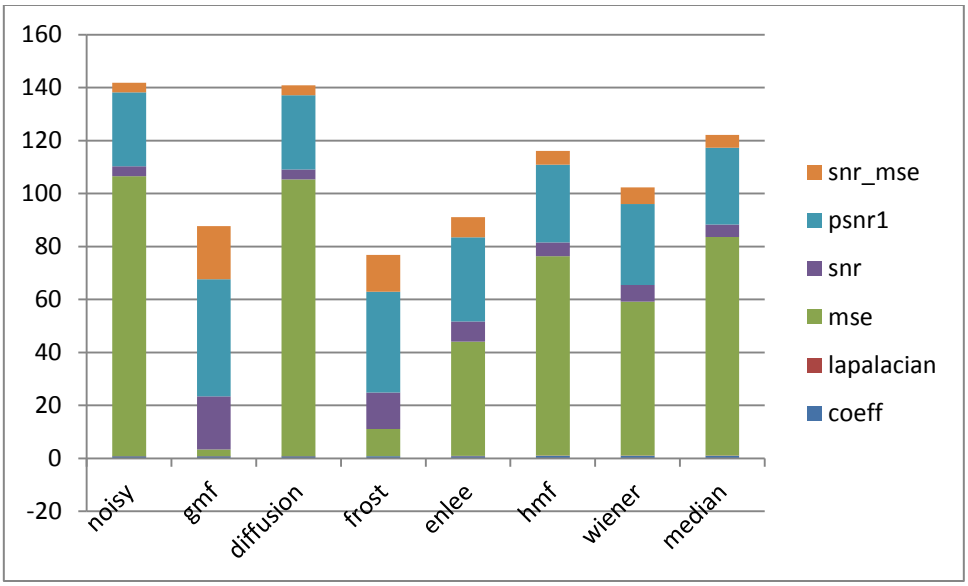


Fig 5.23 Performance of various filters on image2.tiff at noise density 0.05

As shown in fig 5.23 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

2(c) For Noise Density = 0.07

Filtering Results

Table 5.19 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.729209	0.083731	108.2799	3.59331	27.78532	3.59331
gmf	0.77573	0.056729	2.586166	19.81222	44.00424	19.81222
diffusion	0.733314	0.080565	107.4549	3.626527	27.81854	3.626527
frost	0.772118	-0.00991	14.84528	12.22291	36.41492	12.22291
enlee	0.787369	0.052786	47.26477	7.193414	31.38543	7.193414
hmf	0.874499	0.156096	84.5473	4.667792	28.85981	4.667792
wiener	0.90959	0.118508	63.26867	5.926902	30.11892	5.926902
median	0.842854	0.10632	89.96495	4.398056	28.59007	4.398056

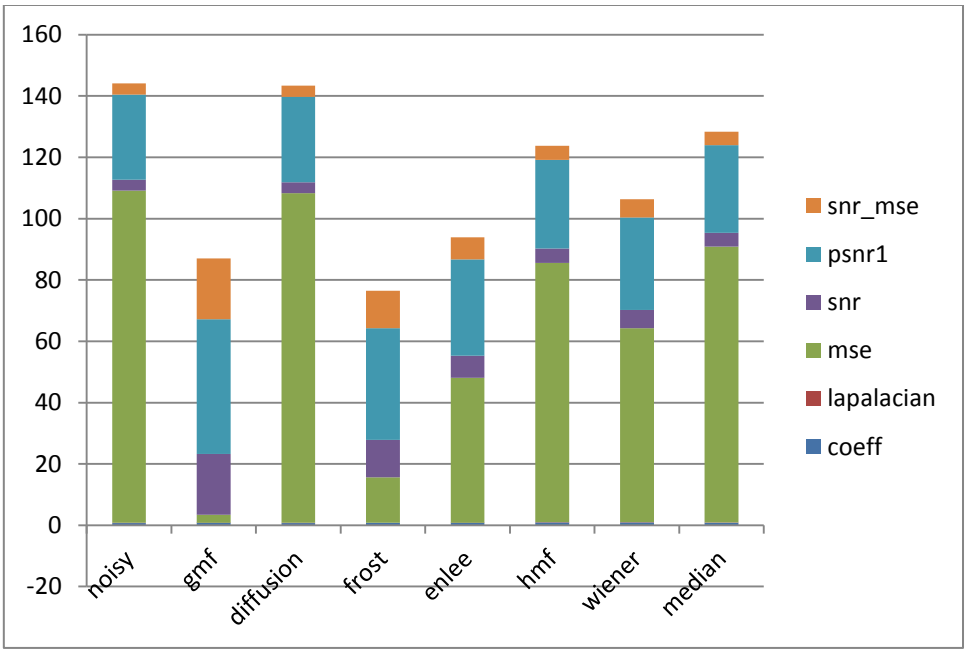


Fig 5.24 Performance of various filters on image2.tiff at noise density 0.07

As shown in fig 5.24 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

(3) For image synth.tiff

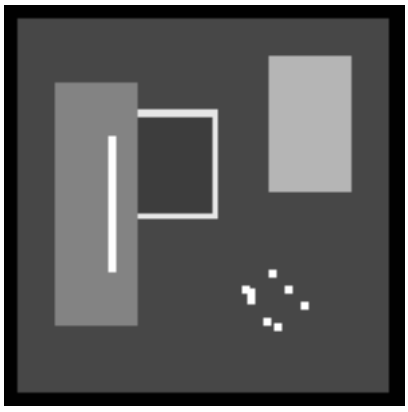


Fig 5.25 synth.tiff

3(a) For Noise Density = 0.03

Filtering Results

Table 5.20 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.669112	0.225217	88.72547	3.957182	28.65032	3.957182
gmf	0.732151	0.248852	2.234311	19.94623	44.63937	19.94623
diffusion	0.669536	0.227375	88.24791	3.98062	28.67376	3.98062
frost	0.484986	0.063077	8.908978	13.93939	38.63252	13.93939
enlee	0.772397	0.174157	35.85844	7.89175	32.58489	7.89175
hmf	0.869773	0.520817	58.48778	5.767014	30.46015	5.767014
wiener	0.926984	0.511978	41.56947	7.24992	31.94306	7.24992
median	0.819141	0.357289	67.75511	5.128244	29.82138	5.128244

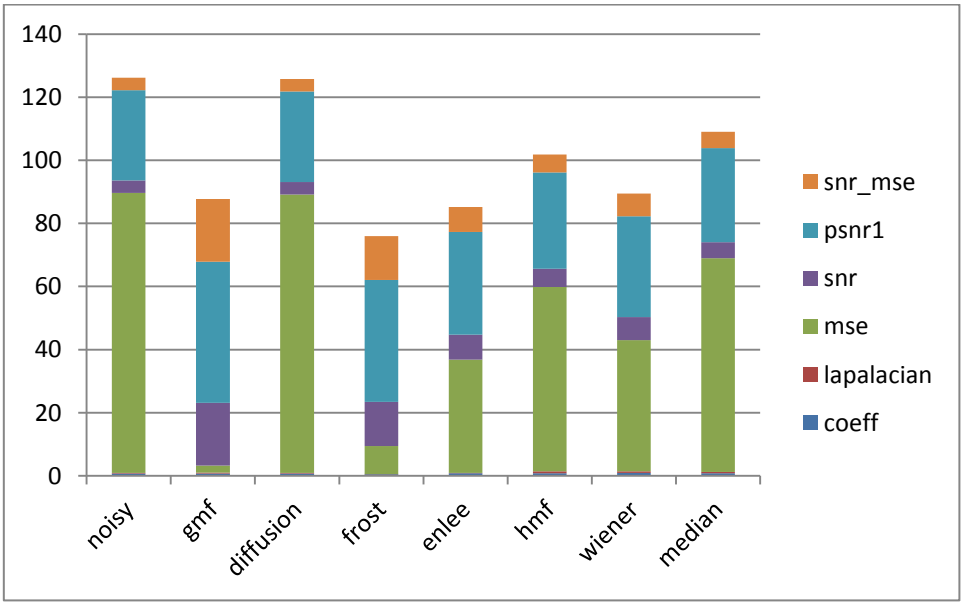


Fig 5.26 Performance of various filters on synth.tiff at noise density 0.03

As shown in fig 5.26 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

3(b) For Noise Density = 0.05

- Filtering Results

Table 5.21 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.620327	0.173458	93.47538	3.730692	28.42383	3.730692
gmf	0.68065	0.202967	2.180178	20.05275	44.74588	20.05275
diffusion	0.629333	0.175347	92.56387	3.77325	28.46639	3.77325
frost	0.494012	0.057371	19.1436	10.61743	35.31057	10.61743
enlee	0.748074	0.130304	39.14142	7.511299	32.20444	7.511299
hmf	0.812352	0.378122	74.43582	4.719845	29.41298	4.719845
wiener	0.881241	0.378715	47.80751	6.642703	31.33584	6.642703
median	0.773273	0.250968	79.284	4.445809	29.13895	4.445809

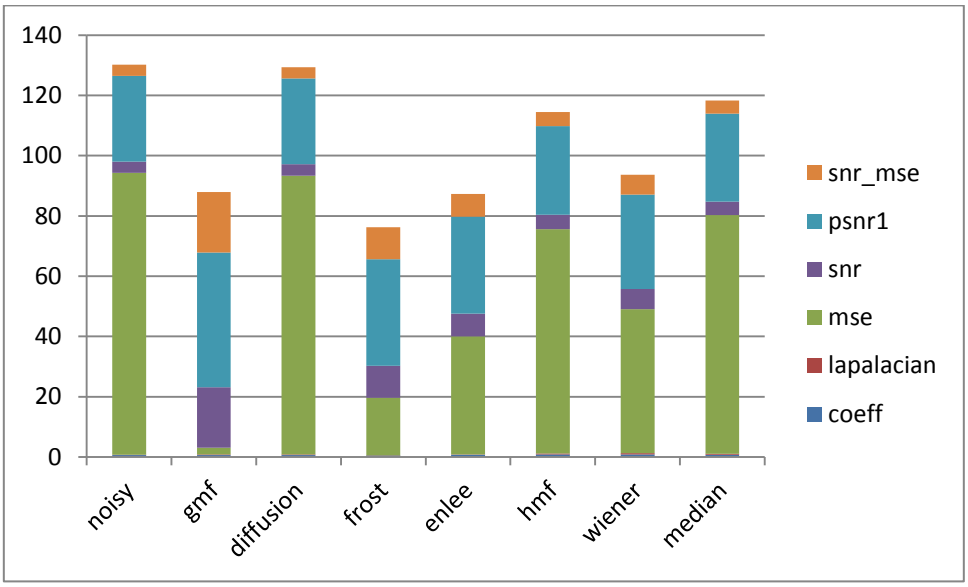


Fig 5.27 Performance of various filters on synth.tiff at noise density 0.05

As shown in fig 5.27 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

3(c) For Noise Density = 0.07

Filtering Results

Table 5.22 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.592596	0.139065	96.7896	3.579378	28.27252	3.579378
gmf	0.641296	0.142439	2.322222	19.77863	44.47177	19.77863
diffusion	0.598347	0.142517	96.34253	3.599484	28.29262	3.599484
frost	0.497322	0.052151	25.83018	9.316391	34.00953	9.316391
enlee	0.729761	0.110001	42.37284	7.166789	31.85993	7.166789
hmf	0.795017	0.26761	84.84867	4.151215	28.84435	4.151215
wiener	0.856547	0.278453	46.45413	6.767421	31.46056	6.767421
median	0.737085	0.196642	78.92507	4.465515	29.15865	4.465515

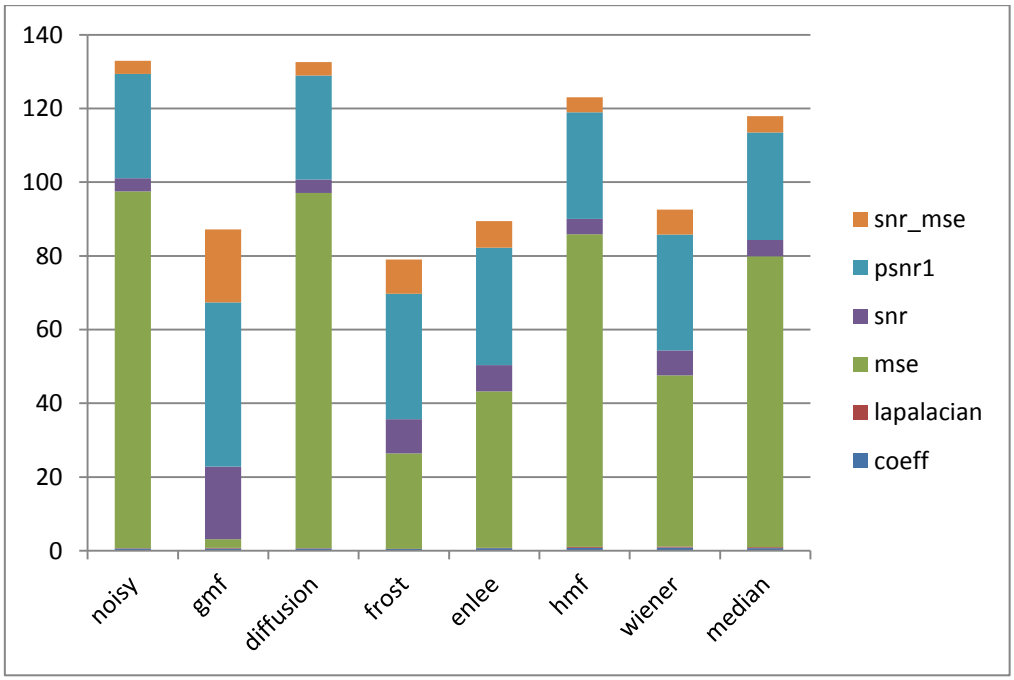


Fig 5.28 Performance of various filters on synth.tiff at noise density 0.07

As shown in fig 5.28 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

(4) for image3.TIF

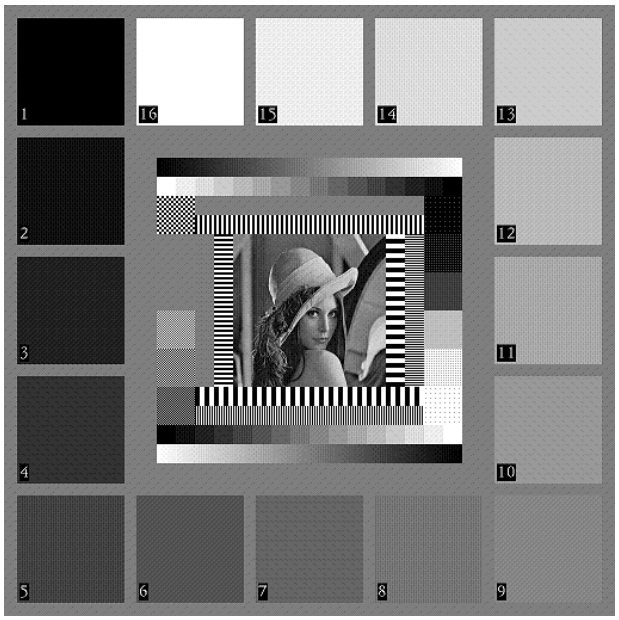


Fig 5.29 image 3.TIF

4(a) For Noise Density = 0.03

Filtering Results

Table 5.23 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr1	snr_mse
noisy	0.853489	0.608835	93.35618	3.920885	28.42937	3.920885
gmf	0.818126	0.342924	3.715412	17.92225	42.43073	17.92225
diffusion	0.855283	0.606175	93.19356	3.928456	28.43694	3.928456
frost	0.763912	0.262734	9.709969	13.75014	38.25863	13.75014
enlee	0.847105	0.324035	56.92839	6.069027	30.57751	6.069027
hmf	0.844618	0.293398	73.32501	4.969794	29.47828	4.969794
wiener	0.881117	0.871042	72.04565	5.046238	29.55473	5.046238
median	0.896835	0.801325	75.73619	4.829281	29.33777	4.829281

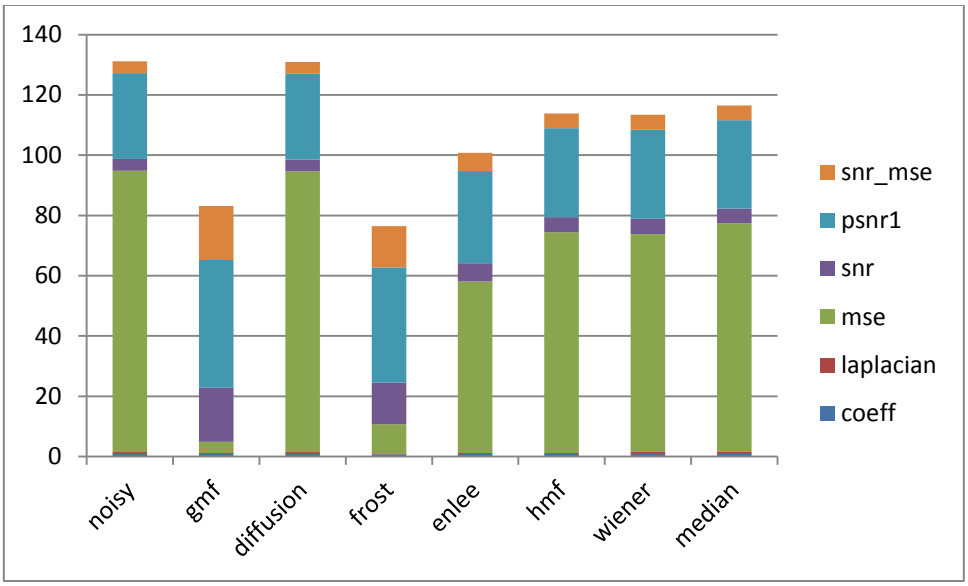


Fig 5.30 Performance of various filters on image3.TIF at noise density 0.03

As shown in fig 5.30 non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

4(b) For Noise Density = 0.05

Filtering Result

Table 5.24 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr1	snr_mse
noisy	0.82287	0.505413	98.00574	3.7098	28.21829	3.7098
gmf	0.812617	0.284442	3.842583	17.77608	42.28457	17.77608
diffusion	0.824766	0.50322	98.46264	3.689601	28.19809	3.689601
frost	0.768239	0.243287	13.70407	12.25382	36.76231	12.25382
enlee	0.831801	0.260046	61.15121	5.758265	30.26675	5.758265
hmf	0.836631	0.255556	80.89708	4.542987	29.05148	4.542987
wiener	0.875609	0.816678	78.80405	4.65683	29.16532	4.65683
median	0.880414	0.712471	84.37938	4.359952	28.86844	4.359952

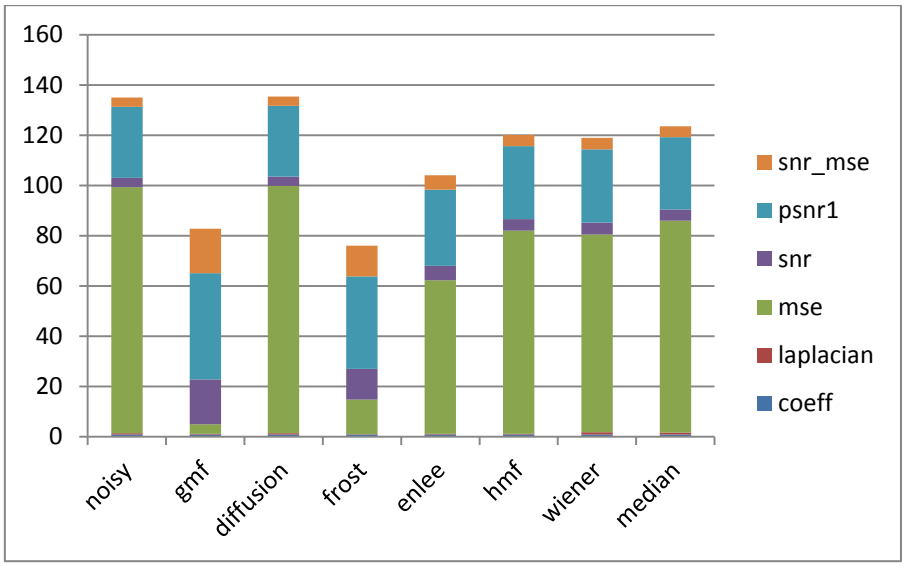


Fig 5.31 Performance of various filters on image3.TIF at noise density 0.05

As shown in fig 5.31 for non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

4(c) For Noise Density = 0.07

Filtering Results

Table 5.25 Statistical results of image with noise and after filtering of noise

	coeff	laplacian	mse	snr	psnr1	snr_mse
noisy	0.798632	0.437128	101.0195	3.578262	28.08675	3.578262
gmf	0.804222	0.249204	4.068882	17.52756	42.03605	17.52756
diffusion	0.801713	0.440349	100.6829	3.592758	28.10125	3.592758
frost	0.769442	0.227898	17.60284	11.16649	35.67498	11.16649
enlee	0.821418	0.233513	62.9727	5.630792	30.13928	5.630792
hmf	0.832848	0.230771	85.22548	4.316621	28.82511	4.316621
wiener	0.867493	0.760599	84.00608	4.379208	28.8877	4.379208
median	0.867816	0.646954	88.80507	4.137938	28.64643	4.137938

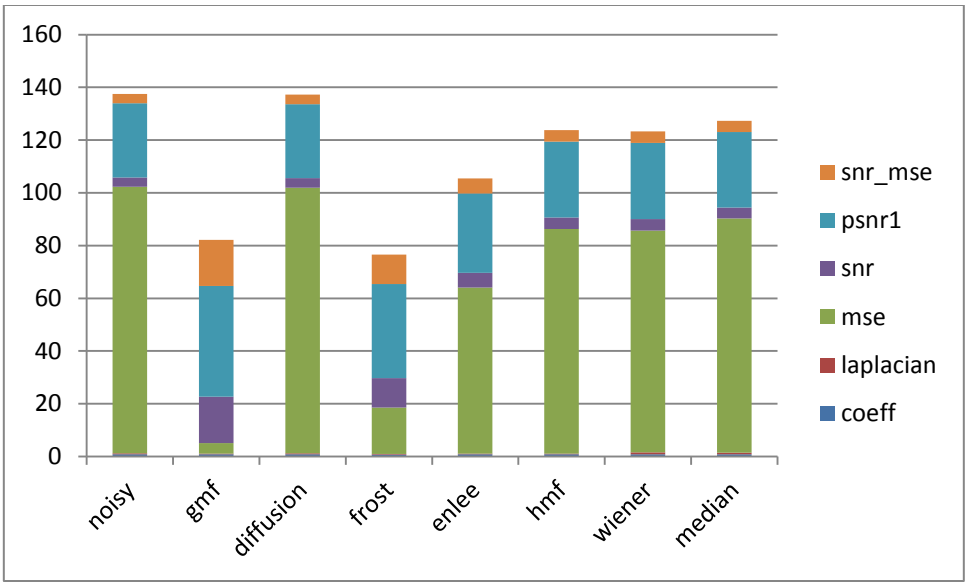


Fig 5.32 Performance of various filters on image3.TIF at noise density 0.07

As shown in fig 5.32 for non linear geometric filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

From the above tables and graphs it is concluded that for the removal Gaussian noise non linear geometric filter is best. This filter gives us the best values of the parameters calculated above. Application of this Gaussian noise is there in fingerprint image enhancement.

5.3 Poisson Noise

(1) For image moon.tiff

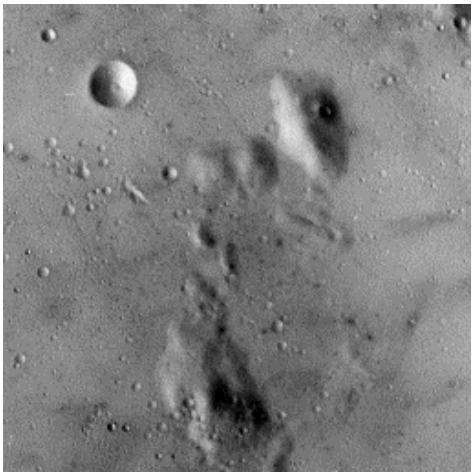


Fig 5.33 moon.tiff

Table 5.26 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.934321	0.594101	46.01678	7.422962	31.50164	7.422962
gmf	0.918493	0.422817	2.704224	19.7317	43.81038	19.7317
diffusion	0.937622	0.602174	44.64648	7.554252	31.63293	7.554252
frost	1.016365	-0.36335	3.96962	18.06464	42.14331	18.06464
enlee	0.954634	0.264278	15.85974	12.04916	36.12784	12.04916
hmf	0.951235	0.477922	34.19888	8.712005	32.79068	8.712005
wiener	0.963179	0.609339	25.54474	9.97911	34.05779	9.97911
median	0.962904	0.601178	26.0069	9.901239	33.97992	9.901239

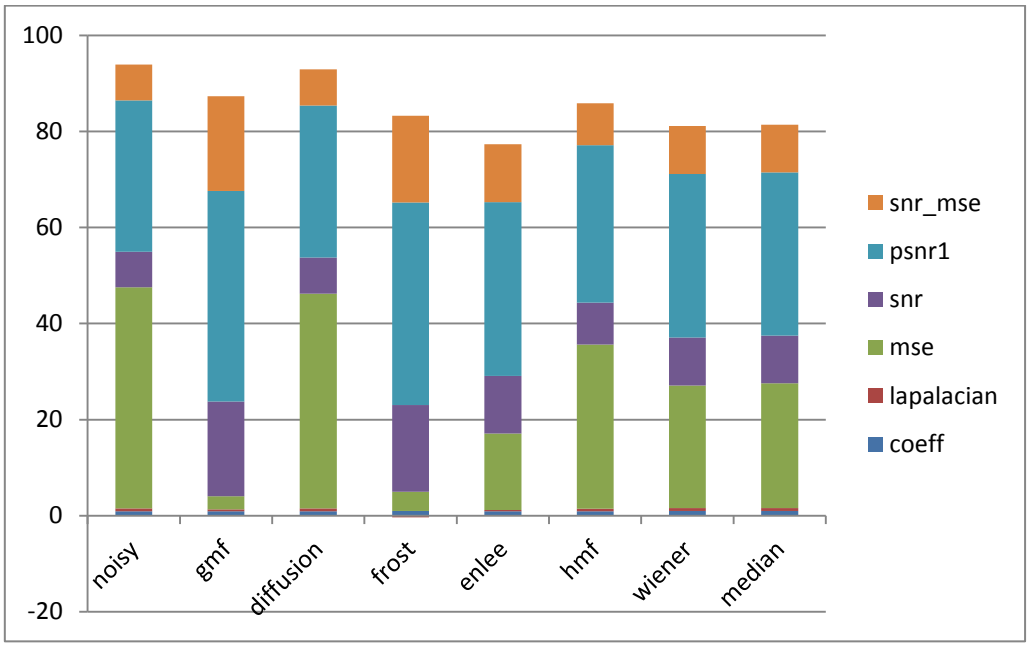


Fig 5.34 Performance of various filters on moon.tiff

As shown in fig 5.34 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

(2) For image2.tiff

Fig 5.35 image2.tiff

Filtering Results

Table 5.27 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.963929	0.446409	43.23343	7.580593	31.77261	7.580593
gmf	0.944325	0.310203	2.02924	20.86546	45.05747	20.86546
diffusion	0.966204	0.454585	41.49127	7.759222	31.95124	7.759222
frost	1.198472	-0.13707	1.496223	22.18882	46.38084	22.18882
enlee	0.983505	0.22846	13.49047	12.63852	36.83053	12.63852
hmf	0.975592	0.442055	29.22217	9.281665	33.47368	9.281665
wiener	0.98452	0.572996	18.83585	11.18894	35.38095	11.18894
median	0.985374	0.527305	20.15642	10.89466	35.08667	10.89466

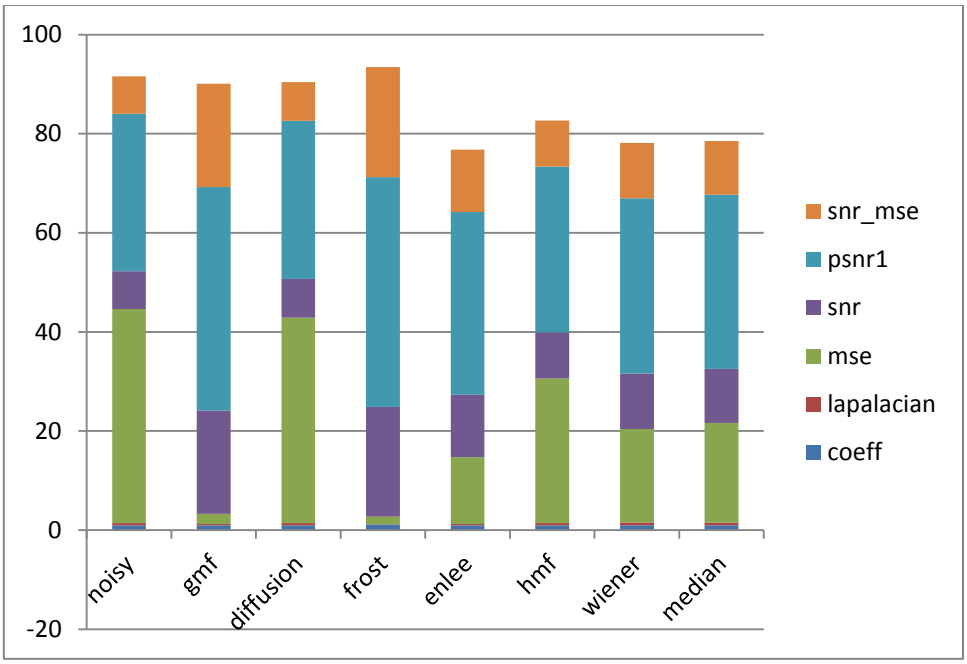


Fig 5.36 Performance of various filters on image2 .tiff

As shown in fig 5.36 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

(3) For image synth.tiff

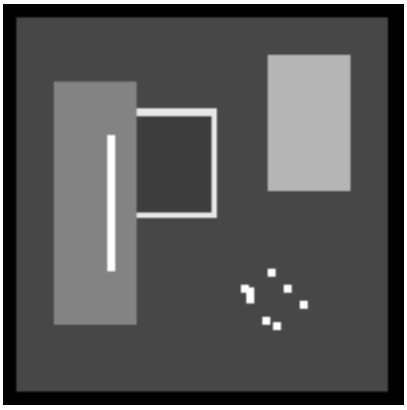


Fig 5.37 synth.tiff

Filtering Results

Table 5.28 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.967655	0.75915	31.43942	8.462919	33.15606	8.462919
gmf	0.932532	0.652788	0.819778	24.3007	48.99384	24.3007
diffusion	0.972556	0.765343	31.02982	8.519872	33.21301	8.519872
frost	0.473211	0.051058	0.068356	35.08993	59.78307	35.08993
enlee	0.945921	0.525558	8.872756	13.95708	38.65022	13.95708
hmf	0.987912	0.878251	5.502889	16.03176	40.7249	16.03176
wiener	0.992194	0.930233	14.53467	11.81361	36.50675	11.81361
median	0.998986	0.940359	10.68982	13.14796	37.8411	13.14796

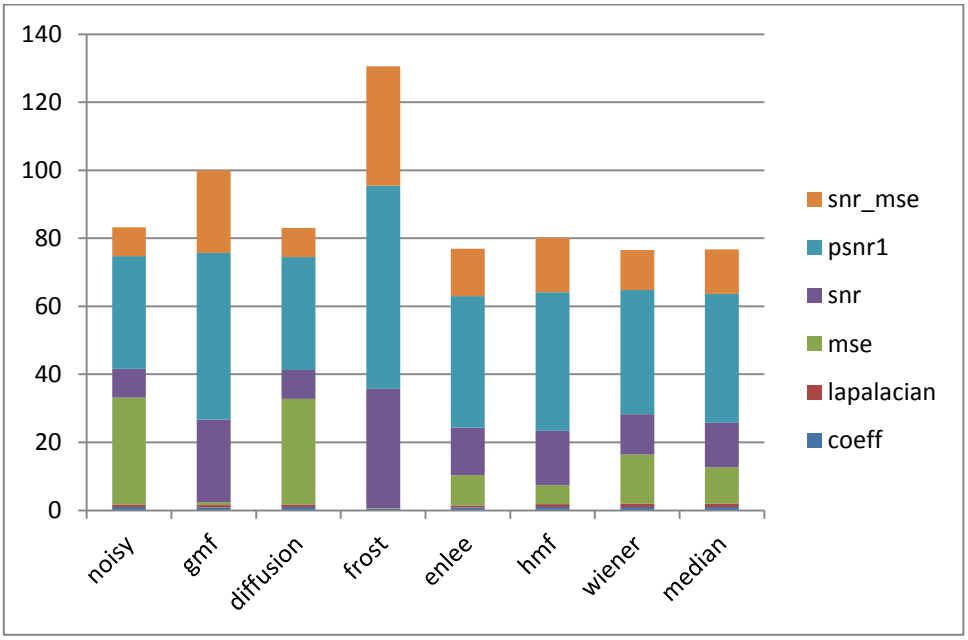


Fig 5.38 Performance of various filters on synth.tiff

As shown in fig 5.38 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

(4) For image3.TIF

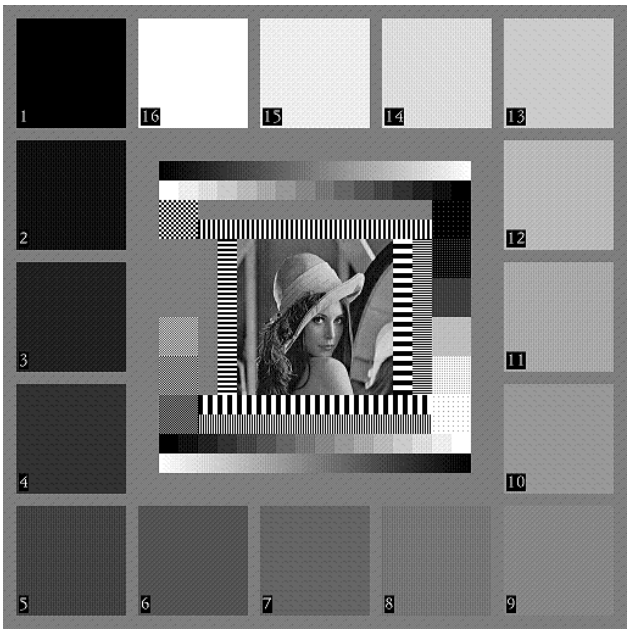


Fig 5.39 image3.TIF

Filtering Results

Table 5.29 Statistical results of image with noise and after filtering of noise

	coeff	lapalacian	mse	snr	psnr1	snr_mse
noisy	0.972717	0.955581	41.76552	7.414136	31.92262	7.414136
gmf	0.835395	0.505033	2.278164	20.04647	44.55495	20.04647
diffusion	0.969128	0.955264	41.4564	7.4464	31.95489	7.4464
frost	0.7717	0.67963	1.988297	20.6375	45.14599	20.6375
enlee	0.872393	0.715754	45.414	7.050418	31.55891	7.050418
hmf	0.882123	0.326368	49.24272	6.698895	31.20738	6.698895
wiener	0.905787	0.934595	55.17252	6.205088	30.71358	6.205088
median	0.968149	0.943094	30.28962	8.809377	33.31787	8.809377

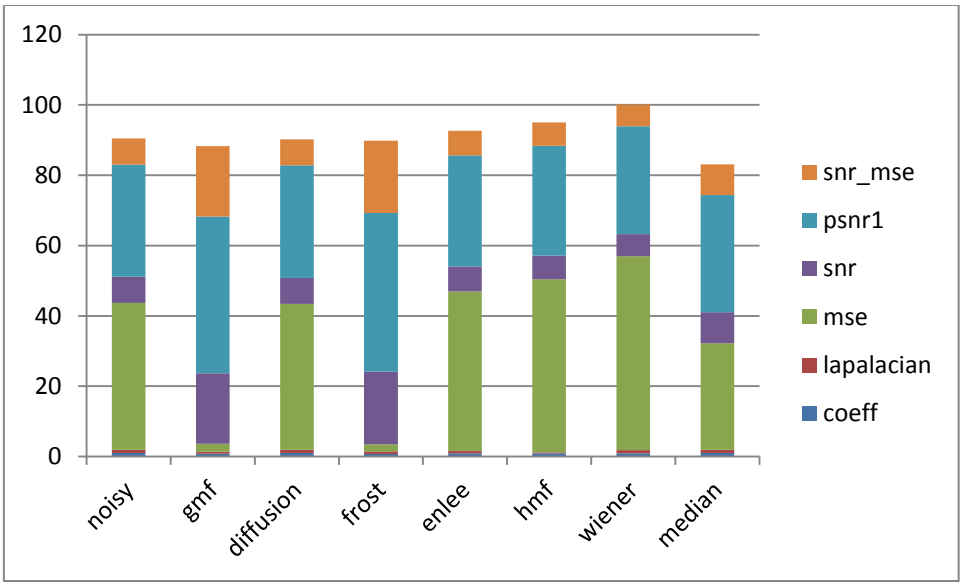


Fig 5.40 Performance of various filters on image3.TIFF

As shown in fig 5.40 for frost filter the value of mse and laplacian is small and the value of psnr , snr, coefficient and snr_mse is large as required.

From the above tables and graph it can be seen that for the removal of poisson noise enhanced frost filter is best suited. This filter removes poisons noise and the application of this filter is in radiography images as this noise is found in those images.

5.4 PROPOSED ALGORITHM

1. Firstly, the image acquisition is done.
2. Then the detection of the noise is done by using mathematical distribution.
3. Find the suitable filter and then implement on it.

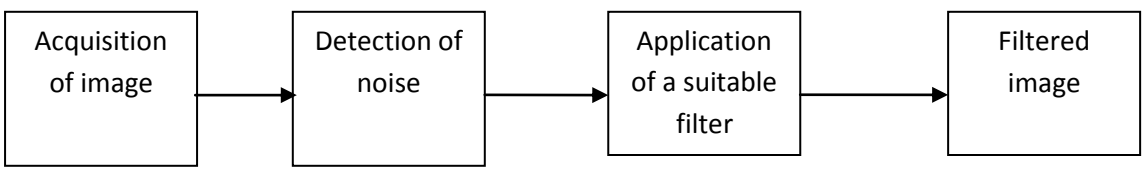


Fig 5.41 Block Diagram

CHAPTER 6
FUTURE SCOPE

In this thesis non linear geometric, diffusion, enhanced frost, enhanced lee, homogeneous mass area, wiener, median filters have been used to remove speckle, Gaussian, Poisson noises. Parameters such as coefficient correlation, laplacian, mean square error, peak signal to noise ratio have been used to know which filter gives us the best values of these parameters. Firstly, speckle noise is removed at different noise density and filter which removes it at its best is enhanced frost filter which removes the noise efficiently. Values of different parameters are calculated from where best filter is analyzed. Secondly, Gaussian noise is removed at different noise density and filter which removes it is non linear geometric filter. This filter gives us the best results of different parameters. Then Poisson noise is removed with the help of these filters and the filter which comes out best is filter is enhanced frost filter.

In this thesis it suggest automatic filtering technique which will work by using mathematical distribution, we can estimate the type of noise in an unknown image and then implement the desired filter on it.

REFERENCES

[1] Yongjian Yu and Scott T. Acton , "Speckle Reducing Anisotropic Diffusion" IEEE Transactions On Image Processing, 2002, Vol.11 , pages:1260-1270.

[2] Kiyo Tomiyasu,"Computer Simulation Of Speckle in a Synthetic Aperture Radar Image Pixel", IEEE Transactions On Geoscience And Remote Sensing, 1983, Vol.21, pages: 357-363.

[3] Armand Lopes, Ridha Touzi. and E. Nezry , "Adaptive Speckle Filters and Scene Heterogeneity", IEEE Transactions On Geoscience And Remote Sensing, 1990, Vol. 28, pages: 992-1000.

[4] Mustafa Karaman, M. Alper Kutay, and Gozde Bozdagi, "An Adaptive Speckle Suppression Filter for Medical Ultrasonic Imaging", IEEE Transactions On Medical Imaging, 1995 , Vol. 14, pages:283-292.

[5] Richard N. Czerwinska Douglas L. Jones William D. O'Brien, Jr, "Ultrasound Speckle Reduction By Directional Median Filtering" , IEEE International conference on image processing 1995, Vol.1, pages:358-361.

[6] Chedsada Chinrungrueng and Aimamorn Suvichakorn, "Fast Edge-Preserving Noise Reduction for Ultrasound Images" IEEE Transactions On Nuclear Science, 2001, Vol.48, pages:849-854.

[7] Yongjian Yy Janelle A. Molloy and Scott T. Acton , "Three-Dimensional Speckle Reducing Anisotropic Diffusion, "IEEE Transactions On Image Processing, 2002, Vol.11, pages: 1260-1270.

[8] Khalifa Djemal, "Speckle Reduction In Ultrasound Images By Minimization Of Total Variation", IEEE International Conference On Image Processing, 2005, Vol.11, pages: 357-360.

[9] G. Deng and L. W.Cahill , "An Adaptive Gaussian Filter For Noise Reduction and Edge Detection", 1994 , IEEE Nucl. Sci. Symp. Med. Im. Conf. pages: 1615-1619.

[10] James C. Brailean, Richard P. Kleihorst, Serafim Efstratiadis, Aggelos K. Katsaggelos And Reginald L. Lagendijk , "Noise Reduction Filters for Dynamic lmage Sequences: A Review",1995, Proceedings of the IEEE, Vol. 83, pages: 1272-1292.

[11] Eero P. Simoncelli and ehdward H. Adelson , “Noise Removal Via Bayesian Wavelet Coring”, 1996, Proc. IEEE Int. Conf. Image Processing, pages: 379-382.

[12] Dimitri Van De Ville, Mike Nachtegael, Dietrich Van der Weken, Etienne E. Kerre, Wilfried Philips and Ignace Lemahieu , “Noise Reduction by Fuzzy Image Filtering”, 2003, IEEE Transactions On Fuzzy Systems, Vol. 11, pages: 429-436.

[13] M. Nachtegael, S. Schulte, D. Van der Weken, V. De Witte, E.E. Kerre , “Fuzzy Filters for Noise Reduction: the Case of Gaussian Noise”, 2005, IEEE International Conference on Fuzzy Systems, pages: 201-206.

[14] Stefan Schulte, Mike Nachtegael, Valérie De Witte, Dietrich Van der Weken, and Etienne E. Kerre , “A Fuzzy Impulse Noise Detection and Reduction Method”, 2006, IEEE Transactions On Image Processing, Vol. 15, pages: 1153-1162.

[15] Triet Le, Rick Chartrand and Thomas J. Asaki, “A Variational Approach to Reconstructing Images Corrupted by Poisson Noise”, 2007, pages: 257-263.

[16] Michael R. Keenan∗ and Paul G. Kotula, “Accounting for Poisson noise in the multivariate analysis of ToF-SIMS spectrum images”, 2004, John Wiley & Sons, Ltd, pages: 203-212.

[17] Piotr S. Windyga , “Fast Impulsive Noise Removal” , 2001, IEEE Transactions On Image Processing, Vol. 10, pages: 173-179.

[18] Krisana Chinnasarn, Yuttapong Rangsanseri and Punya Thitimajshima , “Removing Salt-and-Pepper Noise in Text/Graphics Images” , 1998, IEEE Transactions On Image Processing, pages: 259-269.

[19] Raymond H. Chan, Chung-Wa Ho, and Mila Nikolova, “Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization” , 2005, IEEE Transactions On Image Processing, Vol. 14, pages: 1479-1485.

[20] Kenny Kal Vin Toh, Haidi Ibrahim, and Muhammad Nasiruddin Mahyuddin, “Salt-and-Pepper Noise Detection and Reduction Using Fuzzy Switching Median Filter”, 2008, IEEE Transactions on Consumer Electronics, Vol. 54, pages: 1956-1961.

[21] Chaohong Wu, Z.Shi, V. Govindaraju “Fingerprint Image Enhancement Method Using Directional Median Filter ” ,2004 Elevier Science,pages: 1-15.

[22] L.Meng, Y.Ma, X.Zheng, J.Dun, J.Gu "A New Fingerprint Image Enhancement Based On Decimation-Free Directional Filter Bank", 2007, IEEE International Conference on Signal Processing and Communications ,pages: 1059 – 1062.

[23] [http: dip] "Introduction to Digital Image Processing" available at http://www.dspguide .com /ch14 .htm

[24] [http: dip1] "Basics of Digital Image Processing" available at http://en.wikipedia.org /wiki/Digital image processing.

[25] [http: image formats] "Image Basics" available at http://www. imgfsr.com/ ifsr_ism.html

[26] [http: image noise] "Image Noise" available at http://www.sprawls.org/ppmi/NOISE

[27] [http: noise characteristics] "Noise Characterization" available at http://www.cis.rit .edu/research/thesis/bs/1999/jobes/Thesis.html#Noise%20Characterization

[28] [http: SNR] "Signal to Noise Ratio" available at http://www.bcae1.com/sig2nois.htm http://www.sprawls.org/ppmi2/IMGCHAR/

[29] Zhenghao Shi and KO B. Fung "A Comparison of Digital Speckle Filters" 1994, Canada Centre for Remote Sensing, pages: 2129-2133.

[30] Guy Gilboa, "Image Enhancement and Denoising by Complex Diffusion Processes", 2004 IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 26, No. 8, pages: 1020-1036.

[31] Ari Nieminen, Pekka Heinonen, Yrjo Neuvo, "A New Class Of Detail-Preserving Filters For Image Processing" , 1987, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. Pami-9, No. 1, pages: 74-96.