Evaluation of Gabor filter parameters for image enhancement and segmentation

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

Master of Engineering

in

Electronic Instrumentation and Control Engineering

to

THAPAR UNIVERSITY

PATIALA (PUNJAB)-147004

Submitted by

ABHAY KRISHAN

Regn.No-80751001

Under the esteemed guidance of

Mr. M. D. Singh

Sr. Lecturer

Department of Electrical and Instrumentation Engineering

THAPAR UNIVERSITY

PATIALA (PUNJAB)-147004

July-2009

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled **"Evaluation of Gabor Filter parameters for noise enhancement and segmentation"** in partial fulfillment of the requirements for the award of the degree of **Master of Engineering** in **Electronics and Instrumentation Control** submitted in Electrical & Instrumentation Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Mr. M. D. SINGH**, **Sr. Lecturer,** EIED, Thapar University, Patiala.

The matter presented in this thesis has not been submitted to any other degree of this or any other university.

Abhay

ABHAY KRISHAN

Roll No. 80751001

It is certified that the above statement made by the candidate is correct and true to the best of our knowledge and belief.

Mr. M.D.SINGH

Sr. Lecturer,

EIED,

Thapar University

Patiala

S. Ghosh 15/7/09

Countersigned By:

R.K.Sharma 21/7

Dr. Smarajit Ghosh
Head of Department
EIED, Thapar University
Patiala

Dr. R.K. Sharma
Dean (Academic Affairs)
Thapar University
Patiala

13/8/2009
(Dr D. Baljit Singh)
External Examiner

ACKNOWLEDGEMENT

Words are often too less to reveals ones deep regards. An understanding of the work like this is never the outcome of the efforts of a single person. I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis. First of all I would like to thank the **all mighty god,** one who has always guided me to work on the right path of the life. Without His grace this would never come to be todays reality. This work would not have been possible without the encouragement and able guidance of my supervisor, **Mr. M.D. Singh, Sr. Lecturer,** Electrical & Instrumentation Engineering Department, for their enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. Their feedback and editorial comments were also invaluable for the writing of this thesis.

No words of thanks are enough for my **dear parents** whose support and care makes me stay on earth. Thanks to be with me.

At the end, I would like to thank all the faculty members of the department and my friends who directly or indirectly helped me in completion of my thesis.

ABHAY KRISHAN

(Roll No. 80751001)

ABSTRACT

This work provides the information about the working of Gabor filters for different types of images. This comparative study is based on the output of the noisy and the filtered images using Gabor filter. Three types of Noises: Gaussian, Poisson and Speckle are used to produces Noisy images from noise free image for the purpose of evaluation. Frequency of Gabor filter and variance of Gabor filter is evaluated for approx. 200 images. These parameters are characterized on the basis of some standard image quality matrices like SNR, correlation coefficient and SSIM. The performance of Gabor filter is also evaluated by segmentation 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 4

MATERIALSAND METHODS

Chapter 5

Chapter 6

Chapter 7

Chapter 8

LIST OF ABBREVIATIONS

SNR	Signal to noise ratio
SSIM	Structure similarity
MSE	Mean square error
VAR	Variance
1-D	One-dimensional
2-D	Two-dimensional
JPG	Joint photographic experts group
TIFF	Tagged information file format
FFT	Fast Fourier Transform
Freq_gab	Frequency of Gabor filters
GA	Genetic Algorithm

LIST OF FIGURES

CHAPTER-1
INTRODUCTION

This thesis work is carried out to study the analysis of Gabor filtered output images and the noisy images. For this analysis, we are testing the values of various quality parameters like SNR, Correlation and SSIM. For getting the noisy images, we are using three types of noises, Gaussian, Poisson and Speckle noises. For segmentation purposes, we are selecting a particular area of the original images and this area is different for all the images. For segmentation analysis, we are using the comparison of three output images as: Original image segmentation, Gabor filtered output segmentation and Noised output segmentation. These three output images are having the same dimensions for their comparison.

The various parameters of the Gabor filter play a major role in deciding the output image. The size, phase, orientation and frequency of the output image are selected by the Gabor filter. The image features are measured by employing an appropriate Gabor filter with adaptively chosen size, orientation, frequency and phase for each pixel. An image property called phase divergence is used for the selection of the appropriate filter size. Characteristic features related to the change in brightness, texture and position are extracted for each pixel at the selected size of the filter.

2-D Gabor filter is easier to tune the direction and radial frequency band-width, and easier to tune center frequency, so they can simultaneously get the best resolution in spatial domain and frequency domain. Gabor filter outputs can be modeled as Gaussian’s and develop algorithm for selecting optimal filter parameters.

Organization of thesis is as follows:

This thesis is organized as per the following format.

Chapter-1 **“Introduction”** it includes the background information and overview of this thesis report.

Chapter 2 **“Literature Survey”** involves the description of the entire literature survey that has been done during the present study.

Chapter 3 **“Gabor Filters”** gives a complete idea about the fundamentals of Gabor filters. It also contains the details of the various parameters upon which the working of the Gabor filters depends. It also contains the details of the various types of Gabor filters.

Chapter 4 **“Materials and Methods”** contains the details of all the ten images, the details of all the noises and the details of all the working parameters that we are using in this Thesis work. The working algorithm that has been followed to do the thesis work is also contained in this chapter.

Chapter 5 **“Experimentation Results”** contains the results obtained (in the form of both tables and graphs) for both the noised and the filtered images. SNR, correlation and SSIM are the working parameters that have been taken as the basis for the discussion of results.

Chapter 6 **“Results Summary”** summarized the results that we have obtained from the thesis work in the form of charts.

Chapter 7 **“Discussions of Results”** discusses the best experimental results (both in the forms of charts and tables) for all the ten images for all the three noises.

Chapter 8 **“Conclusions and Future Scope”** concludes this thesis work. This chapter also gives a brief outline of the future scope of the proposed work.

CHAPTER-2
LITERATURE SURVEY

This literature survey deals with the study of various research papers that contributed in the image segmentation using Gabor filters.

There is an increasing demand of image segmentation in various application areas that can be categorized as: edge detection, statistical approach, histogram thresholding, region based methods such as thresholding, region growing, region splitting and merging, color perceptron model, characteristic feature clustering, neural network, physics based technique and fuzzy set theory.

Gabor filters can serve as excellent band-pass filters for one-dimensional signals. A Gabor filter is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function. Gabor filters is having various transforms, image properties, operators, frequencies and various features which are used in detecting image segmentation.

Z-Q Liu, R. M. Rangayyan, C. B. Frank [1] in 1990 proposed the given image is preprocessed by a sequence of Gabor filters. Linear segments at specific orientations are detected by Gabor filters tuned to the corresponding directions. A new directional analysis method using the Gabor filter, optimal in the sense that the product of the effective widths in the spatial and frequency domain reaches the fundamental lower bound. Gabor filters are particularly sensitive to the presence of collinear and elongated segments.

Jiang Wen, You Zhisheng, Li Hui [2] in 1994 segments the Metallograph images using Gabor Filter. They are having the characteristic of certain cells in the visual cortex of some mammals. Two-dimensional Gabor filters are used to extract feature at every at every pixel of the input image. Segmentation demonstrates that a multi-channel filtering technique can be used to successfully characterize the metallic texture and solve the segmentation problem. By adding the gray level of pixels of smoothed image as one additional feature, we separate the untextured regions from textured regions in the same images.

Richard Buse, Zhi-Qiang Liu [3] in 1994 analyzed for handwritten words in grey-scale images using Gabor filters to extract features from the words. They used set of *n* Gabor filters to extract these oriented parts from the grey scale word image. The filtered outputs are then processed to obtain binary images of the extracted parts which are used for the calculation of unary and binary features. Parts are extracted from the word image by using the responses from a bank of Gabor filters.

Andreas Teuner, Olaf Pichler, Bedrich J. Hosticka [4] in 1995 segmented for images used for the filter selection approach were digitized to 256 grey-levels. For discrimination of image features of high importance with respect to the entire image due to their low redundancy. This is based on the analysis of a spectral feature contrast matrix that yields the parameters for a set of dyadic tuned Gabor filters that perform an efficient segmentationof the analyzed image.

Shuzo Yamamoto, Yoshikazu Nakajima, Shinichi Tamura, Yoshinobu Sato, Seiyo Harino [5] in 1999 generated spatial-temporal images of the fluorescent dots in a capillary and applied Gabor filters tuned to the direction of the traces in order to detect them. Gabor filter matched to a line segment with a certain length was applied to the image, after which the filter output was integrated by the connectivity of the trace region. Shorter Gabor filters can be adapted to space variant intensity images and curved line traces. The longer Gabor filter loses such adaptability, if the image has a space invariant gray level characteristic and almost straight line traces, the long Gabor filter is to be preferred. Image quality will be improved by development of better imaging devices or staining methods. It may be better to detect the capillary position by applying a three-dimensional (3-D) Gabor filter with a moderate length matching the curvature of the capillary to a 3-D spatio-temporal image composed of a stacked image sequence.

Ian R Fasel, Marian S Barlett, Javier R Movellan [6] in 2002 presented a systematic analysis of Gabor filters banks for detection of facial landmarks. With a single frequency band, they obtained performances significantly better than those achievable with current systems that use multiple frequency bands. Best performance may be achieved by concentrating a large number of orientations on very low frequency carriers. Better results may be obtained in the feature finding stage by using only a few filters in the lower end of the frequency spectrum.

C.Klimanee, DT Nguyen [7] in 2004 used an array of 8x4 two-dimensional Gabor filters tuned to eight directions and four ridge frequencies. When local frequency and local orientation of a pattern of parallel lines are reasonably well defined, a directional band pass filter tuned to the parallel frequency and perpendicular orientation, can be used to remove noise and artifacts that interfere with the regular pattern. The merging of two ridges can result in a ridge ending to become a ridge bifurcation and the bridging of two valleys endings can result in the ridge bifurcation to disappear together. Use a bank of only eight directional filters equally spaced at 22.5^0 to decompose a fingerprint image into eight directional images. The fingerprint image is divided into non-overlapping and each block is then replaced by its best counterpart chosen among the eight directional images.

Rong Lu, Yi Shen [8] in 2005 proposed image segmentation algorithm based on random neural network and Gabor filtering technique. This uses Gabor functions of different frequencies and orientations for feature extraction. To overcome time consuming problem of feature extraction, a quartered segmentation strategy is introduced using Gabor filters. This can segment images successfully.

Ying-Chun Li, Zhan-Chun Li, Yun-Huan Mei, Jian-Xin Zhang [9] in 2005 devised a more rigorously Gabor based method enhancing the edge information for designing Gabor filters. The Gabor wavelet representation captures salient visual properties such as spatial localization, orientation selectivity, and spatial frequency characteristic. Two dimensional Gabor wavelet representations and the elastic graph matching has shown that the Gabor wavelet representation is optimal for classifying facial actions. After processed by the Gabor amplitude filter, the particles in image can be segmented automatically. Traditional segmentation techniques such as threshold and region going are based on gray –level of segmented regions, work well for images with measurable homogeneous properties. A threshold is determined from the average intensity of high gradient pixels in the obtained intensity image. They can't cover long shape cast in the closed boundaries produced; they should generate much more accurately bounded regions of objects according the size of the objects. This is a feasible approach for segmenting and detecting particles in microscopic images.

Hany Ayad Bastawrous, Takuya Fukumoto, Norihisa Nitta, Masaru Tsudagawa [10] in 2005 detected ground glass opacities (GGO) in Lung CT Images Using Gabor Filters. The main purpose of the preprocessing stage is to emphasize the gray level intensity of the GGO nodules by performing a threshold process followed by intensity inversion and low pass filtering. The main purpose of applying Gabor filter is to enhance the detection of the nodules by emphasizing their frequency components and rejecting other components. After applying Gabor filter, they performed a threshold process followed by labeling in order to extract all connected objects with high intensity values inside the lung area. They used Gabor filters to emphasize the frequency components of the nodules and also reshape their intensity distribution to appear more Gaussian-like.

Hong Wei, Marc Bartels [11] in 2006 projecting the 2.5 D data onto a surface, obtain a texture map as a grey-level image. On image, Gabor wavelet filters are applied to generate Gabor wavelet features. Most buildings and high vegetation can be detected. Gabor wavelet transform can partially remove hill or slope effects in the original data by tuned Gabor parameters. Apply Gabor wavelets to airborne laser scanned LIDAR data to generate their Gabor wavelet representation, which is then grouped into small windows as segmentation units. Large flat roof buildings may be classified as a flat field. This can segment flat fields accurately with very low false negative.

K. L. Mak, P. Peng [12] in 2007 developed an automated inspection system for textile fabrics based on Gabor filters. The Gabor filters are designed on the basis of the texture features extracted optimally from a non-defective fabric image by using a Gabor wavelet network (GWN). This scheme consists of two real-valued Gabor filters and one smoothing filter. GWN with only one wavelet in the hidden layer has been used as the major technique to extract optimally the basic texture features from a non-defective fabric image. Set of fabric images include different types, sizes, and shapes of detects, and different texture backgrounds.

Mohammed Al-Rawi, Jie Yang [13] in 2008 derives Gabor feature matrices that are used to recognize color texture independent of illumination changes. Set of five color textures have been used to test the performance of the Gabor energy matrices for texture classification. Each of the five textures was imaged under four illuminations, white, red, green and blue. Three sales and four orientations are used for Gabor filter to obtain the filtered image.

Jesmin F.khan, Reza R. Adhani, Sharif M.A. Bhuiyan [14] in 2008 grouped the pixels into clusters based on color, texture, brightness and position features extracted by using a Gabor filter customized for each pixel. A novel selection technique called phase divergence, which is to select the controlling size of the filter window. In addition, the Schwarz criterion is employed to determine the appropriate number of clusters suitable for the image.

Xin Pan, Qiu-Qi Ruan [15] in 2008 proposed to extract local invariant features using Gabor function, to handle the variations of rotation, translation and illumination, raised by the capturing device and the palm structure. Gabor features, derived from the convolution of a Gabor filter and palmprint images were encoded into hamming code by pixels. This comes from using the relationship between the local layer sub-blocks and upper layer sub-blocks based on Gabor features defined as local relative variance (LRV), to represent palmprint image. This is high efficient for the local invariant features are simple statistical quantities without sophisticated calculation.

J. Bossu, Ch. Gee, G. Jones, F.Truchetet [16] in 2008 uses a spatial method method based on the crop row frequency to detect crop rows in the image, where the crop row frequency varied slightly from the bottom to top. A simple local inhibition function between the filtered image and the inverse vegetation binarized image, a weed map was deduced. A complete approach for crop/weed discrimination from image processing based on wavelet tranforms and compared it to a Gabor filtering.

Asnor Juraiza Ishak, Aini Hussain, Mohd Marzuki Mustafa [17] in 2008 proposes an image analysis technique that utilizes a combination of Gabor wavelet (GW)and gradient field distribution (GFD) techniques to extract a new set of feature vectors based on their directional texture properties for the classification of weed types. This represents an raw image in its reduced compact form to facilitate and speed up the decision-making process for classification. Weed classification involves assigning weeds to grasses and broadleaves categories. Misclassifications were found in images consisting of highly densed grass weed population that caused the leaves to overlap each other and concealed the grass characteristics. Certain broadleaves images might have stems or branches of the weed visible which can potentially cause misclassification of weed as grass. This extraction has great potential for feature vector representation of weed images.

Minqin Wang, Guoqiang Han, Yongqiu Tu, Guohua Chen, Yuefang Gao [18] in 2008 works on image segmentation Based on Gabor Wavelet. The methods solving the problem include (1) using a linear function; (2) applying a Gauss smoothing function to implement spatial smoothing, which cause the difference between similar texture become small, while the difference between different texture become large. The method of extracting texture features by using Gabor filters is suited with character of human eyes. It can get good segmentation effect. This method can keep the accuracy segmentation and fast processing speed simultaneously.

Gholam Ali Rezaei Rad, Kaveh Samiee [19] passing initial image through each filter will result a new filtered image that has significant properties of the original image. Small values of parameters of Gabor filters will cause a faster computation of each filtered image. Because of the nature of the Gabor filters there is no need to equalize and quantize the original image.

From the above review of research papers, it is quite clear that Gabor filter plays a major role for various types of analysis in the image segmentation fields and for various types of techniques in the image processing. But among these papers, we found one of the techniques, Gabor filter wavelet, was developed for various types of working parameters and for image segmentation purposes for some particular images. Hence, we modified and proposed the same technique for images, in this thesis, taking its application for various images as the main basis.

CHAPTER-3

GABOR FILTERS

A **Gabor filter** is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function. Because of the multiplication-convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function.

Gabor filters are directly related to Gabor wavelets, since they can be designed for number of dilations and rotations. However, in general, expansion is not applied for Gabor wavelets, since this requires computation of biorthogonal wavelets, which may be very time-consuming. Therefore, usually, a filter bank consisting of Gabor filters with various scales and rotations is created. The filters are convolved with the signal, resulting in a so-called Gabor space. This process is closely related to processes in the primary visual cortex. The Gabor space is very useful in e.g., image processing applications such as iris recognition and fingerprint recognition. Relations between activations for a specific spatial location are very distinctive between objects in an image. Furthermore, important activations can be extracted from the Gabor space in order to create a sparse object representation.

The Gabor Filters have received considerable attention because the characteristics of certain cells in the visual cortex of some mammals can be approximated by these filters. In addition these filters have been shown to posses optimal localization properties in both spatial and frequency domain and thus are well suited for texture segmentation problems. Gabor filters have been used in many applications, such as texture segmentation, target detection, fractal dimension management, document analysis, edge detection, retina identification, image coding and image representation. A Gabor filter can be viewed as a sinusoidal plane of particular frequency and orientation, modulated by a Gaussian envelope [22].

$$h(x, y) = s(x, y)g(x, y) \tag{1}$$

$s(x, y)$: Complex sinusoid

$g(x, y)$: 2-D Gaussian shaped function, known as envelope

$$s(x, y) = e^{-j2\pi(u_0 x + v_0 y)}$$

$$g(x, y) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)}$$

3.1 THE TEMPORAL (1-D) GABOR FILTER[22]

Gabor filters can serve as excellent band-pass filters for one-dimensional signals (e.g., speech). A complex Gabor filter is defined as the product of a Gaussian kernel times a complex sinusoid, i.e.

$$g(t) = ke^{j\theta} w(at)s(t) \tag{2}$$

Where

$$w(t) = e^{-\pi t^2}$$

$$s(t) = e^{j(2\pi f_0 t)}$$

$$e^{j\theta} s(t) e^{j(2\pi f_0 t + \theta)} = (\sin(2\pi f_0 t + \theta), j\cos(2\pi f_0 t + \theta))$$

Here k, θ, f_0 are filter parameters.

Complex Gabor filter as two out of phase filters continently allocated in the real and complex part of a complex function, the real part holds the filter

$$g_r(t) = w(t)\sin(2\pi f t_0 t + \theta) \tag{3}$$

And the imaginary part holds the filter

$$g_i(t) = w(t)\cos(2\pi f_0 t + \theta) \tag{4}$$

3.1.1 FREQUENCY RESPONSE

Taking the Fourier transform

$$\hat{g}(f) = ke^{j\theta}\int_{-\infty}^{\infty} e^{-j2\pi ft} w(at)s(t)dt = ke^{j\theta}\int_{-\infty}^{\infty} e^{-j2\pi(f-f_0)t} w(at)dt$$
$$= \frac{k}{a} e^{j\theta}\,\hat{w}\left(\frac{f-f_0}{a}\right) \qquad (5)$$

where $\hat{w}(f) = w(f) = e^{-\pi f^2}$

3.1.2 Gabor energy filters

The real and imaginary components of a complex Gabor filters are phase sensitive, i.e. , as a consequence their response to a sinusoid is another sinusoid. By getting the magnitude of the output (square root of the sum of squared real and imaginary outputs) we can get a response that phase sensitive and thus unmodulated positive response to a target sinusoid input. In some cases it is useful to compute the overall output of the two out of phase filters. One common way of doing so is to add the squared output (the energy) of each filter, equivalently we can get the magnitude. This corresponds to the magnitude (more precisely the squared magnitude) of the complex Gabor filter output. In the frequency domain, the magnitude of the response to a particular frequency is simply the magnitude of the complex Fourier transform, i.e. ,

$$\left\|\hat{g}(f)\right\| = \frac{k}{a}\,\hat{w}\left(\frac{f-f_0}{a}\right) \qquad (6)$$

This is a Gaussian function centered at f_0 and with width proportional to a .

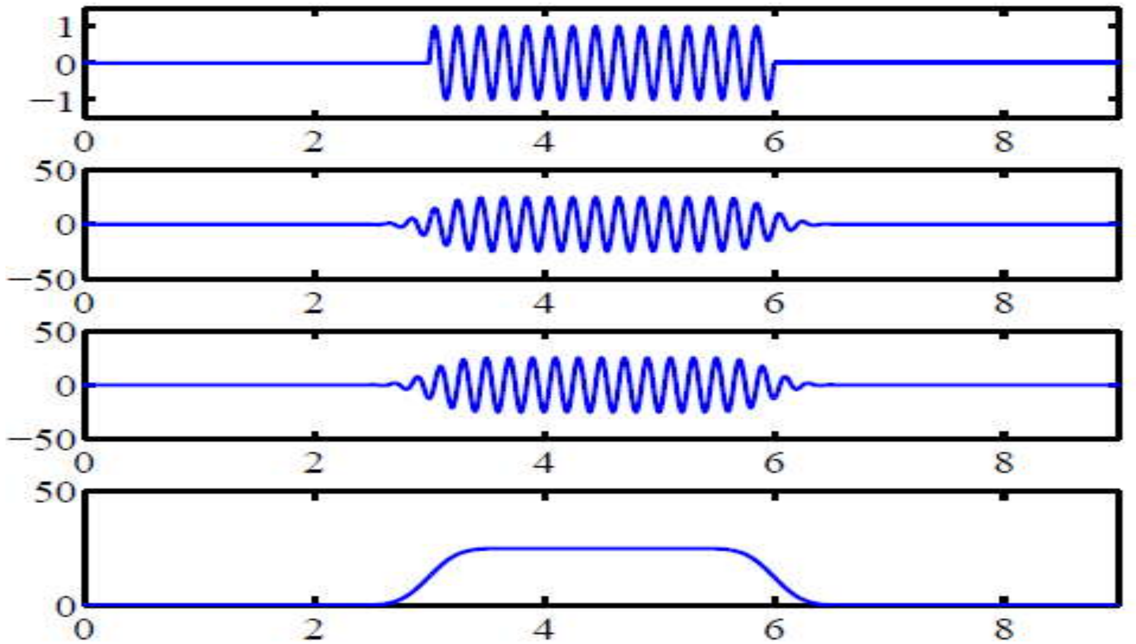


Fig. 3.1

Top: An input signal.

Second: Output of Gabor filter (cosine carrier).

Third: Output of Gabor Filter in quadrature (sine carrier);

Fourth: Output of Gabor Energy Filter

3.1.3 Bandwidth and Peak Response

The peak filter response is at f_0. To get the half-magnitude bandwidth Δ_f, note

$$\hat{w}\left(\frac{f - f_0}{a}\right) = e^{-\pi \frac{f - f_0}{a^2}} = 0.5 \tag{7}$$

The half peak magnitude is achieved for

$$f - f_0 \pm \sqrt{a^2 \log 2\pi} = 0.4697a \approx 0.5a \tag{8}$$

Thus the half-magnitude bandwidth is ((*2*)(*0.4697*) a) which is approximately equal to a. Thus a can be interpreted as the half- magnitude filter bandwidth.

1.1.4 Eliminating the DC response

Depending on the value of f_0 and a, the filter may have a large DC response. A popular approach to get a zero DC response is to subtract the output of a low-pass Gaussian filter,

$$h(t) = g(t) - cw(bt) = ke^{j\theta} w(at)s(t) - cw(bt) \tag{9}$$

Thus

$$\hat{h}(f) = \hat{g}(f) - \frac{c}{b}\hat{w}(\frac{f}{b}) \tag{10}$$

To get a zero DC response we need

$$\frac{c}{b}\hat{w}(0) = \hat{g}(0) \tag{11}$$

$$c = b\hat{g}(0) = b\frac{k}{a}e^{j\theta}\hat{w}(\frac{f_0}{a}) \tag{12}$$

where we used the fact that, $\hat{w}(f_0) = \hat{w}(-f_0)$ Thus,

$$h(t) = g(t) - b\,\hat{g}(0) = ke^{j\theta}(w(at)s(t) - \frac{b}{a}\hat{w}(\frac{f_0}{a})w(bt)) \tag{13}$$

$$\hat{h}(f) = \frac{k}{a}e^{j\theta}(\hat{w}(\frac{f-f_0}{a}) - \hat{w}(\frac{f_0}{a})\,\hat{w}(\frac{f}{a}) \tag{14}$$

It is convenient, to let *b*=*a*, in which

$$h(t) = ke^{j\theta}w(at)(s(t) - \hat{w}(\frac{f_0}{a}) \tag{15}$$

$$h(f) = \frac{k}{a}e^{j\theta}(\hat{w}(\frac{f-f_0}{a}) - \hat{w}(\frac{f_0}{a})\,\hat{w}(\frac{f}{a})) \tag{16}$$

3.2 THE SPATIAL (2-D) GABOR FILTER[22]

Here is the formula of a complex Gabor function in space domain

$$g(x, y) = s(x, y)w_r(x, y) \quad (17)$$

where $s(x, y)$ is a complex sinusoid, known as the **carrier**, and $w_r(x, y)$ is a 2-D Gaussian-shaped function, known as the **envelope**.

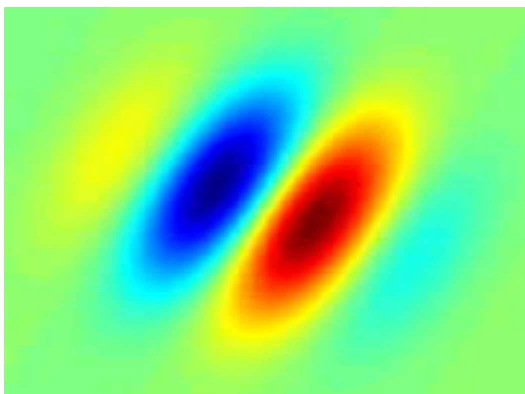


Fig.3.2: Example of 2-D Gabor Filter

3.2.1 The complex sinusoid carrier

The complex sinusoid is defined as follows,

$$s(x, y) = \exp(j(2\pi(u_0 x + v_0 y) + P)) \quad (18)$$

where (u_0, v_0) and P define the spatial frequency and the phase of the sinusoid respectively.

This sinusoid can be represented as two separate real functions, conventionally allocated in the real and imaginary part of a complex function.

The real part and imaginary part of this sinusoid are

$$\mathrm{Re}(s(x,y)) = \cos(2\pi(u_0 x + v_0 y) + P) \tag{19}$$

$$\mathrm{Im}(s(x,y)) = \sin(2\pi(u_0 x + v_0 y) + P) \tag{20}$$

The parameters u_0 and v_0 define the spatial frequency of the sinusoid in Cartesian coordinates. This spatial frequency can also be expressed in polar coordinates as magnitude F_0 and direction ω_0:

i.e.

$$F_0 = \sqrt{u_0^2 + v_0^2}$$
$$\omega_0 = \tan^{-1}\left(\frac{v_0}{u_0}\right)$$
$$u_0 = F_0 \cos\omega_0$$
$$v_0 = F_0 \sin\omega_0$$

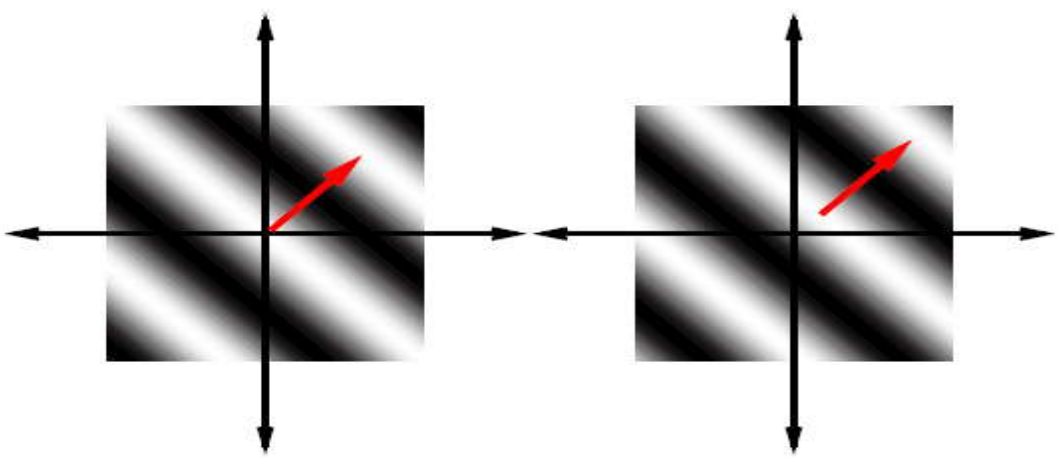


Fig. 3.3: The real and imaginary parts of a complex sinusoidal

Using this representation, the complex sinusoid is

$$s(x,y) = \exp(j(2\pi F_0(x\cos\omega_0 + y\sin\omega_0) + P)) \tag{21}$$

3.2.2 The Gaussian envelope

The Gaussian envelope looks as follows:

$$\omega_r(x,y) = K\exp(-\pi(a^2(x-x_0)_r^2 + b^2(y-y_0)_r^2)) \tag{22}$$

where (x_0, y_0) is the peak of the function , a and b are scaling parameters of the Gaussian, and the $_r$ subscript stands for a rotation operation such that

$$(x-x_0)_r = (x-x_0)\cos\theta + (y-y_0)\sin\theta$$
$$(y-y_0)_r = -(x-x_0)\sin\theta + (y-y_0)\cos\theta$$

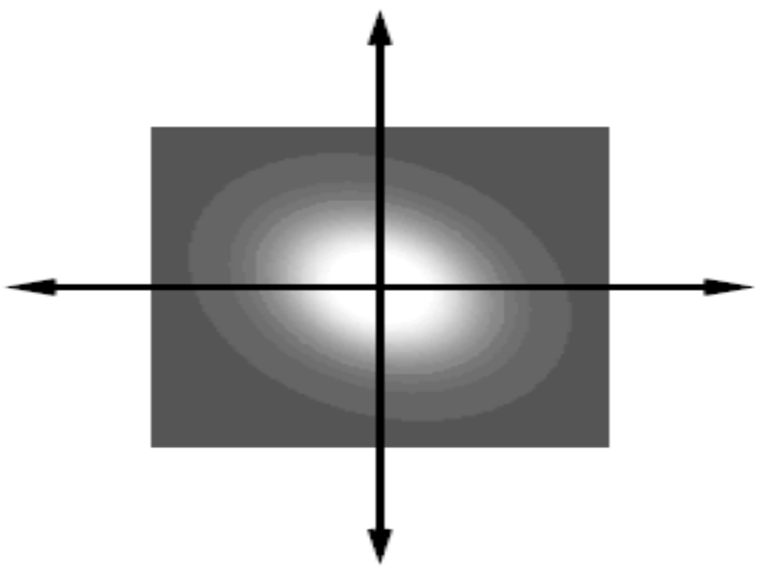


Fig. 3.4: A Gaussian envelope

3.2.3 The complex Gabor function

The complex Gabor function is defined by the following 9 parameters;

- K : Scales the magnitude of the Gaussian envelope.

- (a,b) : Scale the two axis of the Gaussian envelope.
- θ : Rotation angle of the Gaussian envelope.
- (x_0, y_0) : Location of the peak of the Gaussian envelope.
- (u_0, v_0) : Spatial frequencies of the sinusoid carrier in Cartesian coordinates. It can also be expressed in polar coordinates as (F_0, ω_0).
- P : Phase of the sinusoid carrier.

Each complex Gabor consists of two functions in quadrature (out of phase by 90 degrees), conveniently located in the real and imaginary parts of a complex function.

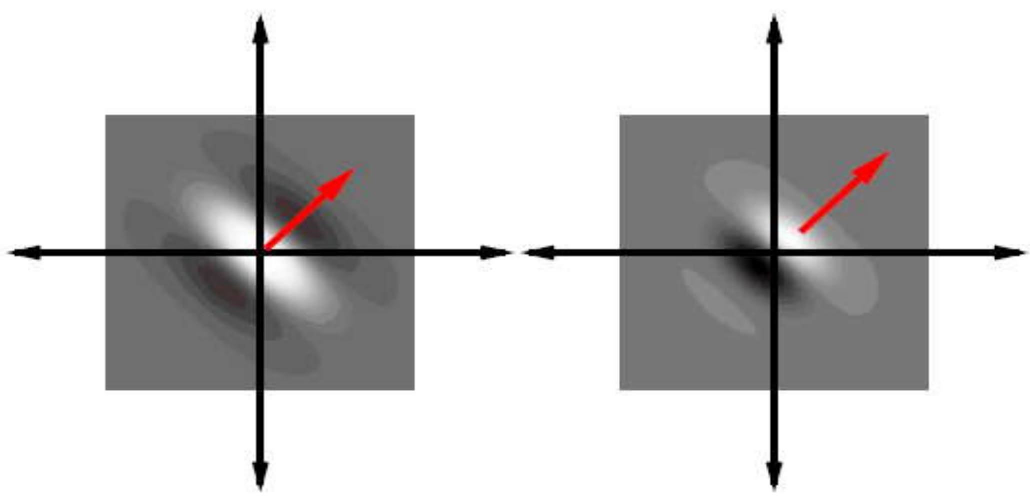


Fig. 3.5
The real and imaginary parts of a complex Gabor function in space domain.

Now we have the complex Gabor function in space domain:

$$g(x,y) = K\exp(-\pi(a^2(x-x_0)_r^2 + b^2(y-y_0)_r^2))\exp(j(2\pi(u_0x+v_0y)+P)) \quad (23)$$

Or in polar coordinates,

$$g(x,y) = K\exp(-\pi(a^2(x-x_0)_r^2 + b^2(y-y_0)_r^2))\exp(j(2\pi F_0(x\cos\omega_0 + y\sin\omega_0)+P)) \quad (24)$$

The 2-D Fourier transform of this Gabor is as follows:

$$\hat{g}(u,v) = \frac{k}{ab}\exp(j(-2\pi(x_0(u-u_0)+y_0(v-v_0))+P))\exp(-\pi(\frac{(u-u_0)_r^2}{a^2}+\frac{(v-v_0)_r^2}{b^2})) \quad (25)$$

Or in polar coordinates,

$$\text{Magnitude}(\hat{g}(u,v)) = \frac{k}{ab}\exp(-\pi(\frac{(u-u_0)_r^2}{a^2}+\frac{(v-v_0)_r^2}{b^2})) \quad (26)$$

$$\text{Phase}(\hat{g}(u,v)) = -2\pi(x_0(u-u_0)+y_0(v-v_0))+P \quad (27)$$

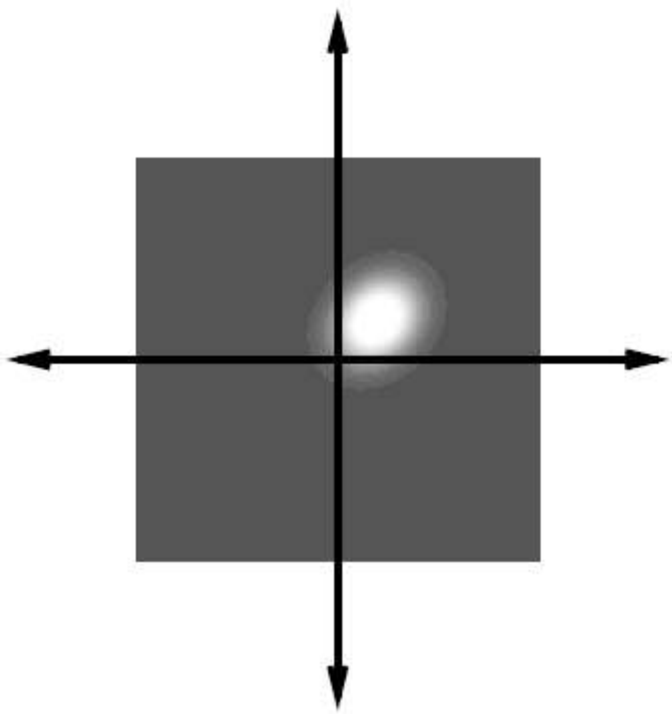


Fig. 3.6: The Fourier transform of the Gabor filter

3.2.4 Half-magnitude profile

The region of points, in frequency domain, with magnitude equal one- half the peak magnitude can be obtained as follows. Since the peak value is obtained for

$(u,v)=(u_0,v_0)$, and the peak magnitude is K/ab, we just need to find the set of points (u,v) with magnitude *K/2ab*.

$$\frac{1}{2}\frac{K}{ab}=\frac{K}{ab}\exp(-\pi(\frac{(u-u_0)_r^2}{a^2}+\frac{(v-v_0)_r^2}{b^2})) \tag{28}$$

$$or, -\log 2=-\pi(\frac{(u-u_0)_r^2}{a^2}+\frac{(v-v_0)_r^2}{b^2}) \tag{29}$$

$$(\frac{(u-u_0)_r}{aC})^2+(\frac{(v-v_0)_r}{bC})^2=1 \tag{30}$$

This is an equation of ellipse centered at (u_0,v_0) rotated with an angle θ with respect to the *u* axis.

3.2.5 Half-magnitude frequency and orientations bandwidths

Frequency and orientations bandwidths of neurons are commonly measured in of the half- magnitude responses. Let u_0, v_0 the preferred spatial frequency of a neuron. In polar coordinates this spatial frequency can be expressed as F_0 and ω_0.

To find the half-magnitude frequency bandwidth, we probe the neuron with sinusoid images of orientation ω_0 and different spatial frequency magnitudes *F*. We increase *F* with respect to F_0 until the magnitude of the response is half the magnitude at (F_0, ω_0). Let's call that value $F_{\max}$. We then decrease *F* with respect to F_0 until the magnitude of the response is half the magnitude at (F_0, ω_0). Call that $F_{\min}$. Half-magnitude frequency bandwidth is defined as follows:

$$\Delta F_{\frac{1}{2}}=F_{\max}-F_{\min} \tag{31}$$

Or, when measured in octaves,

$$\Delta F_{\frac{1}{2}} = \log_2 (F_{max} / F_{min}) \qquad (32)$$

Half-magnitude orientation bandwidth is obtained following the same procedure but playing with the orientation ω instead of the frequency magnitude F.

$$\Delta \omega_{\frac{1}{2}} = \omega_{max} - \omega_{min} \qquad (33)$$

In Gabor functions with $\theta_0 \approx \omega_0$ the frequency bandwidth can be obtained as follows

$$\Delta F_{\frac{1}{2}} = 2aC \approx a \qquad (34)$$

And the orientation bandwidth can be approximated as follows

$$\Delta \omega \approx 2 \tan^{-1}(\frac{bC}{F_0}) \qquad (35)$$

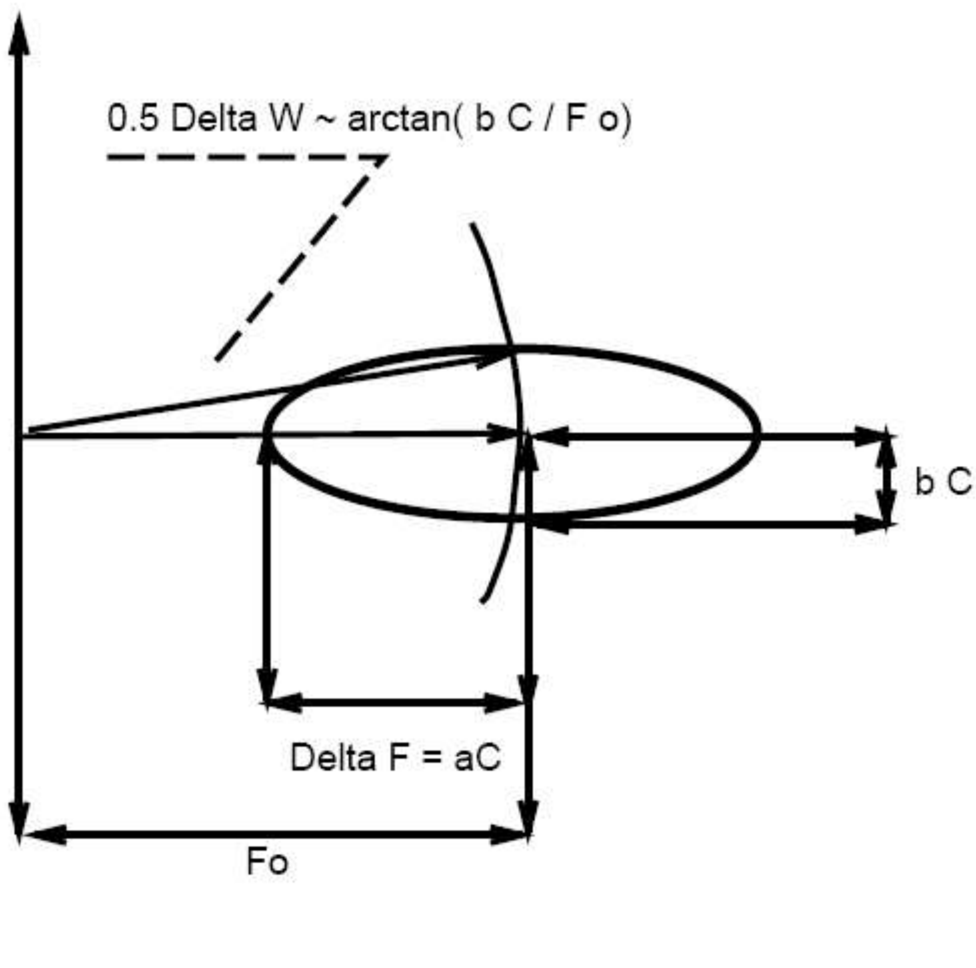


Fig.3.7: A half-magnitude profile and its relationship to the orientation and frequency bandwidths.

3.2.6 Gabor functions for spatial frequency filtering

Consider a massive set of simple cell neurons with Gabor kernel functions with equal parameters except for the location parameters (x_0, y_0).Let all these neurons be distributed uniformly about the foveal field. Each point in the foveal field contains at least two neurons in quadrarture. We can model the operation of such a set of neurons as a convolution operation (assuming a continuous and uniform distribution of filters in all the foveal locations). Since convolution in space domain is product in frequency domain, the set of Gabor functions work as bandpass frequency filters of the foveal image. The peak frequency is controlled by the rotation θ and scale parameters a, b of the Gaussian envelope.

3.2.7 Energy filtering

A quadrature pair (or a Hilbert transform) is a set of two linear operators with the same amplitude response but phase responses shifted by 90 degrees. Strictly speaking sine and cosine Gabor operators are not quadrature pairs because cosine phase Gabors have some DC response, whereas sine Gabors do not. However, one can have quadrature Gabor pairs that looks very much like sine/cosine pairs. Thus the sine and cosine Gabor pair is commonly referred to as a quadrature pair.

A system that sums the square of the outputs of a quadrature pair is called an energy mechanism. Energy mechanisms have unmodulated responses to drifting sinusoids.

Simple cells respond to a drifting sinusoid with a half wave rectified analog to the signal, suggesting that the cells are linear up to rectification. Complex cells respond to a drifting sinusoid in an unmodulated way, as a maintained discharge. The subunits of model complex cells are model simple cells with identical amplitude response.

3.2.8 Functional Interpretations

Section in preparation:

- Minimizes number of neurons needed to achieve a desired frequency resolution.
- Spatially and frequency localized.
- Matched to “logons” likely to occur in images.
- For natural images the Gabor representation is sparser than the pixel representation and than the DOG representation.

3.3 TEMPORAL AND SPATIAL GABOR EXAMPLE [20]

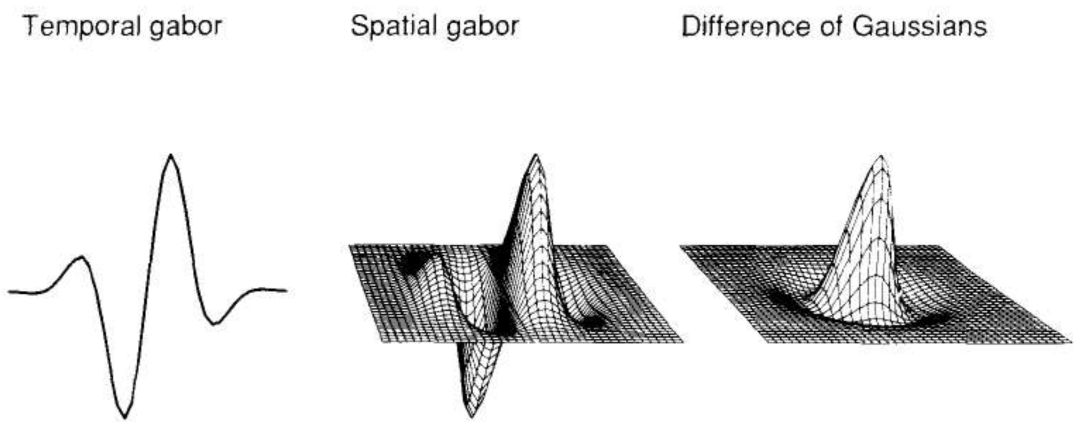


Fig. 3.8

The Temporal Gabor Filter shown here is based on a sinusoid with a period of 56*msec* modified by a Gaussian envelope with a Standard Deviation of 22.4*msec* (Resolution= 4*msec*). The phase difference between the sinusoid and the Gaussian is 0. The spatial Gabor filter is based on a sinusoidal plane wave with a period of 2.8*mm* and a 2-D Gaussian with a Standard Deviation 1.12*mm* (resolution=0.2*mm*, phase=0). The difference of Gaussians filters represents the difference between a pair of 2-D Gaussians with Standard Deviations of 0.6*mm* and 0.9*mm*. In this case, the narrower Gaussian was positive and the broader was negative, so the filter has an “on-center” shape.

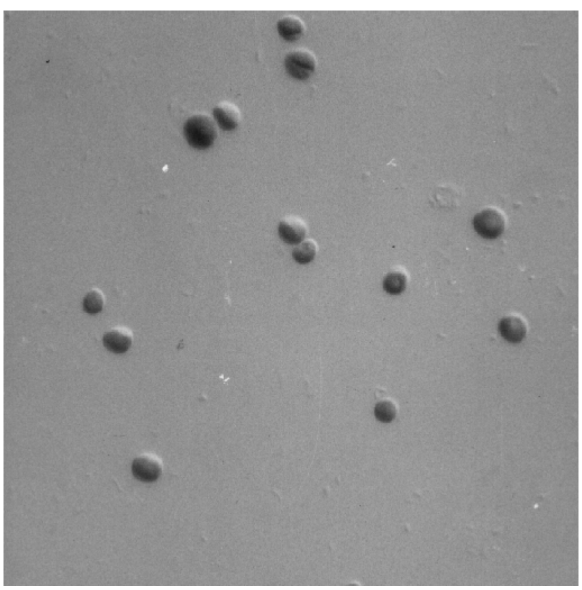
CHAPTER-4
MATERIALS AND METHODS

4.1. IMAGES AS INPUTS

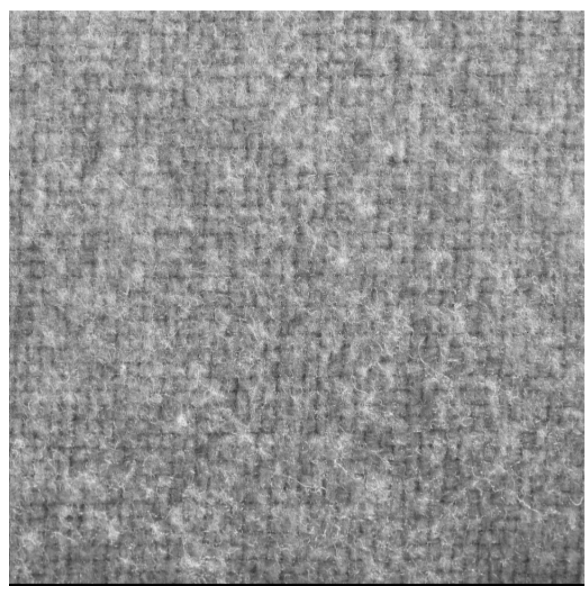
Gabor filters needs some types of images as the input. These images require the process of computer algorithms as per the input image. These computer algorithms yield two types of images from Computer Algorithm: noisy image and magnitude image. The magnitude image is comparing with the noisy image, which gives the advantages of Gabor filters in various parameters.

4.1.1 Input Images

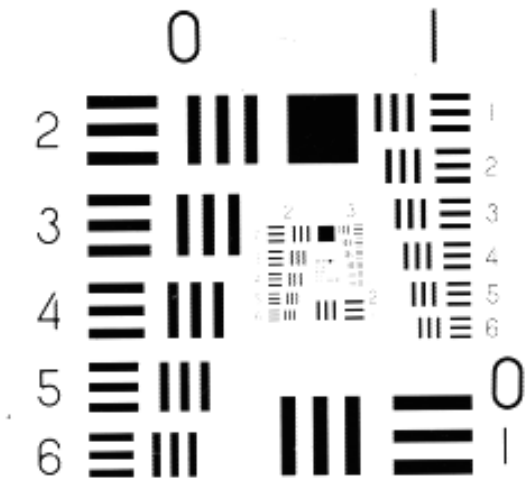
Some of the original test images:-



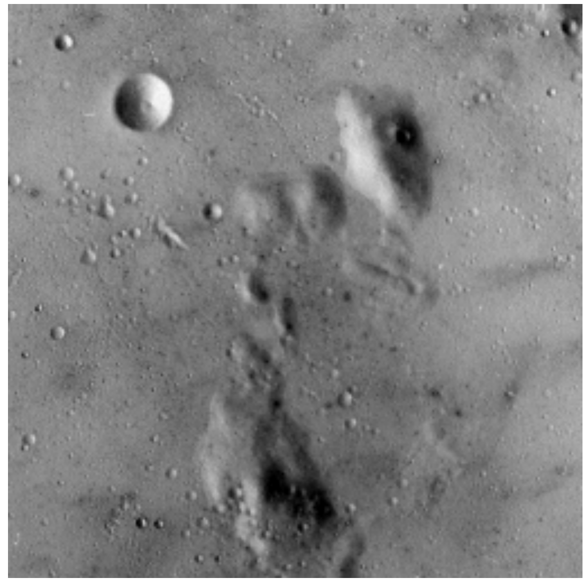
ALGAE.tif



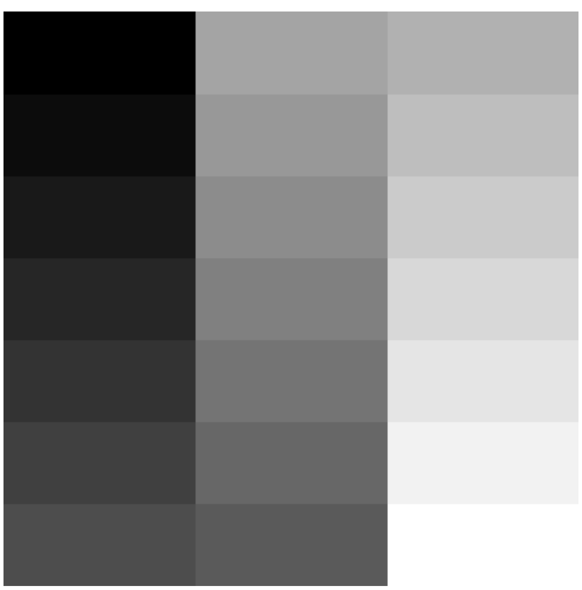
BLANKET.tif



COUNTING.tiff



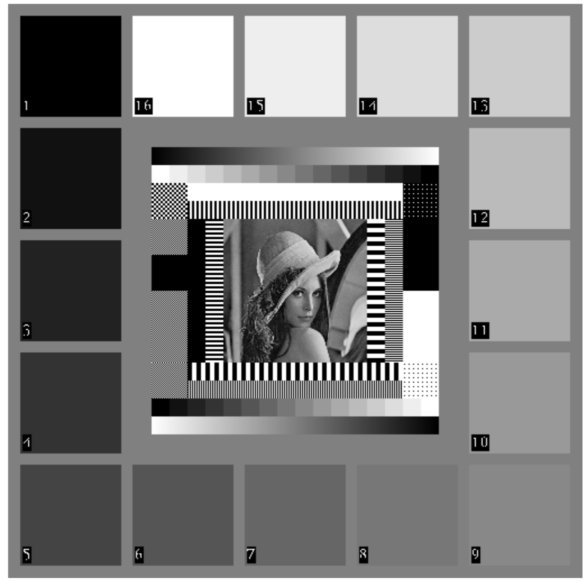
MOON.tiff



GRAY.tiff

JUNGLE.tiff

LENA2D.JPG



TESTPAT.tiff

TOPVIEW.tiff

ULTRASONIC.tif

4.1.2 Size of images

There are two sizes of images that we are using, that are (512 X 512) inches and (256 X 256) inches. That are summarized as:-

ALGAE.tif, BLANKET.tif, GRAY.tiff, JUNGLE.tiff, LENA.jpg, TESTPAT.tiff are having the size of (512 X 512) inches.

COUNTING.tiff, MOON.tiff, TOPVIEW.tiff, ULTRASONIC.tif are having the size of (256 X 256) inches.

4.1.3 Image file formats

A Variety of image file formats are available at present. Like TIFF, JPEG, GIF, BMP, etc. We mainly using TIFF and JPEG in our work, these are explained as follows:

- **TIFF**- stands for Tagged Image File Format. Its extension is recognized both as 'tif' and 'tiff'. These are the file formats used for storing images, including photographs and line art. It grew to accommodate grayscale images, then color images. Today, it is a popular format for high-color-depth images, along with JPEG.
- **JPEG**- stands for Joint Photographic Experts Group. It has '.jpg', 'jpeg' as the allowed extensions. It is the most common format for storing and transmitting photographic images on the World Wide Web and is a commonly used method of compression for photographic images.

4.1.4 Images types

Four types of images are there:

- **Intensity images** – An intensity image is a data matrix whose values represent intensities within some range. For the elements of class uint8 or class uint16 of an intensity image, the integer values lie between [0,255] and [0, 65535], respectively. And if the image is of class double, then the associated values are floating-point numbers. Conventionally, the intensity images with scaled, class double data type have a range of [0, 1]. In MATLAB, an intensity image is stored as a single matrix, with each element of the matrix corresponding to one image pixel.

- **Binary images** - A binary image is a logical array of 0s and 1s. Pixels with the value 0 are displayed as black; pixels with the value 1 are displayed as white. In MATLAB, a binary image must be of class logical that is why the intensity images that happen to contain only 0's and 1's are not taken as binary images.
- **Indexed images** – An indexed image consists of a data matrix, X, and a colormap matrix termed as "map". The "map" is an m-by-3 array of class double containing floating-point values in the range [0, 1]. Its every row specifies the red, green, and blue components of a single color. For these images pixel values are directly mapped to their corresponding colormap values. The color of each image pixel is determined by using the corresponding values of X as an index into map. The value 1 points to the first row in map, the value 2 points to the second row, and so on.
- **RGB images** – An RGB image is also referred as a true- color image. In MATLAB these images are stored in the form of an m-by-n-by-3 data array that defines red, green, and blue components for each individual pixel. 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 RGB images as 24-bit images, where the red, green and blue components are 8 bits each. An RGB array can be of class double, uint8, or uint16. In an RGB array of class double, each color component is a value between 0 and 1. A pixel whose color components are (0,0,0) is displayed as black, and a pixel whose color components are (1,1,1) is displayed as white. The three color components for each pixel are stored along the third dimension of the data array.

4.2. Types of Noises

Noise is considered to be any measurement that is not part of the phenomena of interest. Noise can be generated within the electrical components of the Input amplifier (internal noise), or it can be added to the signal as it travels down the input wires to the amplifier (external noise). There are various types of noises available in MATLAB: 'Gaussian', 'Poisson', 'Speckle', 'localvar', 'Salt & pepper'. Noise is added to the input images to check their various parameters. Out of all these, we are using three noises in our work:

- **Gaussian noise-** This type of noise adds normal distributed noise to the original image. The noise is independent of the image it is applied to. The value of the pixel is altered by the additive Gaussian noise as

  $$J(k,l) = x(k,l) + n$$

  Where n is the noise, $n \sim N(0,v)$, being distributed normally with variance v. the noisy pixels which are generated are anywhere between black and white, distributed according to the Gaussian curve. The width of the curve is adjusted with the mean and the variance parameter. In our work, we are using the default values of the Gaussian noise that are mean=0, variance=0.01 which is quite within the permissible limits of [0, 1].

- **Poisson noise-** Poisson noise is generated from the data instead of adding artificial noise to the data. If I, the original image, is double precision, then input pixel values are interpreted as means of Poisson distributions scaled up by $1e12$. If I is uint8 or uint16, then pixel values are used directly without scaling. Poisson noise generates a noise sequence of integer numbers having a Poisson probability distribution

  $$p(x) = \frac{\mu^x}{x!}.e^{-\mu}$$

- **Speckle noise-** Speckle adds multiplicative noise to the image according to the following formula:

  $$J = I + n * I$$

  where n is an array with the size of an array with the size of the original image, filled with random values resulting from a normal distribution (Gaussian distribution) with mean 0 and are controlled by the variance. With this type of noise, noise generation is dependent on the original image, hence the product in the formula. In dark areas (where values are 0 or close to 0) no noise is generated. Variance is taken to be as default value of 0.04.

4.3. QUALITY METRICS

There are various quality metrics in our work. By evaluating the values of those parameters we can compare the values of the noisy and the magnitude images of the output. The values of those parameters lead to best results as per the details of various values by comparing the noisy and the magnitude output values. These are the various quality metrics:

- **SNR**- SNR stands for signal to noise ratio. SNR is defined as the ratio of the net signal value to the RMS noise. Where the net signal value is the difference between the average signal and background values, and the RMS noise is the standard deviation of the signal value.

$$\text{SNR}=\frac{Signal}{RMSnoise}$$

The net signal is calculated from the difference of the average signal and background values. The RMS or root mean square noise is defined from the signal region.

SNR compares the level of desired signal to the level of background noise. The higher the ratio, the less obtrusive the background noise is. SNR in decibels is defined as:

$$SNR = 10\log(\frac{\sigma_g^2}{\sigma_e^2})$$

Where, σ_g^2 is the variance of the noise free image and σ_e^2 is the variance of error (between the original and the output image).

Brighter regions have a stronger signal due to more light, resulting in higher overall SNR.

- **<u>Correlation</u>**- The operation called correlation is closely related to convolution. In correlation, the value of an output pixel is also computed as a weighted sum of neighboring pixels. The correlation coefficient matrix represents the normalized measure of the strength of linear relationship between variables correlation coefficient. Correlation indicates the strength and direction of linear relationship between 2 signals and its value lie between +1 and -1. The correlation is 1 in the case of an linear relationship, -1 in the case of a decreasing linear relationship and some value in between for all other cases, including the degree of linear dependence between the 2 signals. The closer the coefficient is to either -1 or +1, the stronger the correlation between the signals.

$$COC = \frac{\sum (g-\overline{g})(\hat{g}-\overline{g})}{\sqrt{\sum (g-\overline{g})^2 \sum (\hat{g}-\overline{\hat{g}})^2}}$$

Where g & $\hat{g}$ are the original and noised images and $\overline{g}$ & $\overline{\hat{g}}$ are the means of the original and the noised images.

- **<u>SSIM</u>-** SSIM stands for Structural SIMilarity. The SSIM index is a method for measuring the similarity between two images. The SSIM index is a full reference metric, in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods like <u>PSNR</u> and <u>MSE</u>, which have proved to be inconsistent with human eye perception.

  The SSIM metric is calculated on various windows of an image. The measure between two windows of size N X N *x* and *y* is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\,\mathrm{cov}_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

  with

- μ_x the average of x ;
- μ_y the average of y ;
- σ_x^2 the variance of x ;
- σ_y^2 the variance of y ;
- cov_{xy} the covariance of y ;
- $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ two variables to stabilize the division with weak denominator ;
- L the dynamic range of the pixel-values ;
- $k_1 = 0.01$ and $k_2 = 0.03$ by default.

4.4. Working Algorithm using MATLAB

This working Algorithm is the way of doing our work. This is a step by step procedure that how we implement the images and how we evaluate the working parameters. The main algorithm, followed in order to fulfill the aim of our work, is as follows:

Step 1:

Read the original standard image (ALGAE.tif, BLANKET.tif, COUNTING.tiff, GRAY.tiff, JUNGLE.tiff, LENA.jpg, MOON.tiff, TESTPAT.tiff, TOPVIEW.tiff, ULTRASONIC.tif)

This step is done by using the MATLAB command: imread (‘filename.fmt’);

Step 2:

Apply the Gabor Filter to the original standard image and storing the mainly outputs for the Noisy image and the Magnitude image.

This step is done by using the MATLAB command:

gaborfilter2 (I, S, F, W, P); For Gaussian Noise OR gaborfilter4 (I, S, F, W, P); For Poisson Noise OR

gaborfilter5 (I, S, F, W, P); For Speckle Noise

where I: Original Standard Image

S: Variance

F: Frequency Components

W: Angle Components

P: Phase Components

Step 3:

Resize the output noisy and output magnitude images as per the original standard image size (512 X 512) or (256 X 256) inches as the size of the original image may be. To evaluate various parameters from the original and the output image it is necessary to maintain the same size of the images. If their size is not same the working parameters will not run. To resize the image we used Paint, to get the required size.

Step 4:

Before calculating the values of the various parameters, it is necessary to convert that output image to 1-Dimensional image because that standard image doesn't work by using the parameters formula in their original form.

This step is done by using the MATLAB command:

J=J (:,:,1);

where J is the read standard image file.

Step 5:

Calculate the value of the SNR parameters by using the MATLAB command:

SNR(I,J)

This command gives the values of the SNR, where I is the original image and J is the output image.

Step 6:

Calculate the value of the Correlation parameters by using the MATLAB command:

Correlation(I,J)

This command gives the values of the Correlation, where I is the original image and J is the output image.

Step 7:

Calculate the value of the SSIM parameters by using the MATLAB command:

[mssim ssim_map] = ssim_index(I,J)

This command gives the values of the SSIM parameters, where I is the original image and J is the output image.

Step 8:

After run of all the parameters, all the values of parameters are calculated by changing the Variances and changing the frequency of the Gabor filters. The best value results are collected and plotted all with respect to their particular Variances.

For running the Gabor filters for particular Noises and various parameters, we are using the MATLAB programs. We are total getting twenty results for a particular image, at a particular Variance the frequency changes five times and there are total four Variances taken for a particular image. The best results that we are getting for a particular image are displayed as in the main results chapter.

CHAPTER-5
EXPERIMENTAL RESULTS

This chapter contains the results, obtained after following the Gabor filter Algorithm. The results have been demonstrated in the form of comparison tables. After the comparison tables, a graphical representation has also been done for a quick analysis of results. All the techniques have been tested for all the assumed standard test images.

Image1: ALGAE

a. GAUSSIAN (0,0.01)

Table 5.1

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.486	20.225	0.763	0.8633	0.1156	0.7128
500	4.4518	20.1669	0.7221	0.8561	0.0861	0.7123
700	4.4682	20.1288	0.7571	0.8579	0.1154	0.7138
900	4.4891	20.2421	0.7624	0.86	0.1155	0.7116
1500	4.4752	20.6957	0.765	0.8574	0.1164	0.7148
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4723	21.8606	0.7558	0.7548	0.1149	0.634
500	4.4755	21.176	0.7601	0.7824	0.1147	0.6169
700	4.4801	21.1374	0.7602	0.7795	0.116	0.6213
900	4.4874	20.8774	0.7606	0.7732	0.1152	0.6417
1500	4.4775	21.1344	0.7619	0.7772	0.1164	0.6236
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4599	12.9405	0.7598	0.7214	0.1156	0.4935
500	4.4647	16.6299	0.7613	0.7206	0.1162	0.4799
700	4.4742	16.422	0.7587	0.7174	0.1166	0.4776
900	4.4736	20.4156	0.762	0.7191	0.116	0.4359
1500	4.4762	18.4607	0.7612	0.7319	0.1161	0.4696
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4621	10.6236	0.7591	0.7212	0.116	0.2993
500	4.4635	11.7805	0.7575	0.6903	0.1157	0.262
700	4.4741	11.8081	0.7619	0.7251	0.1156	0.2902
900	4.4829	10.5626	0.7599	0.7179	0.1158	0.2984
1500	4.46	9.1747	0.7619	0.7153	0.1151	0.3059

b. POISSON

Table 5.2

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.7988	20.2422	0.7348	0.8661	0.1954	0.5659
500	6.8096	20.2383	0.7318	0.865	0.1949	0.5664
700	6.8361	19.9922	0.7317	0.8625	0.1959	0.5669
900	6.8317	20.2141	0.7307	0.8664	0.1965	0.5661
1500	6.8087	20.1585	0.7318	0.8611	0.1952	0.5661
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.812	21.0939	0.7326	0.8465	0.1957	0.5278
500	6.8276	21.0827	0.7292	0.8436	0.1952	0.5292
700	6.8467	21.0896	0.7309	0.844	0.1962	0.5289
900	6.7836	21.0807	0.7332	0.849	0.1947	0.529
1500	6.8267	21.0748	0.7339	0.8426	0.1963	0.5295
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.8125	21.1704	0.7301	0.8158	0.1954	0.4579
500	6.8102	21.144	0.731	0.8159	0.1952	0.4612
700	6.8462	20.9381	0.7327	0.807	0.1959	0.4611
900	6.7922	20.9207	0.733	0.8066	0.1947	0.4624
1500	6.7908	21.4214	0.7336	0.8152	0.1961	0.456
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.8274	20.7617	0.7339	0.789	0.1957	0.3948
500	6.8229	22.6137	0.7326	0.7799	0.1953	0.3908
700	6.7942	25.2898	0.7324	0.7871	0.1943	0.3847
900	6.7939	16.9884	0.732	0.7826	0.195	0.3981
1500	6.825	23.0853	0.7332	0.7794	0.1953	0.3875

c. SPECKLE

Table 5.3

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	3.9327	20.4442	0.7914	0.8509	0.0844	0.7031
500	3.9555	18.9699	0.7935	0.8379	0.0843	0.7204
700	3.9535	20.0929	0.7936	0.8494	0.0843	0.7085
900	3.9382	19.9225	0.7907	0.8487	0.0839	0.7097
1500	3.9397	20.2835	0.7931	0.8555	0.0835	0.7053
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	3.9361	21.0364	0.7917	0.7512	0.0837	0.5594
500	3.9372	20.6887	0.7919	0.7463	0.0834	0.5749
700	3.9384	21.1596	0.7915	0.7509	0.084	0.5485

900	3.9237	20.8806	0.7899	0.7484	0.0842	0.5703
1500	3.9467	21.146	0.7928	0.7629	0.0835	0.5529
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	3.9343	14.1927	0.7911	0.706	0.0836	0.3751
500	3.9533	15.3199	0.7951	0.7177	0.0848	0.3713
700	3.9524	18.617	0.7942	0.7325	0.0835	0.3533
900	3.9498	15.7276	0.7919	0.7139	0.0833	0.3663
1500	3.9426	13.1439	0.7944	0.7113	0.0836	0.3811
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	3.9551	6.4981	0.793	0.7146	0.0838	0.2227
500	3.9255	7.7239	0.7917	0.7239	0.0841	0.2161
700	3.9479	7.1029	0.7928	0.7181	0.0836	0.218
900	3.9531	6.6842	0.7933	0.7233	0.0845	0.2222
1500	3.9508	7.6277	0.7911	0.722	0.0836	0.2153

Image2: BLANKET

a. GAUSSIAN (0,0.01)

Table 5.4

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4646	17.8495	0.8037	0.8086	0.4321	0.2782
500	4.4699	17.8177	0.8065	0.8064	0.4339	0.2783
700	4.4635	18.2421	0.8072	0.7982	0.4361	0.2698
900	4.4687	17.2396	0.8066	0.8069	0.4352	0.278
1500	4.2378	17.9706	0.725	0.8099	0.2214	0.2783
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.231	13.6372	0.7385	0.8486	0.2633	0.488
500	4.2288	15.5208	0.7383	0.8647	0.2621	0.5305
700	4.4777	15.214	0.8072	0.8661	0.434	0.5308
900	4.2328	16.802	0.7257	0.8888	0.2213	0.5956
1500	4.4689	15.2098	0.8062	0.87	0.4351	0.5371
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4496	11.0748	0.8064	0.8534	0.4338	0.5416
500	4.4678	11.5171	0.8052	0.8951	0.4351	0.6656
700	4.4587	10.5584	0.8049	0.8545	0.4345	0.5424
900	4.4713	13.9472	0.8059	0.8955	0.4349	0.6648
1500	4.4614	12.5769	0.805	0.8948	0.4341	0.6675
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.2139	7.1938	0.7257	0.8788	0.2217	0.6431
500	4.4743	8.9668	0.8065	0.8788	0.4332	0.6364
700	4.23	7.4665	0.7251	0.8167	0.2216	0.4443
900	4.2189	6.7317	0.7243	0.8313	0.2206	0.4898
1500	4.2331	7.9433	0.7244	0.8787	0.221	0.6391

b. POISSON

Table 5.5

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noised	Filtered	Noised	Filtered	Noised	Filtered
300	6.1147	17.7539	0.9016	0.8096	0.7159	0.2786
500	6.111	17.8113	0.9016	0.811	0.7164	0.2791
700	6.1179	17.8043	0.9079	0.8105	0.7164	0.2787
900	6.1088	17.8621	0.9011	0.81	0.716	0.2789
1500	5.0603	17.7449	0.8146	0.8105	0.4648	0.2791
Freq.(Var=.3)	Noised	Filtered	Noised	Filtered	Noised	Filtered
300	6.0898	15.4166	0.9012	0.9026	0.717	0.6157
500	6.1321	17.0018	0.9013	0.9031	0.7192	0.6196
700	6.1044	17.7785	0.9017	0.9038	0.7172	0.621
900	6.1175	16.7889	0.9011	0.9029	0.7177	0.6185
1500	6.1019	17.5533	0.9013	0.9024	0.7173	0.62
Freq.(Var=.5)	Noised	Filtered	Noised	Filtered	Noised	Filtered
300	6.0848	15.4327	0.9013	0.9243	0.7159	0.7373
500	6.0842	17.1896	0.9001	0.9235	0.7146	0.74
700	6.1147	16.0366	0.9016	0.9235	0.7159	0.7375
900	6.111	15.2534	0.9016	0.9236	0.7164	0.7368
1500	6.1179	15.0606	0.9017	0.923	0.7164	0.7362
Freq.(Var=.7)	Noised	Filtered	Noised	Filtered	Noised	Filtered
300	5.0749	11.0968	0.8157	0.9306	0.4645	0.7805
500	6.1047	14.4666	0.9007	0.9307	0.7182	0.7828
700	6.0898	12.6665	0.9012	0.9311	0.717	0.7826
900	6.1317	12.677	0.9013	0.932	0.7192	0.7832
1500	6.1044	13.1112	0.9017	0.9307	0.7172	0.7833

c. SPECKLE

Table 5.6

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	3.9538	17.4458	0.7412	0.8052	0.3358	0.2769
500	3.9327	17.5741	0.743	0.8063	0.3355	0.2768
700	3.9536	17.4351	0.7432	0.8043	0.3379	0.2764
900	3.9608	17.279	0.7429	0.8048	0.3384	0.2772
1500	3.9304	17.6824	0.7426	0.8056	0.3395	0.2773
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	3.9283	15.5848	0.7434	0.8773	0.3373	0.5728
500	3.9487	14.9095	0.7443	0.8773	0.3379	0.574
700	3.9432	13.1123	0.7434	0.8778	0.3364	0.5715
900	3.9342	14.2625	0.7452	0.877	0.3377	0.5718

1500	3.9296	13.5843	0.7436	0.876	0.3377	0.5714
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	3.9507	9.0666	0.744	0.8696	0.3386	0.6122
500	3.9307	9.8901	0.7431	0.8718	0.337	0.6132
700	3.9572	9.6317	0.7438	0.8729	0.337	0.6144
900	3.9535	9.028	0.7445	0.8711	0.3374	0.6122
1500	3.9519	9.4848	0.7409	0.8711	0.3367	0.612
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	3.9494	6.1051	0.7421	0.8387	0.337	0.5485
500	3.9599	6.1881	0.7422	0.8357	0.3369	0.5477
700	3.9293	6.0909	0.745	0.8371	0.3381	0.5483
900	3.9562	6.001	0.7426	0.8358	0.3361	0.5465
1500	3.9523	6.0274	0.7421	0.8361	0.3376	0.5494

Image3: GRAY

a. GAUSSIAN (0,0.01)

Table 5.7

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4601	12.8116	0.9454	0.9903	0.108	0.8974
500	4.4619	12.8508	0.9444	0.9905	0.1089	0.8963
700	4.4622	12.8656	0.9455	0.9908	0.1081	0.8968
900	4.4657	12.8169	0.9441	0.9906	0.109	0.8974
1500	4.4386	12.8582	0.9447	0.9905	0.108	0.8975
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4811	11.1925	0.9432	0.9937	0.1003	0.7624
500	4.4622	11.4481	0.9455	0.9945	0.1081	0.7616
700	4.4937	11.5436	0.9434	0.9934	0.1005	0.761
900	4.4386	10.9906	0.9447	0.9941	0.108	0.7623
1500	4.4687	10.8372	0.9438	0.9945	0.1	0.7627
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4732	8.2029	0.9436	0.9901	0.1071	0.5335
500	4.4781	8.1882	0.9432	0.9908	0.1061	0.5343
700	4.4601	8.167	0.9454	0.9914	0.108	0.5392
900	4.442	8.1798	0.9415	0.9902	0.1006	0.5357
1500	4.4622	8.2265	0.9455	0.9911	0.1081	0.5335
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4657	6.3586	0.9441	0.983	0.109	0.332
500	4.4386	6.3156	0.9447	0.9839	0.108	0.3323
700	4.4503	6.3327	0.9454	0.9837	0.1083	0.3324
900	4.4732	6.3406	0.9436	0.9833	0.1071	0.3306
1500	4.4579	6.3212	0.945	0.9836	0.1077	0.332

b. POISSON

Table 5.8

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	7.7116	12.6991	0.9892	0.9902	0.4946	0.9456
500	7.6021	12.6813	0.9871	0.9897	0.4836	0.9455
700	7.6926	12.7655	0.9888	0.9901	0.4942	0.9453
900	7.6825	12.7383	0.9889	0.9896	0.4944	0.9455
1500	7.6738	12.6685	0.9888	0.9903	0.4934	0.9454
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	7.6733	14.6641	0.9892	0.9958	0.4947	0.9363
500	7.6931	14.8521	0.9888	0.9957	0.4978	0.9359
700	7.736	14.9486	0.9893	0.996	0.497	0.9361
900	7.6855	14.8028	0.9888	0.9957	0.4953	0.9362
1500	7.6711	14.7953	0.9888	0.9957	0.4943	0.9368
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	7.6943	13.0313	0.9891	0.9981	0.4946	0.8566
500	7.638	12.9867	0.9885	0.9978	0.4915	0.8567
700	7.6867	12.9265	0.9892	0.9975	0.4966	0.8563
900	7.6435	12.968	0.9884	0.9977	0.4928	0.8554
1500	7.6586	12.9346	0.9895	0.998	0.4919	0.8552
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	7.7013	11.4783	0.989	0.9983	0.4946	0.771
500	7.6736	11.4428	0.9884	0.9977	0.4936	0.7696
700	7.687	11.4391	0.9887	0.9984	0.4956	0.7689
900	7.6882	11.4613	0.9888	0.998	0.4937	0.7704
1500	7.7047	11.5307	0.9885	0.9975	0.4941	0.7714

c. SPECKLE

Table 5.9

SNR			Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.5825	12.704	0.9279	0.9964	0.2387	0.9342
500	4.574	12.9791	0.9274	0.9933	0.2404	0.9335
700	4.5873	12.7315	0.9282	0.994	0.24	0.9343
900	4.6015	12.5877	0.9281	0.9961	0.2407	0.9342
1500	4.5719	12.9515	0.9294	0.9954	0.2408	0.934
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.5734	8.1166	0.9283	0.994	0.2401	0.7672
500	4.5766	8.3188	0.9276	0.9941	0.2397	0.7691
700	4.5815	8.5789	0.9289	0.993	0.24	0.7642
900	4.5652	8.1393	0.9286	0.9938	0.2401	0.7649

1500	4.5738	8.464	0.9283	0.9933	0.2404	0.7657
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.5944	6.361	0.9272	0.989	0.2401	0.5618
500	4.5652	6.3877	0.9278	0.9896	0.2397	0.5618
700	4.5825	6.3883	0.9288	0.99	0.2387	0.563
900	4.5829	6.4384	0.928	0.9892	0.2396	0.5616
1500	4.5772	6.3655	0.9281	0.989	0.2403	0.5605
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.5821	5.5724	0.9284	0.978	0.2396	0.4082
500	4.6011	5.5752	0.9283	0.9781	0.24	0.4083
700	4.5827	5.5599	0.9275	0.9777	0.2406	0.4084
900	4.5909	5.5512	0.928	0.9776	0.2402	0.4073
1500	4.5915	5.576	0.9283	0.9776	0.2401	0.4088

Image4: JUNGLE

a. GAUSSIAN (0,0.01)

Table 5.10

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4603	22.3297	0.8212	0.9331	0.2157	0.5457
500	4.4835	22.1344	0.8224	0.9327	0.2151	0.5508
700	4.4713	21.4105	0.82	0.9339	0.213	0.5525
900	4.4755	21.7345	0.8227	0.9349	0.2156	0.55
1500	4.4963	21.4034	0.8224	0.9341	0.2146	0.5522
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4866	23.8267	0.8212	0.9668	0.2153	0.6655
500	4.4647	22.7904	0.8216	0.9659	0.2158	0.6771
700	4.4644	24.0058	0.8211	0.9655	0.2144	0.6624
900	4.4737	23.5765	0.8239	0.9653	0.2158	0.6694
1500	4.4647	23.8863	0.8224	0.9653	0.2144	0.6648
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4141	24.33	0.8022	0.9483	0.2357	0.5607
500	4.433	23.4134	0.8064	0.9395	0.2344	0.563
700	4.4636	22.5385	0.821	0.9399	0.2354	0.5605
900	4.4139	24.9142	0.8002	0.9546	0.2371	0.5665
1500	4.4647	20.9535	0.8227	0.9327	0.2358	0.5564
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4012	15.9684	0.7999	0.9205	0.2348	0.4525
500	4.4778	15.712	0.8229	0.9211	0.2331	0.4427
700	4.4719	16.4543	0.8199	0.9212	0.2328	0.4482
900	4.4841	15.9841	0.8225	0.922	0.235	0.4494
1500	4.4595	16.5276	0.8219	0.9187	0.161	0.3838

b. POISSON

Table 5.11

SNR			Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	7.3067	21.4506	0.9405	0.9351	0.5971	0.556
500	7.3767	21.4085	0.9426	0.9342	0.5994	0.556
700	7.3578	21.5218	0.942	0.9357	0.5988	0.5553
900	7.3638	21.5414	0.9429	0.9328	0.6008	0.5549
1500	7.3876	21.4491	0.9428	0.936	0.6013	0.5556
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	7.3638	26.3869	0.942	0.979	0.6011	0.7445
500	7.3887	25.5521	0.9425	0.9782	0.6026	0.7496
700	7.348	25.7466	0.9419	0.9785	0.5983	0.7476
900	7.3616	24.6808	0.9419	0.9781	0.6009	0.7544
1500	7.3465	24.4812	0.9413	0.9782	0.5981	0.7555
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	7.378	33.3375	0.9434	0.9829	0.6019	0.7615
500	7.3766	33.7411	0.9424	0.9828	0.6017	0.764
700	7.3739	33.7332	0.9432	0.9825	0.6017	0.7625
900	7.3614	29.0246	0.9408	0.9812	0.5978	0.7781
1500	7.3935	34.9371	0.9424	0.9826	0.6004	0.758
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	7.3679	44.7193	0.9429	0.9796	0.6002	0.7189
500	7.354	44.789	0.9415	0.9799	0.6002	0.7169
700	7.3827	41.0633	0.9425	0.979	0.6005	0.7326
900	7.3993	43.0146	0.9425	0.9793	0.6016	0.7235
1500	7.3869	43.4007	0.9431	0.9792	0.601	0.7261

c. SPECKLE

Table 5.12

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.2515	21.6187	0.7869	0.934	0.2192	0.5508
500	4.2742	21.2502	0.7891	0.9339	0.2208	0.5537
700	4.2531	20.8185	0.7861	0.9312	0.2193	0.5558
900	4.4635	20.6612	0.7873	0.9327	0.2196	0.5576
1500	4.2702	21.1216	0.7888	0.9327	0.221	0.554
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.2659	22.8579	0.7878	0.9642	0.2209	0.6609
500	4.248	23.3652	0.7882	0.9632	0.2196	0.6559
700	4.2725	23.4558	0.7884	0.9648	0.2209	0.6543
900	4.2678	23.1007	0.7872	0.9635	0.2206	0.6587

1500	4.2398	22.9977	0.7869	0.963	0.2196	0.6592
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.2637	25.0775	0.7885	0.9473	0.2214	0.553
500	4.2521	22.9689	0.7874	0.964	0.2198	0.5658
700	4.2511	23.3696	0.7873	0.9437	0.2209	0.5643
900	4.2466	23.3304	0.7875	0.9442	0.2205	0.5641
1500	4.245	25.0494	0.7866	0.9445	0.2211	0.5551
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.3653	14.5167	0.7883	0.9031	0.2205	0.4141
500	4.2425	13.9534	0.7875	0.9009	0.2211	0.4199
700	4.2637	14.1571	0.7871	0.9024	0.2201	0.4183
900	4.2688	13.4862	0.7896	0.9024	0.2215	0.4221
1500	4.2615	14.2801	0.7882	0.8998	0.219	0.4151

Image5: MOON

a. GAUSSIAN (0,0.01)

Table 5.13

SNR			Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4801	12.8908	0.8115	0.8965	0.2338	0.5293
500	4.3882	13.1263	0.7803	0.8984	0.1254	0.5289
700	4.4818	13.1573	0.8088	0.8975	0.2352	0.5293
900	4.4915	13.1239	0.8098	0.8978	0.2349	0.5281
1500	4.4943	13.4521	0.8129	0.8982	0.2345	0.5271
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.5065	10.8249	0.8066	0.9337	0.2337	0.6065
500	4.475	13.46	0.8099	0.9343	0.2364	0.6079
700	4.4723	12.4888	0.808	0.9354	0.2321	0.6075
900	4.4497	12.4698	0.809	0.9352	0.237	0.6097
1500	4.4742	12.8783	0.8102	0.9348	0.235	0.6083
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4899	9.0391	0.8066	0.9238	0.2357	0.5607
500	4.4395	9.0516	0.8114	0.9265	0.2344	0.563
700	4.4543	9.2646	0.8062	0.9249	0.2354	0.5605
900	4.5261	8.9887	0.8104	0.9252	0.2371	0.5665
1500	4.48	9.2278	0.8119	0.9242	0.2358	0.5564
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.4816	6.6186	0.8116	0.8993	0.2348	0.4525
500	4.4635	7.6099	0.8066	0.8972	0.2331	0.4427
700	4.4958	6.398	0.8073	0.9004	0.2328	0.4482
900	4.4967	7.2347	0.8103	0.8961	0.235	0.4494
1500	4.4415	6.5833	0.7923	0.8875	0.161	0.3838

b. POISSON

Table 5.14

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.8449	12.9602	0.9207	0.8985	0.5542	0.5315
500	6.7955	12.8598	0.9198	0.8984	0.5528	0.5326
700	6.7904	12.859	0.9223	0.8991	0.5526	0.532
900	6.8281	13.002	0.9189	0.8993	0.5502	0.5307
1500	6.8771	13.0459	0.9204	0.8977	0.5531	0.5302
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.8296	12.4984	0.9203	0.9462	0.5535	0.6666
500	6.8726	13.3386	0.9203	0.9466	0.5541	0.6651
700	6.7629	12.3489	0.9193	0.9473	0.5534	0.6665
900	6.878	12.1579	0.9213	0.946	0.554	0.6661
1500	6.8687	12.0434	0.9223	0.9462	0.557	0.6641
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.8584	12.2187	0.9219	0.9535	0.5551	0.704
500	6.8265	13.3329	0.921	0.9527	0.5543	0.7064
700	6.8624	11.5684	0.9192	0.954	0.5544	0.7052
900	6.8548	13.0254	0.9212	0.9538	0.5545	0.7042
1500	6.8242	12.6339	0.9203	0.9531	0.5524	0.7049
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.8333	10.3696	0.9198	0.9564	0.5518	0.6993
500	6.8344	9.9965	0.9181	0.9544	0.5534	0.7016
700	6.7968	11.0417	0.9197	0.9555	0.5484	0.7017
900	6.8064	11.3508	0.9182	0.956	0.5491	0.6997
1500	6.8462	10.3345	0.9178	0.9563	0.5527	0.7038

c. SPECKLE

Table 5.15

SNR			Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.1191	13.5825	0.7675	0.8972	0.2162	0.5255
500	4.1417	13.5589	0.768	0.8945	0.218	0.5242
700	4.1691	13.7551	0.7696	0.9004	0.2204	0.5242
900	4.149	13.9626	0.7712	0.898	0.2196	0.5231
1500	4.1174	13.3824	0.7701	0.8989	0.2176	0.5267
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.1182	13.4312	0.7654	0.9294	0.2176	0.5871
500	4.0995	13.5288	0.7641	0.9306	0.2182	0.5883
700	4.1561	12.9095	0.7678	0.9288	0.22	0.5853
900	4.1318	13.0067	0.7708	0.9296	0.2193	0.587

1500	4.1337	13.2525	0.769	0.9311	0.22	0.5875
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.1446	8.9517	0.7711	0.9108	0.218	0.5189
500	4.12	8.4163	0.7692	0.913	0.2152	0.5225
700	4.1666	8.6869	0.7699	0.9159	0.2198	0.5214
900	4.0958	9.5633	0.7692	0.9159	0.2182	0.521
1500	4.1477	8.2081	0.7683	0.9106	0.2197	0.526
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.1173	6.2117	0.7672	0.8762	0.2173	0.409
500	4.1186	5.7362	0.7673	0.8755	0.2195	0.4163
700	4.1144	6.2571	0.7666	0.8722	0.2192	0.4078
900	4.1396	5.9818	0.7678	0.8724	0.2182	0.409
1500	4.1499	5.9413	0.7687	0.8733	0.2182	0.4061

Image6: ULTRASONIC

a. GAUSSIAN (0,0.01)

Table 5.16

SNR			Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.2536	13.0824	0.7683	0.885	0.1566	0.4732
500	4.5141	13.4912	0.8338	0.8938	0.3648	0.4782
700	4.4338	13.227	0.8185	0.8955	0.312	0.4788
900	4.259	13.6032	0.7681	0.8972	0.1531	0.4778
1500	4.4847	13.5908	0.8315	0.8949	0.3653	0.4778
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.5186	11.5361	0.8293	0.9396	0.3594	0.6377
500	4.4706	10.8864	0.8319	0.941	0.3623	0.6406
700	4.247	11.0739	0.7647	0.9406	0.1526	0.6433
900	4.4671	11.2908	0.8327	0.9417	0.3625	0.6436
1500	4.5141	11.2547	0.8338	0.9402	0.3648	0.6418
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.2709	8.3958	0.7676	0.9454	0.1529	0.649
500	4.5186	9.3426	0.8293	0.9441	0.3594	0.6394
700	4.4706	9.2301	0.8319	0.9446	0.3623	0.6453
900	4.4982	8.9835	0.8283	0.9453	0.3604	0.65
1500	4.4671	7.5188	0.8327	0.9024	0.3625	0.5041
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	4.5141	7.0975	0.8338	0.9312	0.3648	0.5721
500	4.5034	6.6809	0.8335	0.9321	0.3639	0.5739
700	4.48	7.0717	0.8326	0.9304	0.3642	0.5746
900	4.4847	6.8743	0.8315	0.9279	0.3653	0.575
1500	4.279	7.3245	0.7676	0.9314	0.1574	0.5705

b. POISSON

Table 5.17

SNR			Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	8.8884	13.2111	0.9698	0.8958	0.8198	0.4825
500	8.8016	13.5748	0.9676	0.8949	0.8203	0.4818
700	8.8642	13.2248	0.9693	0.8955	0.8182	0.4828
900	8.8252	13.6523	0.9679	0.8937	0.8169	0.4814
1500	8.8212	13.4555	0.968	0.896	0.8203	0.4816
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	8.8082	11.8709	0.9677	0.9474	0.8197	0.7038
500	8.7671	11.9733	0.9672	0.9479	0.8204	0.7031
700	8.8545	12.1307	0.9693	0.9485	0.822	0.7056
900	8.813	12.5224	0.9682	0.9483	0.819	0.7054
1500	8.859	11.9393	0.9692	0.9482	0.8218	0.7029
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	8.8471	11.7483	0.9688	0.9647	0.8216	0.802
500	8.8095	12.3453	0.9683	0.9644	0.8214	0.8022
700	8.9041	12.4299	0.9681	0.9651	0.8191	0.8018
900	8.8762	12.1799	0.9699	0.9642	0.8195	0.8009
1500	8.7909	12.0562	0.9672	0.9661	0.8162	0.8012
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	8.8609	12.018	0.968	0.9721	0.8188	0.8432
500	8.8948	11.3095	0.9687	0.9736	0.8211	0.8438
700	8.8183	12.1856	0.9689	0.9741	0.8188	0.8457
900	8.7804	12.2269	0.9672	0.9728	0.8186	0.845
1500	8.8312	12.2428	0.9702	0.9739	0.8189	0.8472

c. SPECKLE

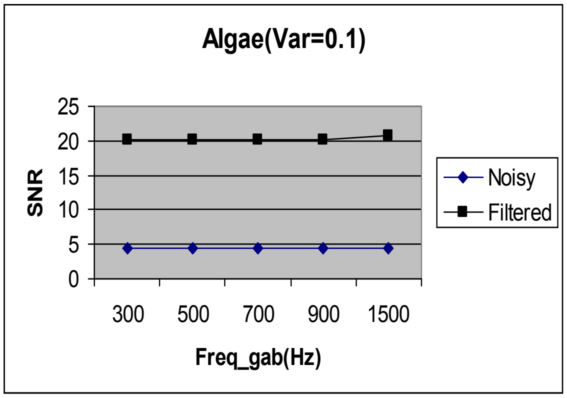
Table 5.18

	SNR		Correlation		SSIM	
Freq.(Var=.1)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.0969	13.1625	0.9109	0.8954	0.5958	0.4815
500	6.0915	13.8385	0.9087	0.8947	0.5912	0.4792
700	6.0859	13.8917	0.9126	0.8955	0.592	0.4795
900	6.0733	14.291	0.9107	0.8949	0.5912	0.4771
1500	6.1028	13.4565	0.8949	0.8954	0.5943	0.4806
Freq.(Var=.3)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.0744	11.3095	0.9091	0.9475	0.5904	0.6914
500	6.1277	12.0146	0.9127	0.9463	0.5974	0.6914
700	6.0958	11.7464	0.9104	0.9476	0.5939	0.6917
900	6.1457	10.5766	0.9123	0.9467	0.5948	0.692

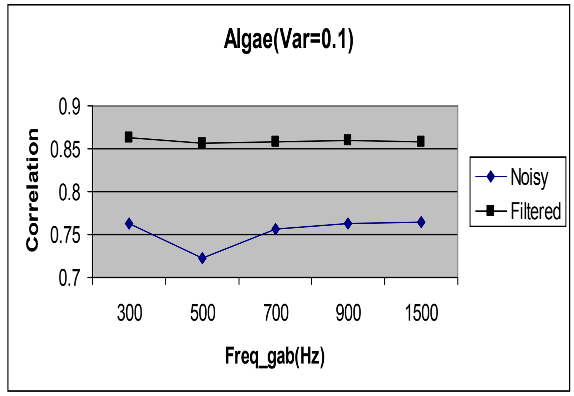
1500	6.0795	11.8136	0.9083	0.946	0.5961	0.6921
Freq.(Var=.5)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.1297	10.9152	0.9089	0.9571	0.5926	0.7648
500	6.0883	10.0438	0.9103	0.958	0.5943	0.7635
700	6.0892	9.8119	0.9099	0.96	0.5962	0.7648
900	6.1057	8.7216	0.9082	0.9587	0.5938	0.7619
1500	6.1492	9.9959	0.9105	0.9574	0.597	0.7639
Freq.(Var=.7)	Noisy	Filtered	Noisy	Filtered	Noisy	Filtered
300	6.0683	8.3242	0.9087	0.9602	0.5931	0.773
500	6.1298	8.8056	0.9118	0.9585	0.5969	0.7711
700	6.101	8.5146	0.9106	0.9597	0.5964	0.7715
900	6.1148	8.3963	0.9135	0.9576	0.5956	0.77
1500	6.053	8.259	0.9074	0.9578	0.5918	0.7691

Following graphs give a simultaneous comparison of SNR, Correlation and SSIM obtained after applying all the proposed algorithms of Gabor Filters, these all graphs having both Noisy and Filtered parameters comparison.

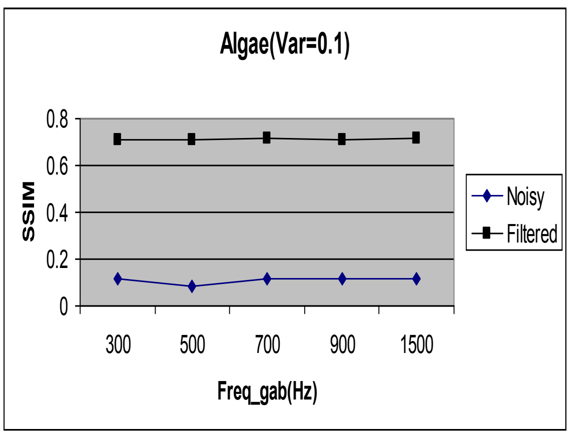
GAUSSIAN NOISE



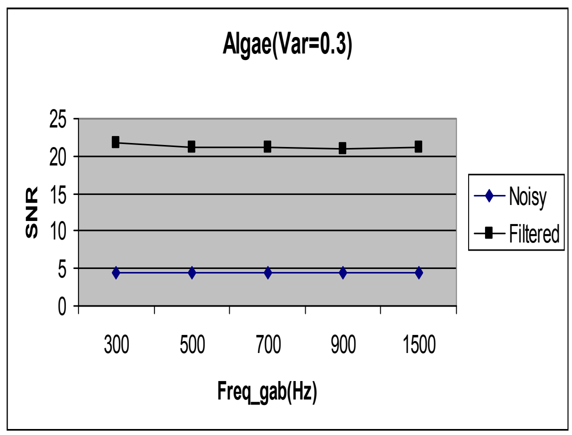
a.



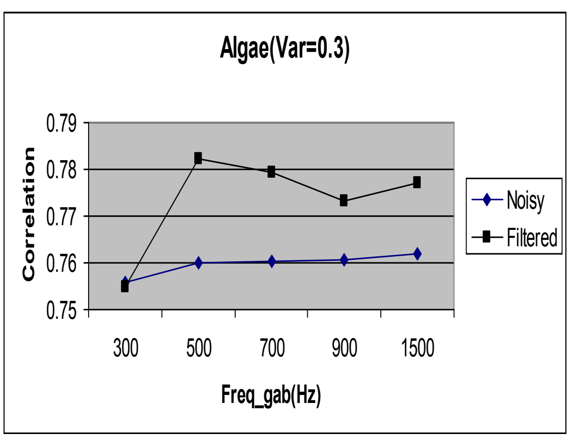
b.



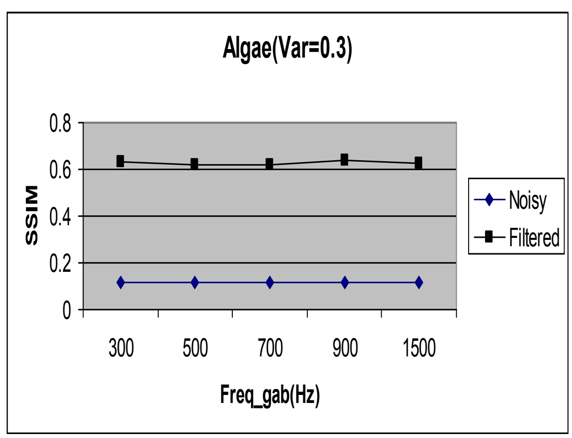
c.



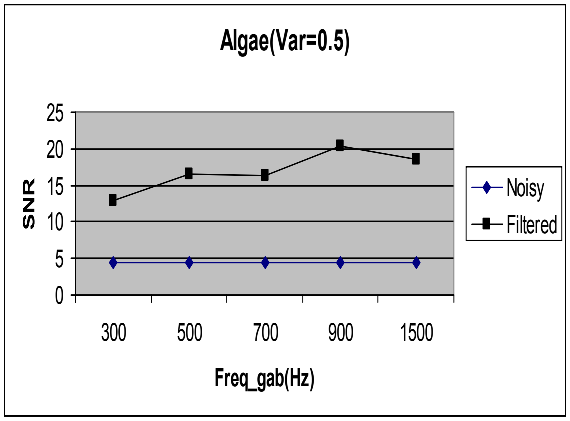
d.

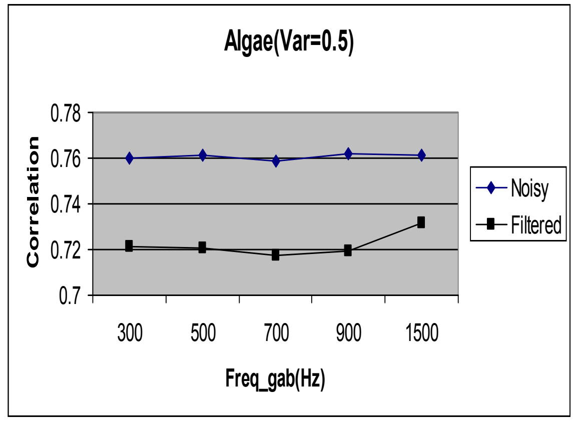


e.

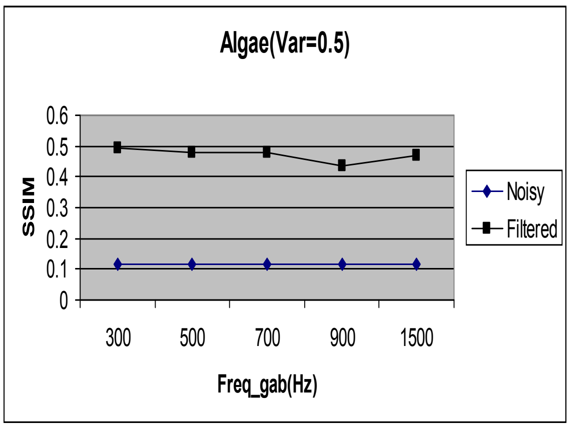


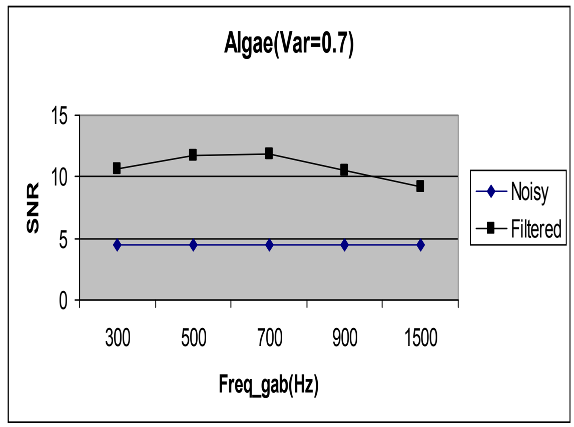
f.



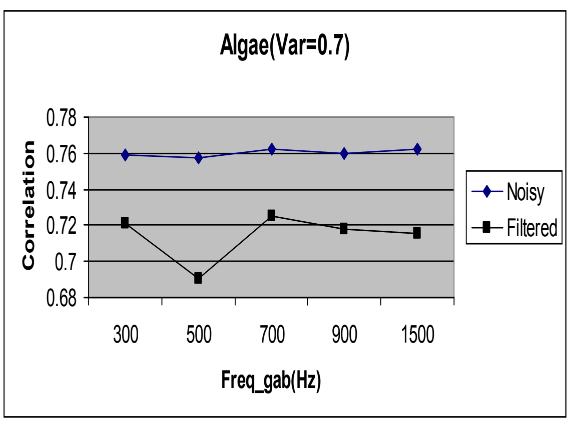


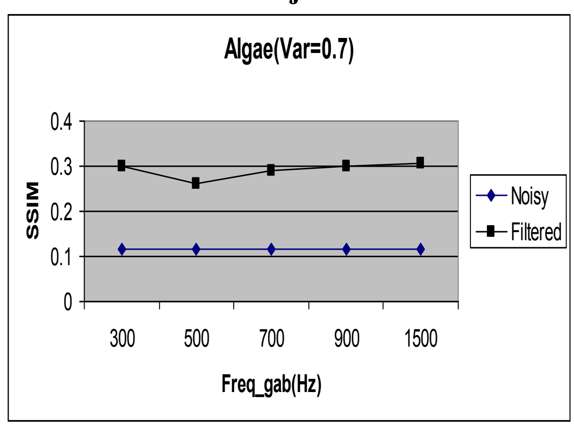
g. **h.**





i. **j.**

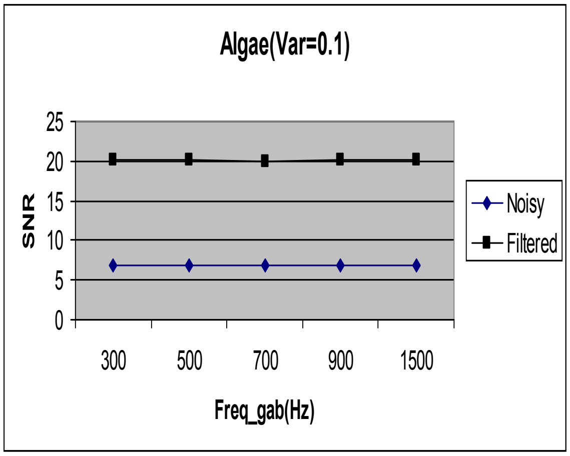


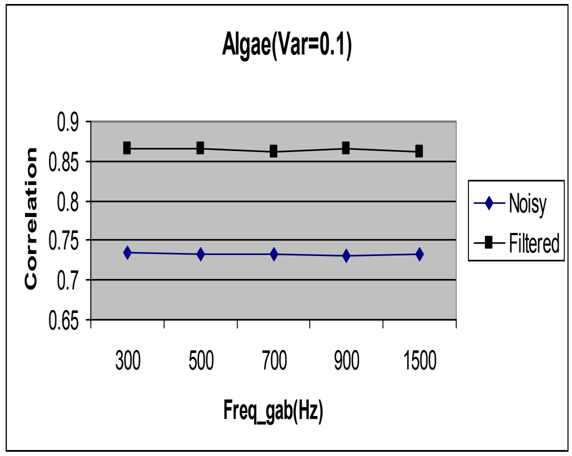


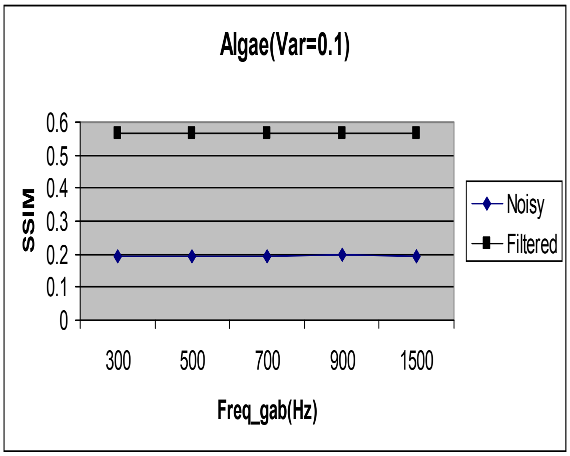
k. **l.**

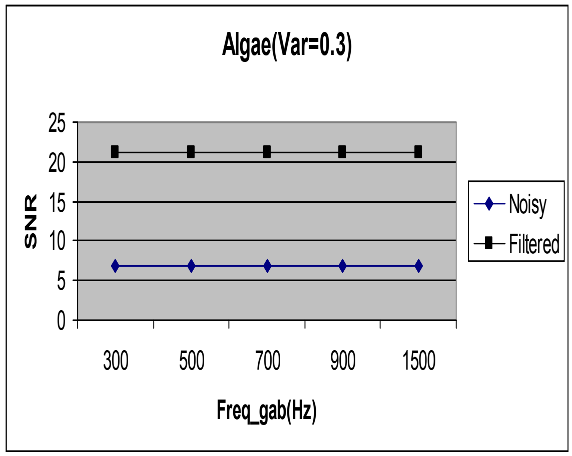
a,b,c: SNR, Correlation and SSIM variation of noisy and filtered values with variance=.1
d,e,f: SNR, Correlation and SSIM variation of noisy and filtered values with variance=.3
g,h,i: SNR, Correlation and SSIM variation of noisy and filtered values with variance=.5
j,k,l: SNR, Correlation and SSIM variation of noisy and filtered values with variance=.7
These above are all graphical results for "ALGAE.tiff" image using Gaussian noise, now following are the details of all the images using all three noises with the same way as above.

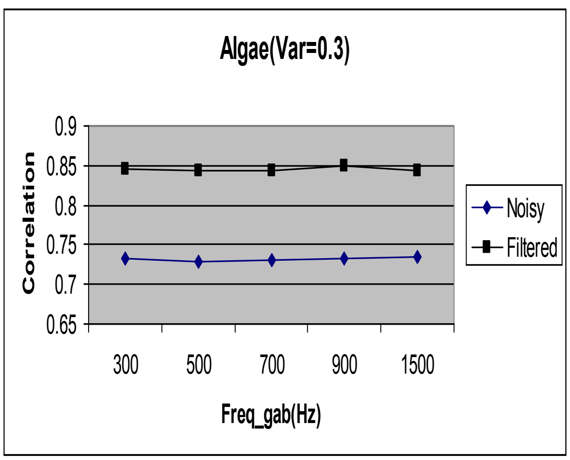
POISSON NOISE

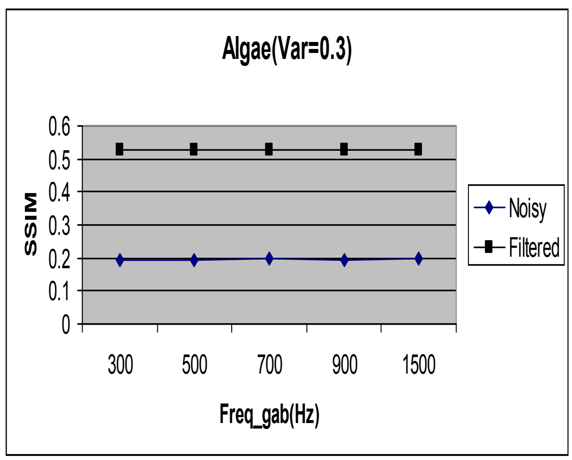


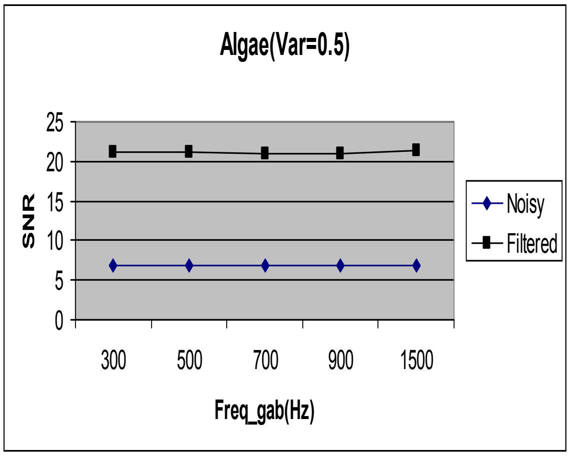


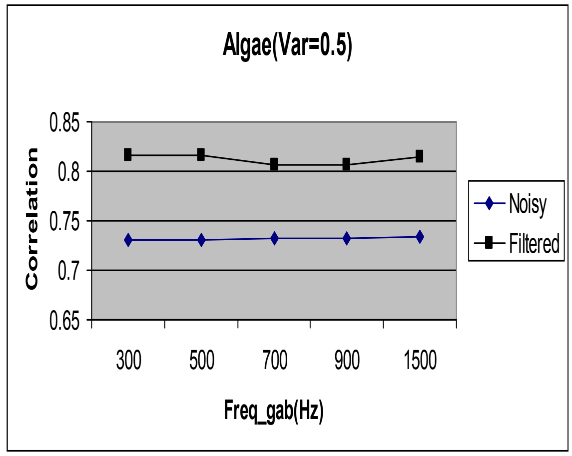


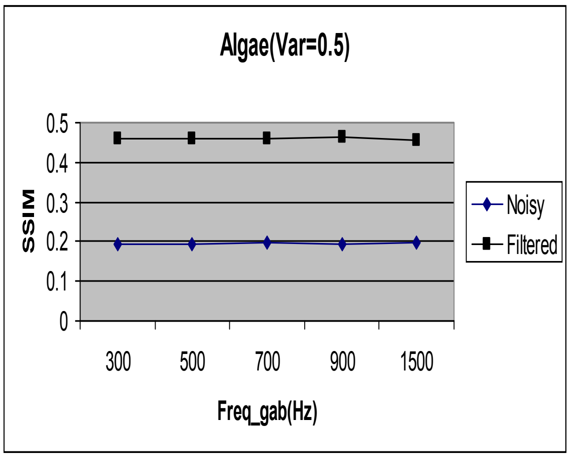


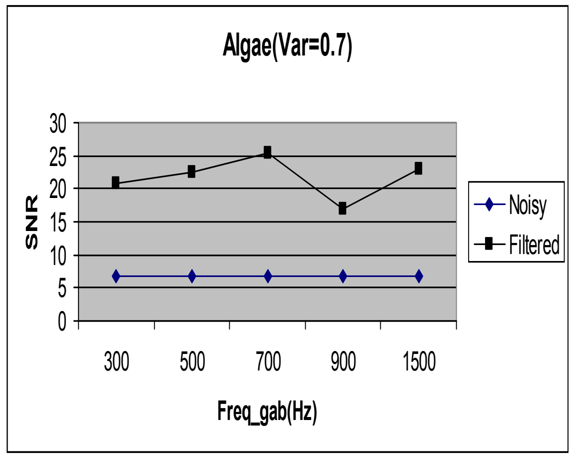


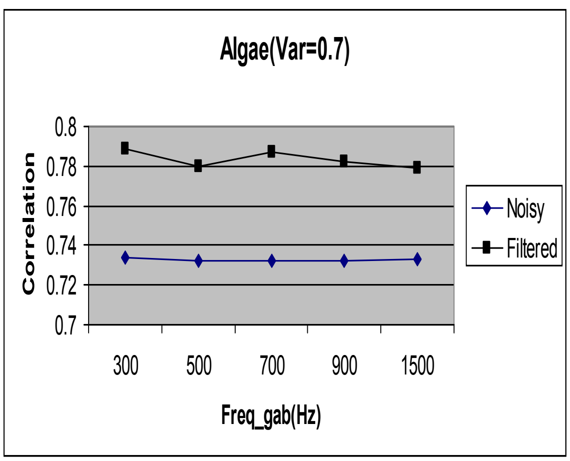


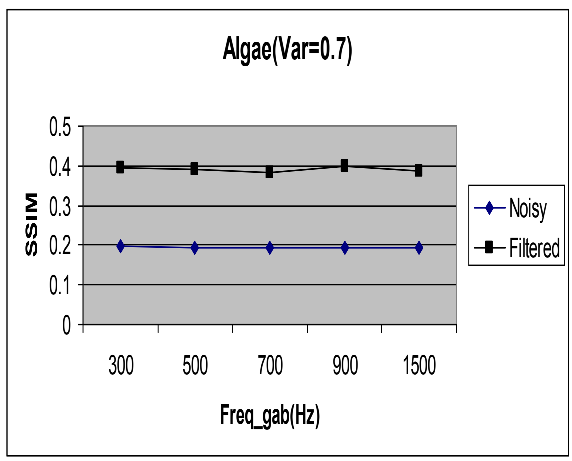

Algae(Var=0.5)
SNR
25
20
15
10
5
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Algae(Var=0.5)
Correlation
0.85
0.8
0.75
0.7
0.65
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

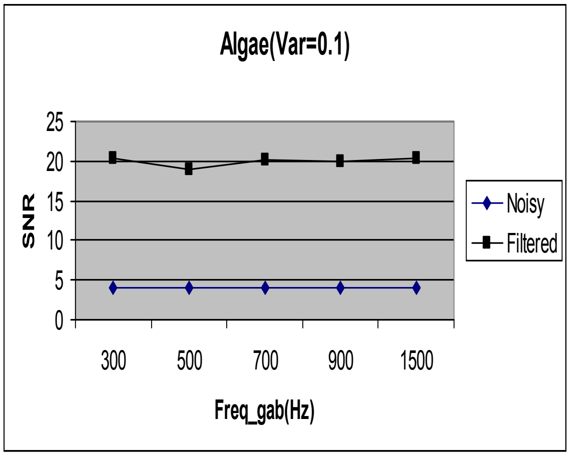

Algae(Var=0.5)
SSIM
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

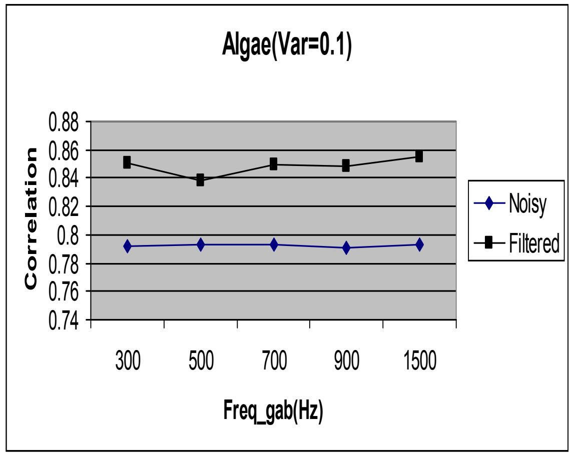

Algae(Var=0.7)
SNR
30
25
20
15
10
5
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

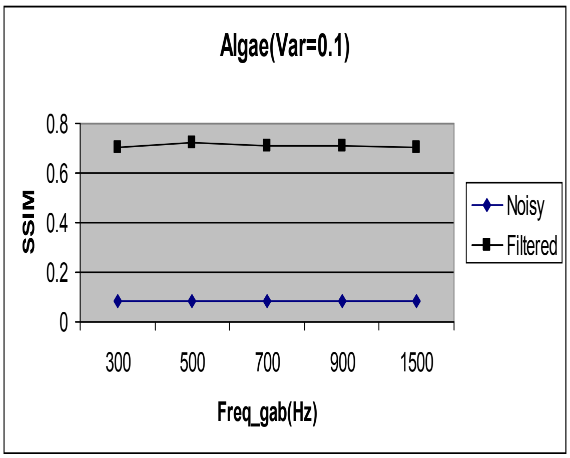

Algae(Var=0.7)
Correlation
0.8
0.78
0.76
0.74
0.72
0.7
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

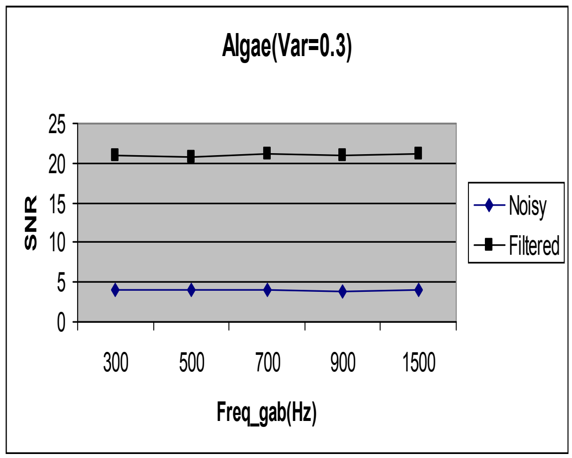

Algae(Var=0.7)
SSIM
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

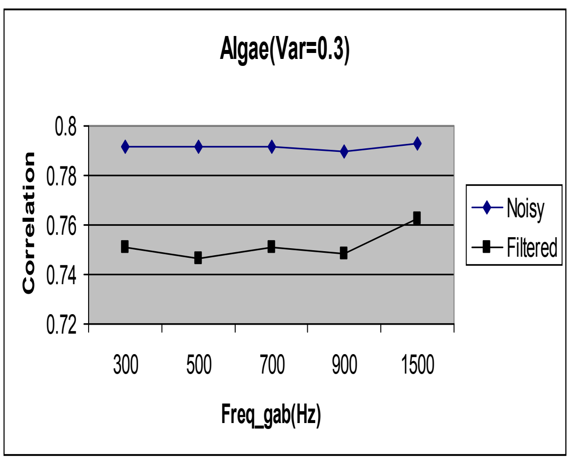
SPECKLE NOISE

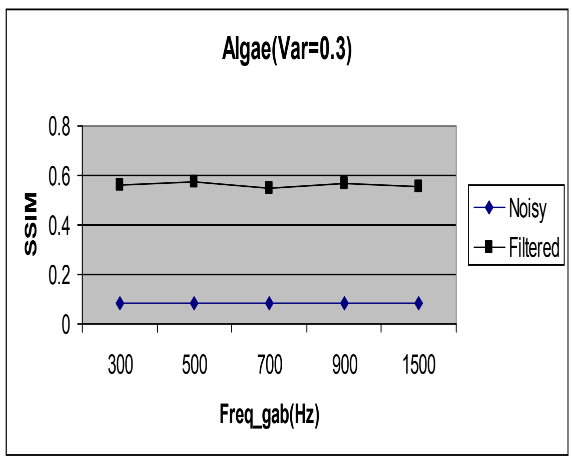


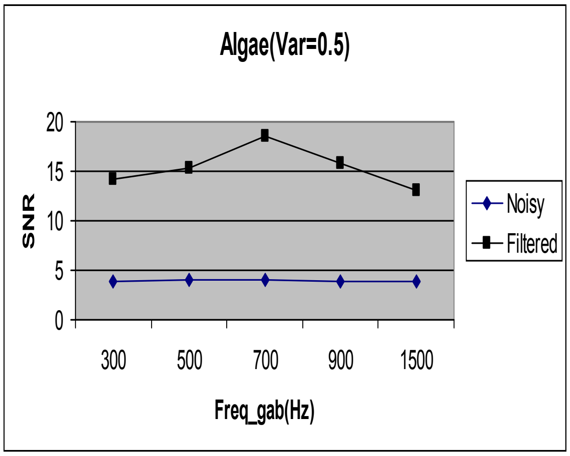


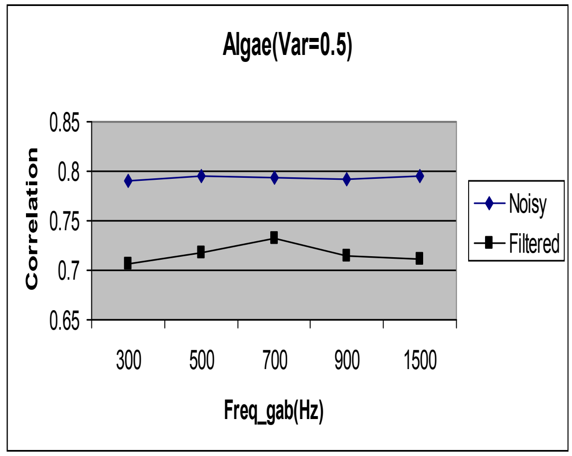


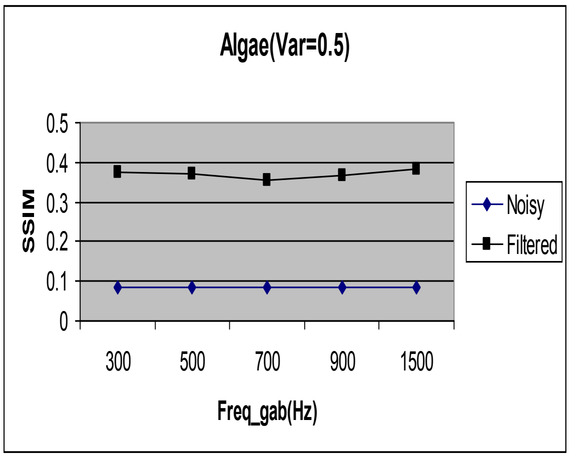


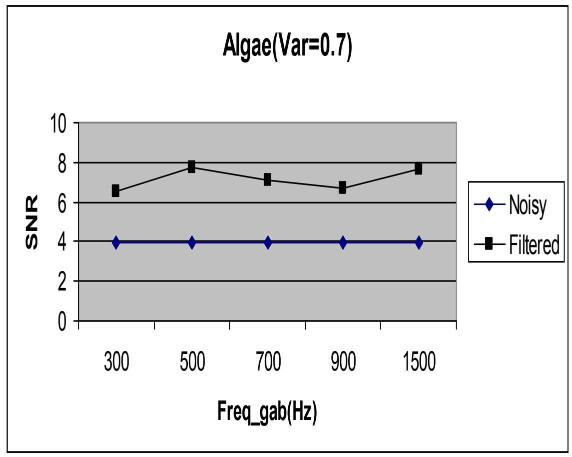


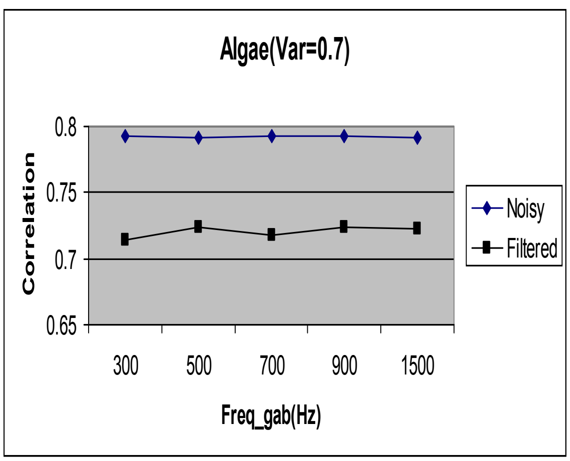


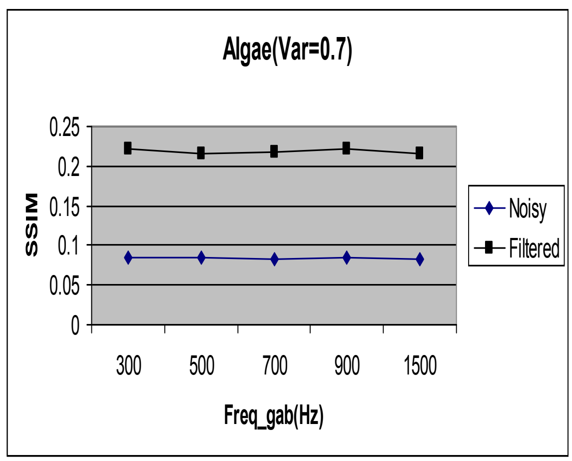

Algae(Var=0.5)
SNR
20
15
10
5
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Algae(Var=0.5)
Correlation
0.85
0.8
0.75
0.7
0.65
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

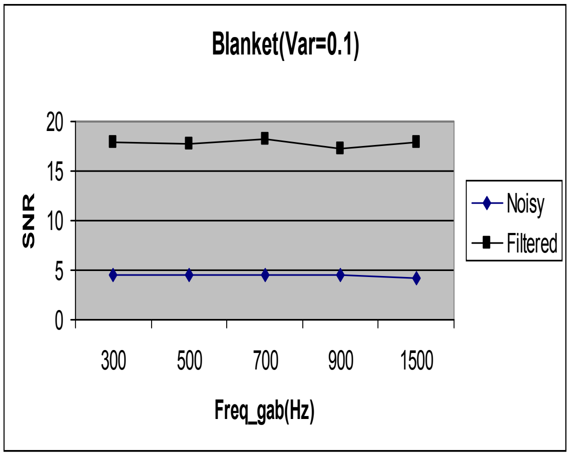

Algae(Var=0.5)
SSIM
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

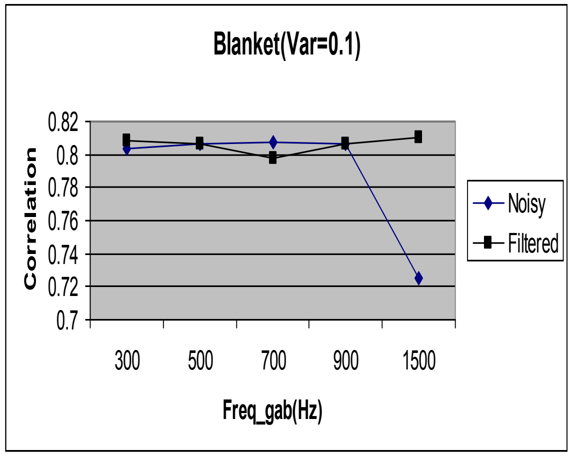

Algae(Var=0.7)
SNR
10
8
6
4
2
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

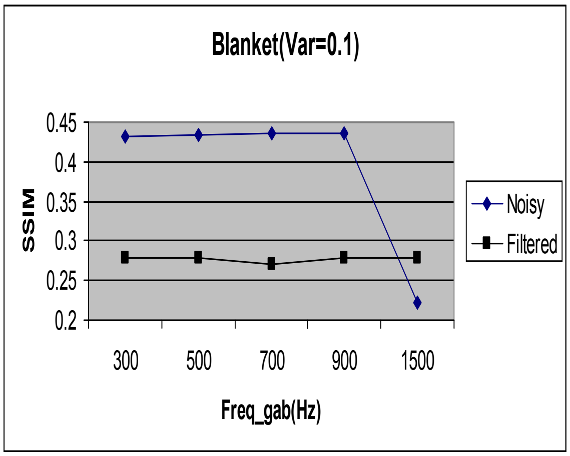

Algae(Var=0.7)
Correlation
0.8
0.75
0.7
0.65
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

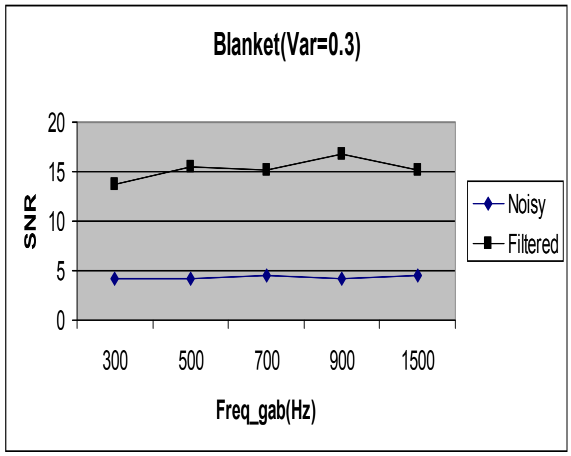

Algae(Var=0.7)
SSIM
0.25
0.2
0.15
0.1
0.05
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

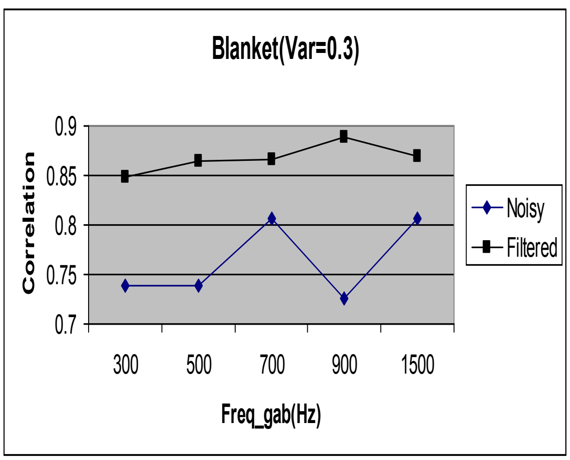
GAUSSIAN NOISE

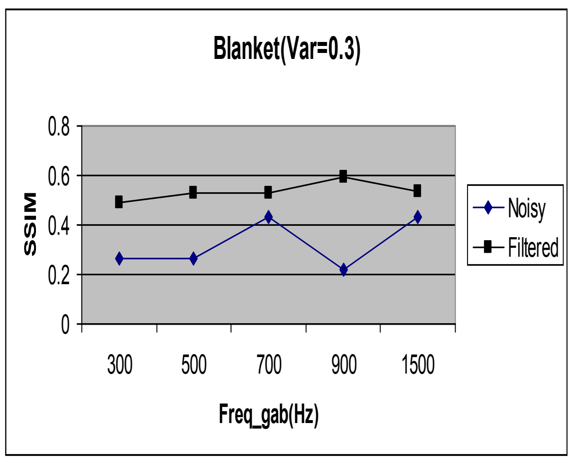


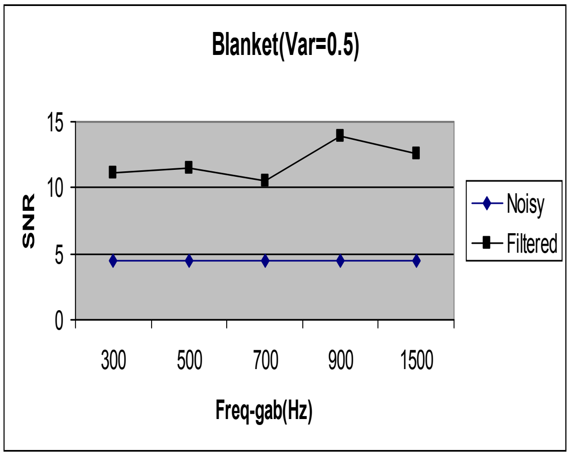


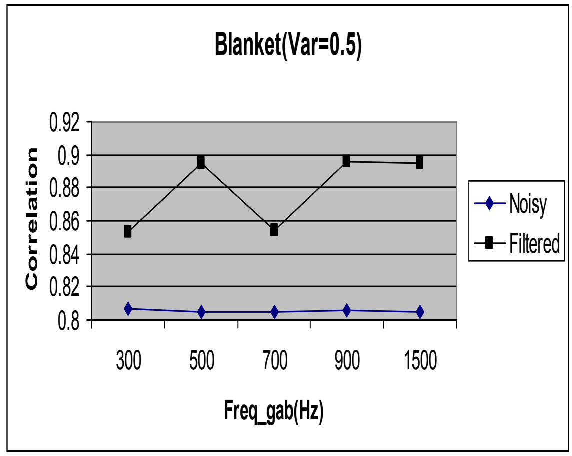


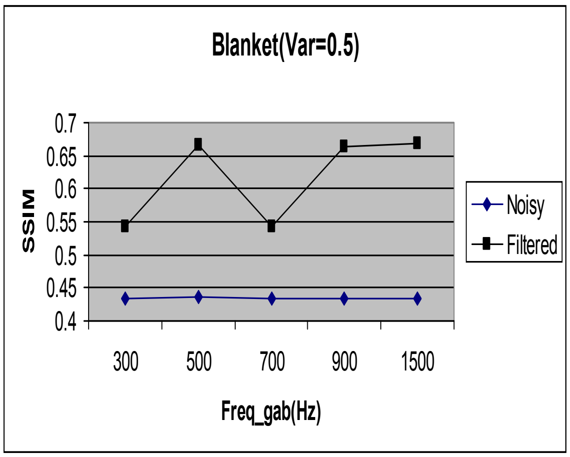


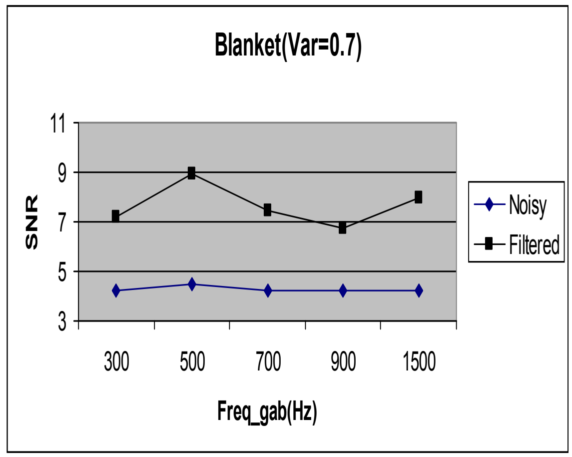


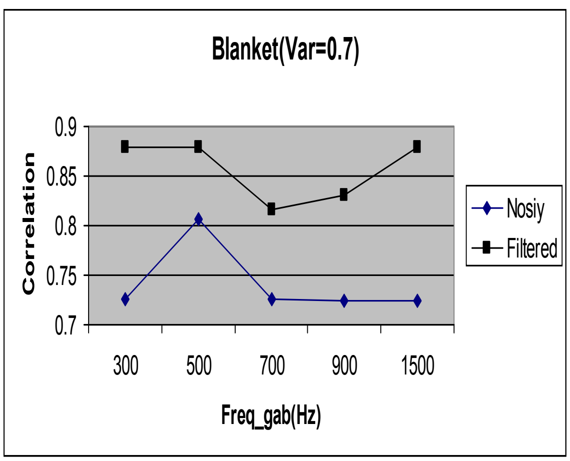


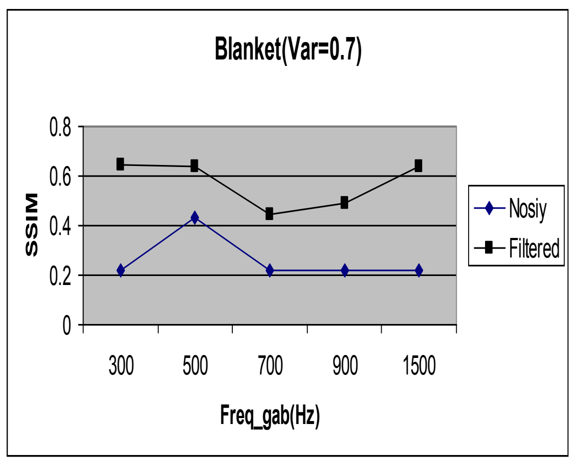

Blanket(Var=0.5)
SNR
15
10
5
0
300
500
700
900
1500
Freq-gab(Hz)
Noisy
Filtered


Blanket(Var=0.5)
Correlation
0.92
0.9
0.88
0.86
0.84
0.82
0.8
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

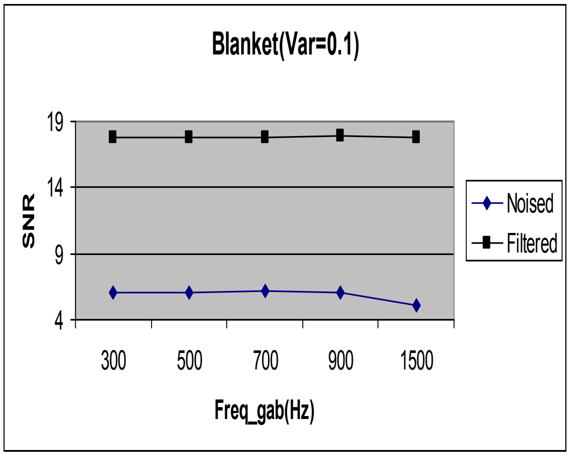

Blanket(Var=0.5)
SSIM
0.7
0.65
0.6
0.55
0.5
0.45
0.4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

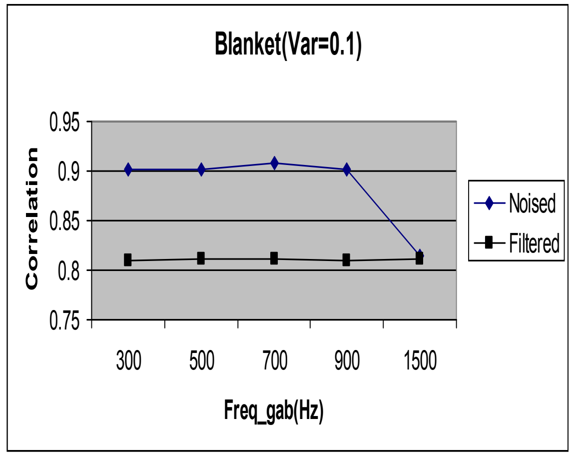

Blanket(Var=0.7)
SNR
11
9
7
5
3
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

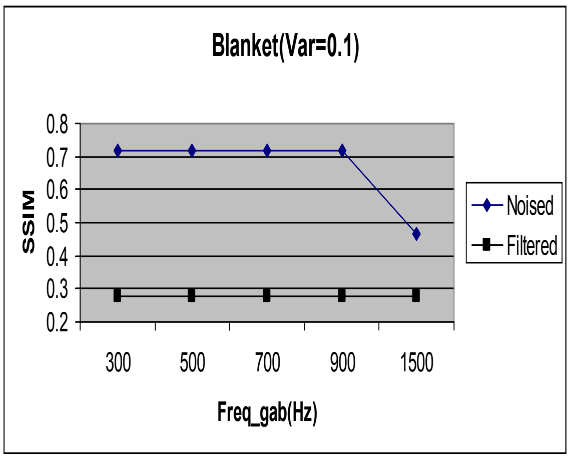

Blanket(Var=0.7)
Correlation
0.9
0.85
0.8
0.75
0.7
300
500
700
900
1500
Freq_gab(Hz)
Nosiy
Filtered

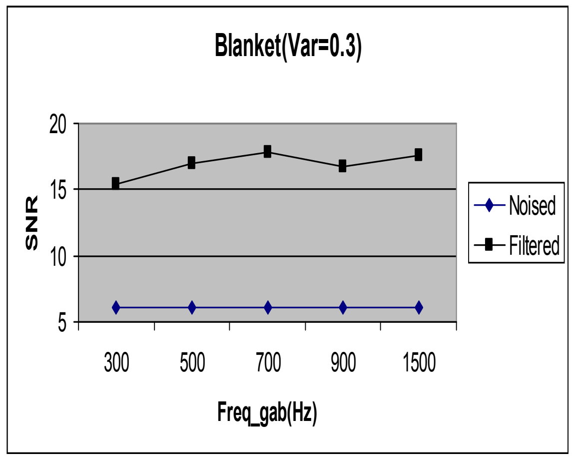

Blanket(Var=0.7)
SSIM
0.8
0.6
0.4
0.2
0
300
500
700
900
1500
Freq_gab(Hz)
Nosiy
Filtered

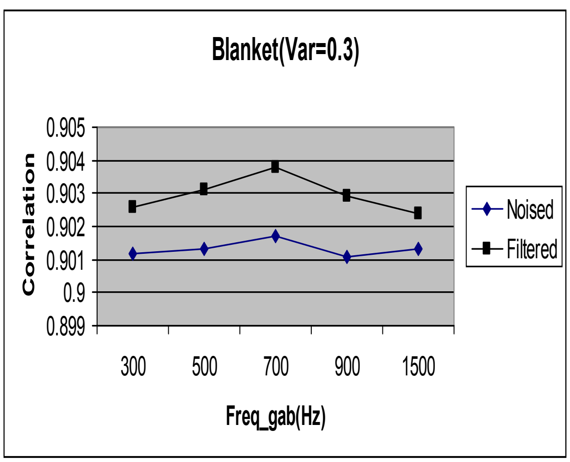
POISSON NOISE

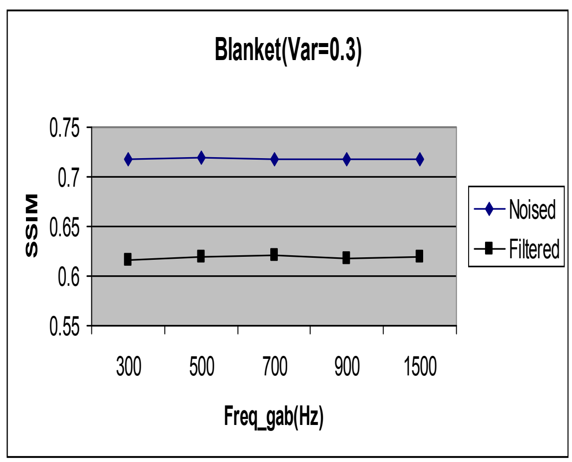


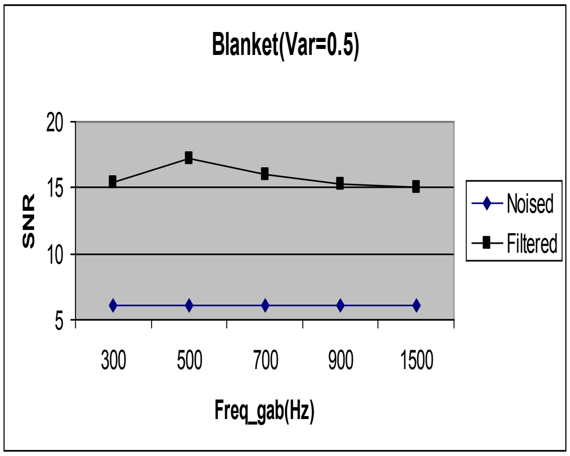


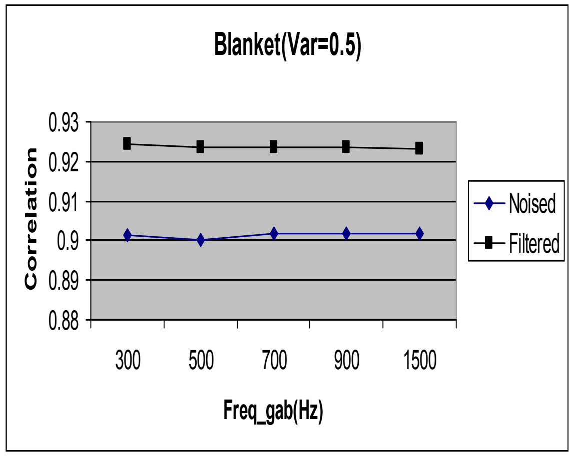


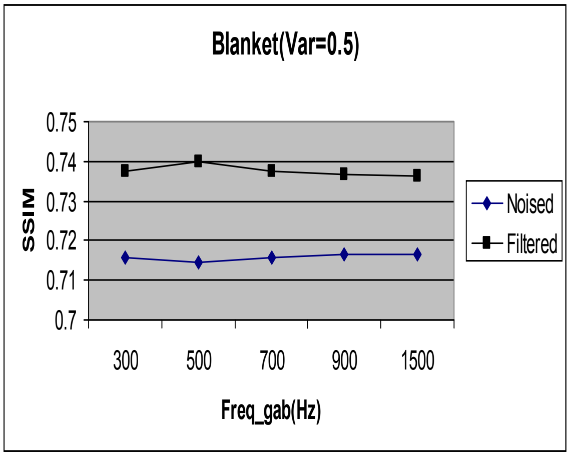


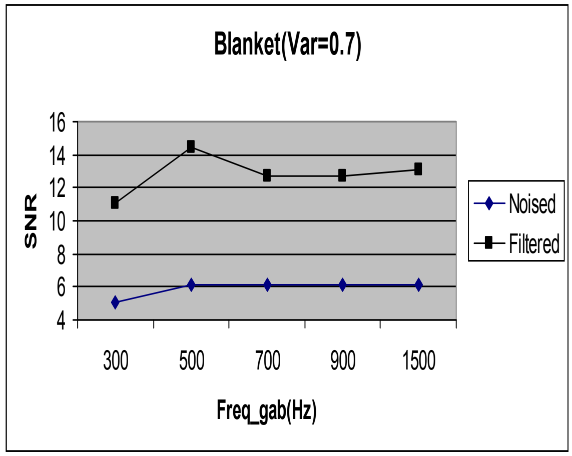


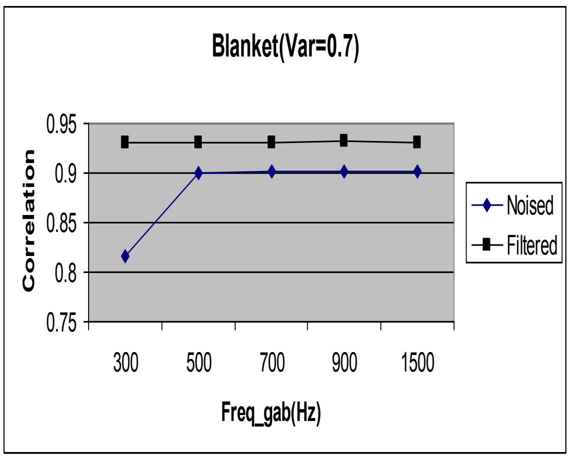


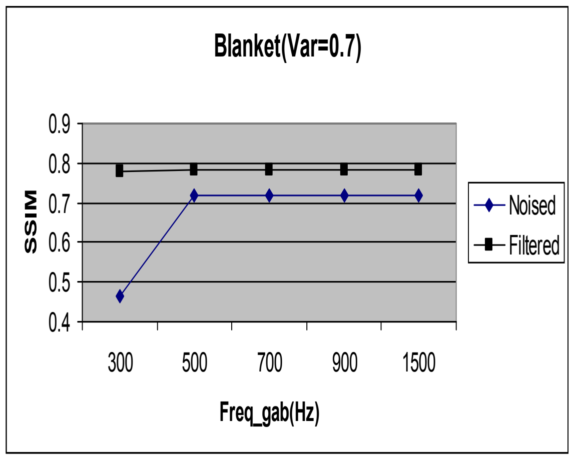

Blanket(Var=0.5)
SNR
20
15
10
5
300
500
700
900
1500
Freq_gab(Hz)
Noised
Filtered


Blanket(Var=0.5)
Correlation
0.93
0.92
0.91
0.9
0.89
0.88
300
500
700
900
1500
Freq_gab(Hz)
Noised
Filtered

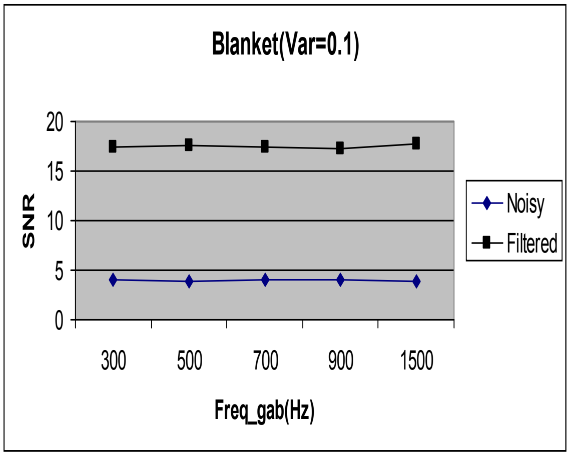

Blanket(Var=0.5)
SSIM
0.75
0.74
0.73
0.72
0.71
0.7
300
500
700
900
1500
Freq_gab(Hz)
Noised
Filtered

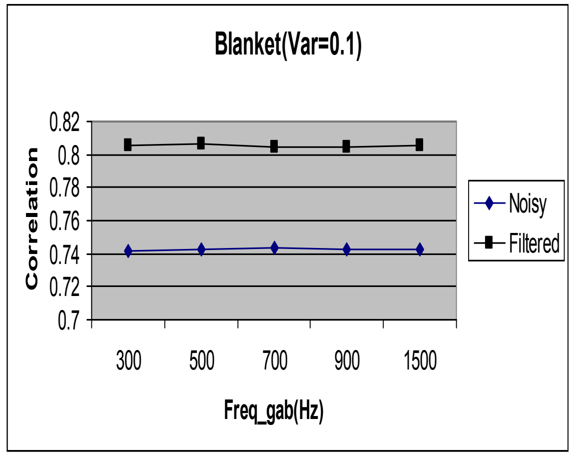

Blanket(Var=0.7)
SNR
16
14
12
10
8
6
4
300
500
700
900
1500
Freq_gab(Hz)
Noised
Filtered

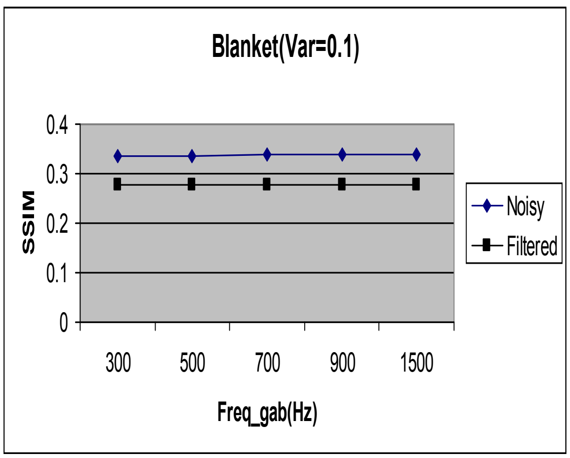

Blanket(Var=0.7)
Correlation
0.95
0.9
0.85
0.8
0.75
300
500
700
900
1500
Freq_gab(Hz)
Noised
Filtered

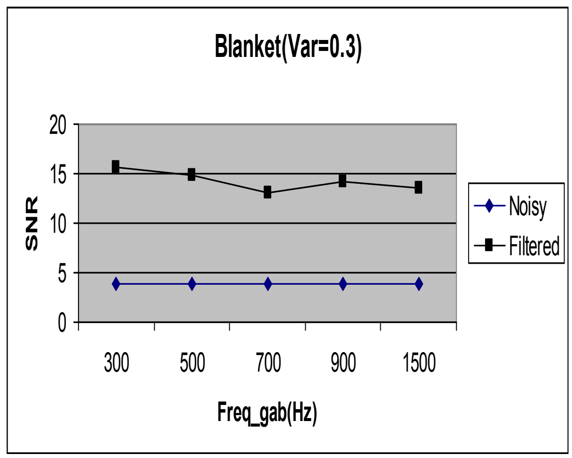

Blanket(Var=0.7)
SSIM
0.9
0.8
0.7
0.6
0.5
0.4
300
500
700
900
1500
Freq_gab(Hz)
Noised
Filtered

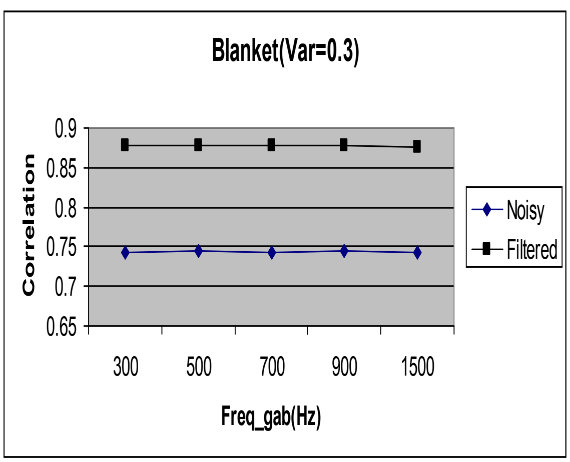
SPECKLE NOISE

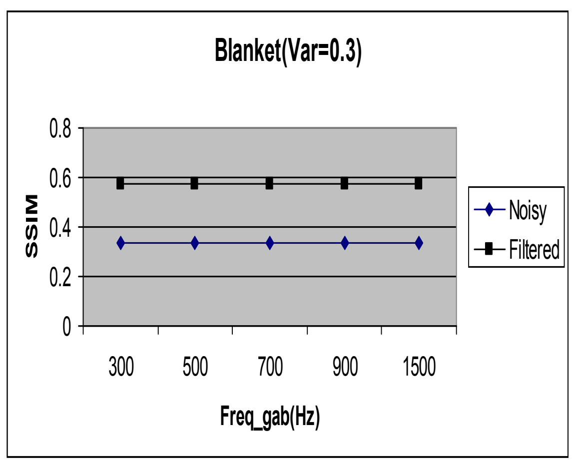


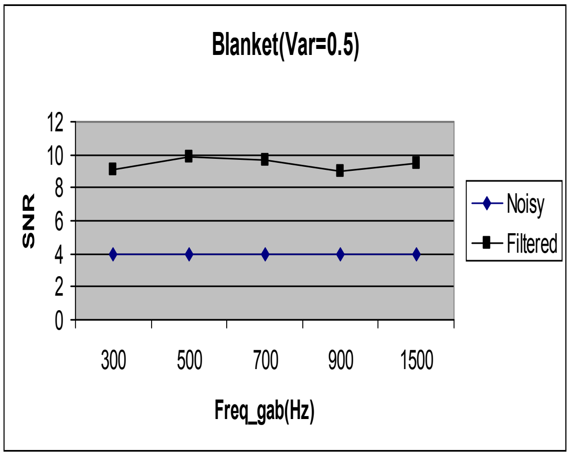


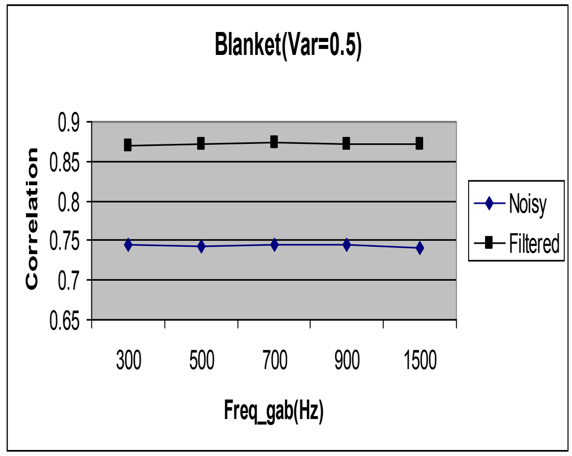


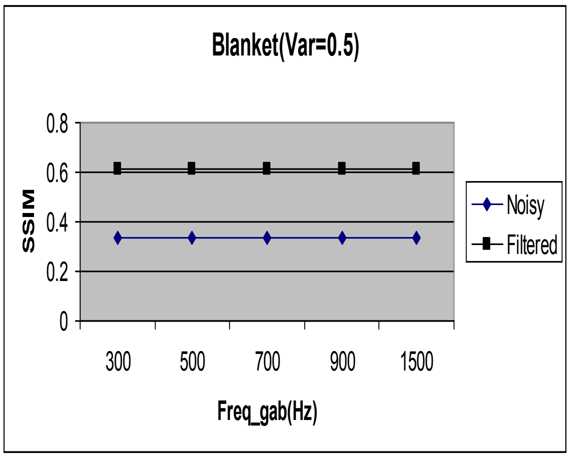


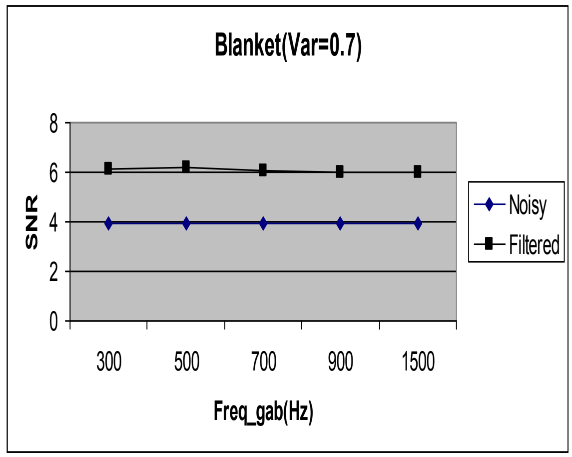


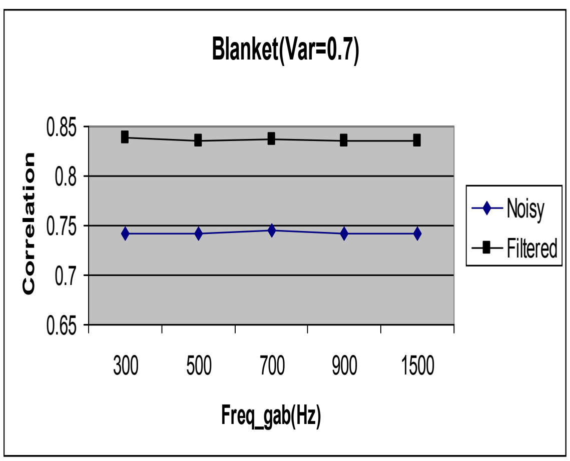


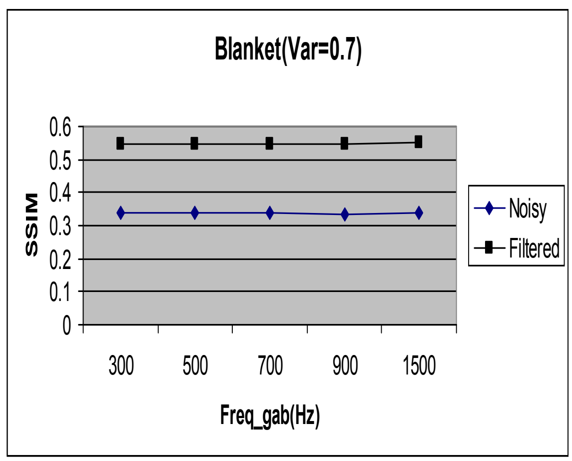

Blanket(Var=0.5)
SNR
12
10
8
6
4
2
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Blanket(Var=0.5)
Correlation
0.9
0.85
0.8
0.75
0.7
0.65
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

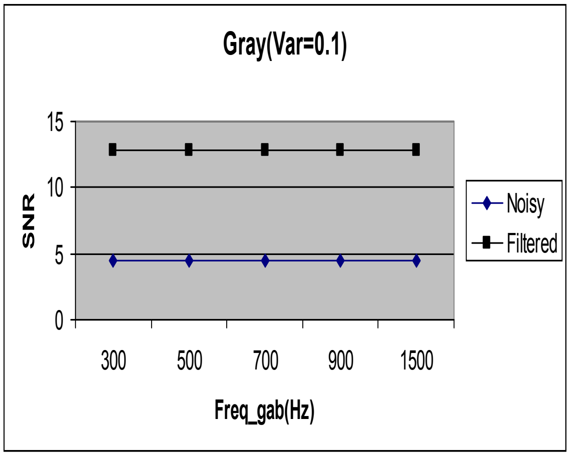

Blanket(Var=0.5)
SSIM
0.8
0.6
0.4
0.2
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

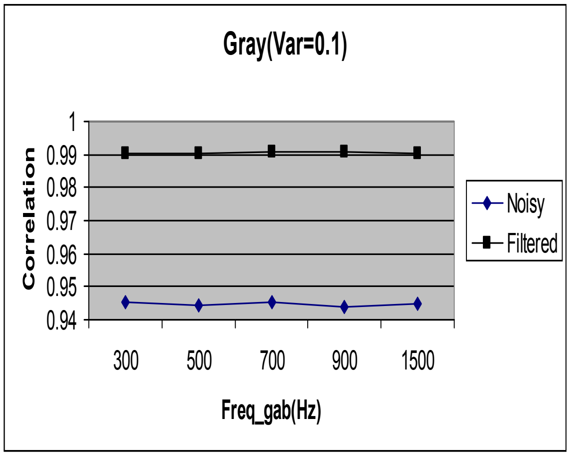

Blanket(Var=0.7)
SNR
8
6
4
2
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

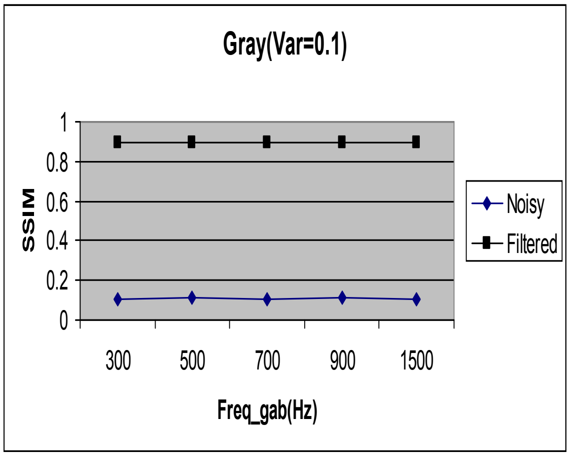

Blanket(Var=0.7)
Correlation
0.85
0.8
0.75
0.7
0.65
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

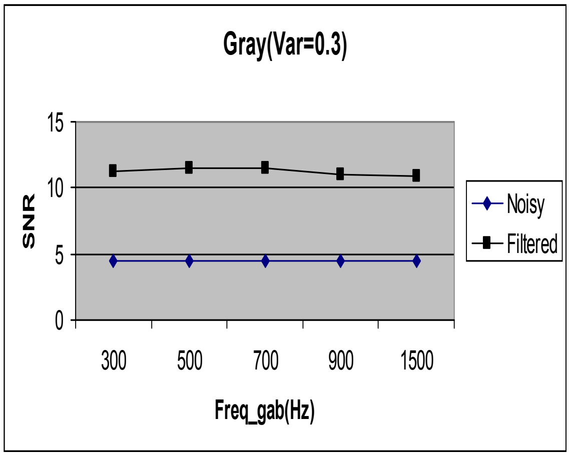

Blanket(Var=0.7)
SSIM
0.6
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

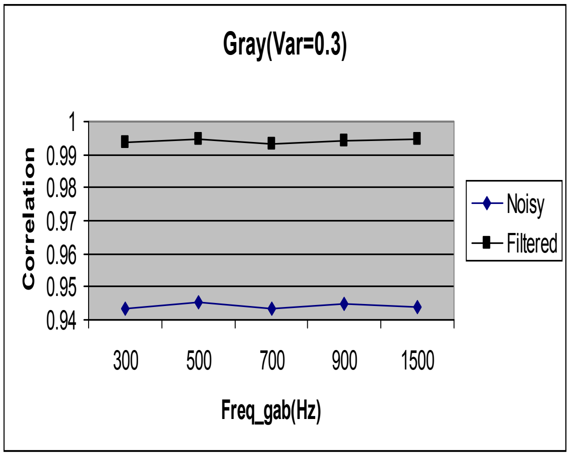
GAUSSIAN NOISE

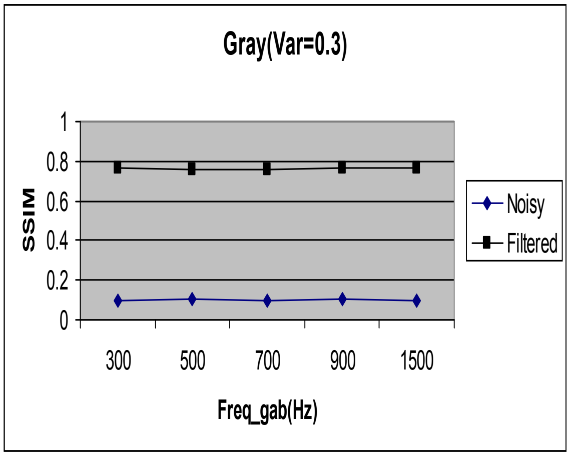


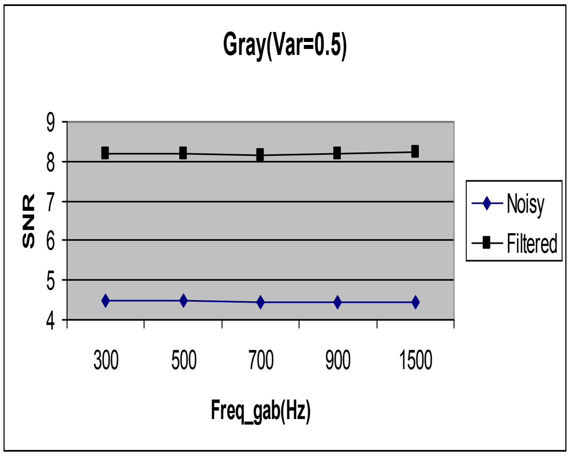


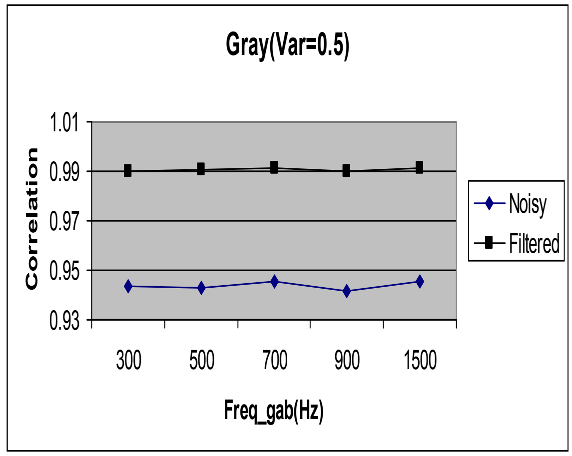


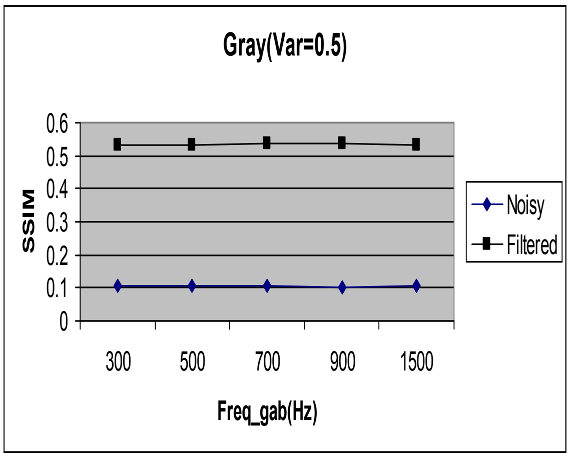


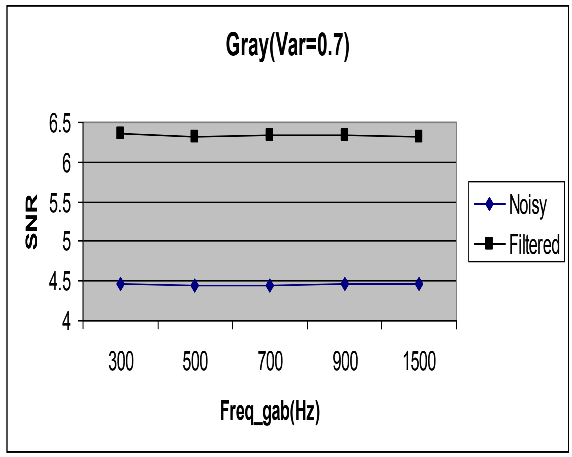


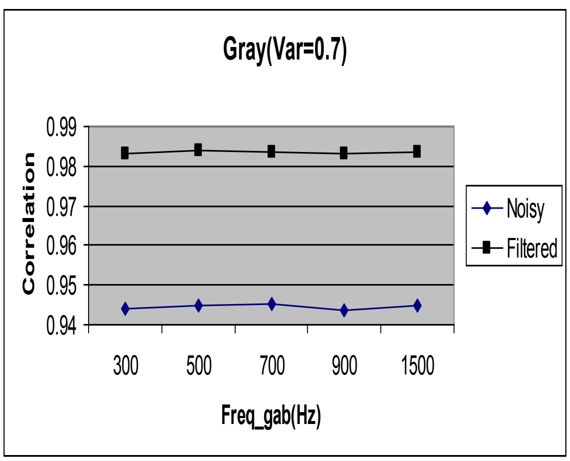


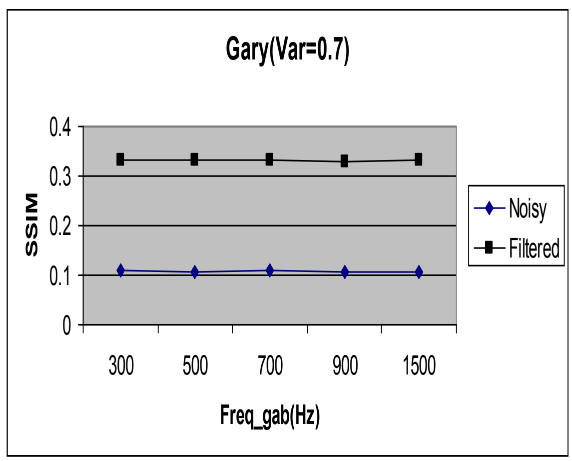

Gray(Var=0.5)
SNR
9
8
7
6
5
4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Gray(Var=0.5)
Correlation
1.01
0.99
0.97
0.95
0.93
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

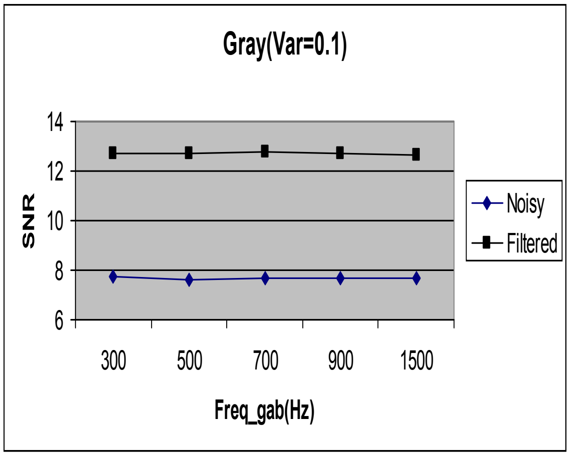

Gray(Var=0.5)
SSIM
0.6
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

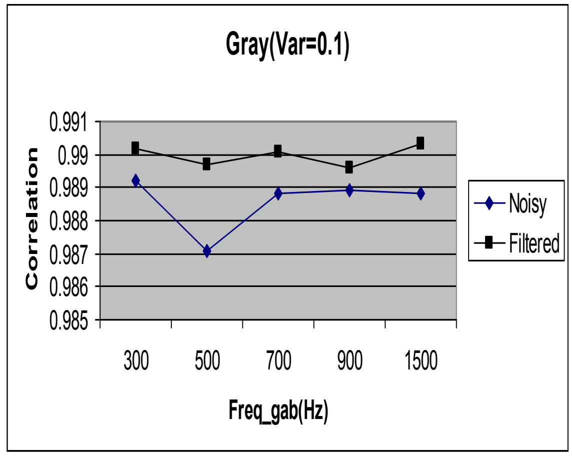

Gray(Var=0.7)
SNR
6.5
6
5.5
5
4.5
4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

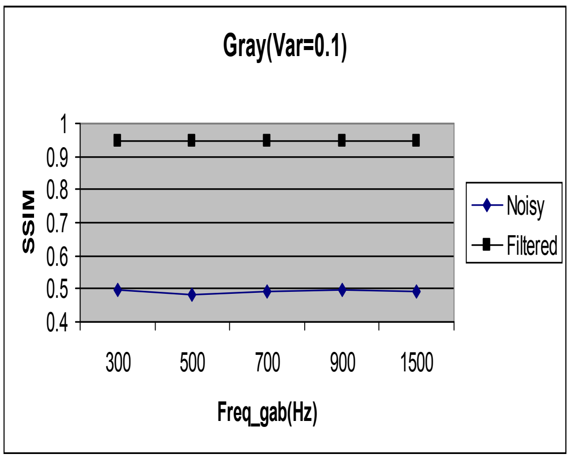

Gray(Var=0.7)
Correlation
0.99
0.98
0.97
0.96
0.95
0.94
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

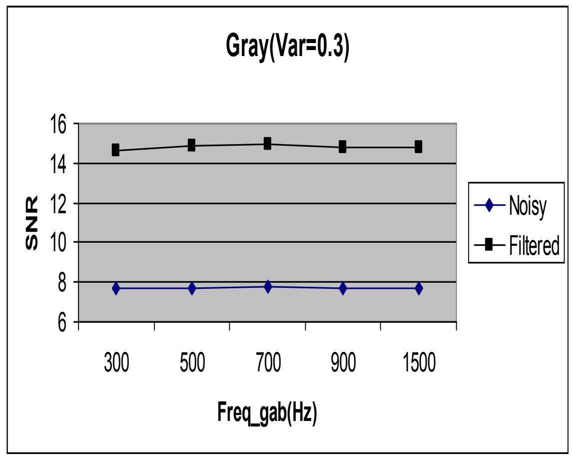

Gary(Var=0.7)
SSIM
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

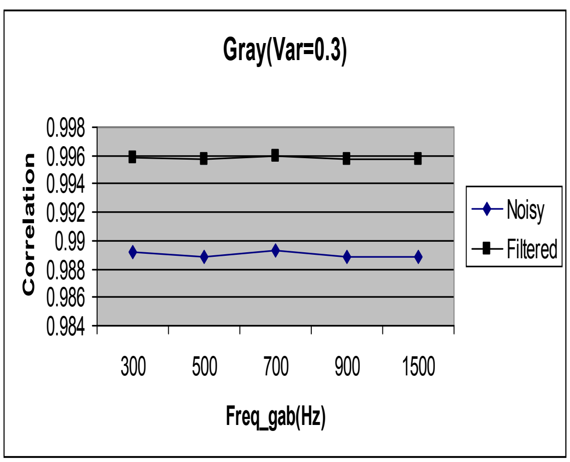
POISSON NOISE

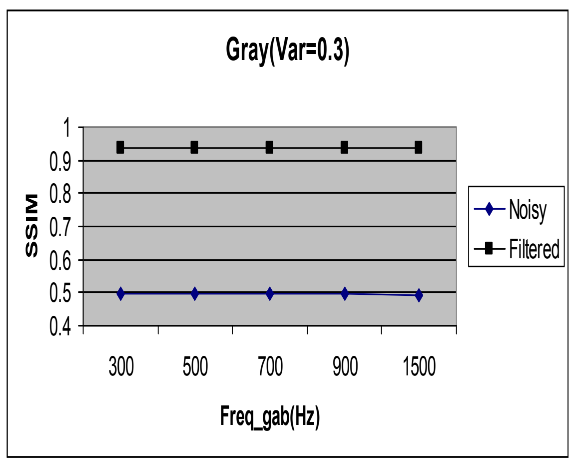


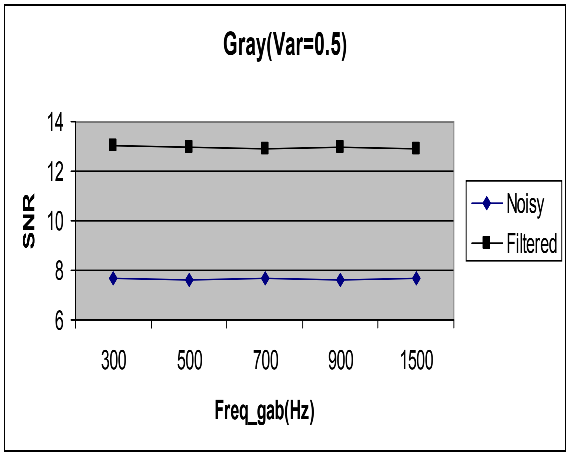


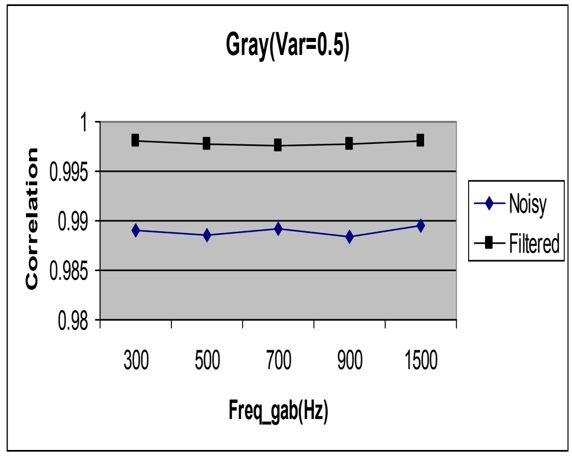


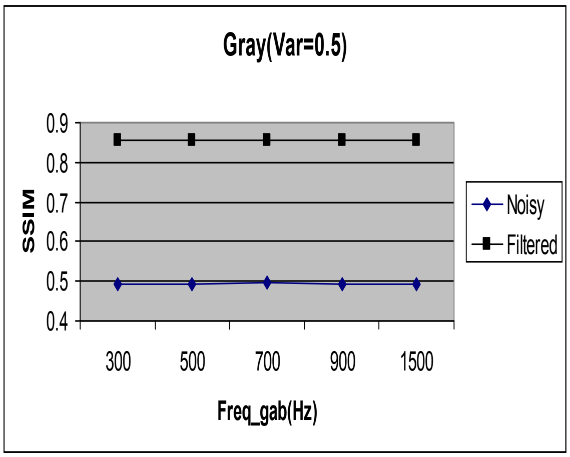


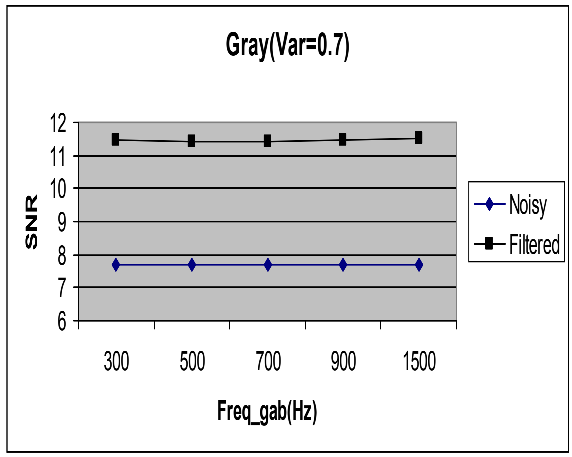


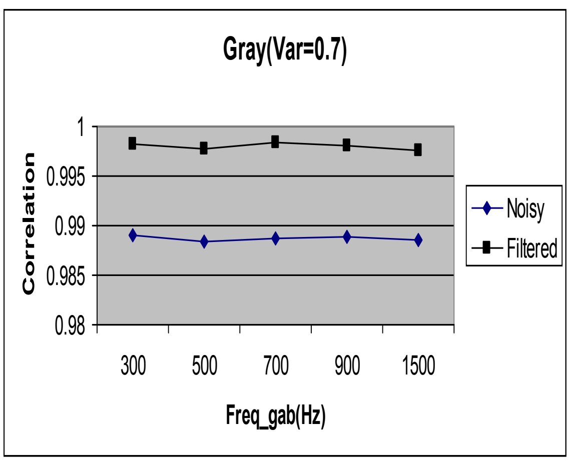


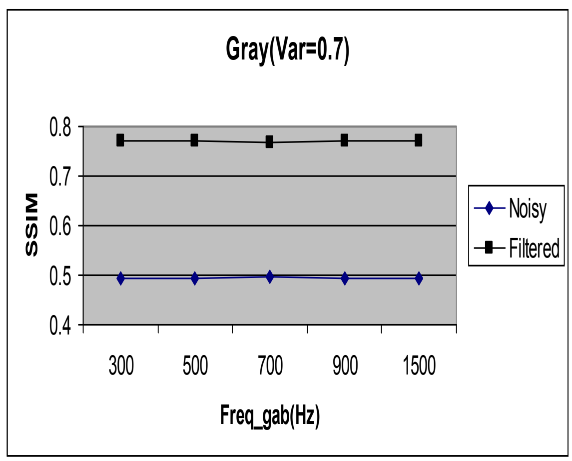

Gray(Var=0.5)
SNR
14
12
10
8
6
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Gray(Var=0.5)
Correlation
1
0.995
0.99
0.985
0.98
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

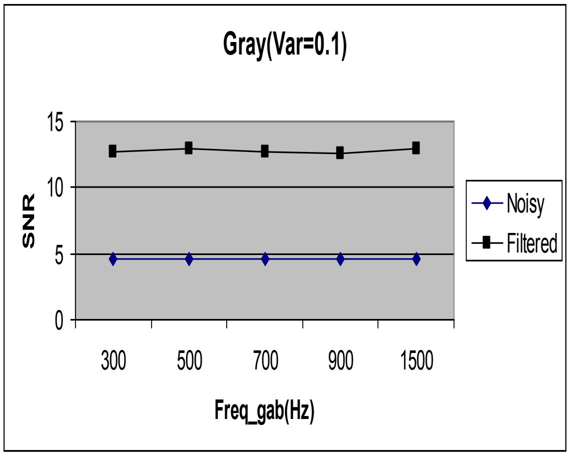

Gray(Var=0.5)
SSIM
0.9
0.8
0.7
0.6
0.5
0.4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

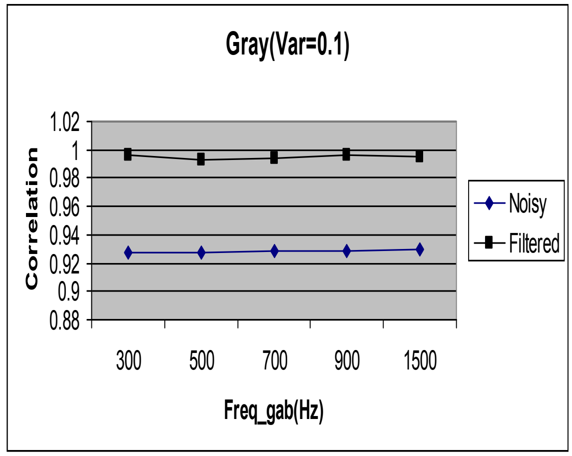

Gray(Var=0.7)
SNR
12
11
10
9
8
7
6
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

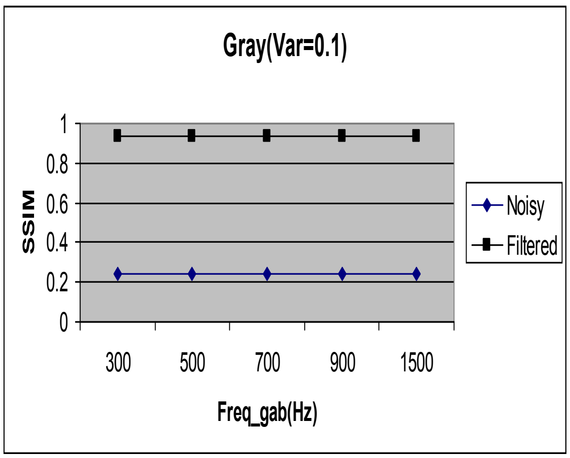

Gray(Var=0.7)
Correlation
1
0.995
0.99
0.985
0.98
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

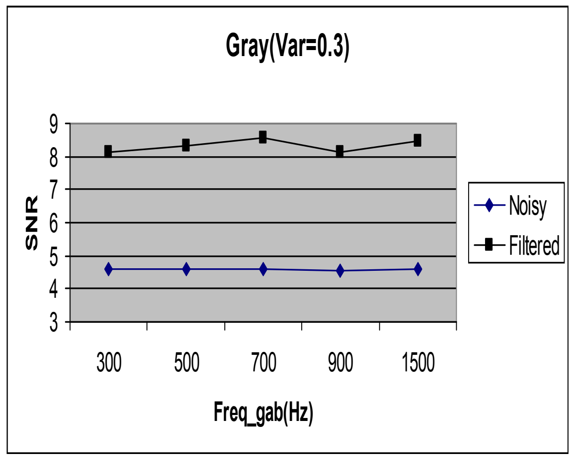

Gray(Var=0.7)
SSIM
0.8
0.7
0.6
0.5
0.4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

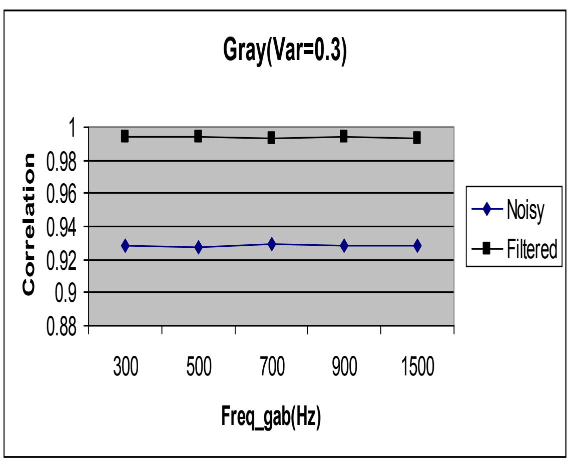
SPECKLE NOISE

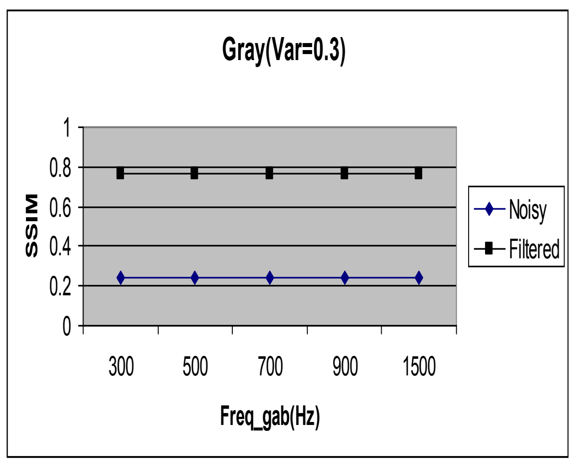


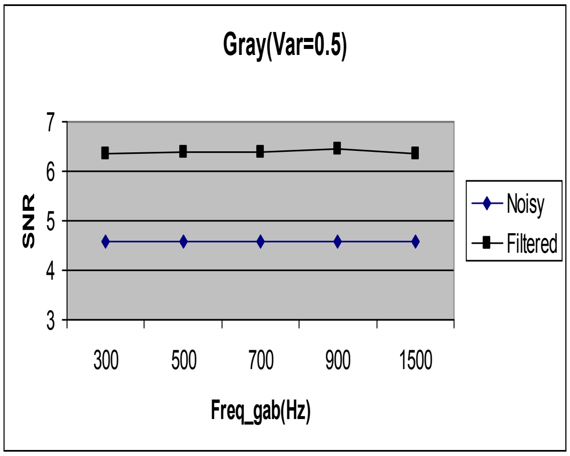


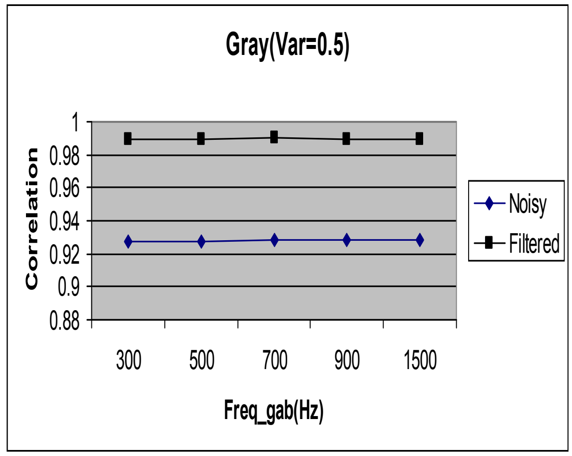


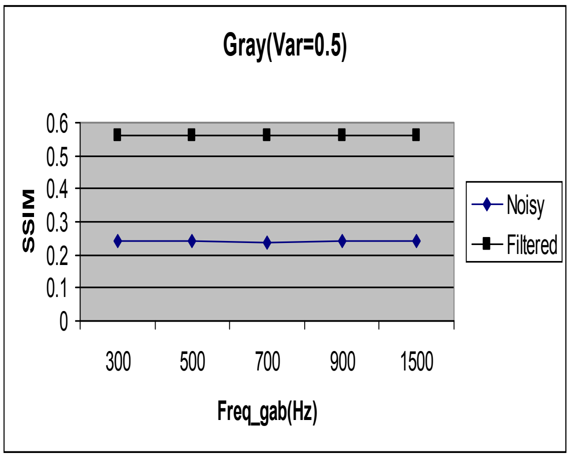


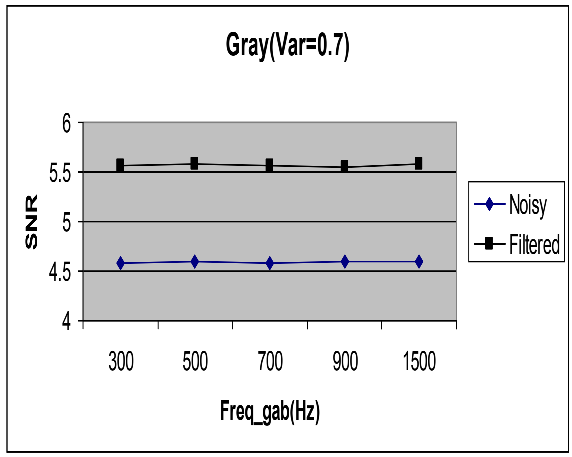


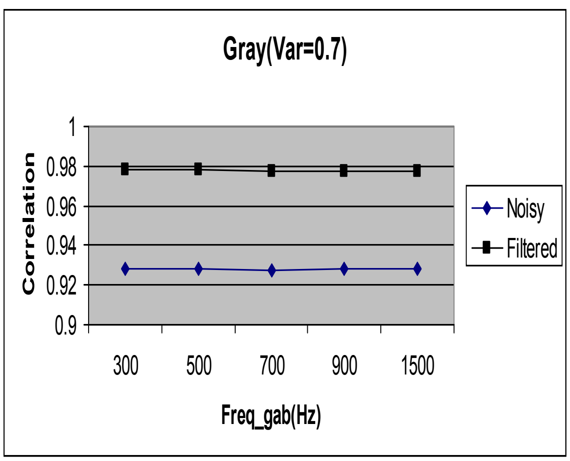


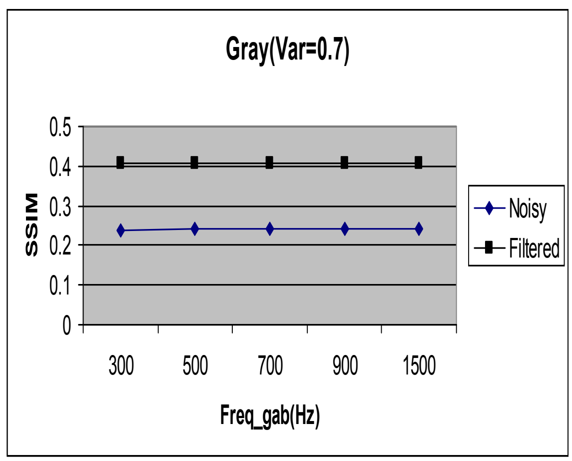

Gray(Var=0.5)
SNR
7
6
5
4
3
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Gray(Var=0.5)
Correlation
1
0.98
0.96
0.94
0.92
0.9
0.88
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

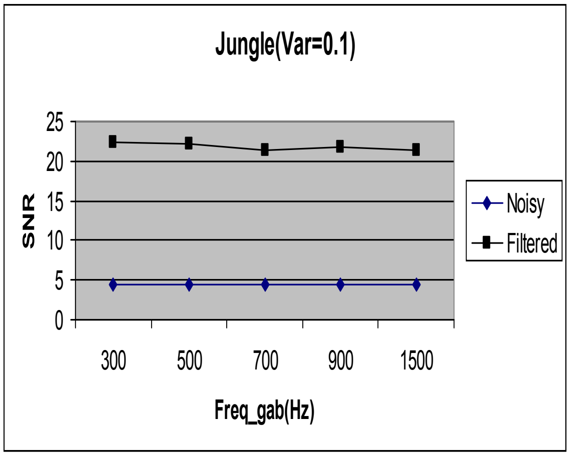

Gray(Var=0.5)
SSIM
0.6
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

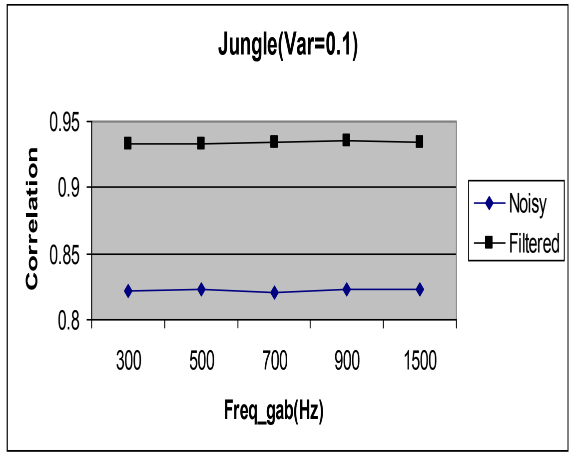

Gray(Var=0.7)
SNR
6
5.5
5
4.5
4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

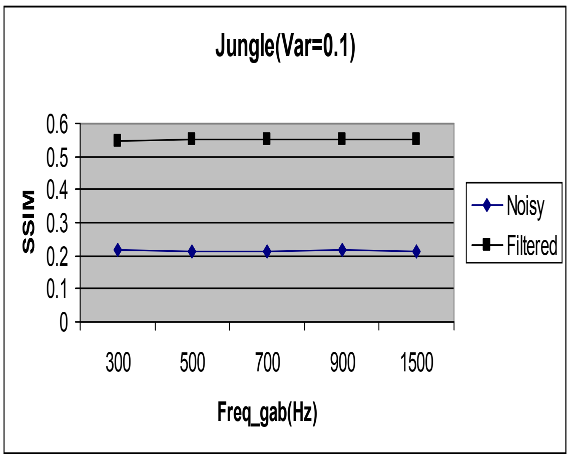

Gray(Var=0.7)
Correlation
1
0.98
0.96
0.94
0.92
0.9
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

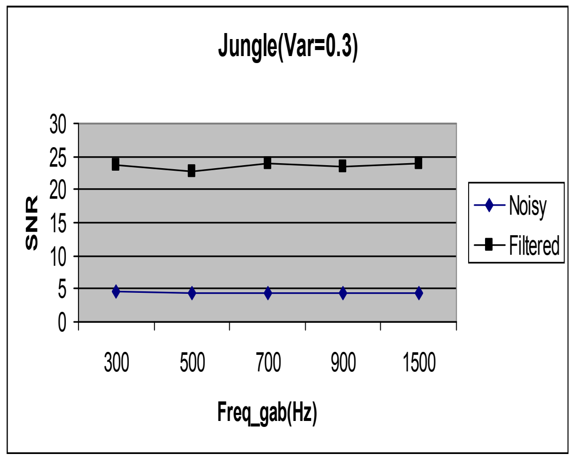

Gray(Var=0.7)
SSIM
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

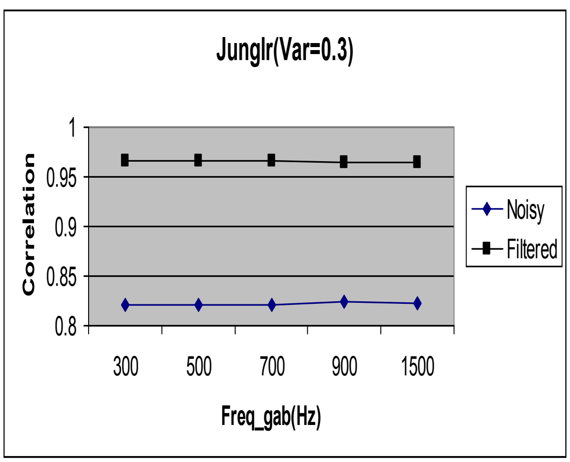
GAUSSIAN NOISE

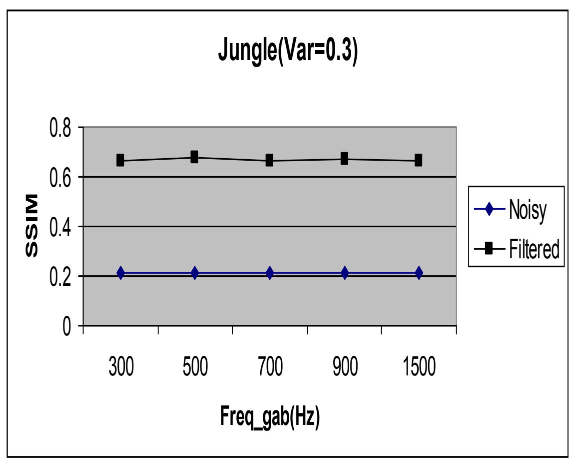


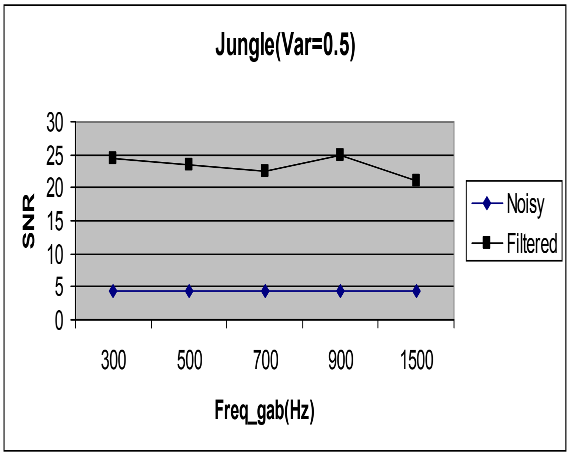
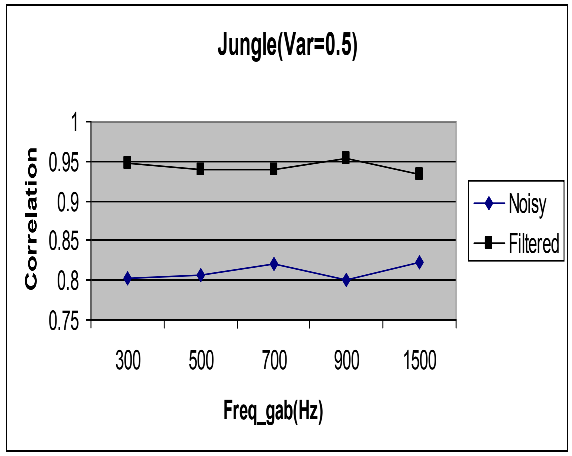
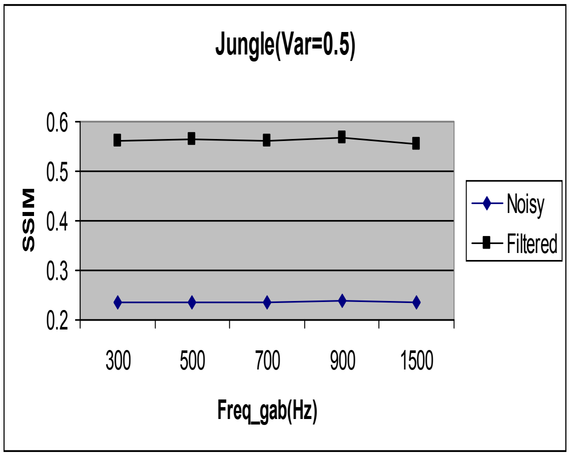
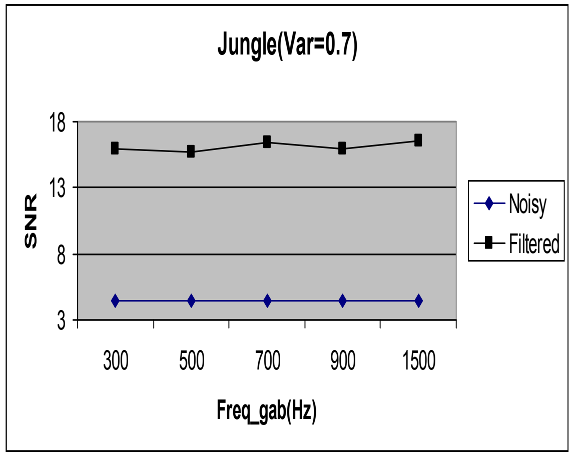
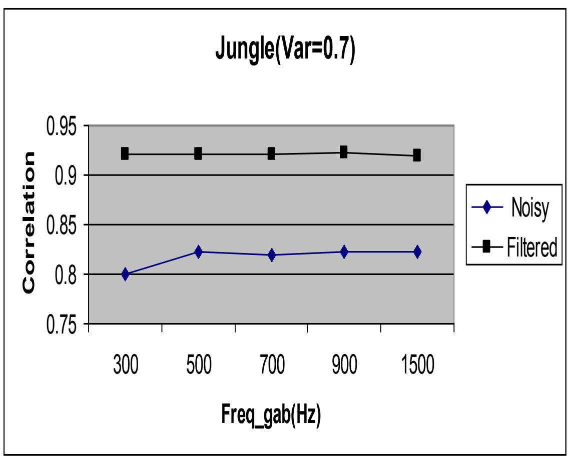
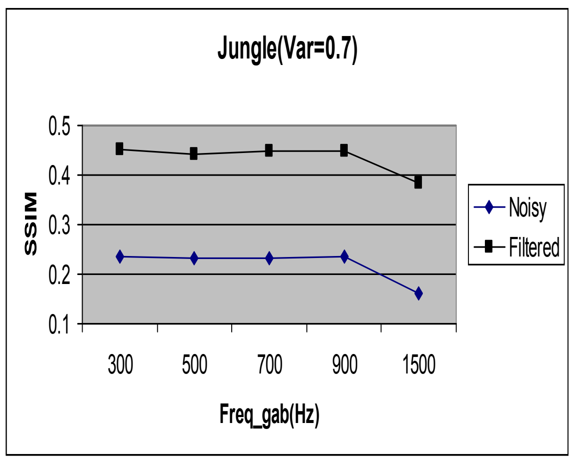




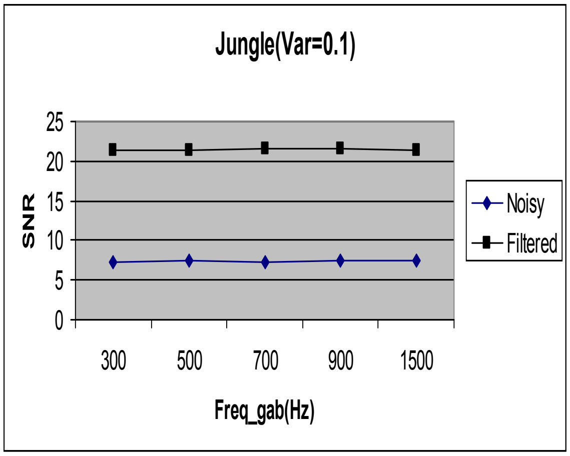


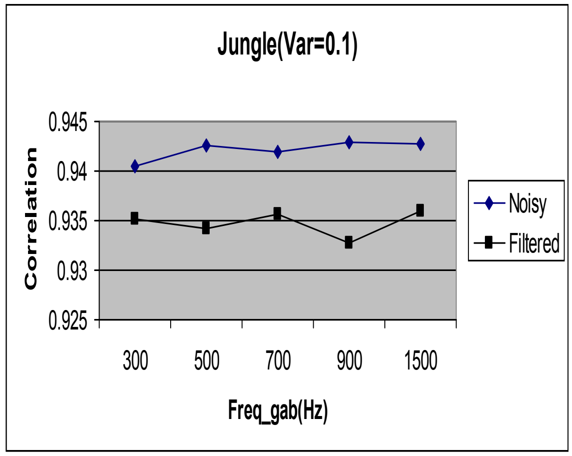


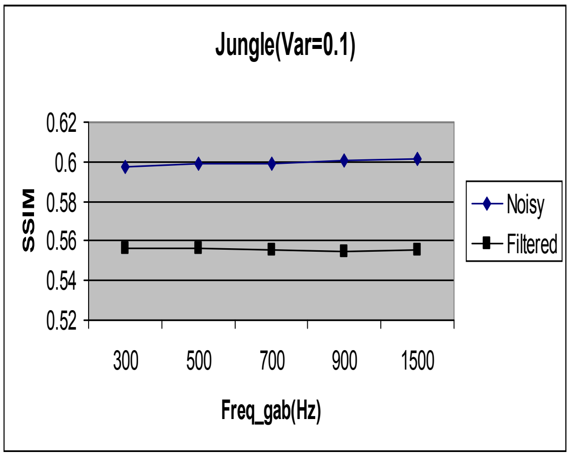


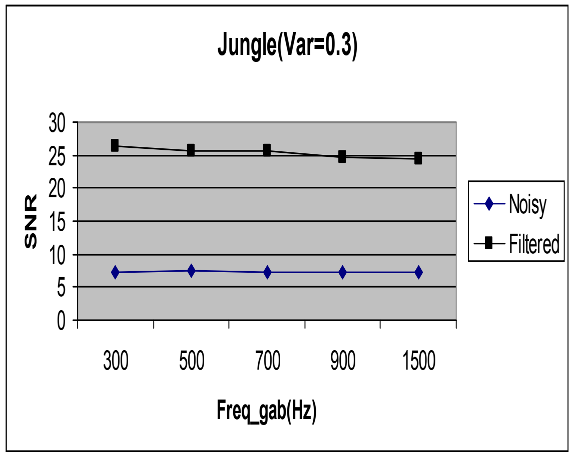

Jungle(Var=0.5)
SNR
30
25
20
15
10
5
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

Jungle(Var=0.5)
Correlation
1
0.95
0.9
0.85
0.8
0.75
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

Jungle(Var=0.5)
SSIM
0.6
0.5
0.4
0.3
0.2
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

Jungle(Var=0.7)
SNR
18
13
8
3
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

Jungle(Var=0.7)
Correlation
0.95
0.9
0.85
0.8
0.75
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

Jungle(Var=0.7)
SSIM
0.5
0.4
0.3
0.2
0.1
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

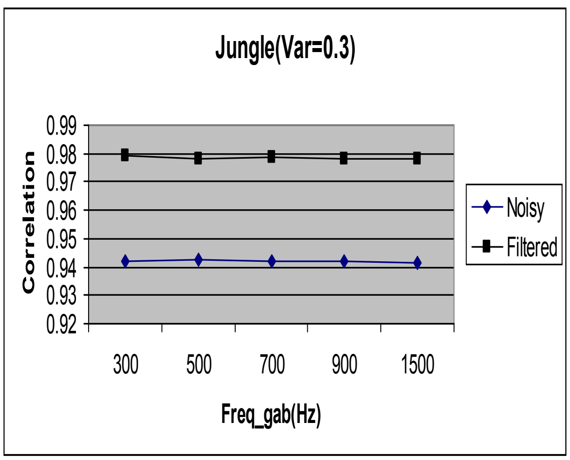
POISSON NOISE

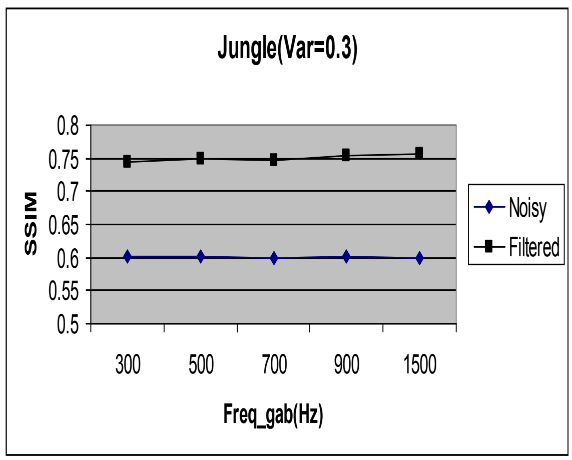


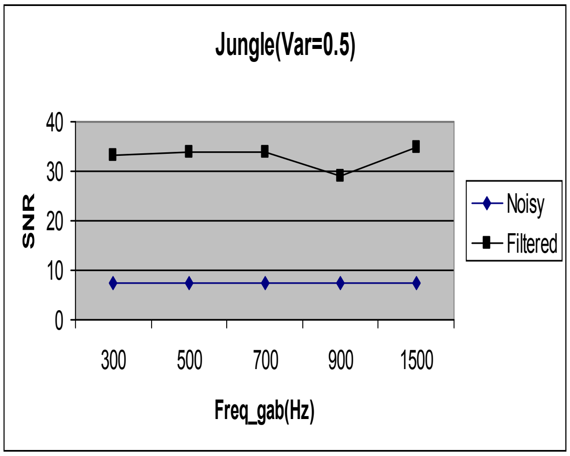


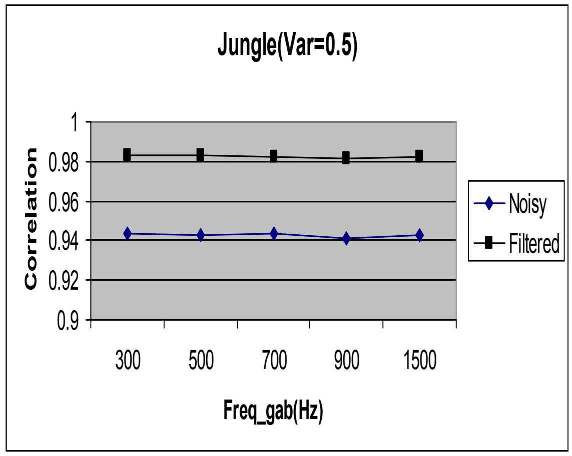


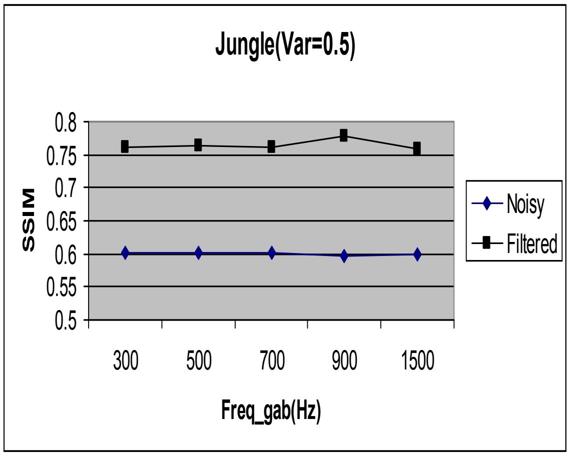


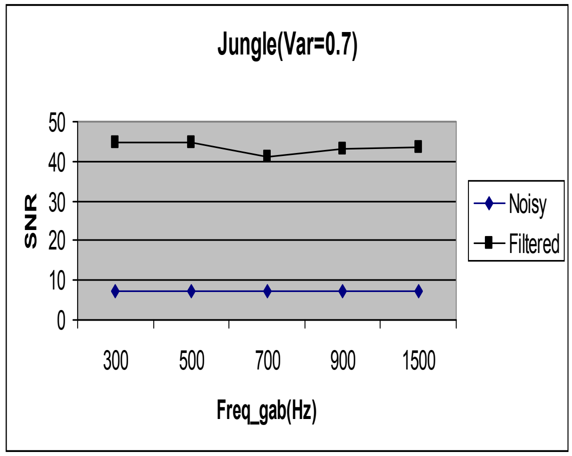


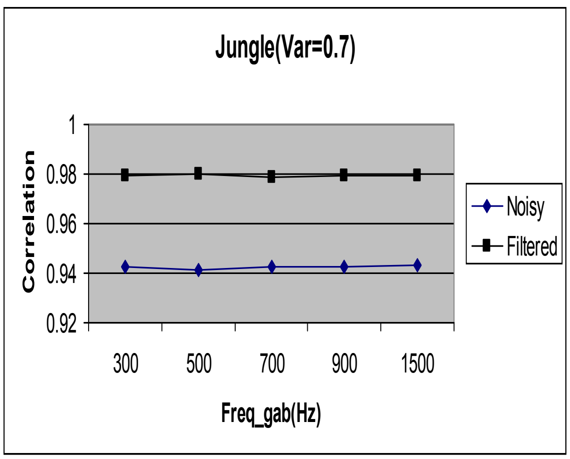


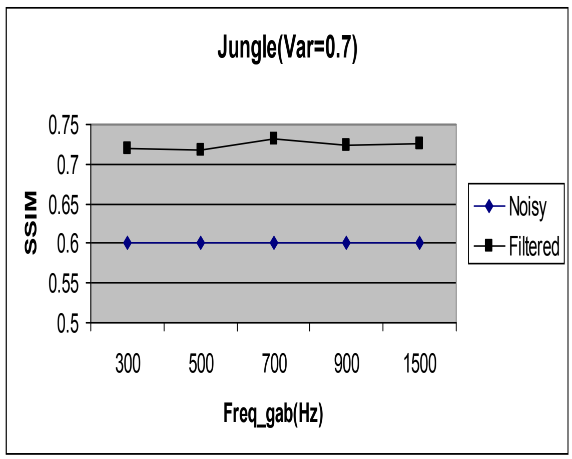

Jungle(Var=0.5)
SNR
40
30
20
10
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Jungle(Var=0.5)
Correlation
1
0.98
0.96
0.94
0.92
0.9
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

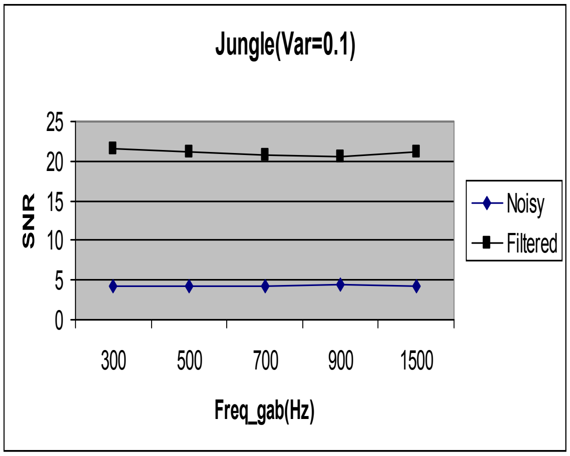

Jungle(Var=0.5)
SSIM
0.8
0.75
0.7
0.65
0.6
0.55
0.5
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

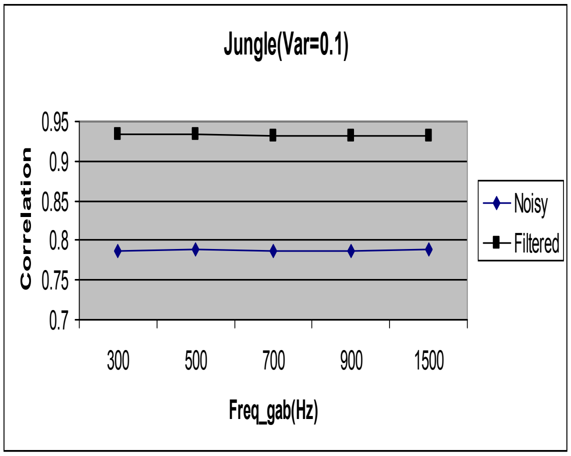

Jungle(Var=0.7)
SNR
50
40
30
20
10
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

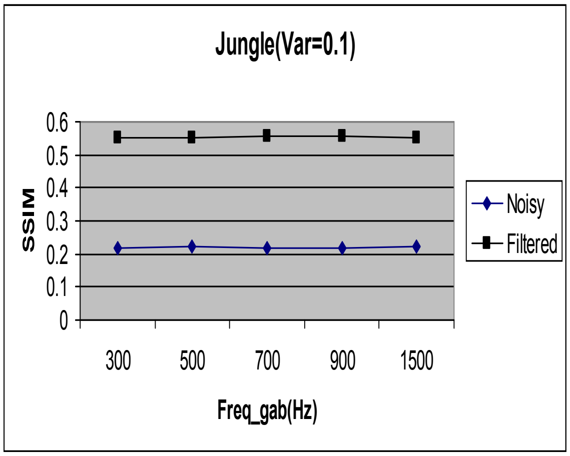

Jungle(Var=0.7)
Correlation
1
0.98
0.96
0.94
0.92
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

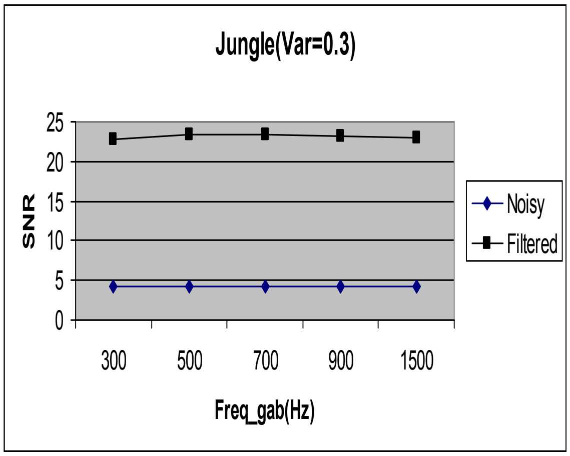

Jungle(Var=0.7)
SSIM
0.75
0.7
0.65
0.6
0.55
0.5
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

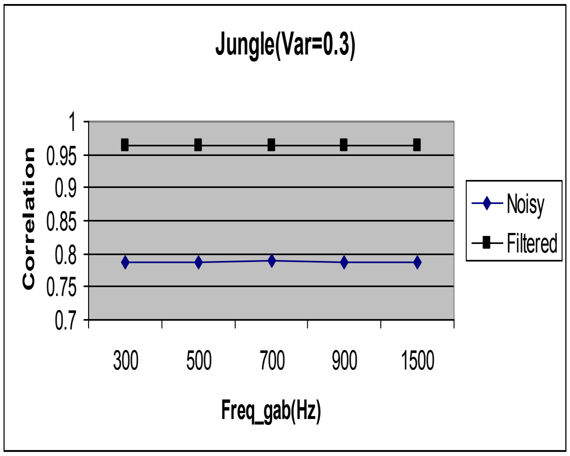
SPECKLE NOISE

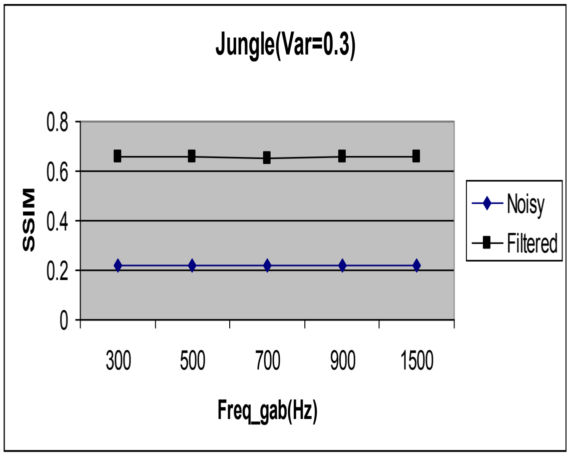


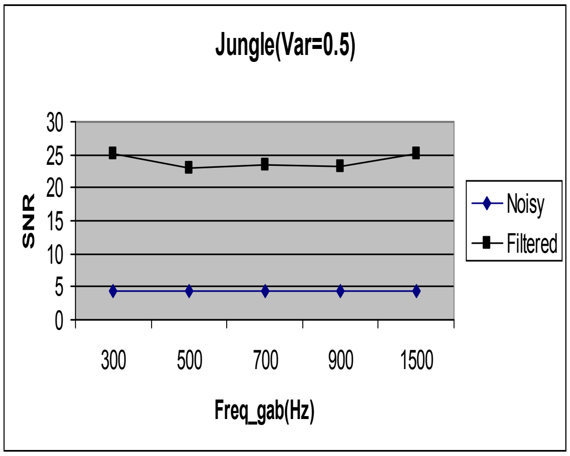


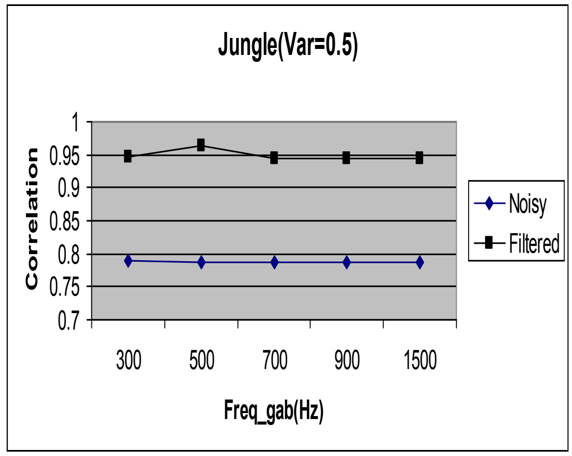


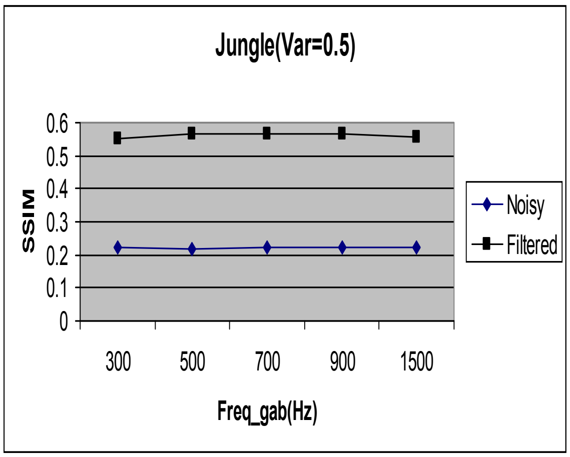


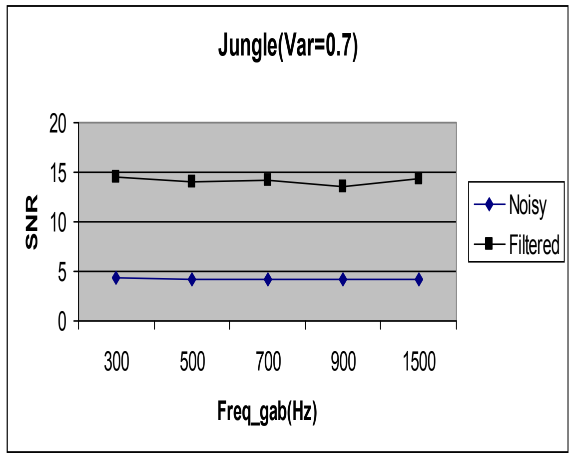


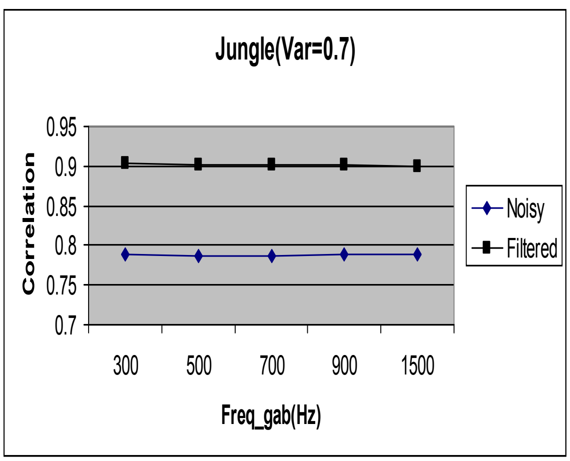


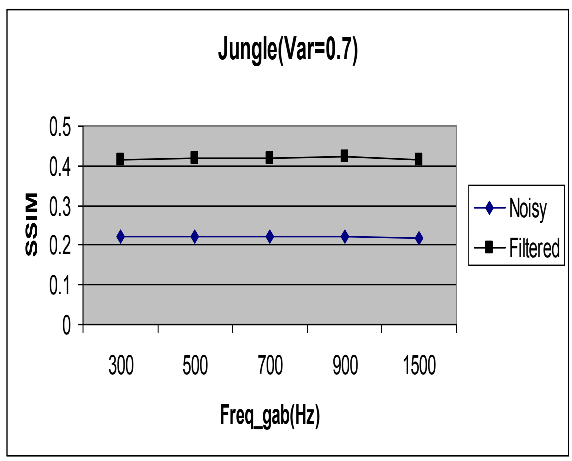

Jungle(Var=0.5)
SNR
30
25
20
15
10
5
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Jungle(Var=0.5)
Correlation
1
0.95
0.9
0.85
0.8
0.75
0.7
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

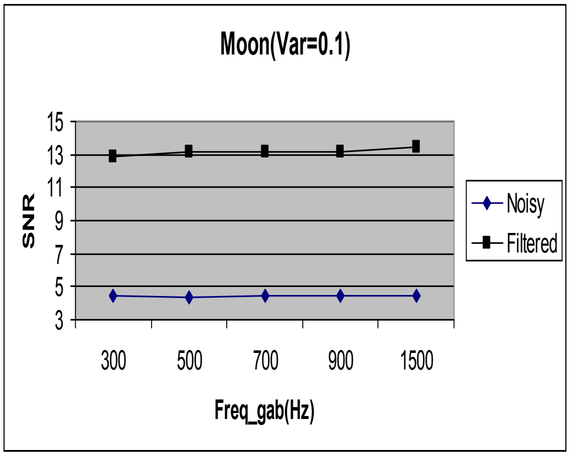

Jungle(Var=0.5)
SSIM
0.6
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

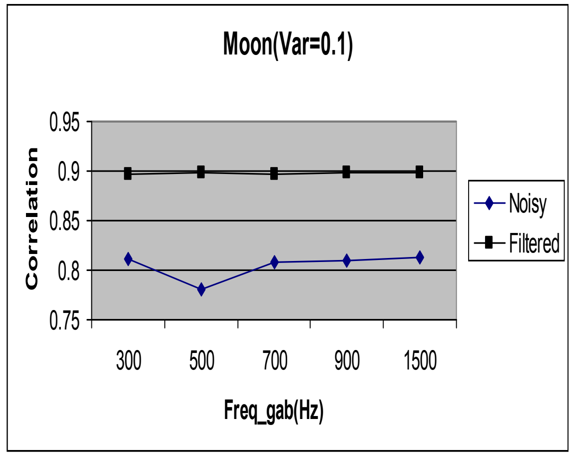

Jungle(Var=0.7)
SNR
20
15
10
5
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

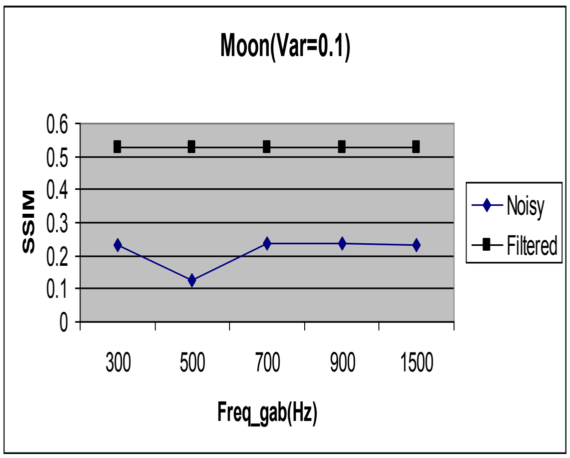

Jungle(Var=0.7)
Correlation
0.95
0.9
0.85
0.8
0.75
0.7
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

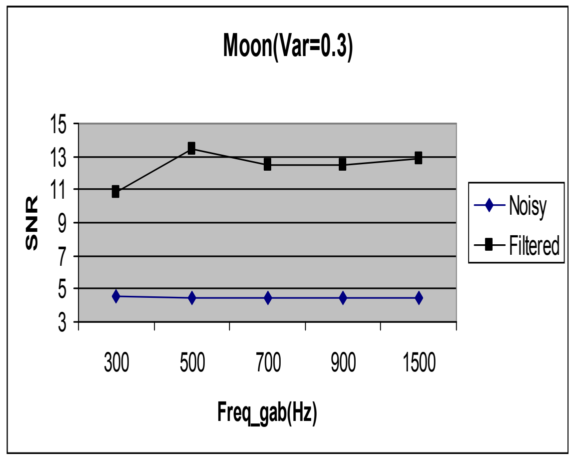

Jungle(Var=0.7)
SSIM
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

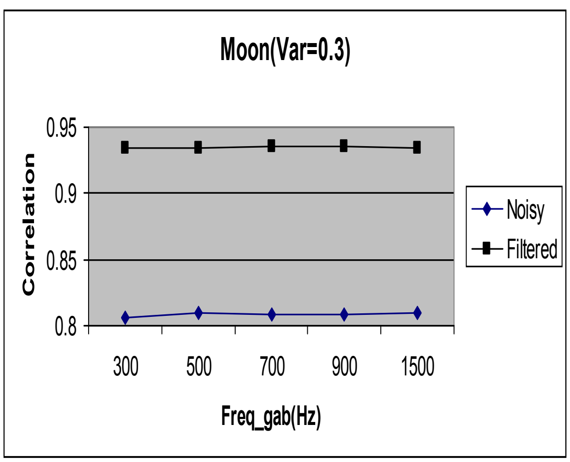
GAUSSIAN NOISE

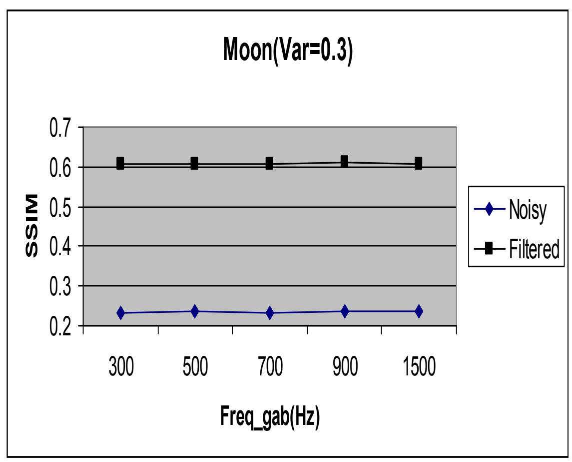


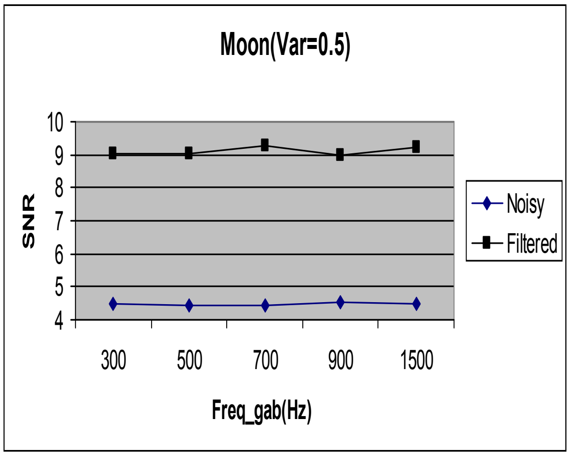


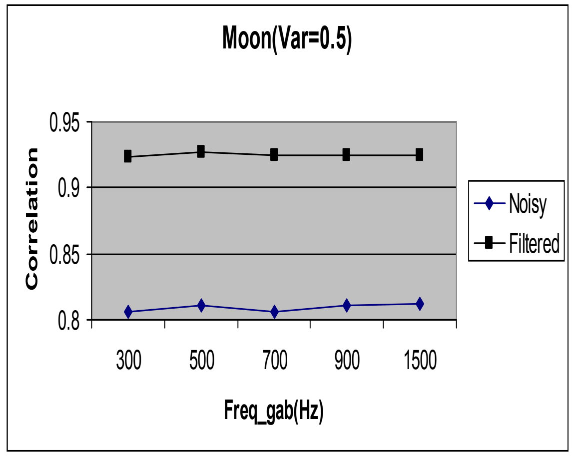


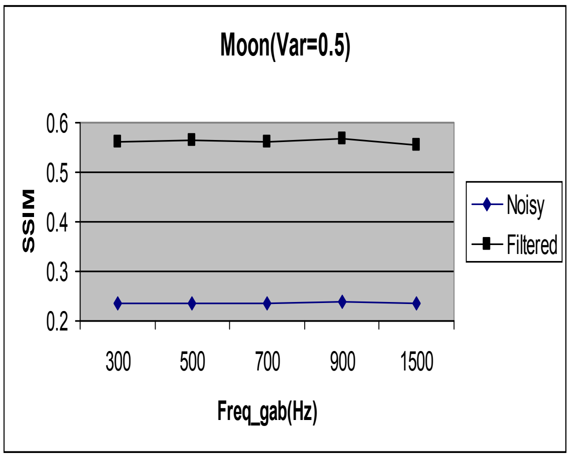


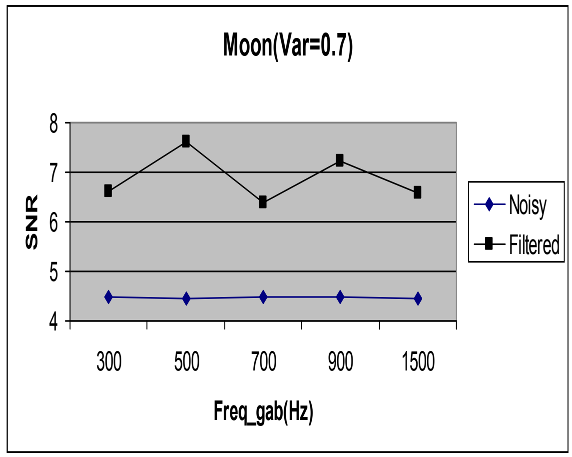


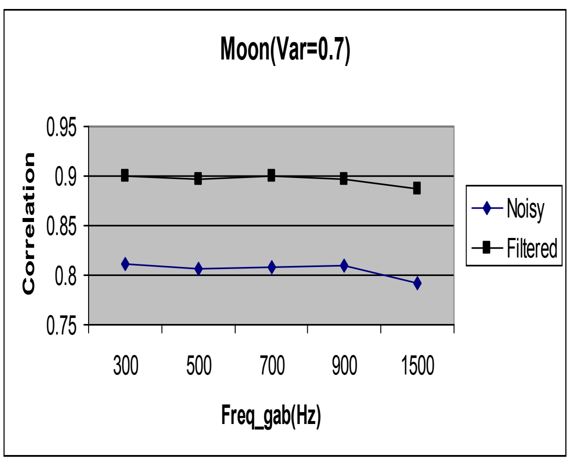


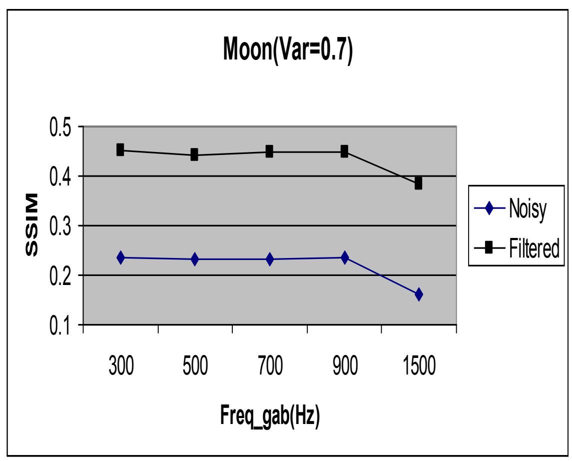

Moon(Var=0.5)
SNR
10
9
8
7
6
5
4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Moon(Var=0.5)
Correlation
0.95
0.9
0.85
0.8
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

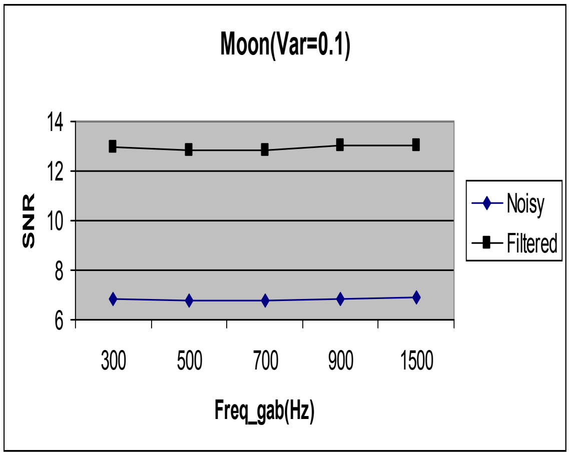

Moon(Var=0.5)
SSIM
0.6
0.5
0.4
0.3
0.2
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

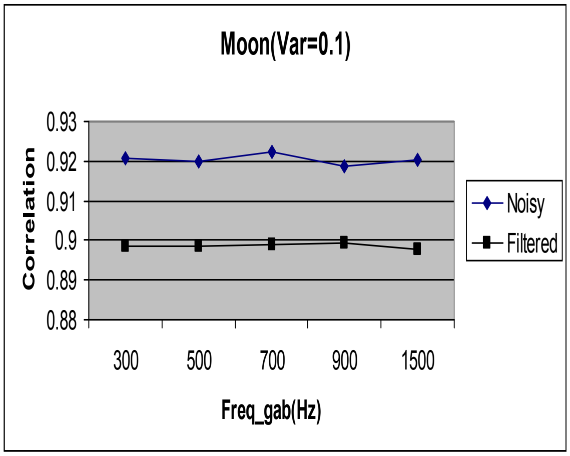

Moon(Var=0.7)
SNR
8
7
6
5
4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

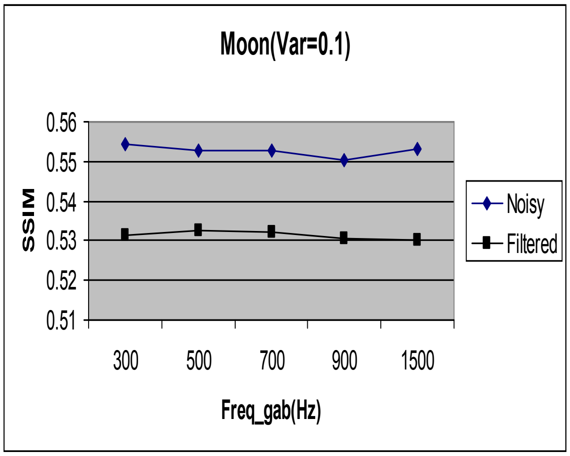

Moon(Var=0.7)
Correlation
0.95
0.9
0.85
0.8
0.75
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

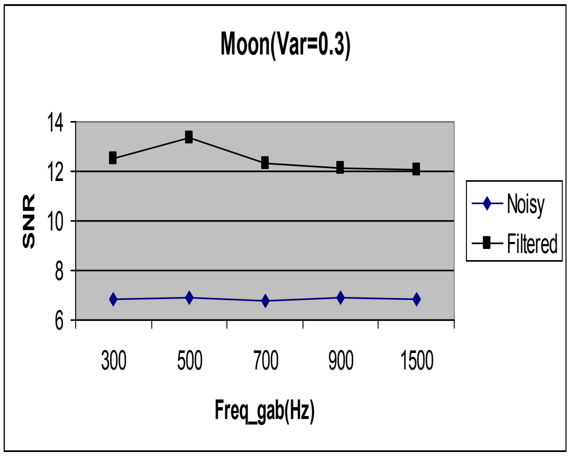

Moon(Var=0.7)
SSIM
0.5
0.4
0.3
0.2
0.1
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

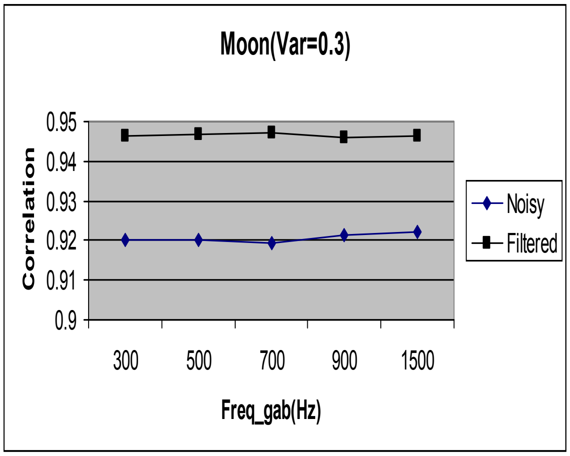
POISSON NOISE

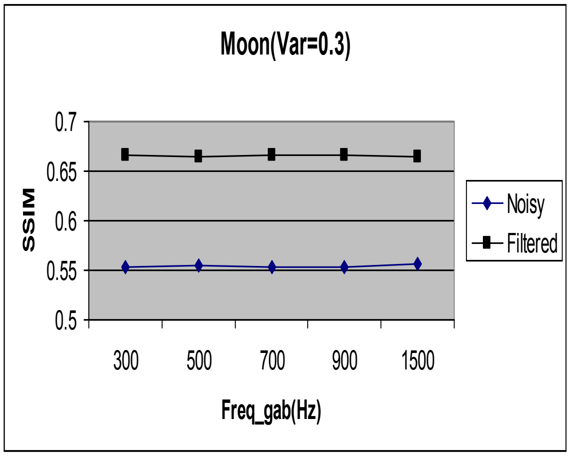


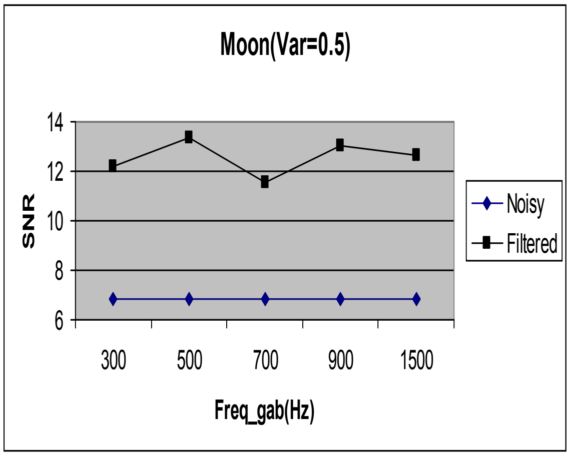


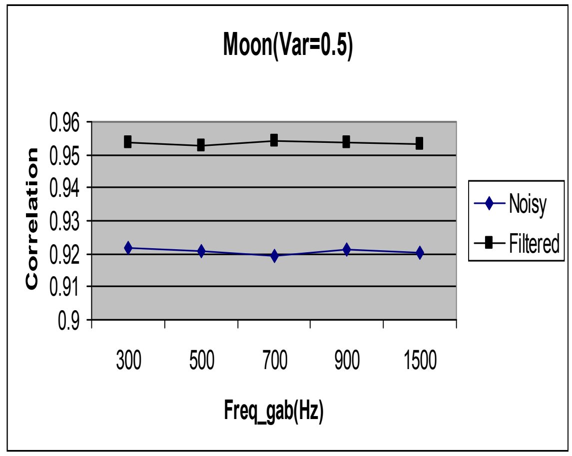


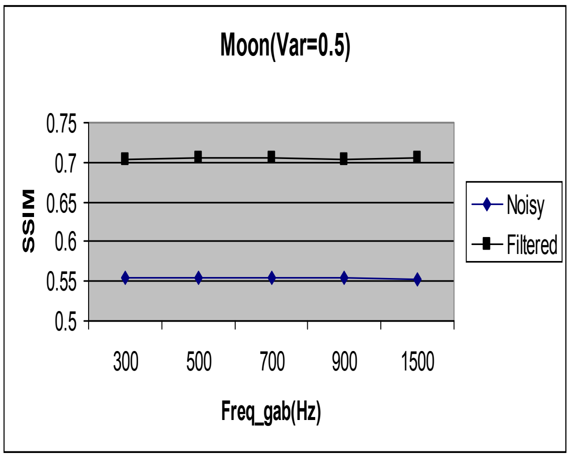


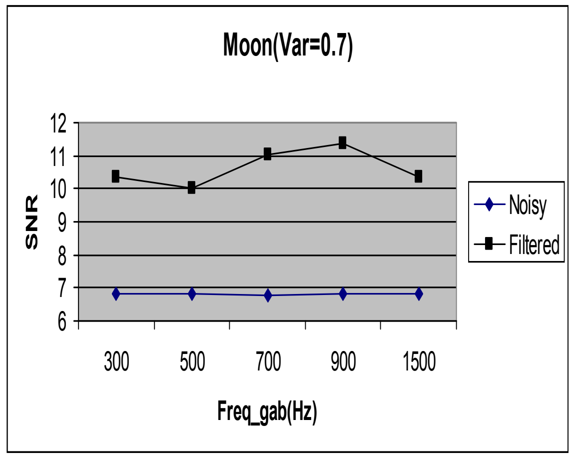


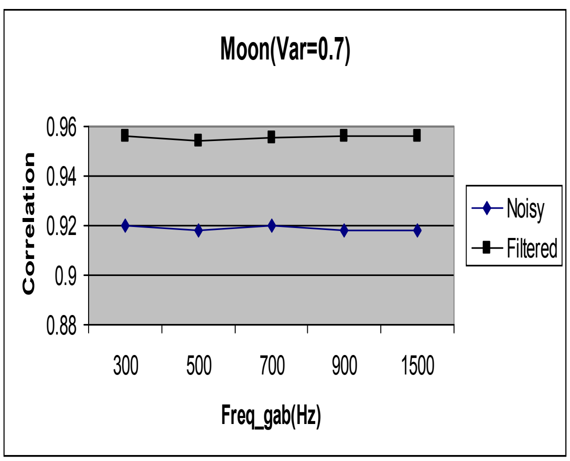


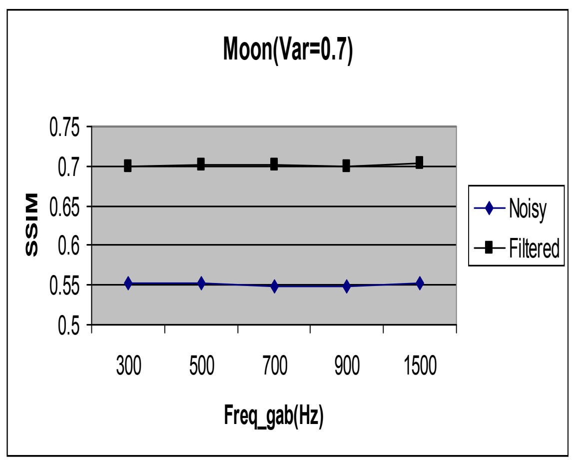

Moon(Var=0.5)
SNR
14
12
10
8
6
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Moon(Var=0.5)
Correlation
0.96
0.95
0.94
0.93
0.92
0.91
0.9
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

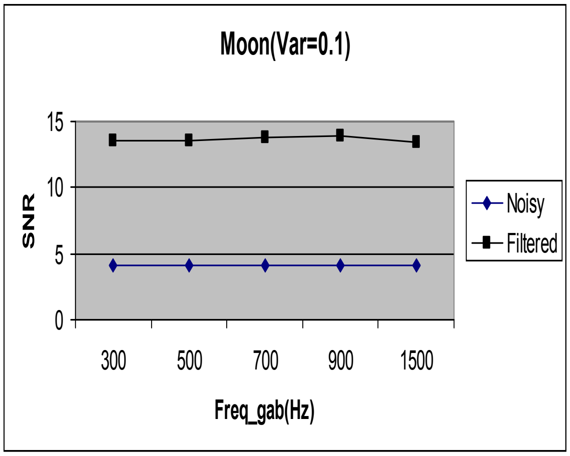

Moon(Var=0.5)
SSIM
0.75
0.7
0.65
0.6
0.55
0.5
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

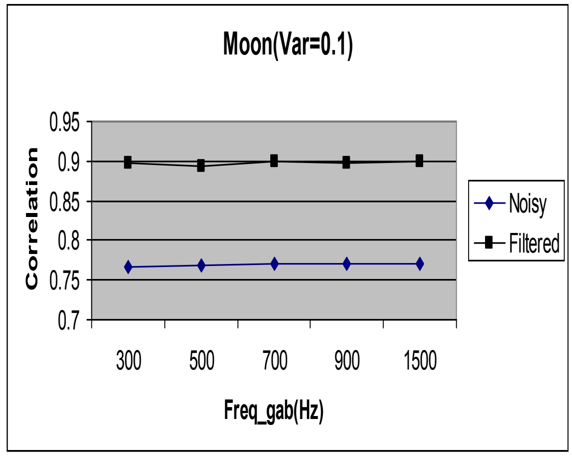

Moon(Var=0.7)
SNR
12
11
10
9
8
7
6
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

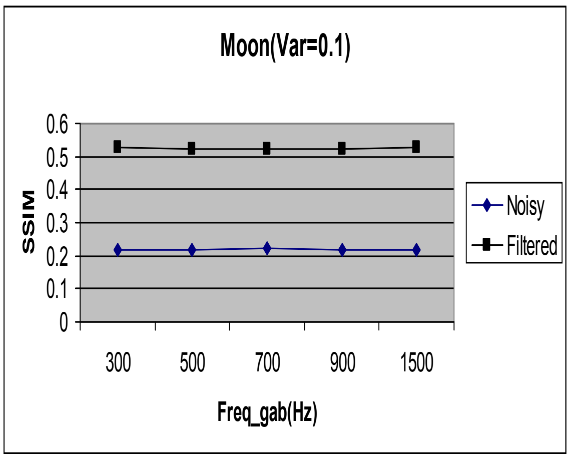

Moon(Var=0.7)
Correlation
0.96
0.94
0.92
0.9
0.88
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

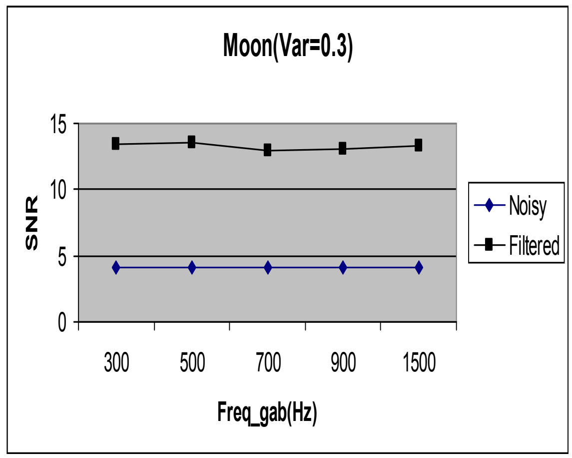

Moon(Var=0.7)
SSIM
0.75
0.7
0.65
0.6
0.55
0.5
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

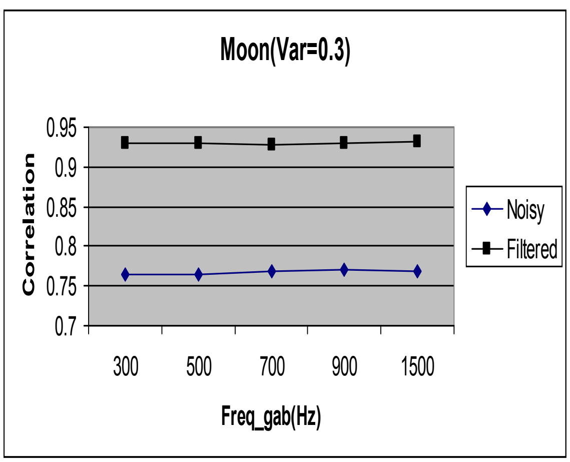
SPECKLE NOISE

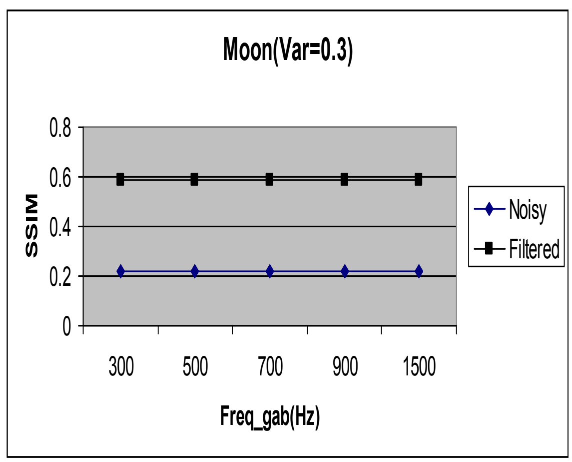


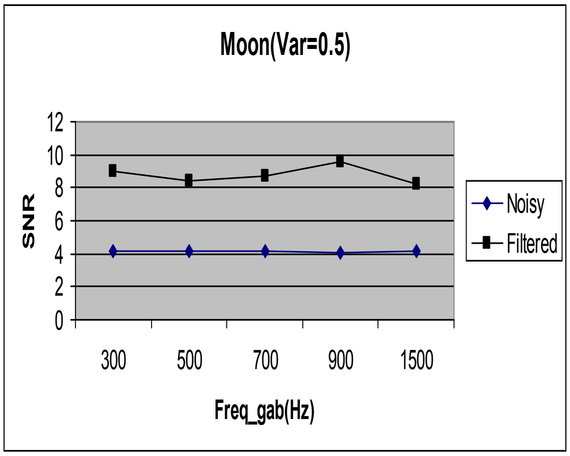


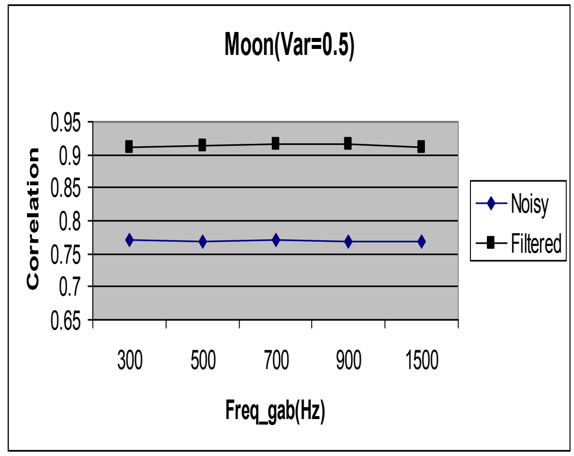


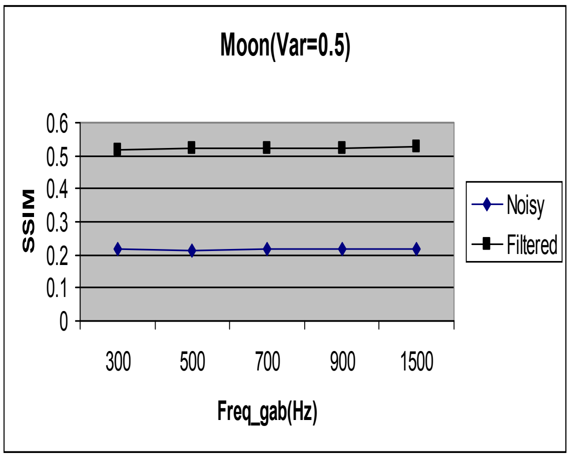


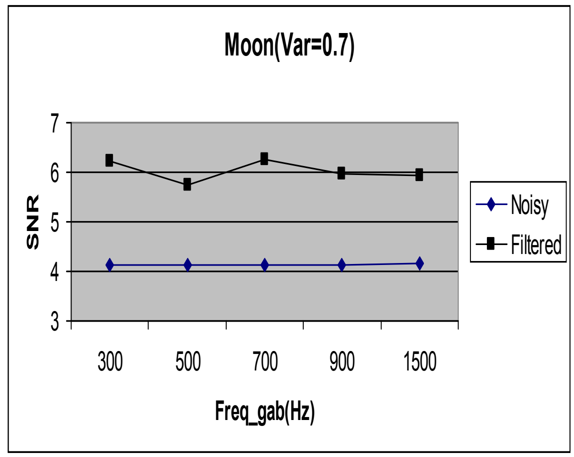


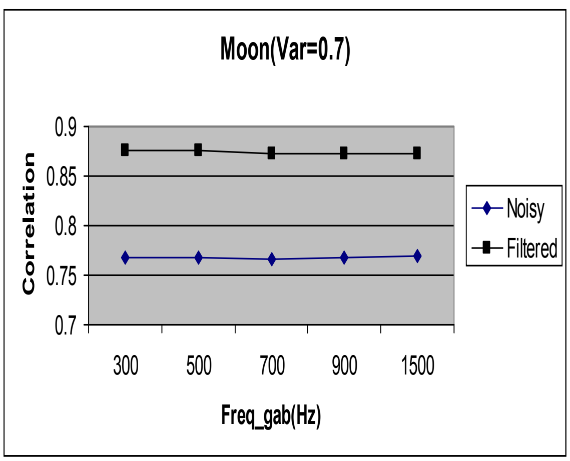


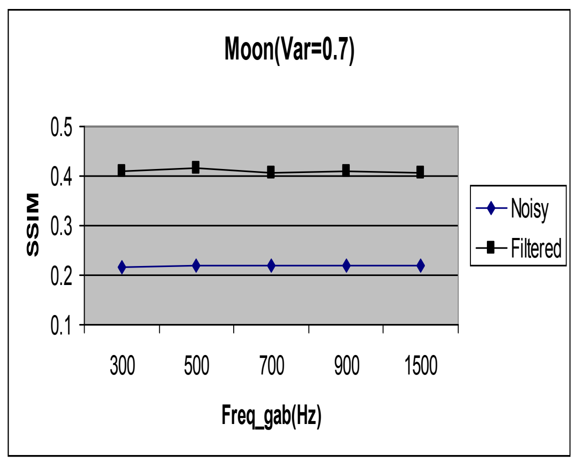

Moon(Var=0.5)
SNR
12
10
8
6
4
2
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Moon(Var=0.5)
Correlation
0.95
0.9
0.85
0.8
0.75
0.7
0.65
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

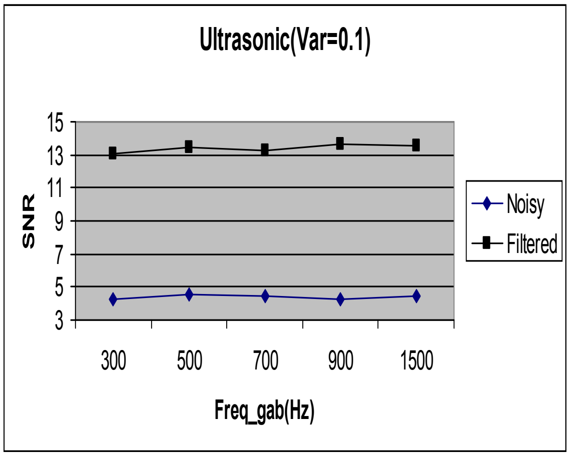

Moon(Var=0.5)
SSIM
0.6
0.5
0.4
0.3
0.2
0.1
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

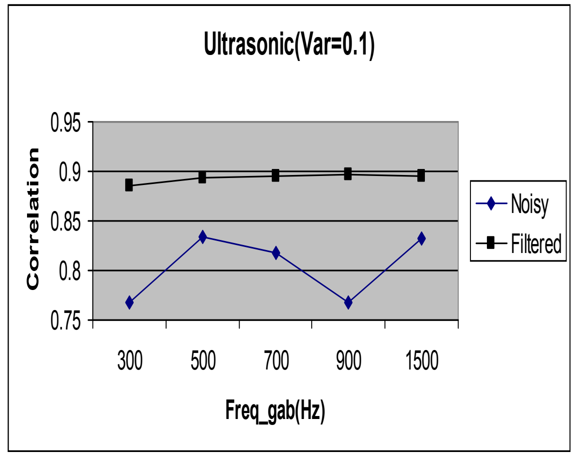

Moon(Var=0.7)
SNR
7
6
5
4
3
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

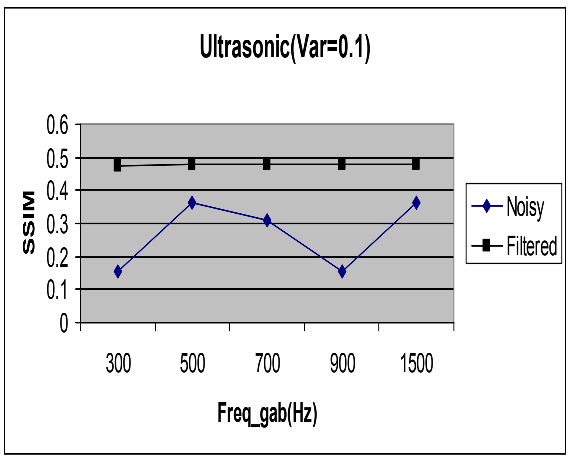

Moon(Var=0.7)
Correlation
0.9
0.85
0.8
0.75
0.7
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

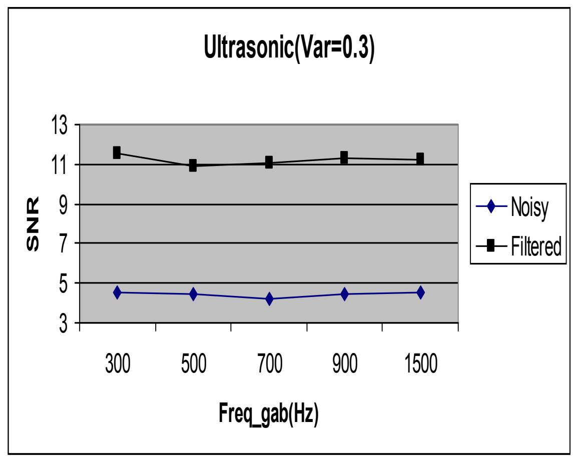

Moon(Var=0.7)
SSIM
0.5
0.4
0.3
0.2
0.1
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

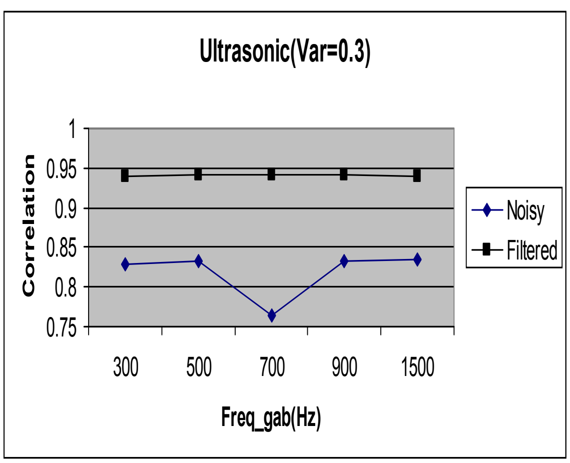
GAUSSIAN NOISE

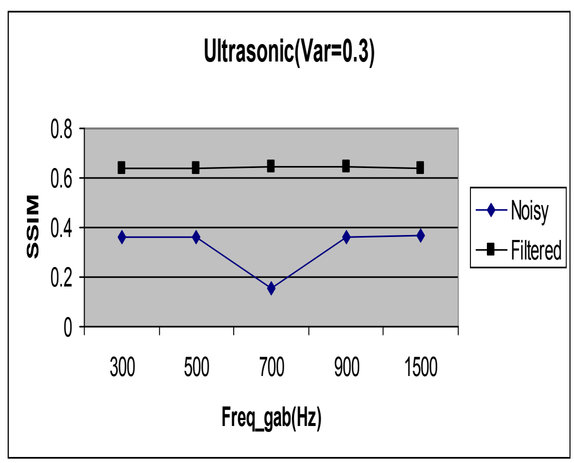


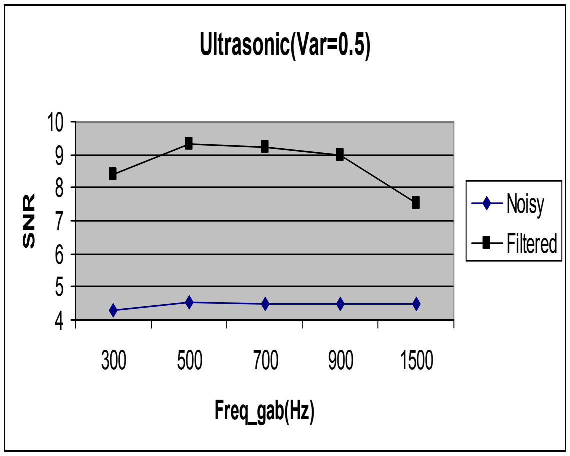


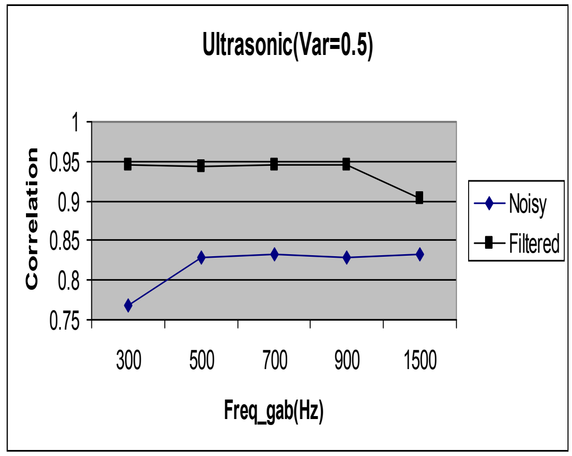


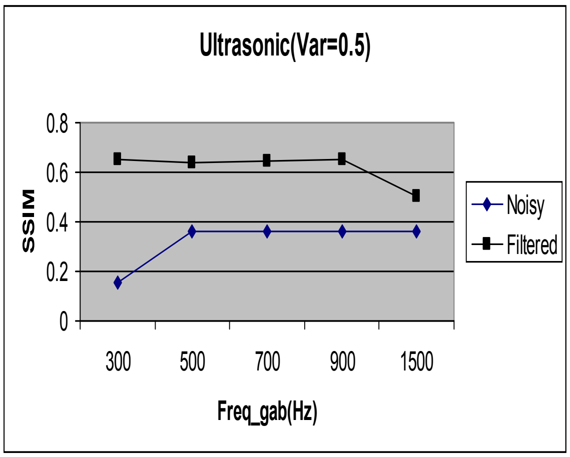


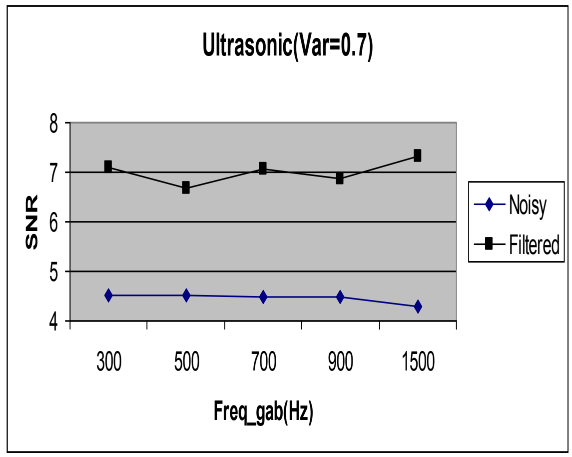


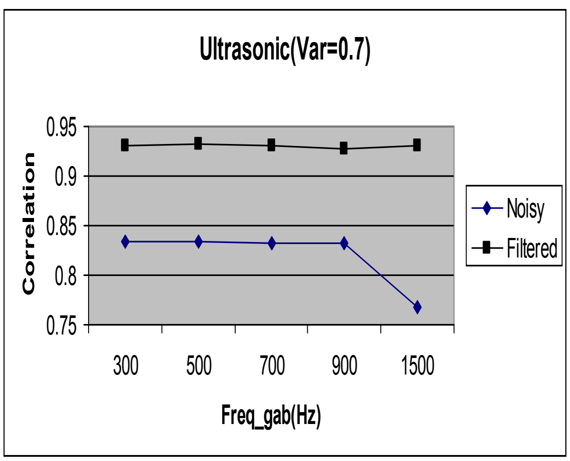


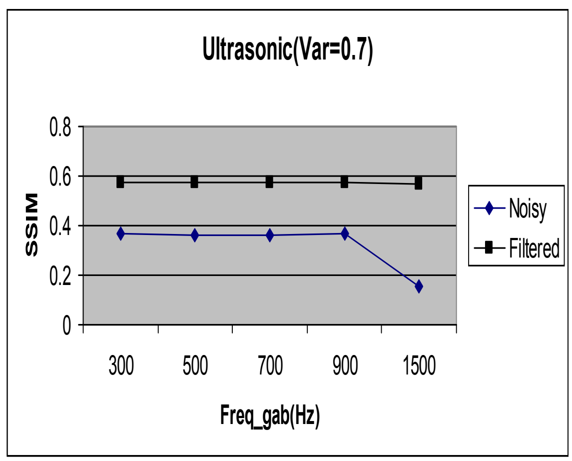

Ultrasonic(Var=0.5)
SNR
10
9
8
7
6
5
4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Ultrasonic(Var=0.5)
Correlation
1
0.95
0.9
0.85
0.8
0.75
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

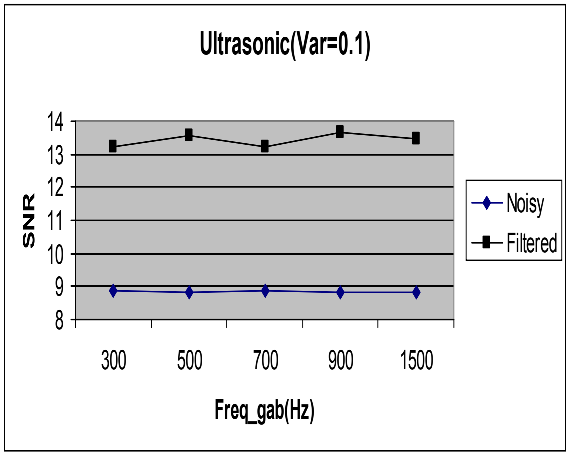

Ultrasonic(Var=0.5)
SSIM
0.8
0.6
0.4
0.2
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

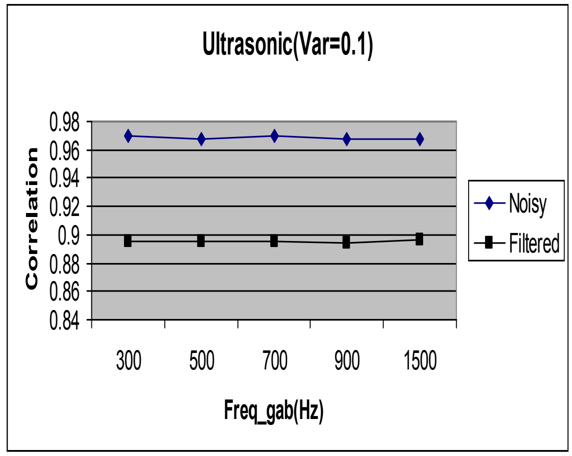

Ultrasonic(Var=0.7)
SNR
8
7
6
5
4
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

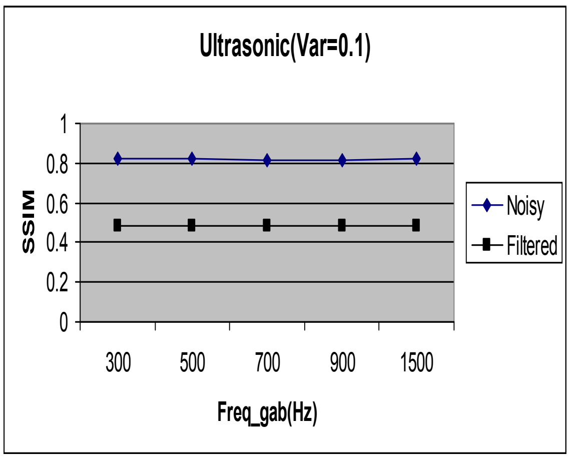

Ultrasonic(Var=0.7)
Correlation
0.95
0.9
0.85
0.8
0.75
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

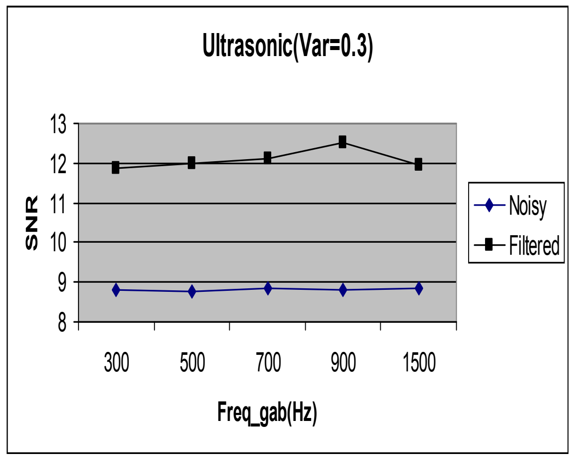

Ultrasonic(Var=0.7)
SSIM
0.8
0.6
0.4
0.2
0
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

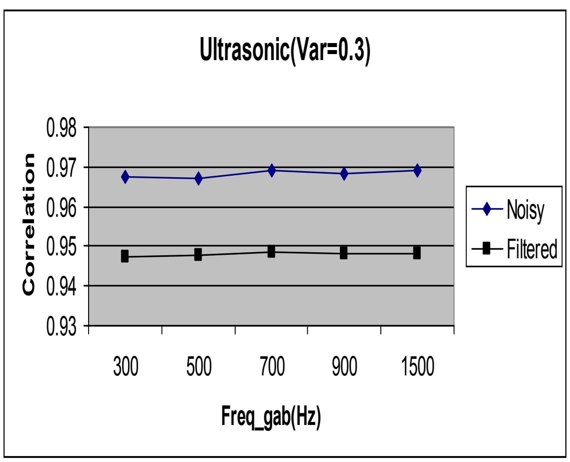
POISSON NOISE

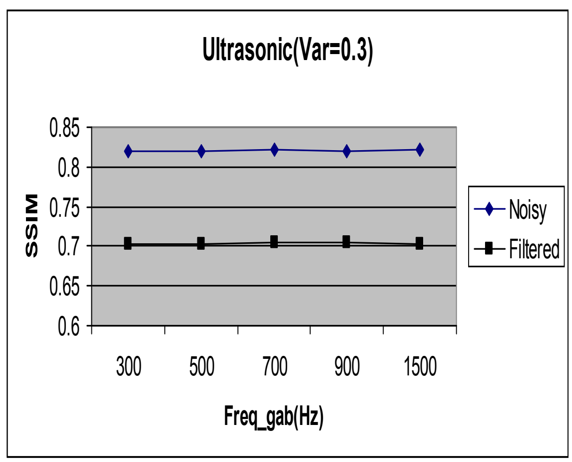


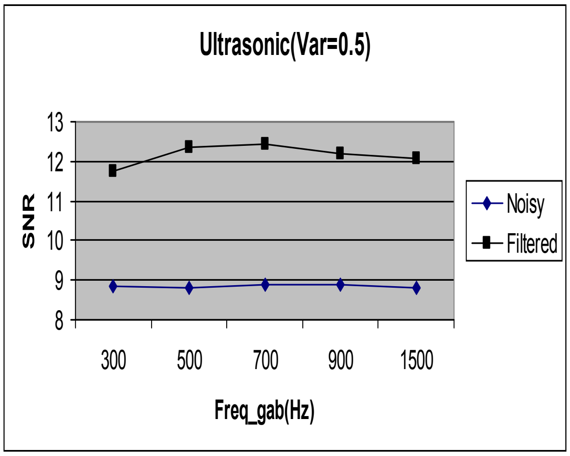


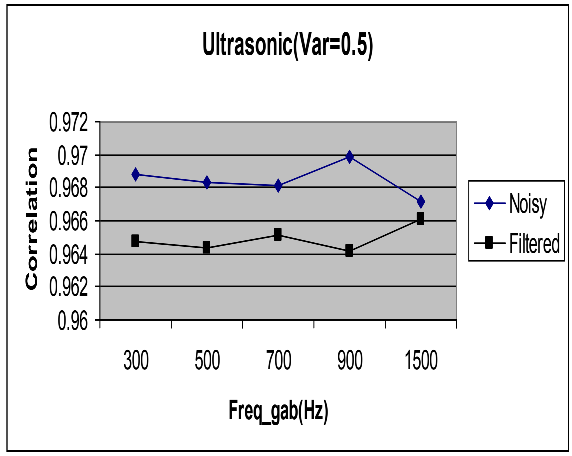


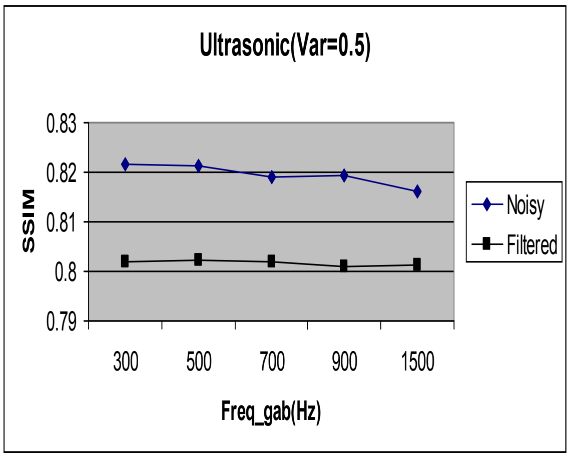


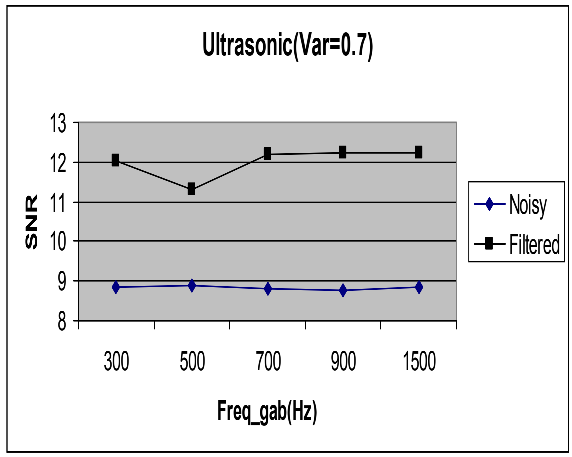


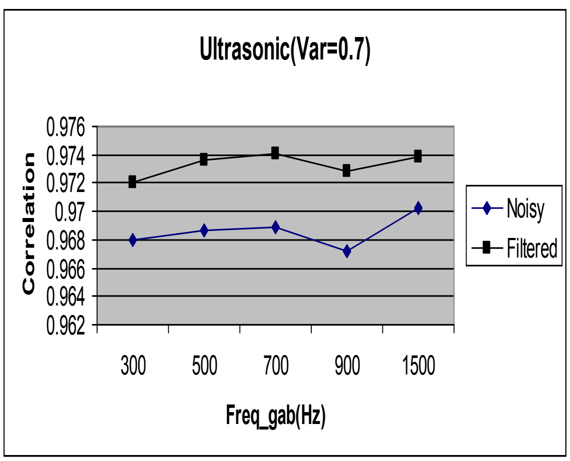


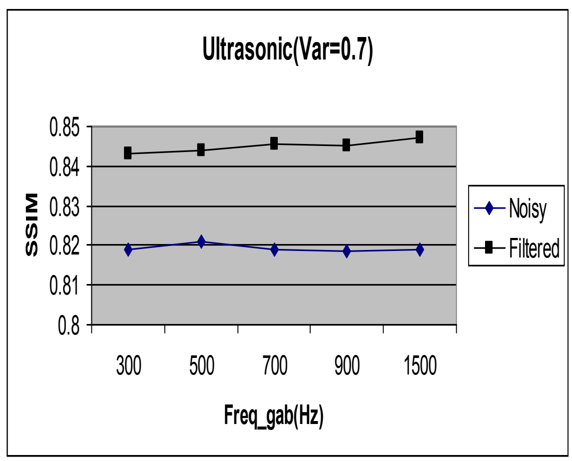

Ultrasonic(Var=0.5)
SNR
13
12
11
10
9
8
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered


Ultrasonic(Var=0.5)
Correlation
0.972
0.97
0.968
0.966
0.964
0.962
0.96
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

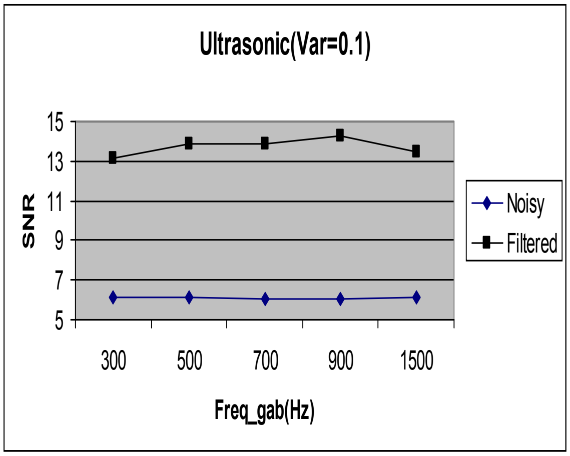

Ultrasonic(Var=0.5)
SSIM
0.83
0.82
0.81
0.8
0.79
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

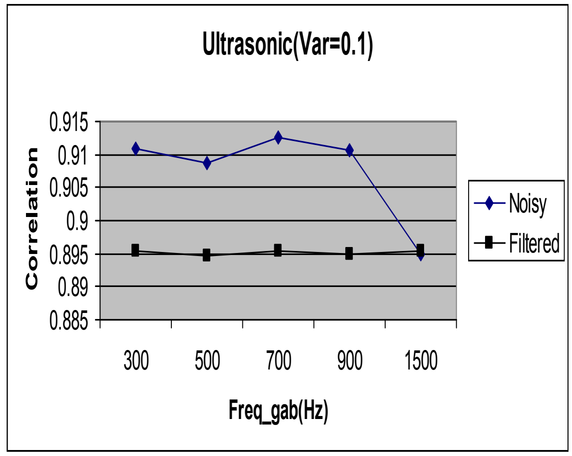

Ultrasonic(Var=0.7)
SNR
13
12
11
10
9
8
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

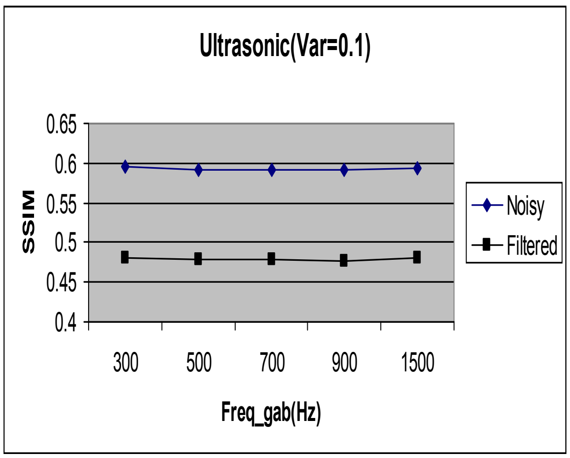

Ultrasonic(Var=0.7)
Correlation
0.976
0.974
0.972
0.97
0.968
0.966
0.964
0.962
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

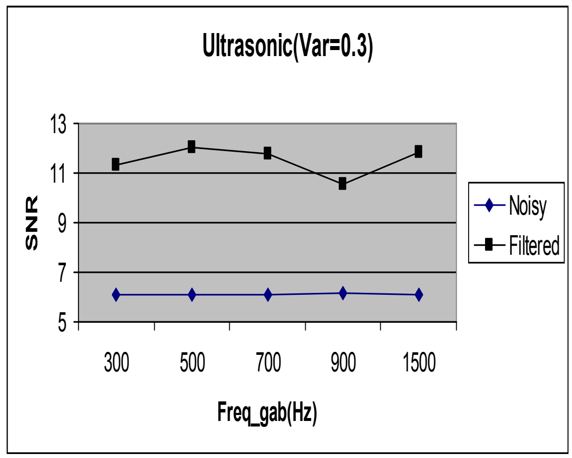

Ultrasonic(Var=0.7)
SSIM
0.85
0.84
0.83
0.82
0.81
0.8
300
500
700
900
1500
Freq_gab(Hz)
Noisy
Filtered

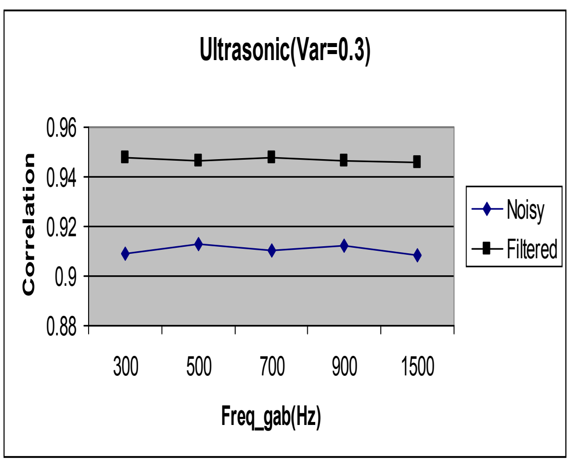
SPECKLE NOISE

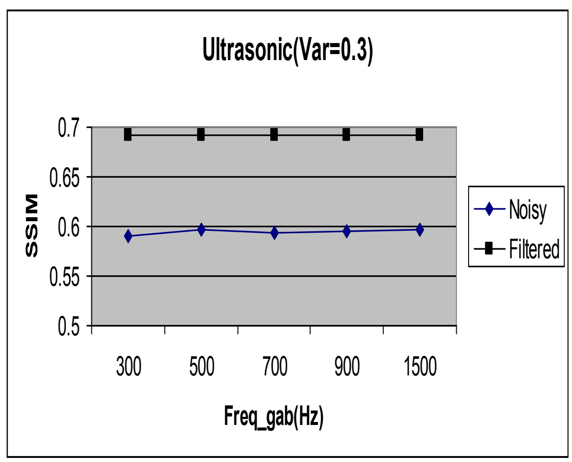


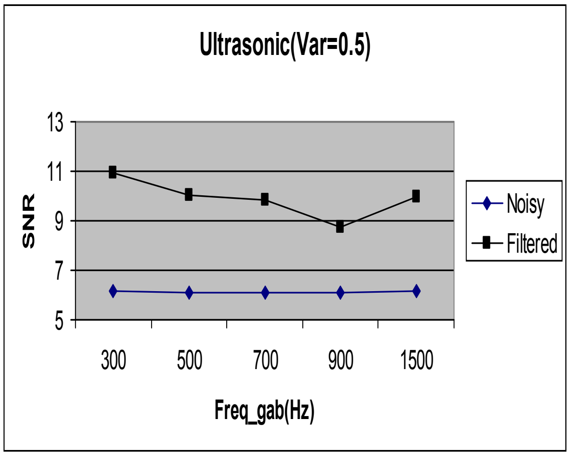


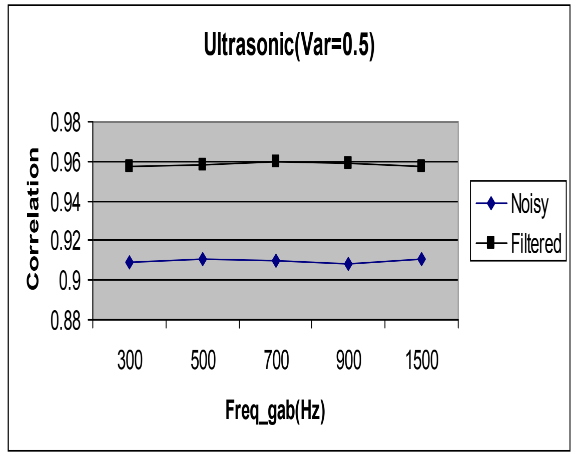


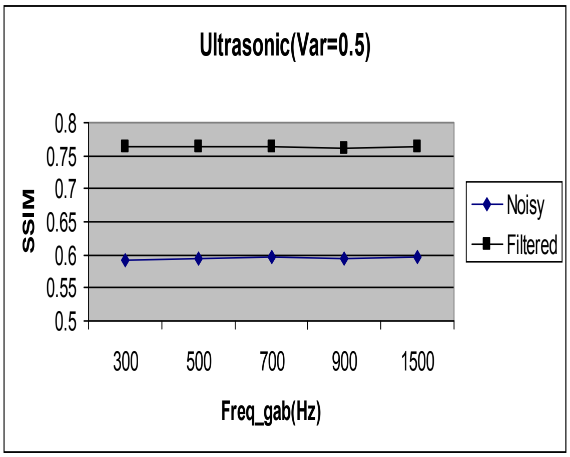


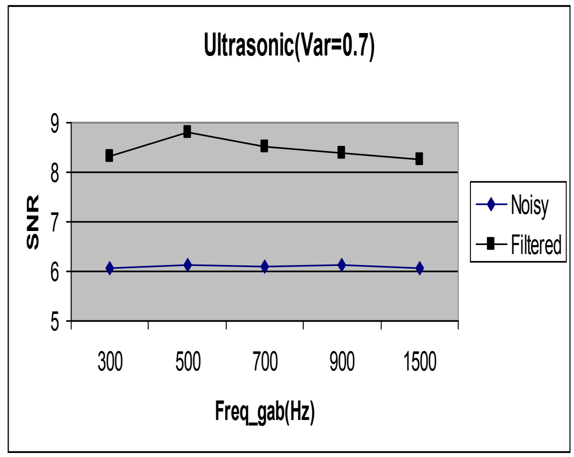


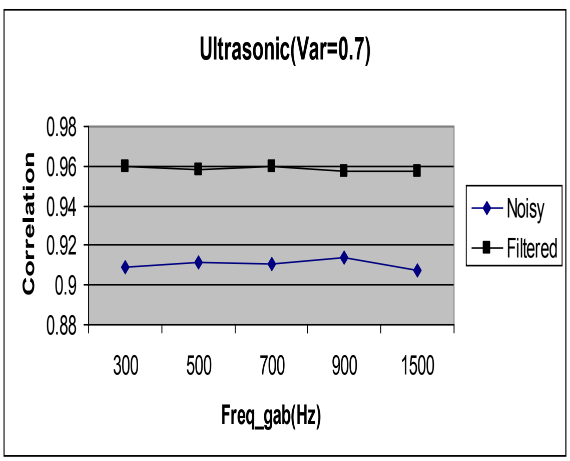


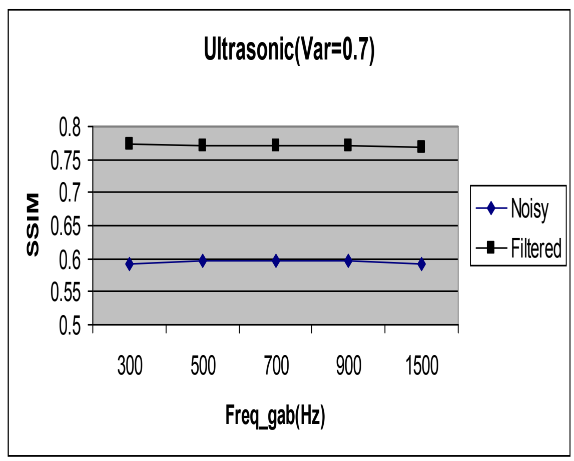










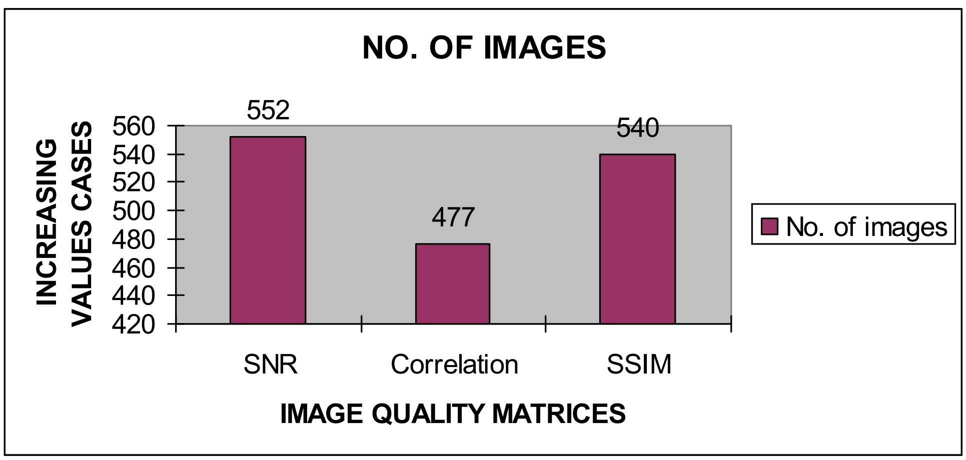


These all above are the graphical results of six images, due to less space we can't display the whole results of all the ten images. Similarly the results of other images can be displayed.

CHAPTER-6
RESULTS SUMMARY

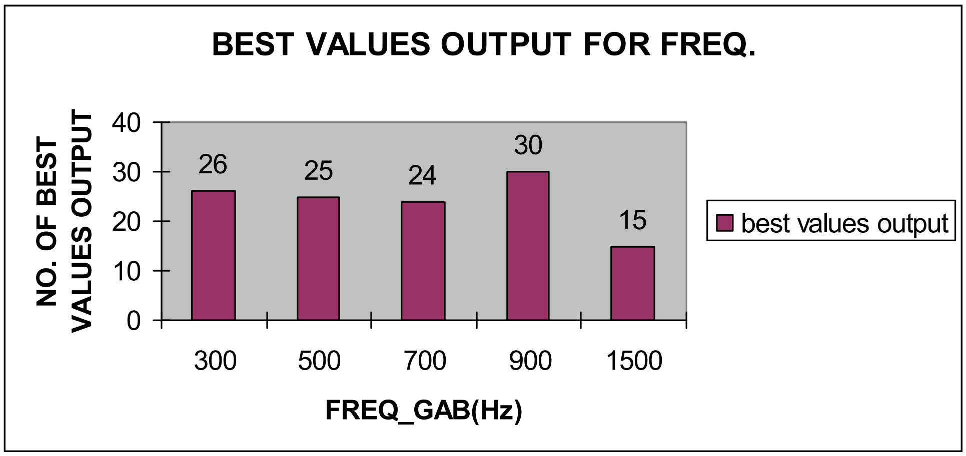
This chapter contains the summary of results obtained by following the Algorithm in Previous Chapters. Here, we have graphically represent the variation regarding the status of working parameters of this Thesis, regarding the best working values of a particular frequency, variation for best values of all the working variances, variation of all best working parameters for every working noise. These are the various plots of all these particular variables:-

- Results for the status of the quality metrics, the number of output of all the parameters that give positive output for filtered image as compared to noisy images.



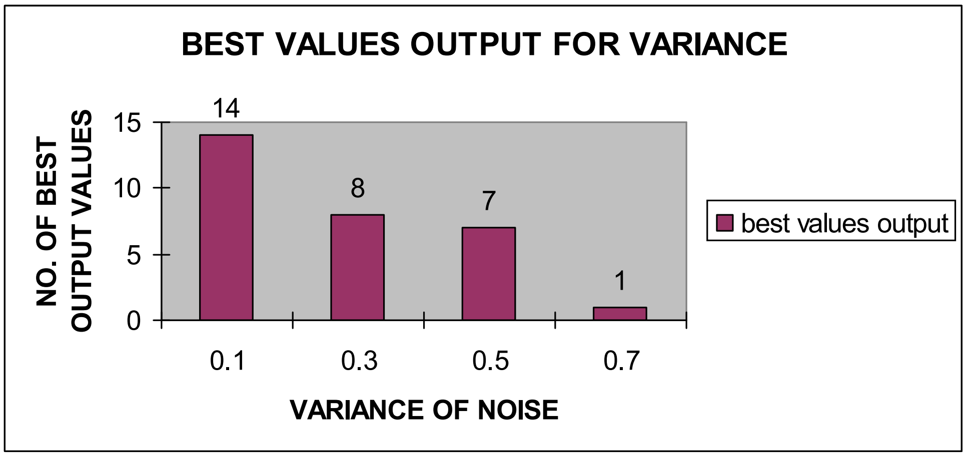
This graph shows the status of the quality metrics, SNR, correlation and SSIM that having positive response as per the comparison between noisy and filtered images. This bar graph represents the no. of cases for positive responses for quality metrics that are having the total no. of 600 cases. For SNR, there are total 552 cases that give better response for Gabor filtered output images as per the comparison between noisy and filtered images. For Correlation, there are total 477 cases that give better response for Gabor filtered output images as per the comparison between noisy and filtered images. For SSIM, there are total 540 cases that give better response for Gabor filtered output images as per the comparison between noisy and filtered images.

- Results for the best working values of particular frequencies, the various frequencies that we are using as input in this work having the graphical form, relating that which particular frequency gives best values for how many observations.



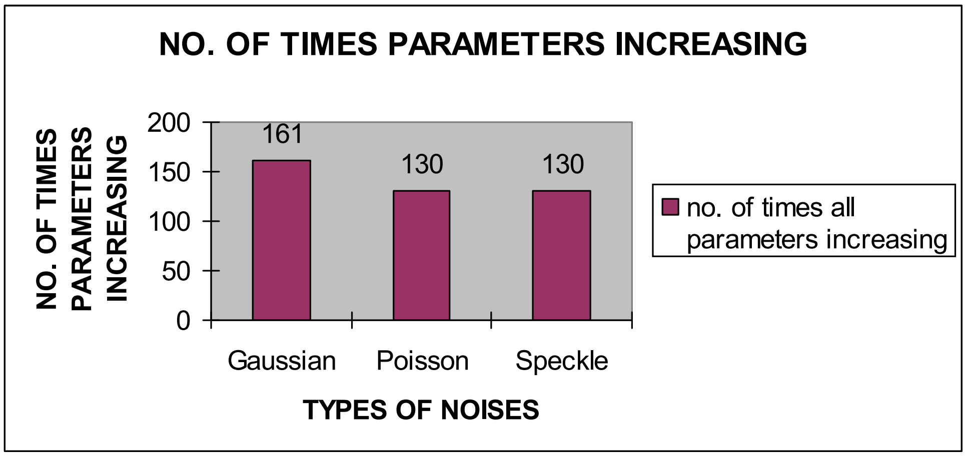
This graph shows the status of the best values output for the frequency of Gabor filter, in Hertz. This bar graph represents the no. of cases for best output value of frequencies; the various frequency values that used are having the total of 120 cases. These 120 cases represent the best value of frequency for a particular 4 variances, a particular variance is testing for various frequencies, that particular frequency is represented here. A particular image is working for 3 different noises and there are total 10 such type of original standard images. That gives the total of 120 such cases. In this representation, 26 cases have the best value output for 300 Hz of Gabor filter, 25 cases have the best value output for 500 Hz of Gabor filter, 24 cases have the best value output for 700 Hz of Gabor filter, 30 cases have the best value output for 900 Hz of Gabor filter, 15 cases have the best value output for 1500 Hz of Gabor filter.

- Results for the best working values of particular variances, the various frequencies that we are using as input in this work having the graphical form, relating that which particular variance gives best values for how many observations.



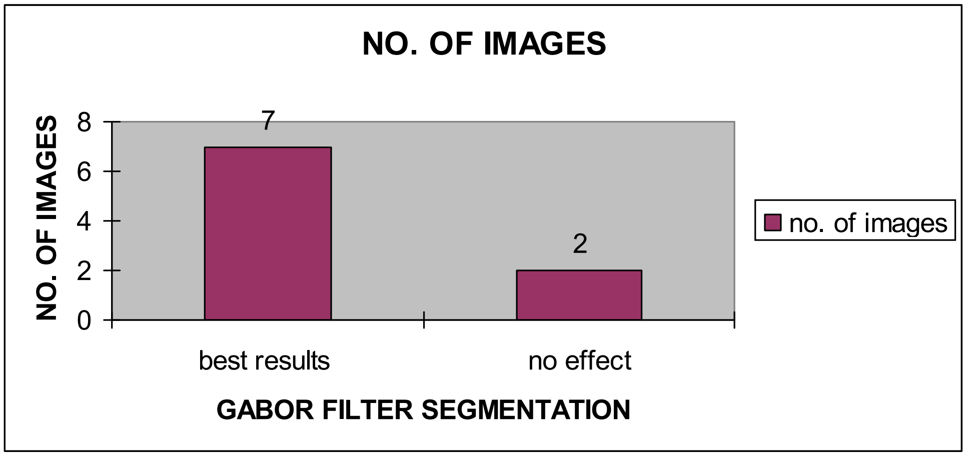
This graph shows the status of the best values output for the variances of Gabor filter. This bar graph represents the no. of cases for best output value of variances; the various variances that used are having the total of 30 cases. These 30 cases represent the best value of variances, a particular variance is working for 3 different noises and there are total 10 such type of original standard images. That gives the total of 30 such cases. In this representation, 14 cases have the best value output for 0.1, 8 cases have the best value output for 0.3, 7 cases have the best value output for 0.5, 1 case have the best value output for 0.7.

- Results for all the three noises, this graphical form display the number of times the quality metrics SNR, Correlation and SSIM increases as per the analysis of noisy and filtered images.



This graph shows the status of the quality metrics, SNR, correlation and SSIM that having positive response as per the comparison between noisy and filtered images. This bar graph represents the no. of cases for positive responses for all quality metrics that are having the total no. of 200 cases for 3 different noises. For Gaussian noise, there are total of 161 cases that give better response for all three quality metrics, as per the comparison between noisy and filtered images. For Poisson noise, there are total of 130 cases that give better response for all three quality metrics, as per the comparison between noisy and filtered images. For Speckle noise, there are total of 130 cases that give better response for all three quality metrics, as per the comparison between noisy and filtered images.

- Results that summarized the image segmentation, it gives information about the status of the images that gives best results after applying Gabor filter.



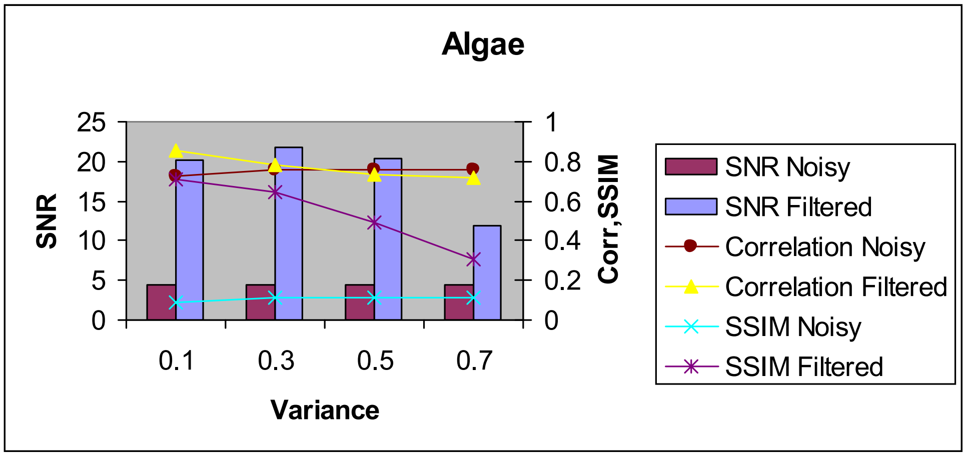
This graph shows the status of the result of the image after image segmentation. This bar graph represents the no. of best results for the image obtained after image segmentation and the no. of images that give no effect after applying Gabor filter to them. We are using total 10 standard input images. Of these 10, 1 is not giving any particular segmentation result after applying segmentation to noisy and Gabor filtered output image, that's the reason we don't include it here. Of rest 9, 7 images give best result for Gabor filtered output image after applying segmentation to them and 2 images give no effect for Gabor filtered output image after applying segmentation to them, they don't give the best result for image segmentation as per the comparison between the Gabor filtered output and the noisy images.

CHAPTER-7
DISCUSSIONS OF RESULTS

This chapter contains the results obtained by following the Algorithm in Previous Chapter. Here, we have graphically represent the variation of various parameters for noisy and filtered images. This is a way of representing the best output values of every parameter with the change of the value of the variance. These are the various plots of all the parameter for the Variance.

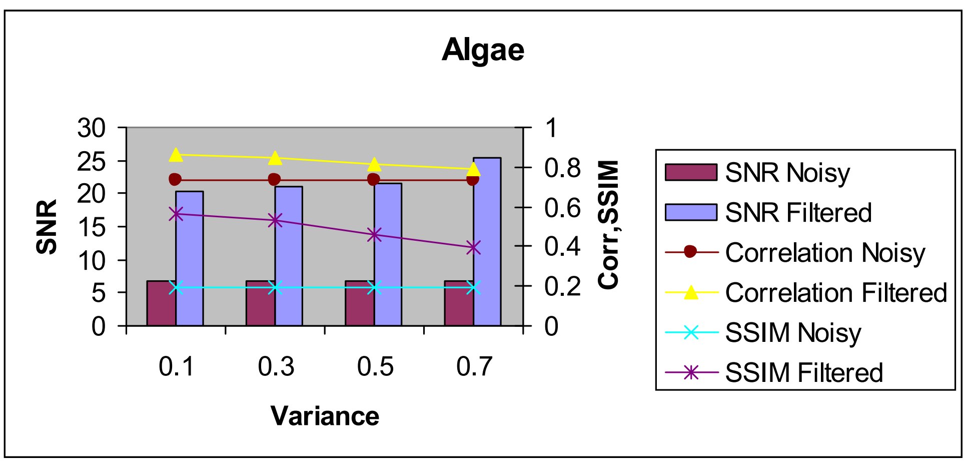
- Result For 'ALGAE.tif' using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.4891	20.2421	0.7221	0.8561	0.0861	0.7123
0.3	4.4723	21.8606	0.7601	0.7824	0.1152	0.6417
0.5	4.4736	20.4156	0.7612	0.7319	0.1156	0.4935
0.7	4.4741	11.8081	0.7591	0.7212	0.1151	0.3059



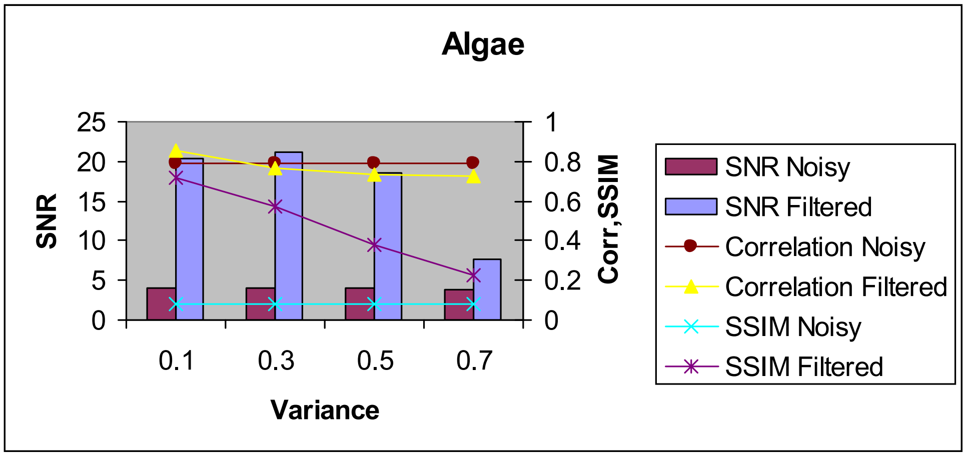
- Result For 'ALGAE.tif' using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	6.7988	20.2422	0.7307	0.8664	0.1949	0.5664
0.3	6.7836	21.0807	0.7332	0.849	0.1947	0.529
0.5	6.7908	21.4214	0.7301	0.8158	0.1947	0.4624
0.7	6.7942	25.2898	0.7339	0.789	0.195	0.3981



- Result For 'ALGAE.tif' using Speckle Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	3.9327	20.4442	0.7931	0.8555	0.0843	0.7204
0.3	3.9467	21.146	0.7928	0.7629	0.0834	0.5749
0.5	3.9524	18.617	0.7942	0.7325	0.0836	0.3811
0.7	3.9255	7.7239	0.7917	0.7239	0.0838	0.2227



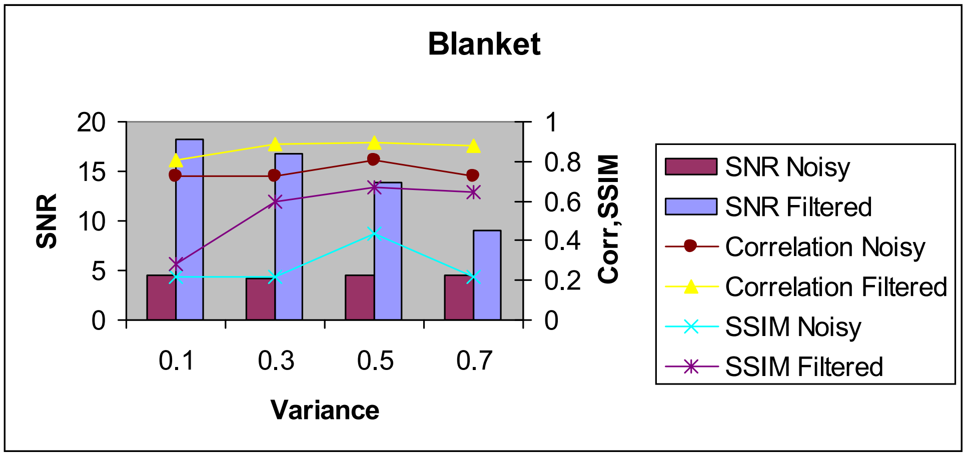
For Gaussian noise, SNR gives best result for the variance of 0.3, correlation coefficient gives best result for the variance of 0.1. Similarly, SSIM gives best result for the variance of 0.1.

For Poisson noise, SNR gives best result for the variance of 0.5, correlation coefficient gives best result for the variance of 0.1. Similarly, SSIM gives best result for the variance of 0.1.

For Speckle noise, SNR gives best result for the variance of 0.3, correlation coefficient gives best result for the variance of 0.1. Similarly, SSIM gives best result for the variance of 0.1.

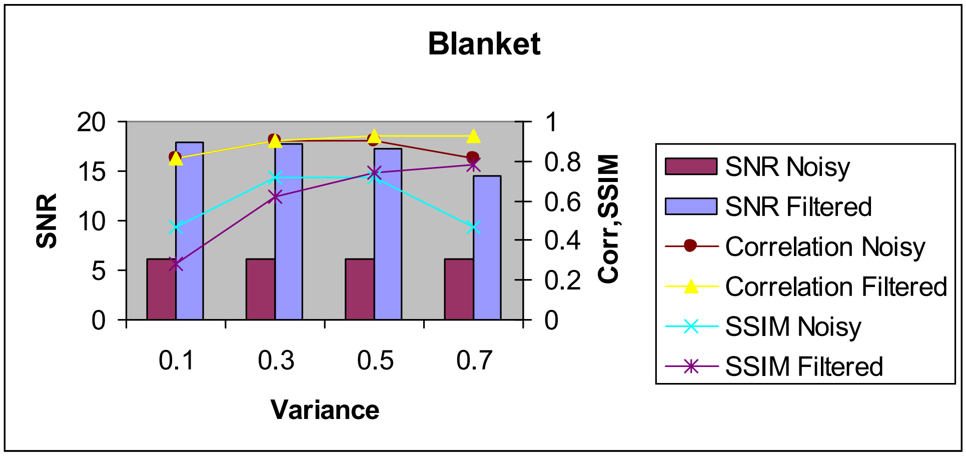
- Result For 'BLANKET.tif' using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.4635	18.2421	0.725	0.8099	0.2214	0.2783
0.3	4.2328	16.802	0.7257	0.8888	0.2213	0.5956
0.5	4.4713	13.9472	0.8052	0.8951	0.4341	0.6675
0.7	4.4743	8.9668	0.7244	0.8787	0.2217	0.6431



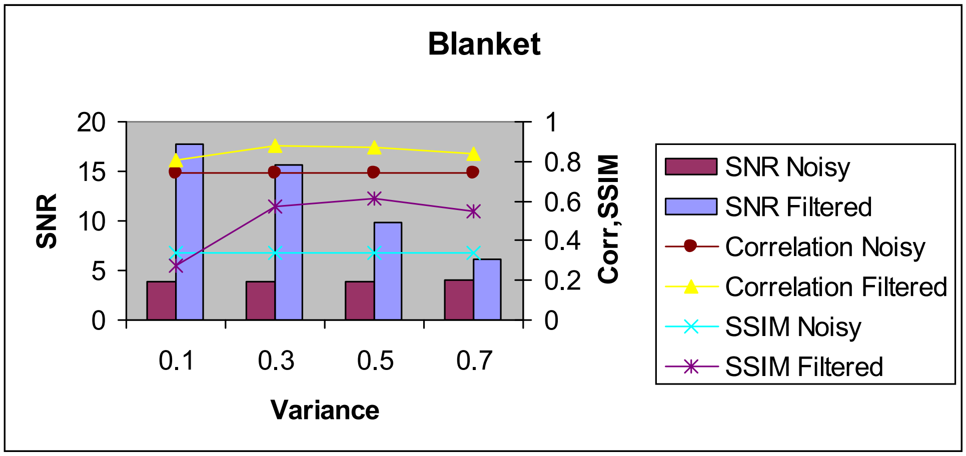
- Result For 'BLANKET.tif' using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	6.1088	17.8621	0.8146	0.8105	0.4648	0.2791
0.3	6.1044	17.7785	0.9017	0.9038	0.7172	0.621
0.5	6.0842	17.1896	0.9001	0.9235	0.7146	0.74
0.7	6.1047	14.4666	0.8157	0.9306	0.4645	0.7805



- Result For ‘BLANKET.tif’ using Speckle Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	3.9304	17.6824	0.7412	0.8052	0.3395	0.2773
0.3	3.9283	15.5848	0.7434	0.8778	0.3379	0.574
0.5	3.9307	9.8901	0.7438	0.8729	0.337	0.6144
0.7	3.9599	6.1881	0.7421	0.8387	0.3376	0.5494



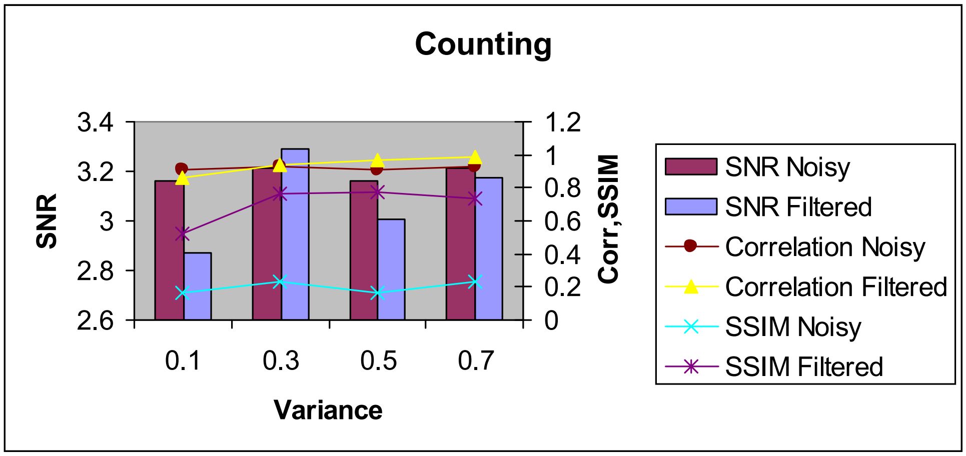
For Gaussian noise, SNR gives best result for the variance of 0.1, correlation coefficient gives best result for the variance of 0.3, SSIM gives best result for the variance of 0.7.

For Poisson noise, SNR gives best result for the variance of 0.1, correlation coefficient gives best result for the variance of 0.7. Similarly, SSIM gives best result for the variance of 0.7.

For Speckle noise, SNR gives best result for the variance of 0.1, correlation coefficient gives best result for the variance of 0.3. SSIM gives best result for the variance of 0.5.

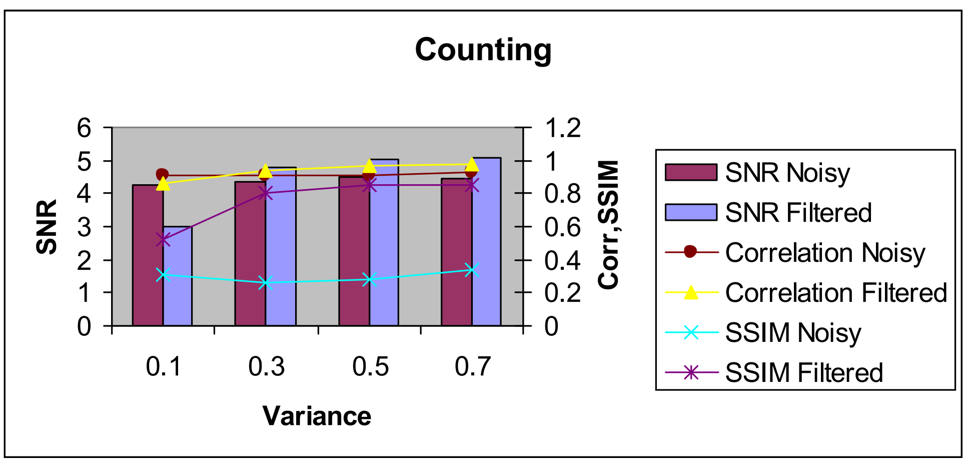
- Result For ‘COUNTING.tiff’ using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	3.1642	2.8713	0.9129	0.862	0.1692	0.5251
0.3	3.215	3.2899	0.9277	0.9411	0.2317	0.7676
0.5	3.1621	3.0057	0.9123	0.9657	0.1664	0.7706
0.7	3.213	3.1724	0.9294	0.9848	0.2327	0.7347



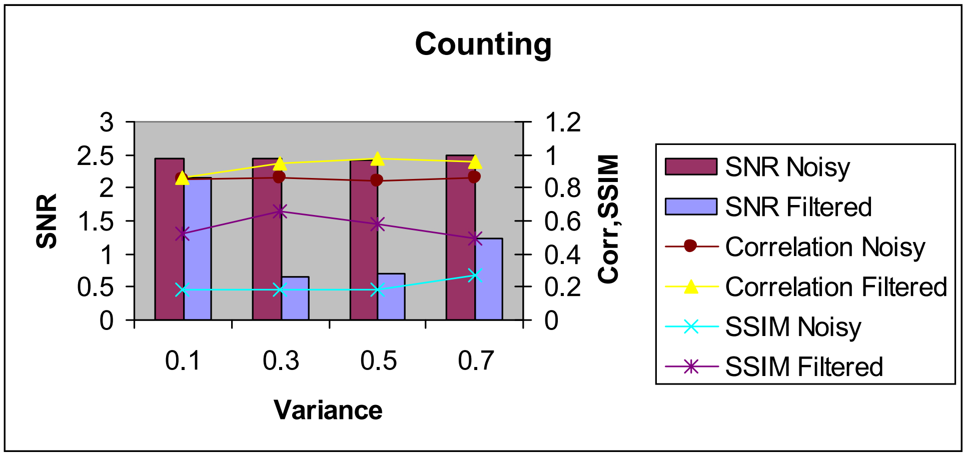
- Result For ‘COUNTING.tiff’ using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.2522	3.0047	0.9087	0.8596	0.3092	0.5271
0.3	4.3571	4.7727	0.9061	0.9416	0.2627	0.8011
0.5	4.4841	5.0335	0.9084	0.9636	0.2849	0.8502
0.7	4.4595	5.0798	0.9306	0.9779	0.3348	0.854



- Result For ‘COUNTING.tiff’ using Speckle Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	2.4544	2.1526	0.8473	0.8617	0.1856	0.5234
0.3	2.4502	0.6544	0.8622	0.9499	0.185	0.6606
0.5	2.4234	0.7037	0.8458	0.9786	0.1856	0.5784
0.7	2.4904	1.2399	0.8613	0.9606	0.2756	0.4919

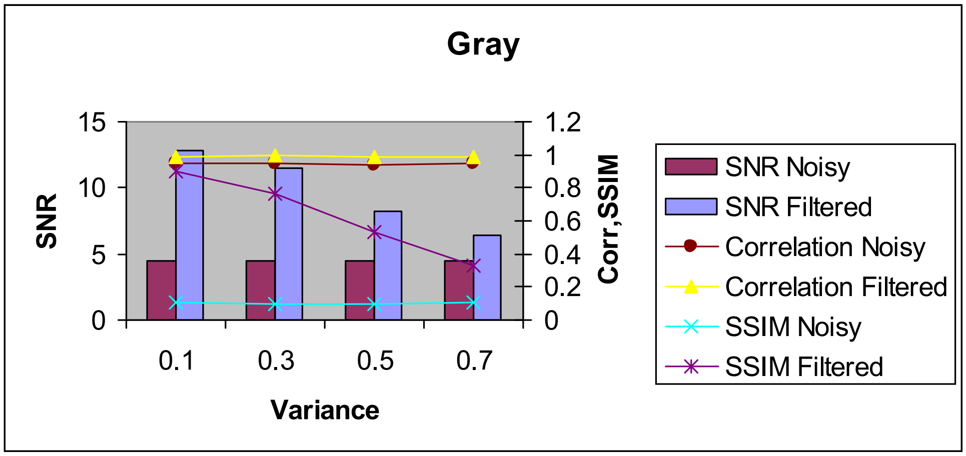


For Gaussian noise, SNR gives best result for the variance of 0.3, correlation coefficient gives best result for the variance of 0.7 and SSIM gives best result for the variance of 0.5. For Poisson noise, SNR gives best result for the variance of 0.7, correlation coefficient gives best result for the variance of 0.5. Similarly, SSIM gives best result for the variance of 0.5.

For Speckle noise, SNR gives best result for the variance of 0.1, correlation coefficient gives best result for the variance of 0.5 and SSIM gives best result for the variance of 0.3.

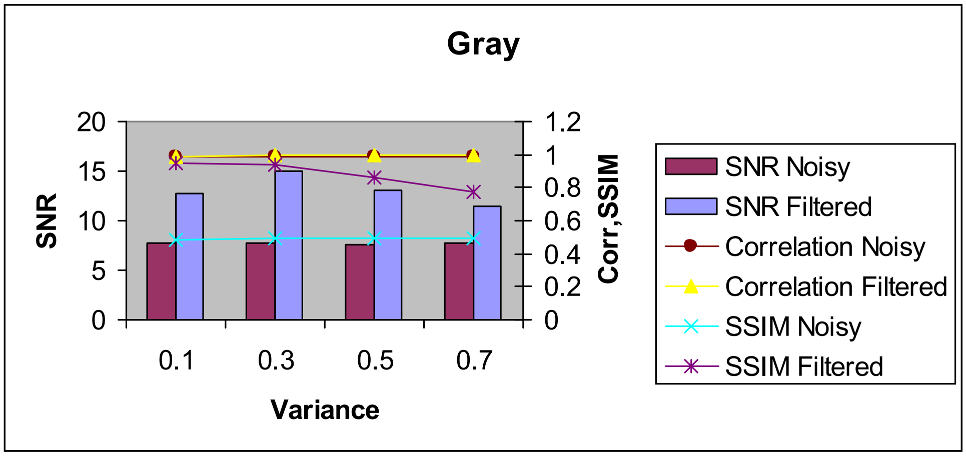
- Result For ‘GRAY.tiff’ using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.4386	12.8582	0.9444	0.9905	0.108	0.8975
0.3	4.4937	11.5436	0.9438	0.9945	0.1	0.7627
0.5	4.4622	8.2265	0.9415	0.9902	0.1006	0.5357
0.7	4.4657	6.3586	0.9436	0.9833	0.108	0.3323



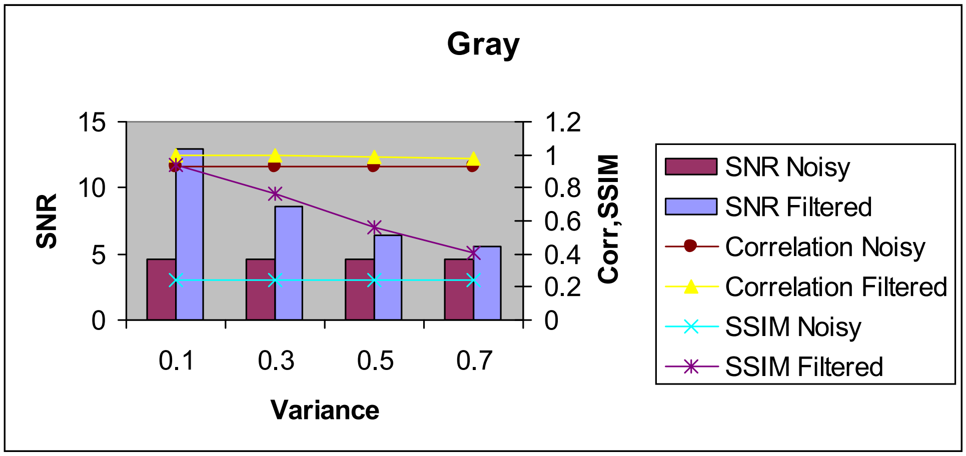
- Result For ‘GRAY.tiff’ using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	7.6926	12.7655	0.9871	0.9897	0.4836	0.9455
0.3	7.736	14.9486	0.9888	0.9957	0.4943	0.9368
0.5	7.638	12.9867	0.9885	0.9978	0.4915	0.8567
0.7	7.7047	11.5307	0.9887	0.9984	0.4941	0.7714



- Result For 'GRAY.tiff' using Speckle Noise.

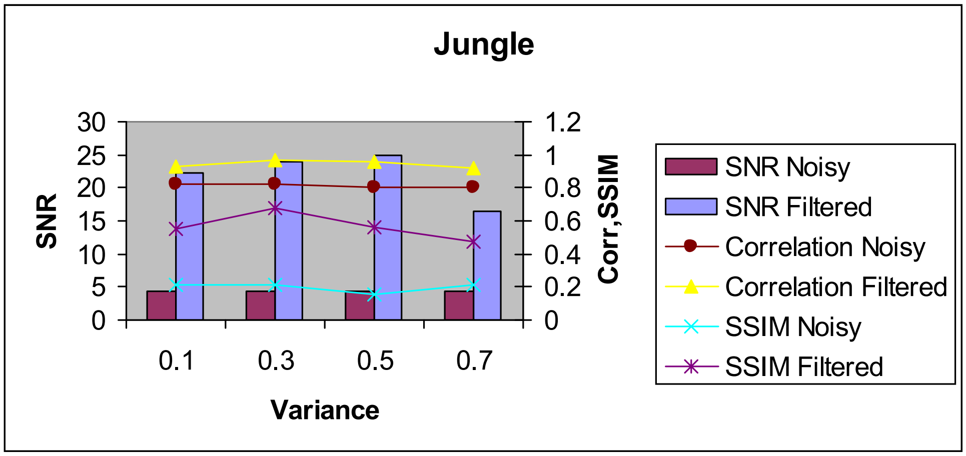
Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.5719	12.9515	0.9279	0.9964	0.2387	0.9342
0.3	4.5815	8.5789	0.9276	0.9941	0.2397	0.7691
0.5	4.5829	6.4384	0.9272	0.989	0.2387	0.563
0.7	4.5821	5.5724	0.9283	0.9781	0.2401	0.4088



For Gaussian noise, SNR gives best result for the variance of 0.1, correlation coefficient gives best result for the variance of 0.3 and SSIM gives best result for the variance of 0.1. For Poisson noise, SNR gives best result for the variance of 0.3, correlation coefficient gives best result for the variance of 0.7 and SSIM gives best result for the variance of 0.1. For Speckle noise, SNR gives best result for the variance of 0.1. Similarly, correlation coefficient gives best result for the variance of 0.1 and SSIM gives best result for the variance of 0.1.

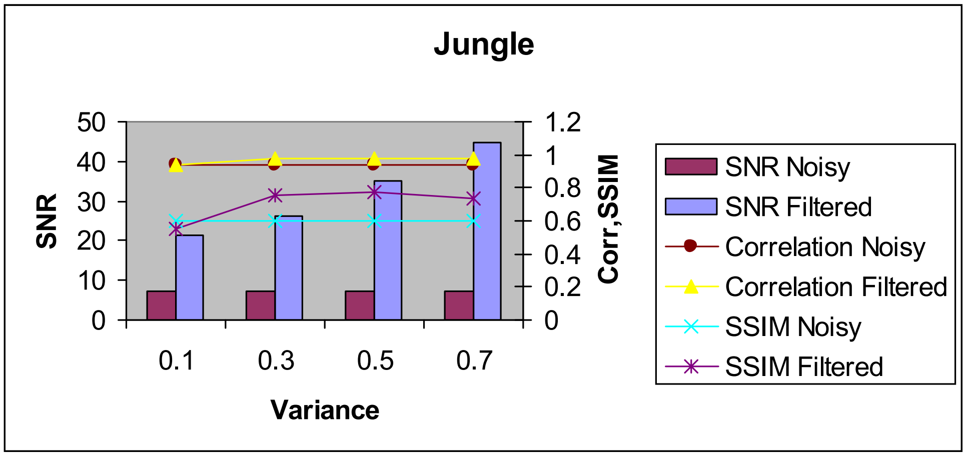
- Result For ‘JUNGLE.tiff’ using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.4603	22.3297	0.82	0.9331	0.213	0.5525
0.3	4.4644	24.0058	0.8212	0.9668	0.2158	0.6771
0.5	4.4139	24.9142	0.8002	0.9546	0.1594	0.565
0.7	4.4595	16.5276	0.7999	0.9205	0.2153	0.4733



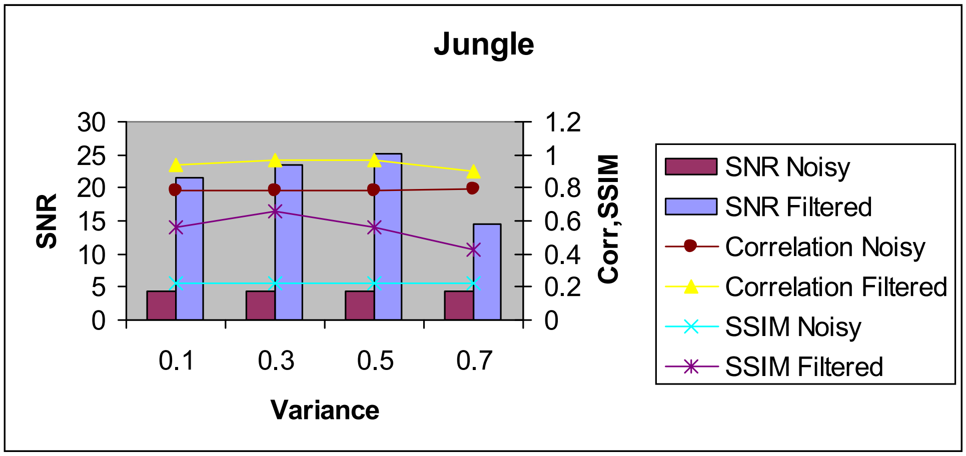
- Result For ‘JUNGLE.tiff’ using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	7.3638	21.5414	0.9405	0.9351	0.5971	0.556
0.3	7.3638	26.3869	0.9413	0.9782	0.5981	0.7555
0.5	7.3935	34.9371	0.9408	0.9812	0.5978	0.7781
0.7	7.354	44.789	0.9415	0.9799	0.6005	0.7326



- Result For ‘JUNGLE.tiff’ using Speckle Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.2515	21.6187	0.7869	0.934	0.2196	0.5576
0.3	4.2725	23.4558	0.7882	0.9632	0.2209	0.6609
0.5	4.2637	25.0775	0.7874	0.964	0.2198	0.5658
0.7	4.2653	14.5167	0.7896	0.9024	0.2215	0.4221



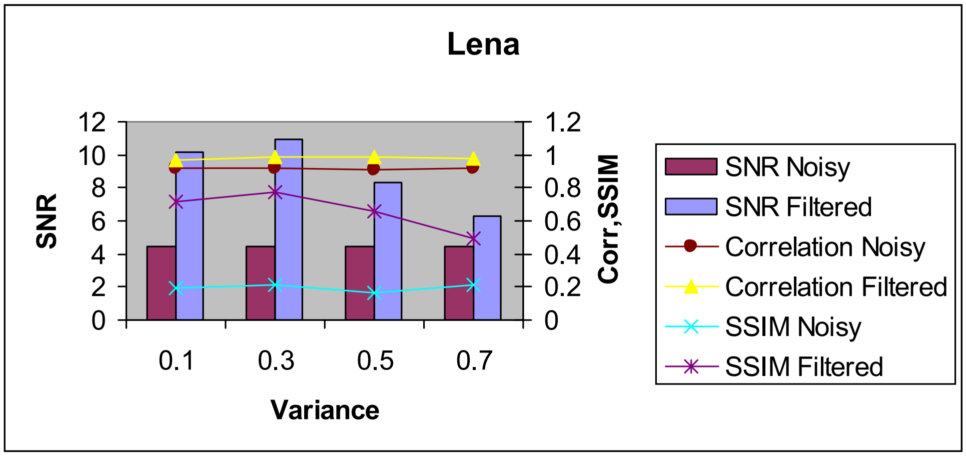
For Gaussian noise, SNR gives best result for the variance of 0.5. Similarly, correlation coefficient gives best result for the variance of 0.5 and SSIM gives best result for the variance of 0.3.

For Poisson noise, SNR gives best result for the variance of 0.7, correlation coefficient gives best result for the variance of 0.5 .Similarly, SSIM gives best result for the variance of 0.5.

For Speckle noise, SNR gives best result for the variance of 0.5. Similarly, correlation coefficient gives best result for the variance of 0.5 and SSIM gives best result for the variance of 0.3.

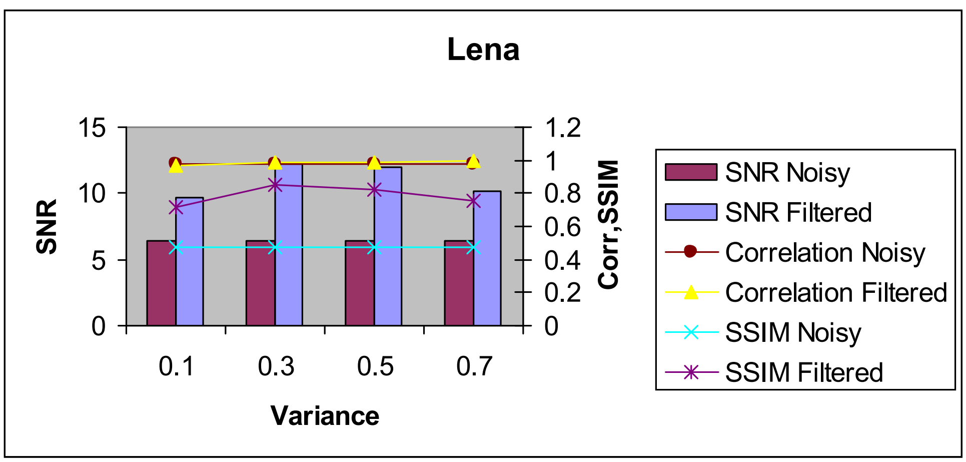
- Result For 'LENA.jpg' using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.4761	10.1857	0.9149	0.9631	0.1915	0.7124
0.3	4.417	10.943	0.918	0.9875	0.2093	0.7777
0.5	4.4701	8.3609	0.9105	0.9855	0.1692	0.6607
0.7	4.467	6.2887	0.9181	0.977	0.2093	0.496



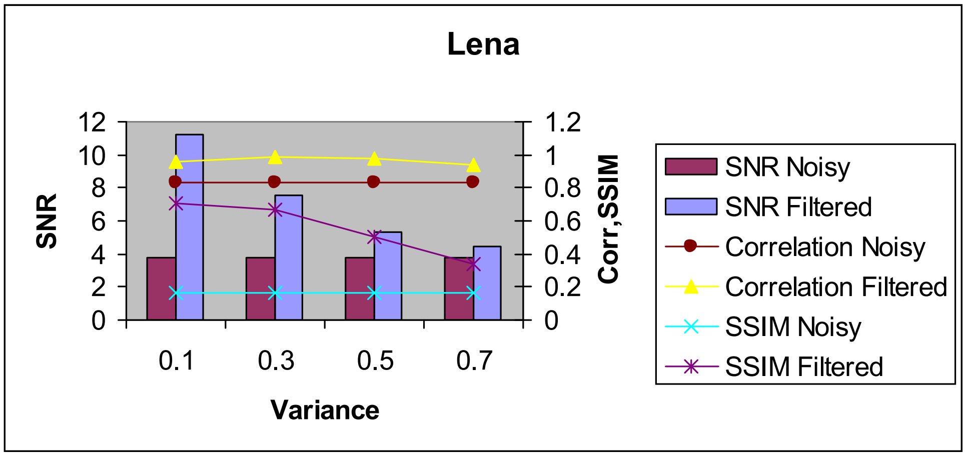
- Result For 'LENA.jpg' using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	6.4321	9.7347	0.9741	0.9635	0.4736	0.717
0.3	6.4148	12.1877	0.9739	0.9897	0.475	0.849
0.5	6.4433	11.9433	0.9734	0.9917	0.4735	0.8237
0.7	6.4161	10.1925	0.9742	0.9935	0.4741	0.758



- Result For ‘LENA.jpg’ using Speckle Noise.

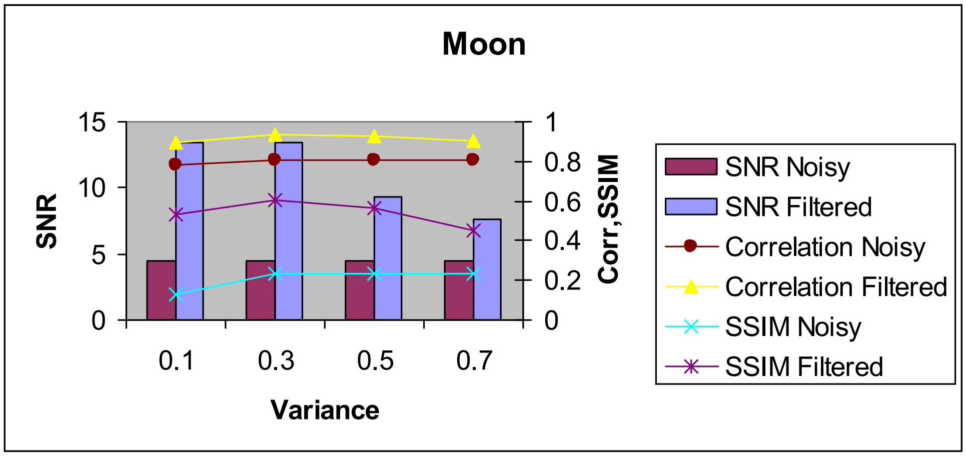
Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	3.7854	11.2583	0.8339	0.9616	0.1646	0.7057
0.3	3.7821	7.5134	0.8336	0.9827	0.1656	0.6684
0.5	3.7835	5.3304	0.8335	0.9733	0.1664	0.4987
0.7	3.8006	4.4302	0.8338	0.942	0.1657	0.3407



For Gaussian noise, SNR gives best result for the variance of 0.3, correlation coefficient gives best result for the variance of 0.5 and SSIM gives best result for the variance of 0.3. For Poisson noise, SNR gives best result for the variance of 0.3, correlation coefficient gives best result for the variance of 0.7 and SSIM gives best result for the variance of 0.3. For Speckle noise, SNR gives best result for the variance of 0.1, Correlation coefficient gives best result for the variance of 0.3 and SSIM gives best result for the variance of 0.1.

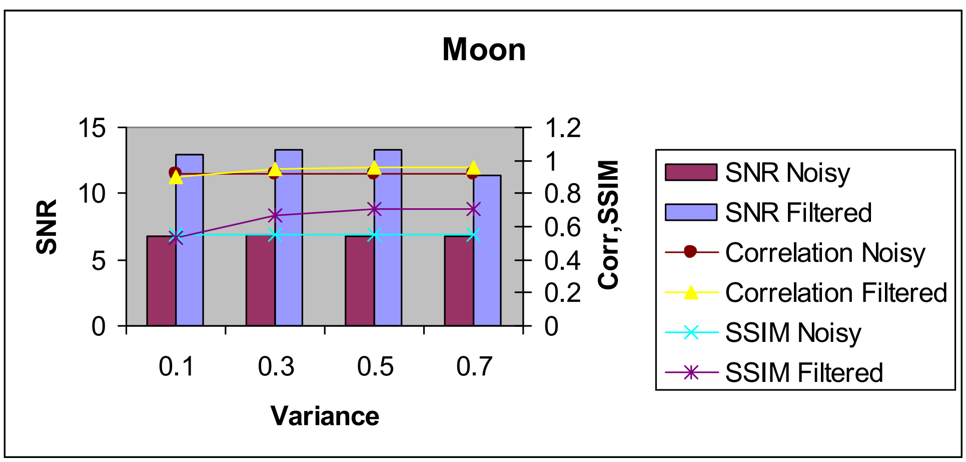
- Result For 'MOON.tiff' using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.4943	13.4521	0.7803	0.8984	0.1254	0.5289
0.3	4.475	13.46	0.8066	0.9337	0.2321	0.6075
0.5	4.4543	9.2646	0.8062	0.9249	0.2371	0.5665
0.7	4.4635	7.6099	0.8073	0.9004	0.2348	0.4525



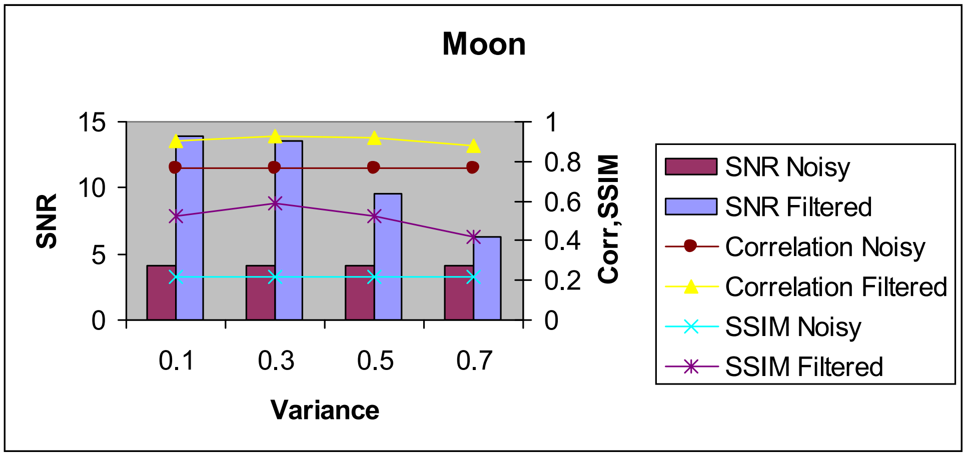
- Result For 'MOON.tiff' using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	6.8281	13.002	0.9189	0.8993	0.5502	0.5307
0.3	6.8726	13.3386	0.9193	0.9473	0.5534	0.6665
0.5	6.8265	13.3329	0.9192	0.954	0.5524	0.7049
0.7	6.8064	11.3508	0.9178	0.9563	0.5484	0.7017



- Result For ‘MOON.tiff’ using Speckle Noise.

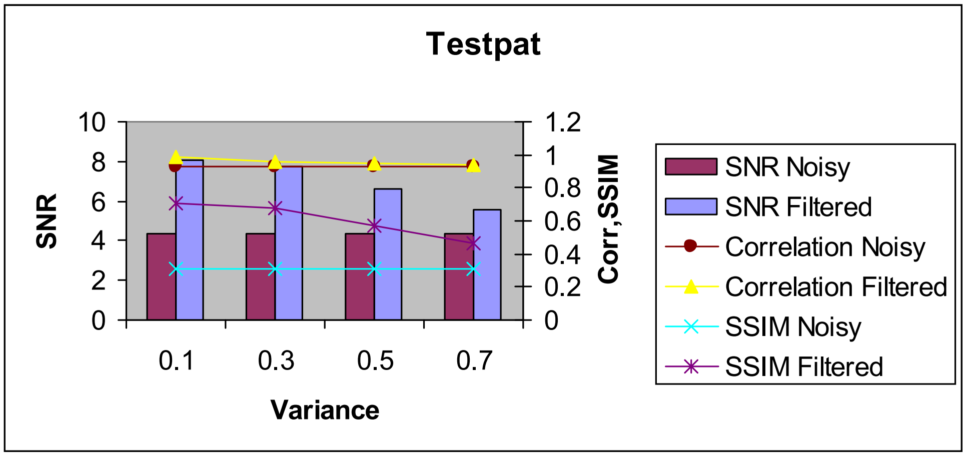
Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.149	13.9626	0.7696	0.9004	0.2162	0.5255
0.3	4.0995	13.5288	0.7641	0.9306	0.2182	0.5883
0.5	4.0958	9.5633	0.7692	0.9159	0.2152	0.5225
0.7	4.1144	6.2571	0.7672	0.8762	0.2195	0.4163



For Gaussian noise, SNR gives best result for the variance of 0.1, correlation coefficient gives best result for the variance of 0.3 and SSIM gives best result for the variance of 0.1. For Poisson noise, SNR gives best result for the variance of 0.5, correlation coefficient gives best result for the variance of 0.7 and SSIM gives best result for the variance of 0.5. For Speckle noise, SNR gives best result for the variance of 0.1. Correlation coefficient gives best result for the variance of 0.3. Similarly, SSIM gives best result for the variance of 0.1.

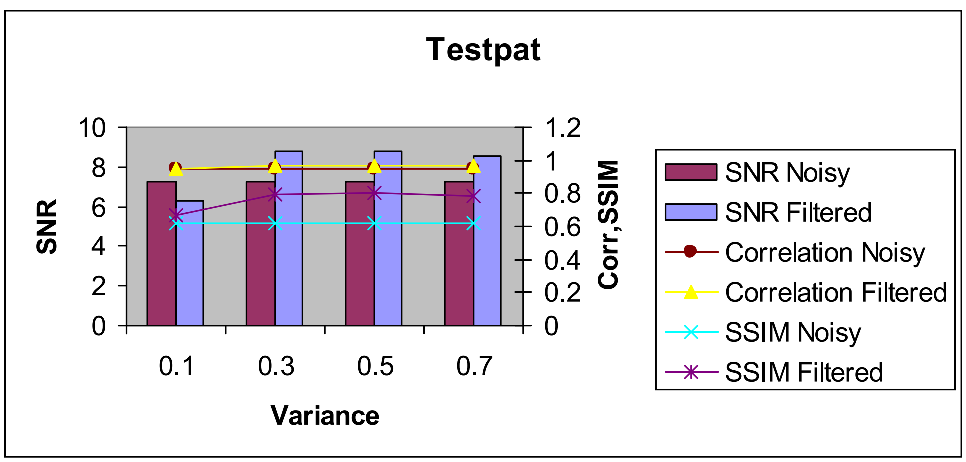
- Result For 'TESTPAT.tiff' using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.3557	8.0651	0.9316	0.9863	0.3111	0.7095
0.3	4.3662	7.7182	0.9316	0.9605	0.3118	0.6772
0.5	4.3836	6.5984	0.9328	0.9513	0.3115	0.5726
0.7	4.3675	5.5273	0.9328	0.9403	0.3109	0.4654



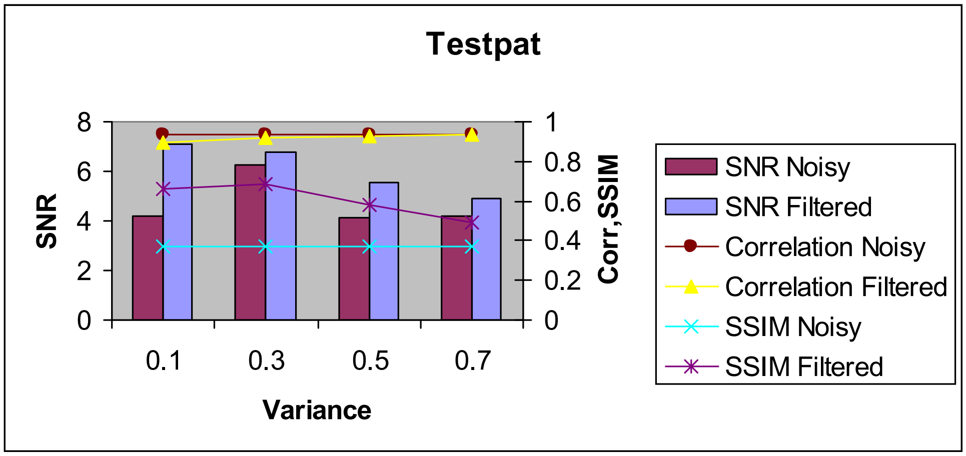
- Result For 'TESTPAT.tiff' using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	7.2689	6.3181	0.9452	0.9445	0.6218	0.6694
0.3	7.2527	8.7533	0.9448	0.9673	0.6231	0.7952
0.5	7.2632	8.8207	0.9445	0.9709	0.6226	0.7987
0.7	7.2485	8.5221	0.9444	0.9654	0.6219	0.7874



- Result For ‘TESTPAT.tiff’ using Speckle Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.2173	7.1188	0.9354	0.8976	0.3748	0.662
0.3	6.2357	6.7493	0.9354	0.9207	0.3731	0.6825
0.5	4.1523	5.5666	0.9352	0.9257	0.3705	0.5807
0.7	4.2157	4.9103	0.9356	0.9323	0.3737	0.4954



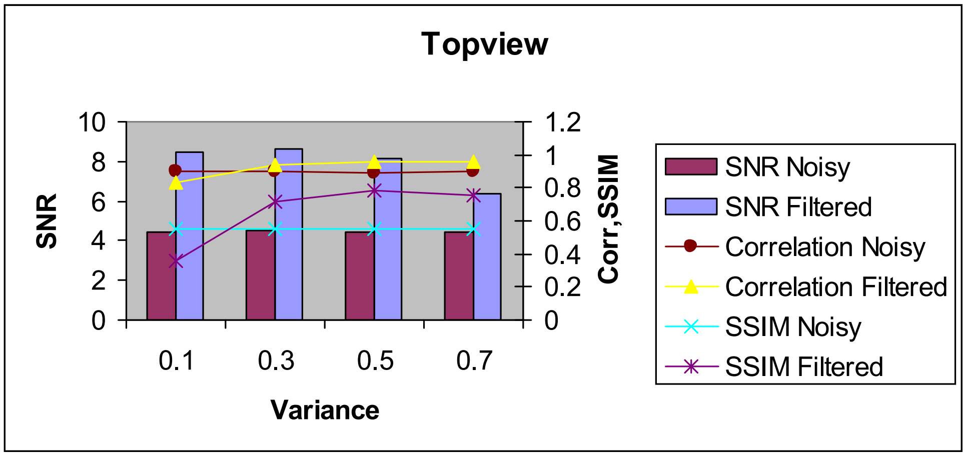
For Gaussian noise, SNR gives best result for the variance of 0.1. Similarly, correlation coefficient gives best result for the variance of 0.1 and SSIM gives best result for the variance of 0.1.

For Poisson noise, SNR gives best result for the variance of 0.5. Similarly, correlation coefficient gives best result for the variance of 0.5 and SSIM gives best result for the variance of 0.5.

For Speckle noise, SNR gives best result for the variance of 0.1. Correlation coefficient gives best result for the variance of 0.7 and SSIM gives best result for the variance of 0.3.

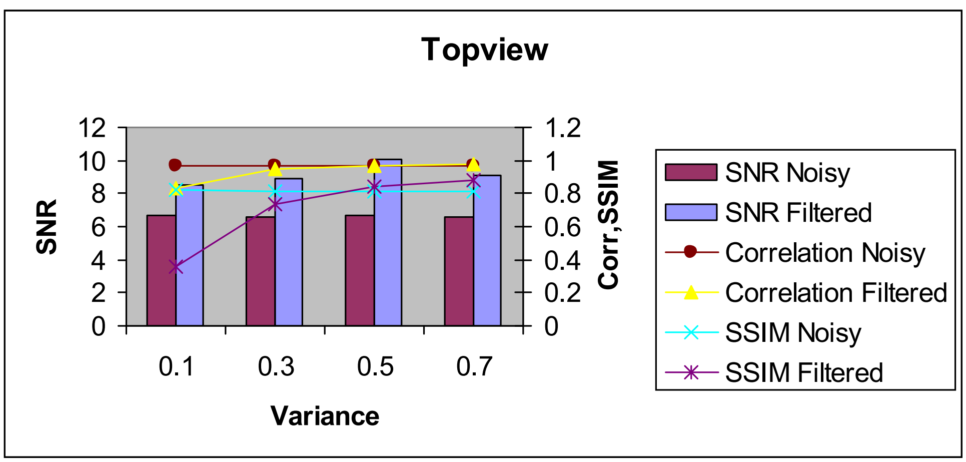
- Result For ‘TOPVIEW.tiff’ using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.4564	8.4836	0.8993	0.8333	0.5515	0.3564
0.3	4.4807	8.6191	0.8969	0.9388	0.5508	0.715
0.5	4.471	8.1613	0.8938	0.9551	0.5516	0.7849
0.7	4.4533	6.3741	0.8959	0.9547	0.5512	0.755



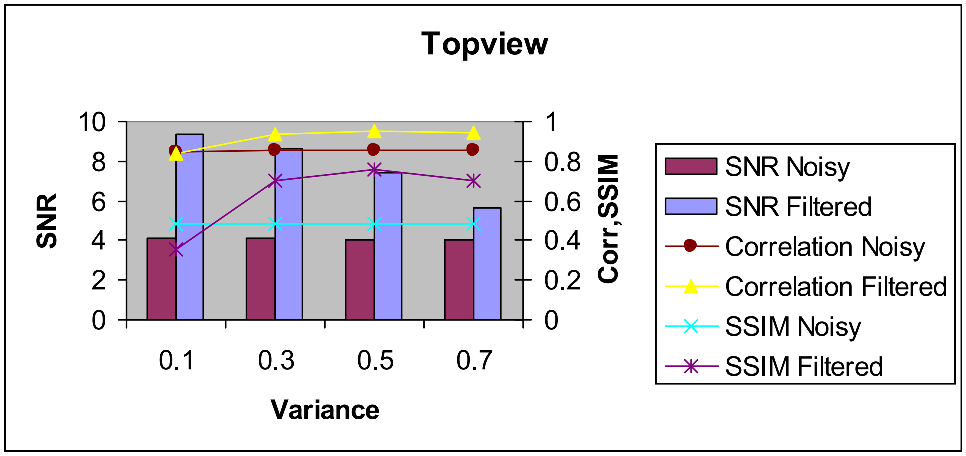
- Result For ‘TOPVIEW.tiff’ using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	6.6428	8.5201	0.9682	0.8368	0.8192	0.3562
0.3	6.6263	8.898	0.9686	0.9439	0.8149	0.7375
0.5	6.662	10.073	0.9675	0.9655	0.8163	0.846
0.7	6.5859	9.0507	0.9684	0.9793	0.8148	0.8778



- Result For ‘TOPVIEW.tiff’ using Speckle Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.0802	9.3336	0.8503	0.8376	0.4809	0.3529
0.3	4.1109	8.6021	0.8519	0.9358	0.4805	0.701
0.5	4.0605	7.3839	0.8533	0.9504	0.4803	0.7564
0.7	4.0436	5.6072	0.8536	0.9403	0.4801	0.7031



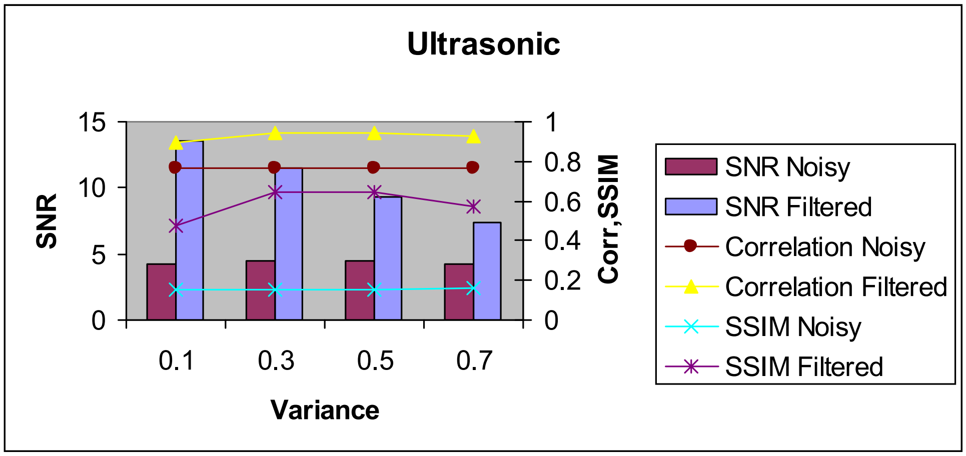
For Gaussian noise, SNR gives best result for the variance of 0.3, correlation coefficient gives best result for the variance of 0.5. Similarly, SSIM gives best result for the variance of 0.5.

For Poisson noise, SNR gives best result for the variance of 0.5, correlation coefficient gives best result for the variance of 0.7. Similarly, SSIM gives best result for the variance of 0.7.

For Speckle noise, SNR gives best result for the variance of 0.1. Correlation coefficient gives best result for the variance of 0.5. Similarly, SSIM gives best result for the variance of 0.1.

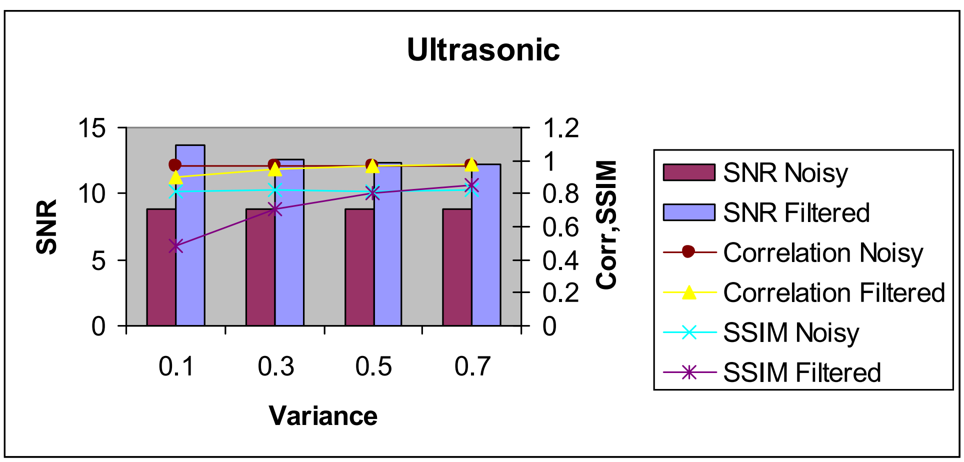
- Result For 'ULTRASONIC.tiff' using Gaussian Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	4.259	13.6032	0.7681	0.8972	0.1531	0.4778
0.3	4.5186	11.5361	0.7647	0.9406	0.1526	0.6433
0.5	4.5186	9.3486	0.7676	0.9454	0.1529	0.649
0.7	4.279	7.3245	0.7676	0.9314	0.1574	0.5705



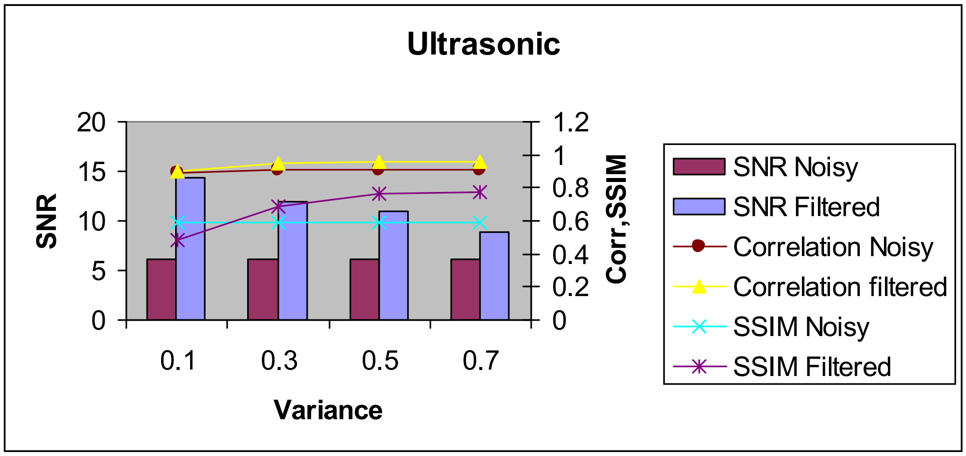
- Result For 'ULTRASONIC.tiff' using Poisson Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation Filtered	SSIM Noisy	SSIM Filtered
0.1	8.8252	13.6523	0.968	0.896	0.8169	0.4814
0.3	8.813	12.5224	0.9672	0.9479	0.8204	0.7031
0.5	8.8095	12.3453	0.9672	0.9661	0.8162	0.8012
0.7	8.7804	12.2269	0.9672	0.9728	0.8189	0.8472



- Result For ‘ULTRASONIC.tiff’ using Speckle Noise.

Variance	SNR Noisy	SNR Filtered	Correlation Noisy	Correlation filtered	SSIM Noisy	SSIM Filtered
0.1	6.0733	14.291	0.8949	0.8954	0.592	0.4795
0.3	6.1277	12.0146	0.9091	0.9475	0.5904	0.6914
0.5	6.1267	10.9152	0.9082	0.9587	0.5926	0.7648
0.7	6.1298	8.8056	0.9087	0.9602	0.5931	0.773



For Gaussian noise, SNR gives best result for the variance of 0.1, correlation coefficient gives best result for the variance of 0.5. Similarly, SSIM gives best result for the variance of 0.5.

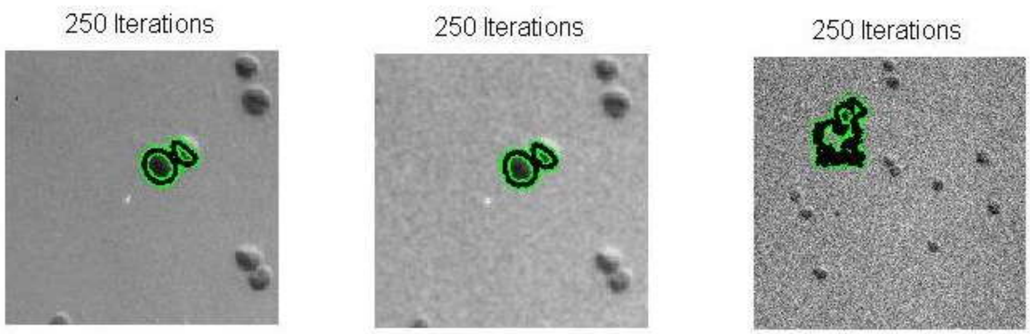
For Poisson noise, SNR gives best result for the variance of 0.1, correlation coefficient gives best result for the variance of 0.7. Similarly, SSIM gives best result for the variance of 0.7.

For Speckle noise, SNR gives best result for the variance of 0.1. Correlation coefficient gives best result for the variance of 0.7. Similarly, SSIM gives best result for the variance of 0.7.

Effect of Gabor filter for image segmentation

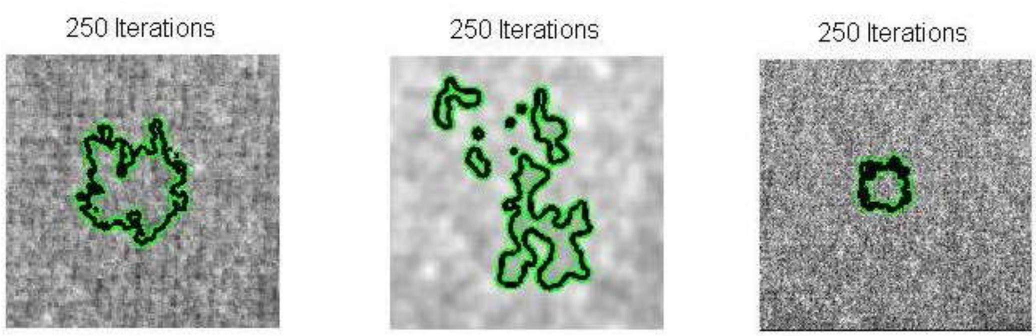
These are the comparison of Image segmentation for various images. There are three segmentation results for every image. These are the segmentation of Original image and the best values output for Noisy and filtered images.

- Result for the image segmentation for “ALGAE.tif”



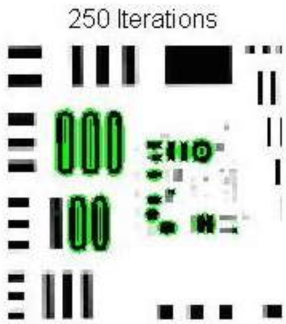
Original Image Filtered Image Noisy image

- Result for the image segmentation for “BLANKET.tif”

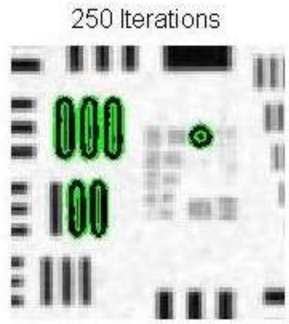


Original Image Filtered Image Noisy image

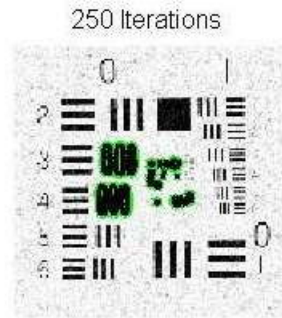
- Result for the image segmentation for “COUNTING.tiff”



Original Image

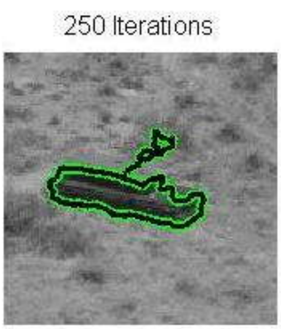


Filtered Image

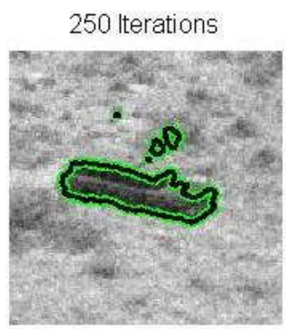


Noisy Image

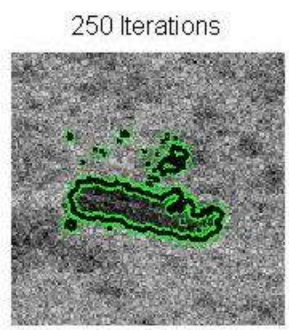
- Result for the image segmentation for “JUNGLE.tiff”



Original Image



Filtered Image

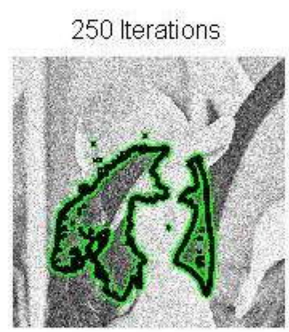


Noisy Image

- Result for the image segmentation for “LENA2D.jpg”

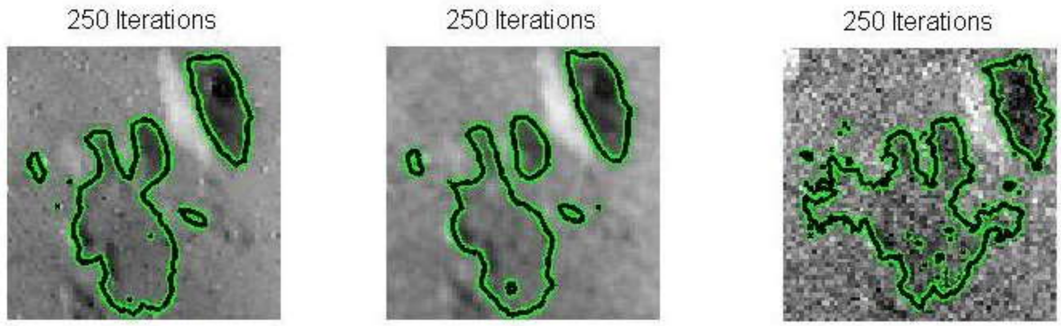
Original Image

Filtered Image



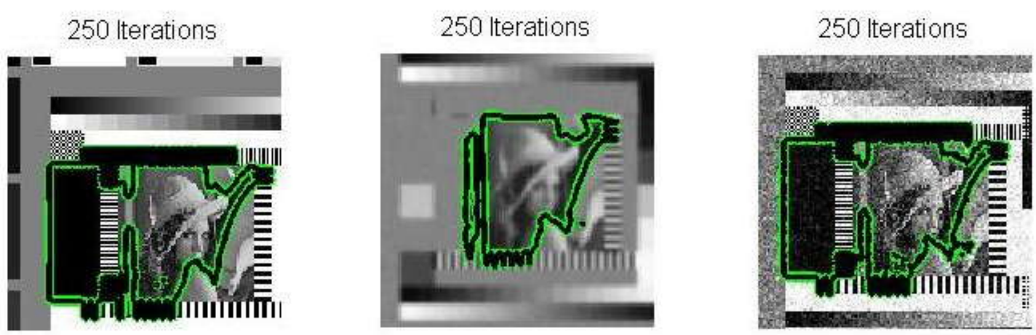
Noisy Image

- Result for the image segmentation for “MOON.tiff”



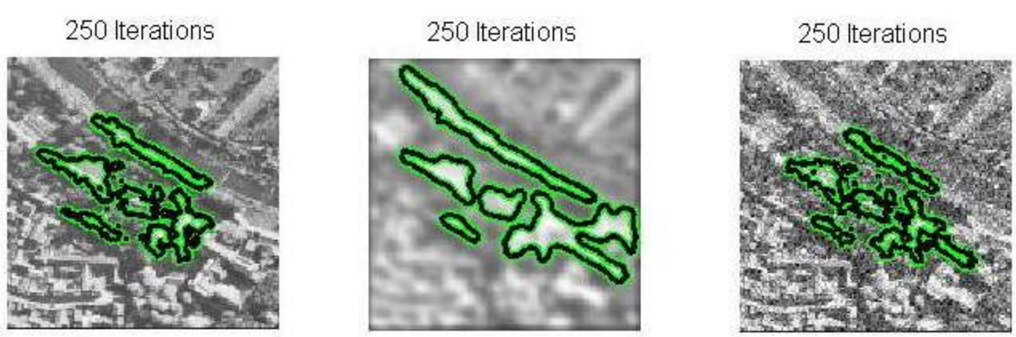
Original Image Filtered Image Noisy Image

- Result for the image segmentation for “TESTPAT.tiff”



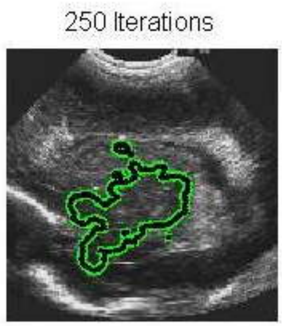
Original Image Filtered Image Noisy Image

- Result for the image segmentation for “TOPVIEW.tiff”

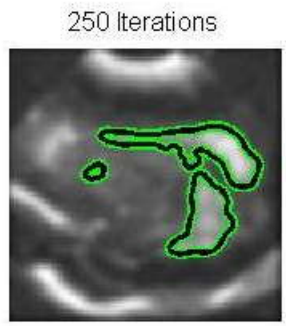


Original Image Filtered Image Noisy Image

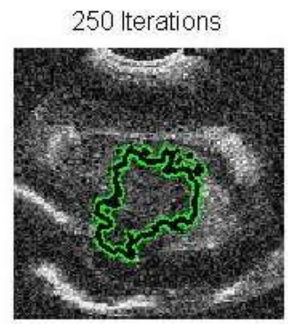
- Result for the image segmentation for "ULTRASONIC.tif"



Original Image



Filtered Image



Noisy Image

After all the image segmentation has been done, we observe that the images that contain noises not gave good segmentation results after applying Gabor filters. The rest images that do not have noisy are given best results after applying Gabor filters as compared to the original images. In these figures, Gabor filtered output images give best results as compared to the original image segmentation. And noisy images segmentation clearly indicates bad segmentation as compared to the other two images (original image and Gabor filtered output image).

CHAPTER-8
CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

This thesis presents evaluation of Gabor filter's parameters for various noisy and filtered images using Gabor filters. A lot of combinations of these noisy and filtered images have been obtained to find the best values as per the analysis of those two images of the quality metrics, SNR, Correlation and SSIM. The input image formats that have been used in this work are TIFF, TIF and JPEG and the output image format is having JPEG.

The analysis of all the obtained experimental results, demonstrates that the image noise is significantly reduced after applying Gabor filter. The parameters which are evaluated were Gabor-filtered output images give best output values for all the quality metrics and also give the best results for the image segmentation.

There are ten images in this thesis work. For the analysis of results, we using the variances values (0.1, 0.3, 0.5, 0.7) and for some frequency values (300, 500, 700, 900, 1500)Hz. Each image gives best response for some particular variances values and for some particular frequency values.

- Of all the frequency values that we are using, the frequency value 900 Hz gives the maximum best values for results.
- The frequency values 300, 500 and 700 Hz gives nearly around the same amount of best results.
- The frequency value 1500 Hz gives the least amount of best results.

For all the noises that we are used in this thesis work, each noise will give some different response from the other noises

- For Gaussian noise, the comparison of noisy and filtered images give the best results mostly for the variances of 0.1 and 0.3.
- For Poisson noise, the comparison of noisy and filtered images give the best results for all the working variances.

- For Speckle noise, the comparison of noisy and filtered images give the best results for the variances of 0.1.

For segmentation results, we can analyze all the three figures in the same plane. These three figures are: segmentation of original image, segmentation of both the noisy and filtered images that are different for all the ten images that gives best results as per the comparison between noisy and filtered images.

8.2 FUTURE SCOPE

The application of Gabor filters has been growing at a very fast rate. Gabor filters can be used for image segmentation, weed image classification, Palmprint recognition, Texture segmentation, for the illumination invariant recognition of color texture, for an automatic inspection system for textile fabrics and many places.

- This work can be further done in the field of texture image segmentation. And also works for various other parameters like MSE etc.
- Moreover, for future work we can use various AI techniques like Radon neural network, Fuzzy, Adaptive, GA in order to attain the best output without performing calculations for each and every combination. This work can be done by using this technique will lead to more efficiency and less tedious work.
- This work can be done using the technique of frequency domain analysis using fractals, FFTs.

REFERENCES:

1. Z.-Q. Liu, R.M. Rangayyan and C.B.Frank, "Analysis directional features in images using Gabor filters" pp.68-74, 1990 COMPUTER-BASED MEDICAL SYSTEMS, PROCEEDINGS OF THIRD ANNUAL IEEE SYMPOSIUM, 1990.

2. Jiang Wen, You Zhisherg and Li Hui, "Segment the Metallograph Images Using Gabor Filter" pp.25-28,vol.1,1994 INTERNATIONAL SYMPOSIUM ON SPEECH, IMAGE PROCESSING AND NEURAL NETWORKS, 1994,Hong Kong.

3. Richard Buse and Zhi-Qiang Liu, "Feature extraction and analysis of handwritten words in Grey-scale images using Gabor filters" pp.164-168,vol.1, PROCEEDINGS ICIP-94, IEEE INTERNATIONAL CONFERENCE on IMAGE PROCESSING, 1994.

4. Andreas Teuner, Olaf Pichler and Bedrich J. Hosticka, "Unsupervised Texture Segmentation of Image Using Tuned Matched Gabor fiters" pp. 863-870, vol. 4, IEEE TRANSACTIONS ON IMAGE PROCESSING, 1995.

5. Shuzo Yamamoto, Yoshikazu Nakajima, Shinichi Tamura, Yoshinobu Sato and Seiyo Harino, "Extraction of fluorescent dot traces from a scanning Laser Opthalmoscope Image sequence by Spatio-Temporal Image analysis: Gabor Filter and Radon Transform Filtering" pp.1357-1363, vol. 46, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, 1999.

6. Ian R Fasel, Marian S Barlett and Javier R Movellan, "A comparison of Gabor filter methods for automatic detection of Facial landmarks" pp.242-246, FIFTH IEEE INTERNATIONAL CONFERENCE ON AUTOMATIC FACE AND GESTURE RECOGNITION, 2002.

7. C.Klimanee and DT Nguyen, "On the design of 2-D Gabor Filtering of Fingerprint Images"pp.430-435, CONSUMERUMER COMMUNICATIONS AND NETWORKING CONFERENCE, 2004. CCNC 2004. FIRST IEEE , 2004.

8. Rong Lu and Yi Shen, "Image segmentation Based on Random Neural Network Model and Gabor filters" pp. 6464-6467, PROCEEDINGS OF 2005 IEEE ENGINEERING IN MEDICINE AND BIOLOGY 27TH ANNUAL CONFERENCE, 2005.

9. Ying-Chun Li, Zhan-Chun Li, Yun-Huan Mei and Jian-Xin Zhang, "Detecting algorithms based Gabor in Microscopic image" pp.5410-5415, PROCEEDINGS OF THE FOURTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, GUANGZHOU, 2005.

10. Hany Ayad Bastawrous, Takuya Fukumoto, Norihisa Nitta and Masaru Tsudagawa, "Detection of ground glass opacities in lung CT images using Gabor filters and neural networks",pp.251-256,INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE OTTAWA, CANADA, 2005.

11. Hong Wei and Marc Bartels, "Unsupervised segmentation Using Gabor wavelets and statistical features in LIDAR Data analysis" pp.667-670,vol.1, PROCEEDINGS OF THE 18TH INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 2005.

12. K. L. Mak and P. Peng, "An automated inspection system for textile fabrics based on Gabor filters"pp.359-369,vol.24,DEPARTMENT OF INDUSTRIAL AND MANUFACTURING SYSTEMS ENGINEERING, THE UNIVERSITY OF HONG-KONG, 2007.

13. Mohammed Al-Rawi and Jie Yang, "Using Gabor Filter for the illumination invariant recognition of color texture"pp.550-555,vol.77,INSTITUTE OF IMAGE PROCESSING AND PATTERN RECOGNITION, SHANGHAI JIAO TONG UNIVERSITY, PR CHINA, 2007.

14. Jesmin F. Khan, Reza R. Adhani and Sharif M.A. Bhuiyan, "A customized Gabor filter for unsupervised color image segmentation"pp.489-501,vol.27, DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, UNIVERSITY OF ALABAMA IN HUNTSVILLE, USA, 2008.

15. Xin Pan and Qiu-Qi Ruan, "Palmprint recognition using Gabor-based local invariant features"pp.2040-2045,vol.72, INSTITUTE OF INFORMATION SCIENCE, BEIJING JIAOTONG UNIVERSITY, PR CHINA, 2008.

16. J. Bossu, Ch. Gee, G. Jones and F.Truchetet, "Wavelet transform to discriminate between crop and weed in perspective agronomic images"pp.133-143, vol.65, 21 BLD OLIVIER DE SERRES, FRANCE, 2008.

17. Asnor Juraiza Ishak, Aini Hussain and Mohd Marzuki Mustafa, "Weed image classification using Gabor wavelet and gradient field distribution"pp.53-61,vol.66, DEPARTMENT OF ELECTRICAL, ELECTRONIC AND SYSTEMS ENGINEERING, FACULTY OF ENGINEERING AND BUILT ENVIRONMENT, UNIVERSITI KEBANGSAAN MALAYSIA, MALAYSIA 2008.

18. Minqin Wang, Guoqiang Han, Yongqiu Tu, Guohua Chen and Yuefang Gao, "Unsupervised Texture Image segmentation Based on Gabor Wavelet and multi-PCNN"pp.376-381,vol.2 SCHOOL OF COMPUTER SCIENCE AND ENGINEERING, SOUTH CHINA UNIVERSITY OF TECHNOLOGY, CHINA 2008.

19. Gholam Ali Rezaei Rad and Kaveh Samiee, “Fast and modified Image segmentation Method Based on Active Contours and Gabor filter”pp.1-5, ELECTRICAL ENGINEERING DEPARTMENT OF IRAN UNIVERSITY OF SCIENCES AND TECHNOLOGY, 2008.

20. Charles E. Connor and Kenneth O. Johnson, “Natural Coding of Tactile Texture: Comparison of Spatial and Temporal Mechanics for roughness Perception”pp.3414-3426, THE JOURNAL OF NEUROSCIENCE, 1992.

21. Rafael C.Gonzalez, Richard E.Woods and Steven L.Eddins, “Digital Image Processing Using MATLAB”, PEARSON EDUCATION (SINGAPORE) PTE LTD., INDIAN BRANCH, 482 F.I.E. PATPARGANJ, DELHI 110092, INDIA, 2004.

22. Javier R. Movellan, “Tutorials on Gabor Filters”, pp.1-20,GNU Free documentation License 1.1,Kolmogorv Project,2002