

A MODIFIED TETROLET BASED IMAGE DE-NOISING FOR REAL TIME EDGE DETECTORS

By: Eyob Teshome

Advisor: Ato Bisrat Derebssa(Msc.)

A thesis submitted to
The Department of Electrical and Computer Engineering

Presented in partial fulfillment of the requirement for the degree of
Masters of Science in Microelectronics Engineering



Addis Ababa University
Addis Ababa Institute of Technology
Addis Ababa, Ethiopia
December 2012

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
ADDIS ABABA INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**A MODIFIED TETROLET BASED IMAGE DE-NOISING FOR
REAL TIME EDGE DETECTORS**

By
Eyob Teshome
Advisor
Ato Bisrat Derebssa(MSc.)

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

**A MODIFIED TETROLET BASED IMAGE DE-NOISING FOR REAL
TIME EDGE DETECTORS**

By
Eyob Teshome

ADDIS ABABA INSTITUTE OF TECHNOLOGY
APPROVAL BY BOARD OF EXAMINERS

Chairman, Dept. of Graduate committee

Signature

Ato Bisrat Derebssa(MSc.)

Advisor

Signature

Internal Examiner

Signature

External Examiner

Signature

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been fully acknowledged.

Eyob Teshome

Name

signature

Place: Addis Ababa

Date of submission

This thesis has been submitted for examination with my approval as a university advisor.

Bisrat Derebssa(MSc.)

Advisor's name

Signature

CHAPTER 1

INTRODUCTION

1.1. BACKGROUND

According to Shehrzad [5], image de-noising needed because a noisy image is not appropriate to analyze. In addition, some fine details in the image may be confused with the noise or vice-versa. Many image processing algorithms such as pattern recognition and edge detection need a clean image to work effectively. Random and uncorrelated noise samples are not compressible. Such concerns underline the importance of de-noising in image and video processing.

Edge detection is one of the most commonly used operations in image analysis, which in some respects is a more difficult problem [5]. Image analysis algorithms draw on information present within an image to extract properties about the scene being imaged.

Singh [3] pointed out that the mostly referred image noise type is zero mean additive white Gaussian noise (AWGN) because it is symmetric, continuous, and has a smooth density distribution. However, many other types of noise exist in practice. Noise can also have different distributions such as Poisson, non-additive Salt-and-Pepper noise and speckle noise. According to Singh et al. [3], “Images are often corrupted with noise during acquisition, transmission, and retrieval from storage media. Image de-noising involves the manipulation of the image data to produce a high quality image. Many image processing algorithms such as pattern recognition, edge detection etc. need a clean image to work effectively. The goal of image de-noising is to remove noise by differentiating it from the signal. The wavelet transform’s energy compactness helps greatly in de-noising. Energy compactness refers to the fact that most of the signal energy is contained in a few large wavelet coefficients, whereas a small portion of the energy is spread across a large number of small wavelet coefficients. These coefficients represent details as well as high frequency noise in the image. By appropriately thresholding these wavelet coefficients, image de-noising is achieved while preserving fine structures in the image.”

The whole system proposed in this paper comprises of two stages;

1. **Tetrolet based image de-noising:** Tetrolets are Haar-type wavelets whose supports are Tetrominoes which are shapes made by connecting four equal-sized squares. The corresponding filter bank algorithm is computationally costly but enormously effective. Using this step the noise artifacts will be removed.
2. **First order gradient based edge detector:** the de-noised image will be fed in to this edge detector step. This helps to find the sharp intensity variation of an image and in this way it obtains the edges of the objects contained on the image.

Implementing such applications on a general purpose computer can be easier but not very efficient in terms of speed. The reason being the additional constraints put on memory and other peripheral device management. Application specific hardware offers much greater speed than a software implementation. There are two types of technologies available for hardware design. Full custom hardware design also called as Application Specific Integrated Circuits (ASIC) and semi-custom hardware device, which are programmable devices like Digital signal processors (DSP's) and Field Programmable Gate Arrays (FPGA's). Full custom ASIC design offers highest performance, but the complexity and the cost associated with the design is very high. The ASIC design cannot be changed; time taken to design the hardware is also very high. ASIC designs are used in high volume commercial applications. In addition, if an error exist in the hardware design, once the design is fabricated, the product goes useless. DSP's are a class of hardware devices that fall somewhere between an ASIC and a PC in terms of the performance and the design complexity. DSP's are specialized microprocessor, typically programmed in C, perhaps with assembly code for performance. It is well suited to extremely complex math intensive tasks such as image processing. Hardware design knowledge is still required, but the learning curve is much lower than some other design choices. Field Programmable Gate Arrays are programmable devices. They are also called reconfigurable devices. Reconfigurable devices are processors which can be programmed with a design. Hardware design techniques such as parallelism and pipelining techniques can be developed on FPGA. This is not possible in dedicated DSP designs. So FPGAs are ideal choice for implementation of real time image processing algorithms.

1.2. STATEMENT OF THE PROBLEM

Generally, edge detection operators behave poorly when they encounter images corrupted by noise. While their behavior may fall within tolerances in specific situations, in general edge detectors have difficulty adapting to different situations. The quality of edge detection is highly dependent on lighting conditions, the presence of objects of similar intensities, density of edges in the scene, and noise. A lot of image de-noising algorithms have been proposed for edge detectors. The commonly used methods such as median filtering [11] and soft-threshold wavelet de-noising [10], only removes Salt-and-Pepper noise, and limited in de-noising performance respectively.

Recently Singh et al. [3] proposed a new approach to the de-noising problem based on the Tetrolet transform. The corresponding de-noising algorithm is very effective in terms of PSNR. However the algorithm has several shortcomings, Firstly, Singh et al. [3] did not determine the criteria to select the best or few best Tetromino partitions among all the possible candidates, which he pointed it as a future work. In other words, he fails to reduce cost of adaptively. Secondly, Singh et al. [3] uses the so called universal thresholding method determine the clean wavelet coefficients. Later Cai-lian Li [2] improved the universal thresholding scheme by using the SureRisk method.

Canny edge detection operator is relatively better than other edge detection operators' in terms of performance but it is computationally expensive; therefore it is hardly used for real time realization of edge detectors. On the other hand, the first order gradient based edge detection operators are less computationally expensive than other edge detection schemes; however they are prone to noise [5].

In general, the main motivation of this research is to use the Tetrolet transform based image de-noising algorithm for first order derivative edge detection operators as a preprocessing scheme by reducing its computational complexity. The whole system will be synthesized and simulated on Xilinx ISE and ModelSim SE software's using Xilinx Virtex®-4 FPGA's platform.

1.3. OBJECTIVE

1.3.1. GENERAL OBJECTIVE

Given a standard image, the general objective is to develop a real-time system that is capable of de-noising as well as detecting the edge of an image.

1.3.2. SPECIFIC OBJECTIVE

- To study and understand a Tetrolet transform and a wavelet transform based image de-noising algorithm.
- To design and simulate an improved image de-noising algorithm using a Tetrolet transform using MATLAB.
- To compare the modified image de-noising algorithm with the previous Tetrolet based image de-noising algorithms.
- To design and simulate a real-time hardware realization of a Tetrolet based image de-noising algorithm.
- To combine the edge detection operator and the de-noising algorithms and compare their performance including our method.
- To compare the software (MATLAB implementation) and the FPGA implementation of the de-noising algorithm via simulation.

1.4. METHODOLOGY

Literature review and assessment: A literature analysis and theoretical assessment of existing systems is done. Different existing systems are studied and compared in order to find out a way to solve the problem.

System Design: After choosing the relevant techniques, the system was designed using the selected technique for each stage.

Implementation and Performance evaluation: In order to check the accuracy of the system, it is tested using standard images. Based on the test result, some modifications are done on the system. The designed algorithm has been coded in MATLAB and verified. The design has been transformed to a model in VHDL that can be synthesized on Xilinx FPGA. Finally, the performance of the algorithm has been compared with benchmark techniques.

1.5. THESIS SCOPE

The scope of this thesis lies in improving the performance of a Tetrolet transform based image de-noising algorithm in terms of performance. Secondly, it is designed to use this de-noising scheme for first order derivative based edge detection operators as a pre-processing scheme.

1.6. CONTRIBUTION AND LIMITATIONS

The ambition to replace human tasks by machines led to invention of different intelligent systems that are capable of processing, analyzing and manipulating information alone. Some of these systems are speech recognition, image processing, computer vision and other. Image edge detection system is one of the image processing applications. It has got several applications in machine vision applications, in medical imaging registration systems, in military tracking systems and others [5]. From this view, the main contribution of this thesis is to develop a system that have a better performance than the previous wavelet based image de-noising algorithms and to use this system as a pre-processing scheme for edge detection algorithms.

The main contribution of this study was to reduce the computational complexity of the Tetrolet based de-noising algorithm and see its performance on the hardware realization.

1.7. ORGANIZATION OF THE THESIS

In the subsequent chapters, literatures related to the work of this thesis will be discussed. The theoretical background of Image Processing algorithms used in this research is then described in chapter 2. In third chapter of the report, the model of the designed system and its implementation detail are presented. The results of the work along with discussions are given in chapter four. Finally, conclusions and recommendations are given.

CHAPTER 2

LITERATURE REVIEW

2.1. INTRODUCTION

Human beings are predominantly visual creatures: we rely heavily on our vision to make sense of the world around us. Humans have evolved very precise visual skills: we can identify a face in an instant, differentiate colors, and process a large amount of visual information very quickly and others. The ambition to create machine that are capable of recognizing information from images led to the development in the field of image processing. Image processing involves changing the nature of an image in order to either improve its pictorial information for human interpretation, or render it more suitable for autonomous machine perception. It has wide applications in computer vision, artificial intelligence, detection systems, monitoring systems and others [4].

According to [5], Edge detection is an initial step in object recognition. The most commonly used edge detection techniques are Gradient-based and Laplacian based edge detection methods. Gradient-based algorithms such as the Roberts filter have a major drawback of being very sensitive to noise.

The main components of overall system are; a preprocessing image de-noising step and the edge detection step. The image de-noising step is responsible for removing the noise artifacts. It uses different features and properties of the image noise in order to remove the unpleasant feature of the image. Some of the properties that it makes use of are: the standard deviation of the noise, the variance of the noise and the sampling interval in case of Poisson noise. Tetrolet based image de-noising strategies that involve smaller computational resources for the purpose of producing fastest algorithm can be an active research area especially to achieve real-time performance. Besides the software implementation, FPGAs were being used to implement and significantly accelerate such algorithms. The second component, the Edge detection system, is used to find the sharp intensity variation of an image.

2.2. LITERATURES ON IMAGE FILTERING METHODS

According to Singh et al [3], there are many different kinds of image filtering methods. They can be broadly classified into two classes:

1. Spatial domain filtering
2. Transform domain filtering

As the name indicates, spatial domain filtering refers to filtering in the spatial domain, while transform domain filtering refers to filtering in the transform domain. The modified Tetrolet transform based image de-noising algorithm fall into transform domain filtering.

2.2.1. SPATIAL DOMAIN FILTERING

Spatial domain filtering can be further divided on the basis of the type of filter used:

- Linear filters
- Non-Linear filters

An example of a non-linear filter is the median filter. Median filtering is quite useful in removing of Salt and Pepper type noise. Spatial filters tend to results blurring in the de-noised image.

2.2.2. TRANSFORM DOMAIN FILTERING

Transform domain filtering can be further divided into two broad classes based on the type of transform used:

- Fourier transforms filters
- Wavelet transforms filters

In general, all wavelet transform de-noising algorithms involve the following three steps.

- Forward Wavelet Transform: Wavelet coefficients are obtained by applying the Wavelet transform.
- Thresholding: Clean coefficients are estimated from the noisy ones.
- Inverse Wavelet Transform: A clean image is obtained by applying the inverse wavelet transform.

There are many ways to perform the thresholding step in wavelet based de-noising technique.

These methods use a threshold and determine the clean wavelet coefficients based on this threshold. There are two main ways of thresholding the wavelet coefficients, namely the hard thresholding method and the soft thresholding method. A summary of various thresholding methods used for de-noising is given below.

I. Wavelet (Hard Thresholding method)

If the absolute value of a coefficient is less than a threshold, then it is assumed to be 0, otherwise it is unchanged. Mathematically it is,

$$X = \text{sign}(Y) (Y .* (\text{abs}(Y) > \lambda)), \quad (2.1)$$

Where Y represents the noisy coefficients, λ is the threshold, X represents the estimated coefficients.

“.” indicates a Matlab operator, which represents a scalar multiplication of matrices.

II. Wavelet (Soft Thresholding Method)

Hard thresholding is discontinuous. This causes ringing / Gibbs effect in the de-noised image. To overcome this, Donoho [6] introduced the soft thresholding method.

If the absolute value of a coefficient is less than a threshold λ , then it is assumed to be 0, otherwise its value is shrunk by λ . Mathematically it is

$$X = \text{sign}(Y) .* ((\text{abs}(Y) > \lambda) .* (\text{abs}(Y) - \lambda)) \quad (2.2)$$

This removes the discontinuity, but degrades all the other coefficients which tend to blur the image.

“.” indicates a Matlab operator, which represents a scalar multiplication of matrices.

III. Wavelet (VisuShrink)

VisuShrink was introduced by Donoho [6]. It uses a threshold value t that is proportional to the standard deviation of the noise. It follows the hard thresholding rule. It is also referred to as universal threshold and is defined as $\sigma\sqrt{2\log M}$.

σ^2 is the noise variance present in the signal and M represents the signal size or number of samples.

IV. Wavelet (SUREShrink)

A threshold chooser based on Stein's Unbiased Risk Estimator (SURE) was proposed by Donoho and Johnstone [7] and is called as SureShrink. It is a combination of the universal threshold and the SURE threshold. This method specifies a threshold value t_j for each resolution level j in the wavelet transform which is referred to as level dependent thresholding. The goal of SureShrink is to minimize the mean squared error, defined as

$$\text{MSE} = \frac{1}{n^2} \sum_{x,y=1}^n (z(x,y) - s(x,y))^2 \quad (2.3)$$

Where $z(x,y)$ is the estimate of the signal while $s(x,y)$ is the original signal without noise and n is the size of the signal. SureShrink suppresses noise by thresholding the empirical wavelet coefficients. The SureShrink threshold t^* is defined as $t^* = \min(t, \sigma\sqrt{2\log n})$,

Where t denotes the value that minimizes Stein's Unbiased Risk Estimator, σ is the noise variance, and n is the size of the image.

V. Wavelet (BayesShrink)

BayesShrink was proposed by Chang, Yu and Vetterli [8]. The goal of this method is to minimize the Bayesian risk, and hence its name, BayesShrink. It uses soft thresholding and is subband-dependent, which means that thresholding is done at each band of resolution in the wavelet decomposition. Like the SureShrink procedure, it is smoothness adaptive. The Bayes threshold, t_B , is defined as 2.4.

$$t_B = \frac{\sigma^2}{\sigma_s} \quad (2.4)$$

Where σ^2 is the noise variance and σ_s is the signal variance without noise. From the definition of additive noise we have 2.5:

$$w(x, y) = s(x, y) + n(x, y) \quad (2.5)$$

Since the noise and the signal are independent of each other, it can be stated that

$$\sigma_w^2 = \sigma_s^2 + \sigma^2. \quad (2.6)$$

σ_w^2 can be computed as shown in equation (2.7):

$$\sigma_w^2 = \frac{1}{n^2} \sum_{x,y=1}^n (w^2(x, y)) \quad (2.7)$$

The variance of the signal, σ_s^2 is computed as (2.8),

$$\sigma_s^2 = \sqrt{\max(\sigma_w^2 - \sigma^2)} \quad (2.8)$$

With σ^2 and σ_s^2 the Bayes threshold is computed from the above Equations.

VI. Tetrolet (Hard Thresholding)

Jens Krommweh [1] proposed a new method for image compression using an adaptive Haar like transform. He called it the Tetrolet transform. It is a simple concept, but quite effective in compression. In the 2D Haar transform, images are divided into 2x2 blocks and the Haar wavelet transform is applied to generate one average and three detailed coefficients. These coefficients capture the detailed information along the horizontal, vertical and diagonal direction. In the Tetrolet transform approach, images are sub-divided into 4x4 blocks. Each 4x4 block is partitioned using Tetrominoes. Following this, the Haar transform is applied to generate 4 average coefficients and 12 detailed coefficients. Tetrominoes are the shapes formed by joining four squares such that they connect with each other at least on one edge.

Singh et al. [3] proposed a new approach to the de-noising problem based on the Tetrolet transform proposed by Jens Krommweh [1] for image compression. It is based on the Haar wavelet transform, but adapts to image characteristics automatically. He came up with a simple Haar transform based de-noising algorithm that works on each 4x4 sub-block of an image independently. He achieves up to 2 dB better peak Signal-to-noise ratio (PSNR) compared with

other published Haar wavelet Transform-based methods. However the software implementation of this method takes a significant amount of computational time.

VII. Tetrolet (SureShrink)

Cai-lian Li [2] improves the Tetrolet based image de-noising scheme using the SureShrink thresholding method instead of the previous universal hard thresholding scheme. As a result, he achieved a better PSNR and visual quality. However its software implementation still takes a tremendous amount of computational time.

2.3. LITERATURES ON EDGE DETECTION

According to [5], Edge detection is an initial step in object recognition. The most commonly used edge detection techniques are Gradient-based and Laplacian based edge detection methods. Gradient-based algorithms such as the Prewitt filter have a major drawback of being very sensitive to noise. Second order derivative operators like the LoG (Laplacian of Gaussian), are also very sensitive to noise. On the other hand, Canny's edge detection algorithm is computationally more expensive compared to Sobel, Prewitt and Robert's operator. However, the Canny's edge detection algorithm performs relatively better than other operators under almost all scenarios. Evaluation of the images showed that under noisy conditions, Canny, LoG (Laplacian of Gaussian), Sobel, Prewitt, Roberts's exhibit better performance, respectively.

In July 2010, a method which combines Sobel edge detection operator and soft-threshold wavelet de-noising to do edge detection on images which include White Gaussian noises was proposed [10]. In recent years, a lot of edge detection methods are proposed. The commonly used methods which combine mean de-noising and Sobel operator or median filtering and Sobel operator [10] are poor in their performance.

2.4. LITERATURES ON TETROLET TRANSFORM USING FPGA

The wavelets transform is gaining momentum to become an alternative tool to traditional time-frequency representation techniques such as the discrete Fourier transform and the discrete cosine transform. Wavelet Transform offers multi-resolution image analysis, which appears to be well matched to the low level characteristic of human vision. The DCT (discrete cosine

transform) is essentially unique but wavelet transform has many possible realizations [15]. The discrete wavelet transform is computationally intensive and operates on large datasets. This factor, coupled with the demand for real time operation in many image processing tasks, made the traditional sequential computers fall short in meeting such requirements. In turn, this necessitated the search for high performance implementations at a reasonable cost. Implementations of the discrete Tetrolet transform can be grouped into two major categories; software implementations using programmable systems, and dedicated hardware implementations using Field Programmable gate arrays or ASIC's. Each implementation category presents different trade-offs in terms of performance, cost, power, and flexibility.

Furthermore, FPGAs inherit design flexibility and adaptability of software implementations. The Tetrolet based de-noising has been realized using Software [3]. However it has not been realized on hardware. That is due to the fact that, Tetrolet Transform was incorporated in to the research arena before a few years ago [1].

2.5. SUMMARY

In the case where an image is corrupted with Gaussian noise, the wavelet based de-noising has proved to be nearly optimal. SureShrink [7] produces the best PSNR compared to VisuShrink [6], BayesShrink [8], Hard and Soft thresholding methods. However, the output from BayesShrink [8] method is much closer to the high quality image and there is no blurring in the output image unlike the other four methods. VisuShrink [6] cannot de-noise multiplicative noise unlike BayesShrink [8]. It has been observed that BayesShrink [8] is not effective for noise variance higher than 0.05. Out of all the wavelet based de-noising schemes the Tetrolet based algorithm shows by far a good PSNR as well as visual quality. However for the previous software implementation of the algorithm takes a significant amount of computational time.

CHAPTER 3

SYSTEM DESIGN AND DEVELOPMENT

3.1. INTRODUCTION

This chapter discusses about the basic algorithm used to develop the Tetrolet based image de-noising for the edge detection operator. The overall system can be divided into two main parts; Tetrolet based image de-noising system and the edge detector system. It also includes the techniques, design prototypes and the parameters for developing this study.

3.2. SYSTEM COMPONENTS

The overall architecture of the Tetrolet based image de-noising for the edge detector system developed in this study is as follows.

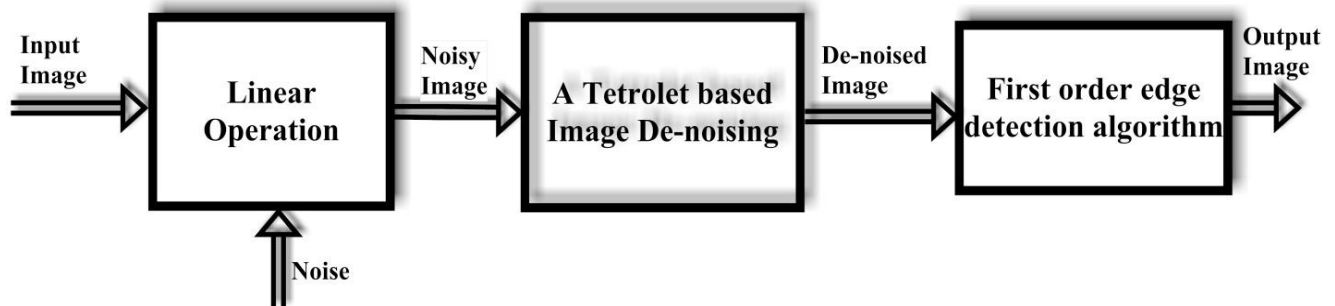


Figure 3.1: Main components of the system

In each component, there are many processes and algorithms. In the following sections, the details development specification and algorithms used will be discussed.

3.2.1. LINEAR OPERATION

The “Linear operation” shown in Figure 3.1 is the addition or multiplication of the noise to the image signal. In the image de-noising process, information about the type of noise present in the original image plays a significant role.

Noise is present in an image either in an additive or multiplicative form. An additive noise follows the rule,

$$w(x, y) = s(x, y) + n(x, y) , \quad (3.1)$$

While the multiplicative noise satisfies,

$$w(x, y) = s(x, y) \times n(x, y) , \quad (3.2)$$

The signs “+” and “×” indicates scalar addition and multiplication of matrices.

Where $s(x,y)$ is the original signal, $n(x,y)$ denotes the noise introduced into the signal to produce the corrupted image $w(x,y)$, and (x,y) represents the pixel location. The above image algebra is done at pixel level. By image noise multiplication, we mean the brightness of the image is varied, whereas for the case of addition the image gets blurred.

I. White Gaussian Noise

Gaussian noise is evenly distributed over the signal [4]. This means that each pixel in the noisy image is the sum of the true pixel value and a random Gaussian distributed noise value. As the name indicates, this type of noise has a Gaussian distribution, which has a bell shaped probability distribution function given by,

$$F(g) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(g-m)^2/2\sigma^2} , \quad (3.3)$$

Where g represents the gray level, m is the mean or average of the function and σ is the standard deviation of the noise.

II. Speckle Noise

Speckle noise [4] is a multiplicative noise. This type of noise occurs in almost all coherent imaging systems such as laser, acoustics, ultra-sound and SAR (Synthetic Aperture Radar) imagery. The source of this noise is attributed to random interference between the coherent returns. Fully developed speckle noise has the characteristic of multiplicative noise. Speckle noise follows a gamma distribution and is given as

$$F(g) = \frac{g^{\alpha-1}}{(\alpha-1)!a^\alpha} e^{-\frac{g}{a}} \quad (3.4)$$

Where variance is $a^2\alpha$ and g is the gray level.

III. Poisson Noise

Medical images are often noisy owing to the physical mechanisms of the acquisition process. The great majority of the de-noising algorithms assume additive white Gaussian noise. However, some of the most popular medical image modalities are degraded by some type of non-Gaussian noise. Among these types, we refer the Poisson noise, which is particularly suitable for modeling the counting processes associated to many imaging modalities such as PET, SPECT, and fluorescent confocal microscopy imaging [12]. Unlike Gaussian image noise, Poisson noise is a data dependent image noise [19].

3.2.2. TETROLET BASED IMAGE DE-NOISING

The main aim of this stage is to remove the noise added in the first stage. The main Processing in this stage is shown in Figure 3.2.



Figure 3.2: Basic steps in Tetrolet based image de-noising step.

I. Tetrolet based image decomposition

The Tetrolet transform is a new geometrical image-based transform, which is recently introduced by Jens Krommweh[1]. Tetrolets are Haar-type wavelets whose supports are Tetrominoes which are shapes made by connecting four equal-sized squares. The adaptive Tetrolet decomposition algorithm is described as Figure 3.3.

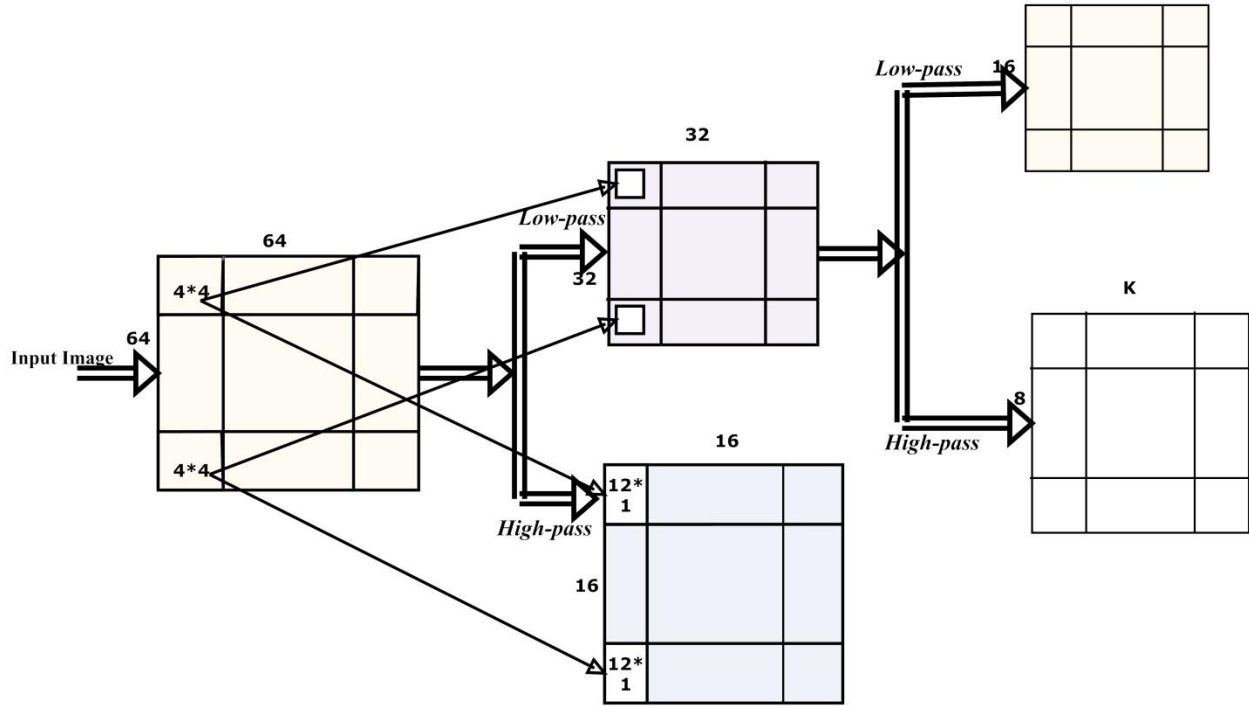


Figure 3.3: Sketch map of Tetrolet Decomposition Frame for 64x64 input image [2]

Definition and Notations: [1] Assume a two-dimensional square data sets I .

$I = \{(i, j) : i, j = 0, \dots, N-1\} \subset \mathbb{Z}^2$ be the index set of a digital image $a = (a[i, j])_{(i,j) \in I}$ with $N = 2^J$, where $J \in \mathbb{N}$.

A 4-neighborhood of an index $(i, j) \in I$ will be $N_4(i, j) := \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$.

Taking the bijective mapping, a one-dimensional index set $J(I)$ shall be:

$$J: I \rightarrow \{0, 1, \dots, N^2-1\} \text{ with } J((i, j)) := jN + i.$$

Tetrominoes are shapes made by connecting four equal-sized squares, each joined together with at least one other square along an edge. Mathematically [1], I_v subset of I contains four indices, i.e. $|I_v| = 4$, and every index of I_v has a neighbor in I_v . For $I_v = \{(i_1, j_1), (i_2, j_2), (i_3, j_3), (i_4, j_4)\}$ bijective mapped to $L: I_v \rightarrow \{0, 1, 2, 3\}$.

The Haar wavelet transform is one of the most simple wavelet transforms. The scaling and wavelet functions for the Haar wavelet transform are defined as follows:

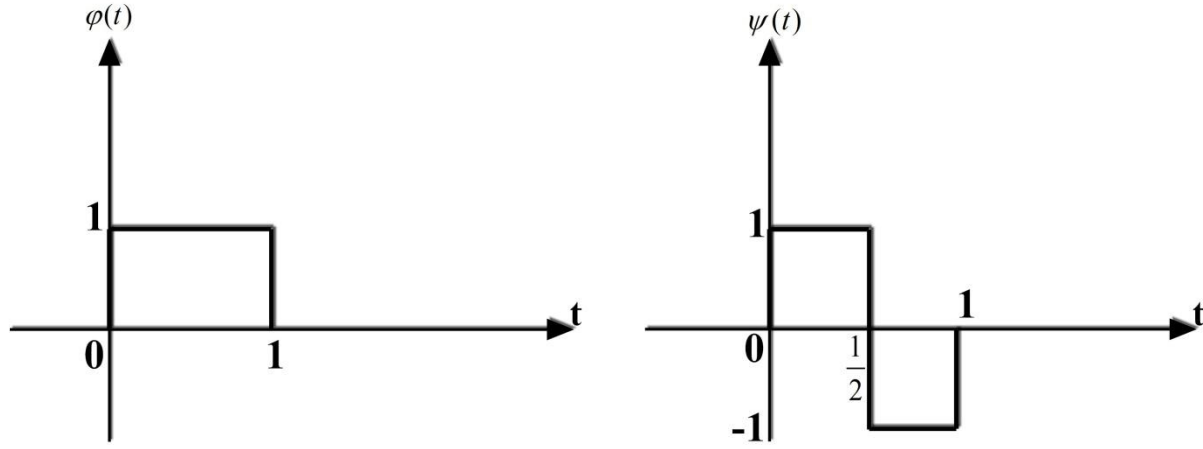


Figure 3.4: Haar Scaling ($\varphi(t)$) and mother wavelet ($\psi(t)$) functions

$$\varphi(t) = 1 \text{ for } 0 < t < 1; \text{ } 0 \text{ otherwise.} \quad (3.5)$$

$$\psi(t) = 1 \text{ for } 0 < t < 0.5; \text{ } -1 \text{ for } 0.5 < t < 1; \text{ } 0 \text{ otherwise.} \quad (3.6)$$

In case of 2D-Haar image decomposition, the low-pass filter and the high-pass filters are given by the averaging sum and the averaging differences of each four pixel values which are arranged in a 2×2 square. With $I_{i,j} = \{(2i, 2j), (2i+1, 2j), (2i, 2j+1), (2i+1, 2j+1)\}$ for $i, j = 0, 1, \dots, N/2 - 1$, we have a dyadic partition $E = \{I_{0,0}, \dots, I_{N/2-1, N/2-1}\}$ of the image index set I . Let L be the bijective mapping, which maps the four pixel pairs of $I_{i,j}$ to the set $\{0, 1, 2, 3\}$, i.e., it brings the pixels into a unique order. Then we can determine the low-pass part as,

$$a^1 = (a^1[i, j])_{i,j=0}^{N/2-1} \text{ with } a^1[i, j] = \sum_{(i', j') \in I_{i,j}} \epsilon[0, L(i', j')] a[i', j'] \quad (3.7)$$

As well as the three high-pass parts for $l = 1, 2, 3$

$$W_l^1 = (w_l^1[i, j])_{i,j=0}^{N/2-1} \text{ with } w_l^1[i, j] = \sum_{(i', j') \in I_{i,j}} \epsilon[l, L(i', j')] a[i', j'], \quad (3.8)$$

Where the coefficients $\epsilon[l, m]$, $l, m = 0, \dots, 3$, are entries from the Haar wavelet transform matrix.

$$W := (\epsilon[l, m])_{l,m=0}^3 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

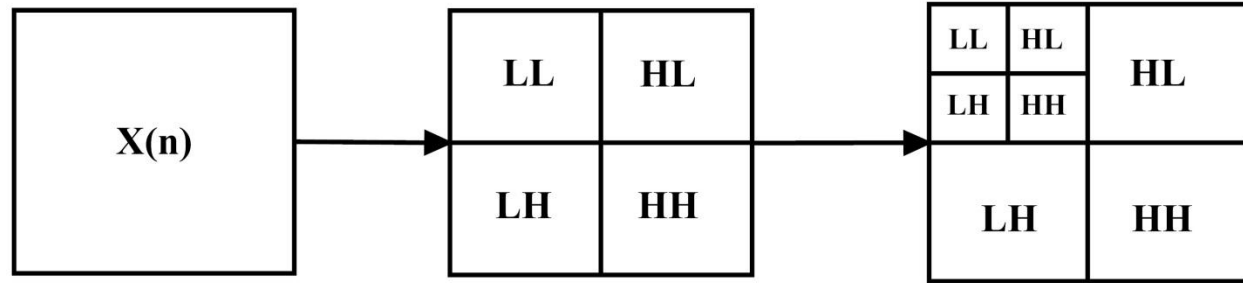


Figure 3.5: 2D Haar Image Decomposition

In the 2D Haar transform, images are divided into 2×2 blocks and the Haar wavelet transform is applied to generate one average and three detailed coefficients. These coefficients capture the detailed information along the horizontal, vertical and diagonal direction. In the Tetrolet transform approach, images are sub-divided into 4×4 blocks. Each 4×4 block is partitioned using Tetrominoes. Following this, the Haar transform is applied to generate 4 average coefficients and 12 detailed coefficients.

Tetrominoes are the shapes formed by joining four squares such that they connect with each other at least on one edge. The tiling problem with Tetrominoes became popular through the famous computer game classic 'Tetris' [16]. Disregarding rotations and reflections there are five different shapes, the so called free Tetrominoes, shown in Figure 3.6. Taking the isometries into account, it is clear that every square $[0, N]^2$ can be covered by Tetrominoes if and only if N is even. There are 117 solutions for disjoint covering of a 4×4 board with four Tetrominoes [17]. For an 8×8 board we compute $117^4 > 10^8$ as a rough lower bound of possible tilings. Thus, in order to handle the number of solutions, it will be reasonable to restrict an image partition of 4×4 squares. As represented in Figure 3.7, we have 22 fundamental solutions in the 4×4 board (disregarding rotations and reflections). One solution (first line) is unaltered by rotations and reflections, four solutions (second line) give a second version applying the isometries. Seven forms can occur in four orientations (third line), and ten asymmetric cases in eight directions (last line). Studies on Tetrominoes can be found in [16].

What if we allow more general partitions such that the local image geometry is taken into account?

The generalization of two-dimensional classical Haar wavelet transform will result in "Tetrolet transform".

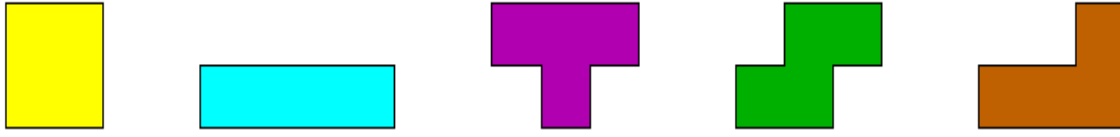


Figure 3.6: The 5 free Tetrominoes: O - I - T - S - L.

Obviously [1], the fixed blocking by the dyadic squares $I_{i,j}$ is very inefficient because the local structures of an image are disregarded. Our idea is to allow more general partitions such that the local image geometry is taken into account. Namely, we use Tetromino partitions. As described in the previous subsection, we divide the image index set into 4×4 blocks. Then instead of using the classical Haar wavelet transform, which corresponds to a partition of the 4×4 block into squares (as in the first line of Figure 3.7), we compute the 'optimal' partition of the block into four Tetrominoes according to the geometry of the image.

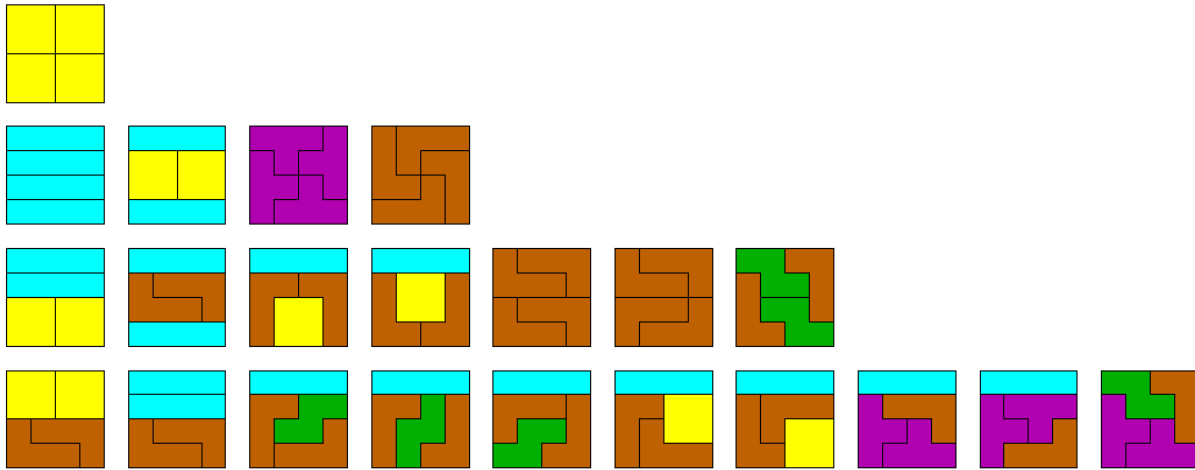


Figure 3.7: The 22 fundamental forms tiling a 4×4 board.

In this Tetrolet transform based de-noising stage, the following steps are undertaken.

The image is extended if its height and width are not multiples of 4. After de-noising takes place, the image is cropped to get the original size. The extended image is divided into 4×4 blocks, and the following steps are performed for each of the blocks:

1. For the standard image decomposition, A Tetrom configuration which can completely cover the block is picked. There are a maximum of 117 possible configurations. The Haar partition is initially chosen, but it is not necessary to always start with it. Unlike the previous Tetrolet based image de-noising algorithms, on this paper, it is only selected 15 best partitions [1], which can represent the total 117 partitions by removing the redundant partitions. Which is proposed to reduce the computational time of the original Tetrolet based image de-noising algorithm, which takes a significant amount of computational time for a software implementation.

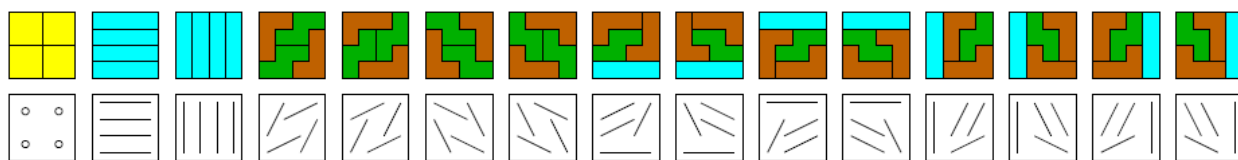


Figure 3.8: The 15 selected Tetromino configurations.

2. The samples of the low pass filter are arranged to minimize their Hamming distance from the corresponding Haar partition. This step is required to remove arbitrary arrangement of samples and prepare average coefficients for the next level of decomposition. Squares of Haar partitions are labeled as 0, 1, 2 and 3. The Hamming distances between the squares of the Haar partition and the 24 different arrangements of the squares of a given Tetromino partition are computed. The particular arrangement of squares which gives the minimum hamming distance is chosen, as described by Jens Krommweh [1].
3. The Haar transform of the arranged samples is calculated.

II. Tetrolet Coefficients Thresholding

Direct thresholding of the Tetrolet coefficients does not produce good results. The Tetrolet coefficients are thresholded using different methods. The Haar coefficients generated in the above step are thresholded. A scaled version of the universal threshold obtained by the formula $T = \sigma\sqrt{2\log(M)} \cdot 0.6$ [6] is used for thresholding. By experiments it is found that the scaled version produces good results. The scale factor is another parameter that can be tuned. Variations are possible here. Any type of thresholding (including Soft and Hard thresholding methods) can be used.

For the edge detection operator, the Tetrolet based image de-noising algorithm is used as a preprocessing scheme. However the scaled version of the universal hard thresholding is not used as a thresholding step for the image de-noising algorithm. Instead the SURE thresholding method is used as a thresholding step for image de-noising algorithm for the combined system. This brings a better performance in terms of PSNR and visual quality compared to the universal hard thresholding method. The combined system is the combination of the Tetrolet based image de-noising algorithm and the first derivative edge detection algorithm.

III. Tetrolet based image reconstruction

For the construction of the image, we need the low-pass coefficients from the coarsest level and the Tetrolet Coefficients as usual. Moreover, the information about the respective covering in each level and block is necessary.

In this stage the following steps are performed.

1. An inverse Tetrolet transform from the threshold coefficients is done to get a sample of the recovered pixels.
2. The whole processes are iterated after picking another way to partition the 4x4 block. The average of all the collected samples is taken.
3. Pixels produced by the above method are the de-noised version of the noisy pixels.

Figure 3.9 shows the modified Tetrolet based image de-noising algorithm using flowchart. After the algorithm starts, we need to initialize temporary memory storage for the output image. Then we will loop through the selected Tetrolet partitions. Inside the loop we will undertake the basic steps of Tetrolet based image de-noising for all partition cases. After the loop ends, we should recover the image from all the selected partitions.

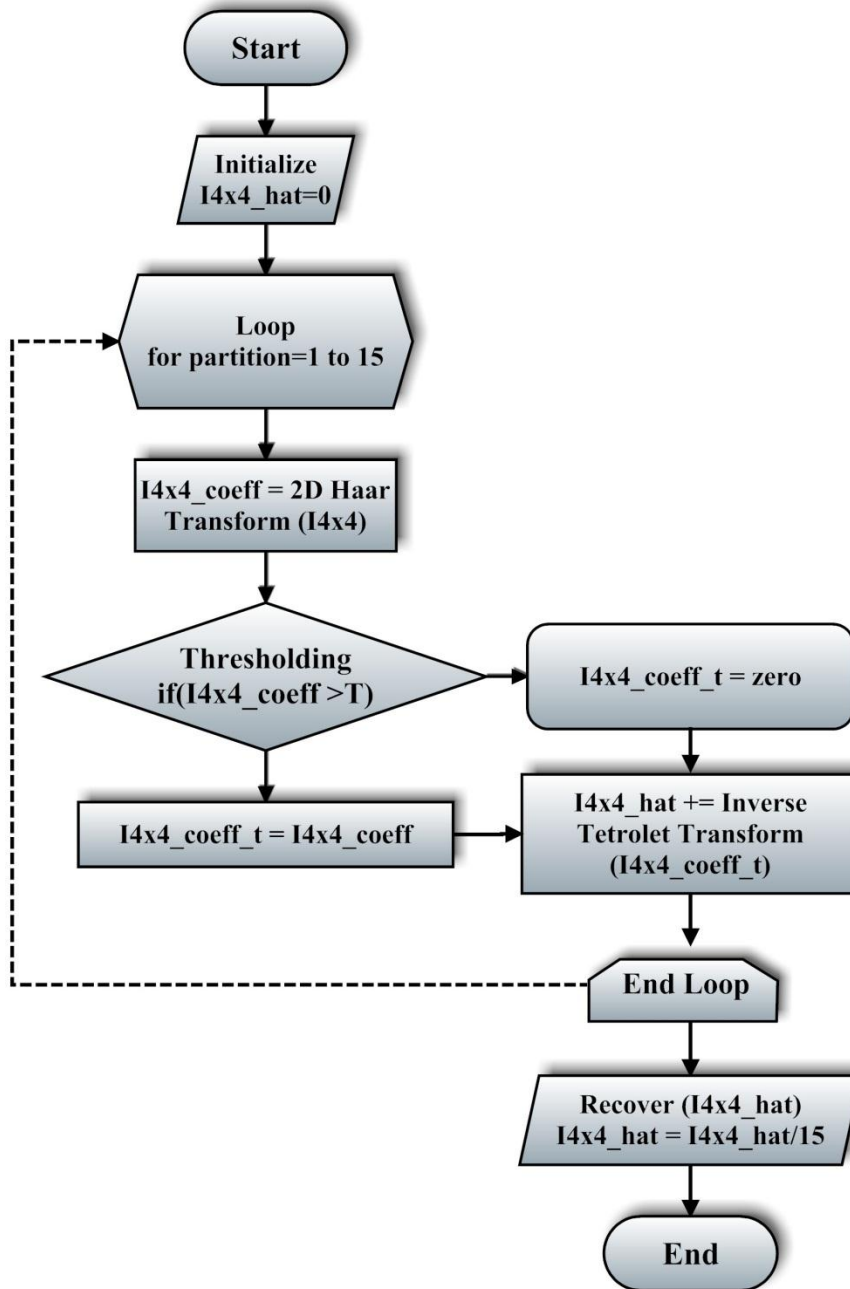


Figure 3.9: The modified Tetrolet based image de-noising algorithm using flowchart.

3.2.3. THE FIRST DERIVATIVE EDGE DETECTION

According to Gonzalez [4], Edge detection is one of the most important operations in image analysis. An edge is a set of connected pixels that lie on the boundary between two regions. The first derivative assumes a local maximum at an edge. For a gradient image $f(x, y)$, at location (x, y) , where x and y are the row and column coordinates respectively, we typically consider the two directional

derivatives. The two functions that can be expressed in terms of the directional derivatives are the gradient magnitude and the gradient orientation. [4] The gradient magnitude is defined by,

$$|\nabla f| = |[G_x, G_y]| = \left| \left[\frac{df}{dx}, \frac{df}{dy} \right] \right| = \sqrt{G_x^2 + G_y^2} \quad (3.9)$$

Where G_x and G_y are the gradient magnitudes in the x and y directions respectively.

This quantity give the maximum rate of increase of $f(x,y)$ per unit distance in the gradient orientation of $|\nabla f|$. The gradient orientation is also an important quantity. The gradient orientation is given by

$$\text{Angle}(\nabla f(x, y)) = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (3.10)$$

The angle is measured with respect to the x- axis. The direction of the edge at (x, y) is perpendicular to the direction of the gradient vector at that point. The other method of calculating the gradient is given by estimating the finite difference.

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h, y) - f(x, y)}{h} \quad (3.11)$$

$$\frac{df}{dy} = \lim_{h \rightarrow 0} \frac{f(x, y+h) - f(x, y)}{h} \quad (3.12)$$

Therefore we can approximate this finite difference as

$$\frac{df}{dx} = \frac{f(x+h, y) - f(x, y)}{h} = f(x + 1, y) - f(x, y), (h_x = 1) \quad (3.13)$$

$$\frac{df}{dy} = \frac{f(x, y+h) - f(x, y)}{h} = f(x, y + 1) - f(x, y), (h_y = 1) \quad (3.14)$$

Using the pixel coordinate notation and considering that i corresponds to the direction of x and j corresponds to the y direction.

$$\frac{df}{dx} = f(i, x + 1) - f(i, j) \quad (3.15)$$

$$\frac{df}{dy} = f(i - 1, y) - f(i, j) \text{ or } \frac{df}{dy} = f(i, j) - f(i + 1, j) \quad (3.16)$$

The most popular classical gradient-based edge detectors are Roberts cross gradient operator, Sobel operator and the Prewitt operator. Due to the similarity of Sobel and Prewitt operator, we will only discuss Sobel edge detector.

I. ROBERTS EDGE DETECTOR

The calculation of the gradient magnitude and gradient orientation of an image is obtained by the partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ at every pixel location. The simplest way to implement the first order partial derivative is by using the Roberts cross gradient operator.

$$\frac{df}{dx} = f(i, j) - f(i + 1, j + 1) \quad (3.17)$$

$$\frac{df}{dy} = f(i + 1, j) - f(i, j + 1) \quad (3.18)$$

The above partial derivatives can be implemented by approximating them to two 2x2 masks. The Roberts operator masks are [5],

$$G_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

These filters have the shortest support, thus the position of the edges is more accurate, but the problem with the short support of the filters is its vulnerability to noise. It also produces very weak responses to genuine edges unless they are very sharp.

III. SOBEL EDGE DETECTOR

Unlike the Roberts edge detection algorithm, the Sobel edge detection algorithm has 3x3 convolution masks. The partial derivatives of the Sobel operator are calculated as [5],

$$G_x = (a_2 + 2a_3 + a_4) - (a_0 + 2a_7 + a_6) \quad (3.19)$$

$$G_y = (a_6 + 2a_5 + a_4) - (a_0 + 2a_1 + a_2) \quad (3.20)$$

Therefore, the Sobel masks are

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Although the Prewitt masks are easier to implement than the Sobel masks, the later has better noise suppression characteristics.

3.3. HARDWARE DESIGN AND SIMULATION

3.3.1. INTRODUCTION

This chapter explains aspects of hardware design, synthesis and simulation of the edge detection after Tetrolet based image de-noising algorithm using FPGA.

3.3.2. HARDWARE PLATFORM

The XC4VSX series of Virtex®-4 FPGA are used for the synthesis and simulation. It is one of most advanced FPGA families in industry, produced by Xilinx [13]. The Virtex FPGA comprises two major configurable elements: configurable logic blocks (CLBs) and input/output blocks (IOBs). CLBs provide the functional elements for constructing logic and IOBs provide the interface between the package pins and the CLBs.

The elementary programmable logic block in Xilinx FPGAs is also called a *slice* [14]. This Virtex-4 FPGA slice includes: Two 4-input LUTs (Look-Up Tables) that can implement any 4-input Boolean function, Two dedicated user-controlled multiplexers for combinational logic, Dedicated arithmetic logic and Two 1-bit registers that can be configured either as flip-flops or as latches. The simplified diagram of a Virtex-4 slice is presented in Figure 3.10.

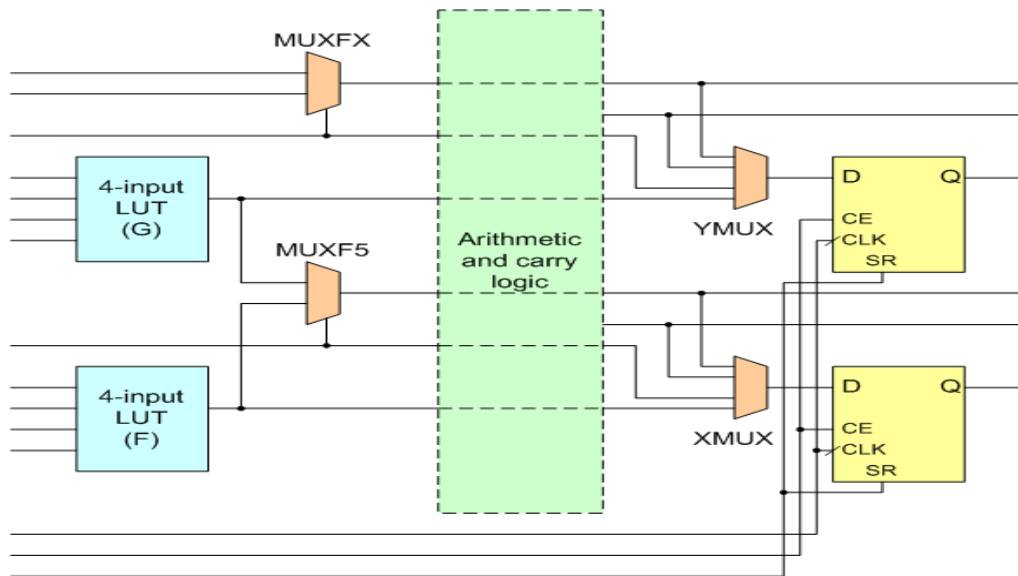


Figure 3.10: A simplified diagram of a Xilinx Virtex-4 FPGA slice.

3.3.3. FPGA DESIGN FLOW

The system design can be processed using the ISE™ Basic Flow as follows:

1. Write the functional description of the design.
2. Write the VHDL modeling of the design.
3. Synthesize the design using Xilinx XST.
4. Run behavioral simulation (also known as RTL simulation).
5. Implement the design: Run the Implement Design process on the top module design, which automatically runs the following processes:
 - Translate, Map and Place and Route

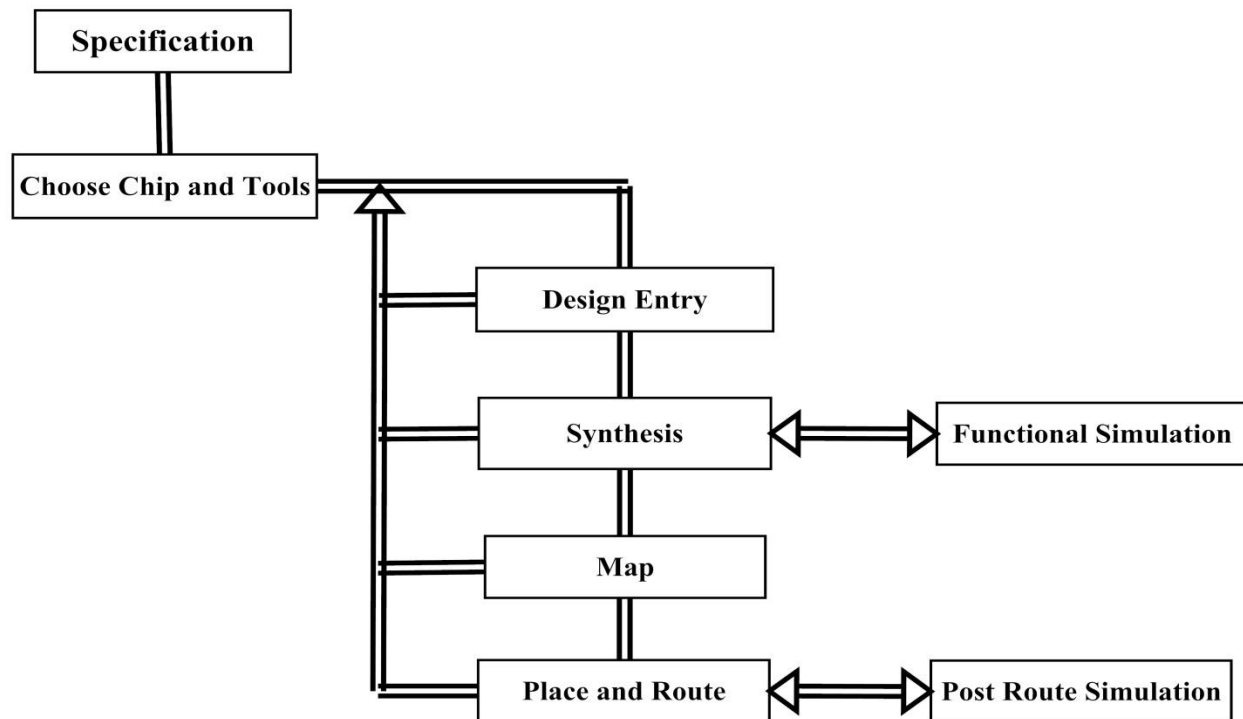


Figure 3.11: FPGA Design Flow

3.3.4. VHDL DESIGN REVIEW

For the sake of modular realization, the whole system is partitioned into three stages. The first stage computes Tetrolet transform coefficients of the input image frame and the thresholding of the detailed Tetrolet Coefficients. The second stage operates on this result to complete the rest of

the image de-noising algorithm. The third and the final stage is the first order gradient based edge detection operator, which works on the output of the second stage.

STAGE ONE: Generally the proposed algorithm of the DFTT is divided into three phases for each Tetrolet partitions: initializing phase, horizontal Pixel phase, and vertical Pixel phase. During the initializing phase the user must provide FTT-2D module with necessary information of the input image. This information is presented to the DTT-2D module through *Control Bus*. Then DTT-2D module apply the internal reset signal *in_Reset* at the DTT-1D then provide it with the necessary information (through *in_Control Bus*) to perform horizontal or vertical pixel, During the Horizontal Pixel and Vertical Pixel phases, the Low and High components for FTT ($L(i)$ and $H(i)$) are calculated. The calculations are performed according to the following equations. To avoid the extra calculation, $\sqrt{2}$ is replaced by 2 for the VHDL description. Note that divide by 2 is implemented as shift to right.

$$L(i) = f(X(i)) = \frac{(X(i) + X(i+1))}{\sqrt{2}} \quad (3.21)$$

$$H(i) = g(X(i)) = \frac{(X(i) - X(i+1))}{\sqrt{2}} \quad (3.22)$$

After decomposing the input image in to High pass and Low pass coefficients for the given Tetrolet partitions, DTT-1D module generates *in_Ready* signal to the DTT-2D module which in turn provide the DTT-1D module with the information for the next line if any. If all lines in the Horizontal Pixel are finished then the process of Vertical Pixel is initiated. All the High pass coefficients will be stored after they are subjected to threshold using the universal hard thresholding method [6].

If a new Tetrolet Partition is required, the customized version of the above process is repeated. The DTT-2D module reinitializes the DTT-1D module with the necessary information of the new image to start new horizontal and vertical phases. When no more partitions are exist, DTT-2D generate *Ready* signal to the outside indicating process end.

For the hardware design of the forward Tetrolet transform image decomposition, only the first three of the Tetrolet partitions are used.

A modified Tetrolet based image de-noising for real-time edge detectors

The overall design in this stage is composed of two-dimensional DTT module (DTT-2D) and the thresholding step designed as synthesizable VHDL code, The DTT-2D module is composed of one-dimensional DTT module (DTT-1D) which represents the main part of the design.

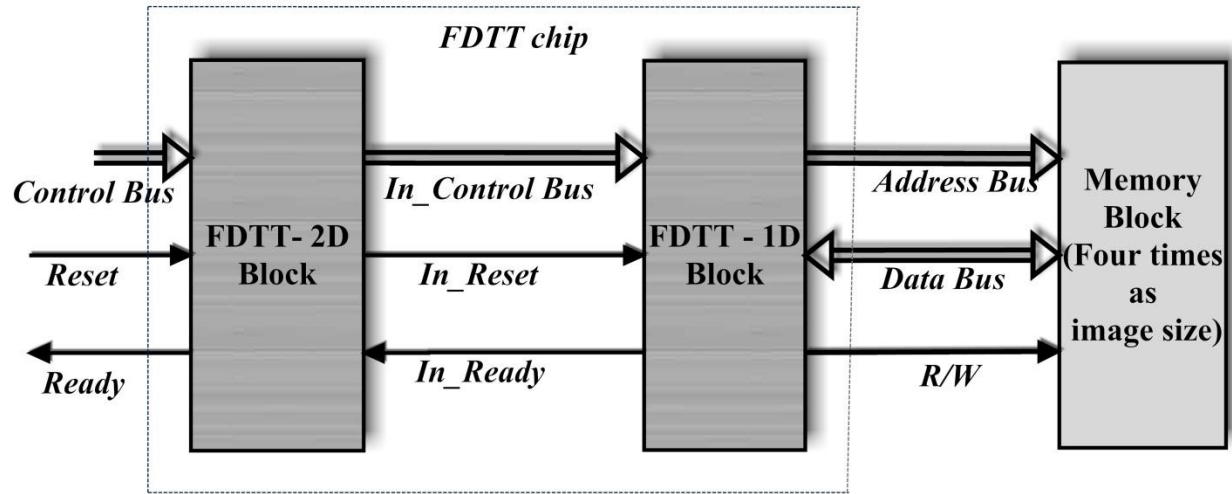


Figure 3.12: Layout of the proposed architecture for the FDTT

STAGE TWO: In this stage, the inverse Tetrolet transform image reconstruction is implemented. The design of the IDTT and FDTT are almost identical except the thresholding step is excluded here. The original pixels values could be retrieved in IDTT process from the High and Low components according to the following equations below. $\sqrt{2}$ are replaced by 2 to avoid extra calculation.

$$X(i) = \frac{(L(i)+H(i))}{\sqrt{2}} \quad (3.23)$$

$$X(i + 1) = \frac{(L(i)-H(i))}{\sqrt{2}} \quad (3.24)$$

STAGE THREE: The third stage mainly comprises of the Sobel edge detection algorithm. The VHDL implementation of this algorithm incorporates the convolution of the 3x3 masks with image pixels, the gradient magnitude and the direction of the intensity variation, the comparison of the magnitude value with the threshold value (useful hints for the Sobel edge detection VHDL implementation in [20]).

3.3.5. SIMULATION WITH TEST BENCHES

The simulation results are produced by integrating ModelSim SE 10.0b and Xilinx ISE environment. There are three test bench configurations provided for each stage for simulation purpose only. Each uses input Text file to initialize memory. Configuration of FDTT or IDTT instances is performed, supplying size of the picture and number of levels of decomposition. The result of the operation is written to a text file.

Memory unit needs to be of size four times as the image size for three Tetrolet partitions, one for the original image and three for the Tetrolet high pass and low pass coefficients. Note that as the Tetrolet partition increases the size of the memory unit also increases. A memory was modeled using non-synthesizable VHDL code for simulation only. Memory *Read* and *write* processes are each given three clock cycles.

In addition, a MATLAB file is used, which can read an image in '(bmp, jpg, gif...)' format and store its contents in 'txt' format. The textual files will be used as input for test bench configurations.

3.4. SUMMARY

Wavelet transform, due to its excellent localization property, has rapidly become an indispensable signal and image processing tool for a variety of applications, including compression and de-noising [6, 7, 8].

Tetrolet transform based image de-noising algorithm is first proposed by Singh et al [3] is a signal estimation technique that exploits the capabilities of Tetrolet transform for signal de-noising. It removes noise by killing coefficients that are insignificant relative to some threshold value. It involves three steps: a linear forward Tetrolet transform, nonlinear thresholding step and a linear inverse Tetrolet transform.

In last part of this section, a hardware description for discrete Tetrolet transform (DTT) module in VHDL language is presented. The module performs discrete Tetrolet transform to images of size 128X128.

CHAPTER 4

RESULT AND DISCUSSIONS

4.1 INTRODUCTION

A set of standard images are used to test the developed system. The results of the test can be classified in to two categories. Which are the result of image de-noising and the result of the combination the Tetrolet based image de-noising and the edge detection algorithm.

4.2. IMAGE DE-NOISING RESULT

Four standard test images (Lena, Boat, Barbara, and Cameraman) are corrupted with different image noises types such as white Gaussian Noise, Speckle noise and Poisson noise. These noisy images are then de-noised using various de-noising algorithms, including the one proposed by this paper. The result is compared based on the performance criteria listed in Section 4.2.1. The white Gaussian noise added to the image is varied with the standard deviation ranging from 10 to 30. The speckle noise added to the image is also varied with variance ranging from 0.02 to 0.08 and the result is also compared based on the performance criteria. Bigger images of size 512x512 are used for visual comparison.

4.2.1. PERFORMANCE CRITERIA

Different algorithms are compared based on the following criteria:

- Visual comparison and subjective analysis - The noisy and de-noised images are presented for subjective comparison for all types of noise.
- Quantitative comparison and objective analysis - Different algorithms are compared based on the PSNR of the de-noised image. The PSNR is calculated as

$$\text{PSNR}(x, y) = 10 \cdot \log_{10}[\max(\max(x), \max(y))^2 / (|x - y|)^2], \quad (4.1)$$

Where x and y are the clean and noisy samples respectively. Higher PSNR indicates better de-noising performance.

4.2.2. SUBJECTIVE ANALYSIS

I. White Gaussian noise

Four standard test images (Lena, Boat, Barbara and Cameraman) of size 512x512 were corrupted with white Gaussian noise having a standard deviation of 30 is de-noised using different de-noising algorithms. The noisy images as well as the de-noised ones (processed using various methods) are presented in this section for visual inspection.

1. Lena Image



(a) Clean Image



(b) Noisy Image



(c) VisuShrink Hard thresholding



(d) VisuShrink Soft thresholding

db1 SURE thresholding with PSNR = 22.5908



(e) SURE thresholding

db1 Bayes thresholding with PSNR = 23.3885



(f) Bayes thresholding

Redundant Haar with PSNR = 24.5044



(g) Redundant Haar Method

db1 Tetrom thresholding with PSNR = 25.9501



(h) Tetrom de-noising

db1 MTetrom with PSNR = 26.3188



(i) Modified Tetrolet de-noising (new)

Figure 4.1: Lena de-noised Image

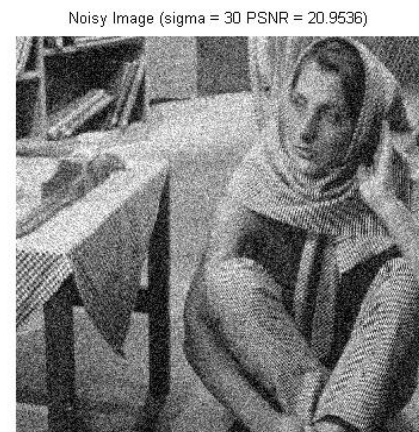
Figure 4.1 shows: Clean Lena image of size 512x512(a), Noisy Lena image (b) and images de-noised by different de-noising methods [c-h] including the method proposed in this paper (i).

De-noised Images (c), (d), (e), (f) and (g) in Figure 4.1 have a higher visual quality compared to the noisy image. Out of the five images (d) and (g) have moderately high visual quality than de-noised images (e) and (f). The Tetrolet based de-noising (h) and (i) have high visual quality than the noisy image and also it have moderately better visual quality than the other wavelet based image de-noising algorithms. The proposed modified Tetrolet based image de-noising and the prior Tetrolet based image de-noising have similar visual appearance.

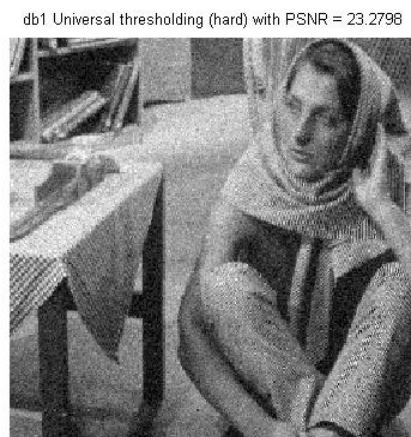
2. The Barbara Image



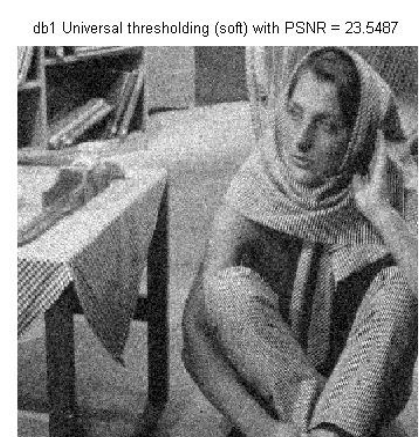
(a) Clean Image



(b) Noisy Image



(c) VisuShrink Hard thresholding



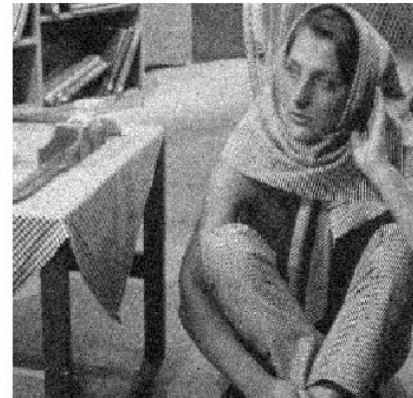
(d) VisuShrink Soft thresholding

db1 SURE thresholding with PSNR = 23.2968



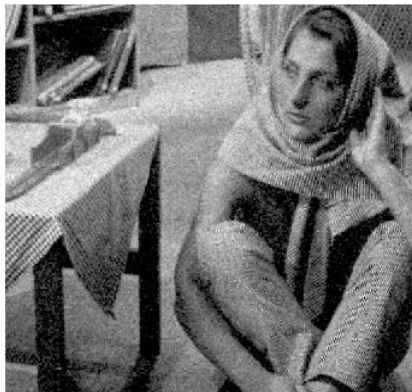
(e) SURE thresholding

db1 Bayes thresholding with PSNR = 23.4972



(f) Bayes thresholding

Redundant Haar with PSNR = 23.2901



(g) Redundant Haar Method

db1 Tetrom with PSNR = 24.7747



(h) Tetrom de-noising

db1 MTetrom with PSNR = 25.1076



(i) Modified Tetrolet de-noising (new)

Figure 4.2: Barbara de-noised Image

Figure 4.2 shows: Clean Barbara image of size 512x512 (a), Noisy Barbara image with noise variance of $\sigma=30$ is added to the clean image (b) and images de-noised by different de-noising methods [c-h] including the method proposed in this paper (i).

De-noised Images (c), (d), (e), (f) and (g) in Figure 4.2 have a moderately higher visual quality than the Noisy images. Images (c) and (g) have fairly better appearance than images (e) and (f). The Tetrolet based image de-noising (h) and (i) have a much higher visual performance compared to the noisy image. It has also a higher visual quality than the other wavelet based methods. The two Tetrolet based methods have a similar visual appearance.

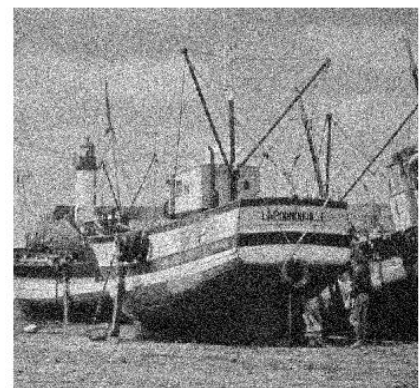
3. The Boat Image

Original Image boat



(a) Clean Image

Noisy Image (sigma = 30 PSNR = 20.7346)



(b) Noisy Image

db1 Universal thresholding (hard) with PSNR = 23.4657



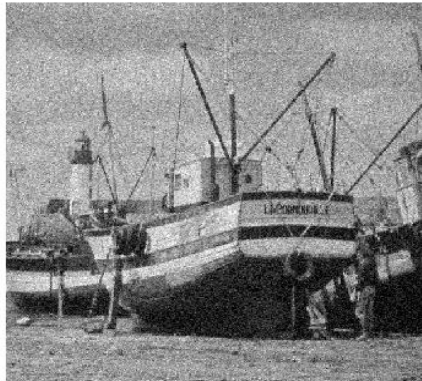
(c) VisuShrink Hard thresholding

db1 Universal thresholding (soft) with PSNR = 23.3105



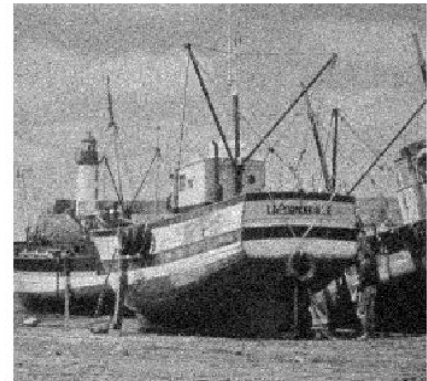
(d) VisuShrink Soft thresholding

db1 SURE thresholding with PSNR = 23.0358



(e) SURE thresholding

db1 Bayes thresholding with PSNR = 23.5962



(f) Bayes thresholding

Redundant Haar with PSNR = 23.8412



(g) Redundant Haar Method

db1 Tetrom thresholding with PSNR = 26.1197



(h) Tetrom de-noising

db1 MTetrom with PSNR = 26.0208



(i) Modified Tetrolet de-noising

Figure 4.3: Boat De-noised Image

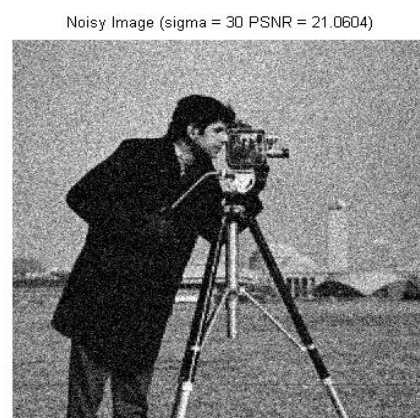
Figure 4.3 shows: Clean Boat image of size 512x512 (a), Noisy Boat image with noise variance of $\sigma=30$ is added to the clean image (b) and images de-noised by different de-noising methods [c-h] including the method proposed in this paper (i).

Images (c), (d), (e), (f) and (g) in Figure 4.3 have a considerably appealing appearance than the noisy image. The Tetrolet based image de-noising (h) and (i) have also higher visual quality compared to the noisy image as well as the wavelet based de-noising algorithms. Concerning the two Tetrolet methods, they have similar visual quality.

3. The Cameraman Image



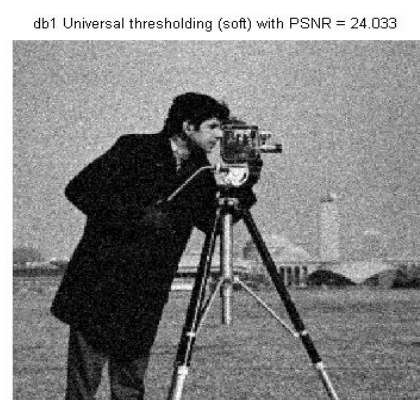
(a) Clean Image



(b) Noisy Image



(c) VisuShrink Hard thresholding



(d) VisuShrink Soft thresholding

db1 SURE thresholding with PSNR = 23.6109



(e) SURE thresholding

db1 Bayes thresholding with PSNR = 24.9151



(f) Bayes thresholding

Redundant Haar with PSNR = 25.2106



(g) Redundant Haar Method

db1 Tetrom thresholding with PSNR = 27.3544



(h) Tetrom de-noising

db1 MTetrom with PSNR = 27.6167



(i) Modified Tetrolet de-noising

Figure 4.4: De-noised Cameraman Image

Figure 4.4 shows: Clean Cameraman image of size 512x512 (a), Noisy Cameraman image with noise variance of $\sigma=30$ is added to the clean image (b) and images de-noised by different de-noising methods [c-h] including the method proposed in this paper (i).

Visually, the de-noised Images (c) and (d) have significantly better compared to the noisy one. The results for the cameraman image are similar to the ones obtained for the Lena and Boat images. The best image in terms of visual quality is obtained by previous Tetrolet based methods. The second best image is produced by the method proposed in this thesis, which is visually appealing than other wavelet based methods. This is due to, the previous Tetrolet transform de-noising have more number of Tetrolet coefficients than the new one.

Note That: having a redundant coefficient favors image de-noising [1].

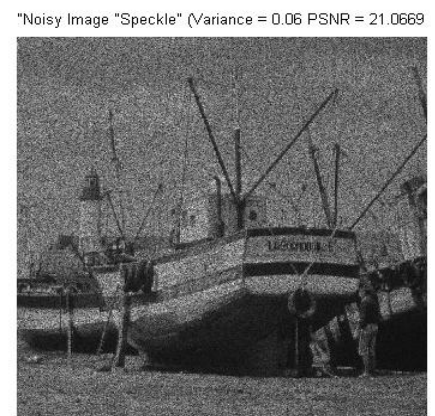
II. Subjective analysis with Speckle noise

Two standard test images (Boat and Cameraman) of size 512x512 were corrupted with Speckle noise having a variance of 0.06 are selected. The noisy images as well as the de-noised ones, processed using various methods, are presented in this section for visual inspection.

1. Boat Image



(a) Clean Image



(b) Noisy Image

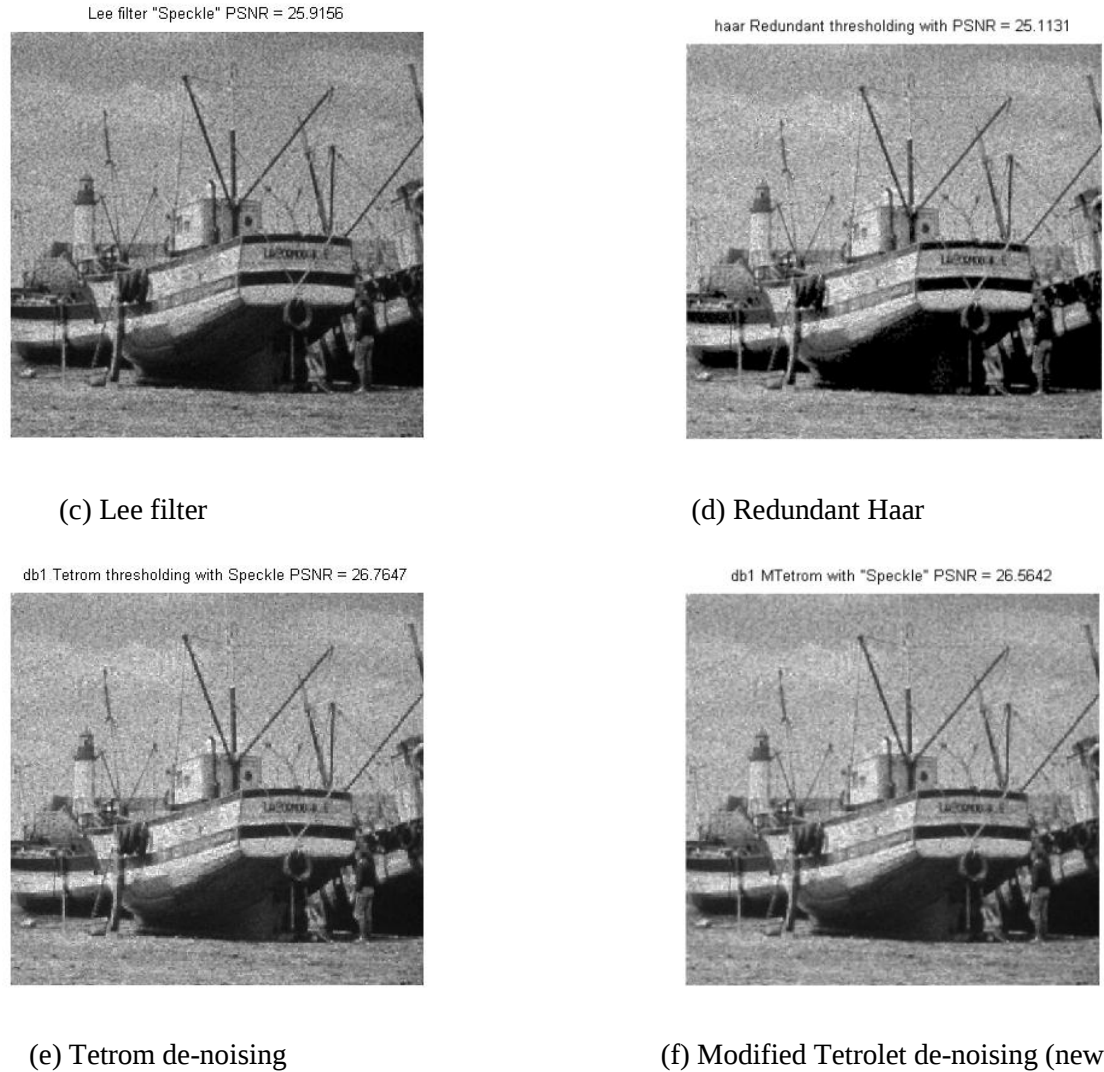


Figure 4.5: De-noised Boat image

Figure 4.5 shows: Clean Boat image of size 512x512 (a), Noisy Boat image with noise variance of $\sigma = 0.6$ is added to the clean image (b) and images de-noised by different de-noising methods [c-e] including the method proposed in this paper (f).

The noisy image have much lesser visual quality than the de-noised images(c, d, e and f), however the visual appearance of these de-noised images are not still impressive. The Tetrolet based de-noising shows a fairly better visual appearance than the Lee filter and the redundant Haar method. Modified Tetrolet method has a similar visual quality when compared with the previous Tetrolet method.

3. Cameraman Image



(a) Clean Image



(b) Noisy Image



(c) Lee filter



(d) Redundant Haar



(e) Tetrom de-noising



(f) Modified Tetrolet de-noising (new)

Figure 4.6: De-noised Cameraman image (Speckle noise)

Figure 4.6 shows: Clean Cameraman image of size 512x512 (a), Noisy Cameraman image with noise variance of $\sigma=0.6$ is added to the clean image (b) and images de-noised by different de-noising methods [c-e] including the method proposed in this paper (f).

The Tetrolet based de-noising have moderately higher visual quality than the noisy image and fairly better visual quality compared to the Lee filter and the redundant Haar method [19]. The previous Tetrolet and the new modified Tetrolet transform based image de-noising have similar performance in terms of visual quality.

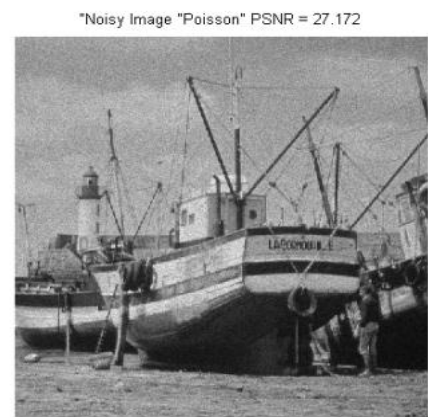
III. Subjective analysis with Poisson noise

Two standard test images (Boat, and Cameraman) of size 512x512 were corrupted with Poisson noise and are selected. The noisy images as well as the de-noised ones, processed using various methods, are presented in this section for visual inspection.

1. Boat Image Example



(a) Clean Image



(b) Noisy Image

Fast Interscale wavelet De-noising "Poisson" PSNR = 32.2331



(c) Fast interscale wavelet de-noising

haar Redundant with PSNR = 27.4665



(d) Redundant Haar

VST.Modified Tetrom "Poisson" PSNR = 31.3427



(e) Tetrolet with variance stabilization

NI.Modified Tetrom "Poisson" PSNR = 31.2043



(f) Nonlinear transient Tetrolet bases

Figure 4.7: De-noised Boat image (Poisson noise)

Figure 4.7 shows: Clean Boat image of size 512x512 (a), Noisy Boat image with Poisson noise is added to the clean image (b), images de-noised by different de-noising methods [c-d] and including the method proposed in this paper [e-f].

The Tetrolet based de-noising with nonlinear invariant basis have a higher visual quality compared to the noisy image, however, it has a lesser visual appearance than the fast interscale wavelet de-noising. The redundant Haar de-noising is not a good method for Poisson noise.

2. Cameraman Image Example

Original Image cameraman



(a) Clean Image

"Noisy Image "Poisson" PSNR = 28.377



(b) Noisy Image

"Noisy Image "Poisson" PSNR = 28.377



(c) Fast interscale wavelet de-noising

Redundant Haar with PSNR = 28.4694



(d) Redundant Haar

VST.Modified Tetrom "Poisson" PSNR = 31.7365



(e) Tetrolet with variance stabilization

NI.Modified Tetrom "Poisson" PSNR = 32.1017



(f) Nonlinear transient Tetrolet bases

Figure 4.8: De-noised Cameraman image (Poisson noise)

Figure 4.8 shows: Clean Cameraman image of size 512x512 (a), Noisy Cameraman image with Poisson noise is added to the clean image (b) and images de-noised by different de-noising methods including the method proposed in this paper [c-f].

The Tetrolet based de-noising with nonlinear invariant basis have a higher visual quality compared to the noisy image, However it has a lesser visual appearance compared to the fast interscale wavelet de-noising. In terms of visual quality, the best image is produced by the fast interscale wavelet de-noising method.

4.2.3. OBJECTIVE ANALYSIS

I. Gaussian noise: The four test images (Lena, Barbara, Boat and Cameraman) were corrupted with white Gaussian noise, and de-noised using different methods. The variance of the white noise is varied from 10 to 30 in steps of 5. The results are the PSNR values averaged over 6 runs. They are presented in Table 4.1, which compares the proposed algorithm with other algorithms such as VisuHard[6], VisuSoft[6], Sure[7], Bayes[8] and redundant Haar[19].

Image	Noisy	VisuHard	VisuSoft	Sure	Bayes	R-haar	Tetrom	Modified Tetrolet
Lena($\sigma=10$)	28.0693	29.079	30.1427	29.4865	30.246	31.9689	31.205	31.1239
Lena($\sigma=15$)	24.9655	27.312	27.4382	27.5498	27.6374	29.5459	29.0147	28.9635
Lena($\sigma=20$)	23.0697	25.6803	25.5598	25.6155	25.9019	27.8561	27.9254	27.6542
Lena($\sigma=25$)	21.7065	24.6571	24.1289	23.8845	24.5491	26.0929	26.8638	26.7565
Lena($\sigma=30$)	20.6621	23.632	23.0409	22.5908	23.3885	24.5044	26.3188	25.9501
boat($\sigma=10$)	27.9854	28.8087	29.6792	28.7772	29.6312	31.1266	30.5671	30.5299
boat($\sigma=15$)	24.9051	26.4414	26.9967	27.0844	27.1592	28.2385	28.0647	28.4175
boat($\sigma=20$)	22.8265	24.8729	25.1729	25.2261	25.5067	26.2588	26.8211	27.1519
boat($\sigma=25$)	21.2891	23.9003	23.8004	23.5379	24.3206	24.8581	25.7944	26.0731
boat($\sigma=30$)	20.0972	23.123	22.7065	22.0943	23.3851	23.8412	24.8993	24.7657
Cameraman($\sigma=10$)	28.5164	30.2152	30.6564	29.9255	30.5437	31.1933	32.3665	32.4179
Cameraman($\sigma=15$)	25.2854	27.8872	27.7153	27.87	27.8698	28.9743	30.0932	29.7179
Cameraman($\sigma=20$)	23.0681	26.2608	25.6806	25.7306	25.9565	27.4975	28.2035	28.005
Cameraman($\sigma=25$)	21.4972	24.4879	24.0977	23.8321	24.4287	26.3213	26.7483	26.5949
Cameraman($\sigma=30$)	20.3223	22.97	22.8984	22.372	23.3155	25.2106	25.5646	25.5227
Barbara($\sigma=10$)	28.2917	27.8177	28.4493	27.7718	29.4684	30.2986	29.9856	29.8586
Barbara($\sigma=15$)	25.3559	25.2711	26.7708	26.4819	26.8245	27.4659	27.4278	27.5487
Barbara($\sigma=20$)	23.406	23.7232	25022	25.0136	24.9582	25.7776	25.9964	26.1233
Barbara($\sigma=25$)	21.9841	22.9104	23.7516	23.6854	23.4914	24.4333	25.2111	25.3271
Barbara($\sigma=30$)	20.8878	22.0513	22.7496	22.5061	22.3024	23.0516	24.1515	24.6833

Table 4.1: PSNR (in dB) Performance Table

The following observations can be drawn from these results:

- For Lena, Barbara and Boat images the Tetrolet method performs, on an average, up to 2 dB better PSNR compared with the VisuHard, VisuSoft, Sure and Bayes. It performs up to 0.5 dB better compared to the redundant Haar. By decomposing the image further additional optimization is possible. Since other methods uses only one level of decomposition, the proposed method has been kept to the same level of decomposition for the sake of uniformity.
- The redundant Haar method results in a slightly better PSNR compared to the proposed method, when the noise variance is 10 and 30 with the exception of cameraman image.
- Regarding Cameraman image, the Tetrolet method performs, on an average, up to 2.6 dB better PSNR compared with VisuHard, VisuSoft, Sure, Bayes and redundant Haar. It also performs up to 1.9 dB better PSNR compared to best of the above methods.
- The Modified Tetrolet based de-noising, which is proposed in this paper, have a similar performance with the previous Tetrolet based de-noising algorithm; However, the newly proposed algorithm sometimes has a slightly better performance in terms of PSNR than the previous Tetrom method[3]. This is due to the fact that, the thresholding used for the newly proposed algorithm are modified so as to produce a better PSNR.

The performance graphs in Figures 4.9 and 4.10 compares the proposed method with other methods. The performance of the proposed method is greater than other de-noising methods in terms of PSNR for Cameraman image. However, the modified Tetrolet de-noising method, which is proposed in this thesis, performs equally with the previous Tetrolet based de-noising. The fact that the proposed method reduces the number of redundant coefficients is compensated using an improved threshold scheme.

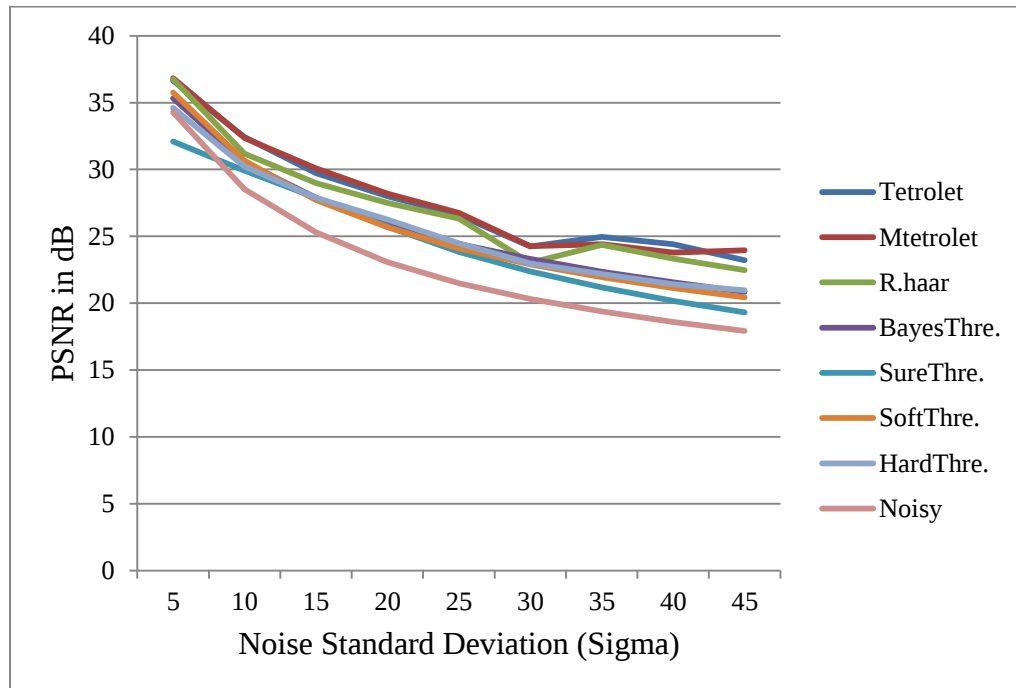


Figure 4.9: Performance Comparison with Different Methods - Cameraman Image

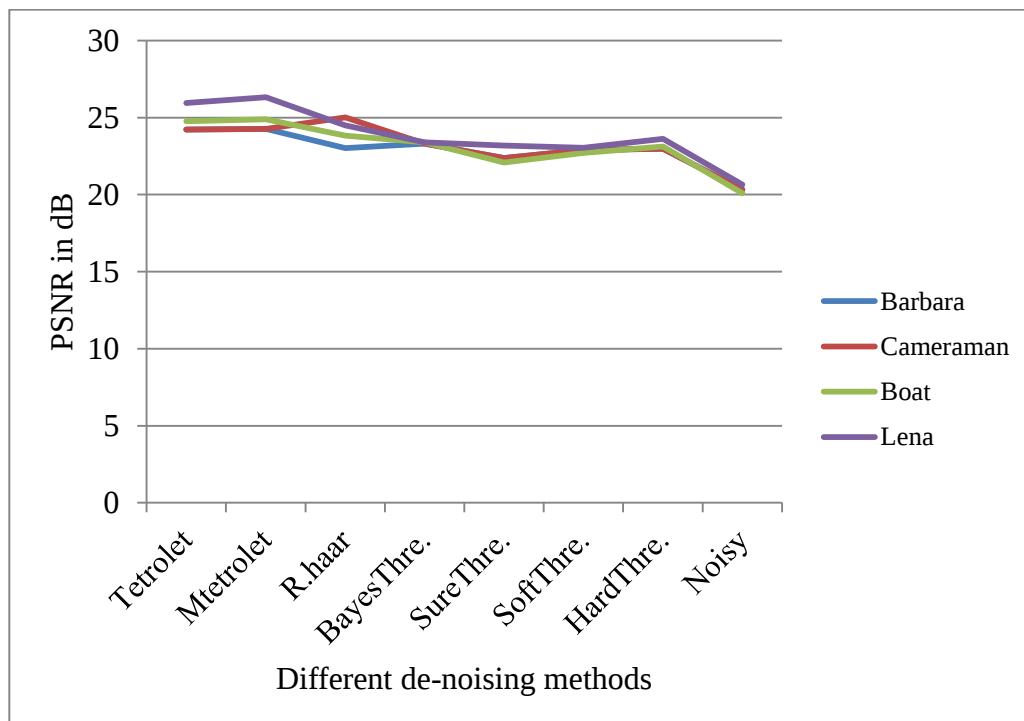


Figure 4.10: Performance Comparison with Different Methods with different Images

II. Objective analysis with Speckle noise

The two test images (Boat and Cameraman) were corrupted with Speckle noise and de-noised using different methods. The variance of the Speckle noise is varied from 0.02 to 0.08. The results are the PSNR values averaged over 6 runs. They are presented in Table 4.2, which compares the proposed algorithm with other algorithms such as Lee Filter, redundant Haar method and Tetrom method.

- The Tetrolet method performs, on an average, up to 1 dB better when compared with the Lee Filter and redundant Haar method for larger noise variance.
- The new algorithm has a similar performance with the previous Tetrolet based de-noising algorithm in terms of PSNR except when the noise variance is 0.02.

Image	Noisy Image	Lee filter	Redundant Haar	Tetrolet method	Modified Tetrolet
Boat(v=0.02)	23.5529	24.9564	24.0963	25.3317	25.2047
Boat(v=0.04)	21.3364	24.6954	23.9617	24.8503	24.8497
Boat(v=0.06)	21.5671	23.4792	25.4068	25.5637	26.1728
Boat(v=0.08)	20.1759	23.5159	23.6613	25.2616	25.4792
Cameraman(v=0.02)	25.5251	25.7311	27.7228	26.3822	26.3473
Cameraman(v=0.04)	23.1082	24.9899	25.5379	25.4805	25.5353
Cameraman(v=0.06)	22.5792	24.425	26.6471	27.157	27.2002
Cameraman(v=0.08)	21.5606	23.9257	24.9251	26.0406	26.4366

Table 4.2: PSNR (in dB) Performance Table

The performance graphs in Figures 4.11 and 4.12 compares the proposed method with other methods. From these figures it can be observed that,

- The Lee filter has a better performance in terms of PSNR for Boat image for lower values of noise variance ($v = 0.02$), whereas for cameraman image the redundant Haar surpass.
- The performance of the proposed method is less than the redundant Haar when the amount of noise is small in the case of Cameraman image.
- Modified Tetrolet based de-noising have a better PSNR for higher values of noise variances for the two test images.

- On the other hand the modified Tetrolet de-noising method, which is proposed in this thesis, performs equally with the previous Tetrolet based de-noising.

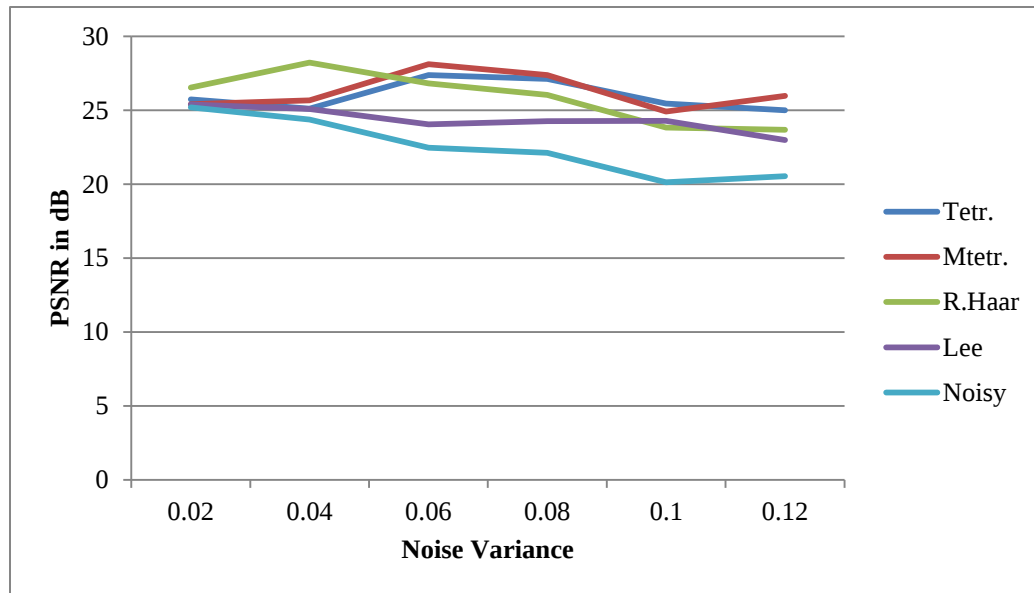


Figure 4.11: Performance Comparison with Different Methods - Cameraman Image

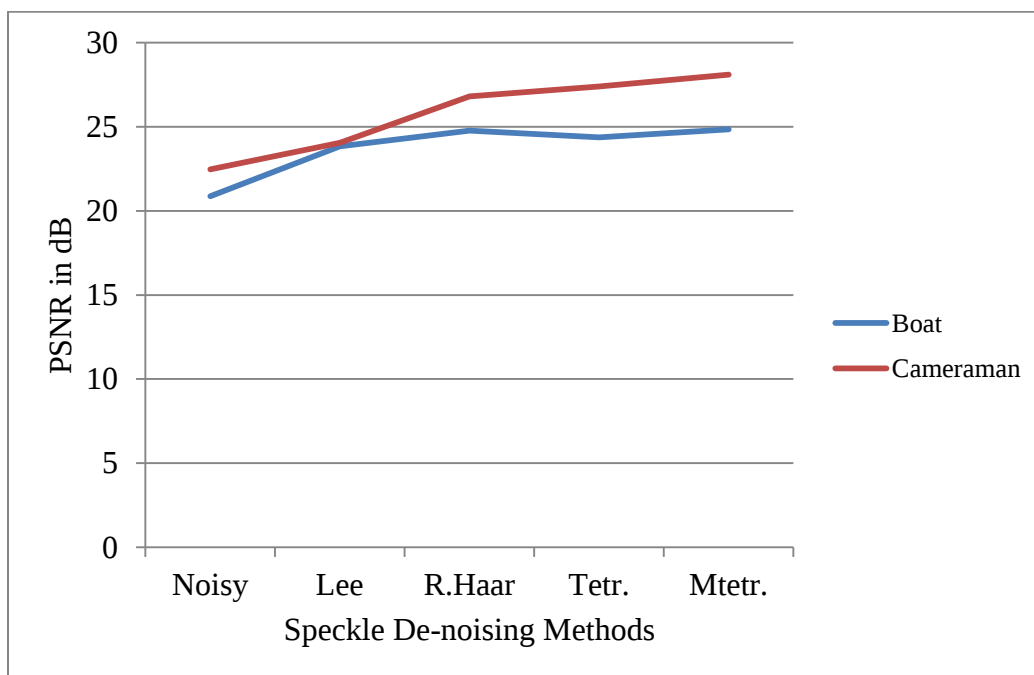


Figure 4.12: Performance Comparison with Different Methods (Speckle $v = 0.06$)

II. Objective analysis with Poisson noise

The two test images (Boat and Cameraman) were corrupted with Poisson noise, and de-noised using different methods. The results are the PSNR values averaged over 6 runs. They are presented in Table 4.3, which compares the Tetrolet based de-noising algorithms with other algorithms such as fast interscale wavelet de-noising [18] and redundant Haar de-noising method [19] for images with Poisson noise.

Image	Noisy Image	fast interscale wavelet de-noising	Mod. Tetrolet de-noising	Mod. Tetrolet de-noising with VST	Redundant Haar de-noising	Mod. Tetrolet de-noising with NLI
Boat	27.191	30.3228	26.0193	29.7922	27.2122	29.7108
Cameraman	28.3831	32.9829	27.7962	32.0254	28.409	32.1864

Table 4.3: PSNR (in dB) Performance Table

The following observations can be drawn from this table:

- Tetrolet de-noising by itself does not give a good result for images corrupted with Poisson noise. Tetrolet de-noising with nonlinear invariant bases performs, on an average, up to 3 dB better when compared with the redundant Haar method.
- The Tetrolet de-noising with nonlinear invariant has a similar performance with fast interscale wavelet de-noising, which is a state of the art Poisson noise de-noising algorithm.
- Generally, the modified Tetrolet transform with variance stabilization method have relatively a better performance in terms of PSNR than the Tetrolet transform with nonlinear invariant basis.

The performance graphs in Figure 4.13 compares the proposed method with other methods. The modified Tetrolet de-noising with nonlinear invariant basis exhibits an equal performance as the fast interscale wavelet de-noising. The cameraman image has a higher PSNR values than other images for the Poisson noise de-noising methods.

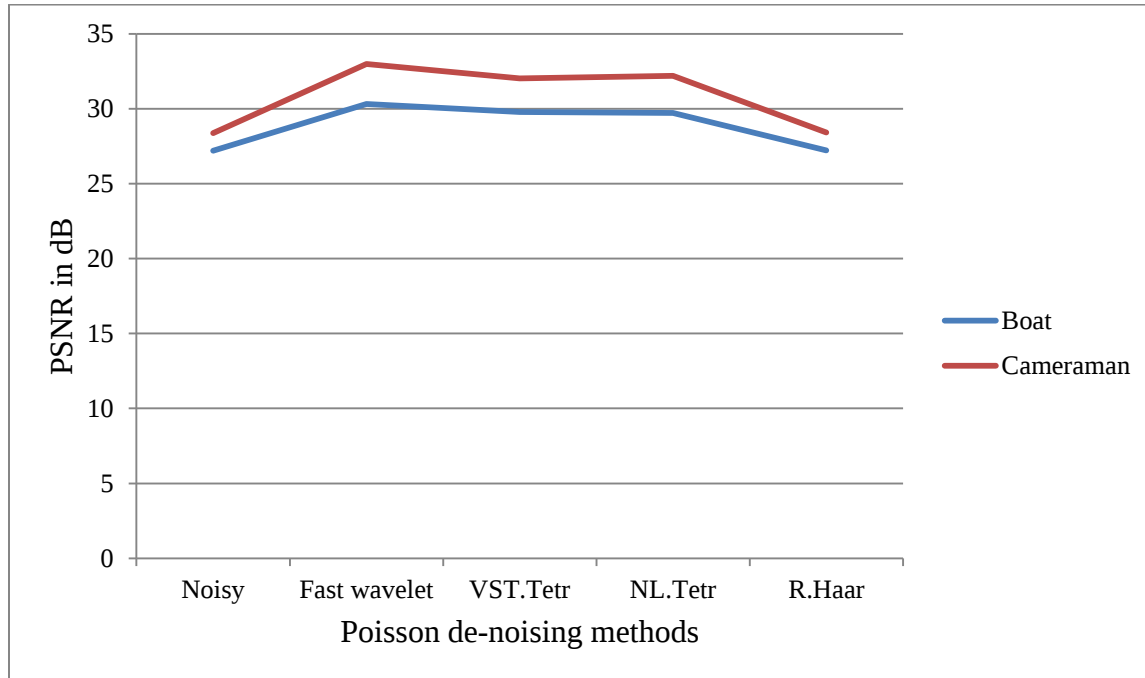


Figure 4.13: Performance Comparison with Different Methods (Poisson)

4.3. EDGE DETECTION AFTER IMAGE DE-NOISING

A standard Cameraman image was used for different de-noising schemes for edge detectors, including the one proposed by us. The result is compared based on the performance criteria listed in Section 4.3.1. Two types of the edge detection methods have been analyzed, which are the Sobel edge detection algorithm and the Roberts edge detection algorithm.

4.3.1. PERFORMANCE CRITERIA

The edge detection algorithms are compared based on the following criteria:

- Visual comparison and subjective analysis – The edge detection for the clean image and the de-noised images are presented. The quality of these images will be evaluated based on their appearance.
- Quantitative comparison and objective analysis - Different algorithms are compared based on the PSNR of the de-noised image. The PSNR is calculated as

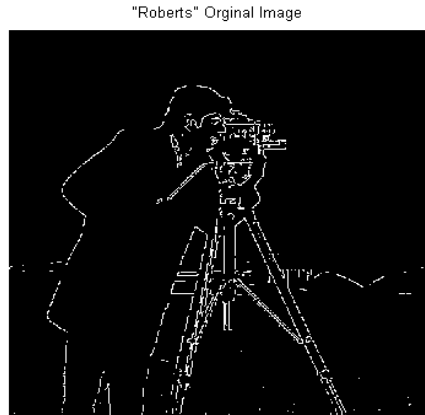
$$\text{PSNR}(x, y) = 10 * \log_{10} \frac{\max(\max(x), \max(y))^2}{\|x - y\|^2}, \quad (4.2)$$

Where x and y are the results of edge detection of clean image and edge detection of de-noised image, respectively.

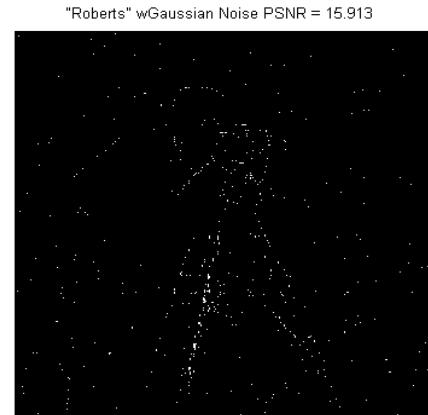
4.3.2. SUBJECTIVE ANALYSIS

I. Subjective analysis of Roberts edge detection

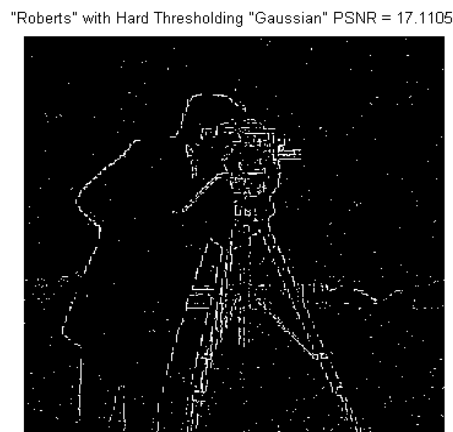
A. White Gaussian noise: The cameraman test images (with a size of 256x256) were corrupted with White Gaussian noise and de-noised using different de-noising methods and then it will be subjected to the Roberts edge detection operator. The results are the PSNR values averaged over 6 runs. Different white noise de-noising methods combined with the Roberts edge detection operator have been compared



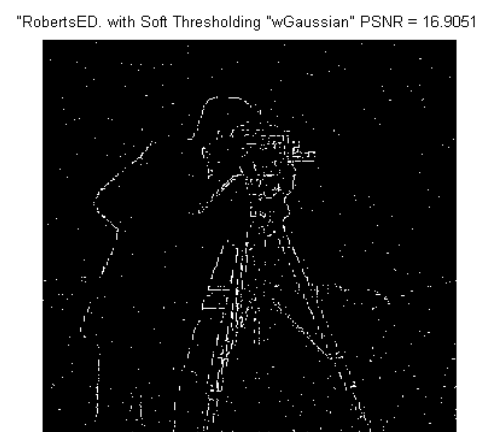
(a) Roberts operator with Clean Image



(b) Roberts operator with Noisy Image



(c) Roberts operator with VisuShrink (Hard)



(d) Roberts operator with VisuShrink (Soft)

"Roberts" with SURE thresholding "Gaussian" PSNR = 16.7503



(e) Roberts operator with SURE thresholding

"Roberts" with Bayes thresholding "Gaussian" PSNR = 17.1957



(f) Roberts operator with Bayes thresholding

"Roberts" Redundant haar "wGaussian" PSNR = 15.9747



(g) Roberts operator with Redundant Haar

Roberts "Tetrom" wGaussian PSNR = 16.4235



(h) Roberts operator with Tetrom

"RobertsED. with Modified Tetrom" "Gaussian" PSNR = 17.2985



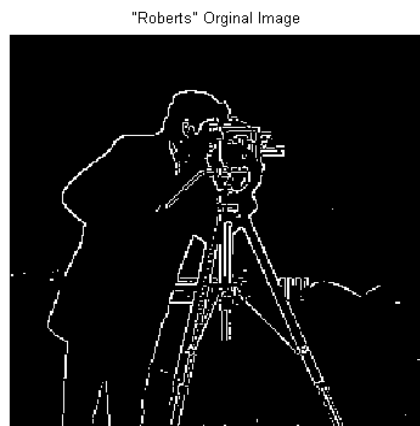
(i) Roberts operator with modified Tetrolet de-noising

Figure 4.14: Cameraman De-noised Image with Roberts Edge detection (Gaussian)

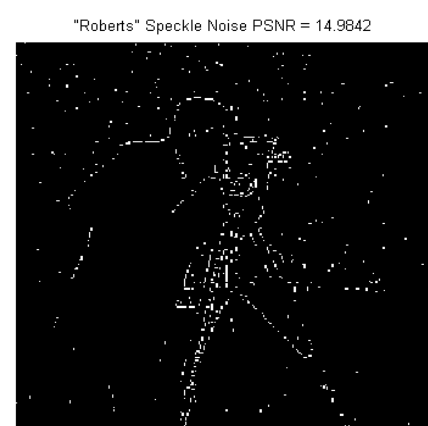
Figure 4.14 shows: Clean Cameraman image with Roberts Edge detection of size 512x512 (a), Noisy Cameraman image with noise variance of 30 is added to the clean image and Roberts edge detection is applied to it. (b) and Roberts edge detection after images de-noised by different de-noising methods [c-h] including the method proposed in this paper (i).

As we can see clearly, out of the seven de-noising algorithms, three of the algorithms such as Hard Threshold before Roberts, Bayes threshold before Roberts and the modified Tetrolet transform de-noising before Roberts have a better performance in terms visual quality. Out of the three the modified Tetrolet based de-noising before Roberts edge detection scored the first rank in terms of subjective analysis. Still the other methods have a comparable visual quality.

2. Speckle noise: The cameraman test images (with a size of 256x256) were corrupted with Speckle noise, and de-noised using different speckle noise de-noising methods and then the image will feed in to the Roberts operator. The results are the PSNR values averaged over 6 runs. Various Speckle noise de-noising methods combined with the Roberts edge detection operator have been compared.



(a) Roberts operator with Clean Image



(b) Roberts operator with Noisy Image

"Roberts" Lee filter "Speckle" PSNR = 14.9385



(c) Roberts with fast Lee filter de-noising

"Roberts" Redundant haar "Speckle" PSNR = 15.2615



(d) Roberts operator with Redundant Haar

Roberts with "Tetrom" Speckle PSNR = 15.1828



(e) Roberts operator with Tetrom

"Roberts" Modified Tetrom "Speckle" PSNR = 15.0494



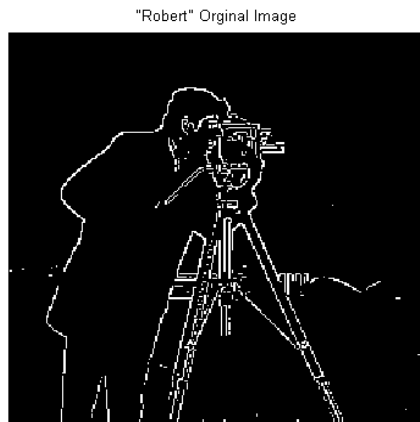
(f) Roberts operator with modified Tetrolet

Figure 4.15: Cameraman De-noised Image with Roberts Edge detection (Speckle)

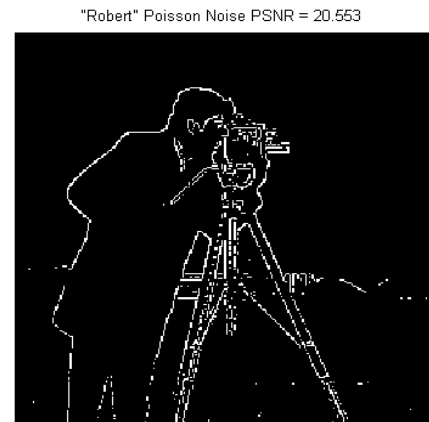
Figure 4.15 shows: Clean Cameraman image with Roberts Edge detection of size 256x256 (a), Noisy Cameraman image with noise variance of 0.6 is added to the clean image and Roberts edge detection is applied to it. (b) and Roberts edge detection after images de-noised by different Speckle noise de-noising methods [c-e] including the method proposed in this paper (f).

The better visual quality is scored when using the previous Tetrolet method (Tetrom) and the Tetrolet based algorithm. However, the visual comparison of these two methods with the Lee filter is not compatible with the PSNR result. Thus the Lee filter is hardly an image de-noising technique; rather it is an image enhancement technique, which only enables the image to look visually appealing than preserving the images crucial image characteristics.

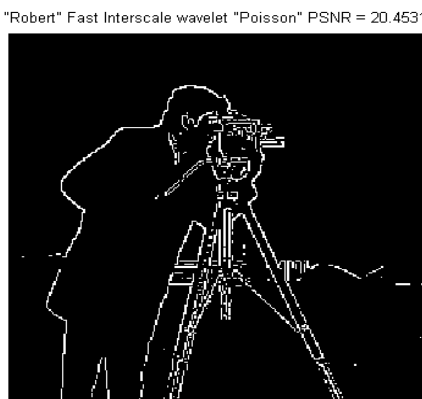
3. Poisson noise: The cameraman test images (with a size of 256x256) were corrupted with Poisson noise, and de-noised using different poisson noise de-noising methods and a Roberts operator will be applied to the de-noised image. The results are the PSNR values averaged over 6 runs. The following de-noising methods combined with the Roberts edge detection operator have been compared.



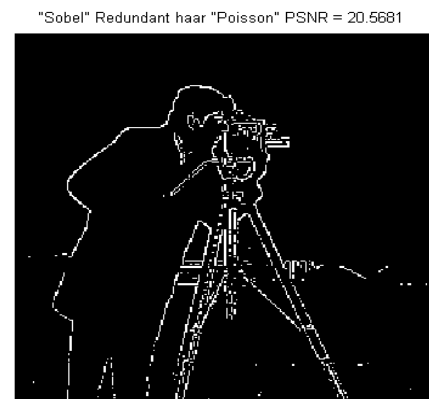
(a) Roberts operator with Clean Image



(b) Roberts operator with Noisy Image

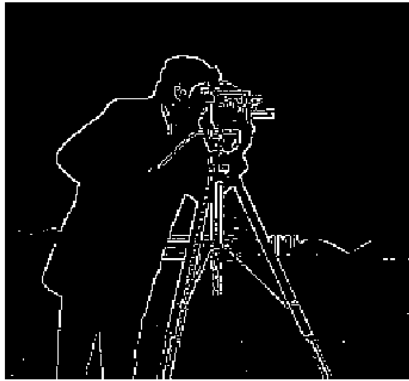


(c) Roberts with fast interscale wavelet de-noising



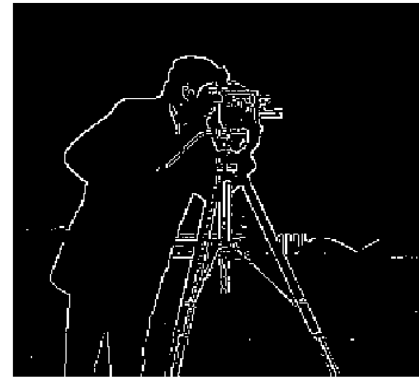
(d) Roberts operator with Redundant Haar

"Robert" VST.Modified Tetrom "Poisson" PSNR = 21.0721



(e) Roberts with Tetrolet de-noising (VST)

"Robert" NI.Modified Tetrom "Poisson" PSNR = 21.407



(f) Roberts with modified Tetrolet de-noising (NLI)

Figure 4.16: Cameraman de-noised Image with Roberts Edge detection (Poisson noise)

Figure 4.16 shows: Clean Cameraman image with Roberts Edge detection of size 256x256 (a), Noisy Cameraman image with Poisson noise is added to the clean image and Roberts edge detection is applied to it. (b) and Roberts edge detection after images de-noised by different Speckle noise de-noising methods [c-d] including the method proposed in this paper [e-f].

Visually, all the edge detected images look appealing, whereas in terms of PSNR the Tetrolet based image de-noising have a PSNR of above 21, specially the Tetrolet transform with nonlinear invariant basis showed a promising result, however still the Fast interscale and the redundant Haar have a competitive result.

II. Subjective analysis with Sobel edge detection

1. White Gaussian noise: The Cameraman test images (with a size of 512x512) were corrupted with Poisson noise, and de-noised using different methods and Sobel operator are used. The results are the PSNR values averaged over 6 runs. The following de-noising methods combined with the Roberts edge detection operator have been compared.

"Sobel" Original Image



(a) Sobel operator with Clean Image

"Sobel" wGaussian Noise PSNR = 15.9946



(b) Sobel operator with Noisy Image

"Sobel" with Hard Thresholding "Gaussian" PSNR = 16.2176



(c) Sobel operator with VisuShrink (Hard)

"Sobel ED. with Soft Thresholding "wGaussian" PSNR = 16.4512



(d) Sobel operator with VisuShrink (Soft)

"Sobel" with SURE thresholding "Gaussian" PSNR = 16.3801



(e) Sobel operator with SURE thresholding

"Sobel" with Bayes thresholding "Gaussian" PSNR = 16.251



(f) Sobel operator with Bayes thresholding

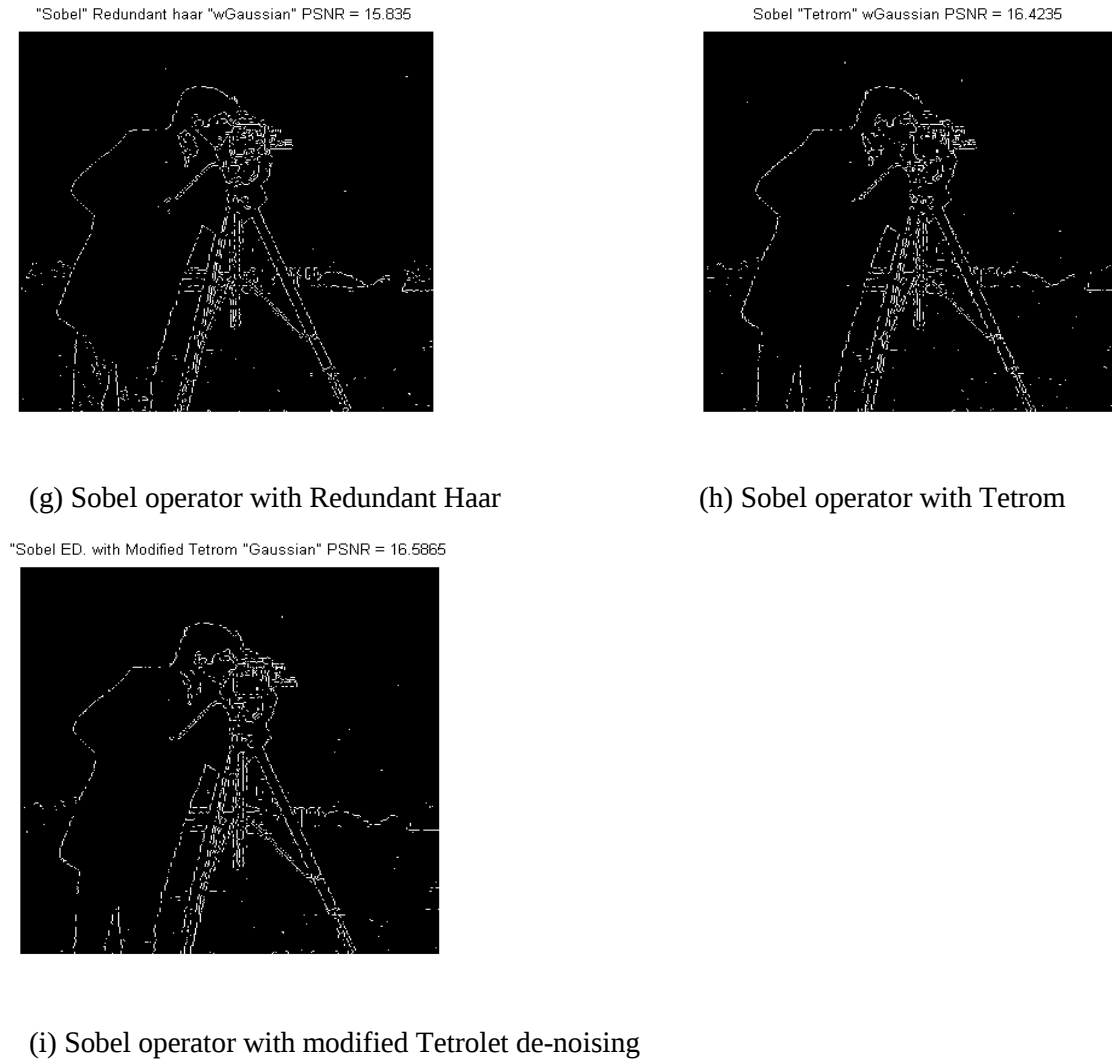
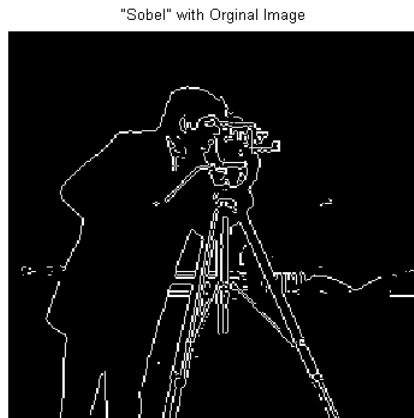


Figure 4.17: Cameraman de-noised Image with Sobel edge detection

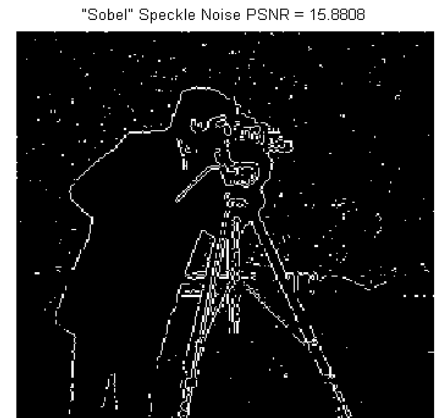
Figure 4.17 shows: Sobel operator for clean image of the cameraman of size 512x512 (a), Sobel operator for Noisy image of the cameraman, noise of variance = 30 (b) and Sobel edge detected de-noised by different de-noising methods including the method proposed in this paper [c-i].

Regarding the Sobel edge detection operator, the Soft threshold before Sobel and the SURE threshold before Sobel scored the highest values in terms of PSNR, however the visual quality is not as good as the Tetrolet based de-noising with Sobel edge detector. The Tetrolet based de-noising with Sobel takes the forth position in terms of PSNR following the noisy image, however the visual appearance is appealing than other de-noising schemes with Sobel edge detection.

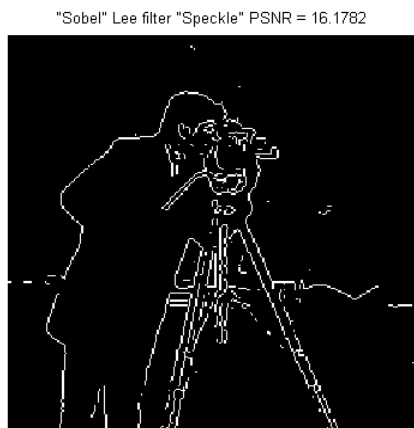
2. Speckle noise: The cameraman test images (with a size of 256x256) were corrupted with Speckle noise subjected to different de-noising methods combined with the Sobel operator. The results are the PSNR values averaged over 6 runs. The following de-noising methods combined with the Sobel edge detection operator have been compared.



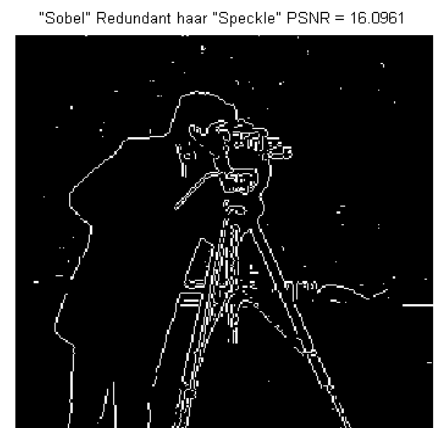
(a) Sobel operator with Clean Image



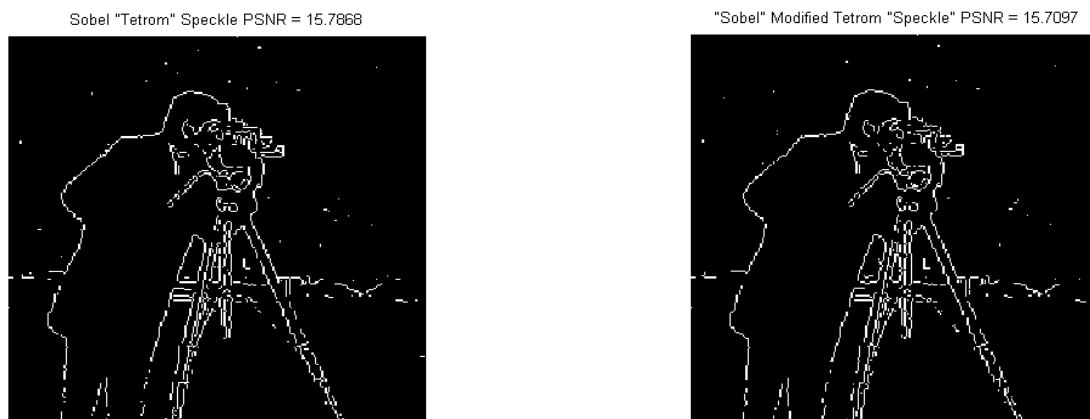
(b) Sobel operator with Noisy Image



(c) Sobel operator with Lee filter



(d) Sobel operator with redundant Haar



(e) Sobel operator with Tetrolet de-noising

(f) Sobel operator with modified Tetrolet de-noising

Figure 4.18: Cameraman De-noised Image with Sobel Edge detection (Speckle)

Figure 4.18 shows: Sobel operator for clean image of the cameraman of size 256x256 (a), Sobel operator for Noisy image of the cameraman, noise of variance = 0.6 of speckle noise (b) and Sobel edge detected images after de-noised by different de-noising methods including the method proposed in this paper [c-f].

The Lee filter before Sobel is visually appealing compared to the other three, the second rank is taken by our modified Tetrolet method with Sobel.

3. Poisson noise: The cameraman test image was (with a size of 256x256) corrupted with Poisson noise, and de-noised using different methods and then Sobel operator are used. The results are the PSNR values averaged over 6 runs.



(a) Sobel operator with Clean Image

(b) Sobel operator with Noisy Image

"Sobel" Fast Interscale wavelet "Poisson" PSNR = 20.3614



(c) Sobel with fast interscale wavelet de-noising

"Sobel" Redundant haar "Poisson" PSNR = 21.2895



(d) Sobel operator with Redundant Haar

"Sobel" VST.Modified Tetrom "Poisson" PSNR = 20.9632



(e) Tetrolet with variance stabilization plus Sobel

"Sobel" NI.Modified Tetrom "Poisson" PSNR = 20.9385



(f) Sobel with nonlinear transient Tetrolet basis

Figure 4.19: Cameraman de-noised Image with Sobel Edge detection (Poisson)

Figure 4.19 shows: Sobel operator for clean image of the cameraman of size 256x256 (a), Sobel operator for Noisy image of the cameraman, Poisson noise is added to it. (b), Sobel edge detected images after de-noised by different de-noising methods [c-d] and the method proposed in this paper [e-f].

The redundant Haar method before Sobel is the best of all the four in terms of PSNR. However the other three methods have a competitive result in terms of visual quality, the Tetrolet transform with variance stabilization, which is proposed in this paper, scored the first rank in terms of visual appearance.

4.2.3. OBJECTIVE ANALYSIS

II. Performance with Roberts edge detection

A. White Gaussian noise: The four test images (Lena, Barbara, Boat and Cameraman) were corrupted with white Gaussian noise, and de-noised using different methods and then the image is fed to the Roberts edge detection operator. The variance of the white Gaussian noise is varied from 10 to 30 in steps of 3. The results are the PSNR values averaged over 6 runs. They are presented in Table 4.4, which compares the proposed algorithm with other algorithms such as Roberts with VisuHard, Roberts with VisuSoft, Roberts with SURE, Roberts with Bayes, and Roberts with redundant Haar method. The following observations can be drawn from these results:

- For noise variances (20 and 30), The Roberts with modified Tetrolet method performs, on an average, up to 0.5 dB better when compared with the Roberts with VisuHard, Roberts with VisuSoft, Roberts with SURE, Roberts with Bayes. However it has approximately similar performance with Roberts with redundant Haar method and Roberts with Previous Tetrom de-noising method.
- Generally, The Roberts with modified Tetrolet method performs better in the case of Cameraman image in terms of PSNR.

Images	Noisy Image	Hard Thre.	Soft Thre.	Sure Thre.	Bayes Thre.	R.haar	Tetrolet	Mod. Tetrolet
Lena($\sigma=10$)	18.0238	16.9394	18.007	16.3865	18.1044	17.1216	18.1173	17.4755
Lena($\sigma=20$)	15.845	15.1981	16.075	16.0349	15.4209	15.4097	15.6387	16.1782
Lena($\sigma=30$)	14.9988	14.5043	15.1763	15.2268	14.4283	14.5044	14.4745	15.0728
Cameraman($\sigma=10$)	20.8732	20.0223	20.8008	18.9065	21.2363	20.7106	21.3163	20.0759
Cameraman($\sigma=20$)	17.3948	17.2376	18.0491	18.0112	17.7155	18.1666	18.2748	18.4846
Cameraman($\sigma=30$)	15.69	15.8885	16.4887	16.4155	15.9169	16.4718	16.7629	17.0321
Boat($\sigma=10$)	17.8872	17.1818	17.7985	16.2981	17.9282	17.2614	17.9117	17.4718
Boat($\sigma=20$)	15.4557	14.9675	15.868	15.7969	15.4464	15.1454	15.2025	15.7593
Boat($\sigma=30$)	14.4708	14.0655	14.8082	14.8455	14.37	14.1048	14.8496	14.7445
Barbara($\sigma=10$)	17.7469	16.5511	17.3948	15.6728	17.7351	16.1818	17.4057	16.2169
Barbara($\sigma=20$)	15.9533	14.2168	15.5736	15.5213	15.3933	14.391	14.5513	14.8607
Barbara($\sigma=30$)	15.3295	13.7197	15.0177	15.2778	14.2361	13.8749	13.4548	14.208

Table 4.4: PSNR (in dB) Performance Table

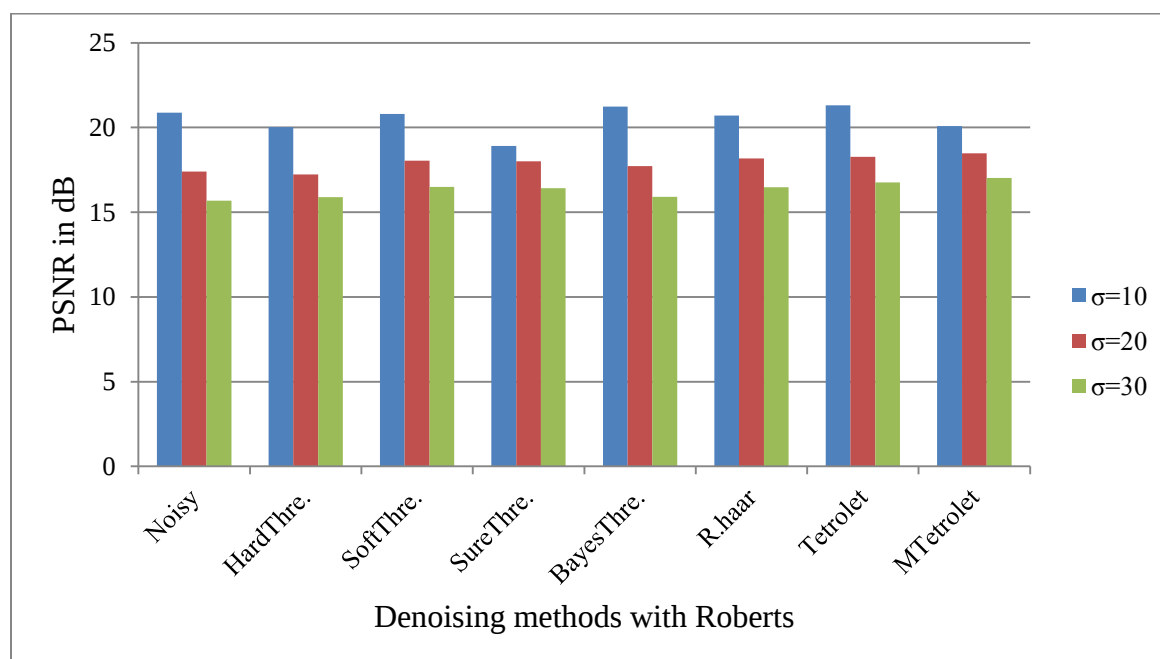


Figure 4.20: Performance comparison with Cameraman image

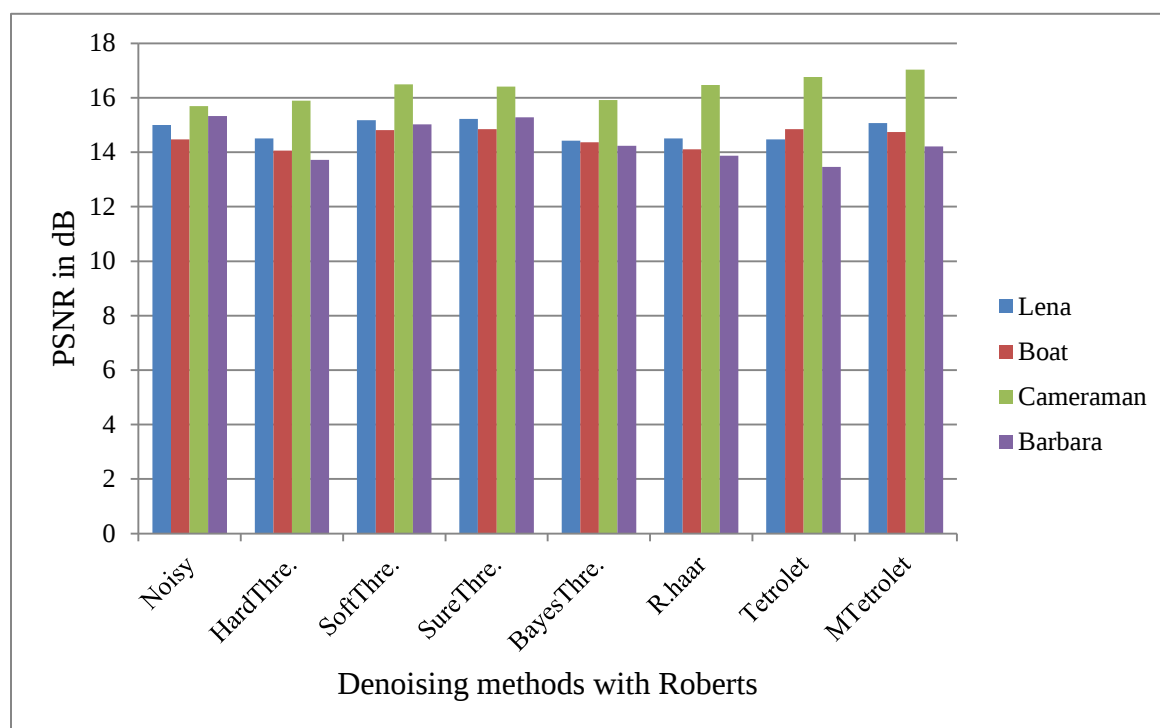


Figure 4.21: Performance comparison with Gaussian noise

The performance graphs in Figures 4.20 and 4.21 compares our method with other methods for different images corrupted with a Gaussian noise of variable standard deviations.

For Lena image, the Tetrom method before Roberts have a highest PSNR value for noise variance of 30, however, for variance other than 30, the modified Tetrom de-noising takes the lead. There is also a similar trend for the other images except Barbara image. For Barbara test image, the Bayes threshold and the previous Tetrolet before Roberts have a highest PSNR result for a noise variance of 30, for other noise scenarios all the de-noising including the method proposed in this thesis have a competitive PSNR values.

B. Speckle noise: The four test images (Lena, Barbara, Boat and Cameraman) were corrupted with Speckle noise, and de-noised using different methods and the de-noised image is subjected to the Roberts edge detector operator. The variance of the Speckle noise is varied from 0.02 to 0.1. The results are the PSNR values averaged over 6 runs. They are presented in Tables 4.5 and 4.6.

The following observations can be drawn from these results:

- The Roberts with modified Tetrolet method performs, on an average, up to 1 dB better when compared with Roberts with Lee Filter and Roberts with redundant Haar method for noise variances 0.08 and 0.1. It has also a similar performance with the previous Tetrolet based de-noising algorithm in terms of PSNR for these noise variance scenarios. For a noise variance less than 0.08, the redundant Haar takes the lead in terms of PSNR.

Variance	Noisy Image	Lee	R.Haar	Tetrom	Mtetr.
0.02	18.7746	16.7755	17.1997	16.2308	16.3349
0.04	17.3048	16.5421	16.6726	16.0349	15.9402
0.06	15.8603	16.129	16.1473	15.576	15.6193
0.08	15.0282	15.9169	15.2391	15.3956	15.3567
0.10	14.3016	15.6242	14.1632	14.7052	14.8142

Table 4.5: PSNR (in dB) Performance Table for Cameraman image

Image	Noisy Image	Lee Filter	R.Haar	Tetrolet.	Mod.Tetrolet.
Lena(V=0.06)	14.1165	14.9883	13.2484	13.8159	13.7275
Cameraman(V=0.06)	14.4987	14.2273	13.0167	13.1337	12.8602
Boat(V=0.06)	13.8544	13.7936	12,8415	13.4695	13.3232
Barbara(V=0.06)	15.0198	14.8953	15.17	15.2711	15.1872

Table 4.6: PSNR (in dB) Performance Table

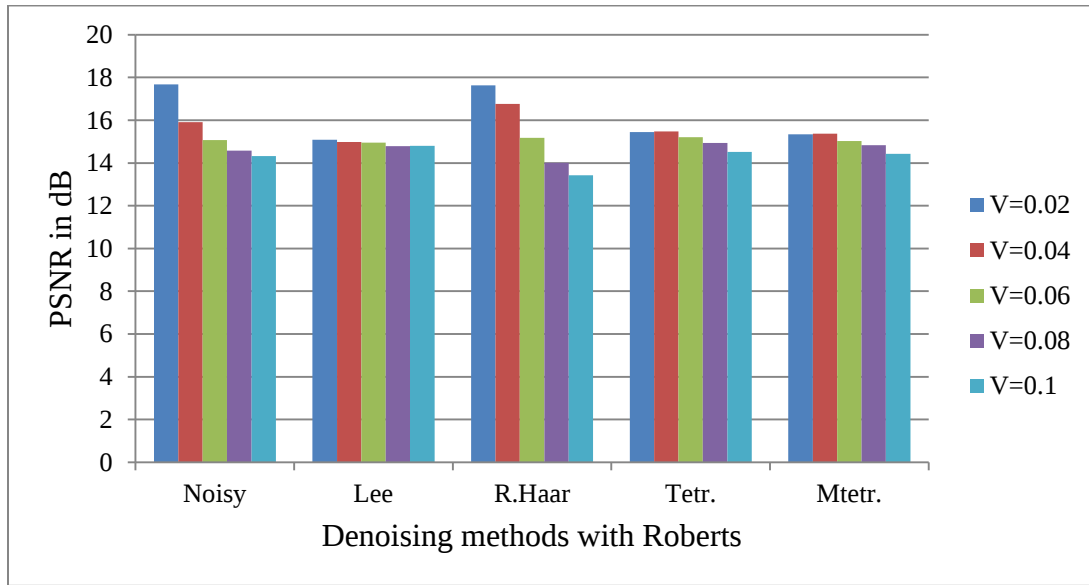


Figure 4.22: Performance comparison with Cameraman image

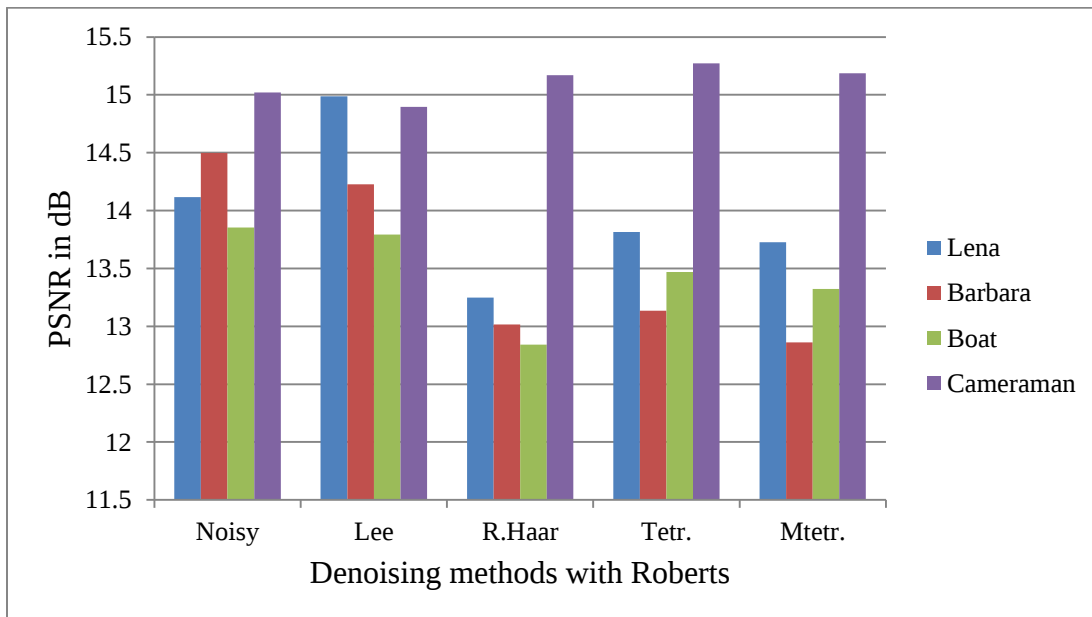


Figure 4.23: Performance comparison with Speckle noise for a variance of 0.6.

The performance graphs in Figures 4.22 and 4.23 compares the proposed method with other methods for Cameraman image corrupted with speckle noise with different variances. For lower values of noise variances, the redundant Haar performs a better PSNR than the other three, whereas for higher values of noise variances the Tetrolet before Roberts and the Lee filter before Roberts take over the lead.

For Cameraman image, all the de-noising algorithms show a high value in terms of PSNR, and Tetrolet de-noising before Roberts is the highest of all. When we come to the Barbara and Boat image the values of the PSNR scores for Redundant Haar, Tetrolet and modified Tetrolet are very low, however they are comparatively high for Lee filter. In all the three images other than Cameraman image, the Lee filter before Roberts shows better performance in terms of PSNR. Still the Tetrolet before Roberts and the modified Tetrolet before Roberts have a competitive result.

C. Poisson noise: The four test images (Lena, Barbara, Cameraman and Boat) were corrupted with Poisson noise, and de-noised using different methods and then the image is subjected to the Roberts edge detection operator. The results are the PSNR values averaged over 6 runs. They are presented in Table 4.7, which compares the proposed method with other algorithms such as Roberts with fast interscale wavelet de-noising, Sobel with Variance Stabilization with modified Tetrolet de-noising, Roberts with Tetrolet de-noising with nonlinear invariant Basis and Sobel with redundant Haar de-noising method.

The following observations can be drawn from these results:

- Tetrolet de-noising specially the one with nonlinear invariant basis brings a promising result. The Roberts with Tetrolet method (nonlinear invariant basis) performs, on an average, up to 0.25 dB better when compared with the Roberts with fast interscale wavelet de-noising and Roberts with redundant Haar method.

Image	Noisy Image	Fast wavelet	VST. Tetrolet	NL. Tetrolet	R. Haar
Lena	17.9695	17.8401	17.373	17.6843	17.9736
Barbara	17.4644	16.5765	16.9722	16.9328	17.4644
Boat	17.3513	17.1778	17.3621	17.539	17.3621
Cameraman	20.6987	20.5752	21.1751	21.4903	20.6985

Table 4.7: PSNR (in dB) Performance Table

The performance graphs in Figure 4.24 compare our method with other methods for different standard images.

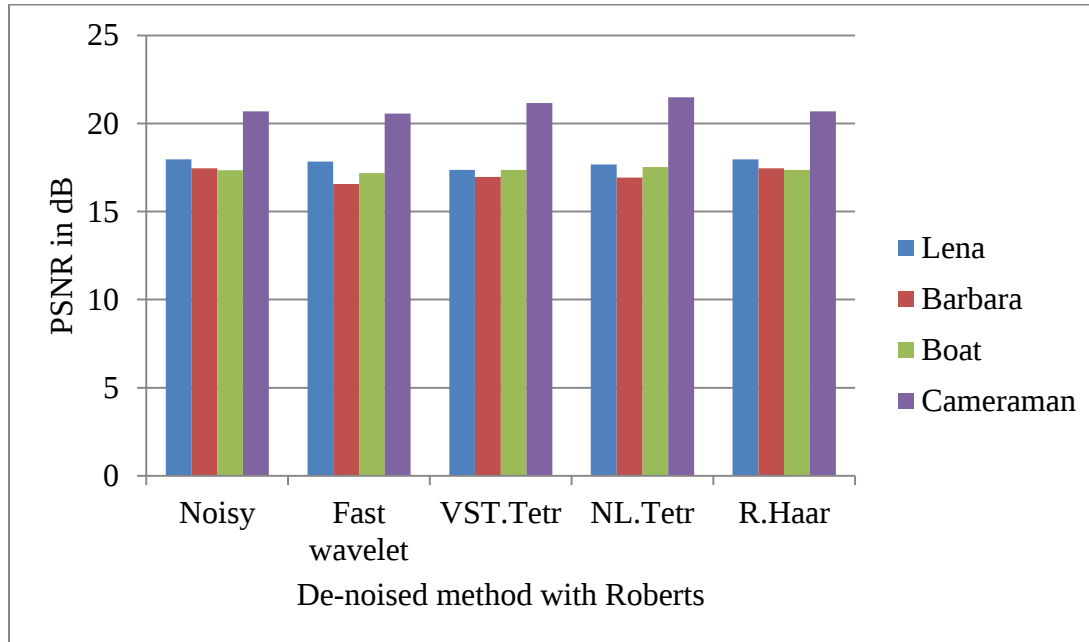


Figure 4.24: Performance comparison with Poisson noise

Out of the four test images, only Cameraman image shows a difference in performance in terms of PSNR among the candidate de-noising algorithms and Tetrolet transform with nonlinear invariant basis before Roberts shows the highest Peak values. For the other three images the performance is almost similar.

I. Performance with Sobel edge detection

A. White Gaussian noise: The four test images (Lena, Barbara, Boat and Cameraman) were subjected to an additive white Gaussian noise, and de-noised using different methods and then the image is fed to the edge detection operator. The variance of the white noise is varied from 10 to 30 in steps of 5. The results are the PSNR values averaged over 6 runs. They are presented in Table 4.8, which compares the proposed algorithm with other algorithms such as Sobel with VisuHard, Sobel with VisuSoft, Sobel with Sure, Sobel with Bayes and Sobel with redundant Haar method.

The following observations can be drawn from these results:

- The Sobel with modified Tetrolet method performs, on an average, up to 0.5 dB better PSNR when compared with the Sobel with VisuHard for noise variances 20 and 30.

A modified Tetrolet based image de-noising for real-time edge detectors

However Sobel with VisuSoft, Sobel with Sure, Sobel with bayesThre and Sobel with redundant Haar method scored a better PSNR than the Tetrolet based methods.

- The modified Tetrolet method with Sobel has a lesser PSNR compared with the modified Tetrolet with Roberts. Which is due to, Sobel operator have a 3x3 convolution kernels, whereas the Roberts operator have a 2x2 convolution masks/kernels, which makes the Roberts operator more suitable for the Tetrolet 4x4 window Tetromino partitions.

Images	Noisy Image	Hard Thre.	Soft Thre.	Sure Thre.	Bayes Thre.	R.Haar	Tetrolet	Mod. Tetrolet
Lena($\sigma=10$)	18.5364	17.3048	18.4335	17.634	18.3876	17.48	18.1173	16.5153
Lena($\sigma=20$)	16.0243	14.9426	16.0005	15.9794	15.7072	15.2169	15.6387	15.2957
Lena($\sigma=30$)	14.7842	14.1681	14.8485	14.8042	14.4634	13.5797	14.4745	14.421
Cameraman($\sigma=10$)	21.1405	20.0424	21.4438	20.629	21.2895	20.8847	21.3163	19.0214
Cameraman($\sigma=20$)	18.5553	17.0287	18.7348	18.6709	18.4706	17.9014	18.2748	17.7588
Cameraman($\sigma=30$)	16.9067	15.5095	17.2061	17.1542	16.5782	16.5578	16.7629	16.7128
Boat($\sigma=10$)	18.0916	16.4887	18.0959	17.2587	18.2971	17.3016	17.9117	15.4884
Boat($\sigma=20$)	15.519	13.9917	15.663	15.6217	15.3205	14.5213	15.2025	14.4173
Boat($\sigma=30$)	14.1456	13.1501	14.3591	14.3338	13.7889	12.9274	13.8496	13.5214
Barbara($\sigma=10$)	17.7545	16.1618	17.4607	16.688	17.6072	16.6379	17.4057	15.5048
Barbara($\sigma=20$)	15.0473	13.6301	15.1567	15.1264	15.0451	13.8688	14.5513	14.2502
Barbara($\sigma=30$)	13.8287	12.9338	13.8669	13.8981	13.6057	12.3864	13.4548	13.5664

Table 4.8: PSNR (in dB) Performance Table (with White Gaussian)

The performance graphs in Figures 4.25 and 4.26 compares our method with other methods with Sobel edge detection operator. For lower values of noise variance ($\sigma=10$), the modified Tetrolet based de-noising with Sobel doesn't give a good PSNR. However, when the noise variance increases, the performance of the Tetrolet method increases. For Cameraman image, Soft Threshold, Bayes Threshold and Tetrolet de-noising shows good PSNR value.

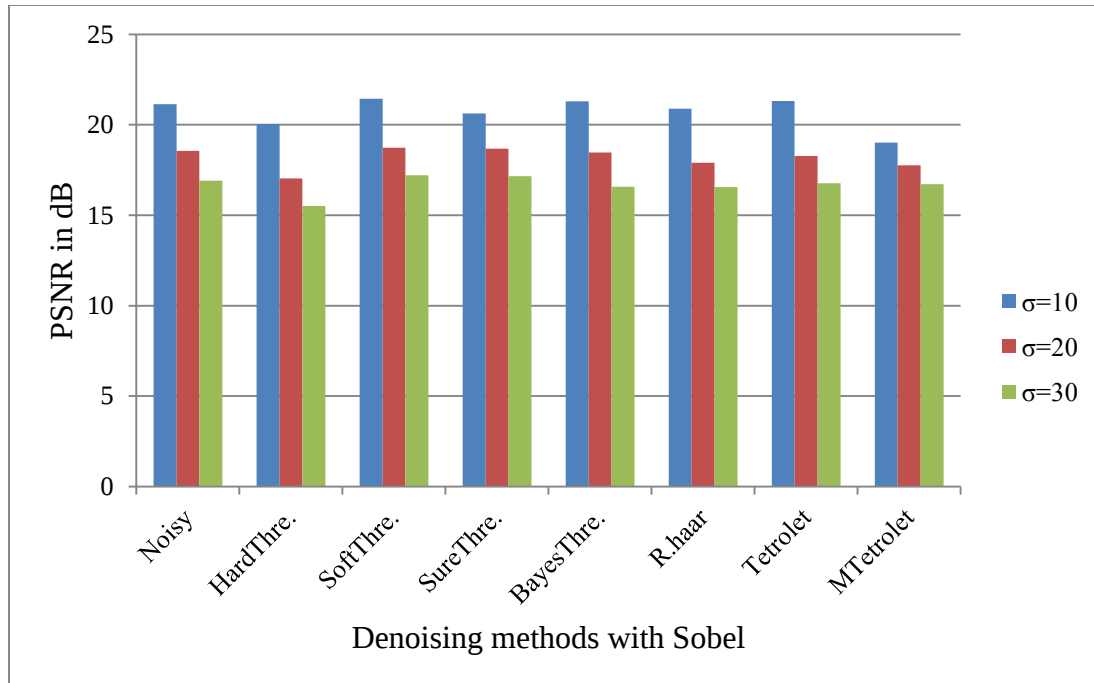


Figure 4.25: Performance comparison with Cameraman image

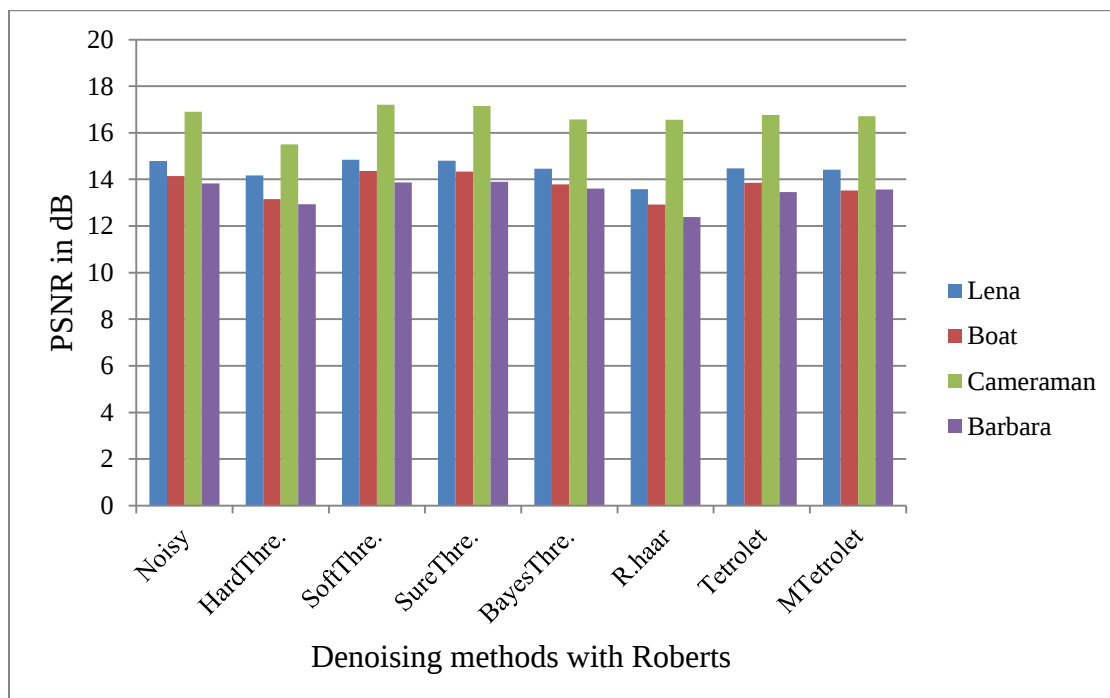


Figure 4.26: Performance comparison with Gaussian Image

B. Speckle noise: The four standard test images (Lena, Barbara, Boat and Cameraman) were corrupted with Speckle noise, and de-noised using different methods and then the de-noised image is subjected to the edge detector operator. The variance of the Speckle noise of the Cameraman image is varied from 0.02 to 0.1 and its performance is compared. The results are the PSNR values averaged over 6 runs. They are presented in Tables 4.9 and 4.10, which compares the modified Tetrolet algorithm with Sobel and other algorithms such as Sobel with Lee Filter and Sobel with redundant Haar method. The following observations can be drawn from these results:

- The Sobel with Tetrolet method performs, on an average, up to 0.7 dB less when compared with Sobel with Lee Filter and Sobel with redundant Haar method for Barbara image. It has nearly similar performance with the previous Tetrolet based de-noising algorithm in terms of PSNR.

Variance(V)	Noisy Image	Lee	R.Haar	Tetrolet	MTetrolet
0.02	18.7746	16.7755	17.1997	16.2308	16.3349
0.04	17.3048	16.5421	16.6726	16.0349	15.9402
0.06	15.8603	16.129	16.1473	15.576	15.6193
0.08	15.0282	15.9169	15.2391	15.3956	15.3567
0.10	14.3016	15.6242	14.1632	14.7052	14.8142

Table 4.9: PSNR (in dB) Performance Table for Cameraman Image (with speckle)

Image	Noisy Image	Lee Filter	R.Haar	Tetrolet	MTetrolet
Lena (V=0.06)	13.5724	14.6199	13.1871	13.856	13.856
Cameraman(V=0.06)	12.8169	13.7212	12.1542	12.926	12.9495
Boat(V=0.06)	13.145	13.7557	12.5303	13.2765	13.2414
Barbara(V=0.06)	15.8603	16.129	16.1473	15.576	15.6193

V = Noise variance

Table 4.10: PSNR (in dB) Performance Table

The performance graphs in Figures 4.27 and 4.28 compares our method with other methods for images corrupted with Speckle noise with different values of noise variance.

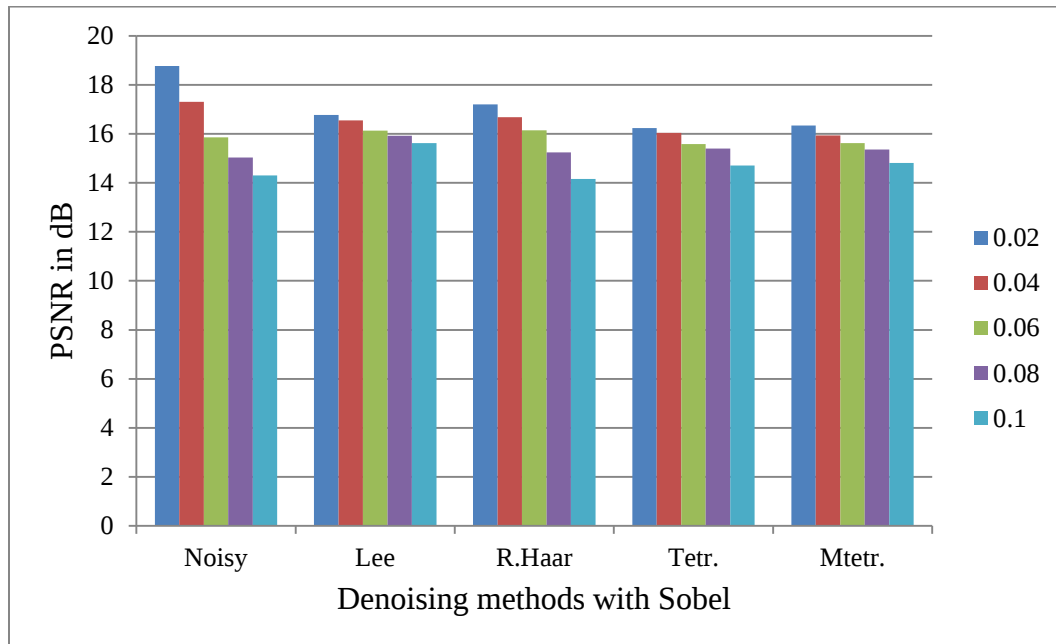


Figure 4.27: Performance comparison with Cameraman image

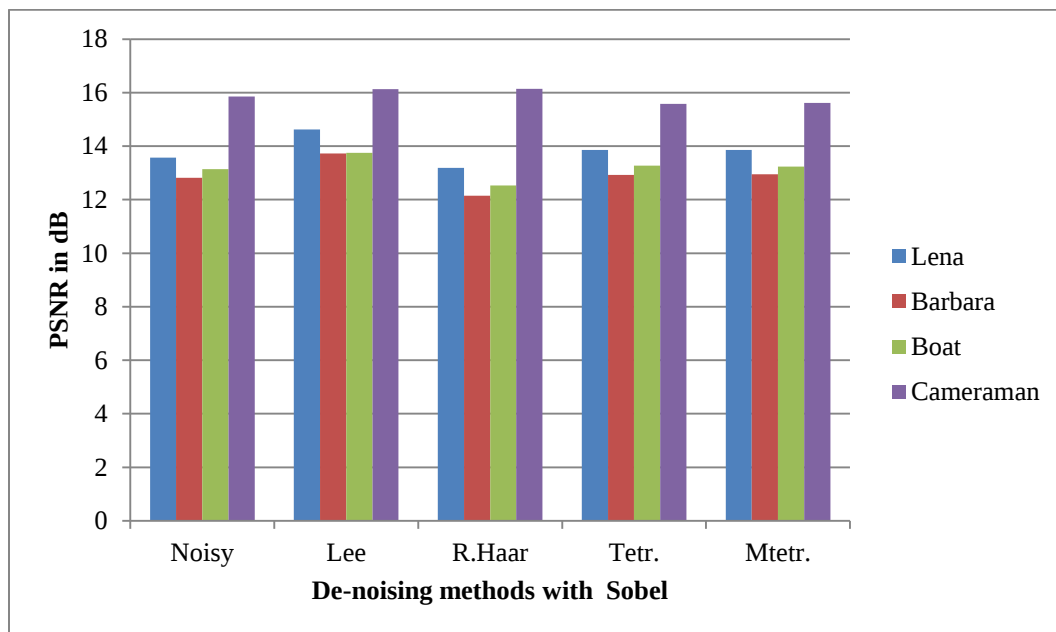


Figure 4.28: Performance comparison with Speckle noise

For lower values of speckle noise variances, the redundant Haar method has a better performance than the other three de-noising methods. For higher values of noise variances, our method is comparable to the other de-noising scheme. The Lee filter before Sobel scores the highest value

of PSNR for higher values of noise variances. Overall, the Cameraman image has the higher values of PSNR with respect to the other Sobel after speckle de-noising algorithms.

C. Poisson noise: The four test images (Lena, Barbara, Boat and Cameraman) were corrupted with Poisson noise and de-noised using different algorithms including our method and then it is fed in to the sobel edge detection operator. The results are the PSNR values averaged over 6 runs. They are presented in Table 4.11, which compares the proposed method with other algorithms such as Sobel with fast interscale wavelet de-noising, Sobel with variance stabilization with modified Tetrolet de-noising, Sobel with Tetrolet de-noising with nonlinear invariant bases and Sobel with redundant Haar de-noising method.

The following observations can be drawn from these results:

- Sobel with Tetrolet de-noising specially the one with nonlinear invariant basis brings a promising result.
- The Sobel with Tetrolet method (with NLI) performs, on an average, up to 0.6 dB better when compared with the Sobel with fast interscale wavelet de-noising. It has also a similar performance with the previous Tetrolet based de-noising algorithm in terms of PSNR.
- Except for Cameraman image, Sobel with redundant Haar method scored the highest PSNR values.

Image	Noisy	Fast wavelet	VST. Tetrolet	NLI. Tetrolet	R. Haar
Lena	18.5081	17.4335	17.8629	18.1997	18.5411
Barbara	17.1439	16.1063	16.6973	16.6973	17.1199
Boat	17.9695	17.2304	17.5127	17.7233	17.982
Cameraman	21.1491	20.4809	21.0636	21.1491	20.6985

Table 4.11: PSNR (in dB) Performance Table (Poisson)

The performance graphs in Figure 4.29 compares our method with other methods for four standard images such as Lena, Barbara, Boat and Cameraman corrupted with Poisson noise.

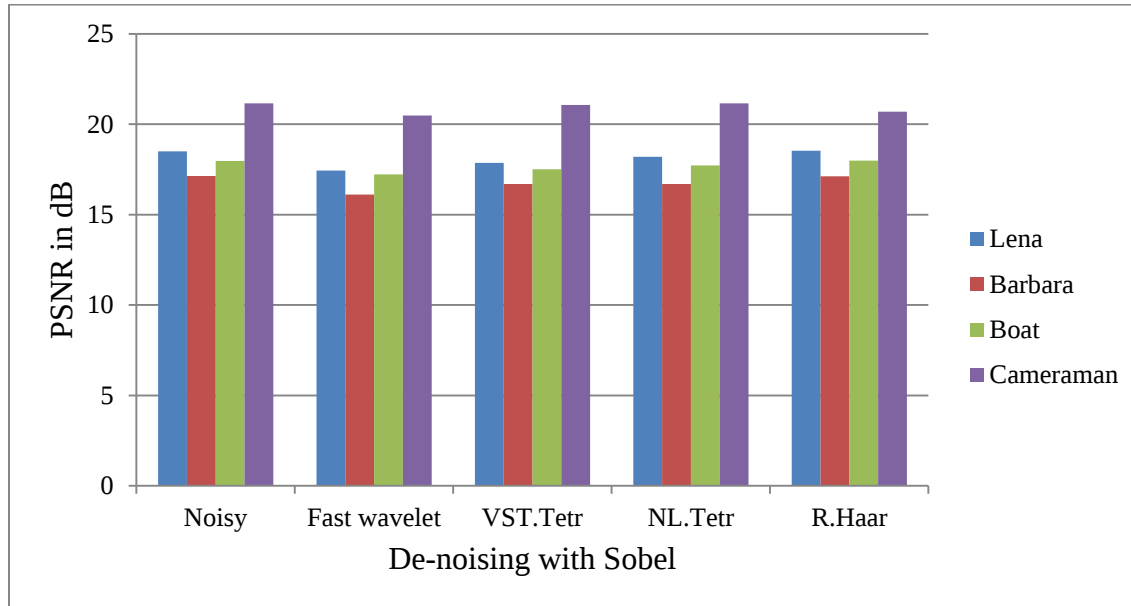


Figure 4.29: Performance comparison with Poisson noise

The trend of the performance with different de-noising methods is similar for all the test images. Out of the four methods, the Tetrolet with nonlinear invariant basis score the highest PSNR values than the other three candidate methods for Cameraman image. Still the three methods have a competitive result.

4.4. HARDWARE SYNTHESIS AND SIMULATION RESULTS

The functional description of VHDL model of each stage of the system is synthesized and simulated using Xilinx Synthesis and ModelSim SE Simulation Tool Environment. The design involves the forward discrete Tetrolet transform (FDTT) and its inverse discrete Tetrolet transform (IDTT) for single number of transformation levels. Each of the first two modules is designed as a hierarchical scheme that uses one-dimensional processing module twice to represent two-dimensional processing. The module can be used repeatedly on the same image for multilevel processing. However, only the first three Tetrolet partitions are used.

The first two stages of the design were synthesized and simulated using a white Gaussian noise corrupted 128x128 Cameraman image. Whereas the Edge detection stage is synthesized using Cameraman image with a size of 64x64, which is due to avoid the compilation error generated by the ISE, which only tolerates a maximum of 64 iterations for a loop, thus the maximum value

of the image size had to be constrained to a certain value range. This issue should be solved if more iteration is required, but it has not been done for this thesis. The synthesis result of stage one, which is the FDTT module, is presented in Appendix A.

After the design is compiled using ISE, XST synthesis tool report is used to determine preliminary device utilization and performance. After the design is mapped by ISE® Design Suite, the actual device utilization is determined.

4.4.1. DEVICE UTILIZATION

These designs were synthesized on Xilinx Virtex®-4 FPGA (XC4VSX Series). The device usage is normally reported in terms of number of look-up tables, I/O buffers, and flip flops. The device usage for FDTT for different Tetrolet partitions is displayed in Table 4.12.

Device Utilization Summary								
Logic Utilization	Available	2D-HAAR		2TETROLET.		3TETROLET		15TETR.
		Used	Utilization	Used	Utilization	Used	Utilization	Utilization
Number of Slices	10240	383	3%	522	5%	797	7%	≈ 31%
Number of Slice Flip Flops	20480	279	1%	365	1%	542	2%	≈ 15%
Number of 4 input LUTs	20480	716	3%	976	4%	1454	7%	≈ 43%
Number of bonded IOBs	320	36	11%	36	11%	36	11%	≈ 11%
Number of GCLKs	32	1	3%	1	3%	1	3%	≈ 3%

Table 4.12: Device utilization FDTT for different Tetrolet Partitions

From the Table 4.12, the results are generated from the Xilinx XST synthesis tool, which is similar to the one generated from the Mapping tool. As the number of Tetrolet partitions increases by 1, the number of Slices doubles, the number of Slice Flip Flops increments and the number of 4 input LUTs increases by 3%. Whereas the number of bounded IOBs and the number of Global Clock buffers does not change with Tetrolet partitions.

The FDTT design is more of a combinational logic then sequential logic, because the number of LUTs utilization is greater than FFs utilization for every Tetrolet partition. The device utilization of IDTT is similar with the FDTT.

4.4.2. RTL SCHEMATIC RESULTS

Figure 4.30 shows the RTL schematic view of forward discrete Tetrolet transform image decomposition algorithm.

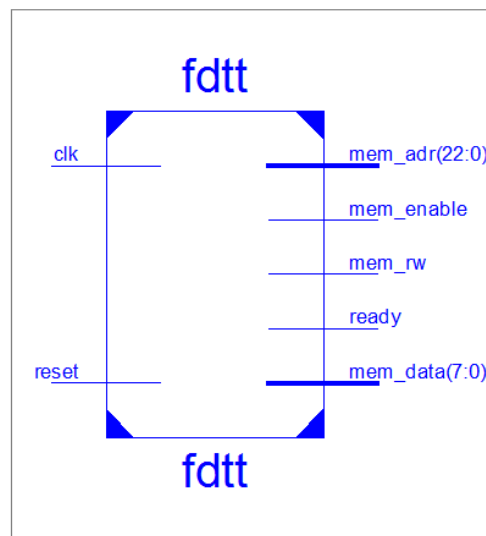


Figure 4.30: RTL schematic view of FDTT

4.4.3. SIMULATION WAVEFORMS

Mentor Graphics ModelSim SE 10.0b software is used to verify the functionality of the designed system with and without delays. The behavioral (functional) simulation refers to functional aspects of the system, without considering the signals delays, which will be known only after implementation. Unlike the functional simulation, the timing simulation however has much help

to investigate timing properties of the design beside the PAR results. The Simulation was implemented for “Cameraman” image with a size of 128×128. Due to the similarity of the VHDL implementation of forward Tetrolet transform and inverse Tetrolet transform, the Simulation waveform of stage two is not presented here.

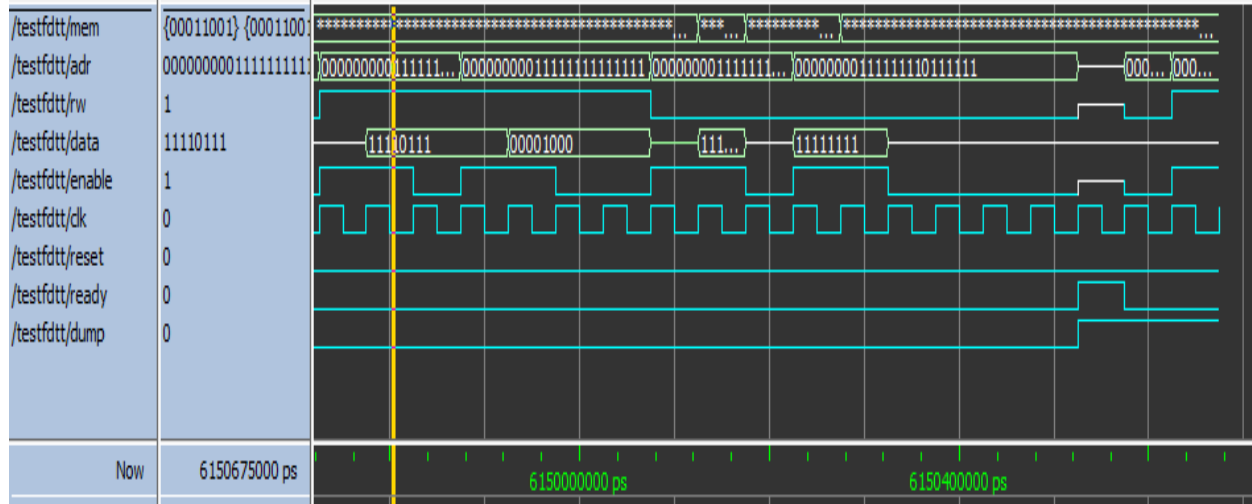


Figure 4.31: ModelSim Timing simulation of the FDTT

4.4.5. REAL-TIME PERFORMANCE

Xilinx XST Synthesis result showed that the FDTT has the maximum frequency (155.027MHz) (See Appendix A). Also comparing the Maximum output required time after clock and Minimum input arrival time before clock, a positive slack is observed, since the difference of the two gives a positive result.

For simulation, 128x128 Cameraman image corrupted with Gaussian noise is used. The ModelSim SE simulation waveform result is used to measure the execution time and the number of clock cycles. To calculate the no. of clock cycles, divide the execution time with the duration of the clock period.

$$\text{Clock Cycles} = \frac{\text{Excution time}}{\text{Time duration of Clock period}} \quad (4.3)$$

Stage of design	No. Tetrolet partitions	Execution time(ns)	No. of Clock cycles	Maximum Frequency(MHz)	Time duration of Clock period (ns)
Stage 1 FDTT (128x128)	1D-Haar	6102625	122053	203.934	50
	One (2D-Haar)	9157075	183142	189.418	50
	Two	12211525	244231	195.409	50
	Three	21374875	427984	155.027	50
	Fifteen	≈ 91633500	≈ 1832670	≈	50

Table 4.14: Review of No. of clock cycle for FDTT and 2D Haar image decomposition

As the number of Tetrolet partitions increases, the number of clock cycles increases, whereas the maximum frequency oscillates between 155.027MHz - 195.409MHz.

Stage of design	Size of image	Execution time(ns)	No. of Clock cycles	Maximum Frequency(MHz)	Time duration of Clock period (ns)`
Stage 3 (Sobel algorithm)	16x16	26100	261	39.277	100
	32x32	102,900	1029	24.446	100
	64x64	410100	4101	12.867	100
	128x128	1638900	16389	-	100
	256x256	6554100	65541	-	100

Table 4.15: Review of No. of clock cycle for Sobel edge detection algorithm.

Regarding the Sobel edge detection operator, the maximum frequency of operation is not as good as the de-noising algorithm. It runs at a maximum frequency of 12.867MHz, 24.446MHz and 39.277MHz for 16x16, 32x32 and 64x64 image sizes respectively. Table 4.15 also shows that, as the number of image sizes doubles, the number of clock cycles increases by four times.

4.5. SUMMARY

In this thesis, different well known image de-noising algorithms are investigated and their performance was comparably assessed using simulation results.

The main aim of this thesis is reducing the computational complexity of the Tetrolet based de-noising and use it as a pre-processing scheme for first order gradient based edge detection schemes.

After the analysis of the test results, the Tetrolet based de-noising proved to be a better algorithm for white Gaussian noise removal, than its wavelet based predecessors. In general, the Tetrolet based image de-noising brings a good improvement in terms PSNR, however for the software realization, the execution time for previous Tetrolet transform based de-noising was unpractical. In this paper, the execution time of the Tetrolet based image de-noising quite improved for the software realization.

In the last part of the result, the result of hardware design for discrete Tetrolet transform (DTT) module in VHDL language is presented. The modules perform forward Tetrolet transform to images of size 128X128 pixels. Synthesis process showed that the DFTT can achieve maximum frequency of (about 155.027MHz) and 1832670 No. of Clock cycles.

For the hardware realization of edge detection algorithm, minimum period for two flip flops that are internal to the device can be clocked is relatively larger. In order to improve this time, optimization of the Sobel edge detection algorithm is recommended.

The Matlab implementation of the modified Tetrolet based de-noising brings a computational time of 5 seconds, whereas the hardware realization brings about execution time of 92 ms (Approximate value).

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1. INTRODUCTION

In this work, Noise removal of image for edge detection algorithm is designed. The designed system is implemented using MATLAB and VHDL. The system is finally compared with benchmark methods. Furthermore, the MATLAB implementation is compared with the FPGA based implementation. Based on the results, the following conclusions and recommendations are made.

5.2. CONCLUSIONS

Generally, the quality of the new Tetrolet based image de-noising is greater than the quality of other benchmark wavelet based de-noising algorithms for white Gaussian noise corrupted images. However, the quality of the new image de-noising algorithm is comparably as equal quality as the previous Tetrolet based image de-noising algorithm.

For the software implementation, the newly proposed algorithm brings down the execution time from the previous Tetrolet based method (80 minutes) to the modified Tetrolet method (5 seconds) for 256x256 image sizes.

FPGA based implementation of Tetrolet based image de-noising results in significant improvement in terms of speed over the software implementation. The synthesis and simulation results showed that, on FPGA the Tetrolet based decomposition or Reconstruction algorithms can achieve a maximum frequency of 155.027MHz and 1832670 the number of clock cycles.

The Software implementation of the modified Tetrolet based de-noising have a computational time of 5 seconds, whereas the hardware realization brings about an execution time of 92ms (Approximation).

5.3. RECOMMENDATIONS

The Matlab simulation results also revealed that, with some modification, the Tetrolet based de-noising have the ability to remove not only the white Gaussian noise but also the speckle and Poisson noise.

For the software realization on Matlab, the modified Tetrolet based image de-noising improves the execution time of previous Tetrom de-noising [3]. However if the vectoring of loops further exploited, the execution time can be improved more.

Regarding the FPGA implementation, only the first three Tetrolet partitions are considered from the 15 selected partitions. Thus for the full realization of the Tetrolet transform, it is recommended to realize and check the performance of the full Tetrolet transform.

The Hardware realization of the forward discrete Tetrolet transform based image decomposition algorithm designed here can be used for the Tetrolet based image compression algorithm with an integration of the remaining compression steps.

The Sobel edge detection is synthesized for only 64x64 image sizes due to the fact that Xilinx ISE only support loops with 64 maximum iterations. This issue is not solved for this thesis. It is left as a future work.

REFERENCES

- [1] J. Krommweh, "Tetrolet transform: A new adaptive Haar wavelet algorithm for sparse image representation ", Journal of Visual Communication and Image Representation, Vol. 21, no. 4, pp. 364–374, 2010.
- [2] Cai-lian Li, Ji-xiang Sun and Yao-hong Kang, "A new algorithm for image denoising based on tetrolet transform", International Conference on Image Processing and Pattern Recognition in Industrial Engineering, Volume 7820, pp. 78201L-78201L-7 (2010).
- [3] Singh, Manish Kumar, "denoising of natural images using the wavelet transform" (2010). Master's Theses, Paper 3895.
- [4] Rafael C. Gonzalez and Richard E. Woods. Digital Image Processing. India: Dorling Kindersley Pvt. Ltd., licensee of Pearson Education in South Asia, 2008, pp. 156, 311, 316-317.
- [5] Shehrzad Qureshi, "EMBEDDED IMAGE PROCESSING ON THE TMS320C6000™ DSP", © 2005 Springer Science+Business Media, Inc.
- [6] Donoho, D.L.; De-noising by soft-thresholding, Information Theory, IEEE Transactions on August 2002. Volume: 41 Issue: 3.
- [7] David L. Donoho and Iain M. Johnstone. "Adapting to Unknown Smoothness via Wavelet Shrinkage." Journal of the American Statistical Association, Vol. 90, No. 432, pp. 1200-1224, Dec. 1995.
- [8] S. Grace Chang, Bin YU, and Martin Vetterli. "Adaptive Wavelet Thresholding for Image Denoising and Compression." IEEE Transactions on Image Processing, Vol. 9, No. 9, pp. 1532-1546, Sep. 2000.

- [10] An improved Sobel edge detection, Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on 9-11 July 2010
- [11] An improved Sobel algorithm based on median filter, Mechanical and Electronics Engineering (ICMEE), 2010 2nd International Conference on August 2010.
- [12] Rodrigues, I. Sanches, J. ; Bioucas-Dias, J. Denoising of medical images corrupted by Poisson noise, Image Processing, 2008. ICIP 2008. 15th IEEE International Conference, 12-15 Oct. 2008
- [13] Xilinx Corporation. 1998. "Xilinx breaks one million-gate barrier with delivery of new Virtex series".
- [14] Xilinx Corporation. 2000. "Virtex Data Sheet".
- [15] Faizal Iqbal, "Wavelet Transform Based Image Compression on FPGA", M.Sc. thesis, College of Engineering, Florida State University, 2004.
- [16] E. D. Demaine, S. Hohenberger, D. Liben-Nowell. Tetris is hard, even to approximate. Tech report, LCS, MIT, 2002.
- [17] M. Korn, Geometric and algebraic properties of polyomino tilings, PhD thesis, Massachusetts Institute of Technology (2004).
- [18] Florian Luisier , Ce'dric Vonesch Thierry Blu, Michael Unser, Fast interscale wavelet denoising of Poisson-corrupted images, Journal of Signal Processing Elsevier North-Holland, Inc. Amsterdam, Volume 90, Issue 2, February, 2010, Pages 415-427
- [19] CNRS, CEREMADE, University Paris-Dauphine.
Internet: <http://www.ceremade.dauphine.fr/~peyre/>
- [20] VHDL Design Representation and Synthesis (2nd Edition), James R. Armstrong, F. Gail Gray . Prentice Hall, April 9, 2000.

APPENDICES

APPENDIX A: FDTT XST SYNTHESIS REPORT (SELECTED)

The synthesis report of the FDTT image decomposition algorithm is presented.

=====

Device utilization summary:

Selected Device : 4vsx25ff668-10

Number of Slices:	883 out of 10240	8%
Number of Slice Flip Flops:	641 out of 20480	3%
Number of 4 input LUTs:	1638 out of 20480	7%
Number of IOs:	36	
Number of bonded IOBs:	36 out of 320	11%
Number of GCLKs:	1 out of 32	3%

TIMING REPORT

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.

FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
GENERATED AFTER PLACE-and-ROUTE.

Clock Information:

-----+-----+-----+

Clock Signal	Clock buffer(FF name)	Load	
clk	BUFGP	641	

-----+-----+-----+

Asynchronous Control Signals Information:

-----+-----+-----+

Control Signal	Buffer(FF name)	Load	
reset	IBUF	81	

-----+-----+-----+

Timing Summary:

Speed Grade: -10

A modified Tetrolet based image de-noising for real-time edge detectors

Minimum period: 6.451ns (Maximum Frequency: 155.027MHz)

Minimum input arrival time before clock: 5.298ns

Maximum output required time after clock: 7.794ns

Maximum combinational path delay: No path found

Timing Detail:

All values displayed in nanoseconds (ns)

=====

Timing constraint: Default period analysis for Clock 'clk'

Clock period: 6.451ns (frequency: 155.027MHz)

Total number of paths / destination ports: 47516 / 750

Delay: 6.451ns (Levels of Logic = 7)

Source: state_FSM_FFd25 (FF)

Destination: source2a_0 (FF)

Source Clock: clk rising

Destination Clock: clk rising

Data Path: state_FSM_FFd25 to source2a_0

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

FDCE:C->Q	28	0.360	1.137	state_FSM_FFd25 (state_FSM_FFd25)
LUT4:I2->O	2	0.195	0.602	access_bus_mux000010 (access_bus_mux000010)
LUT4:I2->O	1	0.195	0.523	counter_or000124 (counter_or000124)
LUT2:I1->O	6	0.195	0.560	counter_or000133 (counter_or0001)
LUT4_D:I3->LO	1	0.195	0.163	destination2_mux0000<16>11122 (N522)
LUT3:I2->O	8	0.195	0.608	destination2_mux0000<16>11115 (N34)
LUT4_D:I3->O	37	0.195	1.110	N41 (N4)
LUT4:I3->O	1	0.195	0.000	source2a_mux0000<8>1 (source2a_mux0000<8>)
FDE:D		0.022		source2a_8

Total 6.451ns (1.747ns logic, 4.704ns route)

(27.1% logic, 72.9% route)

=====

Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'

A modified Tetrolet based image de-noising for real-time edge detectors

Total number of paths / destination ports: 656 / 656

Offset: 5.298ns (Levels of Logic = 2)

Source: reset (PAD)

Destination: source_1 (FF)

Destination Clock: clk rising

Data Path: reset to source_1

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

IBUF:I->O 100 0.965 1.194 reset_IBUF (reset_IBUF)

INV:I->O 368 0.358 2.242 reset_inv1_INV_0 (reset_inv)

FDE:CE 0.540 source_1

Total 5.298ns (1.863ns logic, 3.435ns route)
(35.2% logic, 64.8% route)

Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'

Total number of paths / destination ports: 342 / 34

Offset: 7.794ns (Levels of Logic = 4)

Source: state_FSM_FFd50 (FF)

Destination: mem_enable (PAD)

Source Clock: clk rising

Data Path: state_FSM_FFd50 to mem_enable

Gate Net

Cell:in->out fanout Delay Delay Logical Name (Net Name)

FDC:C->Q 12 0.360 0.848 state_FSM_FFd50 (state_FSM_FFd50)

LUT3:I0->O 1 0.195 0.523 mem_adr<0>_MLTSRCEDGELogicTrst1_SW0 (N02)

LUT4:I3->O 50 0.195 1.161 mem_adr<0>_MLTSRCEDGELogicTrst1 (N46)

LUT2:I1->O 1 0.195 0.360 mem_rw_MLTSRCEDGELogicTrst1 (mem_rw_MLTSRCEDGE)

OBUFT:I->O 3.957 mem_rw_OBUFT (mem_rw)

Total 7.794ns (4.902ns logic, 2.892ns route)
(62.9% logic, 37.1% route)

ABSTRACT

A modified Tetrolet based image de-noising for real time edge detectors

Eyob Teshome

Addis Ababa University, 2012

Image de-noising involves the manipulation of the image data to produce a visually high quality image. Computer vision involves the identification and classification of objects in an image, therefore edge detection is an essential tool. Aiming at the problems of traditional edge detection algorithms such as Sobel, Prewitt, Canny and LoG etc. on noise immunity, this paper presents a combination of an adaptive image de-noising algorithm based on the Tetrolet transform and first order gradient based edge detectors.

An improved image de-noising algorithm based on the Tetrolet wavelet transform is proposed. The Tetrolet transform is an adaptive Haar wavelet transform whose support is Tetrominoes, that is, shapes made by connecting four equal sized squares. The algorithm is proposed to improve de-noising performance for images corrupted by additive white Gaussian noise (AWGN), Poisson noise, and Speckle noise. Which in turn used to improve the first order gradient based edge detection algorithms.

The system is simulated in Matlab, and performance is tested on Standard images. The proposed algorithm improves de-noising performance measured in peak signal-to-noise ratio (PSNR) by 1-2.0 dB over the Haar wavelet transform based de-noising algorithms for images corrupted by additive white Gaussian noise (AWGN) assuming universal hard thresholding. The computational time for the software implementation is also significantly improved compared to the previous Tetrolet based image de-noising algorithm. The local nature of the algorithm makes the proposed method well suited for efficient hardware implementation. Preliminary results showed that FPGA design of the discrete forward Tetrolet transform is running at 155.027MHz targeting Virtex-4 FPGA.

ACKNOWLEDGEMENT

First, I would like to forward my deepest appreciation and thanks to my advisor Ato Bisrat Derebssa(MSc.) for his valuable advice and support.

Then, I would like to thank my father Teshome Biratu, my mother Bezaget Fesesse, my sister Betelhem Teshome and the rest of my family for their support, encouragement and ideas. Without you, it would have been difficult for me.

At last, but not least, I would like to thank my colleague Gosa Demissie, Kassaye Tafesse and Zelalem Tamirat for their encouragement, comments and supports for accomplishing my thesis.

TABLE OF CONTENT

ABSTRACT.....	i
ACKNOWLEDGEMENT.....	ii
TABLE OF CONTENT.....	iii
LIST OF FIGURES.....	v
LIST OF TABLES.....	vii
ABBREVIATIONS.....	viii
CHAPTER 1.....	1
INTRODUCTION.....	1
1.1. BACKGROUND.....	1
1.2. STATEMENT OF THE PROBLEM.....	3
1.3. OBJECTIVE.	4
1.4. METHODOLOGY.....	4
1.5. THESIS SCOPE	5
1.6. CONTRIBUTION AND LIMITATIONS	5
1.7. ORGANIZATION OF THE THESIS	5
CHAPTER 2	6
REVIEWS ON DE-NOISING AND EDGE DETECTION SYSTEM.....	6
2.1. INTRODUCTION	6
2.2. LITERATURES OF DENOISING METHODS.....	7
2.3. LITERATURES OF EDGE DETECTION.....	11
2.4. LITERATURES ON TETROLET TRANSFORM USING FPGA.....	11
2.5. SUMMARY.....	12
CHAPTER 3.....	13
SYSTEM DESIGN AND DEVELOPMENT	13
3.1. INTRODUCTION	13
3.2. SYSTEM COMPONENTS	13
3.3. HARDWARE DESIGN AND SIMULATION.....	25
3.4. SUMMARY.....	29
CHAPTER 4.....	30
RESULT AND DISCUSSIONS.....	30

4.1. INTRODUCTION	30
4.2. IMAGE DENOISING RESULT	30
4.3. EDGE DETECTION AFTER IMAGE DE-NOISING RESULT.....	51
4.4. HARDWARE SYNTHESIS AND SIMULATION RESULTS.....	74
4.5. SUMMARY.....	79
CHAPTER 5.....	80
CONCLUSION AND RECOMMENDATIONS	80
5.1. INTRODUCTION	80
5.2. CONCLUSIONS	80
5.3. RECOMMENDATIONS	81
REFERENCES.....	82
APPENDICES.....	84
APPENDIX A: FDTT XST SYNTHESIS REPORT (SELECTED).....	84

LIST OF FIGURES

Figure 3.1: Main components of the system.....	13
Figure 3.2: Basic steps in Tetrolet based image de-noising step.....	15
Figure 3.3: Sketch map of Tetrolet Decomposition Frame for 64x64 input image [2].....	16
Figure 3.4: Haar Scaling ($\phi(t)$) and mother wavelet ($\psi(t)$) functions.....	17
Figure 3.5: 2D Haar Image Decomposition.....	18
Figure 3.6: The 5 free Tetrominoes: O - I - T - S - L.....	19
Figure 3.7: The 22 fundamental forms tiling a 4×4 board.....	19
Figure 3.8: The 15 selected Tetromino configurations.....	20
Figure 3.9: The modified Tetrolet based image de-noising algorithm using flowchart.....	22
Figure 3.10: A simplified diagram of a Xilinx Virtex-4 FPGA slice.....	26
Figure 3.11: FPGA Design Flow.....	26
Figure 3.12: Layout of the proposed architecture for the fDTT.....	28
Figure 4.1: Lena Image de-noised Image	32
Figure 4.2: Barbara de-noised Image.....	34
Figure 4.3: Boat de-noised Image.....	36
Figure 4.4: Cameraman Image de-noised	38
Figure 4.5: De-noised Boat image.....	40
Figure 4.6: De-noised Cameraman image (Speckle noise).....	41
Figure 4.7: De-noised Boat image (Poisson noise).....	43
Figure 4.8: De-noised Cameraman image (Poisson noise).....	45
Figure 4.9: Performance Comparison with Different Methods - Cameraman Image.....	47
Figure 4.10: Performance Comparison with Different Methods with different Images.....	47
Figure 4.11: Performance Comparison with Different Methods - Cameraman Image.....	49
Figure 4.12: Performance Comparison with Different Methods (Speckle $v = 0.06$)	49
Figure 4.13: Performance Comparison with Different Methods (Poisson).....	51
Figure 4.14: Cameraman de-noised Image with Roberts Edge detection (Gaussian).....	53
Figure 4.15: Cameraman de-noised Image with Roberts Edge detection (Speckle).....	55
Figure 4.16: Cameraman de-noised Image with Roberts Edge detection (Poisson).....	57
Figure 4.17: Cameraman de-noised Image with Sobel Edge detection (White Gaussian).....	59
Figure 4.18: Cameraman de-noised Image with Sobel Edge detection (Speckle).....	61

Figure 4.19: Cameraman de-noised Image with Sobel Edge detection (Poisson).....	62
Figure 4.20: Performance comparison with Cameraman image (Roberts).....	64
Figure 4.21: Performance comparison with Gaussian noise.....	64
Figure 4.22: Performance comparison with Cameraman image.....	66
Figure 4.23: Performance comparison with Speckle noise.....	66
Figure 4.24: Performance comparison with Poisson noise Image.....	68
Figure 4.25: Performance comparison with Cameraman image (Sobel).....	70
Figure 4.26: Performance comparison with Gaussian noise.....	70
Figure 4.27: Performance comparison with Cameraman image.....	72
Figure 4.28: Performance comparison with Speckle image.....	72
Figure 4.29: Performance comparison with Poisson image.....	85
Figure 4.30: RTL schematic view of FDTT.....	70
Figure 4.31: ModelSim Timing simulation of the FDTT.....	77

LIST OF TABLES

Table 4.1: PSNR (in dB) Performance Table (with white Gaussian).....	46
Table 4.2: PSNR (in dB) Performance Table (with speckle).....	48
Table 4.3: PSNR (in dB) Performance Table (with Poisson).....	50
Table 4.4: PSNR (in dB) Performance Table Roberts (with white Gaussian).....	63
Table 4.5: PSNR (in dB) Performance Table for Cameraman Image (with speckle).....	65
Table 4.6: PSNR (in dB) Performance Table Roberts (with speckle).....	66
Table 4.7: PSNR (in dB) Performance Table Roberts (with Poisson).....	67
Table 4.8: PSNR (in dB) Performance Table Sobel (with white Gaussian).....	69
Table 4.9: PSNR (in dB) Performance Table Sobel (with Speckle).....	71
Table 4.10: PSNR (in dB) Performance Table for Cameraman Image (with speckle).....	71
Table 4.11: PSNR (in dB) Performance Table Sobel (With Poisson).....	73
Table 4.12: Device Utilization Summary FDTT for different Tetrolet partitions.....	75
Table 4.14: Review of No. of clock cycle and timing for FDTT and 2D Haar image decomposition.....	78
Table 4.15: Review of No. of clock cycle and timing for Sobel edge detection algorithm for different image sizes.....	78

ABBREVIATIONS

ASIC – Application Specific Integrated Circuits
AWGN – additive white Gaussian noise
CLBs – Configurable Logic Blocks
dB – Decibel
DSP's – Digital Signal Processors
FDTT – Forward Discrete Tetrolet Transform
FPGA – Field Programmable Gate Arrays
GGD – Generalized Gaussian Distribution
HDL – Hardware Description Language
IDTT – Inverse Discrete Tetrolet Transform
IOBs – Input/Output Blocks
ISE – Integrated Software Environment
LoG – Laplacian of Gaussian
LUT – Look-up tables
NLI – Nonlinear Invariant
PAR – Place and Route
PC – Personal Computer
PET – Positron emission tomography
PSNR – Peak signal-to-noise ratio
RTL – Register Transfer Level
SAR – Synthetic Aperture Radar
SPECT – Single-photon emission computed tomography
SURE – Stein's Unbiased Risk Estimate
VHDL – VHSIC Hardware Description Language
VST – Variance Stabilization
XST – Xilinx Synthesis Technology