STUDY AND IMPLEMENTATION OF MORPHOLOGY FOR SPECKLE NOISE REMOVAL AND EDGE DETECTION

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

Master of Engineering

In

Electronic Instrumentation and Control

Submitted by

Arpit Singhal

Roll No. 800951003

Under the Guidance of

Mr. M.D. Singh

Assistant Professor

Department of Electrical and Instrumentation Engineering

Thapar University

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

Patiala, 147004, Punjab, India

July 2011

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled **"Study and implementation of morphology for speckle noise removal and edge detection"** in partial fulfillment of award of degree of **Master of Engineering** in **Electronics Instrumentation and Control** submitted in Electrical and Instrumentation Engineering department, Thapar University, Patiala is an authentic record of my own work carried under the supervision of **Mr. M.D Singh**, Assistant Professor, Department of Electrical and Instrumentation Engineering, Thapar University, Patiala, Punjab.

Date: 14/07/11

Arpit Singhal
800951003

I certify that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 14/07/11

Mr. M.D. Singh
Assistant Professor
Department of Electrical and Instrumentation Engineering
Thapar University, Patiala
Punjab

Countersigned By

S. Ghosh 14/7/11

Dr. Smarajit Ghosh
Head of Department
Department of Electrical and Instrumentation Engineering
Thapar University, Patiala
Punjab

Dr. S. K. Mohapatra
Dean of Academic Affairs
Thapar University, Patiala
Punjab

23/9/2011

Dr. Manjit Singh
Associate Prof. ECE
UCoE, Pbi Univ, PTA.

ABSTRACT

The field of computer vision is concerned with extracting features and information from images in order to make analysis of images easier, so that more and more information can be extracted. The primary goal of this thesis is to remove speckle noise present in images used and then to obtain the useful edges in the output image obtained after noise has been removed. The method use for both noise removal and edge detection is mathematical morphology. On the basis of set theory, mathematical morphology is used for image processing, analyzing and comprehending. It is a powerful tool in the geometric morphological analysis and description.

Noise removal is very important preprocessing step. The existence of speckle is unwanted since it disgrace image quality and it affects the tasks of individual interpretation and diagnosis. Eliminating such noise is an important preprocessing step. Nonlinear filtering techniques are becoming increasingly important in image processing applications, and are often better than linear filters at removing noise without distorting image features. One structure for designing nonlinear filters is mathematical morphology. By using combination of mathematical morphological operations like opening and closing help in removing speckle noise very effectively. The next step after noise removal is edge detection. The need of edge detection is to find the discontinuities in depth, discontinuities in surface orientation, changes in material properties and variations in scene illumination. Again mathematical morphological operations are used for edge detection and enhancement of edges that are detected.

For the evaluation of the performance of proposed algorithm parameters like Signal to Noise Ratio (SNR), Root Mean Square Error (RMSE), Correlation of Coefficient (Coc), and Mean Structural Similarity Index Measure (MSSIM). These parameters are used to calculate the image quality of the output image obtained after speckle noise removed by proposed algorithm and help in comparing the results with that of previously used filters for speckle noise. Another parameter named edge preserving index (EPI) is used for measuring the edges preserved and detected by proposed algorithm with that of previously used edge detectors like Sobel, Prewitt and Canny. Based on the values of these parameters the performance of algorithm in terms of speckle noise removal and edge preservation is discussed.

ACKNOWLEDGEMENT

The real spirit of achieving a goal is through the way of excellence and austere discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various personalities.

With deep sense of gratitude I express my sincere thanks to my esteemed and worthy supervisor, **Mr. M. D. SINGH ,** Assistant Professor**,** Department of Electrical and Instrumentation Engineering, Thapar University, Patiala for his valuable guidance in carrying out this work under his effective supervision, encouragement, enlightenment and cooperation. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. His feedback and editorial comments were also invaluable for writing of this thesis.

I shall be failing in my duties if I do not express my deep sense of gratitude towards **Dr. Smarajit Ghosh,** Professor and Head of the Department of Electrical & Instrumentation Engineering, Thapar University, Patiala who has been a constant source of inspiration for me throughout this work.

I am also thankful to all the staff members of the Department for their full cooperation and help.

This acknowledgement would be incomplete if I do not mention the emotional support and blessings provided by my friends. I had a pleasant enjoyable and fruitful company with them.

My greatest thanks are to all who wished me success especially my parents, sisters whose support and care makes me stay on earth.

Place: Thapar University, Patiala **Arpit Singhal**
Date: **800951003**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF GRAPH

LIST OF ABBREVATION

3-D	Three-dimensional
GIF	Graphics Interchange Format
JPEG	Joint Photographic Experts Group
MATLAB	Matrix Laboratory
MM	Mathematical Morphology
SE	Structuring Element
RGB	Red-Green-Blue
MMC	Morphological Multiscale Characteristic
SARD	Speckle Reducing Anisotropic Diffusion
LSIM	First Order Local Statistical Filter
EPI	Edge Preserving Index
MSSIM	Mean Structural Similarity Index Measure
RMSE	Root Mean Square Error
SNR	Signal to Noise Ratio
ENL	Equivalent Number of Look
SAR	Synthetic Aperture Radar

CHAPTER-1
INTRODUCTION

1.1 Overview

Noise removal and edge detection are the two most important steps in processing of any digital images for improving the information in the picture so that it can be easily understand by human and to make it suitable and readable for any further processing which works on those images. There are many edge detection methods like sobel, prewitt, canny edge detectors and noise removal techniques like filters according to the type of noise present for digital image processing. With advancement in technology in image acquisition and analysis systems new methods are still require for high level of noise removal and for low contrast edge detection. Mathematical morphology (MM) is a new mathematical theory which can be used to process and analyze the images. In the MM theory, images are treated as sets and morphological transformations which derived from Minkowski addition and subtraction are defined to extract features in images.

1.2 Background

Image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or, a set of characteristics or parameters related to the image. Image processing involves changing the nature of an image in order to either

1) Improve its pictorial information for human interpretation, or
2) Render it more suitable for automatic machine perception.

These two aspects represent two separate but equally important applications of image processing. A procedure that satisfies condition 1- a procedure that makes an image looks better- may be the very worst procedure for satisfying condition 2. Human like their images to be sharp, clear, and detailed; machine prefer their image to be simple and uncluttered.

Example of condition 1 may include

- Enhancing the edges of an image to make it appear sharper. Sharpening edges is a vital component of printing.

- Removing noise from an image, noise being random errors in the image. Noise is a very common problem in data transmission: all sorts of electronic components may affect data passing through them, and the results may be undesirable. Noise may take many different forms and each type of noise required a different method of removal.
- Removing motion blur from an image. Motion blur may occur when the shutter speed of the camera is too long for the speed of the object.

Examples of condition 2 may include

- Obtaining the edges of an image. This may be necessary for the measurement of objects in an image. Once we have the edges we can measure their spread and the area contained within them.
- Removing details from an image. For measurement or counting purposes, we may not be interested in all the detail of an image.

The main concern is digital image processing which involves using a computer to change nature of digital images. Digital image processing is the use of computer algorithms to perform image processing on digital images.

Each imaging system suffers with a common problem of "Noise". Unwanted data which may reduce the contrast deteriorating the shape or size of objects in the image and blurring of edges or dilution of fine details in the image may be termed as noise. It may be due to one or more of the following reasons

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

Mathematically there are two basic models of Noise; additive and multiplicative. Additive noise is systematic in nature and can be easily modeled and hence removed or reduced easily. Whereas multiplicative noise is image dependent, complex to model and hence difficult to reduce. When multiplicative noise caused by the de-phased echoes from the scatterers appears, it is called "Speckle Noise". Although it appears as noise but it also contains useful information because it is due to surroundings of the target. Speckle may appear distinct in different imaging systems but it is always manifested in a granular

pattern due to image formation under coherent waves. Speckle is the result of the diffuse scattering, which occurs when a sound wave (RFsound or Ultrasound) pulse randomly interferes with the small particles or objects on a scale comparable to the sound wavelength. In most cases, it is considered a contaminating factor that severely degrades image quality. To improve image analysis, speckle reduction is generally used for two applications:

- Visualization enhancement
- Auto-segmentation improvement.

Edges characterize object boundaries and are useful for segmentation, registration and identification of objects in scenes. Edge points can be thought of as 4 pixel locations of abrupt gray level change. For example, it is reasonable to define edge points in binary images as black pixels with at least one white nearest neighbor. The goal of edge detection is to mark the points in a digital image at which the luminous intensity changes sharply. Sharp changes in image properties usually reflect important events and changes in properties of the world. These include (i) discontinuities in depth, (ii) discontinuities in surface orientation, (iii) changes in material properties and (iv)variations in scene illumination. Edge detection of an image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image.

1.3 Objective of the Thesis

In this thesis the pre-processing steps of speckle noise removal and edge detection are done by using mathematical morphology. Mathematical morphology is a new mathematical theory which can be used to process and analyze the images. It provides an alternative approach to image processing based on shape concept stemmed from set theory, not on traditional mathematical modeling and analysis. Different mathematical operations like dilation, erosion, opening and closing are used in different combinations with structuring element “disk” taken with its size varying for making a non-linear morphology filter which is use to remove speckle noise from the images. After the removal of noise, edges are obtained by using different operations of mathematical morphology. Image quality assessment parameters like Signal to Noise Ratio (SNR),

Root Mean Square Error (RMSE), Correlation of Coefficient (Coc) and Mean Structural Similarity Index Measure (MSSIM) are used for comparing the proposed algorithm with the previously used filters like LEE filter, LSMV filter and SRAD filter .The Edge preserving index (EPI) is used for comparing the proposed algorithm with the previously used edge detectors like sobel, prewitt and canny edge detectors respectively. In brief the main objective is to find a MM based solution for speckle noise reduction and edge detection.

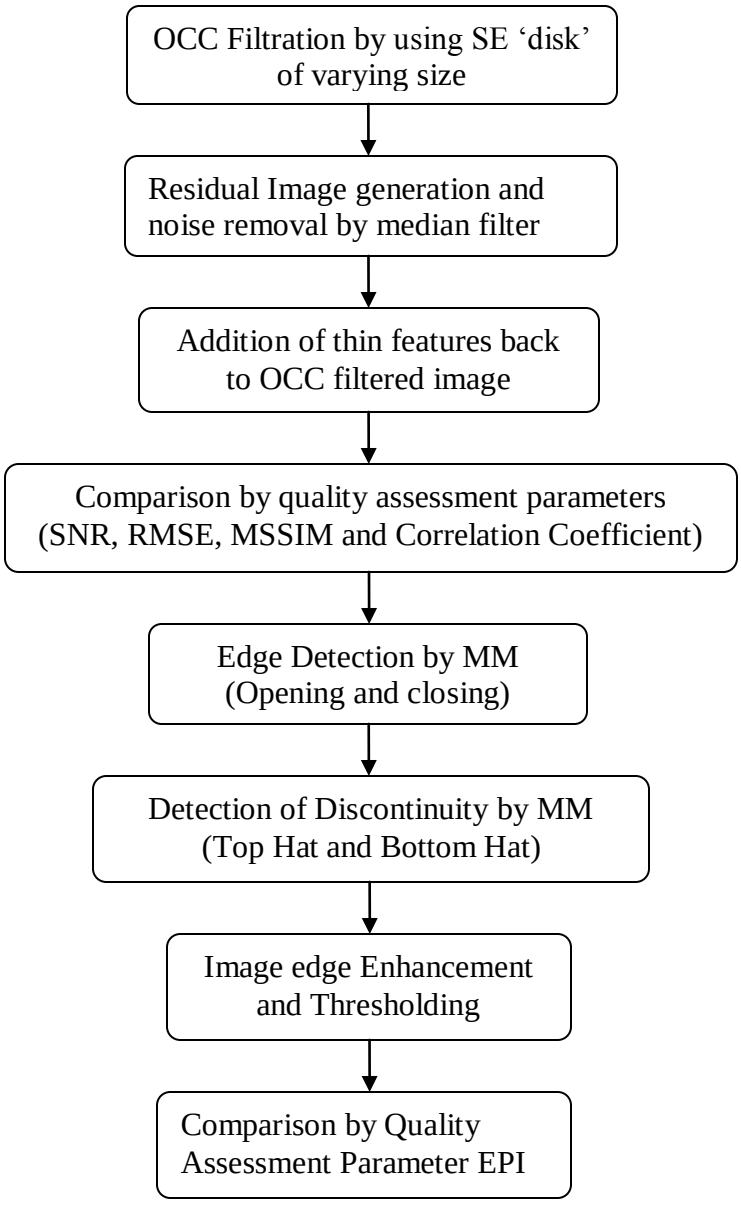


Figure 1.1 Flowchart representing overview of the proposed method

1.4 Organization of Thesis

- Chapter 1 "**INTRODUCTION**", it includes the introduction about the objective and overview of the work done in this thesis.
- Chapter 2 "**LITERATURE SURVEY**", it provides the overview of the work done in this area.
- Chapter 3 "**BACKGROUND**", it gives the basics of topics which are useful in understanding complicated concepts in digital image processing and the work done. These are digital image, noise in image, speckle noise, segmentation, filters and statistical parameters.
- Chapter 4 "**MORPHOLOGY**" explains all the basic concepts of mathematical morphology for both binary and gray-scale images.
- Chapter 5 "**METHODOLOGY**", gives the algorithm use for speckle noise removal from images and algorithm for edge detection.
- Chapter 6 "**RESULTS AND DISSCUSSION**", presents output images, statistical parameters and graphs obtained after applying the proposed method on different images.
- Chapter 7 "**CONCLUSION AND FUTURE SCOPE**", conclude thesis with future scope.

CHAPTER-2
LITERATURE SURVEY

Deng Shiwei et al. [1], 1993 introduced a range image segmentation method using simple morphological operators. They described how to obtain peak and valley in an image using edge-strength operator and peak and valley detection operators. These procedure makes the depth discontinuities contained in the peaks and the orientation discontinuities in the valleys. The depth discontinuities are extracted and the segmentation is accomplished by controlled region growing. The experimental results prove this range image segmentation technique quick and robust

Richard Alan Peters [2], 1995 did an important work on image noise reduction and also limited their tendency to remove important, thin features from an image along with the noise. He introduce new morphological image cleaning algorithm (MIC) that preserves thin features while removing noise. MIC is useful for gray-scale images corrupted by dense, low-amplitude, random, or patterned noise. Such noise is typical of scanned or still-video images. MIC differs from previous morphological noise filters in the way that it manipulates residual images which is the difference between the original image and morphologically smoothed versions. It calculates residuals on a number of different scales via a morphological size distribution. It discards regions in the various residuals that it judges to contain noise. MIC creates a cleaned image by recombining the processed residual images with a smoothed version.

Maragos P [3], 1996 introduced image processing via mathematical morphology by geometry to intuitively understand morphological signal operators and set or lattice algebra to analyze them in the space domain. He provided a unified view and analytic tools for a recently growing part of morphological image processing that is based on ideas from differential calculus and dynamical systems. This part includes both recent and some earlier ideas on using partial differential or difference equations (PDEs) to model distance propagation or nonlinear multiscale processes in images.

Yongjian Yu et.al [4], 2003 generalized the 2D SRAD algorithm developed for despeckling ultrasound images to obtain a SRAD algorithm capable of enhancing volumetric ultrasound data. In this work Firstly, an instantaneous coefficient of variation

edge detector (3D ICOV) appropriate for three-dimensional data set is presented then, the 2D SRAD partial differential equation is formulated by replacing the gradient operator in the traditional diffusion mechanism with the derived 3D ICOV operator. Experiments are conducted firstly by using 3D SRAD algorithm tested on a synthetic 3D dataset as acquired from an ellipsoid using a hypothetical 2D B-scan imager. 2D and 3D algorithms are compared quantitatively by using average edge sharpness, average intensity in a band encircling that object, and coefficient of variation.

Zhao Yuqian et al. [5], 2005 did an important work for object recognition of the human organs which is an important pre-processing step in medical image segmentation and 3D reconstruction. The proposed algorithm used to detect the edges of lungs CT images with salt-and-pepper noise which is more efficient for denoising and the edge detection than the usually used template-based edge detection algorithm such as LOG operator and Sobel edge detector and general morphological edge detection algorithm such as morphological gradient operation and dilation residue edge detector.

Ying-Tung Hsiao et al. [6], 2005 proposed an image segmentation algorithm by integrating mathematical morphological edge detector with region growing technique. The images are first enhanced by morphological closing operations, and then detect the edge of the image by morphological dilation residue edge detector. Moreover, this algorithm deploy growing seeds into the edge image that obtained by the edge detection procedure. By cross comparing the growing result and the detected edges, the partition lines of the image are generated. The proposed image segmentation approach is capable of producing promising segmentation results.

Yuqian Zhao et al. [7], 2006 proposed a novel algorithm to detect complex edge features using multistructure mathematical morphological approach of eight different directions which are obtained by using morphological gradient algorithm that are not possible with conventional mathematical edge detection methods because they are only sensitive to image edge which has the same direction of structure element. The final edge result was got by using synthetic weighted method. The proposed algorithm is more efficient than the usually used single and symmetrical structuring element morphological edge detection operator and differential edge detection operators such as canny operator, LOG operator, Sobel operator and Prewitt operator. The detected edge is more

pinpointed, integral and continual, and the edge information is more abundant. Moreover, the novel proposed algorithm can filter the noise more successfully than other operators

J.-A. Jiang et al. [8], 2007 proposed edge detector based on mathematical morphology to preserve thin edge features in low-contrast regions as well as other apparent edges. A quad-decomposition edge enhancement process, a thresholding process, and a mask-based noise filtering process were developed and used to enhance thin edge features, extract edge points and filter out some meaningless noise points, respectively. Moreover, five bipolar oriented edge masks were also designed to remove most of the incorrectly detected edge features. The proposed method is capable of enhancing the low contrast edges.

Liu Ting et al. [9], 2007 did an important work on an improved edge detection operator for the ultrasound images of cardiac ventricular wall with strong noises and fuzzy edges detected in the motion of their rotation. The proposed algorithm modified the combination of morphological operations, so that the unclear edges of images are avoided. Furthermore, multi-structure elements were also introduced which can reserve integrated edges from different directions of the images. The results demonstrate that this edge detector has a better performance on the edge detection of ventricular wall. It can not only keep the edges more accurate than traditional edge detectors, but also satisfy the request of coherent ventricular wall in the analysis of Ultrasound heart images.

Li-Hui Jiang et al. [10], 2007 introduced a new morphological image-cleaning algorithm that preserves thin features while removing speckle noise is presented and analyzed. In proposed algorithm, the multi-scale top-hat transformation and bottom-hat transformation are added into the conventional multi-scale morphological opening and closing filtering. It differs from previous morphological filters in which it uses residual images to extract and smooth the features. The coefficients of the multi-scale top-hat transformation and bottom-hat transformation are optimized by adaptive weighted technique. They also show that the new filtering algorithm preserves thin features significantly and reduces the speckle index greatly.

Serge Beucher [11], 2007 introduced the concept of binary morphological transformation based on the residual image and was extended on the analysis of the residue evolution in every point on image. Two of the numerical residue, called ultimate

opening and quasi-distance were introduced and through some applications interest and applications of these operators was illustrated.

Xueshun Wang et al. [12], 2008 introduced a new mathematical morphological double-gradient algorithm used for edge detection of decayed wood images. They chose structuring elements appropriately, in order to suppress noises and be adapted to different edges of images. The proposed algorithm is constructed by weight adding combination of morphological operation. The results of simulation in decayed wood image processing demonstrate that the method performs better in noise suppression and edge detection than conventional edge detection operations.

Chaofeng Li et al. [13], 2008 worked on assessment of image quality is for numerous image processing applications, where the goal of image quality assessment (IQA) algorithms is to automatically assess the quality of images in a manner that is consistent with human visual judgment. Structural Similarity Image Metric (SSIM) and Multi-scale Structural Similarity (MS-SSIM) were introduced by them which operate under the assumption that human visual perception is highly adapted for extracting structural information from a scene. Results in large human studies shows that these quality indices perform very well relative to other methods.

Yan Wenzhong [14], 2009 worked on chromosome images were acquired by microscopic imaging of metaphase or prophase cells on specimen slides. Digitized chromosome images usually suffer from poor image quality, particularly lack of contrast and presence of shading and artifacts. So these images must be enhanced. He presented an enhancement algorithm for chromosome images based on mathematical morphology. Firstly, the top-hat transform and bot-hat transform were used to improve the contrast of the images. Then, the iterative threshold segment, closing operation and opening operation were operated on the result respectively. The final result was obtained by the and operation being operated on the two results. To validate the effect of proposed algorithm, the canny edge detection operator was used to detect the edges of chromosomes in the original images and those being enhanced by this algorithm respectively. The experiment proved that the algorithm can improve the contrast of the image effectively and also can remove noise of the image.

Waheeb Abu Ulbeh et al. [15], 2009 proposed a methodology using mathematical morphology for eliminating the noise from the gray image and reconstructing the image in order to get the image without losing any piece of information from it. The methodology was implemented and tested and the experimental results shows that the proposed methodology is efficient comparing with the results obtained by applying standard methods used to eliminate noises such as Gaussian, Sobel, prewitt, laplacian, log, average, unsharp methods which are used as a special filters to eliminate the noise from the gray image.

S.Sudha et al. [16], 2009 worked on a wavelet-based thresholding scheme for noise suppression in ultrasound images. The noise is speckle noise which is to be removed since it restrains information which is valuable for the general practitioner.

Mandeep Singh at al. [17], 2009 worked on various spatial domain filters for speckle suppression in ultrasound images. They proposed that the spatial domain filter are easy to implement on real time systems because they work faster then other methods like multi-resolution or wavelet based filters. The performances of different filters are evaluated by parameters like SNR, COC, QI, SSI and EPI.

T.Ratha Jeyalakshmi et al. [18], 2010 worked on an algorithm for cleaning speckle noise in ultrasound medical images. They use mathematical morphological operations based on Morphological Image Cleaning algorithm (MIC) designed by Richard Alan Peters II. The algorithm uses a different technique for reconstructing the features that are lost while removing the noise. They also used arbitrary structuring elements suitable for the ultrasound images which have speckle noise.

T.Ratha Jeyalakshmi et al.[19], 2010 worked on image segmentation and feature extraction for analysis of ultrasound image of uterine fibroids are the most common pelvic tumors in females. The method used them employs a number of knowledge-based rules to locate the object and also utilizes the concepts in mathematical morphology. It also extracts the necessary features of the fibroid which can be used to prepare the radiological report

M Rama Bai et al. [20], 2010 proposes a novel approach for noise removal cum edge detection for both gray scale and binary images using morphological operations. Two images consisting of noise are processed and the effectiveness of the proposed

approach is experimentally demonstrated. The results demonstrate that the proposed filter cum edge detector approach overcomes the deficiency of conventional methods and efficiently removes the noise and detects the edges.

Ahmed S. Mashaly et al. [21], 2010 proposed an adaptive mathematical morphological filter to reduce the speckle noise in SAR images. The new filter performance is compared with a number of despeckling filters with different parameters. For performance measurements, several parameters were evaluated to test the filter ability to attenuate the speckle noise and keep target information.

Shahana Bano et al. [22], 2010 worked on image segmentation by mathematical morphology based upon the notions of reconstruction and gradient method of an image. They proposed new method based on the notion of regional maxima and makes use of sequential reconstruction algorithm and morphological gradient .They divided their work in two main sections, first is reconstruction of original Image from blurred image by eliminating noise. Second is segment the image by applying morphological gradient method this method produced good result over conventional methods.

Dr. H. B. Kekre et al. [23], 2010 did an important work on edge detection which isextensively used in image segmentation when images are divided into areas corresponding to different objects. They used Sobel, Prewitt and Kirsch edge operators for image segmentation of mammography images for enhancing the tumor area. For comparison purpose Gray level co-occurrence matrix, watershed algorithm, present Sobel, Prewitt and Kirsch edge operators are used

Bhadauria H S et al. [24], 2010 worked on comparative analysis of various edge detection techniques before and after the application of morphological filter. A noisy CT image of abnormal lung infected by Honeycombing is used to evaluate the performance of algorithms. The assessment parameter SNR is used for comparison between gradient based edge detectors and morphology based edge detector.

M. N. Nobi et al. [25], 2011 worked on efficient and simple method for noise reduction from medical images. In their proposed method median filter is modified by adding more features. Experimental results are also compared with the other three image filtering algorithms. The quality of the output images is measured by the statistical quantity measures: peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR) and

root mean square error (RMSE). Experimental results of magnetic resonance (MR) image and ultrasound image demonstrate that the proposed algorithm is comparable to popular image smoothing algorithms.

CHAPTER-3
BACKGROUND

3.1 Definition of an Image

The term image refers to a two-dimensional light intensity f(x, y). Where *x* and *y* denote spatial coordinates and the value of f at any point (x, y) is proportional to the brightness (Or gray level) of the image at that point.

3.1.1 Digital Image

The term digital image refers to an image f(x, y) that has been discretized both in spatial coordinates and brightness. A digital image can be considered a matrix whose row and column indices identify a point in the image and the corresponding matrix element value identifies the gray level at that point. The elements of such a digital array are called pels (picture elements), or more commonly pixels. Images are built up of pixels that contain color information and are aligned with the cartesian coordinate system. The zero point is found at the top-left corner of the image (in PostScript, for example, the zero point is found at the bottom-left corner of the page). The image's width is represented by the variable N the image's height with the variable M as shown below.

$$f(x,y) = \begin{pmatrix} f(0,0)f(0,1).....f(0,N-1) \\ f(1,0)f(1,1).....f(1,N-1) \\ . \quad . \quad ..... \quad . \\ f(M-1,0)F(M-1,1)..F(M-1,N-1) \end{pmatrix} \tag{3.1}$$

3.1.2 Image Types

There are four types of images

- Intensity image
- Binary image
- Indexed image
- RGB image

Most of the image processing operations are carried out using binary or intensity images

3.1.2.1 Intensity image

An intensity image is a data matrix whose values have been scaled to represent intensities. When the elements of an intensity image are of class unit8, or class unit16,

they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating point number.

3.1.2.2 Binary image

A binary image is a logical array of 0s and 1s. It's a digital image that has only two possible values for each pixel. Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color. It's an image that consists of only two brightness levels: black and white.

3.1.2.3 Indexed image

An indexed image consists of two matrices: a color map, and an index to the color map. Most images have only small subset of 16 million possible colors. For convenience of storage and file handling, the image has a associated color map, or color palette, which is simply the list of all the colors used in the image. Each pixel has a value that does not gives its color (as for a red-green-blue [RGB] image), but an index to the color in the map.

3.1.2.4 RGB image

The RGB color model is an additive color model in which red, green, and blue light is added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue. In digital image RGB image is an image in which each pixel is specified by three values one each for the red, blue, and green components of the pixel's color. In MATLAB, an RGB image is represented by an m-by-n-by-3 array of class uint8, uint16, or double. If an RGB is of class double the range of values is [0, 1]. Similarly, the range of values is [0,255] or [0, 65535] for RGB of class unit8, or unit16, respectively.

3.2 Noise in an Image

It is generally desirable for image brightness (or film density) to be uniform except where it changes to form an image. There are factors, however, that tend to produce variation in the brightness of a displayed image even when no image detail is present. This variation is usually random and has no particular pattern. In many cases, it reduces image quality and is especially significant when the objects being imaged are

small and have relatively low contrast. This random variation in image brightness is designated noise. This noise can be either image dependent or image independent. All the digital images contain some visual noise. The presence of noise gives an image a mottled, grainy, textured, or snowy appearance. There are three primary types of noise: Random, fixed pattern and banding. Random noise revolves around an increase in intensity of the picture. It occurs through color discrepancies above and below where the intensity changes. It is random, because even if the same settings are used, the noise occurs randomly throughout the image. It is generally affected by exposure length. Random noise is the hardest to get rid of because we cannot predict where it will occur. The digital camera itself cannot account for it, and it has to be lessened in an image editing program. Fixed pattern noise surrounds hot pixels. Hot pixels are pixel bits that are more intense than others surrounding it and are much brighter than random noise fluctuations. Long exposures and high temperatures cause fixed pattern noise to appear. If pictures are taken under the same settings, the hot pixels will occur in the same place and time. Fixed pattern noise is the easiest type to fix after the fact. Once a digital camera realizes the fixed pattern, it can be adjusted to lessen the effects on the image. However, it can be more dubious to the eye than random noise if not lessened Banding noise depends on the camera as not all digital cameras will create it. During the digital processing steps, the digital camera takes the data being produced from the sensor and creates the noise from that. High speeds, shadows and photo brightening will create banding noise. Gaussian noise, salt & pepper noise, passion noise, and speckle noise are some of the examples of noise.

3.2.1 The Effect: Image Noise

Noise in digital images is most visible in uniform surfaces (such as blue skies and shadows) as monochromatic grain, similar to film grain (luminance noise) and/or as colored waves (color noise). As mentioned earlier, noise increases with temperature. It also increases with sensitivity, especially the color noise in digital compact cameras (example D below). Noise also increases as pixel size decreases, which is why digital compact cameras generate much noisier images than digital SLRs. Professional grade cameras with higher quality components and more powerful processors that allow for

more advanced noise removal algorithms display virtually no noise, especially at lower sensitivities.

Noise is typically more visible in the red and blue channels than in the green channel. This is why the unmagnified red channel crops in the examples below are better at illustrating the differences in noise levels.

3.2.2 Sources of Noise in Digital Images

Digital images are prone to a variety of types of noise. Noise is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene. There are several ways that noise can be introduced into an image, depending on how the image is created. For example:

- If the image is scanned from a photograph made on film, the film grain is a source of noise. Noise can also be the result of damage to the film, or be introduced by the scanner itself.
- If the image is acquired directly in a digital format, the mechanism for gathering the data (such as a CCD detector) can introduce noise [15].
- Electronic transmission of image data can introduce noise.

3.2.3 Noise characteristics

Noise not only changes depending on exposure setting and camera model, but it can also vary within an individual image. For digital cameras, darker regions will contain more noise than the brighter regions; with film the inverse is true.

(a) (1) (2) (3) (4)

FIG 3.1 Noise Characteristics (a) Image with different noisy regions, (1) (2) (3) (4) Regions at 100% zoom

Note how noise becomes less pronounced as the tones become brighter. Brighter regions have a stronger signal due to more light, resulting in a higher overall SNR. This means that images which are underexposed will have more visible noise even if you brighten them up to a more natural level afterwards. On the other hand, overexposed images will have less noise and can actually be advantageous, assuming that you can darken them later and that no region has become solid white where there should be texture.

Noise is also composed of two elements: fluctuations in color and luminance. Color or "chroma" noise is usually more unnatural in appearance and can render images unusable if not kept under control.

The relative amount of chroma and luminance noise can vary significantly from one camera model to another. Noise reduction software can be used to selectively reduce both chroma and luminance noise, however complete elimination of luminance noise can result in unnatural or "plasticy" looking images. Noise fluctuations can also vary in both their magnitude and spatial frequency, although spatial frequency is often a neglected characteristic. The term "fine-grained" was used frequently with film to describe noise whose fluctuations occur over short distances, which is the same as having a high spatial frequency. The example below shows how the spatial frequency can change the appearance of noise.

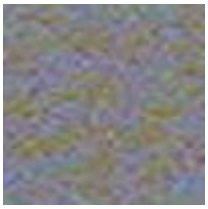
	
Low Frequency Noise (Coarser Texture) Standard Deviation: 11.7	**High Frequency Noise** (Finer Texture) Standard Deviation: 12.5

FIG 3.2: Variation in spatial frequency

If the two patches above were compared based solely on the magnitude of their fluctuations (as is done in most camera reviews), then the patch on the right would seem to have higher noise. Upon visual inspection, the patch on the right actually appears to be

much less noisy than the patch on the left. This is due entirely to the spatial frequency of noise in each patch. Even though noise's spatial frequency is under emphasized, its magnitude still has a very prominent effect. The next example shows two patches which have different standard deviations, but the same spatial frequency.

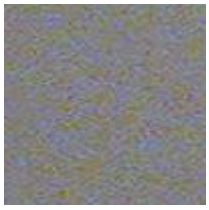
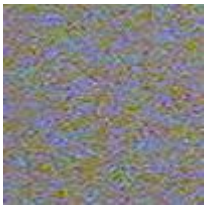
	
Low Magnitude Noise (Smoother Texture) Standard Deviation: 11.7	**High Magnitude Noise** (Rougher Texture) Standard Deviation: 20.8

FIG 3.3: Variation in magnitude

Note how the patch on the left appears much smoother than the patch on the right. High magnitude noise can overpower fine textures such as fabric or foliage, and can be more difficult to remove without over softening the image. The magnitude of noise is usually described based on a statistical measure called the "standard deviation," which quantifies the typical variation a pixel will have from its "true" value. This concept can also be understood by looking at the histogram for each patch:

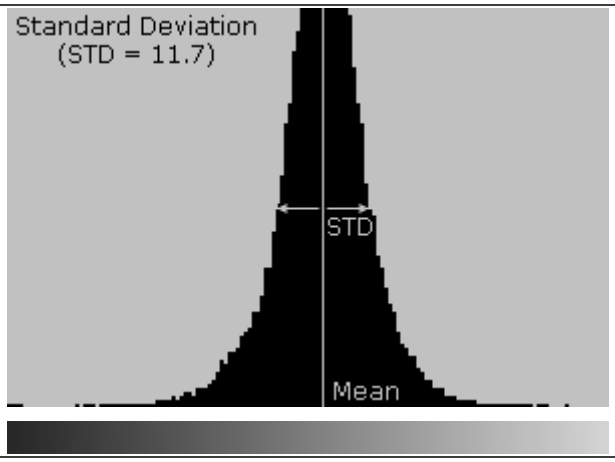
	
	RGB Histogram

FIG 3.4: Low and high magnitude RGB histogram

If each of the above patches had zero noise, all pixels would be in a single line located at the mean. As noise levels increase, the width of this histogram shaped like a bell curve also increases. We present this for the RGB histogram, although the same comparison can also be made for the luminosity (most common measure of overall noise) and individual color histograms.

3.3 Speckle noise

Speckle noise is a granular noise that inherently exists in and degrades the quality of the active radar and synthetic aperture radar (SAR) images [16].

Speckle noise in conventional radar results from random fluctuations in the return signal from an object that is no bigger than a single image-processing element. It increases the mean grey level of a local area.

Speckle noise in SAR is generally more serious, causing difficulties for image interpretation. It is caused by coherent processing of backscattered signals from multiple distributed targets. In SAR oceanography, for example, speckle noise is caused by signals from elementary scatterers, the gravity-capillary ripples, and manifests as a pedestal image, beneath the image of the sea waves.

Speckle is a random, deterministic, interference pattern in an image formed with coherent radiation of a medium containing many sub-resolution scatterers. The texture of the observed speckle pattern does not correspond to underlying structure. The local brightness of the speckle pattern, however, does reflect the local echogenicity of the underlying scatterers. Speckle noise is also known as modal noise which is the noise generated in an optical fiber system by the combination of mode-dependent optical losses and fluctuation in the distribution of optical energy among the guided modes or in the relative phases of the guided modes.

Images are corrupted by speckle noise that affects all coherent imaging systems. Within each resolution cell a number of elementary scatters reflect the incident wave towards the sensor. The backscattered coherent waves with different phases undergo a constructive or a destructive interference in a random manner. The acquired image is thus corrupted by a random granular pattern that delays the interpretation of the image content

and reduces detectability of the features of interest. In medical literature also referred to as “texture”, may present useful diagnostic information. It is therefore advantageous to provide a user interactive denoising method, where the degree of speckle smoothing can be tuned [25].

A speckle image V = $\{v_1,v_2,v_3,.....v_n\}$ is commonly modeled as $v_1 = f_1\vartheta$ where f = $\{ f_1,f_2,f_3.....f_n\}$ is a noise-free image, and $\vartheta = \{\vartheta_1, \vartheta_2, \vartheta_3..... \vartheta_n\}$ is a unit mean random field.

3.4 Basics of Image Segmentation

Segmentation which is a separation of structures having similar attributes from the background and from each other is an essential analysis function in image processing for which numerous algorithms have been developed. Typically, segmentation of an object is achieved either by identifying all pixels or voxels that belong to the object or by locating those that form its boundary. The former is based primarily on the intensity of pixels, but other attributes, such as texture, that can be associated with each pixel, can also be used for segmentation. Techniques that locate boundary pixels use the image gradient, which has high values at the edges of objects [27].

Since classification of pixels is required in image segmentation, it is often treated as a pattern recognition problem and addressed with related techniques. The principle objective of the segmentation process is to divide an image into portions that are homogeneous with respect to one or more characteristics or features. After segmentation process has occurred the segmented portions have similar attributes.

Segmentation is an important tool in image processing and it has been useful in many applications. In digital image processing, segmentation is important for feature extraction, image measurements, and image display. Image segmentation has applications separate from computer vision; it is frequently used to aid in isolating or removing specific portions of an image.

A wide variety of segmentation techniques has been proposed. However, there is no one standard segmentation technique that can produce satisfactory results for all imaging applications. The definition of the goal of segmentation varies according to the goal of

the study and the type of the image data. Segmentation techniques can be divided into classes in many ways, depending on classification scheme:

- Manual, semiautomatic, and automatic.
- Pixel-based (local methods) and region-based (global methods).
- Manual delineation, low-level segmentation (thresholding, region growing, etc), and model-based segmentation (multispectral or feature map techniques, dynamic programming, contour following, etc.)
- Classical (thresholding, edge-based, and region-based techniques)

3.4.1 Classification of Segmentation

The most commonly used segmentation techniques can be classified into two broad categories:

- Region segmentation techniques that find the regions satisfying a given homogeneity criterion, and
- Edge-based segmentation techniques that look for edges between regions with different characteristics.

Thresholding is a common region segmentation method. In this technique a threshold is selected and an image is divided into groups of pixels having values less than the threshold and groups of pixels with values greater or equal to the threshold. There are several thresholding methods: global methods based on gray-level histograms, global methods based on local properties, local threshold selection, and dynamic thresholding. Clustering algorithms achieve region segmentation by partitioning the image into sets or clusters of pixels that have strong similarity in the feature space.

The basic operation is to examine each pixel and assign it to the cluster that best represents the value of its characteristic vector of features of interest.

Region growing is another class of region segmentation algorithms that assign adjacent pixels or regions to the same segment if their image values are close enough, according to some pre-selected criterion of closeness.

The objective of edge-based segmentation algorithms is to find object boundaries and segment the portion enclosed by that boundaries in the image. These algorithms usually operate on edge magnitude and/or phase images produced by an edge operator suited

according to the characteristics of the image. For example, most gradient operators such as Prewitt, Kirsch, or Roberts's operators are based on the existence of an ideal step edge.

3.4.1.1 Region Based Segmentation Techniques

Region based segmentation technique have two major methods. These are thresholding and region growing.

3.4.1.1.1Thresholding

Several thresholding techniques have been developed. Some of them are based on the image histogram; others are based on local properties, such as local mean value and standard deviation, or the local gradient. Global thresholding is the most intuitive approach. In global thresholding only one threshold is selected for the entire image, based on the image histogram. If the threshold depends on local properties of some image regions, for example local average gray value, thresholding is called local. Thresholding is called dynamic or adaptive if the local thresholds are selected independently for each pixel (or groups of pixels). The fig 3.5 shows the basic operation of thresholding [19] on image g(x,y)

$$g(x,y) = \begin{cases} 1 & \text{if } (x,y) > T \\ 0 & \text{if } (x,y) \le T \end{cases} \tag{3.2}$$

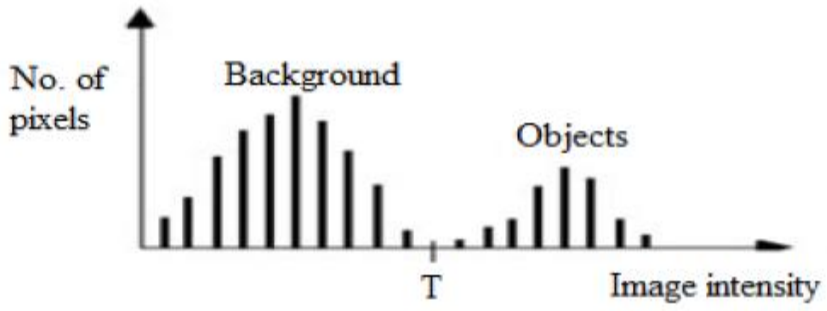


Figure 3.5 Thresholding Process

3.4.1.1.2 Region Growing

As far thresholding methods focus on the difference of pixel intensities whereas the region growing method takes groups of pixels with similar intensities. Region growing, which is also named as region merging starts with a pixel or a group of pixels (called seeds) that belong to the structure of interest being segmented.

Generally an operator chooses the seeds, or it can also provided by an automatic seed finding procedure. In the next step neighboring pixels are examined one at a time and added to the growing region, if they are sufficiently similar based on a uniformity test, (also called a homogeneity criterion). The procedure continues until no more pixels can be added. The object is then represented by all pixels that have been accepted during the growing procedure. One example of the uniformity test is comparing the difference between the pixel intensity value and the mean intensity value over a region. If the difference is less than a predefined value, for example, two standard deviations of the intensity across the region, the pixel is included in the region; otherwise, it is defined as an edge pixel. The results of region growing depend strongly on the selection of the homogeneity criterion. If it is not properly chosen, the regions leak out into adjoining areas or merge with regions that do not belong to the object of interest. Another problem of region growing is that different starting points may not grow into identical regions [6].

The advantage of region growing is that it is capable of correctly segmenting regions that have the same properties and are spatially separated. Another advantage is that it generates connected regions.

3.4.1.2 Edge-Based Segmentation Techniques

Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrasts a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image.

There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories:

- Gradient methods which detects the edges by looking for the maximum and minimum in the first derivative of the image.
- Laplacian methods which search for zero crossings in the second derivative of the image to find edges.

An edge or boundary on an image is defined by the local pixel intensity gradient. A gradient is an approximation of the first order derivative of the image function. For a given image f (x, y), we can calculate the magnitude of the gradient as

$$|G| = \sqrt{G_x^{\,2} + G_y^{\,2}} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \tag{3.3}$$

And the direction of gradient as

$$D = \tan^{-1}\left(\frac{G_Y}{G_X}\right) \tag{3.4}$$

Where Gx and Gy are gradients in directions x and y, respectively [24]. Since the discrete nature of digital image does not allow the direct application of continuous differentiation, calculation of the gradient is done by differencing.

Both magnitude and direction of the gradient can be displayed as images. The magnitude image will have gray levels that are proportional to the magnitude of the local intensity changes, while the direction image will have gray levels representing the direction of maximum local gradient in the original image.

Most gradient operators in digital images involve calculation of convolutions, e.g., weighted summations of the pixel intensities in local neighborhoods. The weights can be listed as a numerical array in a form corresponding to the local image neighborhood (also known as a mask, window or kernel). For example, in case of a Sobel edge operator, there are two masks:

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \qquad \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \tag{3.5}$$

The first mask is used to compute Gx while the second is used to compute Gy. The gradient magnitude image is generated by combining Gx and Gy and |G| is obtained by using respective equation. The results of edge detection depend on the gradient mask. Some of the other edge operators are Roberts, Prewitt, Robinson, and Kirsch [23]. Many edge detection methods use a gradient operator, followed by a threshold operation on the gradient, in order to decide whether an edge has been found. As a result, the output is a binary image indicating where the edges are.

The edge-based techniques are computationally fast and do not require a prior information about image content. The common problem of edge-based segmentation is that often the edges do not enclose the object completely. To form closed boundaries surrounding regions, a post processing step of linking or grouping edges that correspond

to a single boundary is required. The simplest approach to edge linking involves examining pixels in a small neighborhood of the edge pixel and linking pixels with similar edge magnitude and/or edge direction. In general, edge linking is computationally expensive and not very reliable.

3.5 Speckle Reduction Filters

Most of the techniques for speckle reduction filtering in the literature use linear filtering based on adaptive filtering based on computation of local statistics and anisotropic filtering. The common filters used are LEE filter, First order local statistical filter (Lsmv) and speckle reducing anisotropic diffusion (SRAD) filter.

3.5.1 LEE Filter

The Lee filter is a standard deviation based filter that calculates the new pixel values with statistics computed within individual filter windows [21]. In Lee filter, the statistical distribution of the values of the pixels within the moving kernel is utilized to estimate the value of the pixel of interest. This is based on the assumption that the mean and variance of the pixel of interest are equal to the local mean and local variance of all pixels within the user-selected moving kernel. The resulting grey level value R for the smoothed pixel is:

$$R = I_C \times W + I_M \times (1 - W) \tag{3.6}$$

Where

$$W = (1 - C_u^2 / C_i^2) \tag{3.7}$$

$$C_u = \sqrt{1 / ENL} \tag{3.8}$$

$$C_i = S / I_m \tag{3.9}$$

I_c = central pixel of filter window

I_m = mean intensity within filter window

S = Standard deviation of intensity within filter window

The weighting function W is a measure of the estimated noise variation coefficient Cu with respect to the image variation coefficient C_i. The number of looks parameter

ENL is the effective number of looks of the radar image. ENL is used to estimate the noise variance and it controls the amount of smoothing applied to the image by the filter.

Moreover, ENL should be close to the actual number of looks, but it may be changed if the image has undergone resembling. The user may experimentally adjust the ENL value to control the effect of the filter. A small ENL value leads to more smoothing while a large ENL preserves more image features.

3.5.2 First Order Local Statistical Filter (lsmv)

The filter first order statistics uses the variance and the mean of the neighborhood. Hence, the algorithms in this class may be traced back to the following equation

$$f_{i,j} = \bar{g} + k_{i,j}(g_{i,j} - \bar{g}), \tag{3.10}$$

where $f_{i\ j}$ is the estimated noise-free pixel value, $g_{i,j}$ is the noisy pixel value in the moving window, $\bar{g}$ is the local mean value of an *N*1*xN*2 region surrounding and including pixel $g_{i,j}$, $k_{i,j}$ is a weighting factor with $k \in [0..1]$, and i, j, are the pixel coordinates. The factor $k_{i,j}$ is a function of the local statistics in a moving window. It can be found in the literature and may be derived in different form.

$$k_{i,j} = (1 - \bar{g}^2\sigma^2) / (\sigma^2(1 + \sigma_n^2)), \tag{3.11}$$

$$k_{i,j} = \sigma^2 / (\bar{g}^2\sigma^2 + \sigma^2), \tag{3.12}$$

$$k_{i,j} = (\sigma^2 - \sigma_n^2) / \sigma^2. \tag{3.13}$$

The values σ^2 and σ_n^2 represent the variance in the moving window and the variance of noise in the whole image, respectively. The noise variance may be calculated for the logarithmically compressed image by computing the average noise variance over a number of windows with dimensions considerably larger than the filtering window. In each window the noise variance is computed as:

$$\sigma_n^2 = \sum_{i=1}^{p} \sigma_p^2 / \bar{g}_p, \tag{3.14}$$

where σ_p^2 and $\overline{g_p}$ are the variance and mean of the noise in the selected windows, respectively, and p is the index covering all windows in the whole image If the value of $k_{i,j}$ is 1 (in edge areas) this will result to an unchanged pixel, and a value of 0 (in uniform

areas) replaces the actual pixel by the local average, *g*, over a small region of interest. The despeckle filters lsmv have the moving window size as 5×5 [26].

3.5.3 Speckle reducing anisotropic diffusion (SRAD) filter

Casting the probability density function (PDE) approach and adaptive filtering approach Yongjian Yu et.al [4] developed a new model for speckle reduction called as speckle reducing anisotropic diffusion (SRAD) method. SRAD filter can be seen as a mixture of the classical anisotropic diffusion filter and the adaptive speckle Lee filter. The SRAD filter better preserves and enhances edges while efficiently removing speckle in homogeneous regions. The SRAD anisotropic diffusion filter for smoothing a given image can be stated according to the following nonlinear partial differential equation.

For an intensity image $I_0(x, y)$ which is having finite power and no zero values over the image support Ω, the output image $I(x, y; 0)$ is given by the following PDE.

$$\begin{cases} \partial I(x,y;t)/\partial t = \operatorname{div}[c(q)\nabla(x,y;t)] \\ I(x,y;0) = I_0(x,y), (\partial I(x,y;t)/\partial \vec{n}\,|_{\partial\Omega} = 0 \end{cases} \tag{3.15}$$

Where $\partial\Omega$ denotes the border of Ω, $\vec{n}$ is the outer normal to $\partial\Omega$, and the diffusion coefficient c(q) is expressed as

$$c(q) = \frac{1}{1+[q^2(x,y;t) - q_0^2(t)(1+q_0^2(t))]} \tag{3.16}$$

Or

$$c(q) = \exp\{-[q^2(x,y;t) - q_0^2(t)]/[q_0^2(t)(1+q_0^2(t))]\} \tag{3.17}$$

With q_0 (t) standing for the speckle scale function. This coefficient is related to the local statistic of the image (mean and intensity variance over a homogeneous area at each t instant), but to do automatic image processing, it can be approximated by q_0 (t) ≈ q_0 exp[−ρt], where ρ and q_0 are two positive parameters less than or equal to one. q(x, y; t) is called the instantaneous coefficient of variation, and it is calculated from the image pixel intensity I, normalized gradient magnitudes $\|\nabla I\|/I$, and the normalized Laplacian $\nabla^2 I/I$ as:

$$q(x,y;t) = \sqrt{\frac{(1/2)(|\nabla I|/I)^2 - (1/4^2)(\nabla^2 I/I)^2}{[1+(1/4)(\nabla^2 I/I)]^2}} \tag{3.18}$$

The above mentioned function, q which is the instantaneous coefficient of variation q(x, y; t) helps in detecting the edges present in the images corrupted by speckle. At edges and high contrast regions this function produces high values and at homogenous regions it gives low values, $q_0(t)$ is the speckle scale function and q(x, y; t) fluctuates around $q_0(t)$.

To detect an image edge (or a boundary), the gradient magnitude is used by noting that, in (2), with $c(q) \rightarrow 1$ and $q \rightarrow q0$, the filter performs an isotropic diffusion, i.e., Gaussian filtering. For the case $c(q) \rightarrow 0$ and $q \rightarrow \infty$, the diffusion stops at the edges. As the filter proceeds, some instability may appear (when the denominator in (2) approaches zero much faster than the numerator). Fortunately, this can be easily avoided by simple control routines.

3.6 Estimation of Statistical Parameters

The parameters which are used for performance evaluation are Signal to Noise Ratio (SNR), Root Mean Square Error (RMSE), Root Mean Square Error (RMSE), Correlation Coefficient (COC), Mean Structural Similarity Index Measure (MSSIM) and Edge Preserving Index (EPI).

3.6.1 Estimation of SNR

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

$$\mathrm{SNR} = 10\log\left(\frac{\sigma_g^2}{\sigma_e^2}\right) \tag{3.19}$$

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 denoised image). Brighter regions have a stronger signal due to more light, resulting in higher overall SNR [17]

3.6.2 Estimation of RMSE

Mean square error (MSE) is given by [18]

$$\mathrm{MSE} = \sum_{i=j=1}^{N} [f(i, j) - F(i, j)]^2 / N^2 \tag{3.20}$$

$$RMSE = \sqrt{MSE} \tag{3.21}$$

Where, f is the original image F is the filtered image and N is the size of image.

RMSE is an estimator in many ways to quantify the amount by which an filtered/noisy image differs from noiseless image.

3.6.3 Estimation of COC

Correlation indicates the strength and direction of linear relationship between two signals and its value lie between +1 to -1. The correlation is 1 in the case of an increasing linear relationship, -1 in the case of a decreasing linear relationship, and some value in between for all the other cases, including the degree of linear dependence between the two signals. The closer the coefficient is to either -1 or +1, the stronger the correlation between the signals.

$$Coc = \frac{\sum (g - \bar{g})(\hat{g} - \bar{g})}{\sqrt{\sum (g - \bar{g})^2 \sum (\hat{g} - \bar{\hat{g}})^2}} \tag{3.22}$$

Where, g and $\hat{g}$ are original and denoised images respectively and $\bar{g}$ and $\bar{\hat{g}}$ are the mean's of the original image and denoised image respectively [17].

3.6.4 Estimation of MSSIM

MSSIM index is a novel method for measuring the similarity between two images. MSSIM values exhibit much better consistency with the qualitative appearance of the image. MSSIM is given by

$$MSSIM(X, Y) = \frac{1}{M} \sum_{j=1}^{M} SSIM(x_j, y_j) \tag{3.23}$$

$$SSIM(x, y) = \frac{(2\upsilon_x \upsilon_y + C_1)(2\sigma_{xy} + C_2)}{(\upsilon_x^2 + \upsilon_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.24}$$

$$\upsilon_x = \sum_{i=1}^{N} w_i x_i \tag{3.25}$$

$$\sigma_x = \left(\sum_{i=1}^{N} w_i (x_i - \upsilon_x)^2\right)^{\frac{1}{2}} \tag{3.26}$$

$$\sigma_{xy} = \left(\sum_{i=1}^{N} w_i (x_i - \upsilon_x)(y_i - \upsilon_y)\right) \tag{3.27}$$

$$C_1 = (K_1 L)^2 \tag{3.28}$$

$$C_2 = (K_2 L)^2 \tag{3.29}$$

Where, L is the dynamic range of pixel values (255 for 8-bit grayscale images). And K_1 << is a small constant and also K_2 <<1 [13].

Speckle is significant in Ultrasound and Synthetic Aperture Radar (SAR) imaging. Before the implementation of speckle reduction techniques it is necessary to understand the basics of these imaging modalities.

3.6.5 Estimation of EPI

It means edge preservation capacity of image in horizontal and perpendicular direction after treated by different kinds of filter. The higher the EPI value, stronger is the edge preservation capacity.

$$\mathrm{EPI} = \frac{\sum (\Delta I - \overline{\Delta I})(\Delta F - \overline{\Delta F})}{\sqrt{\sum (\Delta I - \overline{\Delta I})^2 \sum (\Delta F - \overline{\Delta F})^2}} \tag{3.40}$$

Where ΔI and ΔF are the high pass filtered versions of the images I and F, obtained with a 3×3 pixel standard approximation of the Laplacian operator. The value of EPI ranges from [0 1] and the largest value of EPI means more ability to preserve edges [17].

CHAPTER-4
MORPHOLOGY

4.1 Mathematical Morphology

For the purposes of image analysis and pattern recognition there is always a need to transform an image into another better represented form. During the past five decades image-processing techniques have been developed tremendously and mathematical morphology in particular has been continuously developing because it is receiving a great deal of attention because it provides a quantitative description of geometric structure and shape and also a mathematical description of algebra, topology, probability, and integral geometry [11]. Mathematical morphology is extremely useful in many image processing and analysis applications. Mathematical morphology denotes a branch of biology that deals with the forms and structures of animals and plants. It analyzes the shapes and forms of objects. In computer vision, it is used as a tool to extract image components that are useful in the representation and description of object shape. It is mathematical in the sense that the analysis is based on set theory, topology, lattice algebra, function, and so on [5]. Another use of mathematical morphology is to filter image. It is a well know non-linear filter for image enhancement [2][3][10].

4.2 Basic Concepts from Set Theory

Let *Z* be a set of integers. The sampling process used to produce the digital image may be viewed as a partitioning in the x y -plane in to a grid, with coordinates of the centre of each grid being a pair of elements from the Cartesian product Z_2. In the terminology of set theory the function f (x, y) is said to be a digital image if (*x*, y) are integers from Z_2 and f is a mapping that assigns an intensity value in the set of real numbers, *R*, to each distinct pair of coordinates (x ,y). If the elements of *R* also are integers, a digital image then becomes a two-dimensional function, whose coordinates and the amplitude values are integers [28].

Let *A* be a set in Z_2, the elements of which are pixel coordinates (*x*, y). If *w* = (x, y) is an element of A, then it is written

$$w \in A \tag{4.1}$$

Similarly, if w is not an element of A, it is written

$$w \notin A \tag{4.2}$$

A set of B pixels satisfying a particular condition is written as

$$B = \{w \mid \text{condition}\} \tag{4.3}$$

For example, the set of all pixels coordinates that do not belong to A, denoted by A_c, and is given by **complement** of A

$$A_c = \{w \mid w / A\} \tag{4.4}$$

The **union** of two sets, denoted by

$$C = A \cup B \tag{4.5}$$

Where the set of all elements belong to either A or B or both.

Similarly, the **intersection** of two sets A and B is the set of elements that belong to both sets, denoted by

$$C = A \cap B \tag{4.6}$$

The **difference** of sets A and B, denoted by $A - B$, is the set of all elements that belong to A but not to B:

$$A - B = \{w \mid w \in A, w / B\} \tag{4.7}$$

Figure 4.1 below illustrates these set operations.

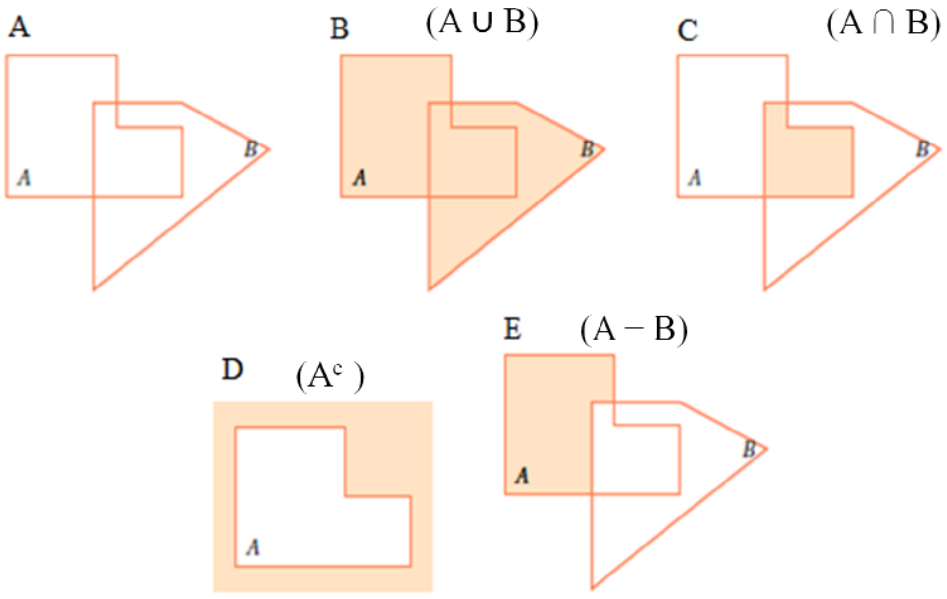


Fig-4.1 Basic concepts from set theory

In addition to the basic operations, morphological operations often require two operations that are specific to sets whose elements are pixel coordinates. The **reflection** of set B, denoted by $B^{\wedge}$, is defined as

$$B^{\wedge} = \{w | w = -b, \text{ for } b \in B\} \tag{4.8}$$

The **translation** of set A by z = (z x, z y), denoted by $(A)_z$, is defined as

$$(A)_z = \{c | c = a + z, \text{ for } a \in A\} \tag{4.9}$$

The union and intersection can be exchanged by logical duality (De-Morgan's Law) as

$$(A \cup B)^C = A^C \cup B^C \tag{4.10}$$

And

$$(A \cup B)^C = A^C \cap B^C \tag{4.11}$$

4.3 Structuring Element

The basic idea in binary morphology is to probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image. This simple "probe" is called structuring element, which is also a binary image (i.e., a subset of the space or grid). A structuring element is a small image used as a moving window whose support delineates pixel neighborhoods in the image plane. Mathematical morphology involves geometric analysis of shapes and textures in images. An image can be represented by a set of pixels [7].

Morphological operators work with two images. The image being processed is referred to as the active image, and the other image, being a kernel, is referred to as the structuring element. Each structuring element has a designed shape, which can be thought of as a probe or a filter of the active image. The active image can be modified by probing it with the structuring elements of different sizes and shapes.

Mathematical morphology use morphological structuring elements in order to measure and distill corresponding shape of an image to attain objective of analysis, to reduce image data and to keep basic shape character.

4.4 Binary Morphological

The theoretical foundation of binary morphology is set theory [8]. In binary images, the points in the set are called foreground and those in the complement are called the

background.

Besides dealing with the usual set-theoretic operations of union and intersection, morphology depends extensively on the translation operation. For convenience, $\cup$ denotes the set-union, $\cap$ denotes set-intersection and + inside the set notation refers to vector addition in the following equations

Given an image A, the translation of A by the point x, denoted by Ax, is defined by

$$A + x = \{a + x: a \in A\} \tag{4.12}$$

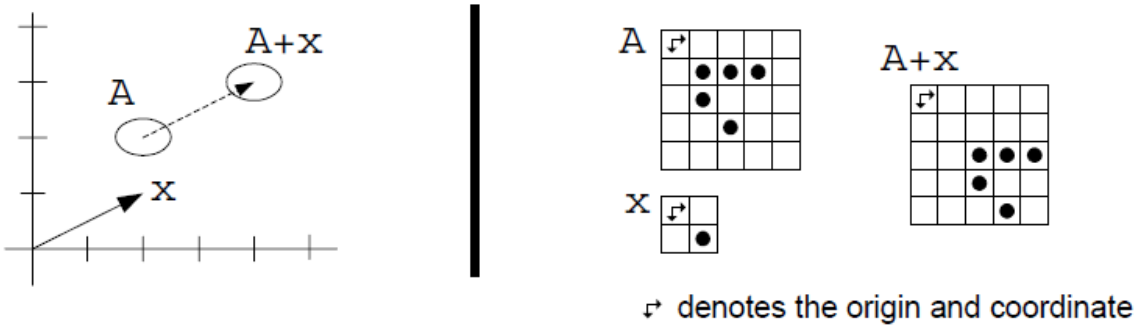


Fig 4.2 Translation operation on Euclidean (left) and digital (right) setting

Figure 4.2 is an example of the translation operation. In the left part, A is an ellipse centered at (2, 2) and x is a vector of (2, 1). Translation of A by x will shift A with the displacement (2, 1) such that A is centered at (4, 3). In the right part, A is an image in Z^2, with f (1; 1); (2; 1); (3; 1); (1; 2); (3; 4) g and x is a vector of (1, 1). Translation of A by x results in {(2; 2); (3; 2); (4; 2); (2; 3); (4; 5)}.

4.4.1 Dilation

Dilation is an operation that grows or thickens objects in a binary image. The specific manner and the extent of this thickening are controlled by structuring element. Dilation is a process that translates the origin of the structuring element throughout the domain of the image and checks to see whether it overlaps with 1-valued pixels. The output image is 1 at each location of the origin of the structuring element if the structuring element overlaps at least one 1-valued pixel in the input image. Mathematically, dilation is defined in terms of set operations. The dilation of A by B, denoted by A $\oplus$ B, is defined as [14]

$$A \oplus B = \{z| B\hat{}z \cap A \neq \Phi\} \quad (4.13)$$

Where Φ is the empty set and B is the structuring element. In words, the dilation of A by B is the set consisting of all the structuring element origin locations where the reflected B overlaps at least some portion of A.

Dilation is commutative, i.e., $A \oplus B = B \oplus A$. It is a convention in image processing to let the first operand of $A \oplus B$ be the image and the second operand is the structuring element, which is usually much smaller than the image. The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus areas of foreground pixels grow in size while holes within those regions become smaller.

- **Working of dilation**

The dilation operator takes two pieces of data as inputs. The first is the image which is to be dilated. The second is a (usually small) set of coordinate points known as a structuring element (also known as a kernel). It is this structuring element that determines the precise effect of the dilation on the input image.

As an example of binary dilation, suppose that the structuring element is a 3×3 square, with the origin at its centre, as shown in matrix. In this matrix, foreground pixels are represented by 1's.

$$\begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \end{pmatrix} \quad (4.14)$$

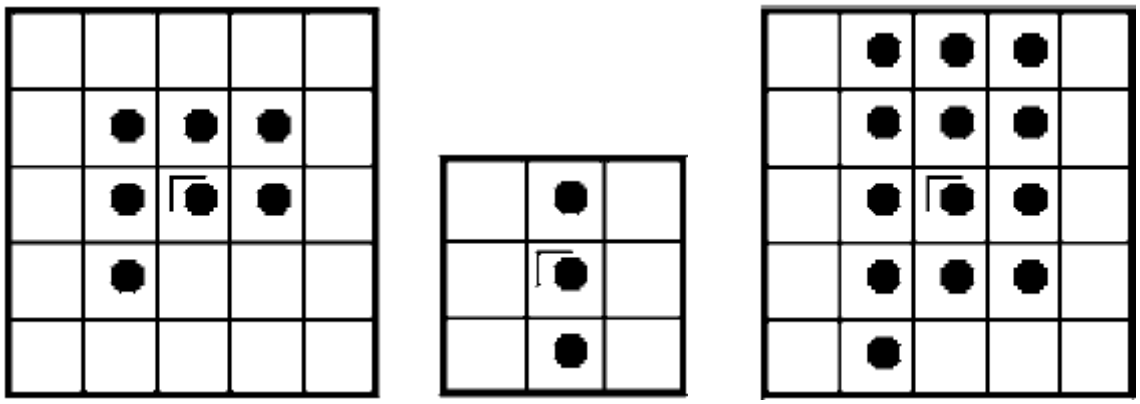
Set of coordinate points-{(-1,-1) (0,-1) (1,-1) (-1,0) (0,0) (1,0) (-1,1) (0,1) (1,1)}.

To compute the dilation of a binary input image by this structuring element, the structuring element superimposed on top of the input image for each foreground pixel so that the origin of the structuring element coincides with the input pixel position. If at least one pixel in the structuring element coincides with a foreground pixel in the image underneath, then the input pixel is set to the foreground value. If all the corresponding pixels in the image are background, however, the input pixel is left at the background value.

Most implementations of this operator expect the input image to be binary, usually with foreground pixels at pixel value 255, and background pixels at pixel value 0.

Such an image can often be produced from a gray scale image using thresholding. The structuring element may have to be supplied as a small binary image, or in a special matrix format, or it may simply be hardwired into the implementation, and not require specifying at all.

The effect of a dilation using this structuring element on a binary image is shown in Figure 4.3 below.



Input Binary Image Structuring Element Dilated Image

Fig-4.3 Process of dilation on binary image

The 3×3 square is probably the most common structuring element used in dilation operations, but others can be used. A larger structuring element produces a more extreme dilation effect, although usually very similar effects can be achieved by repeated dilations using a smaller but similarly shaped structuring element. With larger structuring elements, it is quite common to use an approximately disk shaped structuring element, as opposed to a square one.

4.4.2 Erosion

Erosion is the morphological dual to dilation [22]. Erosion shrinks or thins in a binary image. As in dilation, the manner and extent of shrinking is controlled by the structuring element. The output of erosion has a value 1 at each location of the origin of the structuring element, such that the structuring element overlaps only 1-valued pixels of the input image.

The mathematical definition of erosion is similar to that of dilation. The erosion of A by B, denoted by $A \ominus B$, is defined as

$$A \ominus B = \{z | Bz \cap Ac \neq \Phi\} \quad (4.15)$$

In other words, erosion of *A* by *B* is the set of structuring element origin locations where the translated B has no overlap with the background of A. It is typically applied to binary images, but there are versions that work on gray scale images. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus areas of foreground pixels shrink in size, and holes within those areas become larger.

- **Working of erosion**

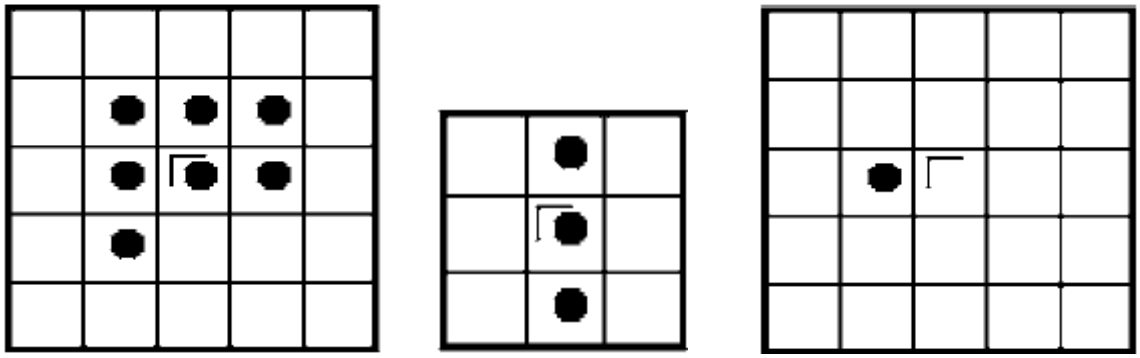
The erosion operator takes two pieces of data as inputs. The first is the image which is to be eroded. The second is a set of coordinate points known as a structuring element. It is this structuring element that determines the precise effect of the erosion on the input image. As an example of binary erosion, suppose that the structuring element is a 3×3 square, with the origin at its center as shown below. 1's represent foreground pixels here.

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{4.16}$$

Set of coordinate points-{(-1,-1) (0,-1) (1,-1) (-1,0) (0,0) (1,0) (-1,1) (0,1) (1,1)}.

To compute the erosion of a binary input image by this structuring element, the structuring element superimposed on top of the input image for each foreground pixel so that the origin of the structuring element coincides with the input pixel coordinates.

If for every pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is. If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value. Erosion is the dual of dilation, i.e. eroding foreground pixels is equivalent to dilating the background pixels. Most implementations of this operator will expect the input image to be binary, usually with foreground pixels at intensity value 255, and background pixels at intensity value 0. The effect of erosion using this type of structuring element on a binary image is shown in Figure 4.4 below.



Input Binary Image Structuring Element Eroded Image

Fig-4.4 Process of erosion in binary image

The 3×3 square is probably the most common structuring element used in erosion operations, but others can be used. A larger structuring element produces a more extreme erosion effect, although usually very similar effects can be achieved by repeated erosions using a smaller similarly shaped structuring element. With larger structuring elements, it is quite common to use an approximately disk shaped structuring element, as opposed to a square one.

Erosions can be made directional by using less symmetrical structuring elements. For example, a structuring element that is 10 pixels wide and 1 pixel high will erode in a horizontal direction only. Similarly, a 3×3 square structuring element with the origin in the middle of the top row rather than the center will erode the bottom of a region more severely than the top.

4.4.3 Opening

The morphological opening of A by B, denoted by A *o* B, is simply erosion of A by B followed by the dilation of the result by *B*.

$$A \; o \; B = (A \ominus B) \oplus B. \tag{4.17}$$

An alternative mathematical formulation of opening is

$$A \; o \; B = \cup \{(B_Z) | \; (B_Z) \text{ is a subset of A}\}. \tag{4.18}$$

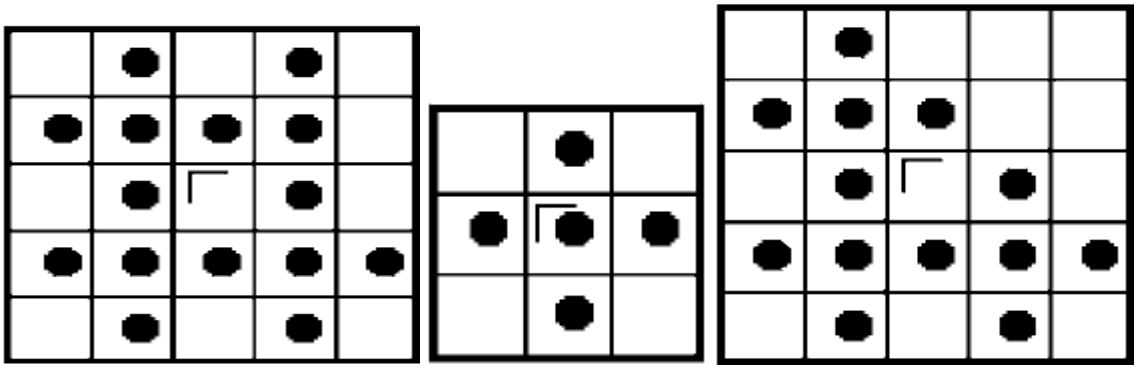
Where $\cup \{.\}$ denotes the union of all sets inside braces, and the notation ($C \subseteq D$) means that C is a subset of D. This formulation has a simple geometric interpretation: A *o* B is the union of all translations of B that fit entirely within A. Morphological opening removes completely regions of an object that cannot contain the structuring element,

smoothes the object contours, breaks thin connections, and removes the protrusions. Opening and closing are two important operators from mathematical morphology. They are both derived from the fundamental operations of erosion and dilation. Like those operators they are normally applied to binary images, although there are also gray level versions. The basic effect of an opening is somewhat like erosion in that it tends to remove some of the foreground (bright) pixels from the edges of regions of foreground pixels. However it is less destructive than erosion in general. As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve foreground regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of foreground pixels.

- **Working of opening**

An opening is defined as erosion followed by a dilation using the same structuring element for both operations. The opening operator requires two inputs: an image to be opened, and a structuring element. Opening is the dual of closing, i.e. opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same element.

While erosion can be used to eliminate small clumps of undesirable foreground pixels, quite effectively, it has the big disadvantage that it will affect all regions of foreground pixels indiscriminately. Opening gets around this by performing both erosion and a dilation on the image. The effect of opening can be quite easily visualized. Imagine taking the structuring element and sliding it around inside each foreground region, without changing its orientation. All pixels which can be covered by the structuring element with the structuring element being entirely within the foreground region will be preserved. However, all foreground pixels which cannot be reached by the structuring element without parts of it moving out of the foreground region will be eroded away. After the opening has been carried out, the new boundaries of foreground regions will all be such that the structuring element fits inside them, and so further openings with the same element have no effect. The property is known as idempotence. The effect of an opening on a binary image using a 3×3 square structuring element is illustrated in Figure 4.5 below.



Input Binary Image Structuring Element Opening

Fig-4.5 Process of opening in binary image

As with erosion and dilation, it is very common to use this 3×3 structuring element. The effect in the above figure is rather subtle since the structuring element is quite compact and so it fits into the foreground boundaries quite well even before the opening operation. To increase the effect, multiple erosions are often performed with this element followed by the same number of dilations. This effectively performs an opening with a larger square structuring element.

4.4.4 Closing

The morphological closing of A by B, denoted by $A \bullet B$, is a dilation followed by erosion:

$$A \bullet B = (A \oplus B) \ominus B. \tag{4.19}$$

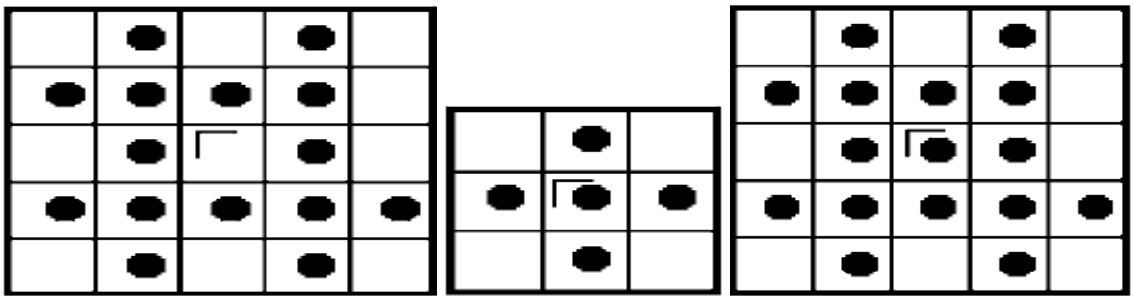
Geometrically, $A \bullet B$ is the complement of the union of all translations of B they do not overlap A. Like opening, morphological closing smoothes the contours of the objects. Unlike opening, however, it generally joins narrow breaks, filling long thin gulfs, and fills holes smaller than the structuring element.

Closing is an important operator in the field of mathematical morphology. Like its dual operator opening, it can be derived from the fundamental operations of erosion and dilation. Like those operators it is normally applied to binary images, although there are gray level versions. Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of foreground (bright) regions in an image (and shrink background color holes in such regions), but it is less destructive of the original boundary shape. As with other morphological operators, the exact operation is determined by a structuring

element. The effect of the operator is to preserve background regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels.

- **Working of closing**

Closing is opening performed in reverse. It is defined simply as dilation followed by erosion using the same structuring element for both operations. The closing operator requires two inputs: an image to be closed and a structuring element. Gray level closing consists straightforwardly of a gray level dilation followed by gray level erosion. Closing is the dual of opening, i.e. closing the foreground pixels with a particular structuring element, is equivalent to closing the background with the same element. One of the uses of dilation is to fill in small background color holes in images. One of the problems with doing this, however, is that the dilation will also distort all regions of pixels indiscriminately. By performing erosion on the image after the dilation, i.e. a closing, some of this effect is reduced. The effect of closing can be quite easily visualized. Imagine taking the structuring element and sliding it around outside each foreground region, without changing its orientation. For any background boundary point, if the structuring element can be made to touch that point, without any part of the element being inside a foreground region, then that point remains background. If this is not possible, then the pixel is set to foreground. After the closing has been carried out the background region will be such that the structuring element can be made to cover any point in the background without any part of it also covering foreground point, and so further closings will have no effect. This property is known as idempotence. The effect of a closing on a binary image using a 3×3 square structuring element is illustrated in Figure 4.6 below.



Input Binary Image Structuring Element Closing

Fig-4.6 Process of closing in binary image

As with erosion and dilation, this particular 3×3 structuring element is the most commonly used, and in fact many implementations will have it hardwired into their code, in which case it is obviously not necessary to specify a separate structuring element. To achieve the effect of a closing with a larger structuring element, it is possible to perform multiple dilations followed by the same number of erosions. Closing can sometimes be used to selectively fill in particular background regions of n image. Whether or not this can be done depends upon whether a suitable structuring element can be found that fits well inside regions that are to be preserved, but doesn't fit inside regions that are to be removed.

4.5 Gray-scale Morphology

Grayscale morphology is a multidimensional generalization of the binary operations. Binary morphology is defined in terms of set-inclusion of pixel sets. So is the grayscale case, but the pixel sets are of higher dimension. In grayscale morphology, images are functions mapped on an Euclidean space or grid E into $\mathrm{R} = (\infty, -\infty)$, where R is the set of real number, ∞ is an element larger than any real number, and $-\infty$ is an element smaller than any real number. In binary morphology, we start with dilation and erosion, which for gray-scale morphology is defined in terms of minima and maxima of pixel neighborhoods [12].

4.5.1Dilation

The gray-scale dilation of f by structuring element b, denoted $f \oplus b$, is defined as:

$$(f \oplus b)(x, y) = \max\{f(x - x', y - y') + b(x', y') \mid (x', y') \in D_b\} \qquad (4.20)$$

Where D_b the domain of b and f (x, y) is is assumed to equal $-\infty$ outside the domain of f .This equation implements a process similar to the concept of spatial convolution [20] Conceptually we can think of rotating the structuring element about its origin and translating it to all locations in the image, just as the convolution kernel is rotated and translated about the image. At each translated location, the rotated structuring element values are added to the image pixel values and the maximum is computed.

- **Working of dilation**

One important difference between convolution and gray-scale dilation is that, in the latter, D_b , a binary matrix, defines which locations in the neighborhood are included in the max operation. In other words, for an arbitrary pair of coordinates (x_o, y_o) in the domain of D_b, the sum $f(x - x_o, y - y_o) + b(x_o, y_o)$ is included in the max computation only if D_b is 1 at those coordinates. If D_b is 0 at (x_o, y_o), the sum is not considered in the max operation. This is repeated for all coordinates $(x', y') \in D_b$ each time that coordinates (x, y) change. Plotting $b(x', y')$ as a function of coordinates x' and y' would look like a digital "surface" with the height at any pair of coordinates being given by the value of b at those coordinates.

In practice, gray-scale dilation usually is performed using flat structuring element in which the value (height) of b is 0 at all coordinates over which D_b is defined. That is,

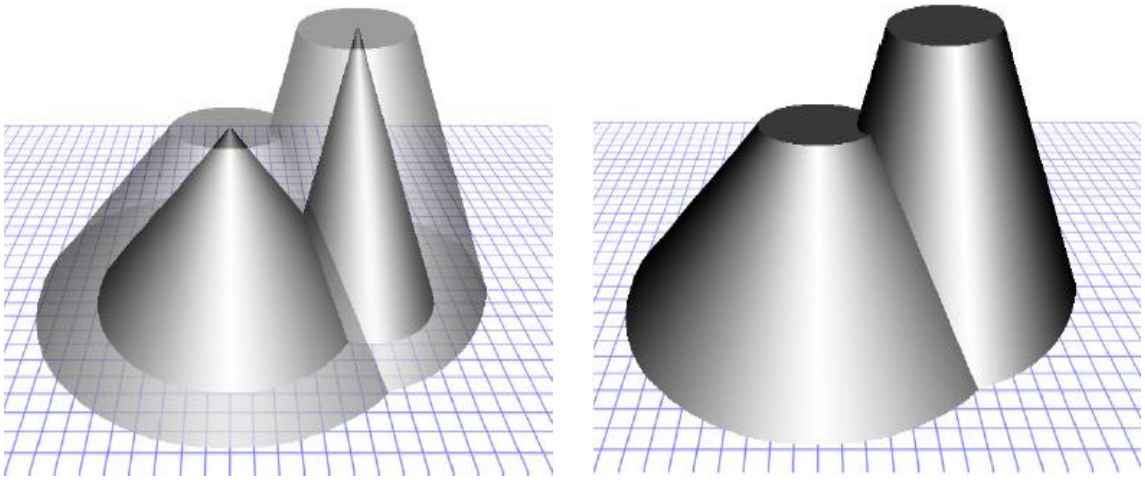
$$b(x', y') = 0 \text{ For } (x', y') \in D_b \tag{4.21}$$

In this case, the max operation is specified completely by the pattern of 0s and 1s in binary matrix D_b, and the gray-scale dilation equation simplifies to

$$(f \oplus b)(x, y) = \max\{f(x - x', y - y') + b(x', y') \mid (x', y') \in D_b\} \tag{4.22}$$

Thus, flat gray-scale dilation is a local-maximum operator, where the maximum is taken over a set of pixel neighbors determined by the shape of D_b.

The effect of dilation on an image using a flat structuring element is shown in the figure 4.7 below.



Dilation over original image Dilated Image

Fig 4.7 Process of dilation in gray scale image

4.5.2 Erosion

The gray-scale erosion of f by the structuring element b, denoted $f \Theta b$, is defined as:

$$(f \Theta b)(x, y) = \min\{f(x + x', y + y') - b(x', y') \mid (x', y') \in D_b\} \quad (4.23)$$

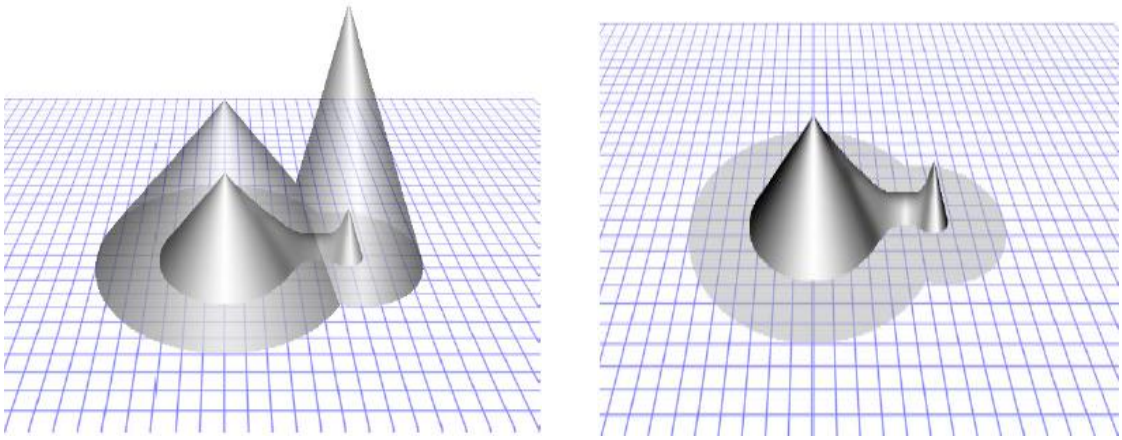
Where D_b the domain of b and f (x, y) is is assumed to equal $+\infty$ outside the domain of *f*. conceptually, we again can think of translating the structuring element to all locations in the image. At each translated location, the structuring element values are subtracted from the image pixel values and the minimum is taken.

- **Working of erosion**

As with dilation, gray-scale erosion is most often performed using flat structuring elements. The equation for flat gray-scale erosion can then be simplified to

$$(f \Theta b)(x, y) = \min\{f(x + x', y + y') \mid (x', y') \in D_b\} \quad (4.24)$$

Thus, flat gray-scale erosion is a local-minimum operator, in which the minimum is taken over a set of pixel neighbor determined by the shape of D_b. Figure 4.8 shows the results of gray-scale erosion using a flat structuring element.



Erosion under original Erosion

Fig 4.8 Process of erosion in gray scale image

4.5.3 Opening

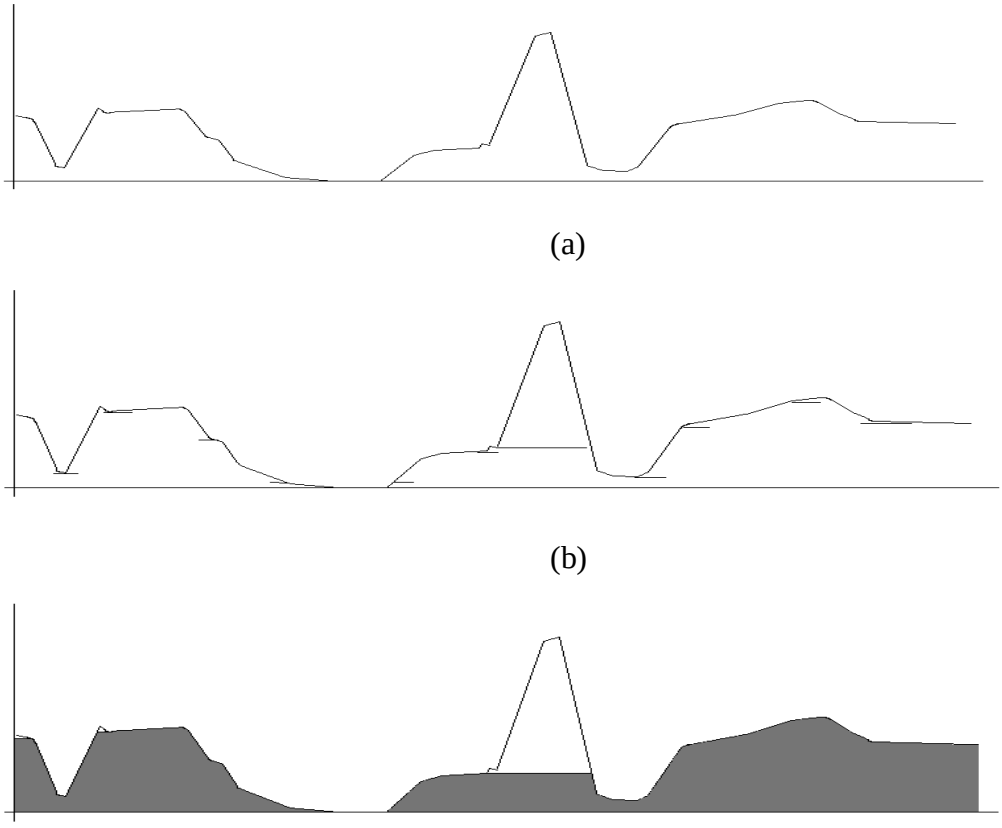
The expression for opening gray-scale images has the same form as their counter-parts. The opening of image f by structuring element b denoted by $f \circ b$, is defined as:

$$f \circ b = (f \ominus b) \oplus b \qquad (4.25)$$

- **Working of opening**

The basic operation of opening can be understood by simple geometric interpretations. Suppose that an image function f (x, y) is viewed as a 3-D surface; that is, its intensity values are interpreted as height values over the xy-plane. Then the opening of f by b can be interpreted geometrically as pushing structuring element b up against the underside of the surface and translating it across the entire domain of f. The opening is constructed by finding the highest point reached by any part of the structuring element as it slides against the undersurface of f.

The figure 4.9 next illustrates the concept in one dimension. In this a one dimensional gray scale image is taken. In this first a structuring element is taken and passed up against the bottom of the curve. Since the structuring element is too large to fit inside the upward peak on the middle of the curve, that the peak is removed by the opening. In general, opening is used to remove the small bright details while leaving the overall gray levels and large bright features relatively undisturbed.



(a)

(b)

(c)

Fig 4.9 Process of Opening in gray-scale image (a) Value along the single row of a gray scale image (b) Flat structuring element at several positions (c) Single row gray scale image after opening

4.5.4 Closing

The expression for closing gray-scale images has the same form as their counter-parts. The closing of image f by structuring element b denoted by $f \circ b$, is defined as [1]

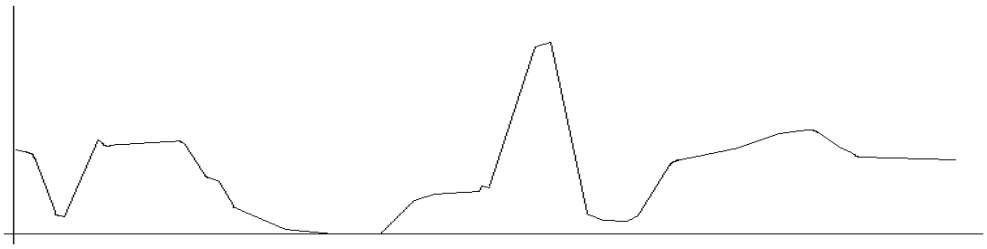
$$A \bullet B = (A \oplus B) \ominus B \tag{4.26}$$

- **Working of closing**

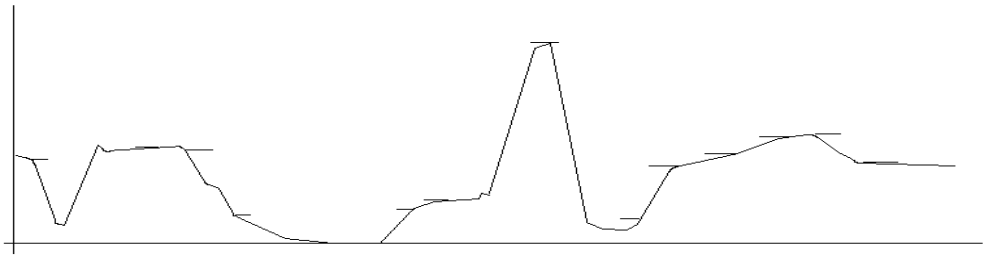
The basic operation of opening can be understood by simple geometric interpretations. Suppose that an image function f (x, y) is viewed as a 3-D surface; that is, its intensity values are interpreted as height values over the x y-plane. Then the opening of f by b can be interpreted geometrically as pushing structuring element b up against the underside of the surface and translating it across the entire domain of f. The opening is

constructed by finding the highest point reached by any part of the structuring element as it slides against the undersurface of f.

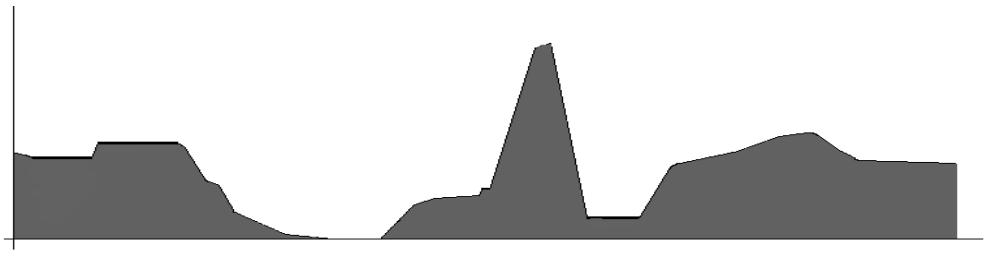
Figure 4.10 below provides the graphical illustration of closing. Note that the structuring element is passed down on top of the curve while being translated to all locations. The closing in figure 4.10 is constructed by finding the lowest points reached by any part of the structuring element as it slides against the upper side of the curve. Here we see that the closing operation suppresses the dark details smaller then the structuring element [9].



(a)



(b)



(c)

Fig 4.10 Process of Closing in gray-scale image (a) Value along the single row of a gray scale image (b) Flat structuring element at several positions (c) Single row gray scale image after closing

CHAPTER-5
METHODOLOGY

5.1 Outline

This chapter gives a sketch of the algorithm used in this thesis. In this thesis noisy gray scale images are used, consider it be *F*. The type of noise present in the images taken is speckle noise. Mathematical morphology operations are used to remove noise from image and then detect the edge of noise free images. There are many mathematical morphology techniques use to detect edges of an image having speckle noise in it but, before edge detection noise removal is necessary step. Usually opening and closing morphological operations in different combination for noise removal and then edge detection.

For speckle noise removal firstly mathematical morphology opening and closing operations are used. Let *S* be the result of smoothing *F* with openings and closings. Assume *S* is noise free. Now if we take difference between *F* and *S*, then the difference image *D* = *F*- *S* contains all the noise in *F* where *D* is called as residual image. However, *S* cannot contain any features with nonzero support that are thinner than the structuring elements used to create it. Thus, *D* contains features as well as noise. If the noise in *F* has a smaller dynamic range than the thin features, then *D* will contain noise at lower amplitude levels and features at higher amplitudes. Thus a median filter is used to remove noise from residual image and to extract thin features smaller then the size of structuring element and are added back to *S*. The algorithm described below is an elaboration of this idea using a morphological size distribution of structuring element to isolate features from noise on different scales.

After noise removal edges are detected since it's easy to detect edges of noise free images then from noisy images. Morphological operations used in different combinations for edge detection are :

1. Erosion residual edge detector

$$E(e) = F - (F \Theta B) \tag{5.1}$$

2 Dilation residue edge detector

$$E(d) = (F \oplus B) - F \tag{5.2}$$

3 Morphological gradient edge detector

$$E(g) = (A \oplus B) - (A \ominus B) \tag{5.3}$$

The results obtained by using the algorithm are compared with different speckle noise removing filters like LEE, FLSMV and SARD and different edge detectors commonly used like sobel , Prewitt and canny edge detector . By changing the threshold values the final images having edges in binary format are obtained and are compared on different parameters with other available methods.

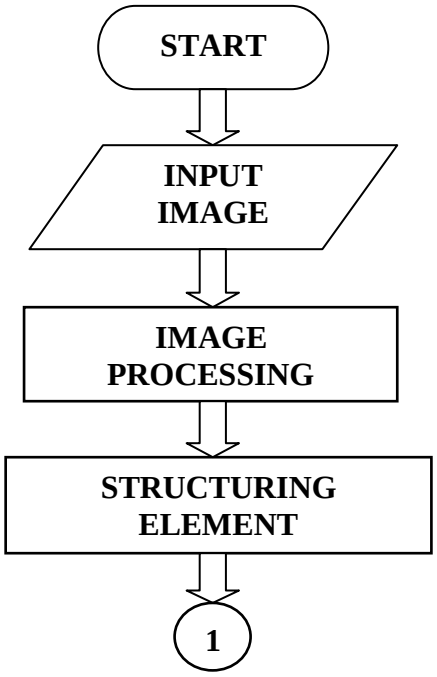
5.2 Methodology

Methodology is divided in two parts.

- Working Algorithm using MATLAB for removing speckle noise from images
- Working Algorithm using MATLAB for finding edges of filtered images

5.2.1 Working Algorithm using MATLAB for removing speckle noise from images

This working algorithm is the way of doing our work for speckle noise removal. This is a step by step procedure that how we preprocess the input images and how we removed noise from images and how our method is compared with other traditional filtering method. The main algorithm, followed in order to fulfill the aim of our work, is as show in figure 5.1 below:



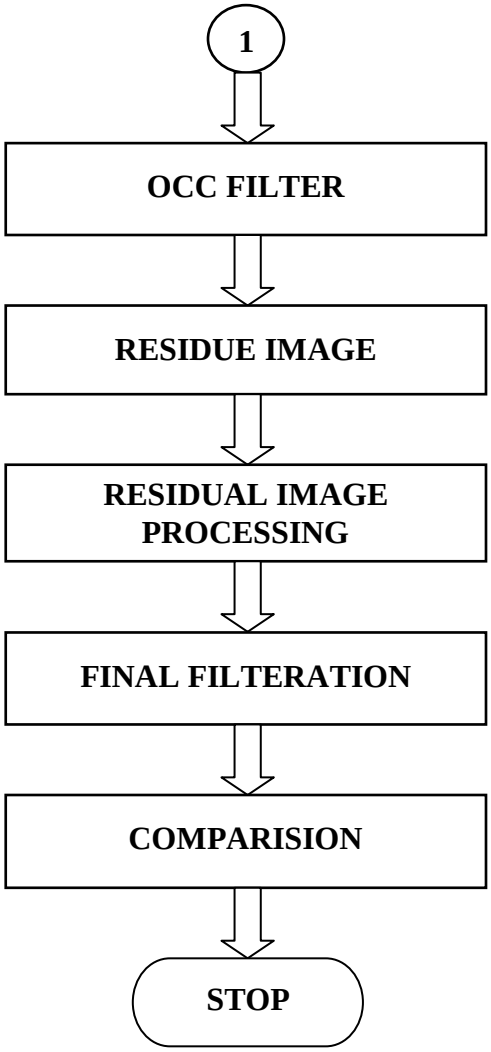


Figure 5.1 Flow chart of working algorithm 1

5.2.1.1 Image Processing

Image processing is the first step in the proposed algorithm. The main process done is the conversion of an image into gray-scale. The test images taken are the gray-scale images whose pixel intensity varies from [0, 255]. Conversion of any other type of image like RGB images are done by the MATLAB function "rgb2gray(x)", where x is the input image.

Secondly speckle noise is added to the test images. The noise level taken is in the range of 0.3 to 0.7 standard deviation.

5.2.1.2 Structuring Element

In morphology structuring element played very important role. The choice of structuring element should be highly meditated. The algorithm designed and tested using only disk shaped SE's. The parameter associated with the SE is its size (i.e. diameter) d. Single size SE can't fetch an appropriate output. It is necessary to filter the image SE's of different sizes and operate on the various residuals separately. So, the size of disk shaped SE is varied from diameter d= {1, 2, 3}.

5.2.1.3 OCC Filter

If F represents the input image, I_Z represents opening of F by (SE) Z and I^Z represents closing of F by (SE) Z, then OCC filter is defined as:

$$OCC\ (F; Z) = (I_Z)^Z + I_Z \tag{5.4}$$

That is, the OCCO filter is the pixel wise average of the open-close of I and the close of I. In the proposed algorithm successive OCC filter is applied increasing simultaneously the diameter of SE. By using OCC filter image with speckle noise is smoothed by applying the OCC filter with increasing SE. With the combination of mathematical morphological operations i.e. opening and closing and by varying the diameter of structuring element the speckle noise is suppressed.

5.2.1.4 Residual Image

As the OCC filter is applied to smooth the noisy image, with increasing SE. Although noise get removed but sometime features which are smaller or can say thinner then the SE will also get removed. To get back those features of the image which get removed while filtering process residual image method is used in the proposed algorithm. The residual image is the difference between the filtered image and the image present previous to it. Suppose we perform OCC operation on original image F ,let the image obtained after applying OCC filter is S which contains less noise level in it, but while applying OCC some important features also get removed . To bring back those features we subtract the F from S. By doing this the image obtained will be containing noise and features lost by applying OCC operation. Now by properly processing the residual image noise is removed from it and features are extracted.

5.2.1.5 Residual Image Processing

The residual image let it be D obtained by operation F - S i.e. the differences between the original image and the filtered image will be having noisy regions and feature regions which are clearly identifiable and easily separable. To remove noise from the residual image median filter is applied. Median filter is a nonlinear digital filtering technique, often used to remove noise.

5.2.1.6 Final filtration

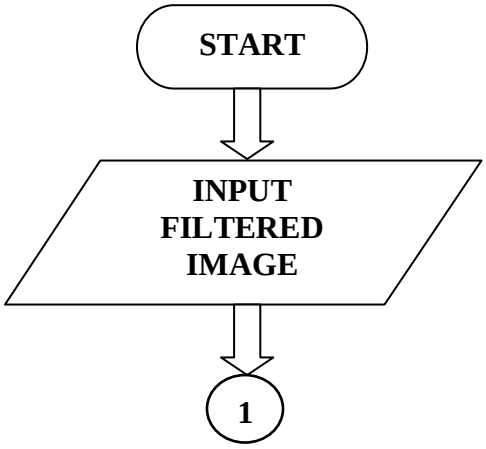
After the noise is removed from the residual images and thin features get extracted that image is added back to the imaged obtained after filtering it by OCC filter. This process help in obtaining back the thin and small features which get eliminated by applying OCC filter back to the image thus preserving the features and removing noise.

5.2.1.7 Comparison

Results are compared with other traditional methods use to filter out speckle noise. These methods are LEE, First Order Local Statistical Filter **(**lsmv) and Speckle reducing anisotropic diffusion (SRAD) filter

5.2.2 Working Algorithm using MATLAB for finding edges of filtered images

This part of algorithm tells that how the edges are obtained of the image obtained after filtering them by previous algorithm. The figure 5.2 below shows the step by step implementation of algorithm for obtaining edges of the test images and how the method proposed is compared with traditional edge detection techniques.



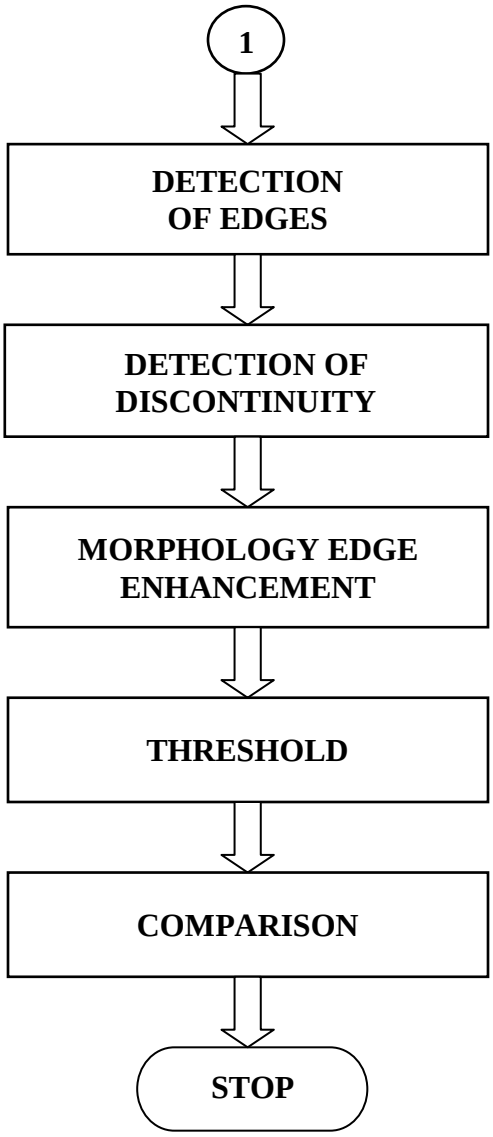


Figure 5.2 Flow chart of working algorithm 2

5.2.2.1 Detection of Edges

In the first step edges of images are obtained which have there intensity change in the image quite high i.e. those edges can be seen by eyes easily. These edges are prominent and thick features of an image which can't get affected by any size of structuring element used during morphological operations. Combination of closing and dilation process is used to perform the edge detection task. The process is that on filtered image is firstly closing followed by dilation morphological operation is done and then the difference is taken between the processed image above and the image before dilation.

Mathematically the equation is like, let M be the filtered image and Z be the structuring element then operation done is as:

$$K = (M \bullet Z) \oplus Z - M \bullet Z \quad (5.5)$$

5.2.2.2 Detection of Discontinuity

In any image there are many thin and small features whose edge detection is quite a difficult task to perform. These thin edges are called peaks and valley of an image. In the proposed algorithm to find such thin edges of image top-hat and bottom-hat mathematical morphology operations are performed. Top-hat morphological operation is given as:

$$TH = K - (K \circ Z) \quad (5.6)$$

And bottom-hat morphological operation is given as:

$$BH = (K \bullet Z) - K \quad (5.7)$$

Where K is the image output obtained in first step and Z is the structuring element.

5.2.2.3 Morphology Edge Enhancement

After getting the top-hat and bottom-hat morphological operation output all thin and think edges are obtained. Now the edges are enhanced to accentuate and enhance the edges. This is done by first taking the difference between TH and BH. And then after taking the difference the output is added to image output obtained at first step i.e. K.

5.2.2.4 Threshold

Thresholding is done after the edges are obtained in the gray-scale image. This step is done to remove little noise left in output gray-scale image, and also the edges seen in binary image format are clear then in gray-scale format.

5.2.2.5 Comparison

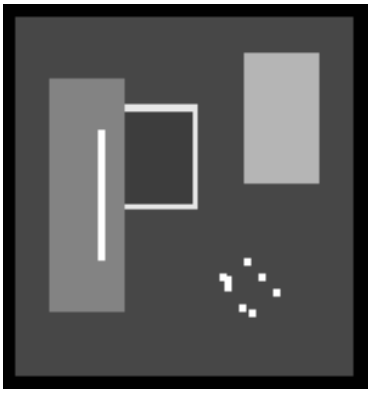
Results are compared with other traditional methods use to find the edges of the image. These methods are Sobel, Prewitt and Canny edge detectors.

CHAPTER-6

RESULT AND DISSCUSSION

6.1 Results

To analyze and evaluate the proposed algorithm for speckle noise removal and edge detection using mathematical morphology 8 different images are shown. On these images speckle noise is applied of increasing amplitude by increasing the value of standard deviation. For the evaluation, the standard deviation in speckle noise is taken from 0.3 to 0.7. Then the proposed algorithm is applied firstly to remove noise. Images obtained after removal of noise by proposed algorithm are shown and are compared with different filters previously present for the removal of speckle noise. These filters are LEE filter, FLSMV filter and SARD filter. Various quality assessment parameters are used to evaluate the filtered image obtained by proposed algorithm and the image obtained after applying other filter use to remove speckle noise. These parameters are SNR, Coc, MSSIM and RMSE. Then edges are acquired of filtered image obtained after applying proposed algorithm and the results are compared with previously used edge detectors like SOBEL detector, PREWITT detector and CANNY detector. EPI parameter is used to compare the results of edges obtained by proposed algorithm and that of other edge detectors used. On calculating the percentage improvement in the values of the quality assessment paramaters, it can be seen that the results obtained by proposed algorithm is better than the previously used filters and edge detectors.



(a)

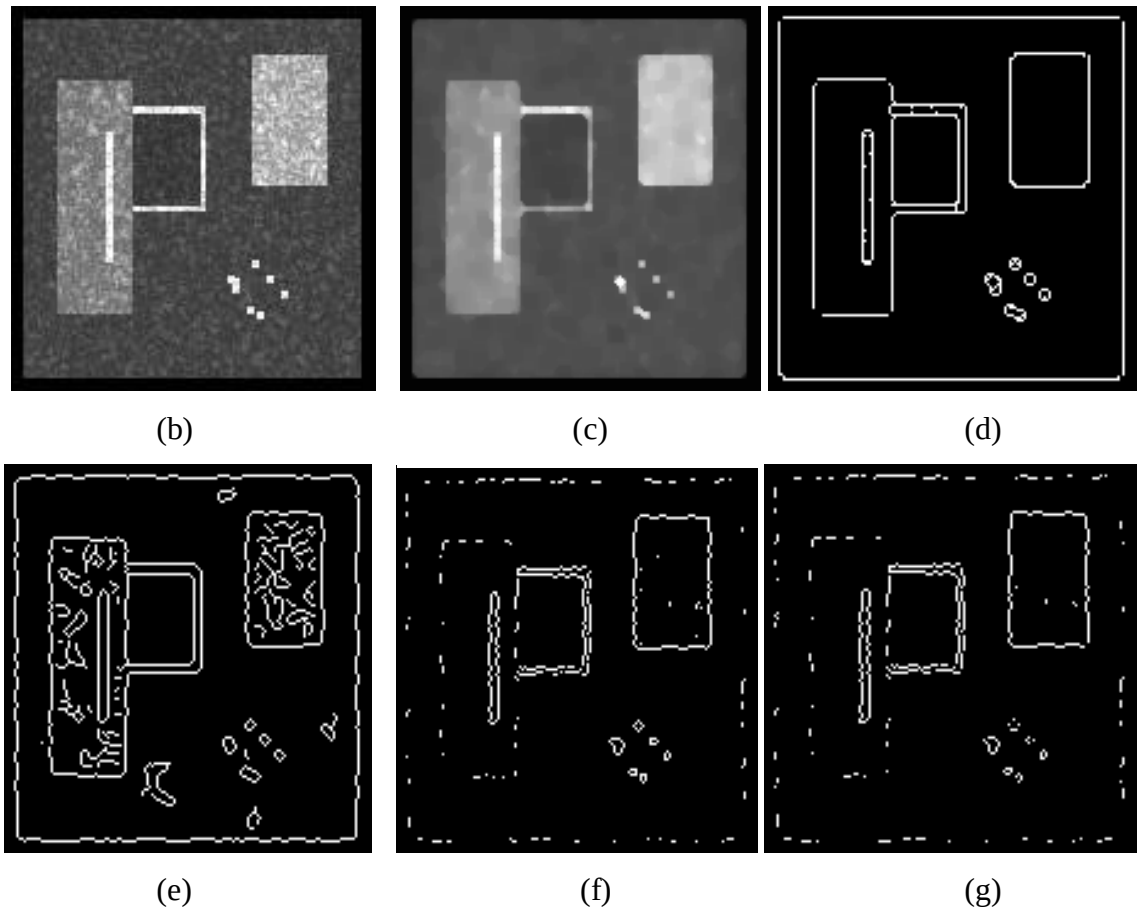


Fig 6.1 Different processes on image 1 (Synth) (a) Original Image (b) Image with Speckle Noise of std. dev. 0.6 (c) Filtered Image by Proposed Algorithm (d) Edges Obtained by Proposed Algorithm (e) Edges Obtained by Canny Operator (f) Edges Obtained by Sobel Operator (g) edges Obtained by Prewitt Operator.

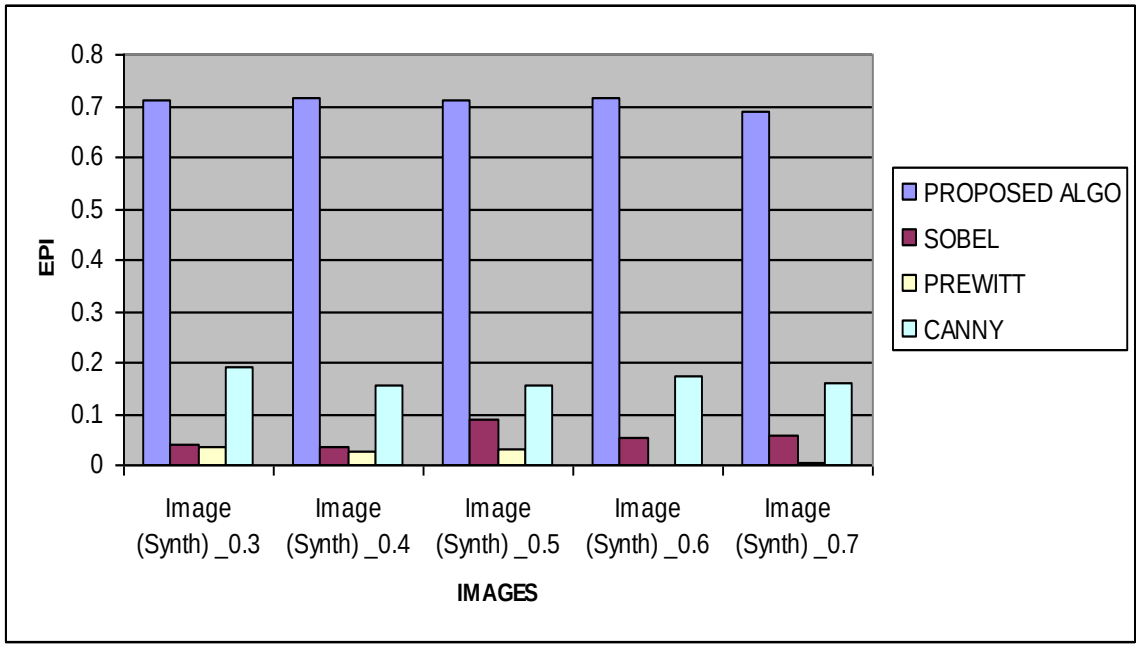
Image (Synth) with speckle noise of standard deviation 0.3	SNR	Coc	MSSIM	RMSE
Noisy Image	12.0347	0.9968	0.8025	3.7166
Filtered by (LEE) filter	10.8477	0.9437	0.8551	4.2609
Percentage Improvement (%)	-9.86%	-5.3%	6.5%	14.6%
Filtered by (lsmv) filter	8.8495	0.8982	0.8335	5.3630
Percentage Improvement (%)	-26.4%	-9.9%	3.8%	44.4%
Filtered by (SARD) filter	9.1412	0.9536	0.8969	5.1859
Percentage Improvement (%)	-24.04%%	-4.4%	11.7%	39.5%
Filtered by proposed algo	17.9019	0.9979	0.9617	1.8914
Percentage Improvement (%)	48.7%	0.2%	19.8%	-49.2%

Image (Synth) with speckle noise of standard deviation 0.4	SNR	Coc	MSSIM	RMSE
Noisy Image	9.756	0.9826	0.7137	4.8315
Filtered by (LEE) filter	10.1784	0.943	0.8153	4.6022
Percentage Improvement (%)	4.3%	-4.1%	14.2%	4.8%
Filtered by (lsmv) filter	8.7421	0.8971	0.8289	5.4297
Percentage Improvement (%)	-10.4%	-8.8%	16.1%	12.3%
Filtered by (SARD) filter	8.9957	0.9546	0.8931	5.2735
Percentage Improvement (%)	-8.8%	-2.8%	25.1%	09.1%
Filtered by proposed algo	17.0832	0.9971	0.9452	2.0784
Percentage Improvement (%)	75.1%	1.4%	32.4%	-57%
Image (Synth) with speckle noise of standard deviation 0.5	SNR	Coc	MSSIM	RMSE
Noisy Image	8.5332	0.9628	0.6481	5.5619
Filtered by (LEE) filter	9.6245	0.9413	0.7817	4.9052
Percentage Improvement (%)	12.7%	-2.3%	20.6%	-11.9%
Filtered by (lsmv) filter	8.7138	0.8988	0.8262	5.4475
Percentage Improvement (%)	2.1%	-6.7%	28.7%	-2.1%
Filtered by (SARD) filter	9.0430	0.9557	0.8917	5.2448
Percentage Improvement (%)	5.9%	-0.8%	37.5%	-5.8%
Filtered by proposed algo	16.8911	0.9939	0.9328	2.1248
Percentage Improvement (%)	97.9%	3.2%	43.9%	-61.8%
Image (Synth) with speckle noise of standard deviation 0.6	SNR	Coc	MSSIM	RMSE
Noisy Image	7.3811	0.9391	0.5814	6.3508
Filtered by (LEE) filter	8.9699	0.9398	0.7439	5.2892
Percentage Improvement (%)	21.5%	0.07%	27.9%	-6.8%
Filtered by (lsmv) filter	8.6872	0.9004	0.8233	5.4641
Percentage Improvement (%)	17.6%	-4.2%	41.6%	-13.97%
Filtered by (SARD) filter	9.0677	0.9603	0.8902	5.2300
Percentage Improvement (%)	22.8%	2.2%	53.1%	-17.7%
Filtered by proposed algo	16.47	0.994	0.9142	2.2304
Percentage Improvement (%)	123.1%	5.8%	57.2%	-64.9%
Image (Synth) with speckle noise of standard deviation 0.7	SNR	Coc	MSSIM	RMSE
Noisy Image	6.4972	0.9056	0.5191	7.0311
Filtered by (LEE) filter	8.4293	0.9303	0.691	5.6289
Percentage Improvement (%)	29.7%	2.7%	33.1%	-19.95%
Filtered by (lsmv) filter	8.7580	0.8986	0.8165	5.4198
Percentage Improvement (%)	37.4%	-0.8%	57.2%	-22.92%
Filtered by (SARD) filter	9.1584	0.9617	0.8865	5.1756
Percentage Improvement (%)	40.9%	6.1%	70.7%	-26.4%
Filtered by proposed algo	16.0281	0.9883	0.8851	2.3468
Percentage Improvement (%)	146.6%	9.1%	70.5%	-66.7%

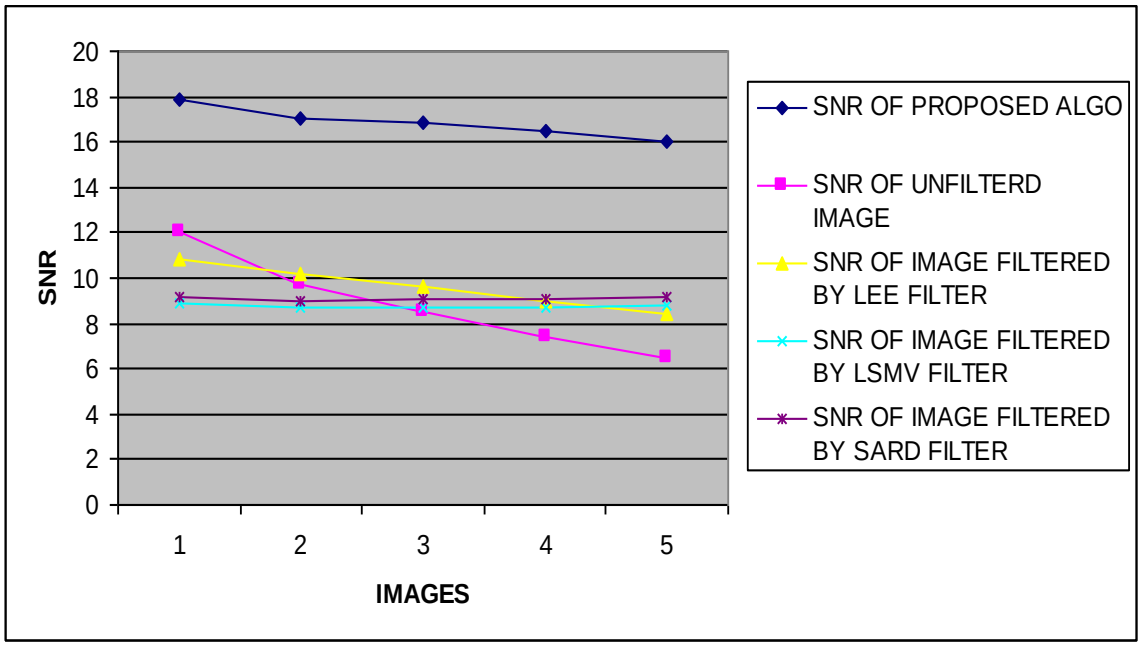
Table 6.1 Statistical results of image 1 (Synth) on parameters SNR, Coc, MSSIM, RMSE and Percentage Improvement

Edge comparison by Edge preserving index (EPI)	Proposed Algo	SOBEL	PREWITT	CANNY
Image (Synth) _0.3	0.7130	0.0412	0.0365	0.1892
Image (Synth) _0.4	0.7156	0.0335	0.0256	0.1544
Image (Synth) _0.5	0.7097	0.0869	0.0308	0.1576
Image (Synth) _0.6	0.7158	0.0524	0.0010	0.1720
Image (Synth) _0.7	0.6899	0.0567	0.0039	0.1591

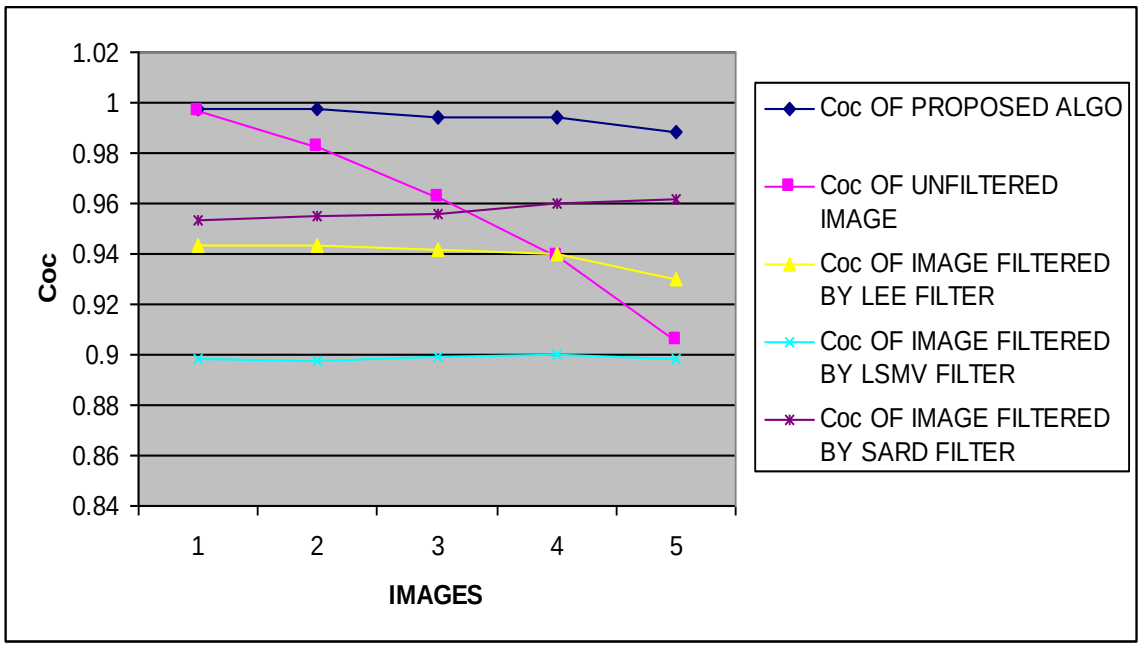
Table 6.2 Statistical results of image 1 (Synth) for parameter EPI



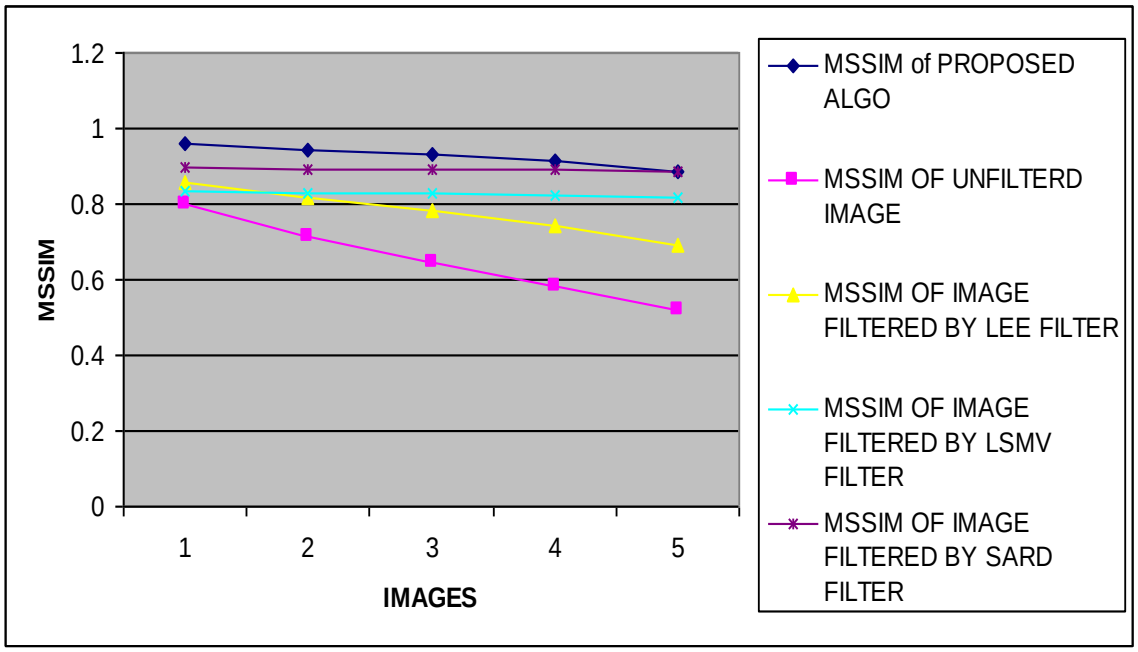
Graph 6.1 Representation of results of EPI parameter on image 1 (Synth)



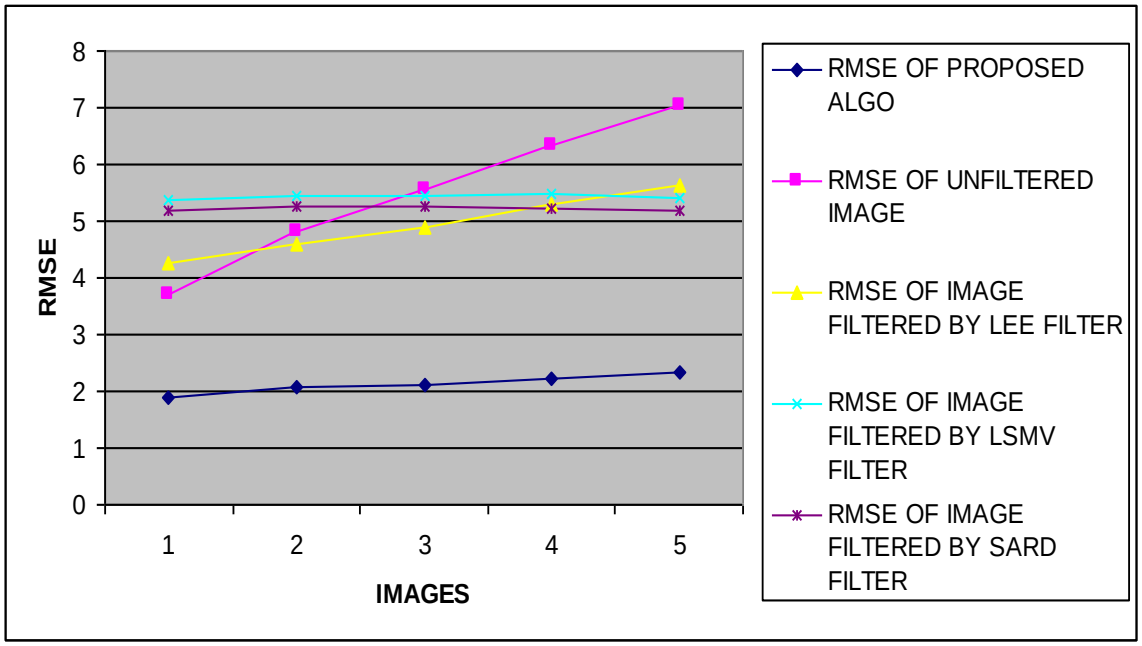
Graph 6.2 Representation of results of SNR parameter on image 1 (Synth)



Graph 6.3 Representation of results of Coc parameter on image 1 (Synth)



Graph 6.4 Representation of results of MSSIM parameter on image 1 (Synth)



Graph 6.5 Representation of results of RMSE parameter on image 1 (Synth)

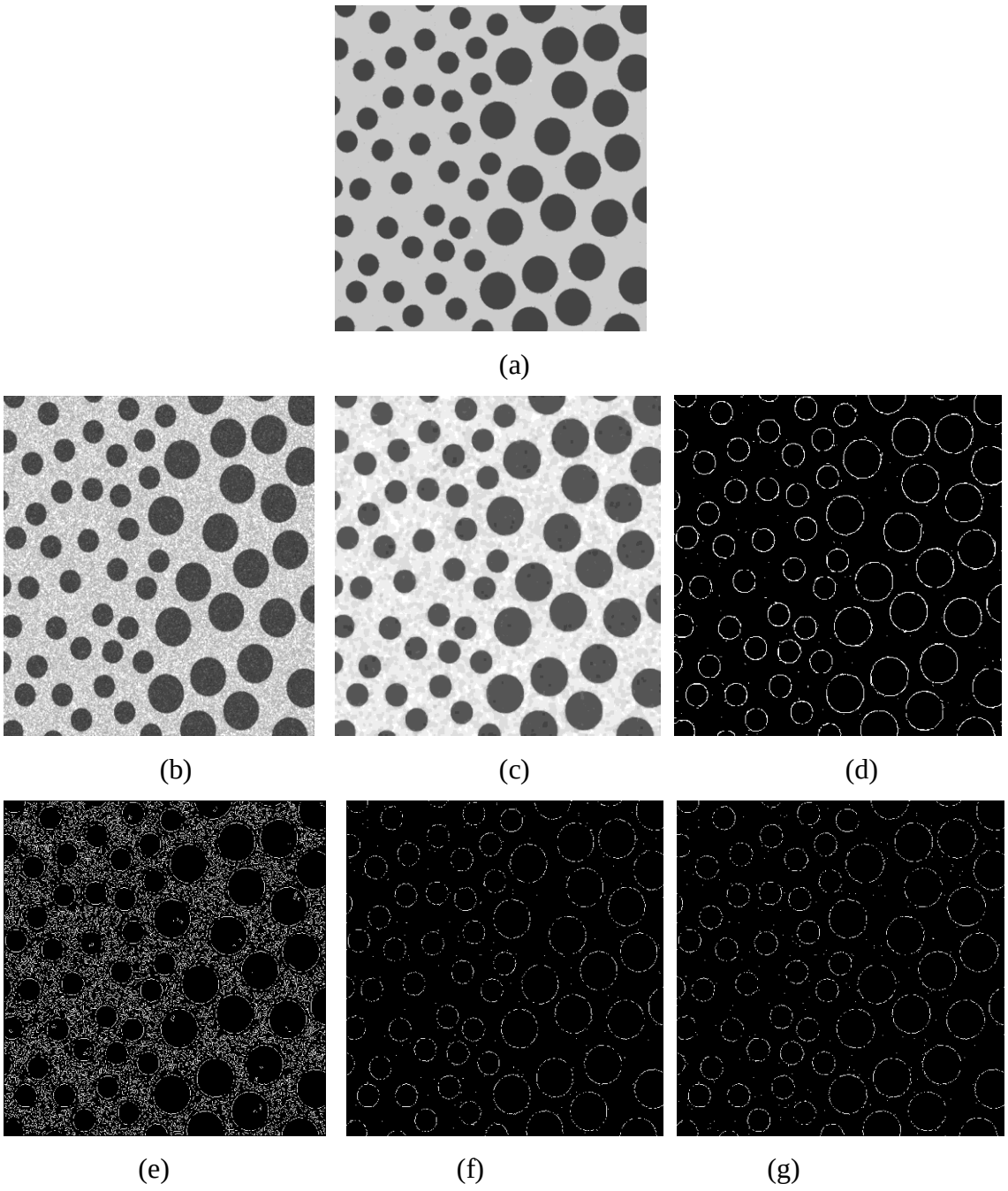


Fig 6.2 Different processes on image 2 (Circles) (a) Original Image (b) Image with Speckle Noise of std. dev. 0.7 (c) Filtered Image by Proposed Algorithm (d) Edges Obtained by Proposed Algorithm (e) Edges Obtained by Canny Operator (f) Edges Obtained by Sobel Operator (g) edges Obtained by Prewitt Operator

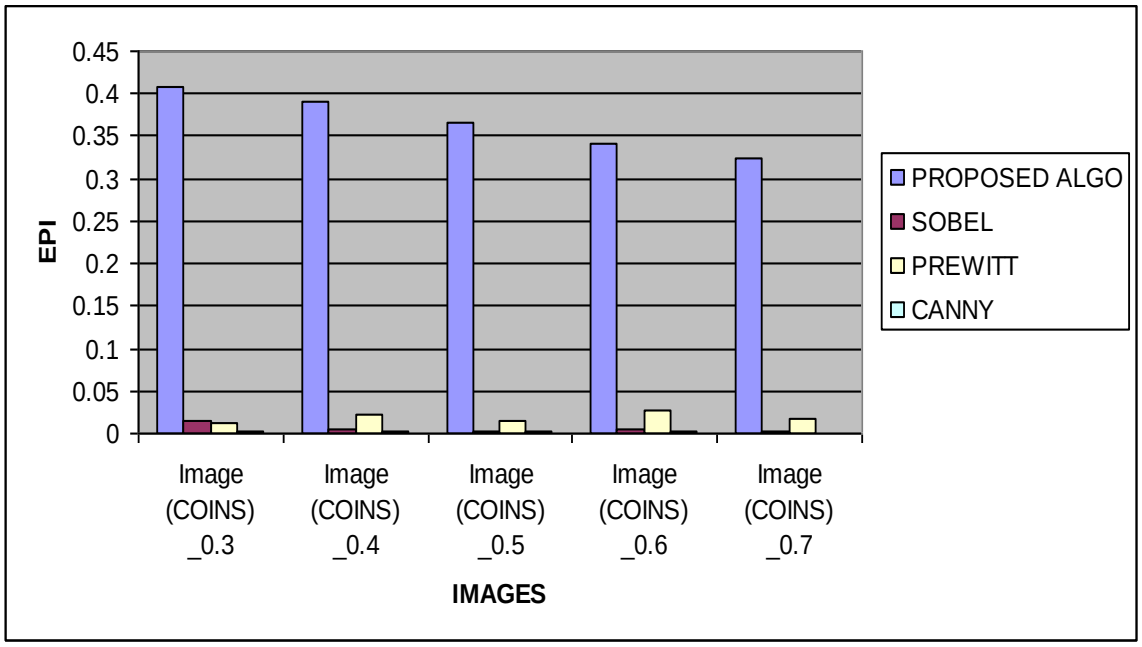
Image (Circles) with speckle noise of standard deviation 0.3	SNR	Coc	MSSIM	RMSE
Noisy Image	7.783	0.9994	0.5609	6.5181
Filtered by (LEE) filter	10.563	0.9956	0.7747	4.7328
Percentage Improvement (%)	35.7%	-0.4%	38.1%	-27.4%
Filtered by (lsmv) filter	9.0978	0.9975	0.8728	5.6025
Percentage Improvement (%)	16.9%	-0.1%	55.6%	-14.1%
Filtered by (SARD) filter	0.2335	0.9948	0.7467	15.5451
Percentage Improvement (%)	-97%	-0.6%	33.1%	138.5%
Filtered by proposed algo	23.6285	0.9976	0.9202	1.0516
Percentage Improvement (%)	203.5%	-0.2%	64.1%	-83.9%
Image (Circles) with speckle noise of standard deviation 0.4	SNR	Coc	MSSIM	RMSE
Noisy Image	6.503	0.9993	0.4657	7.553
Filtered by (LEE) filter	9.0616	0.9954	0.6982	5.6259
Percentage Improvement (%)	39.3%	-0.4%	49.9%	-25.6%
Filtered by (lsmv) filter	8.9372	0.9967	0.8652	5.707
Percentage Improvement (%)	37.4%	-0.3%	85.7%	-24.5%
Filtered by (SARD) filter	0.4705	0.995	0.764	15.1268
Percentage Improvement (%)	-92.8%	-0.5%	64.05%	100.2%
Filtered by proposed algo	21.6986	0.9974	0.8892	1.3132
Percentage Improvement (%)	233.6%	-0.2%	90.9%	-82.7%
Image (Circles) with speckle noise of standard deviation 0.5	SNR	Coc	MSSIM	RMSE
Noisy Image	5.7365	0.9934	0.3987	8.2499
Filtered by (LEE) filter	0.7736	0.9949	0.7745	14.608
Percentage Improvement (%)	-86.6%	0.1%	94.2%	77.1%
Filtered by (lsmv) filter	8.6472	0.9957	0.8558	5.9008
Percentage Improvement (%)	50.7%	0.2%	114.63%	-28.5%
Filtered by (SARD) filter	7.9157	0.995	0.6343	6.4193
Percentage Improvement (%)	37.9%	0.1%	59.1%	-22.2%
Filtered by proposed algo	21.2823	0.9971	0.8559	1.4207
Percentage Improvement (%)	270.9%	0.3%	114.66%	-82.8%
Image (Circles) with speckle noise of standard deviation 0.6	SNR	Coc	MSSIM	RMSE
Noisy Image	5.1499	0.9835	0.3477	8.8262
Filtered by (LEE) filter	7.0868	0.9947	0.5797	7.0621
Percentage Improvement (%)	37.6%	1.1%	66.7%	20%
Filtered by (lsmv) filter	8.3527	0.995	0.848	6.1043
Percentage Improvement (%)	62.1%	1.2%	143.8%	30.9%
Filtered by (SARD) filter	0.9592	0.9952	0.7857	14.2992
Percentage Improvement (%)	86.2%	1.18%	125.9%	62.0%
Filtered by proposed algo	20.3796	0.9968	0.8265	1.5286
Percentage Improvement (%)	295.7%	1.35%	137.7%	82.7%

Image (Circles) with speckle noise of standard deviation 0.7	SNR	Coc	MSSIM	RMSE
Noisy Image	4.7581	0.9942	0.309	9.2335
Filtered by (LEE) filter	6.499	0.995	0.5357	7.5565
Percentage Improvement (%)	36.5%	0.08%	73.3%	-18.2%
Filtered by (lsmv) filter	7.9513	0.9946	0.84	6.393
Percentage Improvement (%)	67.1%	0.04%	171.8%	-30.8%
Filtered by (SARD) filter	1.1381	0.9951	0.7939	14.0077
Percentage Improvement (%)	-76.1%	0.09%	156.9%	51.7%
Filtered by proposed algo	20.1396	0.9967	0.8023	1.5714
Percentage Improvement (%)	323.2%	0.2%	159.6%	-83%

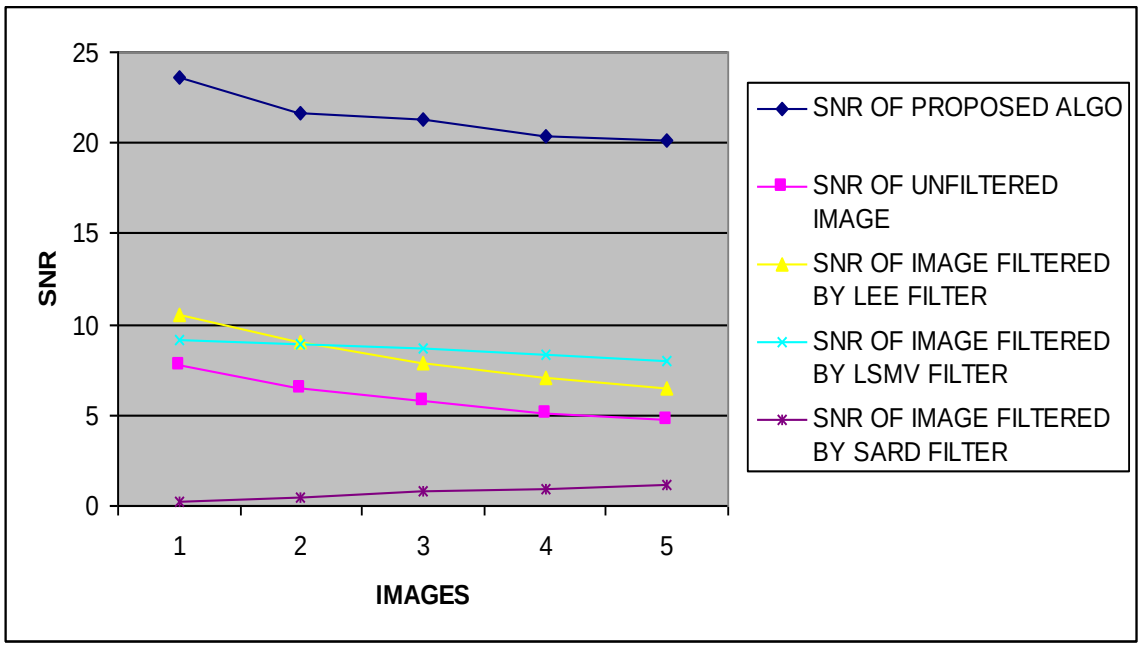
Table 6.3 Statistical results of image 2 (Circles) on parameters SNR, Coc, MSSIM, RMSE and Percentage Improvement

Edge comparison by Edge preserving index (EPI)	Proposed Algo	SOBEL	PREWITT	CANNY
Image (Circles) _0.3	0.4068	0.0141	0.0135	0.0027
Image (Circles) _0.4	0.3897	0.0055	0.022	0.0025
Image (Circles) _0.5	0.367	0.0018	0.0139	0.0023
Image (Circles) _0.6	0.341	0.0043	0.0282	0.0014
Image (Circles) 0.7	0.3239	0.0023	0.0185	0.001

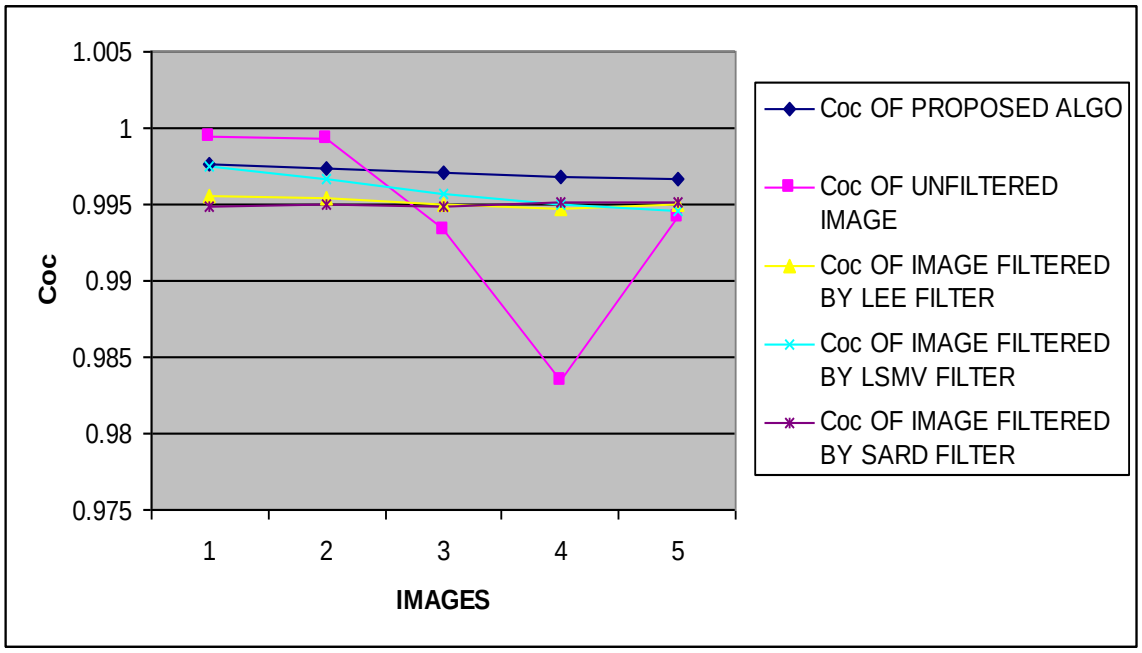
Table 6.4 Statistical results of image 2 (Circles) for the parameter EPI



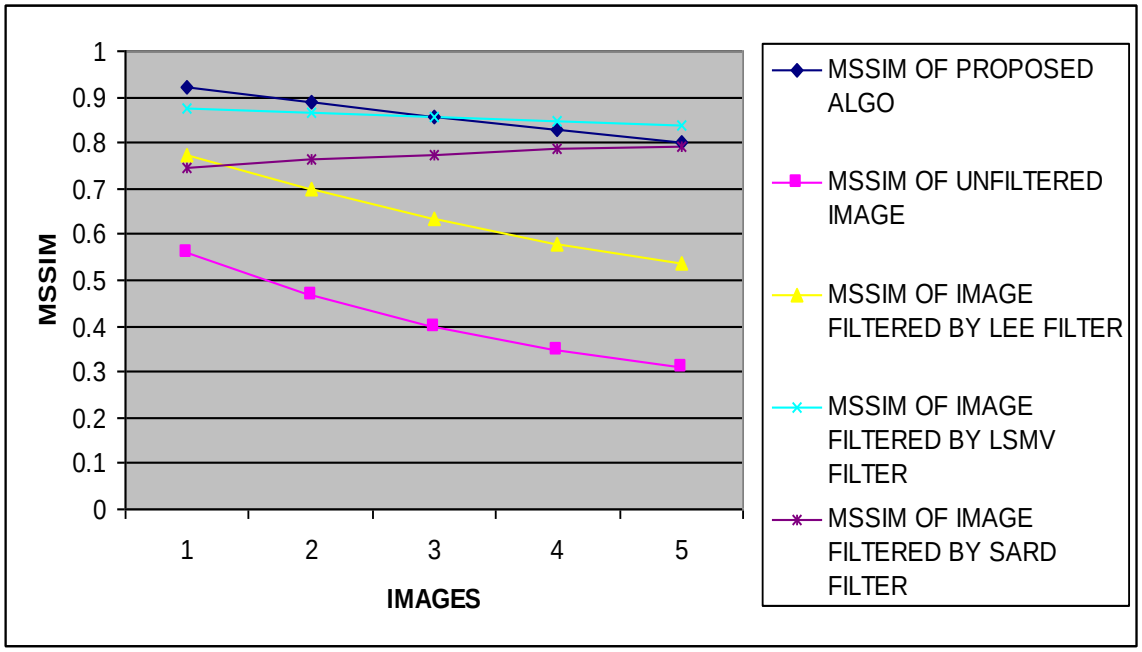
Graph 6.6 Representation of results of EPI parameter on image 2(Circles)



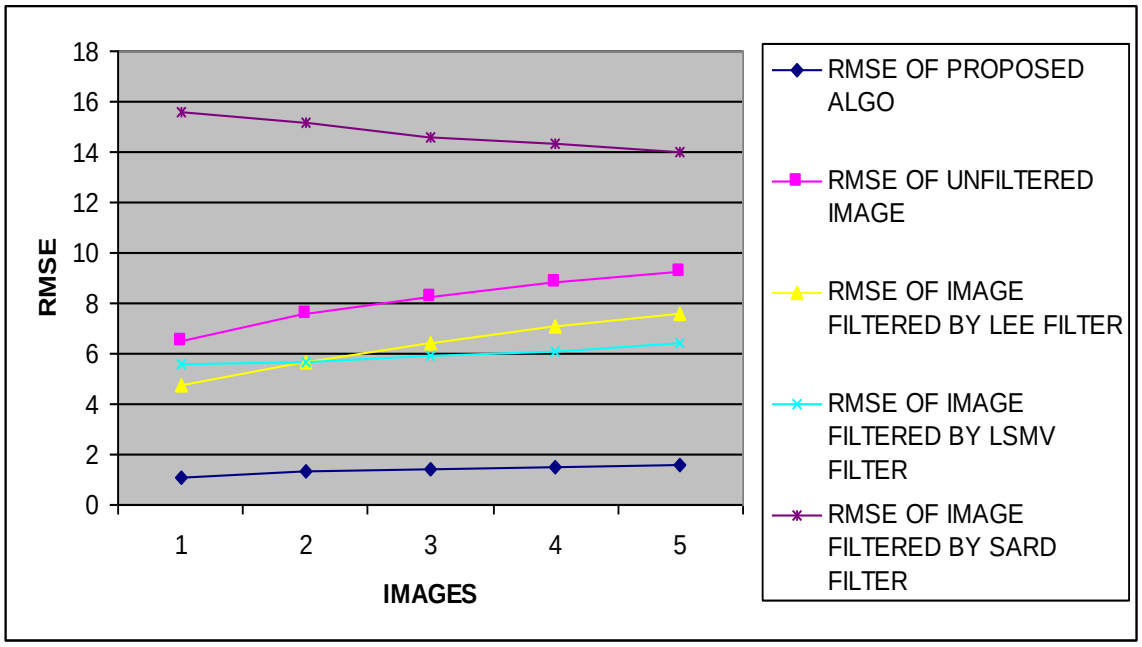
Graph 6.7 Representation of results of SNR parameter on image 2(Circles)



Graph 6.8 Representation of results of Coc parameter on image 2(Circles)



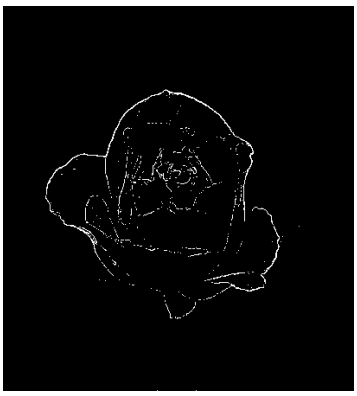
Graph 6.9 Representation of results of MSSIM parameter on image 2(Circles)



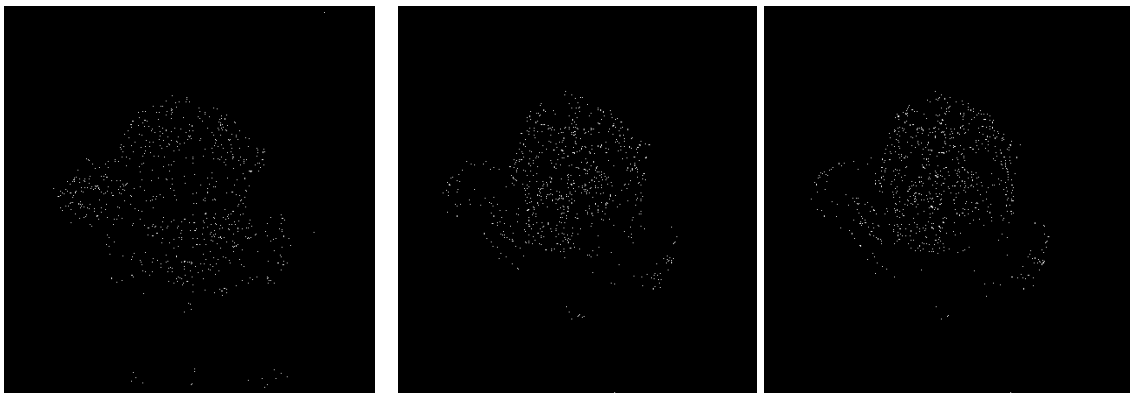
Graph 6.10 Representation of results of RMSE parameter on image 2(Circles)

(b)

(c)



(d)



(e) (f) (g)

Fig 6.3 Different process on image 3 (Flower) (a) Original Image (b) Image with Speckle Noise of std. dev. 0.3 (c) Filtered Image by Proposed Algorithm (d) Edges Obtained by Proposed Algorithm (e) Edges Obtained by Canny Operator (f) Edges Obtained by Sobel Operator (g) edges Obtained by Prewitt Operator

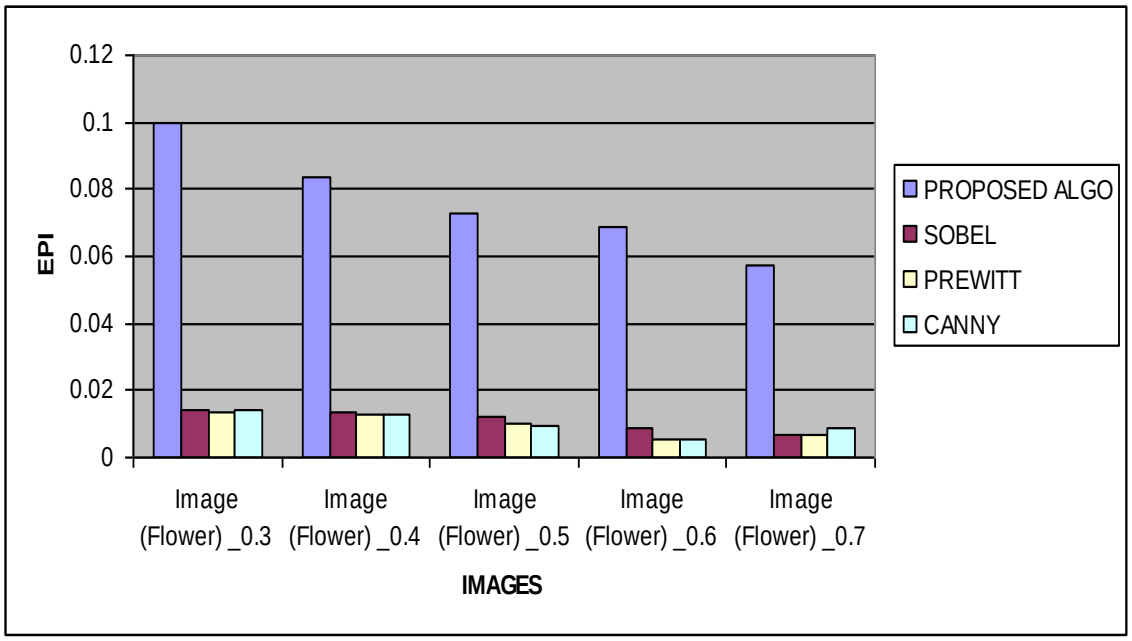
Image (Flower) with speckle noise of standard deviation 0.3	SNR	Coc	MSSIM	RMSE
Noisy Image	12.6732	0.9997	0.9201	3.1524
Filtered by (LEE) filter	12.9682	0.9986	0.8838	3.0471
Percentage Improvement (%)	2.3%	-0.2%	-3.95%	-3.4%
Filtered by (lsmv) filter	11.9274	0.9976	0.8747	3.435
Percentage Improvement (%)	-5.9%	-0.3%	-4.98%	8.9%
Filtered by (SARD) filter	12.3983	0.9997	0.8956	3.2538
Percentage Improvement (%)	-2.2%	0%	-2.7%	3.2%
Filtered by proposed algo	22.8047	0.9994	0.8727	0.9819
Percentage Improvement (%)	79.9%	-0.1%	-5.2%	-68.9%
Image (Flower) with speckle noise of standard deviation 0.4	SNR	Coc	MSSIM	RMSE
Noisy Image	11.168	0.9996	0.8861	3.7489
Filtered by (LEE) filter	12.1023	0.9985	0.871	3.3666
Percentage Improvement (%)	8.3%	-0.2%	-1.8%	-10.2%
Filtered by (lsmv) filter	11.6759	0.9975	0.8738	3.5359
Percentage Improvement (%)	4.5%	-0.3%	-1.4%	-5.7%
Filtered by (SARD) filter	11.0045	0.9996	0.8663	3.8201
Percentage Improvement (%)	-1.5%	0%	-2.3%	1.8%
Filtered by proposed algo	22.7651	0.9992	0.8663	0.9864

Percentage Improvement (%)	103.8%	-0.1%	-2.3%	-73.7%
Image (Flower) with speckle noise of standard deviation 0.5	SNR	Coc	MSSIM	RMSE
Noisy Image	10.2386	0.9995	0.8567	4.1722
Filtered by (LEE) filter	11.2932	0.9983	0.8574	3.6952
Percentage Improvement (%)	10.3%	-0.2%	0.08%	-11.5%
Filtered by (lsmv) filter	11.3282	0.9975	0.8726	3.6804
Percentage Improvement (%)	10.6%	-0.3%	1.8%	-11.8%
Filtered by (SARD) filter	10.1412	0.9995	0.8425	4.2193
Percentage Improvement (%)	-1%	0%	-1.7%	1.1%
Filtered by proposed algo	22.8137	0.9992	0.8604	0.9809
Percentage Improvement (%)	122.8%	-0.1%	0.4%	-76.5%
Image (Flower) with speckle noise of standard deviation 0.6	SNR	Coc	MSSIM	RMSE
Noisy Image	9.6687	0.9991	0.8293	4.4552
Filtered by (LEE) filter	10.7113	0.9982	0.8443	3.9512
Percentage Improvement (%)	10.8%	-0.1%	1.8%	-11.4%
Filtered by (lsmv) filter	11.0581	0.9975	0.8713	3.7966
Percentage Improvement (%)	14.3%	-0.2%	5.06%	-14.8%
Filtered by (SARD) filter	9.6244	0.9992	0.8213	4.478
Percentage Improvement (%)	-0.5%	0.01%	-0.97%	0.5%
Filtered by proposed algo	22.989	0.9991	0.8536	0.9613
Percentage Improvement (%)	137.7%	0%	2.9%	-78.5%
Image (Flower) with speckle noise of standard deviation 0.7	SNR	Coc	MSSIM	RMSE
Noisy Image	9.1311	0.9988	0.8045	4.7397
Filtered by (LEE) filter	10.0558	0.998	0.8312	4.261
Percentage Improvement (%)	10.1%	-0.1%	3.3%	-10.1%
Filtered by (lsmv) filter	10.6357	0.9975	0.8699	3.9858
Percentage Improvement (%)	16.4%	-0.2%	8.1%	-16%
Filtered by (SARD) filter	9.1386	0.9989	0.8038	4.7356
Percentage Improvement (%)	0.08%	0.01%	-5.3%	-0.1%
Filtered by proposed algo	22.7802	0.999	0.8481	0.9847
Percentage Improvement (%)	149.4%	0.02%	5.4%	-79.3%

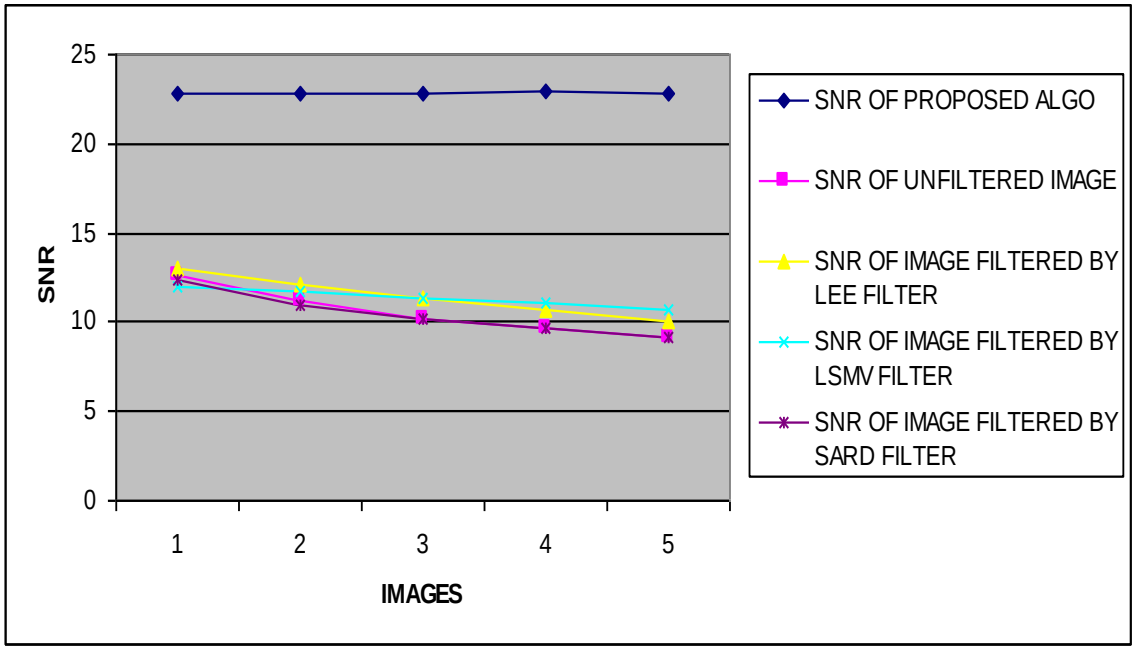
Table 6.5 Statistical results of image 3 (Flower) on parameters SNR, Coc, MSSIM, RMSE and Percentage Improvement

Edge comparison by Edge preserving index (EPI)	Proposed Algo	SOBEL	PREWITT	CANNY
Image (Flower) _0.3	0.0997	0.0141	0.0135	0.0143
Image (Flower) _0.4	0.0836	0.0132	0.0131	0.0126
Image (Flower) _0.5	0.0731	0.0121	0.0098	0.0095
Image (Flower) _0.6	0.0685	0.0088	0.0052	0.0055
Image (Flower) _0.7	0.0575	0.007	0.0069	0.0086

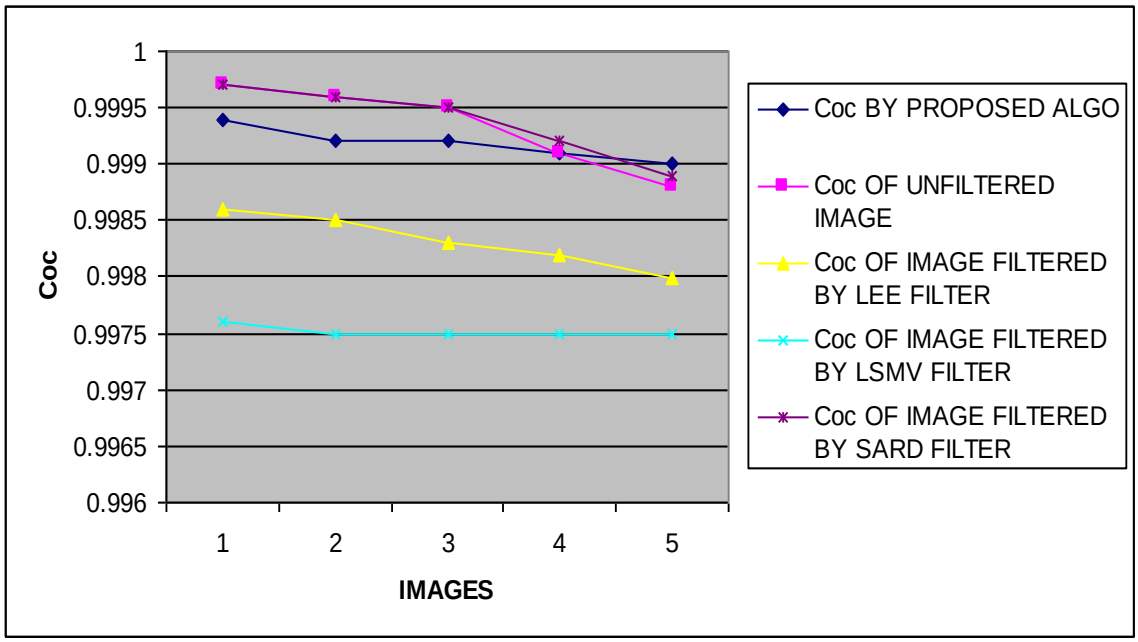
Table 6.6 Statistical results of image 3 (Flower) for the parameter EPI



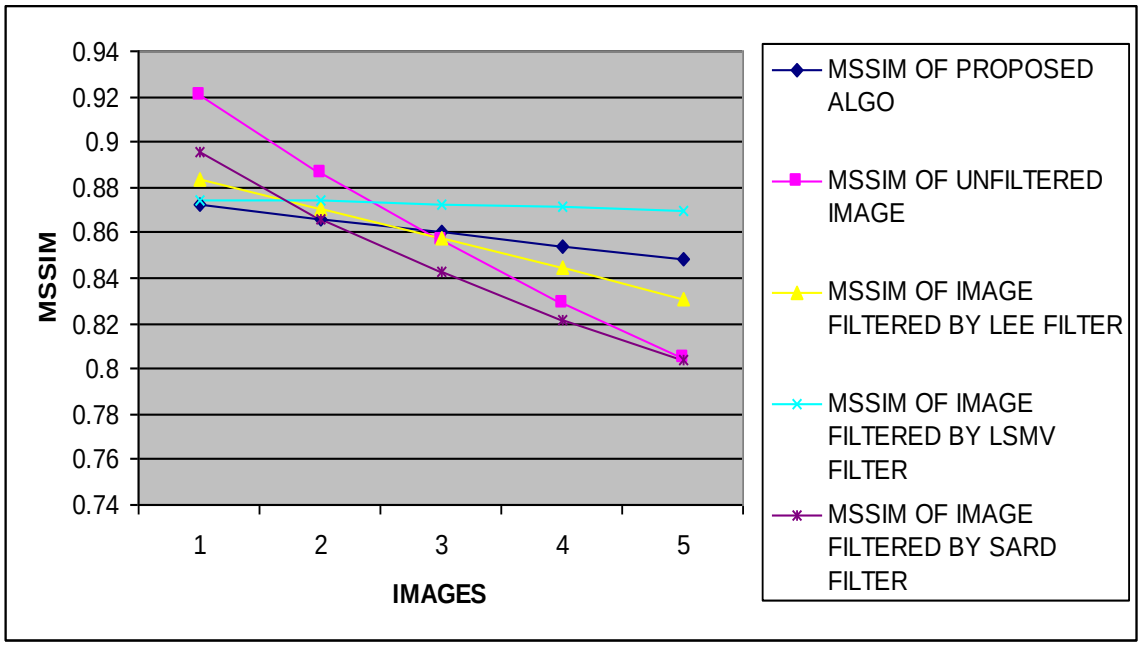
Graph 6.11 Representation of results of EPI parameter on image 3 (Flower)



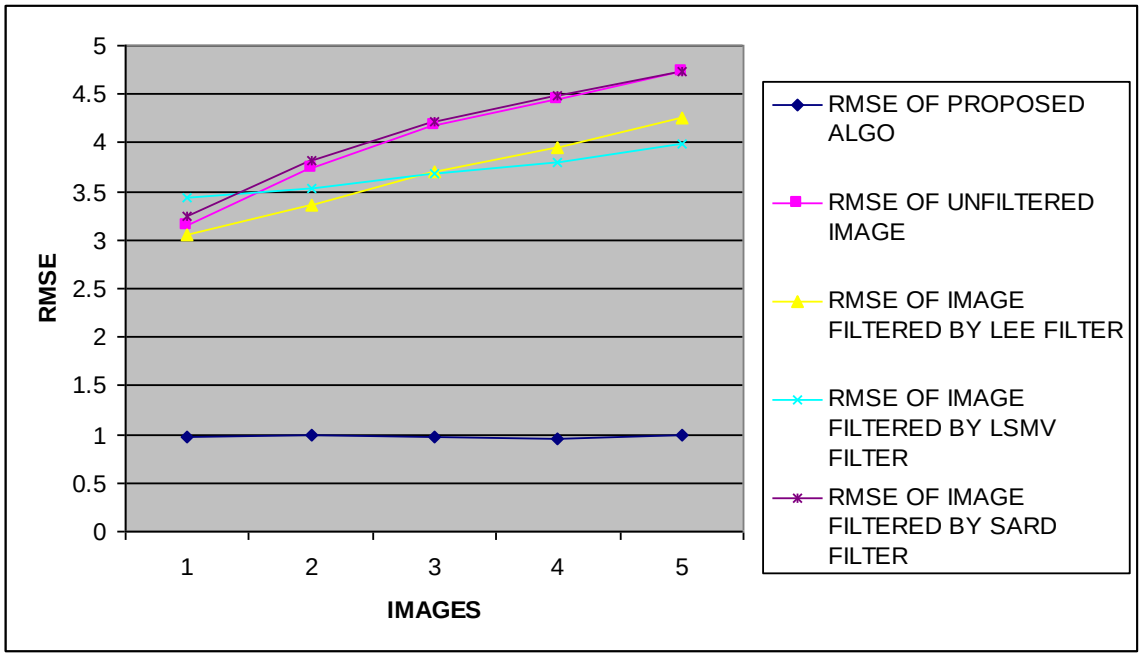
Graph 6.12 Representation of results of SNR parameter on image 3 (Flower)



Graph 6.13 Representation of results of Coc parameter on image 3 (Flower)



Graph 6.14 Representation of results of MSSIM parameter on image 3 (Flower)



Graph 6.15 Representation of results of RMSE parameter on image 3 (Flower)

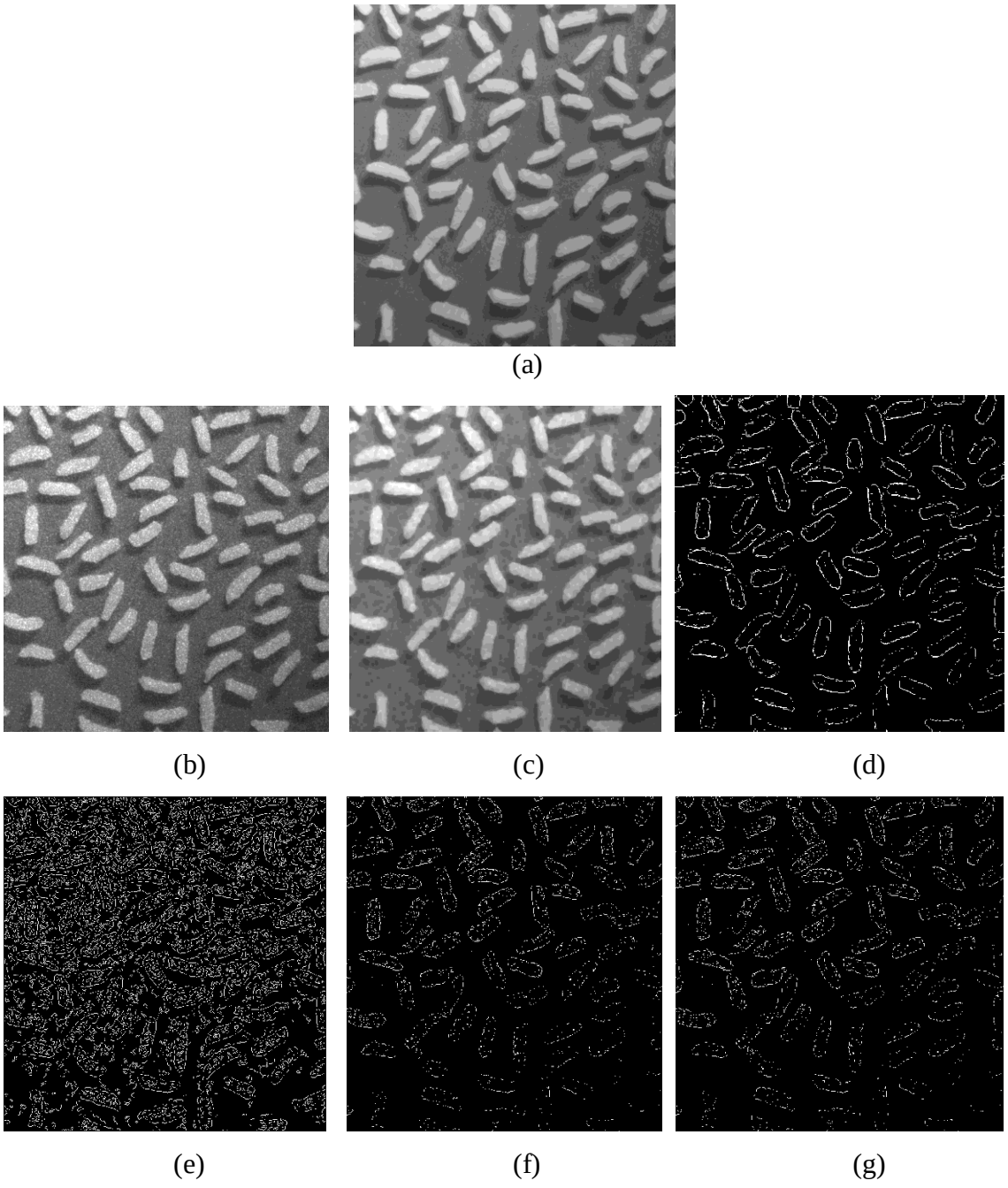


Fig 6.4 Different processes on image 4 (Rice) (a) Original Image (b) Image with Speckle Noise of std. dev. 0.5 (c) Filtered Image by Proposed Algorithm (d) Edges Obtained by Proposed Algorithm (e) Edges Obtained by Canny Operator (f) Edges Obtained by Sobel Operator (g) edges Obtained by Prewitt Operator.

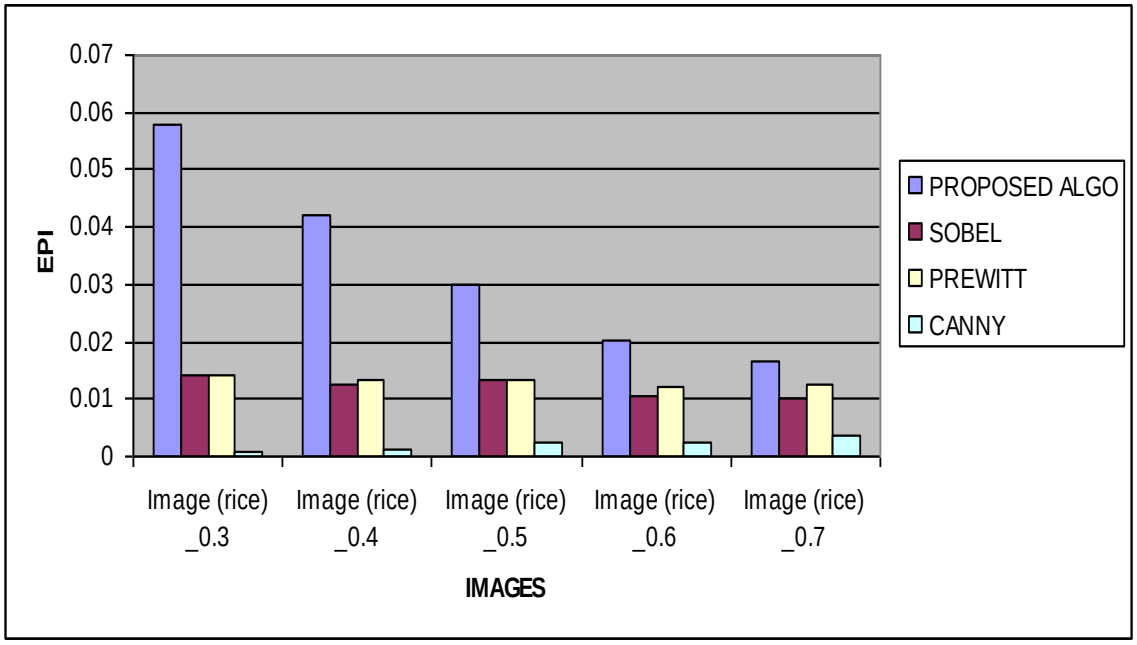
Image (Rice) with speckle noise of standard deviation 0.3	SNR	Coc	MSSIM	RMSE
Noisy Image	9.9677	0.9782	0.7052	5.0685
Filtered by (LEE) filter	13.6567	0.9874	0.8878	3.3146
Percentage Improvement (%)	37%	0.9%	25.8%	-34.7%
Filtered by (lsmv) filter	10.2685	0.9811	0.9044	4.896
Percentage Improvement (%)	3%	0.2%	41.8%	-3.5%
Filtered by (SARD) filter	0.5566	0.9803	0.8424	14.9776
Percentage Improvement (%)	-94.5%	0.2%	19.4%	195.5%
Filtered by proposed algo	22.4123	0.9897	0.919	1.2096
Percentage Improvement (%)	124.8%	1.1%	30.3%	-76.2%
Image (Rice) with speckle noise of standard deviation 0.4	SNR	Coc	MSSIM	RMSE
Noisy Image	8.0436	0.9654	0.5978	6.3254
Filtered by (LEE) filter	11.8192	0.9816	0.8331	4.0956
Percentage Improvement (%)	46.9%	1.6%	39.3%	-35.3%
Filtered by (lsmv) filter	10.163	0.9802	0.901	4.9559
Percentage Improvement (%)	26.3%	1.5%	50.7%	-21.7%
Filtered by (SARD) filter	0.5566	0.9803	0.8424	14.9776
Percentage Improvement (%)	-93.1%	1.5%	40.9%	136.7%
Filtered by proposed algo	21.1905	0.9853	0.8961	1.3923
Percentage Improvement (%)	163.4%	2%	49.8%	-77.96%
Image (Rice) with speckle noise of standard deviation 0.5	SNR	Coc	MSSIM	RMSE
Noisy Image	6.7807	0.9534	0.5062	7.3154
Filtered by (LEE) filter	10.3514	0.9762	0.776	4.8495
Percentage Improvement (%)	52.6%	2.3%	53.2%	33.8%
Filtered by (lsmv) filter	10.0388	0.9796	0.8966	5.0272
Percentage Improvement (%)	48%	2.7%	77.1%	-31.3%
Filtered by (SARD) filter	0.5566	0.9803	0.8424	14.9776
Percentage Improvement (%)	-91.8%	2.8%	66.4%	104.7%
Filtered by proposed algo	21.0994	0.9829	0.8736	1.407
Percentage Improvement (%)	211.1%	3.09%	72.5%	-80.8%
Image (Rice) with speckle noise of standard deviation 0.6	SNR	Coc	MSSIM	RMSE
Noisy Image	5.9362	0.9412	0.4313	8.0623
Filtered by (LEE) filter	9.2044	0.9719	0.7206	5.5341
Percentage Improvement (%)	55%	3.2%	67%	-31.4%
Filtered by (lsmv) filter	9.7818	0.9788	0.8911	5.1782
Percentage Improvement (%)	64.7%	3.9%	106.7%	-35.8%
Filtered by (SARD) filter	0.5566	0.9803	0.8424	14.9776
Percentage Improvement (%)	-90.7%	4.1%	96.1%	85.7%
Filtered by proposed algo	20.8922	0.9755	0.8276	1.441

Percentage Improvement (%)	251.9%	3.6%	91.8%	-82.2%
Image (Rice) with speckle noise of standard deviation 0.7	SNR	Coc	MSSIM	RMSE
Noisy Image	5.3641	0.9252	0.371	8.6113
Filtered by (LEE) filter	8.2838	0.9635	0.6683	6.1529
Percentage Improvement (%)	54.4%	4.1%	78.9%	-28.6%
Filtered by (lsmv) filter	9.6078	0.9768	0.8848	5.283
Percentage Improvement (%)	79.1%	5.5%	138.4%	-38.7%
Filtered by (SARD) filter	0.5566	0.9803	0.8424	14.9776
Percentage Improvement (%)	-89.7%	5.9%	127.06%	73.9%
Filtered by proposed algo	20.8922	0.9755	0.8276	1.441
Percentage Improvement (%)	289.4%	5.4%	123.07%	-83.3%

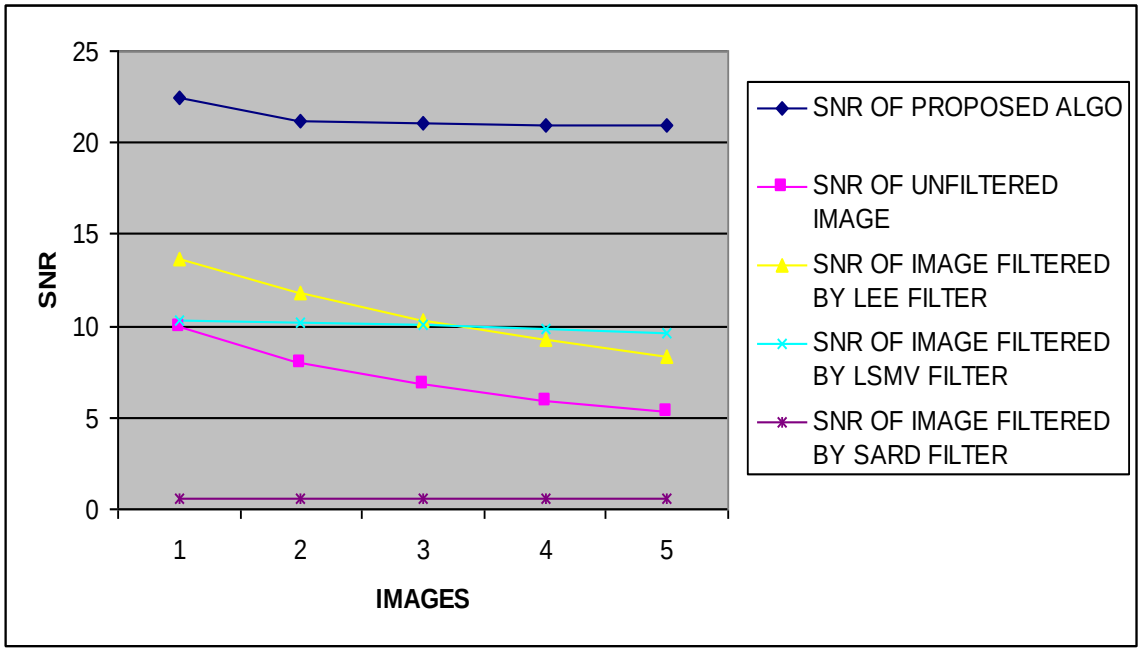
Table 6.7 Statistical results of image 4 (Rice) on parameters SNR, Coc, MSSIM, RMSE and Percentage Improvement.

Edge comparison by Edge preserving index (EPI)	Proposed Algo	SOBLE	PREWITT	CANNY
Image (Rice) _0.3	0.0579	0.0142	0.0143	0.001
Image (Rice) _0.4	0.0422	0.0124	0.0134	0.0012
Image (Rice) _0.5	0.0299	0.0134	0.0134	0.0024
Image (Rice) _0.6	0.0201	0.0105	0.0123	0.0024
Image (Rice) _0.7	0.0164	0.0103	0.0127	0.0038

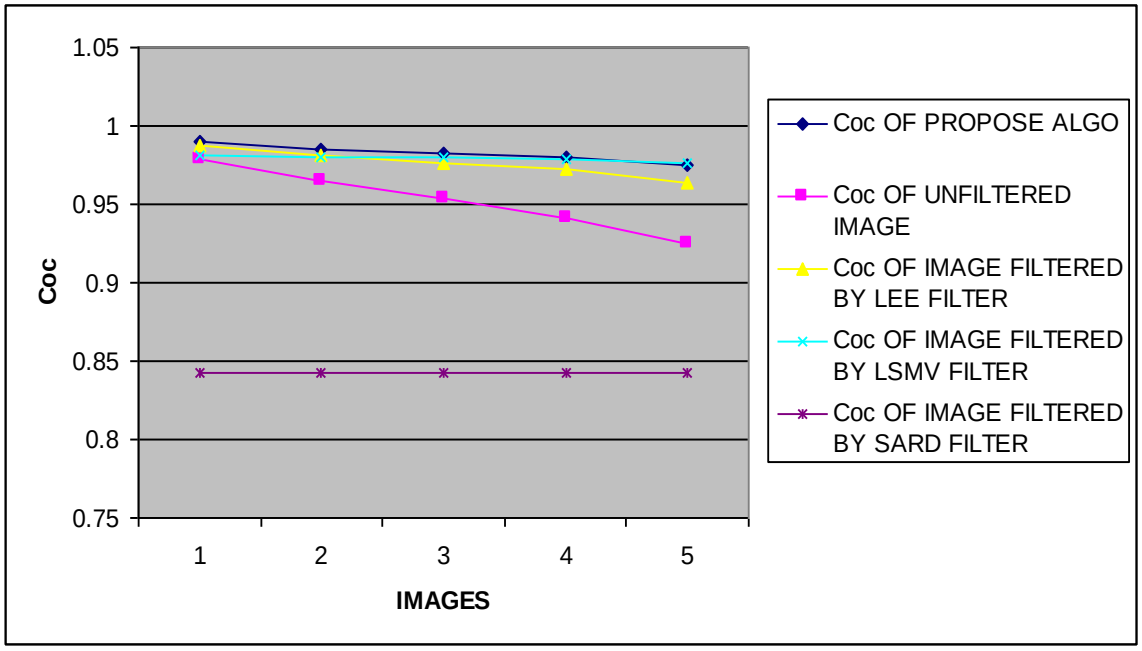
Table 6.8 Statistical results of image 4 (Rice) for the parameter EPI



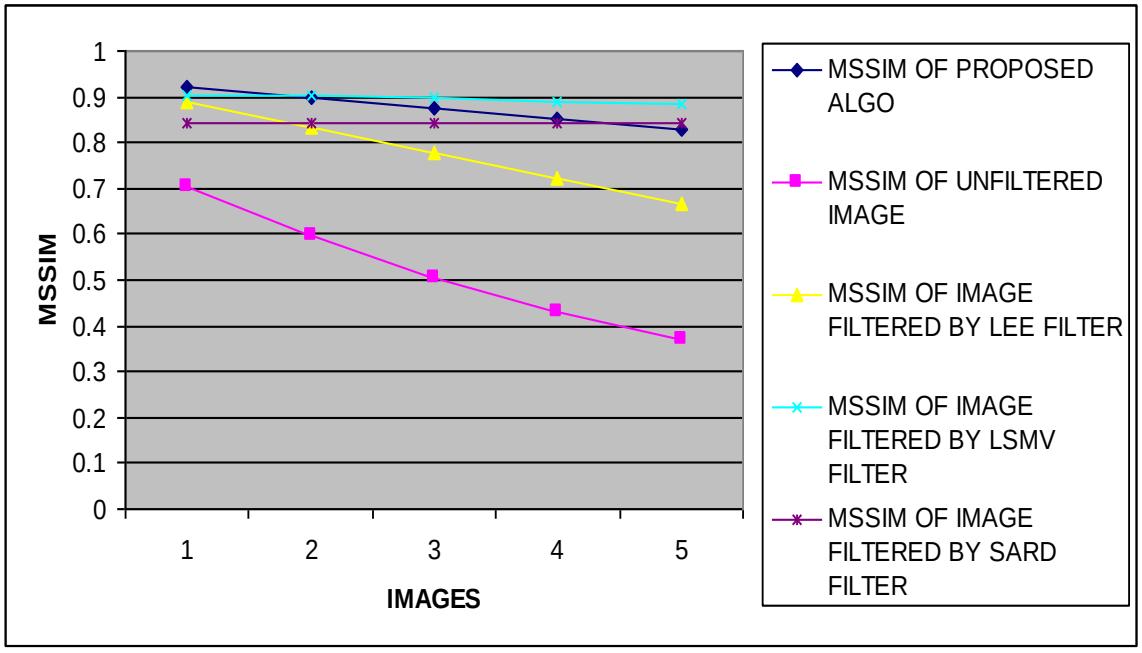
Graph 6.16 Representation of results of EPI parameter on image 4 (Rice)



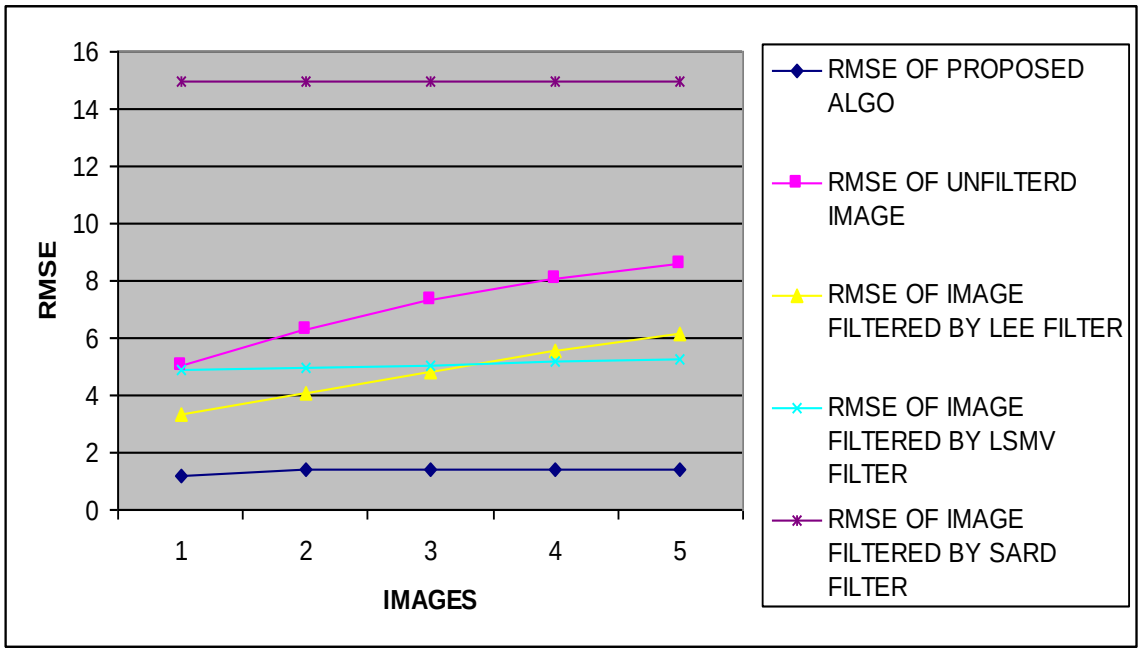
Graph 6.17 Representation of results of SNR parameter on image 4 (Rice)



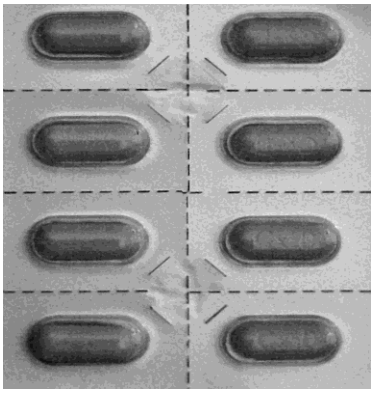
Graph 6.18 Representation of results of Coc parameter on image 4 (Rice)



Graph 6.19 Representation of results of MSSIM parameter on image 4 (Rice)



Graph 6.20 Representation of results of RMSE parameter on image 4 (Rice)



(a)

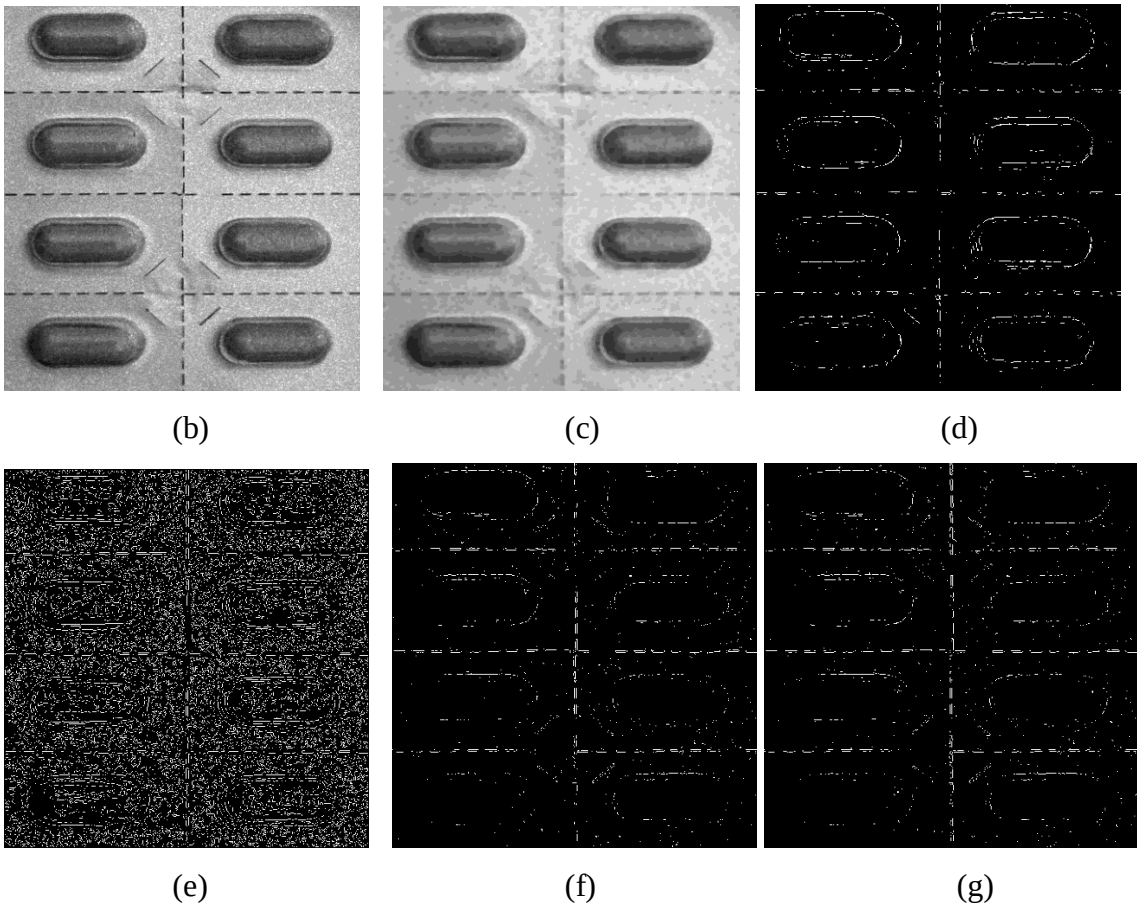


Fig 6.5 Different processes on image 5 (Pills) (a) Original Image (b) Image with Speckle Noise of std. dev. 0.4 (c) Filtered Image by Proposed Algorithm (d) Edges Obtained by Proposed Algorithm (e) Edges Obtained by Canny Operator (f) Edges Obtained by Sobel Operator (g) edges Obtained by Prewitt Operator.

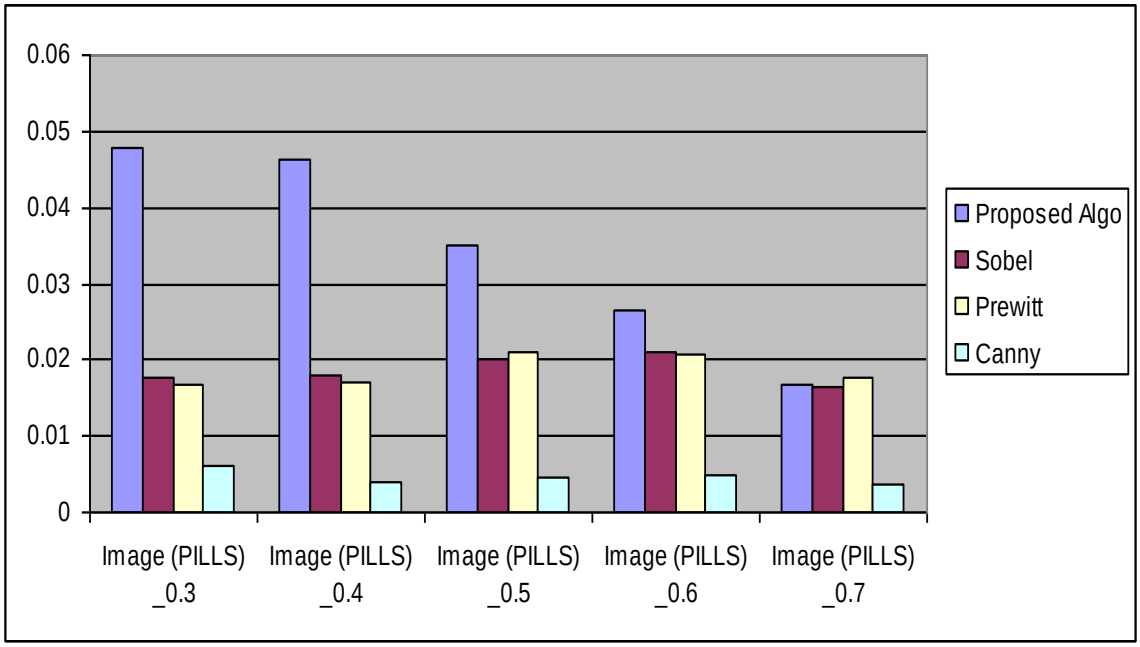
Image (Pills) with speckle noise of standard deviation 0.3	SNR	Coc	MSSIM	RMSE
Noisy Image	8.2107	0.9852	0.6287	6.2047
Filtered by (LEE) filter	11.4639	0.9902	0.8126	4.2663
Percentage Improvement (%)	39.6%	0.5%	29.2%	-31.3%
Filtered by (lsmv) filter	9.4189	0.9887	0.7821	5.3989
Percentage Improvement (%)	14.7%	0.3%	24.3%	-13%
Filtered by (SARD) filter	10.3022	0.9894	0.7573	4.8769
Percentage Improvement (%)	25.4%	0.4%	20.4%	-21.4%
Filtered by proposed algo	22.0018	0.9876	0.7896	1.2681
Percentage Improvement (%)	167.9%	0.2%	25.5%	-79.6%
Image (Pills) with speckle noise of				

standard deviation 0.4	SNR	Coc	MSSIM	RMSE
Noisy Image	6.6200	0.9779	0.5196	7.4517
Filtered by (LEE) filter	9.7807	0.987	0.7419	5.1787
Percentage Improvement (%)	47.7%	0.9%	42.7%	-30.2%
Filtered by (lsmv) filter	9.2544	0.9876	0.7771	5.5021
Percentage Improvement (%)	39.7%	0.99%	49.5%	-26.2%
Filtered by (SARD) filter	10.3738	0.9895	0.7577	4.8369
Percentage Improvement (%)	56.7%	1.1%	45.8%	-35.1%
Filtered by proposed algo	22.0141	0.9862	0.764	1.2663
Percentage Improvement (%)	232.5%	0.8%	47.0%	-83.1%
Image (Pills) with speckle noise of standard deviation 0.5	SNR	Coc	MSSIM	RMSE
Noisy Image	5.7177	0.9697	0.4374	8.2674
Filtered by (LEE) filter	8.4857	0.9839	0.6753	6.0113
Percentage Improvement (%)	48.4%	1.4%	54.3%	27.3%
Filtered by (lsmv) filter	9.0415	0.9862	0.7701	5.6387
Percentage Improvement (%)	58.1%	1.7%	76.0%	-31.8%
Filtered by (SARD) filter	10.4275	0.9893	0.7583	4.807
Percentage Improvement (%)	82.3%	2.0%	73.3%	-41.9%
Filtered by proposed algo	22.0061	0.9851	0.7371	1.2675
Percentage Improvement (%)	284.8%	1.5%	68.5%	-84.7%
Image (Pills) with speckle noise of standard deviation 0.6	SNR	Coc	MSSIM	RMSE
Noisy Image	5.1294	0.9569	0.3729	8.8467
Filtered by (LEE) filter	7.4903	0.9798	0.6138	6.7412
Percentage Improvement (%)	46.0%	2.3%	164.6%	76.2%
Filtered by (lsmv) filter	8.8118	0.985	0.7634	5.7898
Percentage Improvement (%)	71.7%	2.9%	204.7%	65.4%
Filtered by (SARD) filter	10.4616	0.9888	0.7575	4.7882
Percentage Improvement (%)	103.9%	3.3%	203.1%	54.1%
Filtered by proposed algo	21.8383	0.9821	0.7101	1.2922
Percentage Improvement (%)	325.7%	2.6%	190.4%	14.6%
Image (Pills) with speckle noise of standard deviation 0.7	SNR	Coc	MSSIM	RMSE
Noisy Image	4.7053	0.9370	0.3217	9.2894
Filtered by (LEE) filter	6.7238	0.9769	0.5609	7.3631
Percentage Improvement (%)	42.8%	4.2%	74.3%	-20.8%
Filtered by (lsmv) filter	8.4683	0.9834	0.754	6.0233
Percentage Improvement (%)	79.9%	4.9%	134.3%	-35.2%
Filtered by (SARD) filter	10.2667	0.9877	0.7524	4.8968
Percentage Improvement (%)	115.44%	5.4%	133.8%	-47.3%
Filtered by proposed algo	21.9731	0.9810	0.6826	1.2723
Percentage Improvement (%)	366.9%	4.6%	112.1%	-86.4%

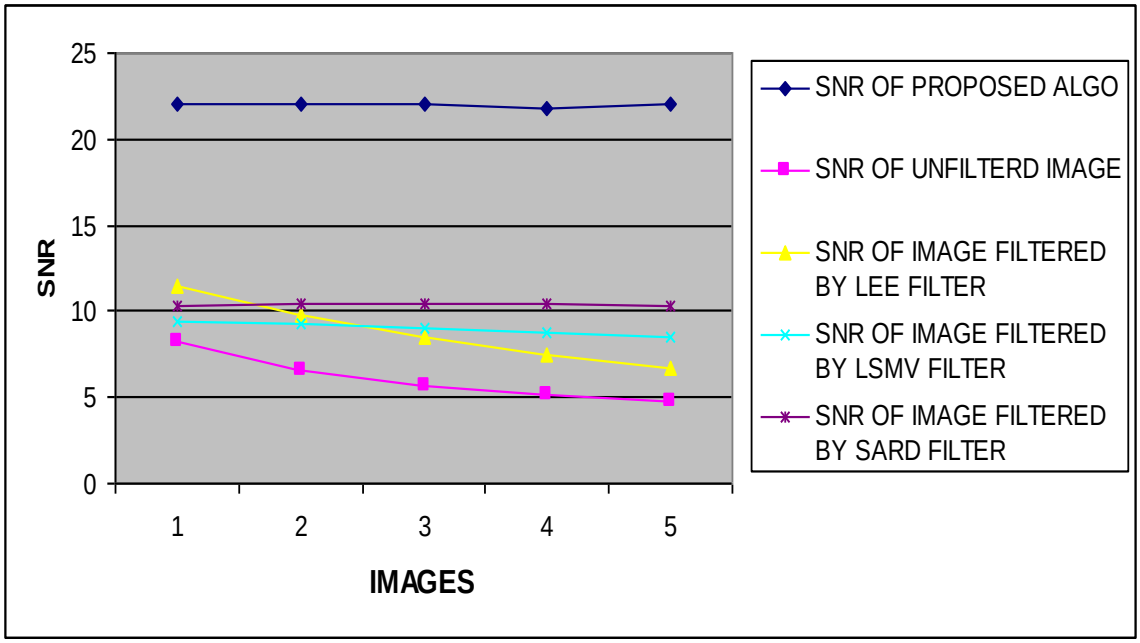
Table 6.9 Statistical results of image 5 (Pills) on parameters SNR, Coc, MSSIM, RMSE and Percentage Improvement.

Edge comparison by Edge preserving index (EPI)	Proposed Algo	SOBEL	PREWITT	CANNY
Image (Pills) _0.3	0.0479	0.0176	0.0168	0.0061
Image (Pills) _0.4	0.0462	0.0181	0.0172	0.004
Image (Pills) _0.5	0.035	0.0202	0.021	0.0046
Image (Pills) _0.6	0.0265	0.0209	0.0206	0.0049
Image (Pills) _0.7	0.0168	0.0163	0.0178	0.0037

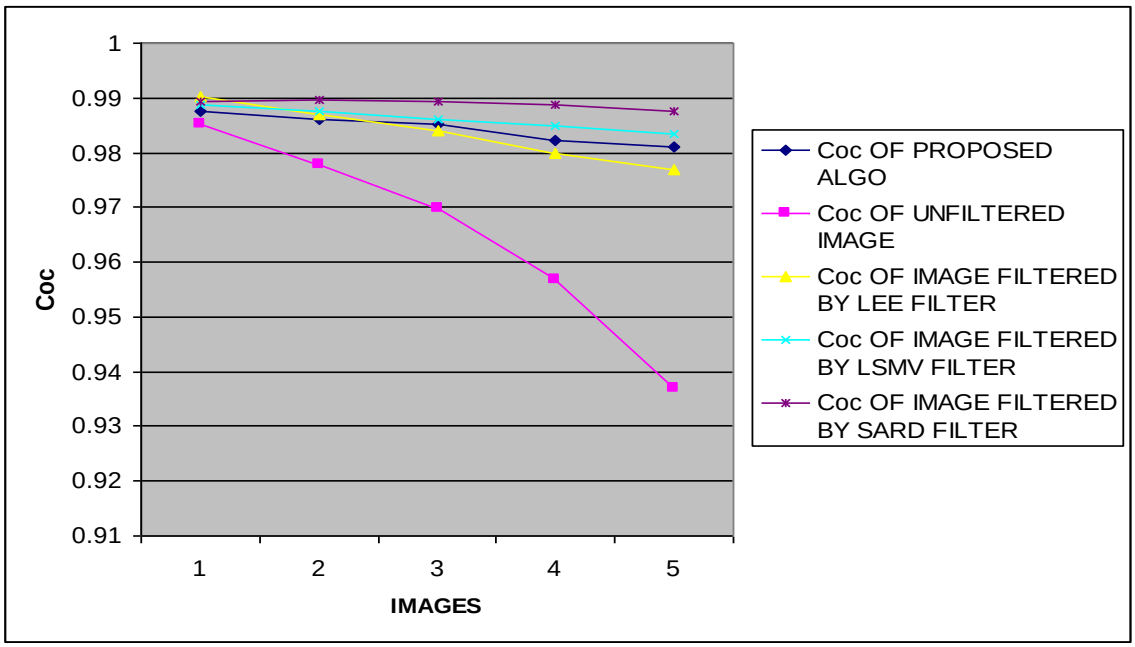
Table 6.10 Statistical results of image 5 (Pills) for the parameter EPI



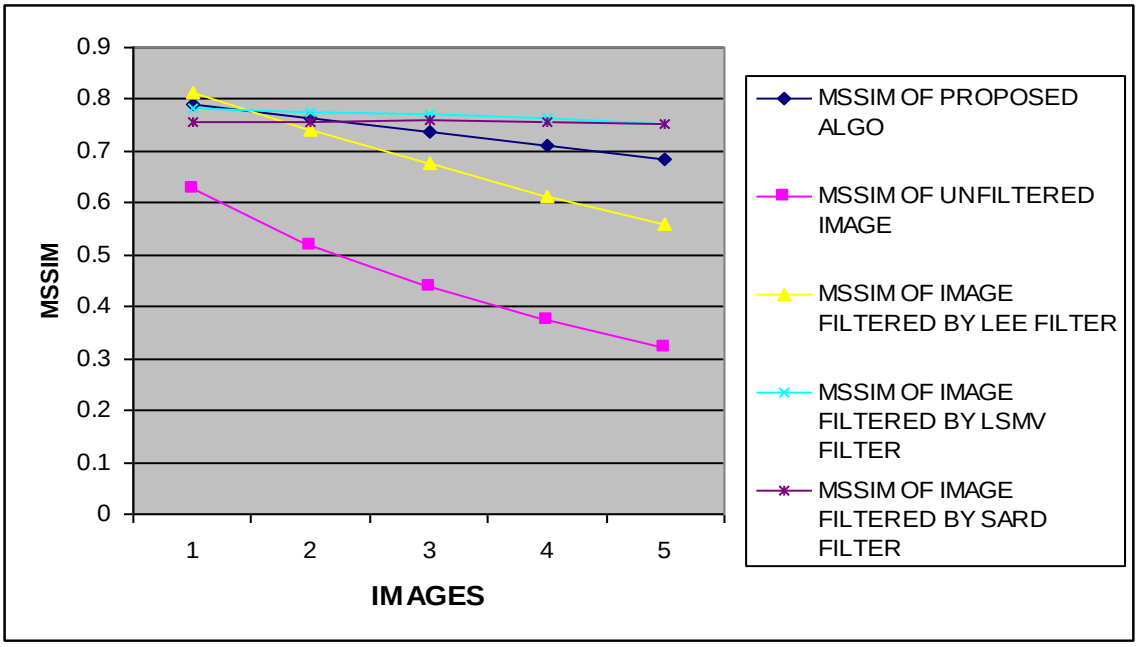
Graph 6.21 Representation of results of EPI parameter on image 5 (Pills)



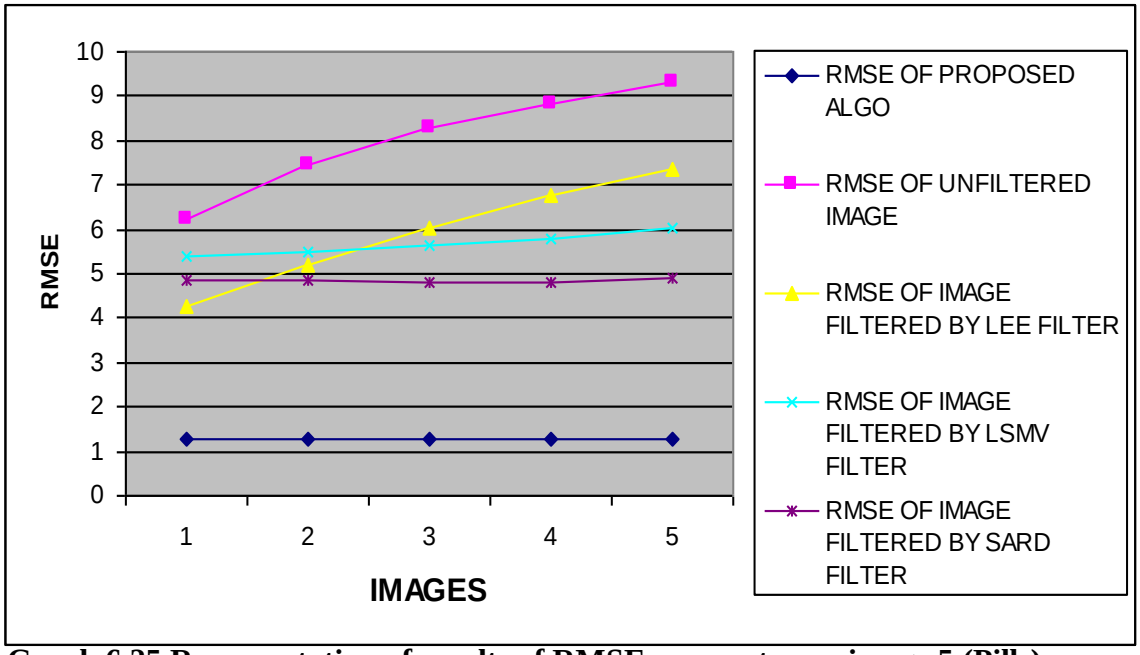
Graph 6.22 Representation of results of SNR parameter on image 5 (Pills)



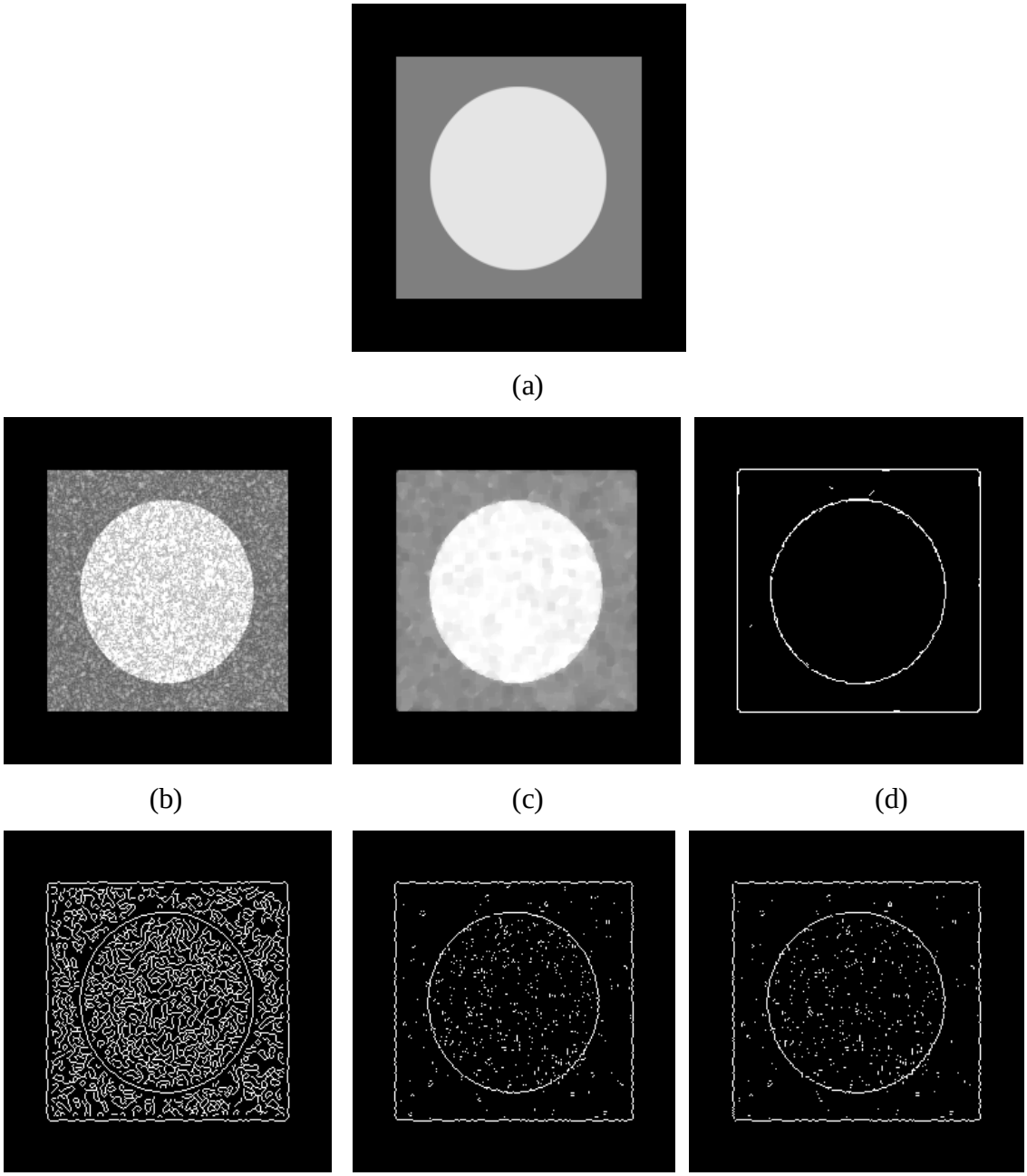
Graph 6.23 Representation of results of Coc parameter on image 5 (Pills)



Graph 6.24 Representation of results of MSSIM parameter on image 5 (Pills)



Graph 6.25 Representation of results of RMSE parameter on image 5 (Pills)



(a)

(b) (c) (d)

(e) (f) (g)

Fig 6.6 Different processes on image 6 (Pattern) (a) Original Image (b) Image with Speckle Noise of std. dev. 0.7 (c) Filtered Image by Proposed Algorithm (d) Edges Obtained by Proposed Algorithm (e) Edges Obtained by Canny Operator (f) Edges Obtained by Sobel Operator (g) edges Obtained by Prewitt Operator.

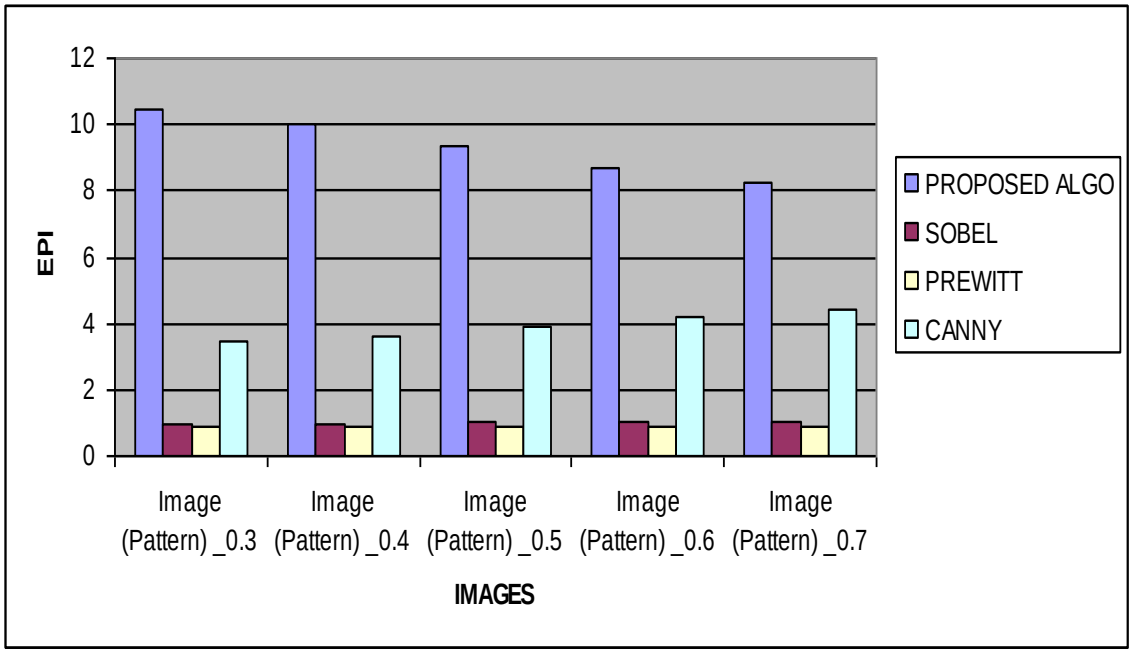
Image (Pattern) with speckle noise of standard deviation 0.3	SNR	Coc	MSSIM	RMSE
Noisy Image	7.5464	1	0.7221	4.7863
Filtered by (LEE) filter	10.4543	0.9909	0.8526	3.4245
Percentage Improvement (%)	38.5%	-0.09%	18.07%	-28.5%
Filtered by (lsmv) filter	8.8512	0.9898	0.9133	4.1187
Percentage Improvement (%)	17.2%	-1.02%	26.4%	-13.95%
Filtered by (SARD) filter	10.4558	0.9934	0.9048	3.424
Percentage Improvement (%)	38.5%	-0.66%	25.3%	-28.5%
Filtered by proposed algo	24.9051	0.9998	0.9643	0.6487
Percentage Improvement (%)	230.02%	-0.02%	33.5%	-86.5%
Image (Pattern) with speckle noise of standard deviation 0.4	SNR	Coc	MSSIM	RMSE
Noisy Image	6.0511	1	0.6677	5.6854
Filtered by (LEE) filter	8.6872	0.9917	0.8043	4.1972
Percentage Improvement (%)	43.5%	-0.83%	20.4%	26.2%
Filtered by (lsmv) filter	8.4875	0.9886	0.9088	4.2948
Percentage Improvement (%)	40.2%	-1.14%	36.1%	-24.5%
Filtered by (SARD) filter	9.9954	0.9938	0.8991	3.6103
Percentage Improvement (%)	65.1%	-0.62%	34.6%	-36.5%
Filtered by proposed algo	23.7257	0.9998	0.9495	0.7431
Percentage Improvement (%)	292.08%	-0.02.%	42.2%	-86.93%
Image (Pattern) with speckle noise of standard deviation 0.5	SNR	Coc	MSSIM	RMSE
Noisy Image	5.1349	1	0.6285	6.3179
Filtered by (LEE) filter	7.3212	0.9923	0.7633	4.912
Percentage Improvement (%)	42.5%	-0.77%	21.4%	-22.3%
Filtered by (lsmv) filter	8.0002	0.9882	0.9042	4.5426
Percentage Improvement (%)	55.8%	-1.18%	43.8%	-28.1%
Filtered by (SARD) filter	9.3703	0.9956	0.8926	3.8798
Percentage Improvement (%)	82.4%	-0.44%	42.02%	-38.6%
Filtered by proposed algo	21.8444	0.9998	0.9348	0.9228
Percentage Improvement (%)	325.4%	-0.02%	48.7%	-81.4%
Image (Pattern) with speckle noise of standard deviation 0.6	SNR	Coc	MSSIM	RMSE
Noisy Image	4.6614	1	0.6021	6.6719
Filtered by (LEE) filter	6.3934	0.9933	0.7339	5.4658
Percentage Improvement (%)	37.1%	-0.67%	21.8%	-18.1%
Filtered by (lsmv) filter	7.286	0.9906	0.9033	4.932
Percentage Improvement (%)	56.3%	-0.94%	50.02%	-26.1%
Filtered by (SARD) filter	8.6647	0.9965	0.8912	4.2081
Percentage Improvement (%)	85.8%	-0.35%	48.01%	-36.93%
Filtered by proposed algo	21.8299	0.9999	0.9316	0.9243
Percentage Improvement (%)	368.3%	-0.01%	54.7%	-86.2%

Image (Pattern) with speckle noise of standard deviation 0.7	SNR	Coc	MSSIM	RMSE
Noisy Image	4.3685	1	0.5838	6.9007
Filtered by (LEE) filter	5.7797	0.9928	0.7053	5.8659
Percentage Improvement (%)	24.6%	-0.72%	20.8%	-15%
Filtered by (lsmv) filter	6.86	0.9898	0.8965	5.1798
Percentage Improvement (%)	57.03%	-1.02%	53.5%	-24.94%
Filtered by (SARD) filter	8.245	0.9962	0.8821	4.4164
Percentage Improvement (%)	88.5%	-0.38%	51.09%	-36.1%
Filtered by proposed algo	21.5271	0.9999	0.9215	0.9571
Percentage Improvement (%)	392.8%	-0.01%	57.8%	-86.2%

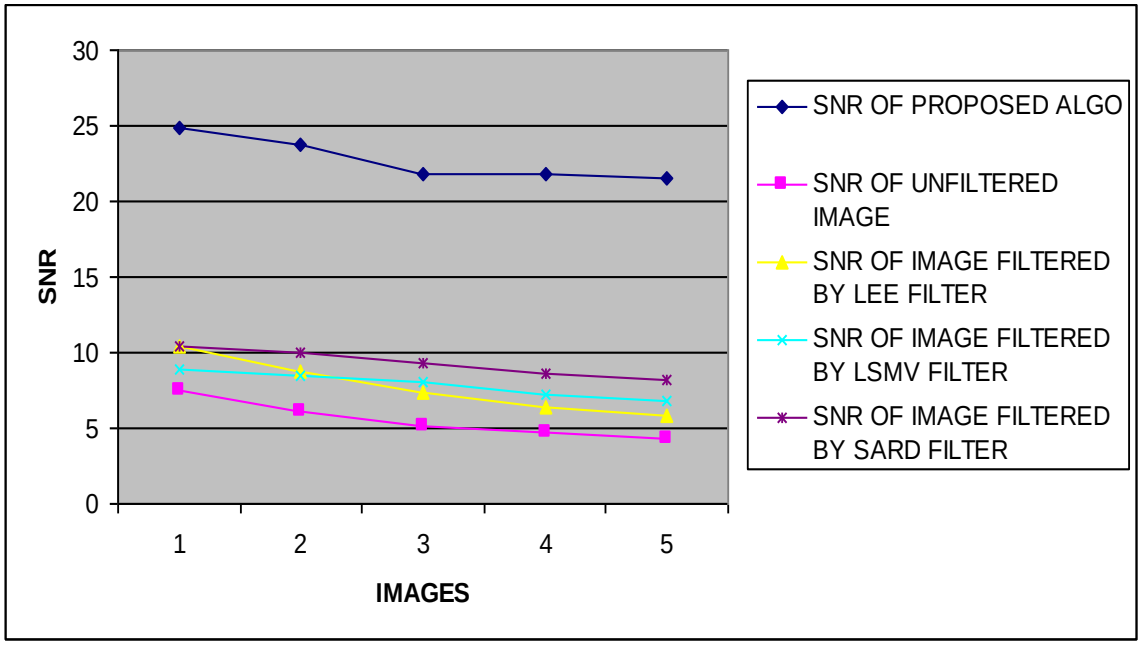
Table 6.11 Statistical results of image 6 (Pattern) on parameters SNR, Coc, MSSIM, RMSE and Percentage Improvement.

Edge comparison by Edge preserving index (EPI)	Proposed Algo	SOBLE	PREWITT	CANNY
Image (Pattern) _0.3	0.7412	0.151	0.1497	0.0493
Image (Pattern) _0.4	0.7316	0.0052	0.014	0.0029
Image (Pattern) _0.5	0.7043	0.0773	0.0626	0.0622
Image (Pattern) _0.6	0.702	0.0161	0.0277	0.0108
Image (Pattern) _0.7	0.6466	0.0118	0.0054	0.0307

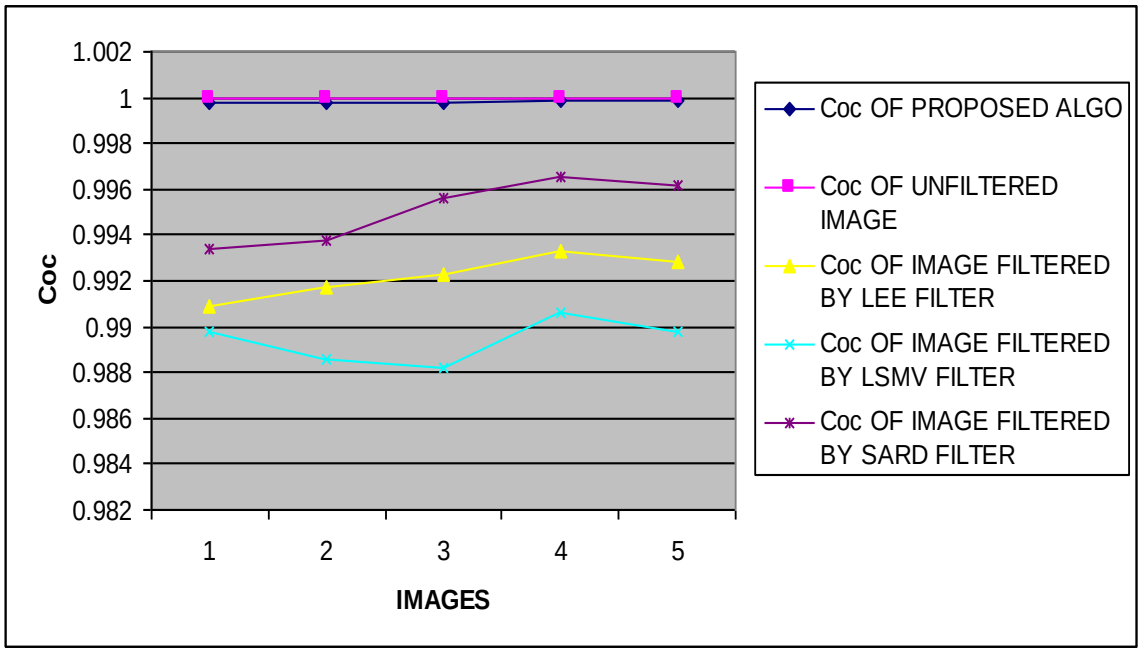
Table 6.12 Statistical results of image 6 (Pattern) for the parameter EPI



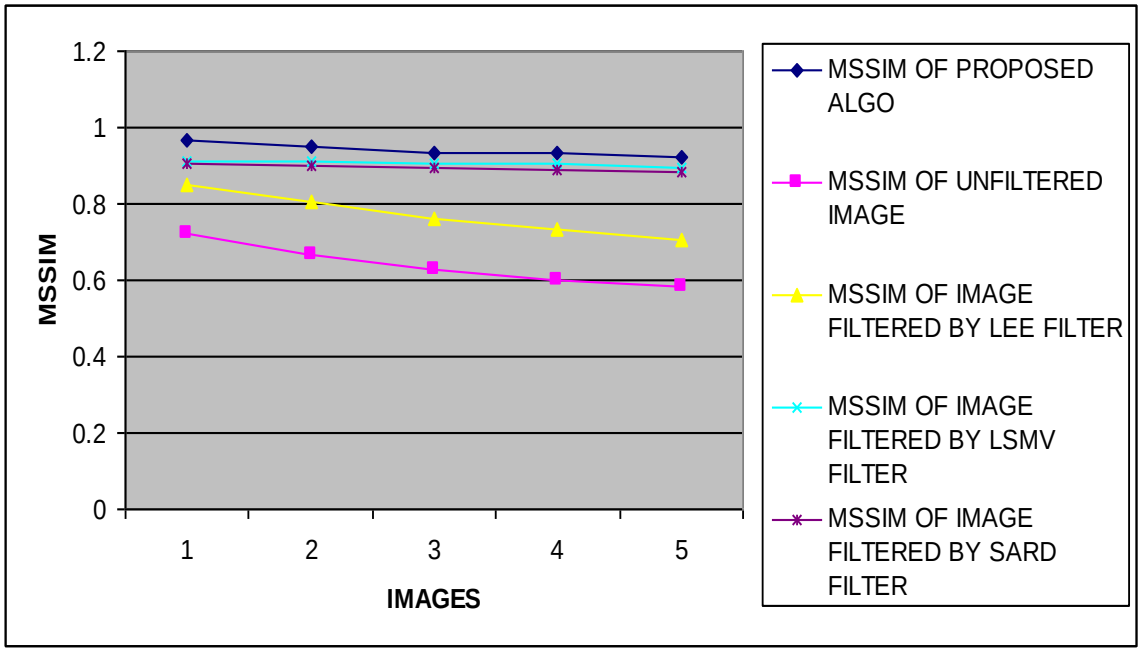
Graph 6.26 Representation of results of EPI parameter on image 6(Pattern)



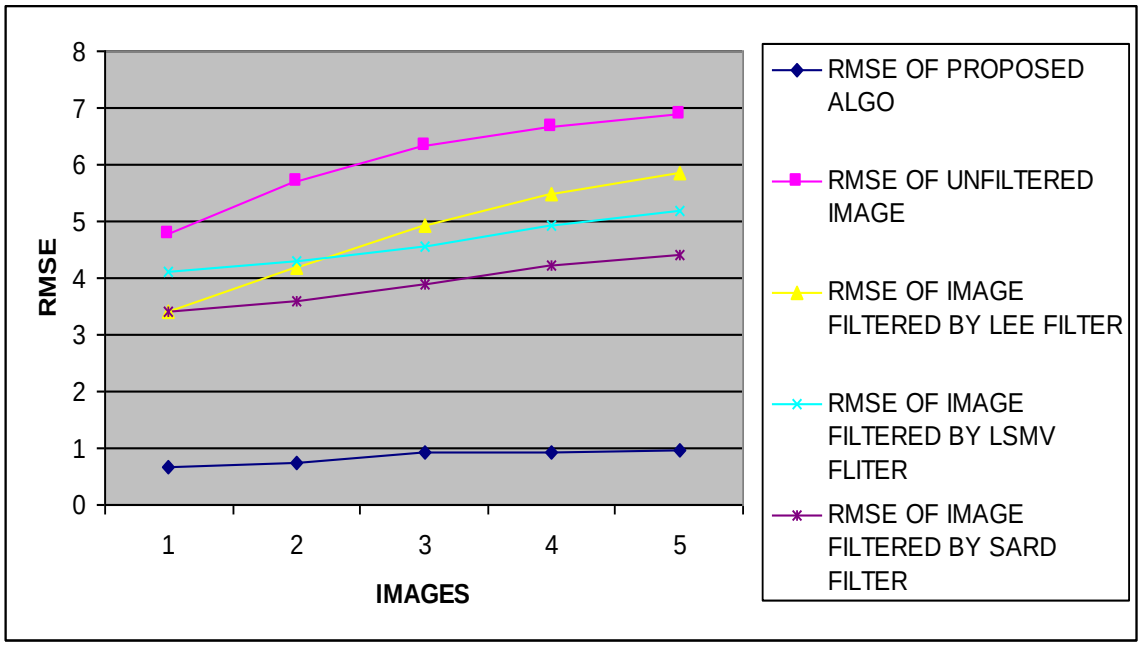
Graph 6.27 Representation of results of SNR parameter on image 6(Pattern)



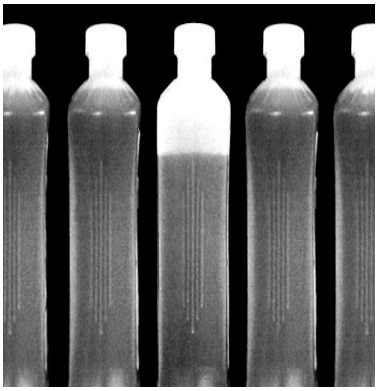
Graph 6.28 Representation of results of Coc parameter on image 6(Pattern)



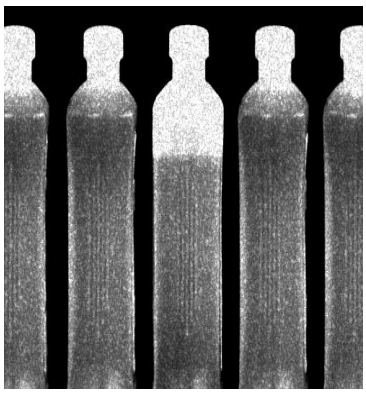
Graph 6.29 Representation of results of MSSIM parameter on image 6(Pattern)



Graph 6.30 Representation of results of RMSE parameter on image 6(Pattern)

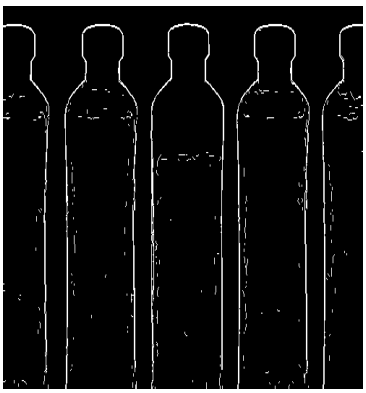


(a)



(b)

(c)



(d)

(e) (f) (g)

Fig 6.7 Different processes on image 7 (Bottles) (a) Original Image (b) Image with Speckle Noise of std. dev. 0.7 (c) Filtered Image by Proposed Algorithm (d) Edges Obtained by Proposed Algorithm (e) Edges Obtained by Canny Operator (f) Edges Obtained by Sobel Operator (g) edges Obtained by Prewitt Operator.

Image (Bottles) with speckle noise of standard deviation 0.3	SNR	Coc	MSSIM	RMSE
Noisy Image	9.3706	0.9896	0.8669	4.5499
Filtered by (LEE) filter	8.8855	0.984	0.8322	4.8113
Percentage Improvement (%)	-5.2%	-0.56%	-4.1%	5.7%
Filtered by (lsmv) filter	6.2019	0.9694	0.7286	6.553
Percentage Improvement (%)	-33.9%	-2.1%	-15.96%	44.02%
Filtered by (SARD) filter	7.4483	0.9767	0.712	5.6771
Percentage Improvement (%)	-20.6%	-1.4%	-17.9%	24.7%
Filtered by proposed algo	15.4467	0.9803	0.8075	2.2605
Percentage Improvement (%)	64.8%	-0.94%	-6.9%	50.4%
Image (Bottles) with speckle noise of standard deviation 0.4	SNR	Coc	MSSIM	RMSE
Noisy Image	7.632	0.983	0.8073	5.5582
Filtered by (LEE) filter	7.8997	0.982	0.8139	5.3896
Percentage Improvement (%)	3.5%	-0.2%	0.8%	-3.1%
Filtered by (lsmv) filter	5.905	0.9694	0.7277	6.7809
Percentage Improvement (%)	-22.7%	-1.4%	-9.9%	21.9%
Filtered by (SARD) filter	7.0264	0.9758	0.7101	5.9596
Percentage Improvement (%)	-7.3%	-0.8%	-12.1%	7.2%
Filtered by proposed algo	15.4703	0.9798	0.8018	2.2544
Percentage Improvement (%)	102.7%	-0.4%	-0.7%	-59.5%
Image (Bottles) with speckle noise				

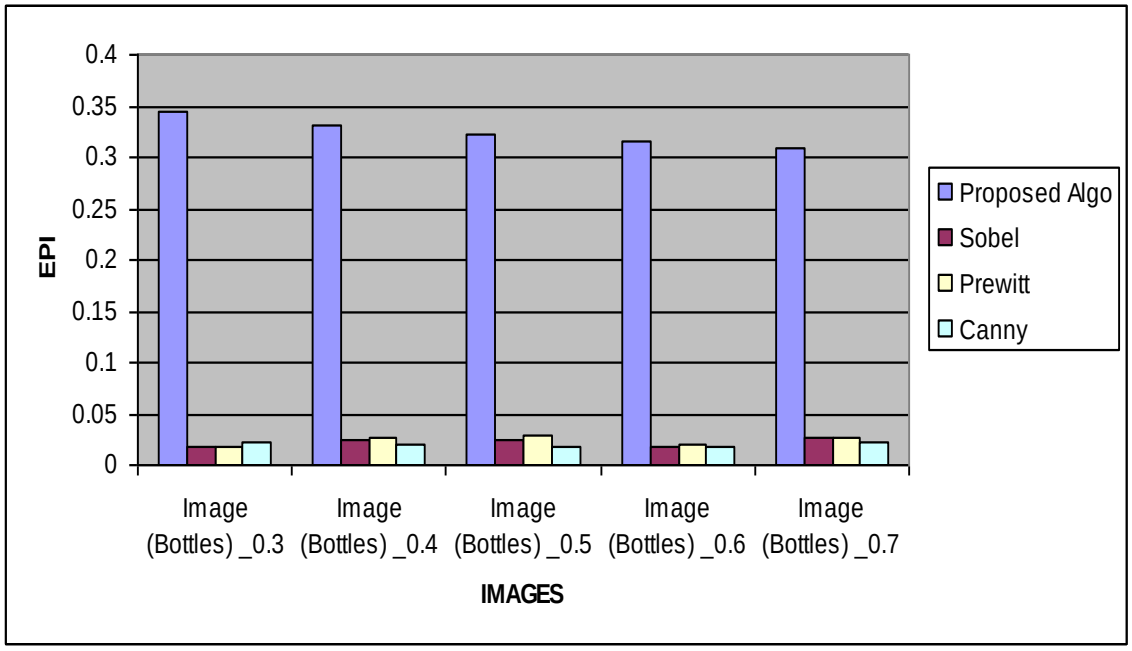
of standard deviation 0.5	SNR	Coc	MSSIM	RMSE
Noisy Image	6.476	0.9764	0.7502	6.3495
Filtered by (LEE) filter	7.1026	0.9794	0.7878	5.9076
Percentage Improvement (%)	9.6%	0.3%	5.01%	-6.96%
Filtered by (lsmv) filter	5.5969	0.9696	0.7267	7.0258
Percentage Improvement (%)	-13.6%	-0.7%	-3.2%	10.6%
Filtered by (SARD) filter	6.6518	0.9756	0.7082	6.2223
Percentage Improvement (%)	2.7%	-0.09%	-5.6%	-2.1%
Filtered by proposed algo	15.4059	0.9781	0.7932	2.2711
Percentage Improvement (%)	137.8%	0.1%	5.7%	-64.3%
Image (Bottles) with speckle noise of standard deviation 0.6	SNR	Coc	MSSIM	RMSE
Noisy Image	5.8034	0.9694	0.7028	6.8607
Filtered by (LEE) filter	6.5338	0.9774	0.7666	6.3073
Percentage Improvement (%)	12.5%	0.8%	9.07%	-8.1%
Filtered by (lsmv) filter	5.294	0.9692	0.7246	7.2751
Percentage Improvement (%)	-8.8%	-0.1%	3.1%	6.04%
Filtered by (SARD) filter	6.2605	0.9754	0.7048	6.509
Percentage Improvement (%)	7.8%	0.62%	0.2%	-5.2%
Filtered by proposed algo	15.6052	0.9761	0.7858	2.2196
Percentage Improvement (%)	168.8%	0.69%	11.8%	-67.7%
Image (Bottles) with speckle noise of standard deviation 0.7	SNR	Coc	MSSIM	RMSE
Noisy Image	5.2318	0.9617	0.6566	7.3274
Filtered by (LEE) filter	6.0145	0.9769	0.7427	6.696
Percentage Improvement (%)	14.9%	1.5%	13.1%	-8.7%
Filtered by (lsmv) filter	5.0762	0.9688	0.7233	7.4598
Percentage Improvement (%)	-2.98%	0.7%	10.1%	1.8%
Filtered by (SARD) filter	5.9724	0.976	0.7026	6.7285
Percentage Improvement (%)	14.1%	1.4%	7%	-8.2%
Filtered by proposed algo	15.809	0.9768	0.7783	2.1681
Percentage Improvement (%)	202.1%	1.5%	18.5%	-70.5%

Table 6.13 Statistical results of image 7 (Bottles) on parameters SNR, Coc, MSSIM, RMSE and Percentage Improvement.

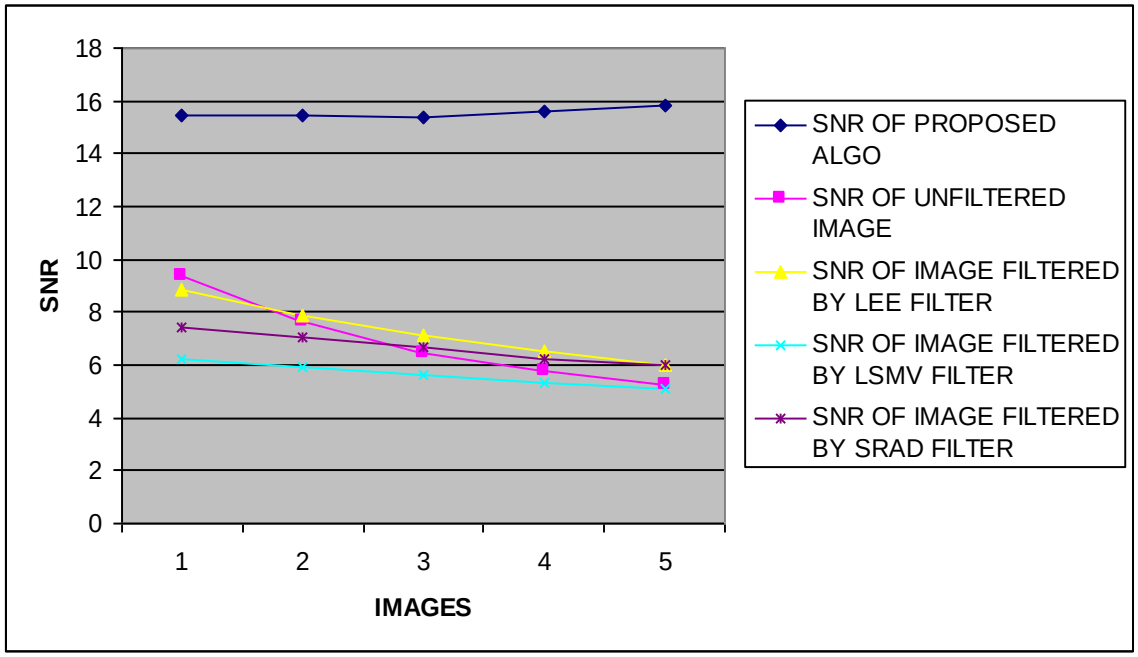
Edge comparison by Edge preserving index (EPI)	Proposed Algo	SOBLE	PREWITT	CANNY
Image (Bottles) _0.3	0.344	0.0173	0.0188	0.0224
Image (Bottles) _0.4	0.3319	0.0239	0.0261	0.0204

Image (Bottles) _0.5	0.3229	0.0252	0.0285	0.0177
Image (Bottles) _0.6	0.3166	0.0187	0.0191	0.0187
Image (Bottles) _0.7	0.3087	0.0264	0.0263	0.023

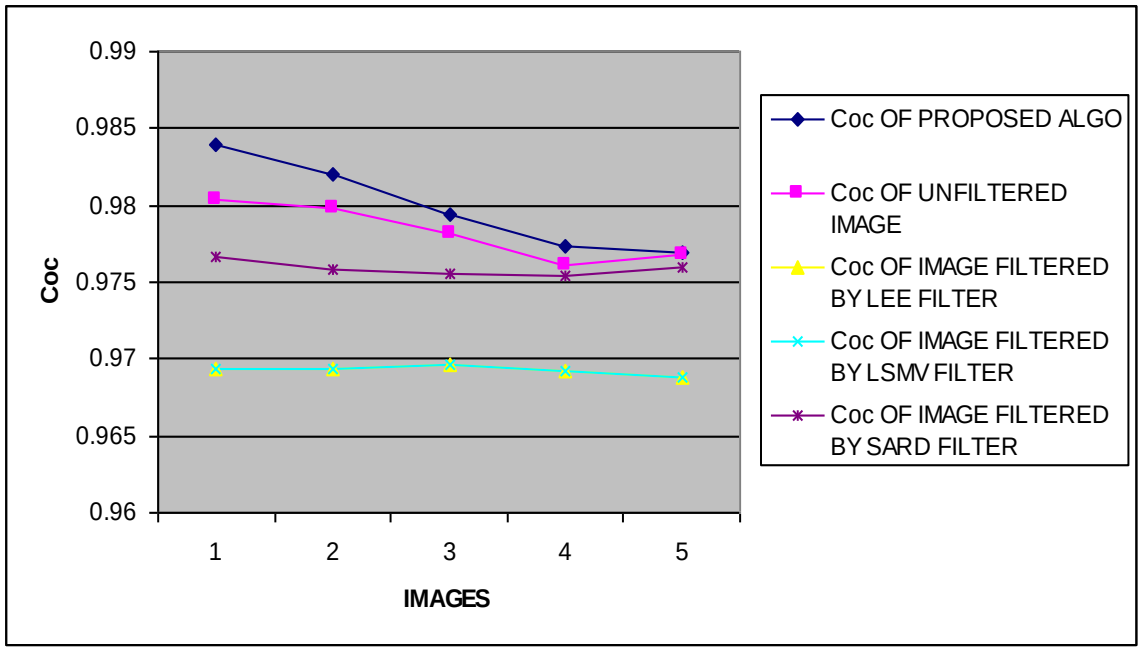
Table 6.14 Statistical results of image 7 (Bottles) for the parameter EPI



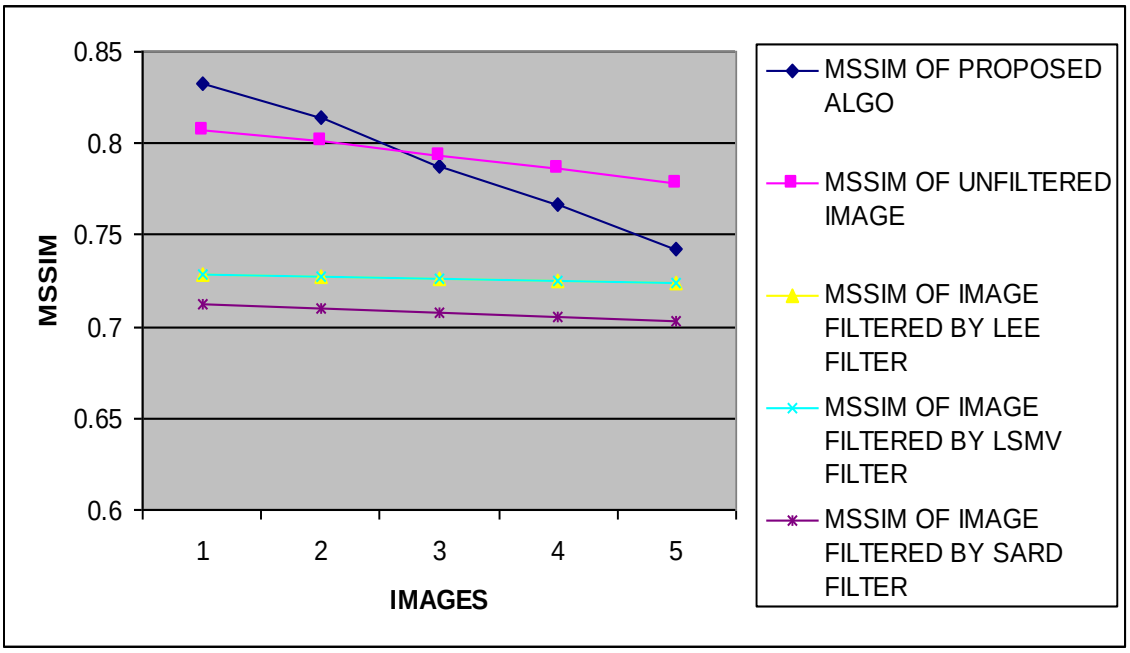
Graph 6.31 Representation of results of EPI parameter on image 7(Bottles)



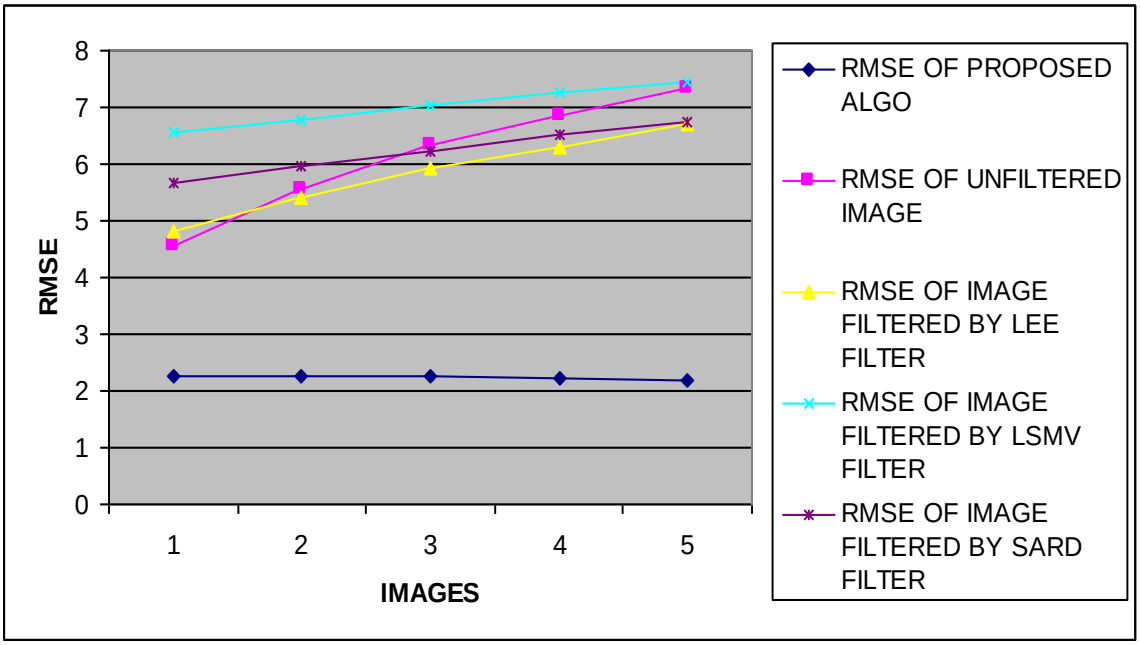
Graph 6.32 Representation of results of SNR parameter on image 7(Bottles)



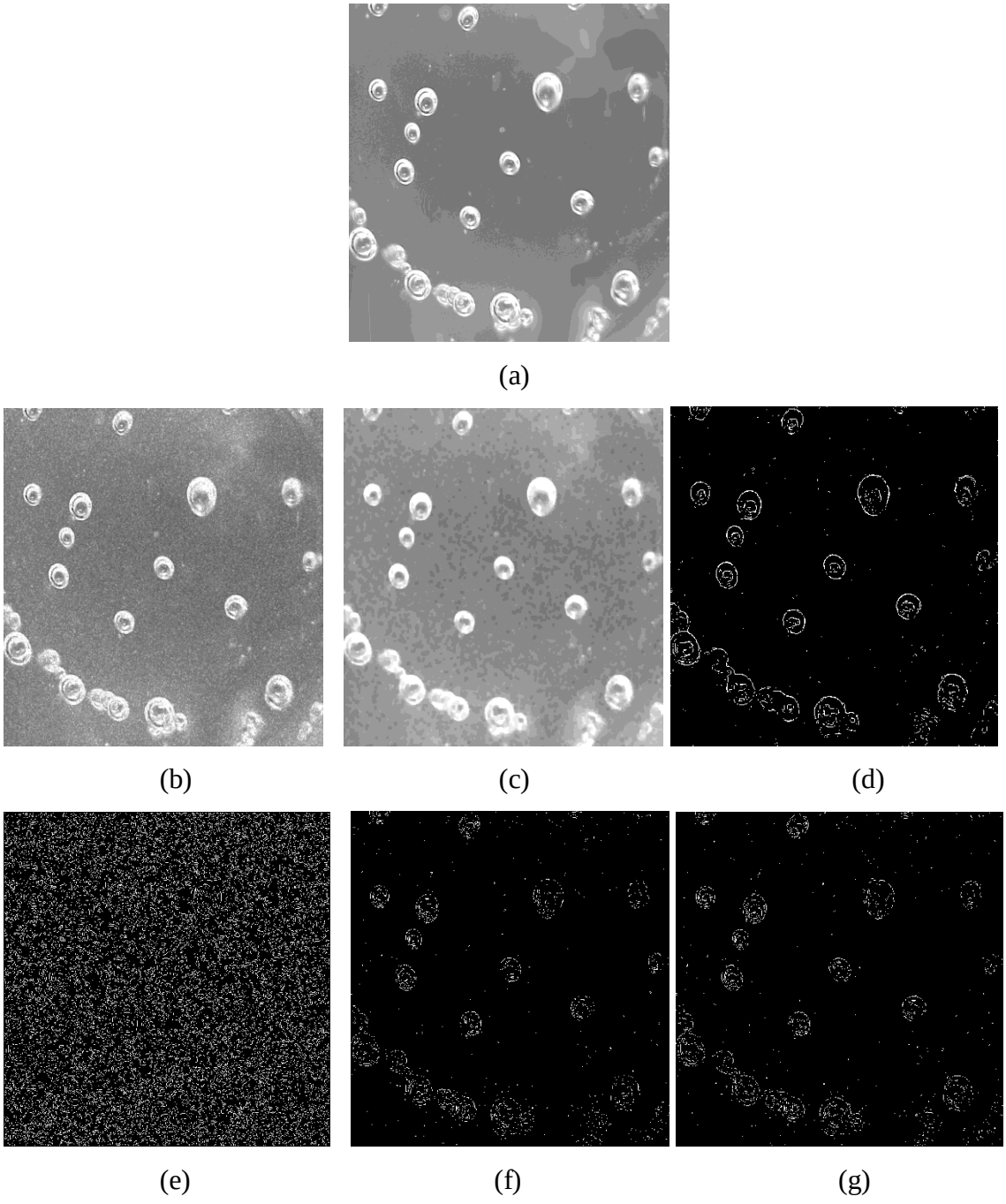
Graph 6.33 Representation of results of Coc parameter on image 7(Bottles)



Graph 6.34 Representation of results of MSSIM parameter on image 7(Bottles)



Graph 6.35 Representation of results of RMSE parameter on image 7(Bottles)



(a)

(b) (c) (d)

(e) (f) (g)

Fig 6.8 Different processes on image 8 (Bubbles) (a) Original Image (b) Image with Speckle Noise of std. dev. 0.5 (c) Filtered Image by Proposed Algorithm (d) Edges Obtained by Proposed Algorithm (e) Edges Obtained by Canny Operator (f) Edges Obtained by Sobel Operator (g) edges Obtained by Prewitt Operator

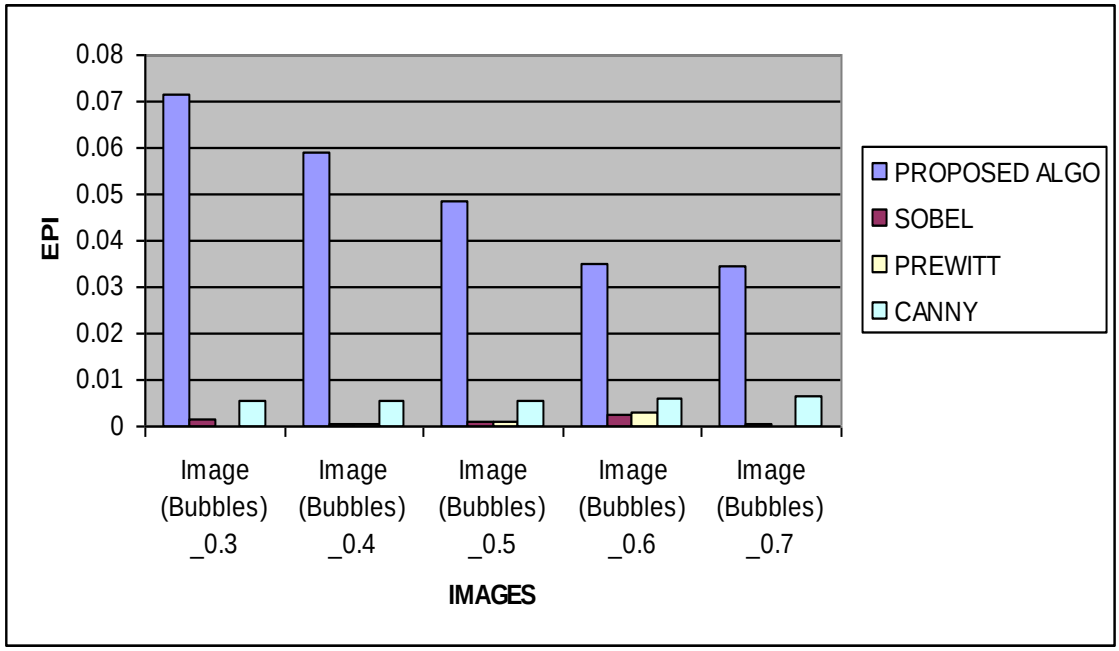
Image (Bubbles) with speckle noise of standard deviation 0.3	SNR	Coc	MSSIM	RMSE
Noisy Image	8.7814	0.896	0.564	5.8103
Filtered by (LEE) filter	12.3123	0.954	0.8151	3.8695
Percentage Improvement (%)	40.2%	6.4%	44.5%	-33.5%
Filtered by (lsmv) filter	12.4609	0.9754	0.9079	3.8038
Percentage Improvement (%)	41.9%	8.8%	60.9%	-34.6%
Filtered by (SARD) filter	11.813	0.9816	0.9011	4.0985
Percentage Improvement (%)	34.5%	9.5%	59.7%	-29.5%
Filtered by proposed algo	20.943	0.9758	0.8929	1.4326
Percentage Improvement (%)	138.4%	8.9%	58.3%	-75.4%
Image (Bubbles) with speckle noise of standard deviation 0.4	SNR	Coc	MSSIM	RMSE
Noisy Image	6.8399	0.8447	0.4369	7.2656
Filtered by (LEE) filter	10.6018	0.9299	0.7334	4.7117
Percentage Improvement (%)	54.9%	10.0%	67.8%	-35.2%
Filtered by (lsmv) filter	12.0613	0.9708	0.9005	3.9829
Percentage Improvement (%)	76.3%	14.9%	106.1%	-45.2%
Filtered by (SARD) filter	12.0902	0.9794	0.8996	3.9697
Percentage Improvement (%)	76.7%	15.9%	105.9%	-45.4%
Filtered by proposed algo	20.6799	0.968	0.8665	1.4766
Percentage Improvement (%)	202.3%	14.5%	98.3%	-79.7%
Image (Bubbles) with speckle noise of standard deviation 0.5	SNR	Coc	MSSIM	RMSE
Noisy Image	5.7263	0.7991	0.3426	8.2595
Filtered by (LEE) filter	9.1196	0.9037	0.6513	5.5884
Percentage Improvement (%)	59.2%	13.0%	90.1%	-32.4%
Filtered by (lsmv) filter	11.5829	0.9671	0.8902	4.2085
Percentage Improvement (%)	100.9%	21.0%	160.5%	-49.1%
Filtered by (SARD) filter	11.3396	0.9785	0.8927	4.328
Percentage Improvement (%)	98.0%	22.4%	160.5%	-47.6%
Filtered by proposed algo	20.9297	0.9574	0.8365	1.4348
Percentage Improvement (%)	165.5%	19.8%	144.1%	-82.7%
Image (Bubbles) with speckle noise of standard deviation 0.6	SNR	Coc	MSSIM	RMSE
Noisy Image	5.0825	0.7616	0.2765	8.895
Filtered by (LEE) filter	7.9699	0.8756	0.5795	6.3793
Percentage Improvement (%)	56.8%	-13.1%	109.5%	-28.3%
Filtered by (lsmv) filter	11.1234	0.959	0.8768	4.4371
Percentage Improvement (%)	118.8%	25.9%	217.1%	-50.2%
Filtered by (SARD) filter	11.276	0.9665	0.8663	4.3598
Percentage Improvement (%)	121.8%	26.9%	213.3%	51%
Filtered by proposed algo	20.8677	0.9481	0.8055	1.4451
Percentage Improvement (%)	310.5%	24.4%	191.3%	83.8%

Image (Bubbles) with speckle noise of standard deviation 0.7	SNR	Coc	MSSIM	RMSE
Noisy Image	4.646	0.7298	0.2266	9.3533
Filtered by (LEE) filter	7.0114	0.8496	0.5118	7.1236
Percentage Improvement (%)	50.9%	16.4%	125.8%	-23.8%
Filtered by (lsmv) filter	10.5739	0.9525	0.858	4.7269
Percentage Improvement (%)	127.5%	30.5%	278.6%	-49.5%
Filtered by (SARD) filter	10.6025	0.9502	0.8166	4.7113
Percentage Improvement (%)	128.2%	30.2%	260.3%	-49.7%
Filtered by proposed algo	21.1078	0.9347	0.7727	1.4057
Percentage Improvement (%)	354.3%	28.0%	240.9%	-85%

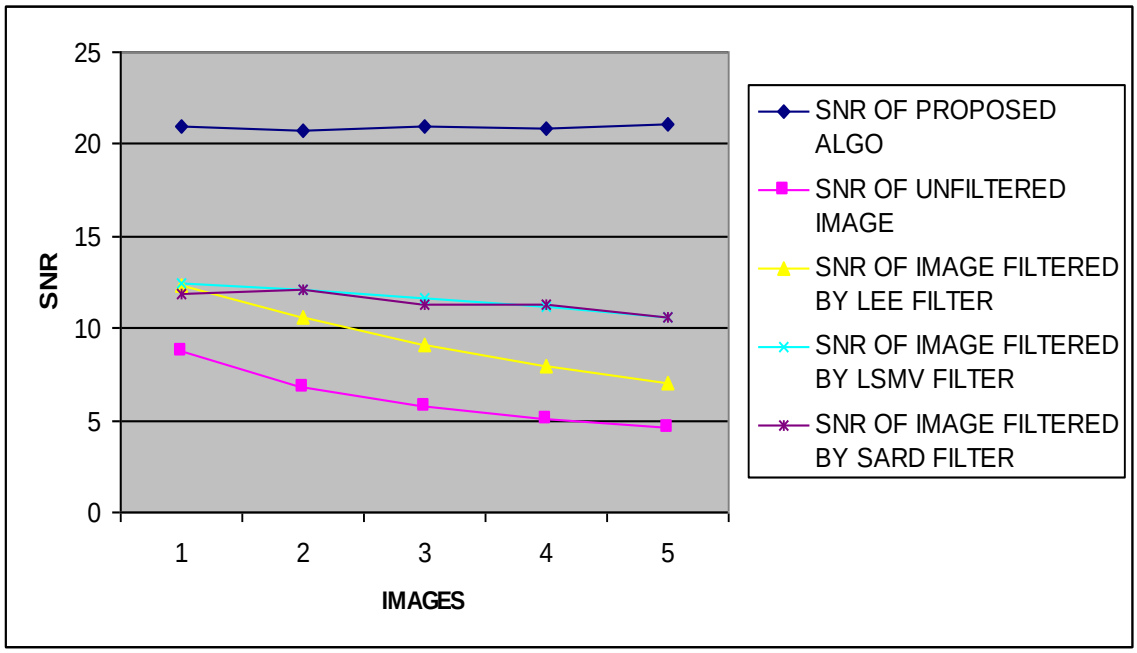
Table 6.15 Statistical results of image 8 (Bubbles) on parameters SNR, Coc, MSSIM, RMSE and Percentage Improvement.

Edge comparison by Edge preserving index (EPI)	Proposed Algo	SOBLE	PREWITT	CANNY
Image (Bubbles) _0.3	0.0713	0.0015	0.0001	0.0053
Image (Bubbles) _0.4	0.0591	0.0004	0.0006	0.0055
Image (Bubbles) _0.5	0.0487	0.0008	0.0009	0.0055
Image (Bubbles) _0.6	0.0349	0.0023	0.0031	0.0058
Image (Bubbles) _0.7	0.0347	0.0005	0.00005	0.0064

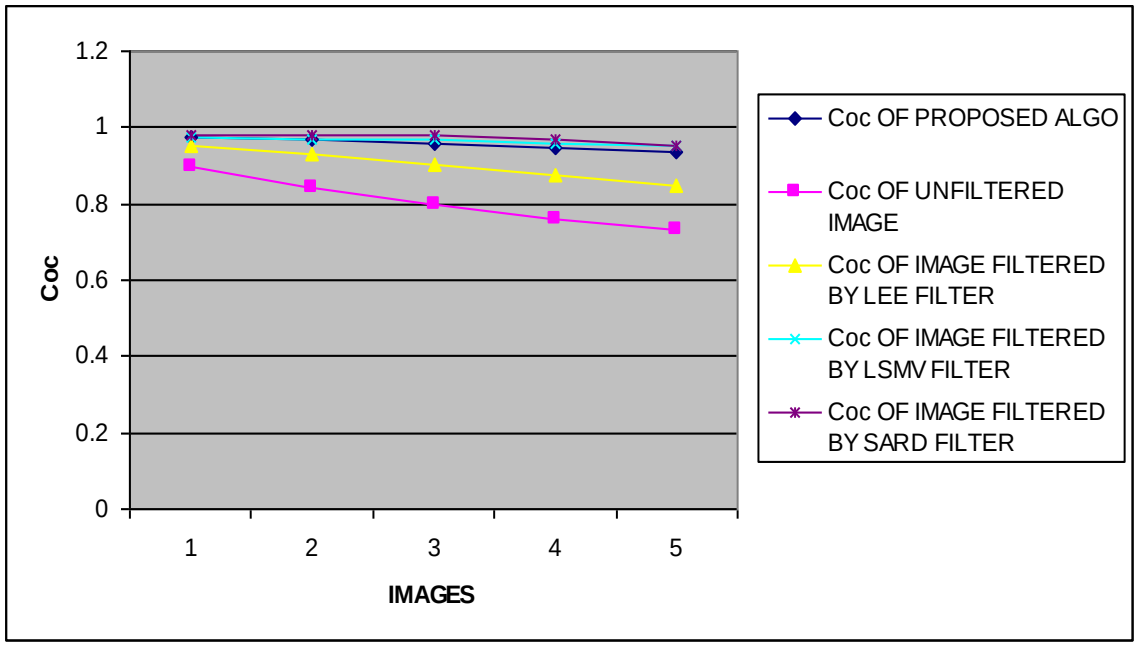
Table 6.16 Statistical results of image 7 (Bubbles) for the parameter EPI



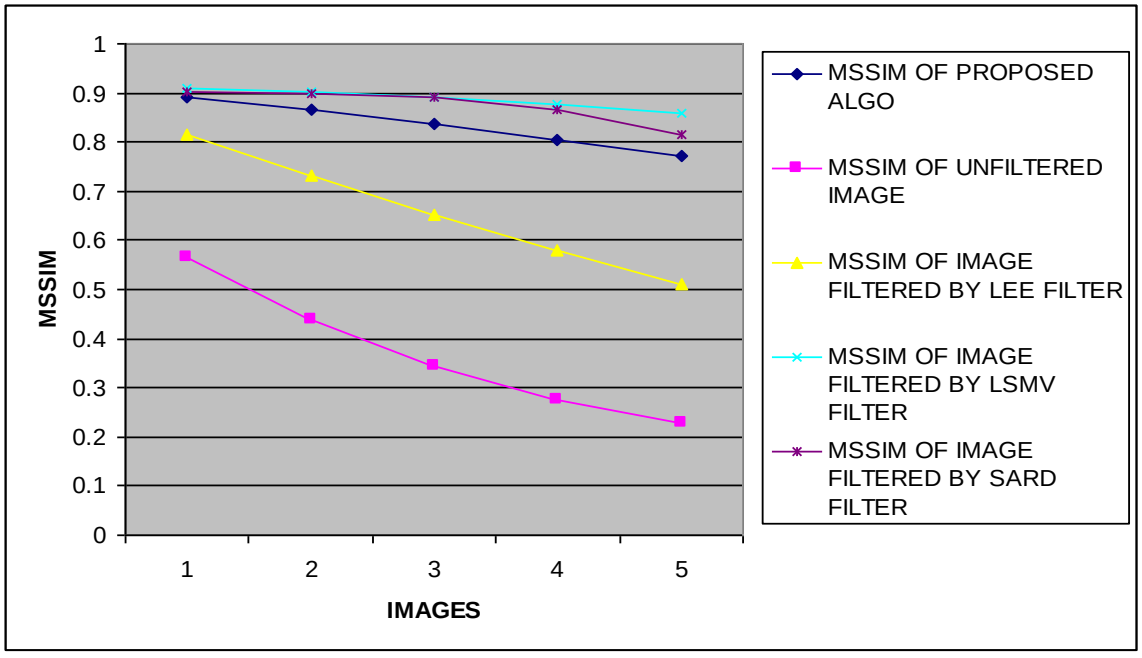
Graph 6.36 Representation of results of EPI parameter on image 8(Bubbles)



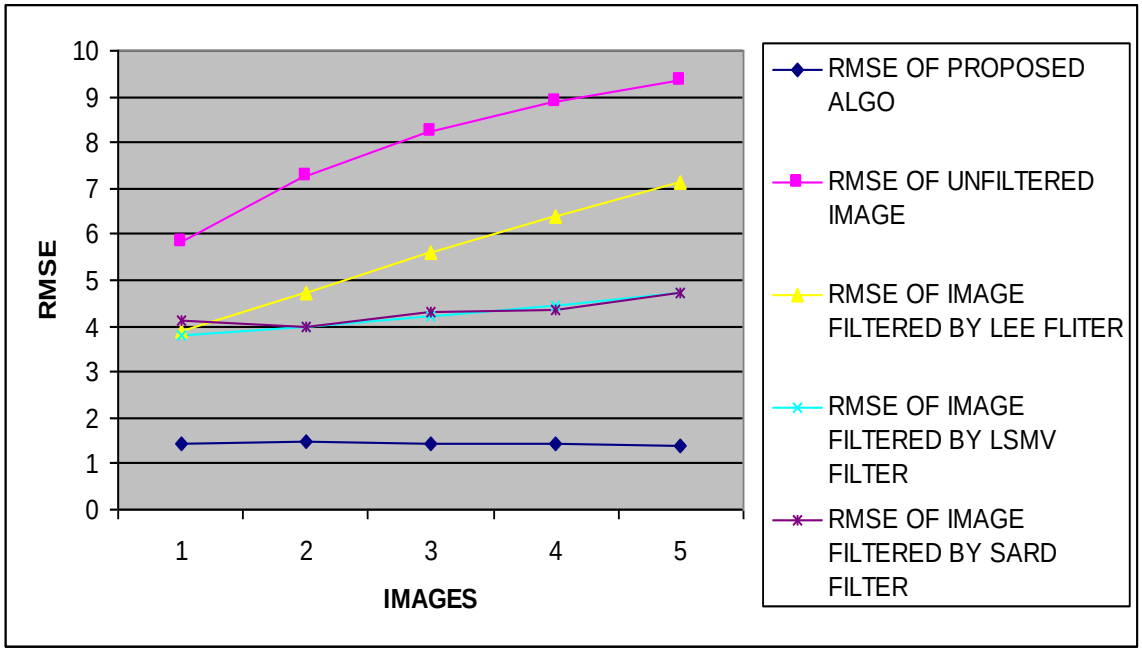
Graph 6.37 Representation of results of SNR parameter on image 8(Bubbles)



Graph 6.38 Representation of results of Coc parameter on image 8(Bubbles)



Graph 6.39 Representation of results of MSSIM parameter on image 8(Bubbles)



Graph 6.40 Representation of results of RMSE parameter on image 8(Bubbles)

6.2 Summary of results

After simulating different images on proposed algorithm we conclude that the proposed method is removing the noise and finding the edges of images with greater efficiency as shown from figures as compared to other methods or techniques of image speckle noise filtering and edge detection.

After statistical analysis it is clear that SNR, Coc, MSSIM, RMSE, Percentage Improvement and EPI are much better from proposed algorithm then that of other method used for filtering and edge detection.

CHAPTER-7
CONCLUSION AND FUTURE SCOPE

In this thesis a novel MM based algorithm is proposed to remove speckle noise present in the image and then find the edges of the filtered image. In proposed algorithm 5 step approach is used to reduce the speckle noise. Firstly OCC filtration method is applied three times with an increasing size of disk shape structuring element varying from 1 to 3 to remove speckle noise. Then residual image is obtained by doing subtraction between original and filtered image. After that from residual image the speckle noise left is removed by applying median filter and then thin features are extracted out from filtered residual image. These features are added back to the filtered image since by OCC method thin features of image get eliminated. The second part of proposed algorithm deals with finding out the edges of filtered image which is a 4 step approach. Firstly by combination of MM opening and closing operations edges are obtained. Then thin edges are obtained by applying top-hat and bottom-hat MM method. After that edges are enhanced to make low contrast edges visible and then thresholding is done. Finally the proposed algorithm filtering efficiency is compared with the previously used filters to remove speckle noise like LEE, LSMV and SRAD by using different quality assessment parameters like SNR, MSSIM, RMSE and correlation coefficient. The efficiency of proposed algorithm to find edges is compared with edge detectors like SOBEL, PREWITT and CANNY by using quality assessment parameter EPI. Also the percentage of improvement is calculated which shows improvement occur by applying the proposed algorithm on different parameters. It can be seen that the values of SNR, correlation, MSSIM and EPI are increasing and value of RMSE is decreasing by applying proposed algorithm then to already available methods.

The proposed algorithm has been tested on 25 images. The method can be applied to many practical areas, such as military object detection using SAR and medical application like ultrasound etc.

The major advantage of this algorithm is high degree of improvement in SNR and RMSE values of filtered image also a very good improvement in EPI which shows a good preservation of edges.

In some cases however the proposed method does not give large difference between the results obtained by it and that of previously used filters or edge detectors. This case comes when the image is highly noisy. For such case a high efficiency of algorithm can be a future scope of the work.

Generally there is not any universal method for speckle noise removal and edge detection of all kinds of images. So by applying some technique like changing the type and varying the size of structuring elements, it is possible to remove speckle noise depending on the type of images and level of noise in that images and also can be made possible to find edges of all kinds of practical images.

REFERENCES

[1] Deng Shiwei and Yuan Baozong, "Range Image Segmentation Using Mathematical Morphology", *IEEE Region 10 Conference On TENCON Proceedings on Computer, Communication, Control and Power Engineering*, vol-2, pp: 1009 –1011, 1993.

[2] Richard Alan Peters, "A New Algorithm for Image Noise Reduction Using Mathematical Morphology", *IEEE Transaction on Image Processing*, vol. 4, No.5, pp: 554-568, 1995.

[3] Maragos P., "Differential Morphology and Image Processing," *IEEE Trans Image Processing*, vol. 5, pp: 922–937, 1996.

[4] Scott T. Acton, Janelle A. Molloy and Yongjian Yu, "Three-Dimensional Speckle Reducing Anisotropic Diffusion", *IEEE Conference Record of the 37th Asilomar Conference on Signals, Systems and Computers*, vol.2, pp: 1987- 1991, 2003.

[5] Zhao Yuqian, Gui Wei-hua, Chen Zhencheng, Tang Jing-tian and Li Ling-yun, "Medical Images Edge Detection Based on Mathematical Morphology*", Proceedings of the IEEE Engineering in Medicine and Biology 27th Annual International Conference Shanghai, China*, pp: 6492 – 6495, 2005.

[6] Ying-Tung Hsiao, Cheng-Long Chuang, Joe-Air Jiang and Cheng-Chih Chien, "A Contour Based Image Segmentation Algorithm Using Morphological Edge Detection", *IEEE International Conference on Systems, Man and Cybernetics*, vol.3, pp: 2962– 2967, 2005.

[7] Yuqian Zhao, Weihua Gui and Zhencheng Chen, "Edge Detection Based on Multi-Structure Elements Morphology", *Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China*, vol-2, pp: 9795 – 9798, 2006.

[8] J.-A. Jiang, C.-L. Chuang, Y.-L. Lu and C.-S. Fahn, "Mathematical Morphology Based Edge Detectors for Detection of Thin Edges in Low-Contrast Regions", *The Institution of Engineering and Technology*, vol.1, Issue: 3, pp: 269–277, 2007.

[9] Liu Ting, Luo Xiaogang, Peng Chenglin and Wen Li, "Improved Morphological Edge Detection Algorithm for Ultrasound Heart Ventricular Wall Image in the Motion of Its Rotation", *1rd International conference on bioinformatics and biomedical engineering* , pp: 960-963, 2007.

[10] Li-Hui Jiang, Zhen-Ni Jin, Fan Zhang and Rui-Hua Liu, "A New Algorithm for Speckle Suppression Using Mathematical Morphology and Adaptive Weighted Technique" *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong*, pp: 19-22, 2007.

[11] Serge Beucher, "Numerical Residue", *Image and Vision Computing,* vol.25, ISMM05 Special Issue, pp: 405-415, 2007.

[12] Xueshun Wang, Dawei Qi and Yuanxiang Li, "Edge Detection of Decayed Wood Image Based on Mathematical Morphological Double Gradient Algorithm", *Proceedings of the IEEE International Conference on Automation and Logistics Qingdao,* pp: 1226-1231, 2008.

[13] Chaofeng Li and Alan C. Bovik, "Three-Component Weighted Structural Similarity Index", *SPIE,* 7242-24 Vol. 1, pp.1 of 9, 2008.

[14] Yan Wenzhong, "Mathematical Morphology Based Enhancement for Chromosome Images", *3rd International conference on bioinformatics and biomedical engineering,* pp: 1, 2009.

[15] Waheeb Abu Ulbeh, Akram Moustafa and Ziad A. Alqadi, "Gray Image Reconstruction", *European Journal of Scientific Research,* vol.27, No.2, pp:167-173, 2009.

[16] S. Sudha, G. R. Suresh and R. Sukanesh, "Speckle Noise Reduction in Ultrasound Images by Wavelet Thresholding Based on Weighted Variance", *International Journal of Computer Theory and Engineering*, vol. 1, No. 1, pp: 1793-8201, 2009.

[17] Mandeep Singh, Dr Sukhwinder Singh and Dr. Savita Kansal, "Comparative Analysis of Spatial Filters for Speckle Reduction in Ultrasound Images", *World Congress on Computer Science and Information Engineering*, vol. 6, pp: 228-232, 2009.

[18] T. Ratha Jeyalakshmi and K.Ramar, "A Modified Method for Speckle Noise Removal in Ultrasound Medical Images", *International Journal of Computer and Electrical Engineering*, vol. 2, No. 1, pp: 1793-8163, February, 2010.

[19] Ratha Jeyalakshmi and Ramar Kadarkarai, "Segmentation and Feature Extraction of Fluid-Filled Uterine Fibroid–A Knowledge-Based Approach", *Maejo International Journal of Science and Technology*, *4*(03), pp: 405-416, 2010.

[20] M Rama Bai, Dr V Venkata Krishna and J SreeDevi," A New Morphological Approach for Noise Removal cum Edge Detection**",** *International Journal of Computer Science Issues*, vol. 7, pp: 1694-0814, Issue: 6, 2010.

[21] Ahmed S. Mashaly, Ezz Eldin F. AbdElkawy and Tarek A. Mahmoud, "Speckle Noise Reduction in SAR Images Using Adaptive Morphological Filter", *10th International Conference on Intelligent Systems Design and Applications,* 2010.

[22] Shahana Bano, M. Surendra Prasad Babu and C.Naga Raju, "A New Proposed Method for Image Segmentation Based on Gray Scale Morphological Transformations", *International Journal of Computer Science Issues*, vol. 1, Issue: 2, 2010.

[23] Dr. H. B. Kekre and Ms. Saylee M. Gharge, "Image Segmentation using Extended Edge Operator for Mammographic Images", *International Journal on Computer Science and Engineering*, vol. 02, Issue: 4, pp: 1086-1091, 2010.

[24] Bhadauria H S and Dewal M L, "Comparison of Edge Detection Techniques on Noisy Abnormal Lung CT Image Before and After Using Morphological Filter", *Published in International Journal of Advanced Engineering & Application,* vol. 1, pp: 277, 2010.

[25] M. N. Nobi and M. A. Yousuf, "A New Method to Remove Noise in Magnetic Resonance and Ultrasound Images", *Journal Of Scientific Research*, Res. 3(1), pp: 81-89, 2011.

[26] Christos Loizou, Constantinos S. Pattichis and Costas Pattichis, "Despeckle filtering algorithms and software for ultrasound imaging", *published in Morgan & Claypool Publishers Series,* 2008.

[27] Rafael C. Gonzalez, Richard E. Woodes, and Steven L. Eddins, "Digital Image Processing",3rd edition, published by Pearson Education(Singapore) Pte. Ltd.

[28] J. Serra, Ed., "Image Analysis and Mathematical Morphology", *Vol.* 2: *Theoretical Advances***.** New York: Academic, 1988.