

# **HYBRID FRACTIONAL FOURIER AND DUAL TREE COMPLEX WAVELET TRANSFORM FOR IMAGE DENOISING**

## **A DISSERTATION**

*Submitted in partial fulfillment of the  
requirements for the award of the degree*

*of*

**INTEGRATED DUAL DEGREE**

**(Bachelor of Technology & Master of Technology)**

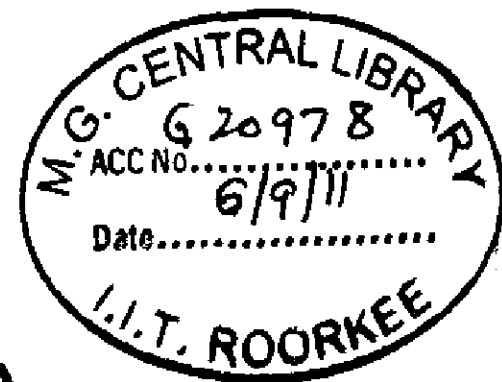
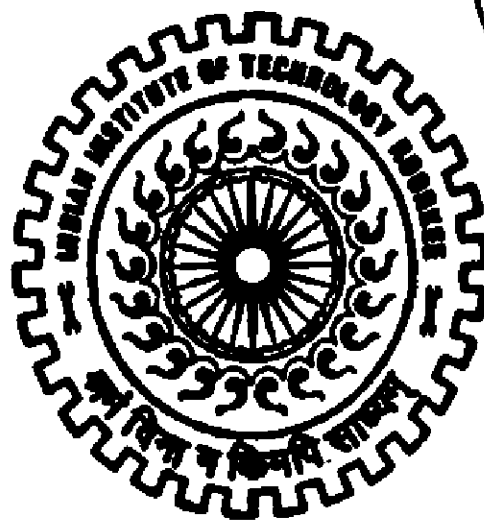
*in*

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**(With Specialization in Wireless Communication)**

*By*

**ARPIT GUPTA**



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE  
ROORKEE - 247 667 (INDIA)**

**JUNE, 2011**

## CANDIDATE'S DECLARATION

---

I hereby declare that the work, which is presented in this dissertation report entitled, **“HYBRID FRACTIONAL FOURIER AND DUAL TREE COMPLEX WAVELET TRANSFORM FOR IMAGE DENOISING”** towards the partial fulfilment of the requirements for the award of the degree of **Integrated Dual Degree** (Bachelor of Technology and Master of Technology) in Electronics and Communication Engineering with specialization in **Wireless Communication**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee (INDIA) is an authentic record of my own work carried out during the period from July 2010 to June 2011, under the guidance of **Dr. DEBASHIS GHOSH, Associate Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee.**

I have not submitted the matter embodied in this dissertation for the award of any other Degree or Diploma.

Date: 30/6/2011

Place: Roorkee

  
ARPIT GUPTA


## CERTIFICATE

---

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 30/6/2011

Place: Roorkee

  
**Dr. Debashis Ghosh,**  
Associate Professor,  
E & C Department,  
IIT Roorkee,  
Roorkee-247667(INDIA)

## ACKNOWLEDGEMENTS

---

---

First of all, I would like to express my deep sense of respect and gratitude towards my guide **Prof. Debashis Ghosh**, who has been the guiding force behind this work. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I would like to thank all faculty members and staff of the Department of Electronics and Computer Engineering, IIT Roorkee for their generous help in various ways for the completion of this thesis.

I am greatly indebted to all my batch mates, who have helped me with ample moral support and valuable suggestions. Most of all I would like to thank my family. Finally, I would like to extend my gratitude to all those persons who directly or indirectly contributed towards this work.

ARPIT GUPTA

## LIST OF ABBREVIATIONS

---

---

DSP	Digital Signal Processing
DIP	Digital Image Processing
SNR	Signal to Noise Ratio
AWGN	Additive White Gaussian Noise
NLM	Non-Local Means
FT	Fourier Transform
STFT	Short Time Fourier Transform
WT	Wavelet transform
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
DTCWT	Dual Tree Complex Wavelet transform
FrFT	Fractional Fourier Transform
FrFT-DTCWT	Hybrid Fractional Fourier and Dual Tree Complex Wavelet Transform
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
TFD	Time Frequency Distribution
LPF	Low Pass Filter
HPF	High Pass Filter

## ABSTRACT

---

---

Denoising plays an important role in image processing which is used to recover a signal/image that has been corrupted by noise. In this Thesis we have shown various denoising algorithms based on spatial and frequency domain filtering, Discrete Wavelet Transform, Dual Tree Complex Wavelet Transform (DTCWT) and Fractional Fourier Transform (FrFT). Based on the study of various algorithms a new hybrid FrFT and DTCWT algorithm has been proposed.

After a thorough study of various denoising techniques, these techniques are then implemented in MATLAB for different types of noises such as Gaussian, Salt and Pepper and Speckle Noise at various noise levels and there simulation results are compared based on Mean Square Error (MSE) criteria and visual interpretation and it has been shown that denoising algorithms depends on the type of noise present in image, hence it is necessary to have prior knowledge about the type of noise present in image so as to select the appropriate denoising algorithm.

Combining the advantages of DTCWT and FrFT, a new hybrid algorithm has been proposed and it proves to be best when noise is of Gaussian or Speckle type whereas Median filter proves to be best when noise is of Salt and Pepper type.

# Table of Contents

Chapter 1	Introduction.....	1
1.1	Need for denoising .....	1
1.2	Types of Noises.....	1
1.2.1	Gaussian Noise.....	2
1.2.2	Salt and Pepper Noise .....	2
1.2.3	Speckle Noise.....	3
1.3	Various Image Denoising Techniques .....	4
1.4	Statement of Problem .....	4
1.5	Thesis Organization.....	5
Chapter 2	Image denoising with Filters in Spatial & Frequency Domain.....	6
2.1	Background .....	6
2.2	Spatial Domain Filters.....	7
2.2.1	Mean Filter.....	7
2.2.2	Median Filter.....	7
2.2.3	Non-Local Means.....	8
2.2.4	Wiener Filter (Adaptive Filter) .....	10
2.3	Frequency Domain Wiener Filter.....	11
Chapter 3	Image denoising using Discrete Wavelet Transform.....	14
3.1	Background .....	14
3.2	Continuous Wavelet Transform (CWT).....	15

3.3	Discrete Wavelet transform (DWT).....	16
3.4	Denoising Algorithm: .....	20
3.5	Thresholding .....	20
3.5.1	VisuShrink [5] .....	22
3.5.2	BayesShrink[5] .....	23
Chapter 4	Hybrid Fractional Fourier and Dual Tree Complex Wavelet Transform for Image Denoising. ....	25
4.1	Fractional Fourier Transform.....	25
4.1.1	Definition of Fractional Fourier Transform[20,21].....	26
4.1.2	Denoising using FrFT [22] .....	26
4.2	Dual Tree Complex Wavelet Transform.....	28
4.3	Denoising using Hybrid FrFT and DT CWT.....	30
Chapter 5	Simulation Results and Discussion.....	31
5.1	Simulation Results .....	31
5.2	Discussion.....	44
Chapter 6	Conclusion:.....	45
Works Cited	.....	46

## List of Figures

Figure 1-1 Clean Image .....	2
Figure 1-2 Corrupted Image with Gaussian Noise (Mean=0, Variance=400) .....	2
Figure 1-3 Clean Image .....	3
Figure 1-4 Corrupted Image with Salt & Pepper Noise (D=0.2) .....	3
Figure 1-5 Clean Image .....	4
Figure 1-6 Corrupted Image with Speckle Noise (V=25).....	4
Figure 2-1 Example of self-similarity in an image .....	8
Figure 3-1 STFT windows, narrow in time domain means wide in frequency domain and vice-versa. ....	14
Figure 3-2 Time Frequency occupation by wavelet transform .....	15
Figure 3-3 1D signal decomposition.....	17
Figure 3-4 1D Multilevel decomposition.....	18
Figure 3-5 A one level 2-Dimensional DWT.....	19
Figure 3-6 One level 2D DWT Reconstruction .....	20
Figure 3-7An example of (a) linear signal thresholded using, (b) hard-thresholding, and (c) soft-thresholding. ....	21
Figure 3-8 DWT of 2D data.....	22
Figure 4-1 Noise Separation in the $\alpha^{\text{th}}$ domain [20].....	28
Figure 4-2 one level complex dual tree.....	28



Figure 5-1 Lena and Peppers Images to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Gaussian Noise with Variance =  $20^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid..... 33

Figure 5-2 Ultrasound and SAR Images to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Gaussian Noise with Variance =  $20^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid ..... 34

Figure 5-3 Lena and Peppers Image to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Gaussian Noise with Variance =  $25^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid..... 36

Figure 5-4 Ultrasound and SAR Images to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Gaussian Noise with Variance =  $25^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid..... 37

Figure 5-5 Lena and Peppers Image to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Salt & Pepper Noise with  $D=.03$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid..... 39

Figure 5-6 Ultrasound and SAR Images to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Salt & Pepper Noise with  $D=.03$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid..... 40

Figure 5-7 Lena and Peppers Image to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Speckle Noise with Variance =  $1^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid..... 42

Figure 5-8 Ultrasound and SAR Image to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Speckle Noise with Variance =  $1^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid..... 43

## List of Tables

Table 4-1 First level Coefficients of the analysis filter.....	29
Table 4-2 Remaining Levels Coefficients of the Analysis Filters .....	29
Table 5-1 Comparison of various denoising algorithms in presence of Gaussian Noise with $\sigma=20$ for different images .....	32
Table 5-2 Comparison of various denoising algorithms in presence of Gaussian Noise with $\sigma=25$ for different Images.....	35
Table 5-3 Comparison of various denoising algorithms in presence of Salt & Pepper Noise with Noise Density $D=.03$ for different Images .....	38
Table 5-4 Comparison of various denoising algorithms in presence of Speckle Noise with Noise Variance $V=1$ for different Images .....	41

## Chapter 1 Introduction

### 1.1 Need for denoising

Digital images are essential part of contemporary life. Some of the most frequently used images types are binary (takes only two discrete values '0' for black and '1' for white), gray scale images (monochrome or one color image) and color images (three band monochrome images). However, due to natural phenomenon such as transmission errors or imperfect acquisition tend to distort the images. This distortion includes blurring of image or adding additive or multiplicative noise as discussed in subsection 1.2. Thus arises the need for image denoising in order to suppress or completely remove the noise from the degraded image. Image denoising is mostly used in field of photography in which an image somehow gets degraded and needs to be denoised before it can be printed. In order to apply different denoising techniques (mentioned in subsection 1.3 and discussed in detail in Chapter 2, 3 and 4) we need to know the degradation process to develop a model for it. Image denoising finds application in fields such as astronomy, medical imaging, forensic science and many other fields of contemporary life.

### 1.2 Types of Noises

In this subsection we discuss noise commonly present in an image. Noise is undesired information that contaminates the image. Noise is present in an image either in an additive or multiplicative form [1]

An additive noise follows the rule

$$w(i, j) = s(i, j) + n(i, j) \quad (1.1)$$

while the multiplicative noise satisfies

$$w(i, j) = s(i, j) \times n(i, j) \quad (1.2)$$

where  $s(i, j)$  is the original signal,  $n(i, j)$  denotes the noise introduced into the signal to produce the corrupted image  $w(i, j)$ , and  $(i, j)$  represents the pixel location.

### 1.2.1 Gaussian Noise

Gaussian noise is normally distributed over the signal/image i.e 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, whose probability distribution function given by:

$$F(g) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(g-m)^2} \quad (1.3)$$

where  $g$  represents the gray level,  $m$  is the mean or average of the function, and  $\sigma$  is the standard deviation of the noise. When introduced into an image (peppers), Gaussian noise with zero mean and variance as 400 would look as in Figure 1-2.



Figure 1-1 Clean Image



Figure 1-2 Corrupted Image with Gaussian Noise (Mean=0, Variance=400)

### 1.2.2 Salt and Pepper Noise

Salt and pepper noise is an impulse type of noise, also referred to as intensity spikes. This is caused generally due to errors in data transmission. The corrupted pixels are set alternatively to the minimum or to the maximum value ('0' or '255' in case of gray scale images), giving the image a "salt and pepper" like appearance. Unaffected pixels remain unchanged. It has only two possible values 'a' and 'b', both with probability less than 0.1.

For an 8-bit image gray scale image, the typical value for pepper noise is 0 and for salt noise 255.

When introduced in an image (peppers), Salt and Pepper noise with noise density,  $D = 0.2$  would look as shown in Figure 1-4. It will affect  $D \times N$  pixels where  $N$  is the number of pixels in image.



Figure 1-3 Clean Image



Figure 1-4 Corrupted Image with Salt & Pepper Noise ( $D=0.2$ )

### 1.2.3 Speckle Noise

Another common form of noise is speckle noise. This noise is, in fact, caused by errors in data transmission. The corrupted pixels are either set to the maximum value, which is something like a snow in image or have single bits flipped over. This kind of noise affects the ultrasound images/SAR images. Speckle noise has the characteristic of multiplicative noise. Speckle noise follows a gamma distribution and is given as:

$$F(g) = \frac{g^{\alpha-1}}{(\alpha-1)!a^\alpha} e^{-g/a} \quad (1.4)$$

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

It adds multiplicative noise to the image  $s$ , using the equation  $w = s + (n)x(s)$ , where  $n$  is uniformly distributed random noise with mean 0 and variance  $V$ . When introduced in an image (peppers), Speckle noise with variance  $V = 25$  would look as shown in Figure 1-6



Figure 1-5 Clean Image

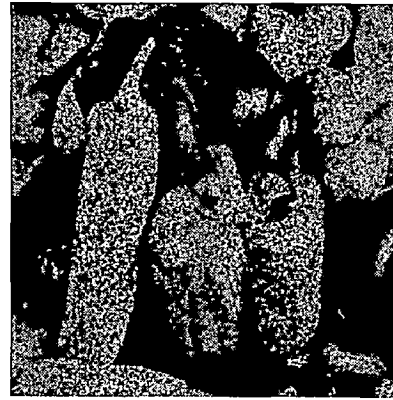


Figure 1-6 Corrupted Image with Speckle Noise ( $V=25$ )

### 1.3 Various Image Denoising Techniques

Over the years several denoising techniques have been implemented. These includes applying filters in spatial domain such as Mean, Median, Wiener and Non Local means filter [2], filters in frequency domain like Wiener filter [3,4] which acts as a low pass filter and remove the high frequency components present as noise, filtering in Wavelet domain [5] using several threshold and shrinkage techniques [4,6,7] on discrete wavelet and complex wavelet transform[8,9-11] and applying filters in Fractional domain such as in Fractional Fourier Transform [12]. All these techniques have been discussed in detail in the following chapters. There are few other techniques which have not been discussed in this thesis.

### 1.4 Statement of Problem

The basic idea behind this thesis is to propose a new denoising algorithm after a thorough study of various conventional denoising algorithms.

There are various denoising techniques to reconstruct a noise free image from noise present in a clean image. Selecting the appropriate method plays a major role in getting the desired denoised image. The denoising methods depends on the problem i.e. type of noise. For example, a method that is used to denoise images corrupted with Gaussian noise may not be suitable for denoising images corrupted with Salt & Pepper noise. In this thesis, a study is made on the various denoising algorithms and each is implemented in MATLAB. In order to quantify the performance of the various denoising algorithms, several high quality images are taken and some known noise is added to it. Then denoising algorithms are applied to these noisy images, which produces an image close to the original high quality image. The

performance of each algorithm is compared by computing Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) besides the visual interpretation.

At the end after comparing various techniques and noticing the advantages and disadvantages of various denoising algorithms a new algorithm has been proposed which gives better performance over the other denoising algorithms both visually and in terms of MSE and PSNR. In this new algorithm we combined the benefits of Fractional Fourier Transform and Dual Tree Complex Wavelet Transform and tried to remove their drawbacks.

## **1.5 Thesis Organization**

In Chapter 2 we have discussed several spatial domain filters such as Mean, Median, Non-Local Means and Wiener Filter and a frequency domain wiener filter, these filters can again be regrouped as Linear, Non-Linear and Adaptive filters. We have shown how these filters can be implemented and the results of their implementation are then shown in chapter 5.

In Chapter 3 we introduced the concept of Image denoising using Discrete Wavelet Transform. We thoroughly discussed what is meant by DWT, how DWT can be implemented for Image denoising and the various thresholding and shrinkage method that plays an important role in denoising of an image.

In Chapter 4 we define Fractional Fourier Transform and Dual Tree Complex Wavelet Transform and their implementation and algorithms for image denoising, their superiority over other conventional methods and then on the basis of this discussion we proposed a new hybrid algorithm combining advantages of both FrFT and DT CWT, thereby removing their shortcomings.

In Chapter 5 implementation of all the denoising techniques discussed in earlier chapters is done in MATLAB and the simulation results are presented along with the discussion of the results obtained for various denoising techniques. The algorithms are applied on four different Images with all 3 types of noises i.e. Gaussian, Salt & Pepper and Speckle Noise.

Finally In Chapter 6 we have concluded our thesis based on the results obtained and the scope for future work followed by references.

## Chapter 2 Image denoising with Filters in Spatial & Frequency Domain

The need to remove noise without degrading the edges and other high frequency components of the image significantly has motivated the development of efficient edge-preserving noise smoothing techniques. Over the years significant progress and development has been made in developing several image denoising techniques. In this chapter we have discussed Mean, Median, Non-Local Means (NLM) and Wiener filters applied in spatial domain, and implementation of Wiener filter in frequency domain. The results of all these filters are later discussed in Chapter 5 when applied on noisy images.

### 2.1 Background

Filters play an important role in image denoising process. The basic concept behind image denoising using linear filters is digital convolution and moving window principle. Let  $w(x)$  be the input noisy signal which is to be filtered, and  $z(x)$  be the filtered output. Thus filtered output can be expressed mathematically in simple form as:

$$z(x) = \int w(t)h(x-t)dt \quad (2.1)$$

Where  $h(t)$  represents the impulse response of the filter and integral is the convolution of input noisy signal with filter.

For discrete case:

$$z(i) = \sum_{t=-\infty}^{\infty} w(t)h(i-t) \quad (2.2)$$

as seen the limits is  $\infty$  but the weights  $h(t)$  are zero outside some range  $(-k, +k)$ , thus above equation can be written as:

$$z(i) = \sum_{t=i-k}^{i+k} w(t)h(i-t) \quad (2.3)$$

that is output  $z(i)$  at point  $i$  is given by a weighted sum of neighborhood pixels of  $i$  where the weights are given by filter response  $h(t)$ . To find the output at the next pixel  $i+1$ , the function  $h(t)$  is shifted by one and the weighted sum is recalculated. The total output is created by a series of shift-multiply-sum operations, and this forms a discrete convolution. For the 2-dimensional case,  $h(t)$  is  $h(t, u)$ , and output in discrete form is given as:



$$z(i, j) = \sum_{t=i-k}^{i+k} \sum_{u=j-k}^{j+k} w(t, u)h(i-t, j-u) \quad (2.4)$$

In DIP the weight,  $h(t, u)$  may be defined arbitrarily thus different types of filters such as Mean, Median, Wiener and NLM filters can be implemented.

## 2.2 Spatial Domain Filters

### 2.2.1 Mean Filter

A mean filter reduces the noise in the noisy image by taking the mean of the adjacent neighboring pixels i.e. it's a simple sliding window spatial filter that replaces the centre pixel of the window by taking the average mean of all the pixels in the window being considered. Generally a 3x3 or 5 x 5 sliding window is used. This filter works on the shift multiply sum principle.

The local mean filter uses equal weights,  $h(t, u)$  to yield a smoothed estimate:

$$z(i, j) = \frac{1}{NM} \sum_{t=i-k}^{i+k} \sum_{u=j-k}^{j+k} w(t, u) \quad (2.5)$$

where NM is the number of pixels in the window and  $w(t, u)$  is the noisy image. It can be easily shown that in the case of AWGN with zero mean and variance of  $\sigma^2$  the noise power in the filtered output is reduced by number of pixels in the window i.e. NM, thereby resulting in improvement of output SNR by a factor of NM. This implies that for better SNR larger window mask should be selected however if we take large window size it introduces denoising artifact of image blurring.

Mean filtering is more effective when the noise present in an image is of impulsive type. The mean filter works like a low pass filter (LPF), and it does not allow the high frequency components present in the noise to pass through. It is to be noted that larger windows of size 5× 5 or 7×7 produces more denoising but make the image more blurred. A trade-off is to be made between the kernel size and the amount of denoising.

### 2.2.2 Median Filter

A median filter is a type of nonlinear filters unlike the mean filter and the wiener filter (described in section 2.2.4). Median filter also follows the moving window principle similar to the mean filter. A  $n \times n$  (where n is odd) kernel of pixels is scanned over pixel matrix of

the entire image. The median of all the pixels in the window is calculated and the centre pixel value of the window is replaced with the calculated median. Median filtering is done by, first sorting all the pixel values from the surrounding neighbourhood in either ascending or descending order and then replacing the centre pixel which is being considered with the median value of the neighborhood window.

The median is more robust compared to the mean. Thus, a single very unrepresentative pixel in a neighborhood will not affect the median value. Since the median value is the value of one of the pixels in the neighbourhood (window), the median filter does not create new pixel values. For this reason the median filter is much better at preserving sharp edges than the mean filter. These advantages led median filters in denoising uniform noise along with sharp impulse type of noise as well from an image.

### 2.2.3 Non-Local Means

NLM denoising method is based on the self-similarity of images in spatial domain [2]. The non-local means algorithm assumes that the image contains an extensive amount of redundancy. These redundancies can then be exploited to remove the noise in the image.

An example of such a type of self-similarity is displayed in Figure 2-1 below. The figure shows three pixels  $p$ ,  $q1$ , and  $q2$  and their respective neighborhoods. The neighborhoods of pixels  $p$  and  $q1$  are similar, but the neighborhoods of pixels  $p$  and  $q2$  are not similar. Adjacent pixels tend to have similar neighborhoods, but non-adjacent pixels can also have similar neighborhoods only in some of the particular cases as for example, in Figure 2-1 most of the pixels in the same column as  $p$  will have similar neighborhoods to  $p$ 's neighborhood. The self-similarity assumption can be exploited to denoise an image. Pixels with similar neighborhoods can be used to determine the denoised value of a pixel.

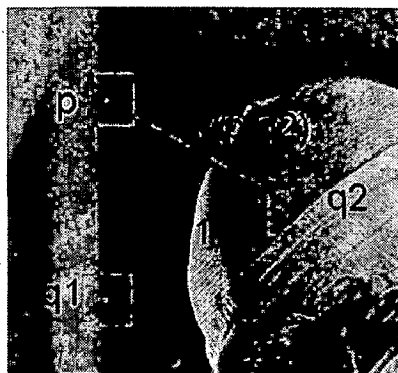


Figure 2-1 Example of self-similarity in an image

## Implementation:

Given a discrete gray noisy image  $w$ , the estimated value  $NL[w](i)$ , for each pixel  $i$ , is computed as a weighted average of all the pixels in the image:

$$NL[w](i) = \sum_{j \in I} h(i,j)w(j) \quad (2.6)$$

where the weights  $\{h(i,j)\}$  depends on the similarity between the pixels  $i$  and  $j$ , and satisfies  $0 \leq h(i,j) \leq 1$  and  $\sum_j h(i,j) = 1$ . Each pixel is a weighted average of all the pixels in the image. The weights are based on the similarity between the neighborhoods of pixels  $i$  and  $j$ . To compute similarity between the pixels, a square neighborhood  $N_i$  with radius  $R_{sim}$  is defined centered about a pixel  $i$ . This similarity is measured as a decreasing function of the weighted Euclidean distance,  $\|y(N_i) - y(N_j)\|_{2,a}^2$  where  $a > 0$  is the standard deviation of the Gaussian kernel. These weights are defined as:

$$h(i,j) = \frac{1}{Z(i)} \exp(-\|y(N_i) - y(N_j)\|_{2,a}^2 / k^2) \quad (2.7)$$

where  $Z(i)$  is the normalizing constant given as:

$$\sum_j \exp(-\|y(N_i) - y(N_j)\|_{2,a}^2 / k^2) \quad (2.8)$$

and  $k$  is the weight-decay control parameter and  $k \approx 10\sigma_n$ , which controls the decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances.

The parameter  $a$  is the neighborhood filter with radius  $R_{sim}$ . The weights of  $a$  are computed by the following formula:

$$\frac{1}{R_{sim}} \sum_{i=m}^{R_{sim}} \frac{1}{(2i+1)^2} \quad (2.9)$$

where  $m$  is the distance of the weight which is from the center of the filter.

Implementing NLM basically has three parameters.

- The first parameter,  $k$ , is the weight-decay control parameter which controls where the weights lay on the decaying exponential curve. If  $k$  is set too low, not much of the noise will be removed. Setting  $k$  too high results in blurring of the image. When

an image contains white noise with a standard deviation of  $\sigma_n$ ,  $k$  should be set between  $10 \sigma_n$  and  $15 \sigma_n$ .

- The second parameter,  $R_{sim}$ , is the radius of the neighborhoods used to find the similarity between two pixels. If  $R_{sim}$  is too large, no similar neighborhoods will be found, but if it is too small there will be too many similar neighborhoods. Common values for  $R_{sim}$  are 3 and 4 to give neighborhoods of size  $7 \times 7$  and  $9 \times 9$ , respectively.
- The third parameter,  $R_{win}$ , is the radius of a search window. Because of the involvement of lot computation of taking the weighted average of every pixel for every pixel, it will be reduced to a weighted average of all pixels in a window. The window is centered at the current pixel being computed. Common values for  $R_{win}$  are 7 and 9 to give windows of size  $15 \times 15$  and  $19 \times 19$ , respectively. With this change the algorithm will take a weighted average of  $15^2$  pixels rather than a weighted average of  $N^2$  pixels for a  $N \times N$  image.

#### 2.2.4 Wiener Filter (Adaptive Filter)

Wiener filter is a type of adaptive filter[13] which is an improvement over the mean filter. The main difference between the mean filter and the adaptive filter is that the weight matrix of the filter varies after each iteration in the adaptive filter while it remains constant throughout the iterations in the mean filter. Adaptive filters are capable of denoising non-stationary images, that is, images that have abrupt changes in intensity.

##### Implementation:

Let  $s(i,j)$  be the original image and  $n(i,j)$  is the noise and  $w(i,j)$  is the noisy image which need to be filtered i.e

$$w(i,j) = s(i,j) + n(i,j) \quad (2.10)$$

Let the filtered image be  $z(i,j)$  which should be equal to  $s(i,j)$  ideally.

$$z(i,j) = \sum_{m=-N}^N \sum_{n=-N}^N h(m,n)w(i+m,j+n) \quad (2.11)$$

The weights of the Wiener filter,  $h(m,n)$ , can be found by minimizing:

$$J = E[\{s(i,j) - z(i,j)\}^2] \quad (2.12)$$

where  $E$  denotes expectation. The solution for  $h(m, n)$  is obtained in a vector form as:

$$\mathbf{h} = \mathbf{R}^{-1}\mathbf{p} \quad (2.13)$$

$R(m, n)$  and  $p(m, n)$  correspond to the autocorrelation function of  $w(i, j)$  and cross-correlation function of  $s(i, j)$  and  $w(i, j)$ , respectively, which are given by:

$$R(m, n) = E[w(i, j)w(i - m, j - n)] \quad (2.14)$$

$$p(m, n) = E[s(i, j)w(i - m, j - n)] \quad (2.15)$$

respectively. In practice, the following criterion

$$J = \frac{1}{M^2} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \{s(i, j) - z(i, j)\}^2 \quad (2.16)$$

is often defined due to its easy computation, and the solution for filter weights ( $h$ ) as defined in equation (2.13) is obtained. In this case,  $R(m, n)$  and  $p(m, n)$  are calculated as

$$R(m, n) = \frac{1}{M^2} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} (w(i, j)w(i - m, j - n)) \quad (2.17)$$

$$p(m, n) = \frac{1}{M^2} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} (s(i, j)w(i - m, j - n)) \quad (2.18)$$

respectively, from  $M \times M$  images of  $s(i, j)$  and  $w(i, j)$ .

### 2.3 Frequency Domain Wiener Filter

One of the main difficulties associated with spatial domain filters is their computational complexity required to perform the convolution. Frequency domain filters overcome this problem as in Fourier domain convolution is transformed into multiplication. Most of the information in an image is mainly present in low frequencies, and the noise is generally located in high frequencies or spread across all frequencies (white noise), thus frequency domain filters in order to perform denoising represents some form of low pass filters to reduce most of the high-frequency components in order to denoise the image. However, it has a drawback as along with noise it also removes some information of an image that are contained in high frequency. Thus, it becomes difficult to suppress the noise without degradation of some of the significant features of the image like edges and textures. Therefore, most of the general frequency-based image denoising methods results in overly

smoothed denoised images where the noise has been reduced but also edges and other high-frequency features of the image have been blurred.

**Implementation [13]:**

Let  $Z(u, v), H(u, v), S(u, v)$  and  $W(u, v)$  represent the discrete Fourier transforms (DFTs) of  $z(i, j)$  (filtered output),  $h(i, j)$  wiener filter weights,  $s(i, j)$  (clean image) and  $w(i, j)$  (noisy image), respectively.

In frequency domain the output of wiener filter is given as:

$$z(u, v) = H(u, v)W(u, v) \quad (2.19)$$

(As convolution in spatial domain is multiplication in frequency domain)

by minimizing the wiener cost function by partial differentiating it w.r.t  $H(u, v)$  and equating to zero:

$$J = E[\{S(u, v) - Z(u, v)\}^2] = E[\{S(u, v) - H(u, v)W(u, v)\}^2] \quad (2.20)$$

we get:

$$H(u, v) = \frac{E[W(u, v)S^*(u, v)]}{E[|W(u, v)|^2]} \quad (2.21)$$

where  $*$  denotes complex conjugate. When noise is white noise, the numerator reduces to

$$E[W(u, v)S^*(u, v)] = E[(S(u, v) + N(u, v))X S^*(u, v)] = E[|S(u, v)|^2] = P_S(u, v) \quad (2.22)$$

and the denominator reduces to

$$E[|W(u, v)|^2] = P_S(u, v) + P_N(u, v) \quad (2.23)$$

where  $P_S(u, v)$  and  $P_N(u, v)$  correspond to the power spectra of  $s(i, j)$  and  $n(i, j)$ , respectively.

Therefore:

$$H(u, v) = \frac{P_S(u, v)}{P_S(u, v) + P_N(u, v)} \quad (2.24)$$

The output of the Wiener filter is given by:

## Chapter 3 Image denoising using Discrete Wavelet Transform

### 3.1 Background

The limitations of Fourier transform make the use of Wavelet Transform important. Fourier transform is unable to provide information in time and frequency domain simultaneously. This means that Fourier transform provides us with the frequency components, but it does not tell when and where the impulse occurred. However, in Image processing, we require time and frequency information of the signal simultaneously. To overcome this problem, we divide the signal into different parts and analyze frequency components of these parts separately. One of the solutions to this problem was Short Term Fourier Transform (STFT). In STFT we move a window across the signal to analyze the frequency domain of the signal for each part. The problem still remains since STFT has a fixed resolution. Thus measuring frequency and time cannot be done simultaneously at a desired resolution. A wide window gives better frequency resolution but poor time resolution. A narrower window, on the other hand, gives good time resolution but poor frequency resolution.

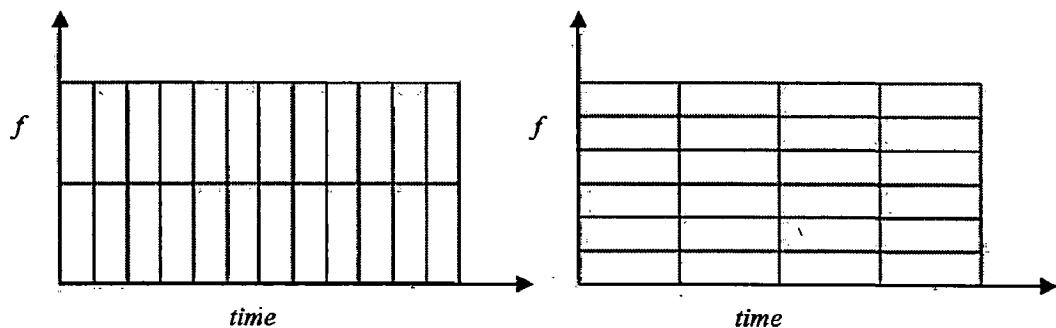


Figure 3-1 STFT windows, narrow in time domain means wide in frequency domain and vice-versa.

The solution to this problem was provided in terms of Wavelet Transform. Wavelet transform uses a scalable modulated window to move across the signal. Using a small scale for high frequency parts and a big scale for low frequency parts enables wavelet transform to provide a good time and frequency resolution [14].

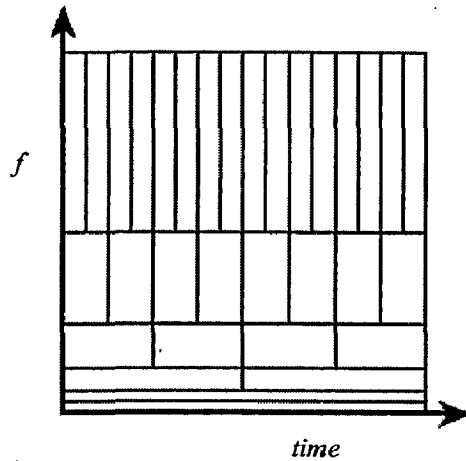


Figure 3-2 Time Frequency occupation by wavelet transform

### 3.2 Continuous Wavelet Transform (CWT)

CWT is defined basis functions which are very short (high frequency) and long (low frequency) depending on the resolution of the frequency analysis, where the basis functions are obtained from a single prototype wavelet also known as Mother wavelet

$$W_{a,b}(t) = \frac{1}{\sqrt{a}} W\left(\frac{t-b}{a}\right) \quad (3.1)$$

where  $a \in R^+$ ,  $b \in R$ . For large  $a$ , the basis function becomes a stretched version of the prototype wavelet which is a low frequency function, while for small  $a$  the basis function becomes a contracted wavelet which is a high frequency function. The continuous wavelet transform (CWT) is then defined as the convolution of  $x(t)$  with a wavelet function,  $W(t)$ , shifted in time by a translation parameter  $b$  and a dilation parameter  $a$ .

$$X_w(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} W\left(\frac{t-b}{a}\right) x(t) dt \quad (3.2)$$

At high frequencies the CWT is sharper in time while at low frequencies the CWT is sharper in frequency.  $W(t)$  denotes the mother wavelet. The parameter  $a$  represents the scale index that is the reciprocal of the frequency. The parameter  $b$  indicates the time, shifting (or translation).



### 3.3 Discrete Wavelet transform (DWT)

CWT is redundant since the parameters  $(a, b)$  are continuous thus we need to discretize the grid on the time-scale plane corresponding to a discrete set of continuous basis functions to get DWT.

$$W_{j,k}(t) = \frac{1}{\sqrt{a_j}} W\left(\frac{t-b_k}{a_j}\right) \quad (3.3)$$

The above equation can be written as:

$$W_{j,k}(t) = a_0^{-\frac{j}{2}} w(a_0^{-j}t - kb_0) \quad (3.4)$$

Where  $a_j = a_0^j$  and  $b_k = kb_0 a_0^j$  and  $j, k \in Z, a_0 > 1$  and  $b_0 \neq 0$  and in the discrete form of the wavelet as shown in equation (3.4),  $j$  controls the dilation and  $k$  controls the translation. Two popular choices for the discrete wavelet parameters  $a_0$  and  $b_0$  are 2 and 1 respectively, a configuration that is known as the *dyadic grid* arrangement

Wavelet analysis decomposes a signal into time shifted and scaled versions of a mother wavelet,  $W_{j,k}(t)$ . Wavelet analysis results in perfect reconstruction, in which a decomposed signal or image is reassembled into its original form without loss of information. We normally use two types of basis functions for decomposition and reconstruction. These functions are:

- Scaling Function:

$$\Phi_{j,k}(t) = 2^{-\frac{j}{2}} \Phi_0(2^{-j}t - k) \quad (3.5)$$

- Wavelet:

$$W_{j,k}(t) = 2^{-j/2} \Psi_0(2^{-j}t - k) \quad (3.6)$$

An example of a simple wavelet function is called the Haar wavelet. The Haar transform stands for the simplest algorithm enabling signal or image denoising. In Haar's case it is a square wave. The Haar mother wavelet  $W(t)$  and scaling function  $\Phi(t)$  are defined as follows [15]:

$$\phi(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$W(t) = \begin{cases} 1 & 0 \leq t \leq 1/2 \\ -1 & 1/2 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (3.7)$$

Every basis function  $W$  is orthogonal to every basis function  $\Phi$ . Wavelets are functions defined over a finite interval and have an average value of zero.

The discrete wavelet transform (DWT) is commonly implemented with sets of filters that divide a signal frequency band into sub bands. At each scale in DWT, the approximation coefficients are generated from a low pass filter and are associated with the low frequency part of the signal while the detail coefficients are output from a high-pass filter and capture the high frequency components of the signal.

The 1-D forward wavelet transform of a discrete-time signal  $x(n)$  ( $n = 0, 1, \dots, N$ ) is performed by convolving signal  $x(n)$  with both a half-band low-pass filter  $L$  and high-pass filter  $H$  and down sampling by two as shown below Figure 3-3. This provides the coefficients  $c_f(k)$  and  $d_j(k)$  for the decomposition of the signal into its scaling function and wavelet function components.

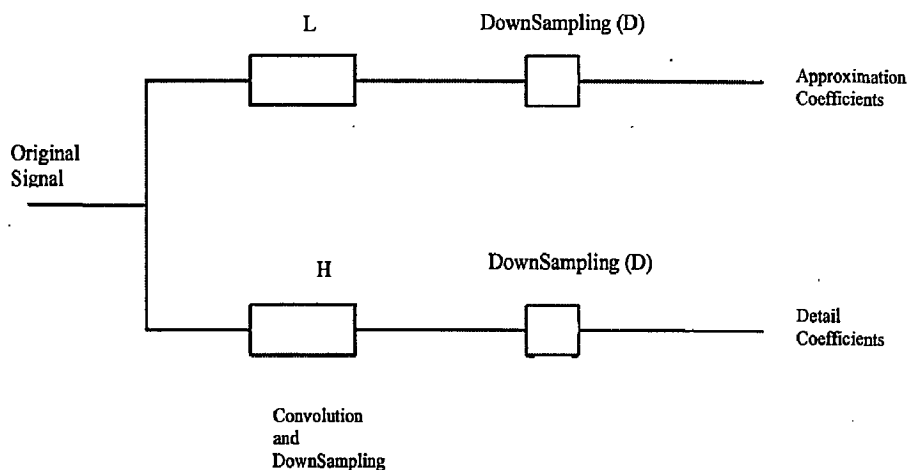


Figure 3-3 1D signal decomposition

$$c(n) = \sum_{k=0}^{L-1} h_0(k)x(n-k) \text{ and } d(n) = \sum_{k=0}^{L-1} h_1(k)x(n-k) \quad (3.8)$$

where  $c(n)$  represent the approximation coefficients for  $n = 0, 1, 2, \dots, N - 1$  and  $d(n)$  are the detail coefficients,  $h_0$  and  $h_1$ , are coefficients of the discrete-time filters  $L$  (Low Pass filter) and  $H$  (high pass filter).

$$\{ h_0(n)_{n=0}^{L-1} = (h_0(0), h_0(1), \dots, h_0(L - 1))$$

$$\{ h_1(n)_{n=0}^{L-1} = (h_1(0), h_1(1), \dots, h_1(L - 1))$$

The high pass filter leads to  $W(t)$  and the low pass filter leads to a scaling function  $\Phi(t)$ . A filter bank is a set of filters in which the analysis bank often has two filters, low pass  $H_0$  and high pass  $H_1$ . They split the input signal into frequency bands. The filtered outputs from both filters give a double signal length. To overcome this we have to down sample or decimate the signal by 2.

In multilevel decomposition, this process is repeated, with successive approximations (the output of the low-pass filter in the first bank) being decomposed in turn i.e. approximate coefficients that are obtained from LPF are decomposed again, so that one signal is broken down into a number of components. This is called the Mallat algorithm or Mallat-tree decomposition. A three-level decomposition is shown in Figure 3-4. In this illustration,  $a_3$  represents the approximation coefficients, while  $d_3, d_2$  and  $d_1$  represent the detail coefficients resulting from the three-level decomposition.

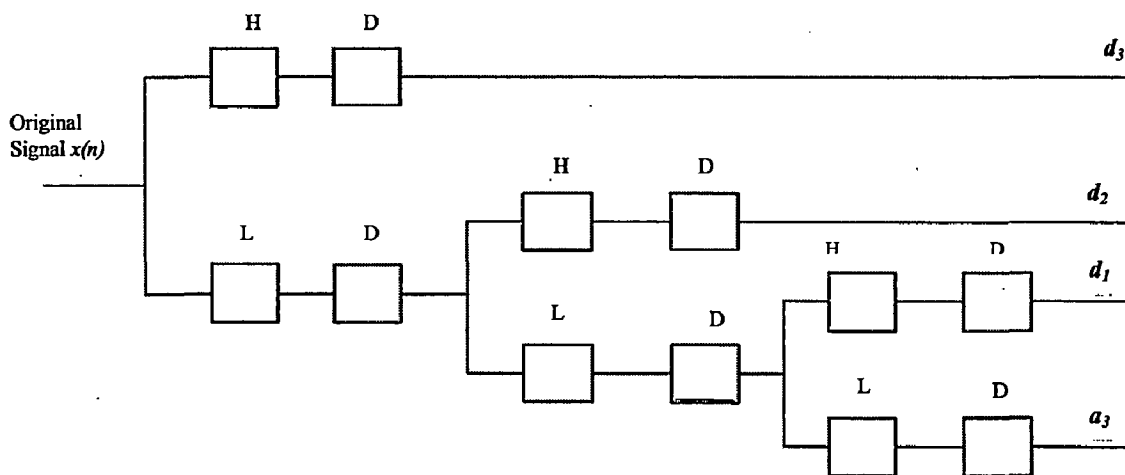


Figure 3-4 1D Multilevel decomposition

2-D signals such as digital images require a two-dimensional wavelet transform. The 2-D DWT analyzes an image across rows and columns in order to separate horizontal, vertical and diagonal details. In the first stage the rows of 2D ( $N \times N$ ) signal are filtered using a high pass

and low pass filters. In the second stage 1-D convolution of the filters with the columns of the filtered image is applied. Each of the branches in the tree is shown in the Figure 3-5 therefore produces an  $(N/2) \times (N/2)$  sub image. This leads at each level to 4 different sub bands  $HH$ ,  $HL$ ,  $LH$  and  $LL$ .

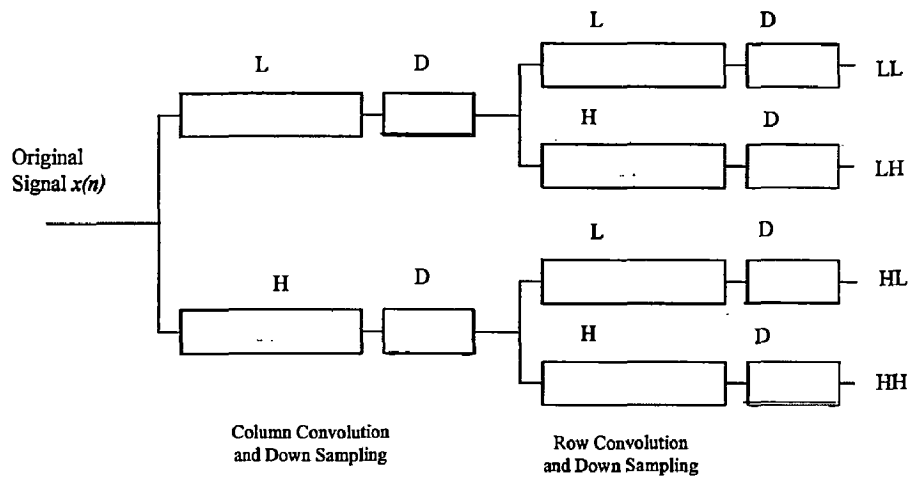


Figure 3-5A one level 2-Dimensional DWT

To reconstruct the image from its 2-D DWT sub images (LH, HL, HH) the detail coefficients are recombined with the low pass approximation using up sampling and convolution as shown in Figure 3-6. In the first stage the columns of the up sampled sub images are convolved with the impulse responses  $h_0^T(k)$  and  $h_1^T(k)$  and in the second stage the rows of the up sampled sums are convolved with the same impulse responses.

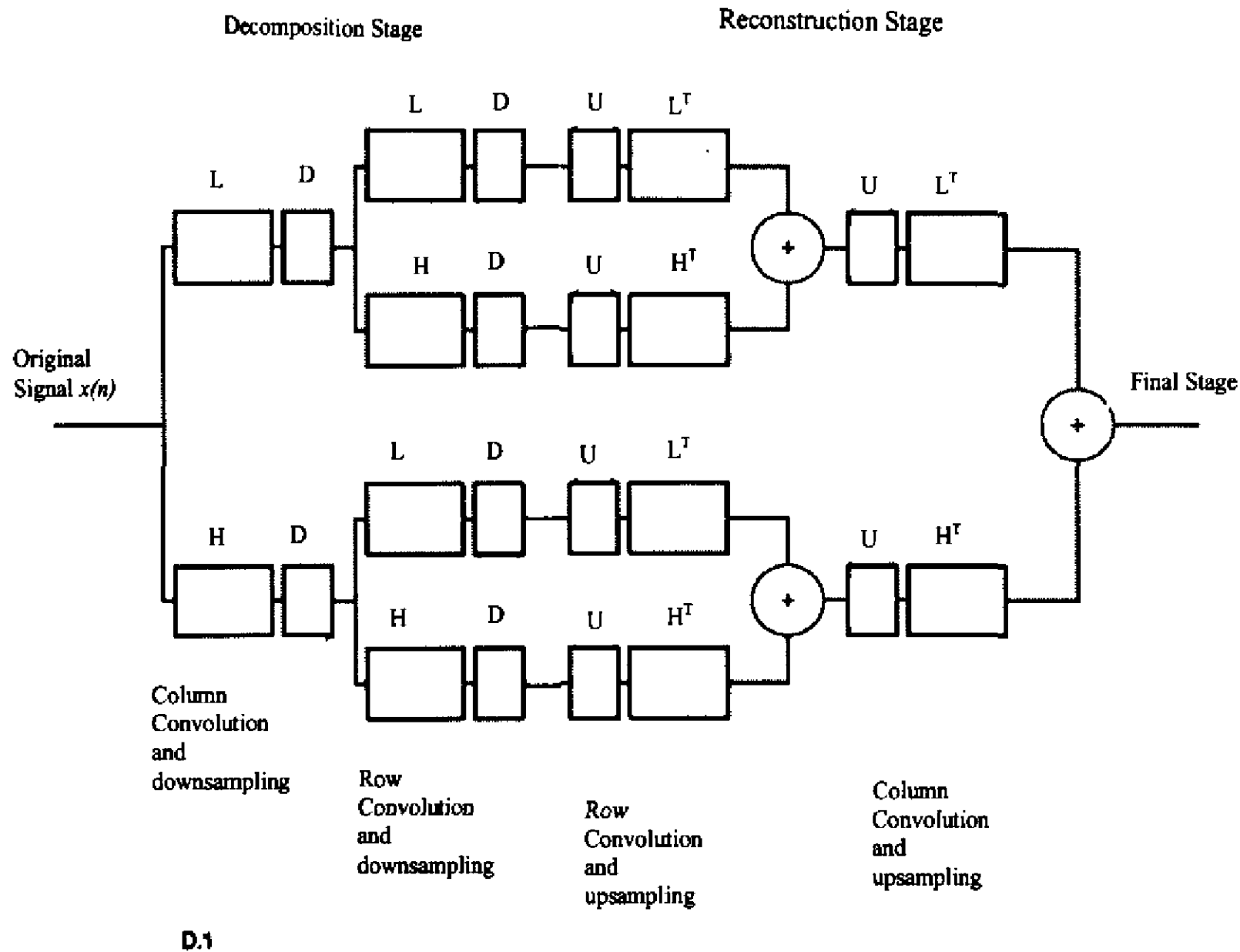


Figure 3-6 One level 2D DWT Reconstruction

### 3.4 Denoising Algorithm:

- Choice of a wavelet (e.g. Haar, symmlet, etc) and number of levels for the decomposition.
- Estimation of a threshold
- Choice of a shrinkage rule (VisuShrink, BayesShrink etc) and application of the threshold (hard and soft threshold) to the detail coefficients.
- Application of the inverse DWT using the thresholded coefficients.

### 3.5 Thresholding

Donoho and Johnstone [16] worked on filtering of additive AWGN using wavelet thresholding. The term wavelet thresholding means comparing the detail coefficients with a given threshold value, and shrinking these coefficients close to zero to take away the effect of noise in the data. The image is reconstructed from the modified coefficients. This process is also known as the inverse discrete wavelet transform. During thresholding, a wavelet coefficient is compared with a given threshold and is set to zero if its magnitude is less than the threshold; otherwise, it is retained or modified depending on the threshold rule. Some

typically used methods for image noise removal include VisuShrink and BayesShrink [16-18].

Prior to the discussion of these methods, it is necessary to know about the two general categories of thresholding. They are hard- thresholding and soft-thresholding types.

### Hard Thresholding

Hard thresholding (Figure 3-7(b)) deletes all coefficients that are smaller than the threshold  $\lambda$  and keeps the others unchanged. The hard thresholding is defined as follows:

$$c_h(k) = \begin{cases} \text{sign}c(k) (|c(k)|) & \text{if } |c(k)| > \lambda \\ 0 & \text{if } |c(k)| \leq \lambda \end{cases} \quad (3.9)$$

where  $\lambda$  is the threshold and the coefficients that are above the threshold are the ones that are retained. The coefficients whose absolute values are lower than the threshold are set to zero.

### Soft Thresholding

Soft thresholding (Figure 3-7(c), for  $\lambda=0.5$ ) deletes the coefficients under the threshold, but modifies the ones that are left. The general soft shrinkage rule is defined by:

$$c_s(k) = \begin{cases} \text{sign} c(k) (|c(k) - \lambda|) & \text{if } |c(k)| > \lambda \\ 0 & \text{if } |c(k)| \leq \lambda \end{cases} \quad (3.10)$$

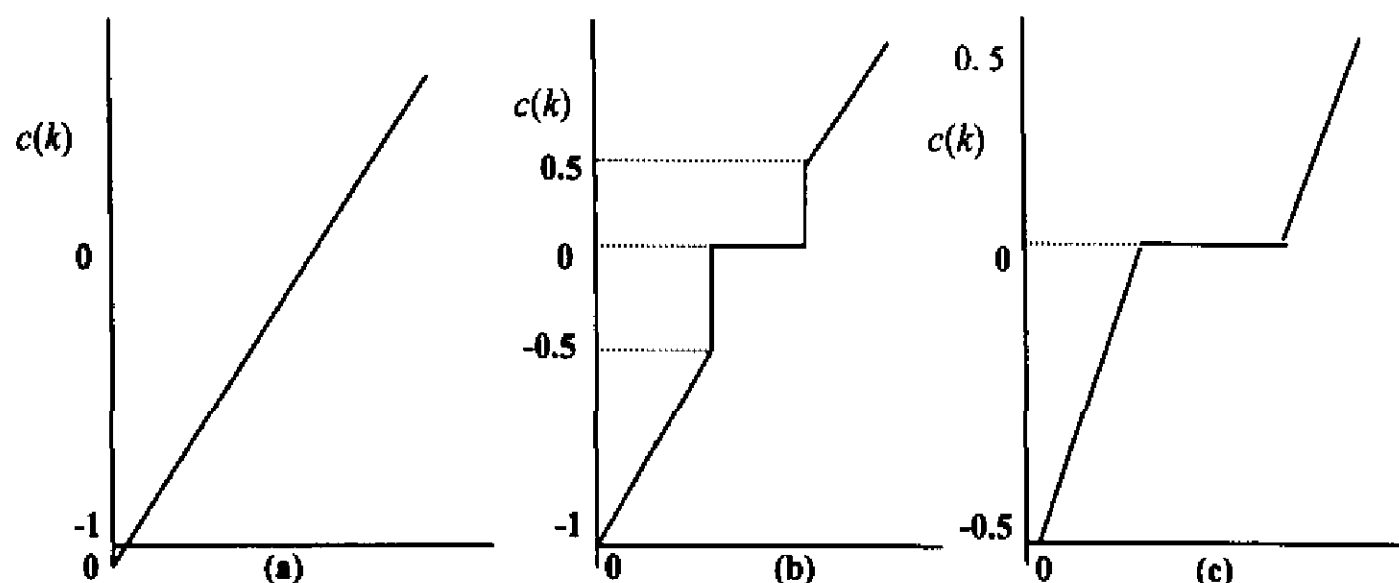


Figure 3-7 An example of (a) linear signal thresholded using, (b) hard-thresholding, and (c) soft-thresholding.

In practice, it can be seen that the soft method is much better and yields more visually pleasant images. This is because the hard method is discontinuous and yields abrupt artifacts in the recovered images.

Now let us discuss the thresholding methods i.e. VisuShrink and BayesShrink in detail. For all these methods the image is first subjected to a discrete wavelet transform, which decomposes the image into various sub-bands. Graphically it can be represented as shown in Figure 3-8.

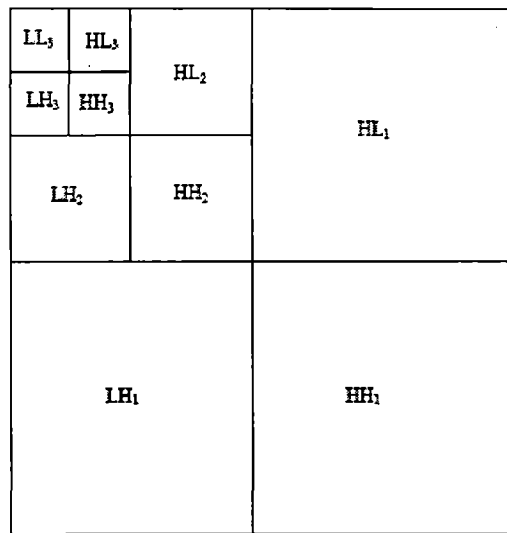


Figure 3-8 DWT of 2D data

The sub-bands  $HH_k, HL_k, LH_k, k = 1, 2, \dots, j$  are called the detail sub bands, where  $k$  is the scale and  $j$  denotes the largest level of decomposition.  $LL_k$  is the low resolution component that gives the approximate coefficients. Thresholding is now applied to the detail components of these sub bands to remove the unwanted coefficients, which contribute to noise. And as a final step in the denoising algorithm, the inverse discrete wavelet transform is applied to build back the modified image from its coefficients.

### 3.5.1 VisuShrink [5]

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:

$$\lambda = \sigma(\sqrt{2 \log(n)}) \tag{3.11}$$

$\sigma^2$  is the noise variance present in the signal and  $n$  represents the signal size or number of samples. An estimate of the noise level  $\sigma$  was defined based on the median absolute deviation [8,10] given by:

$$\hat{\sigma} = \frac{\text{median} (\{|g_{j-1,k}|:k=0,1,\dots,2^{j-1}-1\})}{0.6745} \quad (3.12)$$

VisuShrink does not deal with minimizing the mean squared error [17]. VisuShrink is known to yield recovered images that are overly smoothed. This is because VisuShrink removes too many coefficients. VisuShrink follows the global thresholding scheme where there is a single value of threshold applied globally to all the wavelet coefficients.

### 3.5.2 BayesShrink[5]

BayesShrink was proposed by Chang, Yu and Vetterli [17]. 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. It is a smoothness adaptive shrinkage method. The Bayes threshold,  $t_B$ , is defined as:

$$t_B = \frac{\sigma^2}{\sigma_s} \quad (3.13)$$

where  $\sigma^2$  is the noise variance and  $\sigma_s^2$  is the signal variance without noise. The noise variance  $\sigma^2$  is estimated from the subband  $HH_1$  in Figure 3-9 by the median estimator shown in Equation (3.12). From the definition of additive noise we have

$$w(x,y) = s(x,y) + n(x,y) \quad (3.14)$$

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

$$\sigma_w^2 = \sigma_s^2 + \sigma^2 \quad (3.15)$$

$\sigma_w^2$  can be computed as shown below:

$$\sigma_w^2 = \frac{1}{n^2} \sum_{x,y=1}^n w^2(x,y) \quad (3.16)$$

The variance of the signal,  $\sigma_s^2$  is computed as:



$$\sigma_s = \sqrt{\max(\sigma_w^2 - \sigma^2, 0)} \quad (3.17)$$

With  $\sigma^2$  and  $\sigma_s^2$ , the Bayes threshold is computed from Equation (3.13). Using this threshold, the wavelet coefficients are thresholded at each band shown in Figure 3-8.

## Chapter 4 Hybrid Fractional Fourier and Dual Tree Complex Wavelet Transform for Image Denoising.

In order to understand the hybrid technique for Image denoising we first need to understand both the techniques separately. In this chapter we will first describe how FrFT and DTCWT can be used for Image denoising and then we will proceed to describe the new proposed Hybrid method.

### 4.1 Fractional Fourier Transform

The fractional Fourier transform is a generalization of the Fourier transform, which is introduced by Namias [19] at first and has many applications in signal and image processing spatially in the area of denoising using Low Pass Filter such as an optimal Wiener filter. The fractional Fourier transform can be viewed as the chirp-basis from its definition, but it can also be interpreted as a rotation in the time-frequency plane for better understanding. With the order from 0 increasing to 1, the fractional Fourier transform shows the characteristics of the signal changing from the time domain to the frequency domain.

It has many advantages over conventional Fourier Transform. As in Fourier transform we cannot obtain the local Time Frequency character while FrFT can give information about signal in a domain between time and frequency.

Let us first explain what is meant by Fractional Transform.

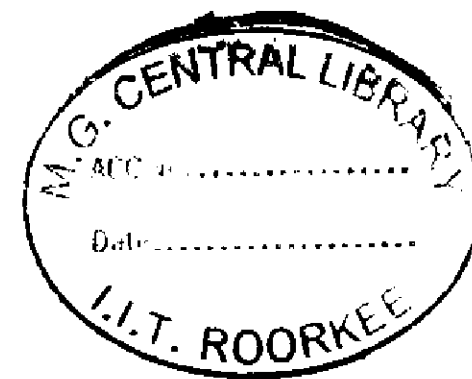
Consider a transform  $T$  on a function  $f(x)$  as

$$T \{f(x)\} = F(u) \quad (4.1)$$

where  $F(u)$  is the  $T$  transform of  $f(x)$ . Now let us define a new transform:

$$T^\alpha \{f(x)\} = F_\alpha(u) \quad (4.2)$$

We call here the “ $\alpha$ -order Fractional  $T$  transform” and the parameter  $\alpha$  is called the “fractional order”. This kind of transform is called “fractional transform”. It is basically operating Transform  $T$  on  $f(x)$  ‘ $\alpha$ ’ number of times where ‘ $\alpha$ ’ can be a fraction as well. This type of Fractional Transform satisfies following constraints:



$$T^0\{f(x)\} = f(x) \quad (4.3)$$

$$T^1\{f(x)\} = F(u) \quad (4.4)$$

It also satisfies the additive property:

$$T^\beta\{T^\alpha\{f(x)\}\} = T^{\beta+\alpha}\{f(x)\} = F_{\beta+\alpha}(u) \quad (4.5)$$

#### 4.1.1 Definition of Fractional Fourier Transform[20,21]

The  $\alpha^{\text{th}}$  order FrFT of a function denoted by  $\{F^\alpha\{f(x)\}\}(x)$  is defined for  $\alpha \in [-2,2]$

$$\{F^\alpha\{f(x)\}\}(x) = \int_{-\infty}^{\infty} B_\alpha(x, x')f(x')dx' \quad (4.6)$$

where the kernel  $B_\alpha(x, x')$  is given by:

$$B_\alpha(x, x') = \frac{\exp\left[-i\left(\frac{\pi\hat{\Phi}}{4} - \frac{\phi}{2}\right)\right]}{|\sin\phi|^{\frac{1}{2}}} X \exp[i\pi(x^2 \cot\phi - 2xx' \operatorname{cosec}\phi + x'^2 \cot\phi)] \quad (4.7)$$

where  $\phi = \alpha\pi/2$  and  $\hat{\Phi} = \operatorname{sgn}(\sin\phi)$ . We see that for  $\alpha=0$  and  $\alpha=2$  kernel reduces to  $B_0(x, x') = \delta(x-x')$  and  $B_2(x, x') = \delta(x+x')$ .

Some of the essential properties of FrFT are:

- FrFT operation is linear
- First order FrFT  $F^1$  corresponds to conventional FT and  $F^0$  means no transform is performed.  $F^2$  gives  $f(-x)$  and  $F^3$  gives Inverse Fourier Transform and  $F^4$  performs no transform.
- FrFT is periodic with  $\alpha$  ranging from  $[-2,2]$  or  $[0,4]$
- FrFT is additive.
- Inverse FrFT of order  $\alpha$  is given by  $F^{-\alpha}$

#### 4.1.2 Denoising using FrFT [22]

In many Image processing applications, signals which we wish to recover are degraded by a noise. We may design some digital filter (explained in Chapter 2 and 3) in Spatial, Frequency

or Wavelet Domain for noise removal. Another approach to denoise an image is designing a filter in Fractional Fourier domain.

The conventional filter in time domain can be written as:

$$x_0(t) = \int_{-\infty}^{\infty} h(t-t')x_i(t')dt' \quad (4.8)$$

where  $x_0(t)$ ,  $x_i(t)$  and  $h(t)$  output signal, input signal and impulse response of the filter respectively. This can also be written in frequency domain as

$$x_0(t) = IFT(FT(x_i(t)).H(w)) \quad (4.9)$$

Where  $H(w)$  is the FT of  $h(t)$

Fractional filter is the generalization of conventional filter and is defined as:

$$x_0(t) = F^{-\alpha}\{F^{\alpha}\{x_i(t)\}.H_{\alpha}(u)\} \quad (4.10)$$

Where  $H_{\alpha}(u) = F^{\alpha}\{h(t)\}$

We can perform the  $\alpha^{th}$  order Fractional Fourier Transform operation that corresponds to rotation of the Wigner Distribution by an angle  $\phi=\alpha\pi/2$  in the clockwise direction, we can find the fractional domain in which signal and noise do not have overlap. Then we can rotate the Wigner Distribution, that is, do the Fractional Fourier Transform in that domain, then filtering out the undesired noise using the LPF. This is shown in Figure 4-1.

In Figure 4-1, we see that on performing filtering operation in spatial or frequency domain the noise cannot be removed completely as noise overlaps with the signal both in frequency and spatial domain.

In Figure 4.1 we can see that if we rotate the Wigner Distribution (WD) in  $0.5^{th}$  domain i.e.  $\alpha=0.5$  noise and signal wont overlap and thus a LPF can be applied to get the noise free signal and then inverse FrFT is applied to get the signal in spatial domain.

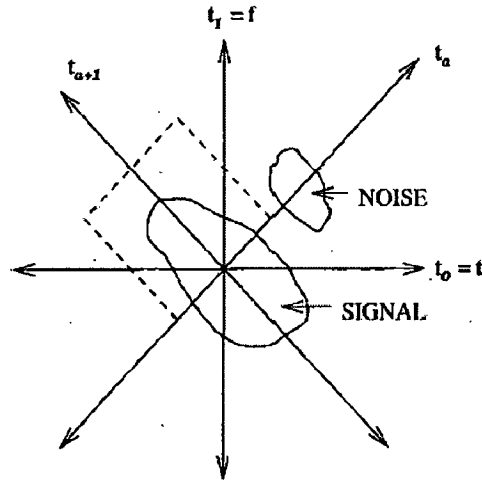


Figure 4-1 Noise Separation in the  $\alpha^{\text{th}}$  domain [20]

## 4.2 Dual Tree Complex Wavelet Transform

The 1-D dual-tree wavelet transform is implemented using a pair of filter banks operating on the same image data simultaneously. The filter bank upper one in Figure 4-2 ('tree a') represents the real part of a complex wavelet transform. The lower one in Figure 4-2 represents the imaginary part ('tree b'). The transform is an expansive (or oversampled) transform. The transform is two times expansive because for an  $N$ -point signal it gives  $2N$  DWT coefficients.

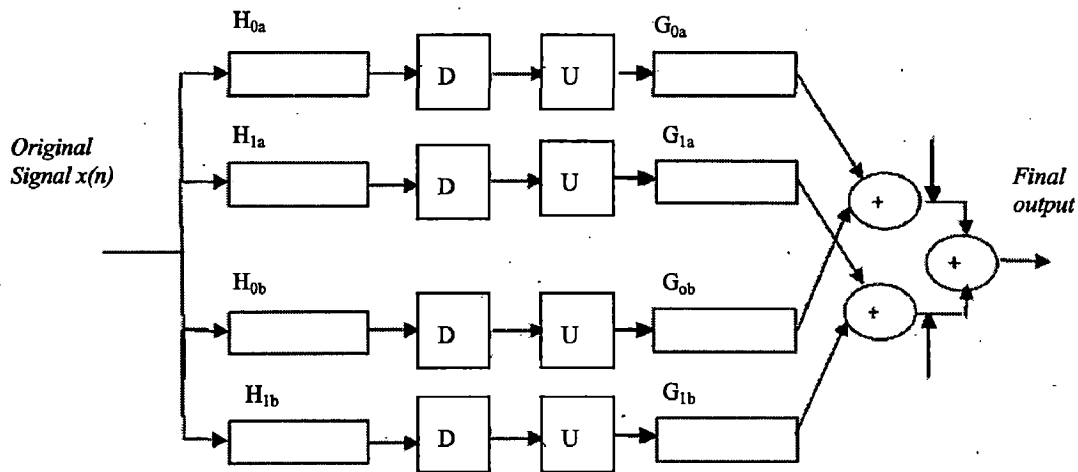


Figure 4-2 One level Complex dual tree.

The approximation and detail coefficients for the two trees are denoted respectively  $(a_A, d_A)$  and  $(a_B, d_B)$ . The detailed coefficients  $d_A$  and  $d_B$  can be interpreted as the real and imaginary parts of a complex process  $z = d_A + id_B$ . DT CWT is not really a complex wavelet transform;

since it does not use any complex wavelet instead it is implemented with real wavelet filters applied in two trees separately. The reconstruction is done in each tree independently, by using the dual filters, the results are averaged to obtain  $x(n)$ . The algorithm for DTCWT is same as for DWT as discussed in Chapter 3. In the implementation of DT CWT in this thesis we use a filter of length 10, the table of coefficients of the analyzing filters in the first stage (Table 4.1) and the remaining levels (Table 4.2) are shown [23]. The coefficients of the synthesis filters are the transposes of the analysis filters (orthogonal filters).

**Table 4-1 First level Coefficients of the analysis filter**

$H_{0a}$	$H_{1a}$	$H_{0b}$	$H_{1b}$
0	0	0.01122678	0
-0.08838834	-0.01122679	0.01122679	0
0.08838834	0.01122679	-0.08838834	-0.08838834
0.69587998	0.08838834	0.08838834	-0.08838834
0.69587998	0.08838834	0.69587998	0.69587998
0.08838834	-0.69587998	0.69587998	-0.69587998
-0.08838834	0.69587998	0.08838834	0.08838834
0.01122679	-0.08838834	-0.08838834	0.08838834
0.01122679	-0.08838834	0	0.01122679
0	0	0	-0.01122679

**Table 4-2 Remaining Levels Coefficients of the Analysis Filters**

$H_{00a}$	$H_{01a}$	$H_{00b}$	$H_{01b}$
0.03516384	0	0	-0.03516384
0	0	0	0
-0.08832942	-0.11430184	-0.11430184	0.08832942
0.23389032	0	0	0.23389032
0.76027237	0.58751830	0.58751830	-0.76027237
0.58751830	-0.76027237	0.76027237	0.58751830
0	0.23389032	0.23389032	0
-0.11430184	0.08832942	-0.08832942	-0.11430184
0	0	0	0
0	-0.03516384	0.03516384	0

Despite the fact that DT CWT is expensive it has many benefits over DWT such as it provides a high degree of shift-invariance and better directionality compared to the real DWT.

### **4.3 Denoising using Hybrid FrFT and DT CWT**

As seen in the preceding subsections about the benefits of using FrFT and DT CWT over conventional methods of Image denoising. There may be a noise model which has both overlapping as well as non overlapping content of noise thus using hybrid model in which filtering in FrFT domain removes overlapping noise while filtering in Wavelet domain removes non overlapping noise can be very effective.

We now propose a new hybrid model which can be implemented via the following steps:

Step1. Take the FrFT of the noisy image.

Step2. Filter out the noise using LPF.

Step3. Take the Inverse FrFT to get the denoised image back in spatial domain.

Step4. Now apply the DTCWT to get detailed and approximate coefficients.

Step5. Choose proper threshold and apply soft thresholding on detail coefficients.

Step6. Now reconstruct the Image using Inverse DTCWT with the modified wavelet coefficients to give the final denoised Image.

- [19] Namias, V., "The fractional order Fourier transform and its application to quantum mechanics", *J. Inst. Math. Appl*, Vol. 25, pp. 241—265, 1980
- [20] H.M. Ozaktas and O. Aytur "fractional fourier domians" *IEEE trans. on signal processing*, vol. 46, pp 119-124, 1995
- [21] A.C. McBride and F.H.kerr "on Namias Fractional Fourier transform" *IMA J.Appl. Math*, vol. 39, pp. 158-175, 1987
- [22] S. R. Subramaniam, T. K. Hon, A. Georgakis, G. Papadakis, "Fractional Fourier-Based Filter for Denoising Elastograms "32nd Annual International Conference of the IEEE EMBS, pp 4028-4031, 2010
- [23] Sathesh ,Samuel Manoharan," A Dual Tree Complex Wavelet Transform Construction and Its Application To Image Denoising," *International Journal of Image Processing (IJIP)* Volume(3), Issue(6), 2010



- [9] N. G. Kingsbury, "The Dual-Tree Complex Wavelet Transform: a new technique for Shift Invariance and Directional filters", *In the Proceedings of the IEEE Digital Signal Processing Workshop*, 1998.
- [10] N. G. Kingsbury, "Image processing with Complex wavelets", *Phil.Trans. Royal Society London- Ser.A*, vol. 357, No. 1760, pp. 2543-2560, Sep 1999.
- [11] Yushu Liu, Mingyan Jiang, "Image denoising algorithm based on DTCWT and adaptive windows," *3<sup>rd</sup> IEEE International Conference on Computer Science and Information Technology, (ICCSIT)*, vol. 8, pp 207-210, 2010
- [12] Hui Wan, Ran Tao, Yue Wang, "Fractional Fourier-Contourlet Deblurring of SpaceVariant Degradation Coupled with Noise," *First International Conference on Pervasive Computing Signal Processing and Applications (PCSPA)*, pp 632-635, 2010.
- [13] Hiroko Furuya, Shintaro Eda, Testuya Shimamura, "Image Restoration via Wiener Filtering in the Frequency Domain", *WSEAS transactions on signal processing*, , Volume 5 Issue 2, Wisconsin, USA, February 2009
- [14] Amara Graps, "An Introduction to Wavelets," *IEEE Computational Science and Engineering*, Vol 2, No. 2., CA, USA, 1995
- [15] Eva Hostalkova, Oldrich Vysata, Ales Prochazka, " Multi-Dimensional Biomedical Image Denoising Using Haar Transform," *15th International Conference on Digital Signal Processing*, pp 175 - 178 , July 2007.
- [16] David L. Donoho and Iain M. Johnstone, "Adapting to Unknown Smoothness via Wavelet Shrinkage," *Journal of American Statistical Association*, 90(432):1200-1224, December 1995.
- [17] S. Grace Chang, Bin Yu and Martin Vetterli, "Adaptive Wavelet Thresholding for Image Denoising and Compression," *IEEE Trans .on Image Processing*, Vol 9, No. 9, pg 1532-1546, Sept 2000.
- [18] D. L. Donoho and I. M. Johnstone, "Denoising by soft thresholding", *IEEE Trans. on Inform. Theory*, Vol. 41, pp. 613-627, 1995.

## Works Cited

[1] Matlab, "Image Processing Toolbox"

<http://www.mathworks.com/access/helpdesk/help/toolbox/images/images.html>

[2] A.Buades, B. Coll and J. M. Morel, "A non-local algorithm for image denoising". *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.2, pp. 60-65, USA, 2005,

[3] S.Suhaila, T.Shimamura," Power estimation method for image denoising by frequency domain Wiener filter," *The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, Vol. 3, pp. 608 - 612 , 2010.

[4] X.Huang, G.A. Woolsey," Image denoising using Wiener filtering and wavelet thresholding" *IEEE International Conference on Multimedia and Expo*, vol 3, pp 1759-1762, 2000.

[5] Mr. Sachin Ruikar, Dr. D.D.Doye," Image Denoising Using Wavelet Transform" *2nd International Conference on Mechanical and Electrical Technology (ICMET)*, pp509-515, 2010.

[6] Amir HajiRassouliha1, Mehdi Amoon1, Mohammad Hossein Doostmohammadi1, Ali Sadr1, Ahmad Ayatollahi1, Gholam Ali Rezairad1," Comparison of Using Different Types of Wavelet or FFT for de-nosing of Body Surface Potential Signals" *Third International Conference on Broadband Communications, Information Technology & Biomedical Applications*, pp233-238, 2008

[7] S. Jiang, X. Hao, "Hybrid Fourier-Wavelet Image denoising," *Electronics Letter, Institution of Engineering and Technology*, Vol 43, No. 20

[8] J. Neumann and G. Steidl, "Dual-Tree Complex Wavelet Transform in the Frequency Domain and an Application to Signal Classification," *International Journal of Wavelets, Multiresolution and Information Processing IJWMIP*, Volume: 3, Issue 1 pp. 43-65, 2005

## **Chapter 6 Conclusion**

In this thesis a new hybrid FrFT-DTCWT technique has been proposed for image denoising. On implementation of various denoising algorithms on several images degraded with different types of noise we conclude that the proposed method gives better result as compared to the earlier denoised algorithms for Gaussian and speckle noise whereas for Salt and Pepper noise median filter remains the best algorithm for denoising. Another conclusion is made that for denoising we need to know the degradation model in order to apply an appropriate algorithm for denoising.

Future work includes to see variation in MSE for different number of levels and to implement various thresholding schemes in DTCWT and to research for modification of the proposed method to denoise SAR images as well.

## 5.2 Discussion

In presence of Gaussian noise we can see that the trend of denoising algorithms remains approximately the same even with the variation of noise present in an image. Out of the various earlier defined algorithms DTCWT and NLM are the best algorithms which give approximately the same result in terms of MSE. However, NLM involves a lot of mathematical computation and is therefore time consuming. Therefore we prefer DTCWT. Also we can see that 'db4' and 'sym4' wavelets gives the best result in VisuShrink and BayesShrink shrinkage methods and BayesShrink outperforms VisuShrink. In VisuShrink we notice that MSE is minimum for Level2, MSE first decreases from level1 to level2 and then keeps on increasing with the increase in number of level of decomposition whereas in BayesShrink method MSE decrease up to a certain value and then become approximately constant with the increase in number of level of decomposition. However denoised images obtained after BayesShrink and VisuShrink are blurry compared to DTCWT and NLM. The new proposed hybrid scheme outperforms all the other algorithms both in terms of MSE as well as visual comparison. We notice that the proposed technique preserves texture and edges of the image which is not the case in DTCWT and NLM where the image becomes smooth and also the MSE calculated with proposed method is quite low.

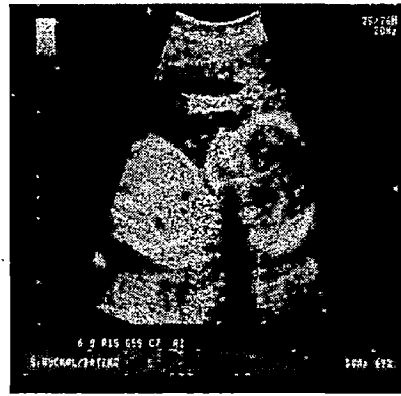
In presence of Salt and Pepper noise, Median filter provides the best result, however proposed method also provides good results when compared to the rest of the other denoising techniques (other than median filter).

In presence of Speckle noise proposed algorithm proves to be the best technique. DTCWT and NLM techniques also give good results. VisuShrink technique follows the same trend as in case of Gaussian noise whereas BayesShrink method is unable to give any satisfactory results.

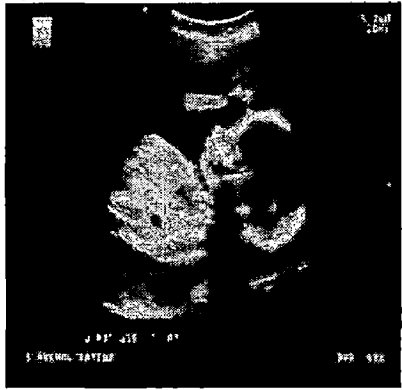
In comparison to DTCWT the proposed algorithmic gives an improvement of '0.5-0.8 db' in presence of Gaussian noise, about '1.5-2.5 db' in presence of Salt and Pepper noise and about '2-3 db' in presence of Speckle noise.



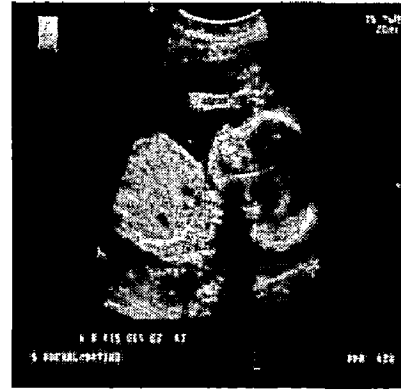
(a)



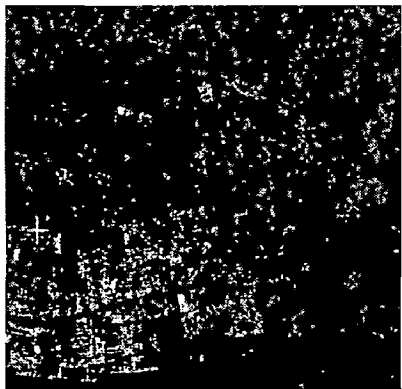
(b)



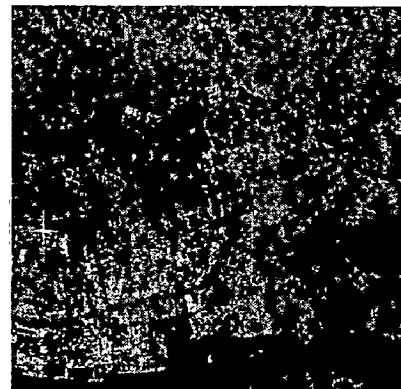
(c)



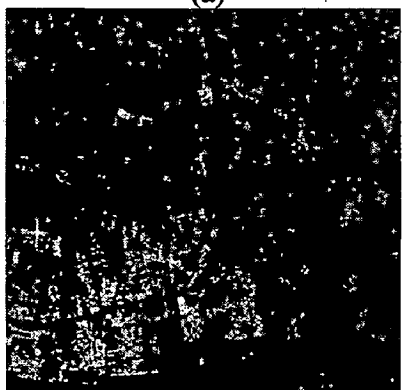
(d)



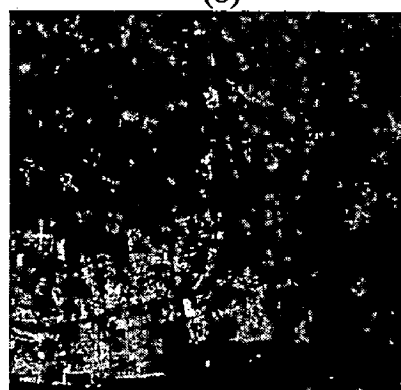
(a)



(b)



(c)



(d)

**Figure 5-8** Ultrasound and SAR Image to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Speckle Noise with Variance =  $1^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid



(a)



(b)



(c)



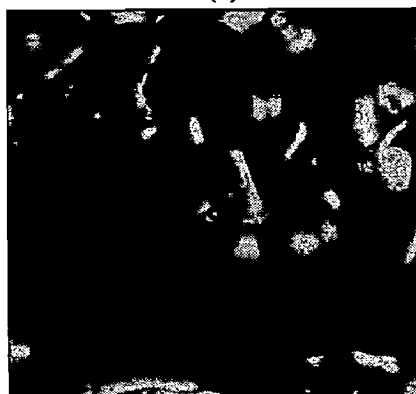
(d)



(a)



(b)



(c)

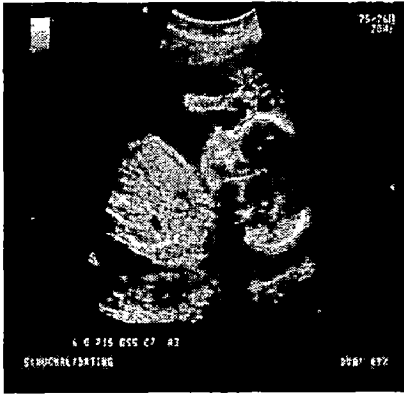


(d)

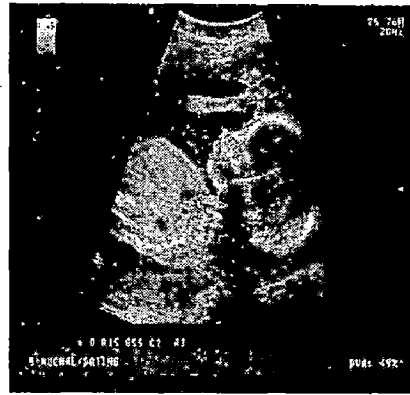
**Figure 5-7 Lena and Peppers Image to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Speckle Noise with Variance =  $1^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid**

Table 5-4 Comparison of various denoising algorithms in presence of Speckle Noise with Noise Variance  $V=1$  for different Images

Algorithms/Images	Lena			Peppers			Ultrasound			SAR		
	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	
Noisy	1467	486.96	472.36	149.72	228.79	330.10	381.91					
Median	425.34	184.55	142.50	184.55	90.62	86.36						
Wiener	376.07	155.07										
NL Means												
DWT	Level1	Level2	Level3	Level1	Level2	Level3	Level1	Level2	Level3	Level1	Level2	Level3
VisuShrink	Haar	415.18	211.76	213.49	155.76	170.81	158.00	123.21	157.15	438.54	607.58	725.71
	Db2	395.42	176.86	171.18	141.10	127.54	143.71	94.86	120.33	383.01	554.67	676.67
	Db4	392.98	167.99	160.58	130.07	106.98	136.98	89.79	112.64	352.62	533.06	655.03
	Sym2	395.42	176.86	171.18	141.10	127.54	143.71	94.86	120.33	383.01	554.67	676.67
	Sym4	391.14	168.30	157.59	132.31	86.69	104.92	139.34	89.50	113.88	352.16	528
BayesShrink	Haar	503.27	312.12	284.49	316.41	286.89	283.69	329.04	326.63	399.79	380.08	378.05
	Db2	468.89	263.85	233.80	308.80	274.79	271.03	344.33	320.94	380.01	360.53	358.56
	Db4	516.02	317.31	286.97	308.33	277.14	273.22	346.80	322.58	380.68	361.58	359.65
	Sym2	468.89	263.85	233.80	308.80	274.79	271.03	344.33	320.94	380.01	360.53	358.56
DTCWT	Sym4	478.40	271.85	240.76	308.36	271.05	343.78	320.08	317.18	372.90	353.61	351.71
		139.71	75.26					78.52				317.15
FrFT		118.99	55.67					58.20				205.33
	Proposed	94.33	36.95					39.37				185.44



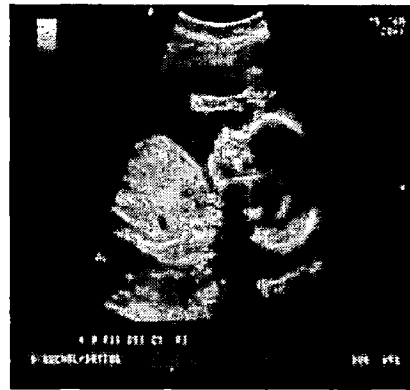
(a)



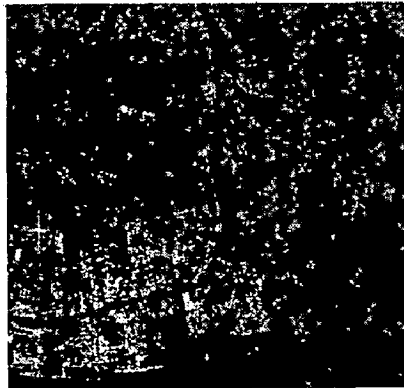
(b)



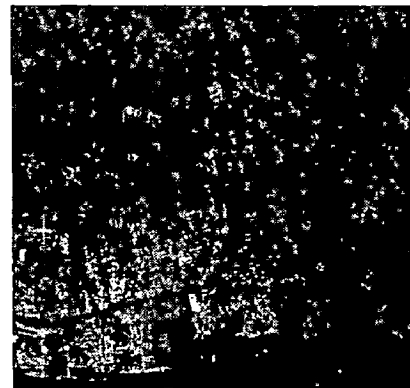
(c)



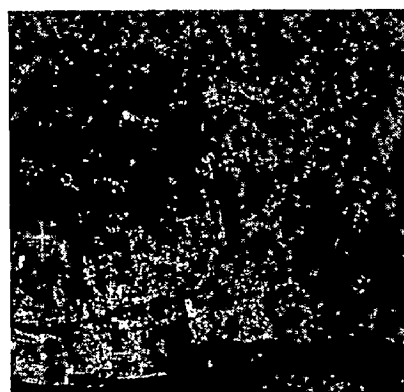
(d)



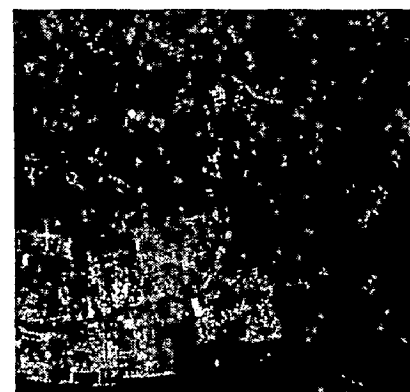
(a)



(b)



(c)



(d)

**Figure 5-6** Ultrasound and SAR Images to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Salt & Pepper Noise with  $D=0.03$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid





(a)



(b)



(c)



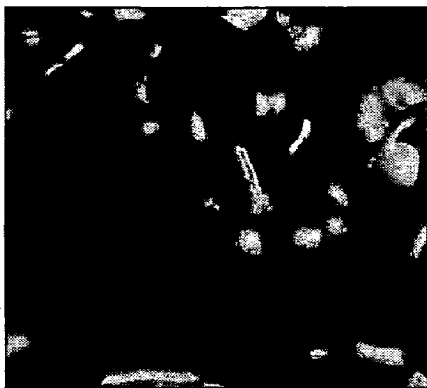
(d)



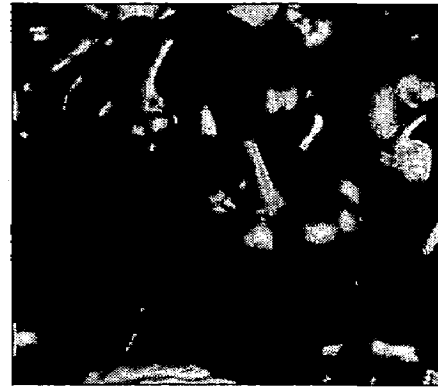
(a)



(b)



(c)



(d)

**Figure 5-5 Lena and Peppers Image to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Salt & Pepper Noise with  $D=0.03$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid**

Table 5-3 Comparison of various denoising algorithms in presence of Salt & Pepper Noise with Noise Density D=0.03 for different Images

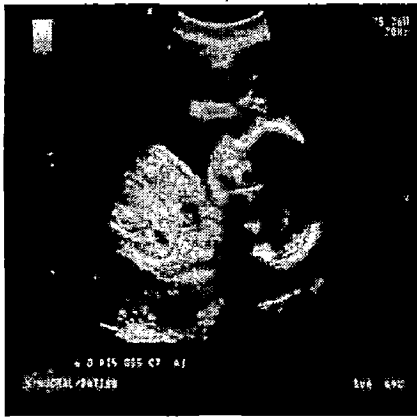
Algorithm/Images	Lena			Peppers			Ultrasound			SAR			
	Level1	Level2	Level3	Level1	Level2	Level3	Level1	Level2	Level3	Level1	Level2	Level3	
	MSE			MSE			MSE			MSE			
Noisy	559.74	559.74	559.74	658.68	658.68	658.68	710.50	710.50	710.50	634.29	634.29	634.29	
Median	21.14	21.14	21.14	7.76	7.76	7.76	24.02	24.02	24.02	248.49	248.49	248.49	
Wiener	368.08	368.08	368.08	474.81	474.81	474.81	522.37	522.37	522.37	514.03	514.03	514.03	
NL Means	196.74	196.74	196.74	298.15	298.15	298.15	374.23	374.23	374.23	463.82	463.82	463.82	
DWT	Level1	Level2	Level3	Level1	Level2	Level3	Level1	Level2	Level3	Level1	Level2	Level3	
VisuShrink	Haar	189.38	158.23	204.8	209.35	147.45	186.09	226.40	153.97	179.26	457.35	632.30	753.98
	Db2	176.41	128.41	162.66	208.10	140.39	154.44	222.23	142.35	158.26	408.01	586.49	714.96
	Db4	165.37	119.29	144.84	183.15	125.56	131.28	209.77	128.04	141.18	364.02	551.20	687.60
	Sym2	176.42	128.41	162.66	208.10	140.39	154.44	223.23	142.35	158.26	408.01	586.49	714.96
BayesShrink	Sym4	166.89	121.71	146.61	193.92	132.96	142.57	216.41	133.12	144.30	370.34	553.48	687.14
	Haar	555.85	554.57	554.37	655.78	655.46	655.41	708.46	707.73	707.59	602.33	595.35	594.49
	Db2	548.08	544.75	544.27	653.86	652.97	652.83	706.03	704.56	704.40	589.60	582.34	581.52
	Db4	539.70	533.75	532.95	649.49	647.29	646.97	699.47	695.96	695.35	583.41	575.43	574.63
DTCWT	Sym2	548.08	544.74	544.26	653.86	652.97	652.83	706.03	704.56	704.40	589.60	582.34	581.52
	Sym4	538.49	532.52	531.67	648.53	646.13	645.73	698.88	695.45	694.80	583.85	576.32	575.45
FrFT	110.93			118			114.92			433			
Proposed	79.32			79.22			73.38			218.54			
	62.36			68.45			65.82			207.33			



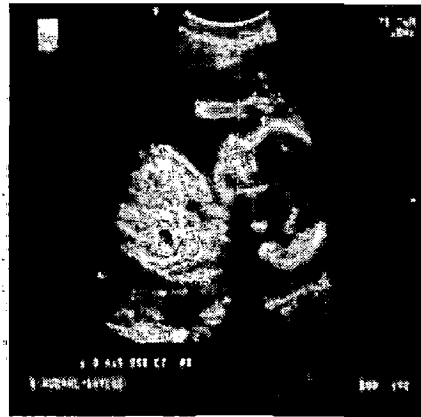
(a)



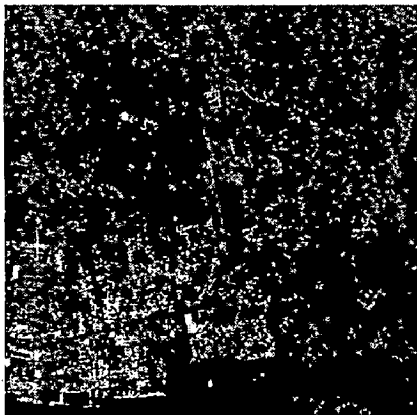
(b)



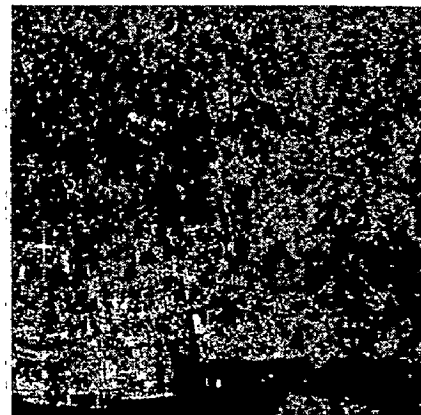
(c)



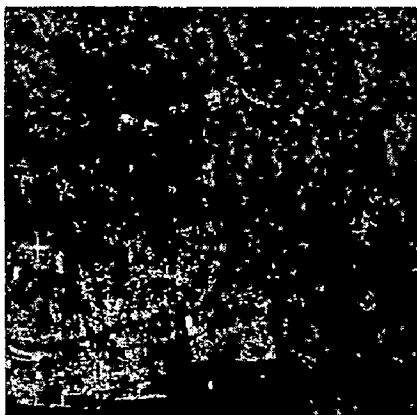
(d)



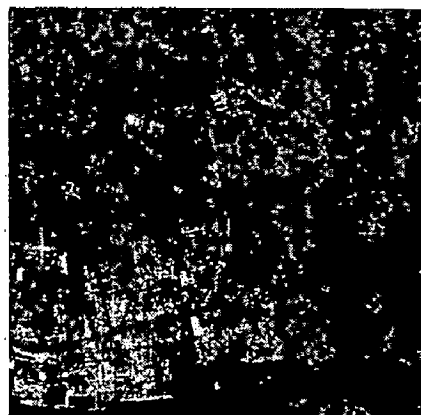
(a)



(b)



(a)



(d)

**Figure 5-4** Ultrasound and SAR Images to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Gaussian Noise with Variance =  $25^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid



(a)



(b)



(c)



(d)



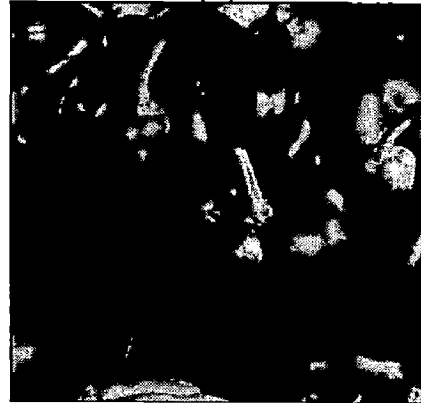
(a)



(b)



(c)

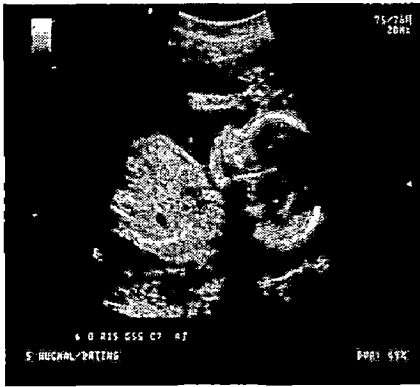


(d)

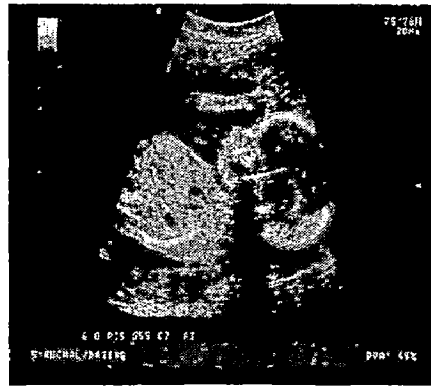
**Figure 5-3** Lena and Peppers Image to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Gaussian Noise with Variance =  $25^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid

Table 5-2 Comparison of various denoising algorithms in presence of Gaussian Noise with  $\sigma=25$  for different Images

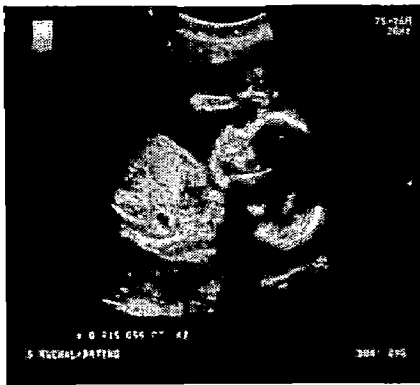
Algorithm/Images	Lena			Peppers			Ultrasound			SAR			
	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE	
Noisy	624.89	623.77	622.35	622.35	623.77	622.35	623.77	622.35	623.77	622.35	623.77	622.35	
Median	134.7648	133.01	117.50	117.50	133.01	117.50	133.01	117.50	133.01	117.50	133.01	117.50	
Wiener	123.83	120.06	116.95	116.95	120.06	116.95	120.06	116.95	120.06	116.95	120.06	116.95	
NL Means	76.59	68.30	57.12	57.12	68.30	57.12	68.30	57.12	68.30	57.12	68.30	57.12	
DWT													
VisuShrink	Level1	200.23	155.41	195.91	189.95	171.60	196.24	134.21	160.39	160.39	171.60	196.24	134.21
	Level2	182.11	118.36	150.64	171.56	126.80	182.21	105.56	123.41	123.41	126.80	182.21	105.56
	Level3	177.84	108.71	138.14	163.65	106.08	176.20	100.20	114.19	114.19	106.08	176.20	100.20
	Level1	182.11	118.36	150.64	171.54	126.72	182.21	105.56	123.41	123.41	126.72	182.21	105.56
	Level2	177.84	108.71	138.14	163.65	106.08	177.32	99.07	116.89	116.89	106.08	177.32	99.07
BayesShrink	Level1	200.23	155.41	107.61	187.39	102.11	193.06	107.03	94.40	94.40	102.11	193.06	107.03
	Level2	182.11	118.37	88.40	174.73	82.61	181.86	90.50	77.54	77.54	82.61	181.86	90.50
	Level3	177.84	108.71	80.48	164.65	72.80	174.16	83.72	70.05	70.05	72.80	174.16	83.72
	Level1	182.11	118.37	88.40	172.20	81.56	181.86	90.50	77.54	77.54	81.56	181.86	90.50
	Level2	177.84	108.71	80.48	164.65	72.80	176.82	85.10	71.49	71.49	72.80	176.82	85.10
DTCWT		72.94			53.95			62.30				243.24	
FrFT		87.25			66.34			78.36				199.48	
Proposed		62.96			47.37			54.30				186.43	



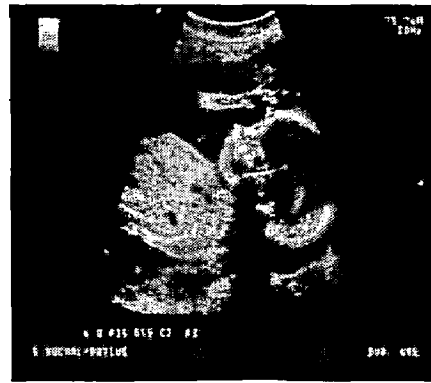
(a)



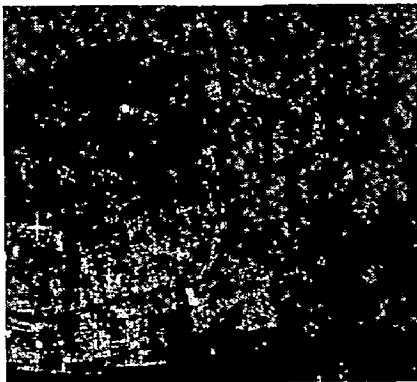
(b)



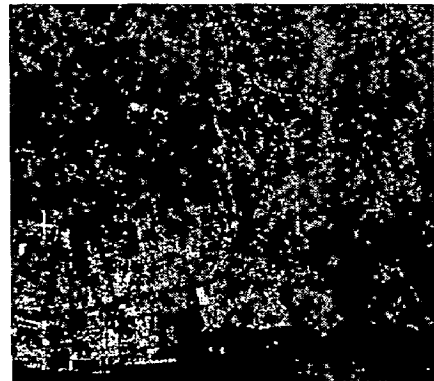
(c)



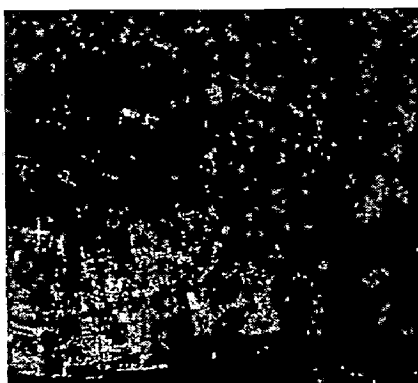
(d)



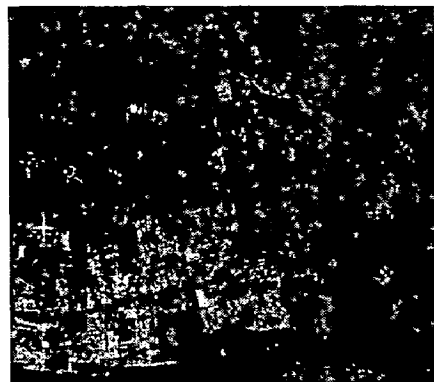
(a)



(b)



(c)



(d)

**Figure 5-2** Ultrasound and SAR Images to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Gaussian Noise with Variance =  $20^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid



(a)



(b)



(c)



(d)



(a)



(b)



(c)



(d)

**Figure 5-1 Lena and Peppers Images to compare denoising by DTCWT and Hybrid FrFT-DTCWT when corrupted with Gaussian Noise with Variance =  $20^2$  (a) Original Image (b) Noisy Image (c) Denoised by DTWCT (d) Denoised by Hybrid**

Table 5-1 Comparison of various denoising algorithms in presence of Gaussian Noise with  $\sigma=20$  for different images

Algorithm/Image	Lena			Peppers			Ultrasound			SAR		
	Level1	Level2	Level3	Level1	Level2	Level3	Level1	Level2	Level3	Level1	Level2	Level3
Noisy		MSE			MSE			MSE			MSE	
Median		398.86			399.21			399.76			399.24	
Wiener		94.59			80.06			94.19			320.41	
NL Means		82.39			75.43			77.47			206.62	
DWT		54.06			46.27			53.75			217.34	
VisuShrink	Haar	145.82	142.86	194.53	134.87	169.46	138.78	121	158.36	392.74	607.11	733.80
	Db2	126.02	105.54	149.04	121.41	83.37	124.76	91.45	120.60	333.60	550.58	684.77
	Db4	121.55	96.02	136.89	109.72	71.46	119.33	85.58	111.90	300.03	527.77	669.20
	Sym2	126.58	104.87	148.49	121.41	83.37	124.15	124.76	120.60	333.60	550.60	684.77
BayesShrink	Sym4	121.37	95.99	135.63	110.99	70.75	119.60	84.75	114.12	300.84	523.48	662.86
	Haar	138.11	92.30	86.40	127.19	77.36	71.54	129.85	80.16	253.56	239.75	238.05
	Db2	124.73	74.65	68.38	118.80	62.70	56.06	121.17	68.30	235.40	221.24	219.81
	Db4	120.74	68.24	61.83	109.70	55.59	49.18	115.98	62.43	223.82	210.13	208.82
DTCWT	Sym2	124.73	74.65	68.38	118.80	62.70	56.06	121.17	68.30	235.41	221.24	219.81
	Sym4	120.77	68.74	62.16	110.32	54.84	47.88	117.41	63.72	223.69	210.17	208.80
FrFT		48.60			42.16			45.99			198.72	
Proposed		59.25			53.46			58.93			149.37	
		43.40			37.12			42.51			139.77	



## Chapter 5 Simulation Results and Discussion

### 5.1 Simulation Results

In this chapter we have taken four different images which are corrupted with Gaussian, Salt & Pepper and Speckle noise separately at different noise levels and then image denoising and restoration algorithms are applied. The simulation results are then being compared on the basis of Mean Square Error (MSE).

$$MSE = \frac{1}{MN} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} (y(m, n) - z(m, n))^2 \quad (5.1)$$

where  $y(m, n)$  and  $z(m, n)$  represent the original image and the de-noised image respectively.  $M$  and  $N$  are the number of rows and columns in image and in our simulation we have used a square image of size 512X512 i.e.  $M=N=512$ . PSNR is then given by:

$$PSNR = 10 \log_{10} \frac{I_{max}^2}{MSE} \quad (5.2)$$

where  $I_{max}$  is the maximum intensity value of image which is 255 in case of 8 bit gray scale image.

We have compared different image denoising techniques by applying them to four different images (Lina, Peppers, an Ultrasound Image and a Satellite Image) in presence of Gaussian noise at two different noise variances ( $20^2$  and  $25^2$ ), Salt and Pepper Noise at a noise density of 0.03 and Speckle noise at a noise variance of 1. The results are compared on basis of MSE and are tabulated in Table 5.1, Table 5.2, Table 5.3 and Table 5.4. The visual comparison is done only for DTCWT and Proposed Hybrid method for all four images.

The discussion on the results obtained is done in the next subsection.