Image interpolation and denoising in discrete wavelet transform domain

Shaojie Wu
University of Nevada, Las Vegas
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
IMAGE INTERPOLATION AND DENOISING IN DISCRETE WAVELET
TRANSFORM DOMAIN

by

Shaojie Wu

Bachelor of Science
Beijing Union University, China
1996

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree
Department of Electrical and Computer Engineering
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
May 2001
The Thesis prepared by

SHAOJIE WU

Entitled

Image Interpolation and Denoising in Discrete Wavelet Transform Domain

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Graduate College Faculty Representative
ABSTRACT

Image Interpolation and Denoising in Discrete Wavelet Transform Domain

by

Shaojie Wu

Dr. Shahram Latifi, Examination Committee Chair
Professor of Electrical & Computer Engineering
University of Nevada, Las Vegas

Traditionally, processing a compressed image requires decompression first. Following the related manipulations, the processed image is compressed again for storage. To reduce the computational complexity and processing time, manipulating images in the transform domain, which is possible, is an efficient solution.

The uniform wavelet thresholding is one of the most widely used methods for image denoising in the Discrete Wavelet Transform (DWT) domain. This method, however, has the drawback of blurring the edges and the textures of an image after denoising. A new algorithm is proposed in this thesis for image denoising in the DWT domain with no blurring effect. This algorithm uses a suite of feature extraction and image segmentation techniques to construct filter masks for denoising. The novelty of the algorithm is that it directly extracts the edges and texture details of an image from the spatial information contained in the LL subband of DWT domain rather than detecting the edges across multiple scales. An added advantage of this method is the substantial reduction in computational complexity. Experimental results indicate that the new algorithm would
yield higher quality images (both qualitatively and quantitatively) than the existing methods.

In this thesis, new algorithm for image interpolation in the DWT domain is also discussed. Being different from other methods for interpolation, which focus on Haar wavelet, new interpolation algorithm also investigates other wavelets, such as Daubecuies and Bior. Experimental results indicate that the new algorithm is superior to the traditional methods by comparing the time complexity and quality of the processed image.
TABLE OF CONTENTS

ABSTRACT ........................................................................................................................iii
LIST OF FIGURES ...........................................................................................................vii
ACKNOWLEDGMENTS ..................................................................................................ix
CHAPTER 1 INTRODUCTION ........................................................................................1
CHAPTER 2 BACKGROUND ..........................................................................................5
  Introduction ................................................................................................................5
  Wavelets and their Characteristics ...........................................................................6
  Discrete Wavelet Transform ...................................................................................10
CHAPTER 3 IMAGE DENOISING IN DWT DOMAIN ..............................................15
  Introduction ..............................................................................................................15
  Denoising Algorithm in DWT Domain .................................................................18
  Edge Detection ........................................................................................................18
  Types of Edge Detection Operators in Spatial Domain .....................................20
    Gradient-Based Methods ................................................................................20
    Laplacian-Based Methods ..............................................................................27
  Edge Detection in Discrete Wavelet transform Domain ..................................30
  Feature Information Projection onto HL, LH and HH Subbands ....................32
  LL Subband Segmentation and Noise Deduction .............................................36
    Introduction to Image Segmentation ..............................................................36
    LL Subband Segmentation .............................................................................39
  Experimental Results ............................................................................................43
  Summary ................................................................................................................47
LIST OF FIGURES

Figure 2.1 The Haar Scaling Function and its Wavelet........................................8
Figure 2.2 Analysis Filter Bank with lowpass and Highpass Filters..................12
Figure 2.3 2-D Wavelet Decomposition.............................................................13
Figure 2.4 2-D Representation of the Wavelet Algorithm with 1 Iterations........14
Figure 2.5 Model for Two-Level Discrete Wavelet Decomposition..................14
Figure 3.1 The Zero Crossing of $f^-(x)$ at $x_p$ creates a phantom edge...........26
Figure 3.2 The original Cameraman and Edges Information using Sobel Operator...26
Figure 3.3 Edges Information Extracted Using Prewitt operator.........................27
Figure 3.4 Experimental Results for Test Image: Gardener..............................33
Figure 3.5 HH, LH and HL subbands after $F_{edge}$ Filtering.............................35
Figure 3.6 Model for Histogram Peaking Finding Segmentation Method..........39
Figure 3.7 Segmentation Result of LL subband of Gardener............................42
Figure 3.8 Experimental results for Test Image: Gardener...............................45
Figure 4.1 Diagram of Linear Interpolation Methods.........................................49
Figure 4.2 M-th Order Image Interpolation.........................................................50
Figure 4.3 Propagation of Extrema Points Across the Scale...............................53
Figure 4.4 Parent-Child Relationship of DWT for Different Scales....................54
Figure 4.5 General Model for Interpolation in DWT Domain for 2-D image........55
Figure 4.6 Original Image: Gardener.................................................................58
Figure 4.7 Interpolating Image Using Bilinear and Db2 Wavelet.......................58
Figure 4.8 Original Image: Cameraman..............................................................58
Figure 4.9 Interpolating Image Using Bilinear and Bior2.4 Wavelet..................58
Figure 4.10 Original Image: Bay Area.................................................................59
Figure 4.11 Interpolating Bay Area Using Bicubic and Db2 Wavelet..................59
Figure 4.12 Original image: Planetary Nubulae.................................................59
ACKNOWLEDGMENTS

I would like to express my debt of gratitude to my advisor. Dr. Shahram Latifi, for his incessant guidance, precious advice and being such an awesome advisor. His patience and clarity of thought are sources of inspiration for me. I would also like to thank Dr. Henry Selvaraj, Dr. Sahjendra Singh, Dr. John Wang and Dr. Emma Regentova for their valuable suggestions and comments.

I would like to extend my thanks and gratitude to all the faculty and staff in the Electrical and Computer Engineering Department for their kind help. I would like to express my gratitude to the Department and Graduate College for providing me with financial support.

I also want to say thanks to my wife Ni Ying, my parents and those who gave me their constant and unconditional support and encouragement throughout the course of my study.
CHAPTER 1

INTRODUCTION

Data compression techniques have been widely used to store and exchange digital information [19,20]. The basic goal of image compression is to convert an original high bit image into a compressed form with a low bit rate. Under the available storage and communication capacity, image compression achieves the best possible quality. The need for data compression is evident. Astronomical images can be extremely large, making the potential gains from image compression very important. For instance, the Space Telescope Science Institute (STScI) has digitized photographic plates covering the entire sky, generating 1500 images each having $14000 \times 14000$ 16-bit pixels. Several astronomical groups are now constructing cameras with mosaics of large CCDs; these instruments will be used in projects that generate data at a rate exceeding 100Mbytes every 5 minutes for many years [13]. Compression of such images can reduce the volume of data that is necessary to store. Even for some “small” images, compression is also necessary. For example, without compression, a letter sized document image sampled at 300 dpi would need about 2 Mbytes of space to store. This is true whether or not the image contains a single line or 100 lines. It would take 2.5 minutes to transmit it over a 56K modem through a telephone line. However, by using an efficient compression technique on the image, only several Kbytes for storage and several seconds for
transmission via the same phone line are needed. Moreover, compressed images have their own advantages over original images.

1. Compression conserves storage space so that larger inventories are allowed.
2. Good compression techniques make progressive transmission possible.
3. Compression techniques make the transmission of the same image with a lower bit rate possible; this means reducing the transmission bandwidth.

It is well known that for image manipulation, the processing speed as well as memory and I/O utilization are major issues to be considered. Because of these reasons above, now most image data are usually archived and transmitted in the compressed domain.

Numerous techniques exist for compressing the image used by digital computers, such as JPEG, quadtrees, weighted finite automata (WFA), wavelets, and iterated function systems[14]. During the past decade, wavelets have made quite a splash in the field of image compression. In fact, the FBI has already adopted a wavelet-based standard for fingerprint image compression[15]. The evolving next-generation image compression standard, JPEG-2000, which will dislodge the current popular JPEG standard, will also be based on wavelets. Wavelets are good tools for decomposing signals, such as images, into a hierarchy of increasing resolutions: as we consider more and more resolution layers, we get a more and more detailed look at the image. Wavelets can be regarded as "mathematical microscopes" that permit one to "zoom in" and "zoom out" of images as multiple resolutions. The remarkable thing about the wavelet decompositions is that it enables this zooming feature at absolutely no cost in terms of redundancy: for an $M \times N$ image, they are exactly $M \times N$ wavelet coefficients—exactly the same as the number of
original image pixels. Because of all these features of wavelets, wavelet transform has been widely used as effective tools in image compression and transmission.

Interpolation is an important application in image processing, computer graphics and computer vision. This technique can be used to smooth the surface of an object, enlarge an image, or recover a signal from the limited samples[28]. There are many image interpolation schemes used in the spatial domain such as Bilinear, nearest neighbor, and bicubic interpolations. Noise is any undesired information that contaminates an image. Noise appears in images from variety of sources, such as digital acquisition process, which converts an optical image into a continuous electrical signal that is then sampled. Digital acquisition is the primary process by which noise appears in digital images[15]. Spatial denoising methods can be effectively used to remove various types of noise in digital images such as noise removal using spatial filters.

Traditionally, to interpolate and denoise a compressed image, we usually need to fully decompress the image, do the related manipulation, then recompress it to save it back into the storage media. Therefore, processing an image in its compressed form is one efficient way to handle image data and save the computer resources as well.

The main objectives of the thesis are to demonstrate image interpolation in the DWT domain, and to develop new methods for an image denoising in the DWT domain. These algorithms can provide fast manipulation methods and good image quality for an image compressed by using DWT based image coding.

The thesis is organized into five chapters. Chapter 1 gives the introduction. Chapter 2 introduces the background and the related research. Chapter 3 presents the image denoising scheme in DWT domain together with experimental results and analysis. Fast
algorithms for image interpolation in the DWT domain are described in Chapter 4. Finally, conclusion and direction for the future work are summarized in Chapter 5.
CHAPTER 2

BACKGROUND

In this chapter, an overview of Wavelet Transform is presented, the concepts of wavelets and its multiresolution analysis will be introduced briefly. The general properties of wavelet system are also listed. In particular, the discrete wavelet transform (DWT), a very useful tool for image processing, is explained.

2.1 Introduction

Wavelets are a family of functions that satisfy certain requirements. The name wavelet should bring to mind a function that waves above and below the $x$-axis and should therefore integrate to zero. Another connotation of wavelet suggests that it should be small, or in other words the function has to be well localized. Other constraints on the functions insure quick and easy calculation of the direct and inverse wavelet transform.

The early work of wavelets can be dated back to the 1980s by Morlet, Grossmann, Meyer and Mallat, their study of wavelet transform was motivated by the fact that some seismic signal could be well presented by translation and dilation of a simple, oscillatory function called wavelet. Later, a wavelet paper written by Ingrid Daubechies in 1988 attracted the attention for its wide applications in many areas such as signal processing, statistics and numerical analysis. As we know, before the introduction of wavelet transforms, the Fourier transform is the most well known, and the most widely used.
transform. Fourier transform was developed by Baptiste Joseph Fourier to explain the distribution of temperature and heat condition. This transform allows for the decomposition of an image into a weighted sum of 2-D sinusoidal terms, which proves extremely valuable for periodic, time-invariant, or stationary phenomena. Hence, the goal of wavelets is to represent the signal based on a set of basis functions. Wavelet can be defined as an oscillating function of time or space and allows simultaneous time and frequency analysis with a flexible mathematical foundation. One important feature distinguishes wavelet transform from Fourier analysis is that it provides localization in both time and frequency and is therefore a tool for the analysis of transient, nonstationary or time-varying signals.

2.2 Wavelets and their Characteristics

Wavelets are based on the concept that already exists in different disciplines. The ‘scale-space’ for multiscale visual modeling has been proposed by computer scientists before the appearance of wavelets. However, wavelets provide a rigorous formulation of these concepts. The most significant advancement made from the traditional scale-space is that multiscale representations are designed to be invertible, compact and computationally efficient. These observations show that wavelet is an evolution of scale-space, but with a solid mathematical foundation.

There is a broad variety of wavelets, such as continuous wavelets, orthogonal wavelets, bi-orthogonal wavelets, etc. Like sines and cosines in Fourier analysis, wavelets are used as basis functions that are localized in time and frequency to represent
other functions. The simplest possible orthogonal wavelet system is the Haar Wavelet. We discuss it as an introductory example. It is shown as follows:

\[ \psi_{0,0}(x) = \begin{cases} 
1 & 0 \leq x < \frac{1}{2} \\
-1 & \frac{1}{2} \leq x < 1 
\end{cases} \quad (2.1) \]

once the mother wavelet \( \psi(x) \) is fixed, a basis can be made of translations and dilations of the mother wavelet:

\[ \psi_{j,k}(x) = \psi_{0,0}(2^j x - k) \quad (2.2) \]

where \( j \) is the dilation function and \( k \) is the translation function, defined as an orthogonal basis in \( L^2(\mathbb{R}) \) (the space of all square integrable functions). Notice that the functions become more and more localized in time as \( j \) increases. This localization action allows us to represent local changes accurately using few coefficients. And the effect of \( k \) is to translate or shift the wavelets. This means that any elements in \( L^2(\mathbb{R}) \) may be represented as a linear combination (possibly infinite) of these basis functions. The orthogonality of \( \psi_{j,k} \) is easy to check. It is apparent that

\[ \int \psi_{j,k}(x) \psi_{j',k'}(x) dx = 0 \quad (2.3) \]

where \( j = j' \) and \( k = k' \) are not satisfied simultaneously. If \( j \neq j' \), then nonzero values of the wavelet \( \psi_{k',j'} \), are contained in the set where the wavelet \( \psi_{k',j} \) is constant. That makes the integral equal to zero.

If \( j = j' \) but \( k \neq k' \), then at least one factor in the product \( \psi_{k,j} \psi_{k',j} \) is zero. Thus the functions \( \psi_{k,j} \) are orthogonal. The constant that makes this orthogonal basis orthonormal is \( 2^{j/2} \). Indeed, from the definition of norm in \( L^2 \):

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
the set \( \{ \psi_{j,k}, j \in \mathbb{Z} \} \) defines an orthonormal basis for \( L^2 \). For the Haar wavelet basis the scaling function is very simple. It is unity on the interval \( [0,1) \), ie.

\[
\phi(x) = 1(0 \leq x \leq 1)
\]

In figure 2.1, we show the Haar wavelet basis and its "scaling function" \( \phi(x) \).

Figure 2.1: the Haar scaling function and its wavelet.

Let \( y = (y_0, y_1, \ldots, y_{2^n-1}) \) be the data vector of size \( 2^n \). The data vector can be associated with a piece-wise constant function of \( f \) on \( [0,1) \) generated by \( y \) as follows:

\[
f(x) = \sum_{k=0}^{2^n-1} y_k 1(2^{-n} \leq x \leq (k+1)2^{-n})
\]

(2.6)

the date function \( f \) is obviously in the \( L^2[0,1) \) space, and the wavelet decomposition of \( f \) has the form

\[
f(x) = c_{0,0}\phi(x) + \sum_{j=0}^{n-1} \sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(x)
\]

(2.7)
the sum respects to \( j \) is finite because \( f \) is a step function, and everything can be exactly described by resolution up to the \((n-1)\)-st level. For each level the sum, which respects to \( k \) is also finite because the domain of \( f \) is finite. In particular, no translation of the scaling function \( j \) is required.

An obvious disadvantage of the Haar wavelet is that it is not continuous and therefore it is not the best choice for representing the smooth functions. This is the reason why other wavelets have been developed.

In brief, the wavelet systems have three general characteristics:

1. They are a two-dimensional expansion set for some class of one or multi-
dimensional signals.

2. The wavelet expansion gives a time-frequency localization of the signal. It means most of the energy of the signal is well presented by a few wavelet coefficients, \( c_{j,k} \).

3. The calculation of the coefficients from the signal can be implemented in efficient ways. Many wavelet transforms can be calculated with \( O(N) \) operations. Generally wavelet transforms require \( O(N \log(N)) \), same as fast Fourier transform (FFT).

4. All the wavelet systems are generated from a signal scaling function or a signal with simple scaling and translation.

5. Almost all useful wavelet systems also satisfy the multiresolution conditions.

6. The lower resolution coefficients can be calculated from the higher resolution coefficients by a tree-structured algorithm called a filter bank, i.e., efficient calculation.
2.3 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is very similar to the Discrete Fourier Transform (DFT), but the DWT does not use the sine and cosine functions to analyze the image data. The discrete wavelet transform is really a family of transforms that satisfy specific conditions. From our perspective, the *Discrete Wavelet Transform* can be described as a transform that are shifted and expanded version of themselves. In other words, the discrete wavelet transform means the signal decomposition with a family of real orthonormal bases generated from the mother wavelet $\psi_{m,n}(x)$ by scaling and translation,

$$\psi_{m,n} = 2^{-m/2} \psi(2^{-m} x - n) \quad (2.8)$$

This allows the analysis of image signal without windowing effects resulting from the need to handle the infinite sine function on finite image signal using the DFT. Where $m,n$ are integers.

Then the signal $f(x)$ can be expended as,

$$f(x) = \sum_{m,n} c_{m,n} \psi_{m,n}(x) \quad (2.9)$$

where wavelet coefficients $c_{m,n}$ can be calculated by inner product,

$$c_{m,n} = \int f(x) \phi_{m,n}(x) \, dx \quad (2.10)$$

To construct the wavelet function $\psi_{m,n}(x)$, a scale function $\phi(t)$ needs to be determined. The $\phi(t)$ can be expressed in terms of a weighted sum of shifted $\phi(2t)$ as

$$\phi(t) = \sum_k h(k) \sqrt{2} \phi(2t - k) \quad (2.11)$$

And the relation between the mother function $\psi(x)$ and $\phi(t)$ is

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\[ \psi(x) = \sum_k g(k) \sqrt{2} \phi(2x - k) \] (2.12)

where \( g(k) \) should satisfy,

\[ g(k) = (-1)^k h(1 - k) \] (2.13)

We can see, the coefficients \( g(k) \) and \( h(k) \) are very important since we need to use them to calculate the wavelet coefficients \( c(k) \) and detailed coefficients \( d(k) \) decomposition.

One of the most common models for a discrete wavelet transform uses the high-pass and low-pass filters. The filters must be the perfect reconstruction filters, which means that any distortion introduced by the forward transform will be canceled in the inverse transform. A discrete two-channel multirate filter bank convolves a signal \( C_{(i)} \) with a low-pass filter \( h[n] \) and a high-pass filter \( g[n] \) and then subsamples the output:

\[ c_{(i+1)} = c_i \ast h(n) \] (2.14)
\[ d_{(i+1)} = d_i \ast g(n) \] (2.15)

Figure 2.2 illustrates the decomposition and reconstruction process, an analysis filter bank with one input \( c_{(i)} \) and two outputs \( c_{(i+1)} \) and \( d_{(i+1)} \) which provide a recursive algorithm for wavelet decomposition through \( h_0(k) \) and \( g(k) \). In the upper path, \( c_{(i)} \) is passed through a low-pass filter \( h_0(k) \) and decimated by a factor of \( M \), the symbol \( \downarrow M \) indicates the operation of down-sampling by \( M \). In the lower path, \( c_{(i)} \) is passed through a high-pass filter \( h(k) \) and also decimated by a factor of \( M \). For the reconstruction process, the input signal \( c_{(i+1)} \) (resp. \( d_{(i+1)} \)) is upsampled by a factor of \( M \).
and filtered by using a low-pass filter $h(n)^*$ (resp. high-pass filter $g(n)^*$). The symbol \( \uparrow L \) indicates the operation of up-sampling by $L$. A reconstructed signal $c_{(i)}$ is obtained by filtering the zero expanded signals with a dual low-pass filter $h(n)^*$ and a dual high-pass filter $g(n)^*$. If $z(x)$ denotes the signal obtained from $x$ by inserting a zero between every sample, the reconstructed original function is obtained by summing the two filtered images. This can be written as:

$$c_{(i)} = z(c_{(i+1)}) \ast h(n)^* + z(d_{(i+1)}) \ast g(n)^*$$  \hspace{1cm} (2.16)
filtering. The output of the analysis system is a set of four subimages [18]: the so called LL (low low), LH (low high), HL (high low) and HH (high high) subbands, which correspond to different spatial frequency bands in the image. The decomposition of "Seagull" into four such subbands is shown in Figure 2.4. We can see that the LL subband is a coarse (low resolution) version of the image, and that the HL, LH, and HH respectively contain details with vertical, horizontal, and diagonal orientations. The total number of pixels on the four subbands is equal to the original number of pixels.

In order to perform the wavelet decomposition of an image, one recursively applies the scheme of Figure 2.3 to the LL subband. Each stage of this recursion produces a coarser version of the image as well as three new detailed images at that particular scale. Figure 2.5 shows the model for two Level discrete wavelet decomposition.

In general, the magnitude of the wavelet coefficient indicates the presence or absence of a particular pattern while its sign indicates the phase of the pattern. The underlying is the same regardless of the small shifts and therefore wavelet analysis has been applied to texture extraction in image processing for more than a decade.

Figure 2.3: 2-D wavelet decomposition

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 2.4: The 2 dimensional representation of the wavelet algorithm with one iteration.

Figure 2.5: Model for two-Level discrete wavelet decomposition
CHAPTER 3

IMAGE DENOISING IN DISCRETE WAVELET TRANSFORM DOMAIN

The uniform wavelet thresholding is one of the most widely used methods for image denoising in the Discrete Wavelet Transform (DWT) domain. The edges and the textures of an image, however, will blur by using this method. It is well known that textures and edges could tolerate some noise but not blurring, whereas smooth regions tolerate blurring but not noise. By implementing the feature extraction, image segmentation technique and projection operation in DWT domain, an algorithm is developed for image denoising.

3.1 Introduction

In many image processing applications, it is often important to filter random noise from an image. Furthermore, the image is commonly stored and transmitted in the compressed form to save memory space and Input/Output bandwidth. Traditionally, denoising a compressed image is a three-step process beginning with decompression. The image is then manipulated and compressed again for storage. To reduce the computational complexity and processing time, denoising a corrupted image in transform domain is an efficient solution.

For Fourier based denoising methods, the standard filter used is the Weiner Filter, which attenuates the high frequency portion of the spectrum; however, this results in
some loss of the image details. Alternatively, the technique of thresholding coefficients in the wavelet transform domain has shown promise. The key concept of wavelet thresholding is that the wavelet representation can separate the signal and the noise, then the DWT compacts the energy of the signal into a small number of DWT coefficients with large amplitudes, and it spreads the energy of the noise over a large number of minute amplitude DWT coefficients. Hence, a thresholding operation attenuates noise energy by removing those small coefficients while maintaining signal energy by keeping those large coefficients unchanged or modified. Based on this idea, Donoho and Johnstone [2-4] developed the method of wavelet shrinkage denoising, which is defined by two thresholding rules[5]. For a given function $p(y)$, the hard thresholding operator is defined as follows:

$$
\hat{P}(y) = \begin{cases} 
    p(y), & |p(y)| > \lambda \\
    0, & \text{otherwise}
\end{cases} 
$$ (3.1)

The soft thresholding consists of replacing each wavelet coefficient $P(y)$ by the value $\hat{P}(y)$ where the next equation is TRUE

$$
\hat{P}(y) = \begin{cases} 
    \text{sgn}(p(y))(|p(y)| - \lambda), & |p(y)| \geq \lambda \\
    0, & \text{otherwise}
\end{cases} 
$$ (3.2)

Uniform wavelet thresholding has shown some success, however, problems arise due to the uniformity of the algorithm. Along with discarding the noise coefficients, uniform wavelet thresholding also attenuates or removes some signal coefficients: a loss in detail results and image clarity is reduced. Most images usually contain three types of information which are classified as smooth regions, texture regions and edges depending on the region type, thus it is advisable to adapt to the changing spatial characteristics.
For example, in a smooth area, noise is more visible, yet in a detailed region such as textures and edges, a small amount of noise is acceptable in order to preserve the image clarity. The threshold should then adapt in a spatial manner.

The wavelet transform breaks an image down into four subsampled, or decimated, images. These results consist of four images. One image that has been highpass filtered in both the horizontal and vertical directions, one that has been highpass filtered in the vertical and lowpass filtered in horizontal, and one that has been lowpassed in the vertical and highpassed in the horizontal, one that has been lowpass filtered in both directions, known as "HH", "LH", "HL", and "LL". Figure 2.5 shows the model for a two-level decomposition, Figure 2.4 illustrates a one level wavelet decomposition of an image. We can see the lowpass-lowpass image in the upper-left corner, the lowpass-highpass images on the diagonals, and the highpass-highpass in the lower-right corner. Further, a second level wavelet transform is shown in the LL corner of Figure 2.5 creating seven subimages. This process is called multiresolution decomposition.

In the resulting images the transform contains spatial information. This differs from the spectrum of the other transforms because visible correlation to the image itself exists.

The basis for our algorithm presented is to use the spatial information contained in the discrete wavelet transform (DWT) domain to manipulate the wavelet coefficients. To accomplish this, the edges of the image are detected in the DWT domain, and the edge information is projected onto each coefficient's subbands. Segmentation methods are utilized to classify the image into different regions. Finally, adapted denoising techniques are applied to each region.
3.2 Denoising Algorithm in DWT Domain

One of the most widely used denoising methods in discrete wavelet transform (DWT) domain is wavelet coefficients thresholding. S. Grace Chang and Martin Vetterli[1] developed an adaptive thresholding method for image denoising which is achieved by soft-thresholding the coefficients with thresholds that are spatially and scale-wise adaptive. Whether the uniform or adaptive thresholding method is used, the signal coefficients are selected based on the amplitude of the coefficients; however, the spatial information contained in DWT domain is not considered. The new algorithm spatially extracts the features from lowpass-lowpass subband of DWT domain, the extracted features are then projected onto HL, LH and HH subbands to select the coefficients attributed to the signal and discard coefficients attributed to noise. The lowpass-lowpass subband is classified into different regions and the adaptive noise filtering methods are used.

3.2.1 Edge Detection

Edge detection is useful for locating the boundaries of objects within an image. Any abrupt change in image frequency over a relatively small area is defined as an edge. Image edges typically occur at the object boundaries, where the amplitude of the object abruptly changes to the amplitude of the background or another object.

Usually, the edges of the image are detected by using edge enhancement filters which are opposite of smoothing filters. Whereas smoothing filters are low pass filters, edge enhancement filters are high pass filters and their effect is to enhance or boost edges. This may mean a simple high pass filter, but sometimes may be more general, including a
thresholding of the points into edge and non-edge categories, and even linking up of edge pixels into connected boundaries in the image [15].

There are various types of operators in use today, many are implemented with convolution masks, and most are based on discrete approximation to differential operators. Differential operations measure the rate of change in the image brightness function. A large change in image brightness over a short distance indicates the presence of an edge. Some edge detection operators return orientation information (information about the direction of the edge), whereas others only return information about the existence of an edge at each point.

Edge detection methods are used as a first step in the line detection process. Edge detection is also used to find complex object boundaries by marking potential edge points corresponding to places in an image where rapid changes in brightness occur. After these edge points have been marked, they can be merged to form lines and object outlines.

With many of these operators, noise in the image can create problems. This is why it is best to preprocess the image to eliminate, or at least minimize noise effects. To deal with the noise effects, we must make tradeoffs between the sensitivity and the accuracy of an edge detector. For example, if the parameter is set so large that the edge detector is very sensitive, it will tend to find any potential edge points that are attributed to noise. If we make it less sensitive, it may miss valid edges. The parameters that we can set include the edge detection mask and the value of the gray-level threshold. A large mask is less sensitive to noise and a lower gray-level threshold will tend to reduce noise effects.

Edge detection operators are based on the idea that edge information in an image is found by looking at the relationship a pixel has with its neighbors. If a pixel’s gray-level
value is similar to those around it, there is probably not an edge at that point. However, if a pixel has neighbors with widely varying gray levels, it may represent an edge point. In other words, an edge is defined by a discontinuity in gray-level values. Ideally, an edge separates two distinct objects. In practice, apparent edges are caused in color of texture or by the specific lighting conditions presented during the image acquisition process.

3.2.2 Types of edge detection operators in spatial domain

There are various types of operators in use today for edge detection, they can be divided into two groups: one group is Laplacian-based methods and the other group is Gradient-based methods. In this chapter, we will discuss different edge detection operators briefly.

3.2.2.1 Gradient-Based Methods

The core of gradient edge detection is the gradient operator, $\nabla$. In continuous form, applied to a continuous-space image, the gradient is defined as:

$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} i_x + \frac{\partial f(x, y)}{\partial y} i_y$$

(3.3)

where $i_x$ and $i_y$ are the unit vectors in the $x$ and $y$ directions. Notice that the gradient is a vector, having both magnitude and direction. Its magnitude, $|\nabla f(x_0, y_0)|$, measures the maximum rate of change in the intensity at the location $(x_0, y_0)$. Its direction is that of the greatest increase in intensity.

To produce an edge detector, we can simply extend the 1-D case. Consider the effect of finding the local extrema of $\nabla f(x, y)$ or the local maxima of

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}$$

(3.4)
the most commonly used method of producing edge segments from the Eq. 3.4 consists of two stages, thresholding and thinning. The gradient magnitude at every point is compared to a threshold value, $T$. All points satisfying the following criterion are classified as candidate edge points:

$$|\nabla f_c(x, y)| \geq T$$

(3.5)

The set of candidate edge points tends to form stripes, which have positive width. Since the desire is usually for zero-width boundary segments, a subsequent processing stage is needed to thin the stripe to the final edge contours. Edge thinning can be accomplished in a number of ways, but thinning by nonmaximum suppression is usually the best choice. Generally, we wish to suppress any point that is not a local maxima in gradient magnitude. Since 1-D local neighborhood search typically produces a single maximum, those points that are local maxima will form edge segments only one point wide. One approach classifies an edge-stripe point as an edge point if its gradient magnitude is a local maximum in at least one direction. However, this thinning method sometimes has the side effect of creating false edges near strong edge lines[21]. It is also somewhat inefficient because of the computation required to check along a number of different directions. A better thinning approach checks only a single direction, the gradient direction, to test whether a given point is a local maximum in gradient magnitude. The points that pass this scrutiny are classified as edge points.

As we discussed earlier, in the continuous-space image, $x$ and $y$ represent the horizontal and vertical axes, respectively. In the discrete space, the representation of $f_c(x, y)$ is $f(n_1, n_2)$, with $n_1$ describing the horizontal position and $n_2$ describing the
vertical. A digital gradient may be computed by convolving two windows with an image. Based on Eq. (3.3), the gradient estimate is

\[ \hat{\nabla} f(n_1, n_2) = f_1(n_1, n_2) + f_2(n_1, n_2) \]

where

\[ f_1(n_1, n_2) = f(n_1, n_2) * h_1(n_1, n_2) \]

\[ f_2(n_1, n_2) = f(n_1, n_2) * h_2(n_1, n_2) \]

The above two masks are necessary because the gradient requires the computation of an orthogonal pair of directional derivatives. The gradient magnitude and direction estimates can be computed as followings:

\[ \left| \hat{\nabla} f(n_1, n_2) \right| = \sqrt{f_1(n_1, n_2)^2 + f_2(n_1, n_2)^2} \]

\[ \angle \hat{\nabla} f(n_1, n_2) = \tan^{-1} \left( \frac{f_2(n_1, n_2)}{f_1(n_1, n_2)} \right) \]

there are many possible derivative-approximation masks for use in gradient estimation. Two simplest approximation schemes for the horizontal derivative are, for the first and central difference, respectively,

\[ f_1(n_1, n_2) = f(n_1, n_2 + 1) - f(n_1, n_2 - 1) \]

\[ f_2(n_1, n_2) = f(n_1 + 1, n_2) - f(n_1 - 1, n_2) \]

the masks

\[ h_1(n_1, n_2) = -1 \quad 0 \quad 1 \quad \text{and} \quad h_2(n_1, n_2) = 0 \]

\[ 1 \]
are used. Based on the gradient operation, other operators are developed, which are described below:

- Sobel Operator

The Sobel edge detection masks look for edges in both the horizontal and vertical directions and then combine this information into a single metric. These masks are as follows:

\[
\begin{align*}
\text{ROW MASK} & \quad \text{COLUMN MASK} \\
\begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1
\end{bmatrix} & \quad \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{bmatrix} \tag{3.14}
\end{align*}
\]

These masks are each convolved with the image at each pixel location and now we have two numbers: \( s_1 \), corresponding to the result from the row mask, and \( s_2 \), from the column mask. We use these numbers to compute two metrics, the edge magnitude and the edge detection, which are defined as follows:

**EDGE MAGNITUDE**

\[
\sqrt{s_1^2 + s_2^2} \tag{3.15}
\]

**EDGE DETECTION**

\[
\tan^{-1}\left[ \frac{s_1}{s_2} \right] \tag{3.16}
\]

The edge detection is perpendicular to the edge itself because the direction specified is the direction of the gradient, along which the gray levels are changing. It is well known that moving across an edge, the gradient will start at zero, increase to a maximum, and then decrease back to zero, by manipulating the edge magnitude and edge detection for every pixel, the edges of an image are found.
Figure 3.2 illustrates an original image and the edge information using Sobel operator.

- Kirsch Compass Masks

The Kirsch edge detection is defined by taking a single mask and rotating it to the eight major compass orientations: North, Northwest, West, East, Northeast, Southwest, South and Southeast, so it is called Compass masks. The masks are defined as follows:

\[
\begin{align*}
    k_0 &= \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}, \\
    k_1 &= \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}, \\
    k_2 &= \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, \\
    k_3 &= \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
    k_4 &= \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}, \\
    k_5 &= \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}, \\
    k_6 &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}, \\
    k_7 &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}
\end{align*}
\]

The edge magnitude is defined as the maximum value found by the convolution of each of the masks with the image and the edge direction is defined by the mask that produces the maximum magnitude; for instance, \(k_0\) corresponds to a vertical edge, where \(k_5\) corresponds to a diagonal edge in the Northwest/Southwest direction. The edges of the image for every direction are obtained by combing every convolution result of eight masks.

- Prewitt Operator

The Prewitt is similar to the Sobel, but with different mask coefficients. The masks are defined as follows:
These masks are each convolved with the image. At each pixel location we find two numbers: $p_1$, corresponding to the result from row mask, and $p_2$, from the column mask. We use these results to determine two metrics, the edge magnitude and the edge direction, which are defined as follows:

**EDGE MAGNITUDE**

$$\sqrt{p_1^2 + p_2^2}$$  \hspace{1cm} (3.17)

**EDGE DETECTION**

$$\tan^{-1}\left[\frac{p_1}{p_2}\right]$$  \hspace{1cm} (3.18)

as with the Sobel edge detection, the direction lies $90^\circ$ from the apparent direction of the edge. The result for edge extraction from the same image using Prewitt operator is illustrated in figure 3.3.
Figure 3.1 The zero crossing of $f'(x)$ at $x_p$ creates a phantom edge.

Figure 3.2 The original image (left) and edges information using Sobel operator (right).
3.2.2.2 Laplacian-Based methods

In the previous section, we discussed the Gradient-based edge detection methods, in this chapter, we will introduce the Laplacian-based methods.

For a continuous function $f(x, y)$ of two variables $x$ and $y$, the laplacian is defined as:

$$\nabla^2 f(x, y) = \nabla \cdot \nabla f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$  \hspace{1cm} (3.19)

The gradient operator applied to a continues function produces a vector at each point whose direction gives the direction of maximum change of the function at that point, and whose magnitude gives the magnitude of this maximum change [19]. The zero crossings of $\nabla^2 f(x, y)$ occur at the edge points of $f_c(x, y)$ because of the second derivative action. Laplacian-based edge detection has the nice property that it produces edges of zero

Figure 3.3 Edges information extracted using Prewitt Operator
thickness, making edge-thinning steps unnecessary. This is because the zero crossings themselves define the edge locations.

The continues laplacian is isotropic, favoring no particular edge orientation. Consequently, its second partial terms in Eq. (3.3) can be oriented in any direction as long as they remain perpendicular to each other. Consider an ideal, straight, and noise-free edge oriented in an arbitrary direction. Let us realign the first term of Eq. (3.3) parallel to that edge and the second term perpendicular to it. The first term then generate no response at all because it acts only along the edge. The second term produces a zero crossing at the edge position along its edge-crossing profile.

An edge detector based solely in the zero crossing of the continues laplacian produces closed edge contours if the image, \( f(x,y) \), meets certain smoothness constraints \([20]\). The contours are closed because edge strength is not considered, so even the slightest, most gradual intensity transition produces a zero crossing. In effect, the zero-crossing contours define the boundaries that separate regions of nearly constant intensity in the original image. The second derivative zero crossings occur at the local extrema of the first derivate\([29,30]\), but many zero crossings are not local maxima of the gradient magnitude. Some local minima of the gradient magnitude give rise to phantom edges, which can be largely eliminated by appropriately thresholding the edge strength. In Figure 3.1, a 1-D example of a phantom edge is illustrated.

For the digital signal, the laplacian operator, which in one dimension reduced to the second derivative, is also computed by convolving a mask with the image. One of the masks that is used may be derived by comparing the continuous and digital cases as follows:
\[ v^*(i) = v(i) - v(i-1) = [v(i) - v(i-1)] - [v(i-1) - v(i-2)] \]  
\[ = v(i-2) - 2v(i-1) - v(i) \]  
\[ = (1 - 2 1)(v(i-2) v(i-1) v(i)) \]  

In this form, the Laplacian at \( i \) is computed from values centered about \( i-1 \). To keep the Laplacian symmetric, it is normally shifted and given at \( i \) as:

\[ (1 -2 1) (v(i-1) v(i) v(i+1)) \]  
also, the sign is typically changed to give:

\[ (-1 2 -1) (v(i-1) v(i) v(i+1)) \]  
and this is a common form of the one-dimensional digital laplacian although mathematically it is the negative of the Laplacian. Different choices are available when extending this mask to two dimensions. Two standard masks are:

\[ \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \]  

- **Laplacian Operators**

  A classical laplacian-based method is Laplacian operator, unlike Kirsch Compass Masks, the laplacian masks are rotationally symmetric, Which means edges at all orientations contribute to the result. They are applied by selecting one mask and convolving it with the image. The sign of the result (positive or negative) from two adjacent pixel locations provides directional information, and tells us which side of the edge is brighter, the Laplacian masks are defined as:
Laplacian Masks

\[
\begin{bmatrix}
0 & -1 & 0 \\
-1 & 4 & -1 \\
0 & -1 & 0
\end{bmatrix}
\quad \begin{bmatrix}
1 & -2 & 1 \\
-2 & 4 & -2 \\
1 & -2 & 1
\end{bmatrix}
\quad \begin{bmatrix}
-1 & -1 & -1 \\
-1 & 8 & -1 \\
-1 & -1 & -1
\end{bmatrix}
\quad (3.24)
\]

These masks differ from the laplacian type previous described is that the center coefficients have been decreased by one. With these masks we are trying to find edges and are not interested in the image itself. An easy way to picture the difference is to consider the effect each mask would return when applied to an area of constant value. The preceding convolution masks would return a value of zero. If we increase the center coefficients by one, each would return the original gray level. Therefore, if we are only interested in edge information, the sum of the coefficients should be zero. If we want to retain most of the information that is in the original image, the coefficients should sum to a number greater than zero. The larger this sum, the less the processed image is changed from the original image. Consider an extreme example in which the center coefficient is very large compared with the other coefficients in the mask. The resulting pixel value will depend most heavily upon the current value, with only minimal contribution from the surrounding pixel values.

3.2.3 Edge Detection in Discrete Wavelet Transform Domain

The significant of detecting edges in the DWT domain for denoising is obvious, once we distinguish edges from other regions of an image, we could preserve them and add them back after coefficients thresholding. Detecting edges directly on the wavelet transform data usually uses very complicated algorithms\cite{6,7,8}, Xu, el\cite{9} use the direct multiplication of wavelet transform data at adjacent scales to distinguish important edges from noise. This technique is based on the fact that sharp edges have large signal
amplitudes over many wavelet scales, and noise dies out swiftly with increasing scale. S. Mallat developed the method of multiscale edge detection. The main idea is that the nonsampled wavelet decomposition implements the discretized gradient of the image smoothed at different scales. The points of sharp variances occur at local maxima (called modulus maxima) of the gradient norm in the direction of gradient. The modulus maxima which propagates across scales could be characterized by the local Lipschitz regularity. If the function $f$ is Lipschitz $\alpha$ at point $(x_0, y_0)$, then for $(x, y)$ in its neighborhood,

$$M_s f(x, y) \leq K s^\alpha$$ (3.25)

where $M_s f$ denotes the modulus of the wavelet transform at scale $s$, and $K$ is a constant. Since the noise sequence is singular almost everywhere and its induced modulus maxima does not show coherence across the scales, this property allows one to distinguish the edge’s points from noise points. This is accomplished by associating local maxima points across the scale. Hence, the chain of modulus maxima is the edge information. These methods need to compute the wavelet coefficients across many scales, to reduce the computational complexity and save the computational efforts, the edge detection method presented in this thesis only uses the lowpass-lowpass subband of wavelet decomposition for the first level. From Figure 1.1, we can see that the LL subband contains spatial information, rather than detecting edges directly on HH, LH and HH subbands with complicate algorithms. This property provides the motivation and possibility to use the spatial technique on LL subband in DWT domain.

The Sobel boundary detection algorithm is chosen as the edge detection method[15]. According to our experiments, it has less sensitivity to noise and less computation
efforts, the main idea of this boundary detection algorithm is briefly summarized in the previous section, and the reader could refer to [10, 11] for more details. Figure 3.4 (a) is used as the test image, with additive $iid \ N(0, \sigma)$ noise. One level wavelet decomposition is shown in figure 3.4(b). The edges information of LL subband after Sobel detection is illustrated in Figures 3.5(a,b). The original image is very noisy; however, the result of the edge detection on the LL subband is very clear. This is caused by two reasons: First, the Sobel boundary detection algorithm has less sensitivity to noise. Second, in DWT, LL subband is achieved by convolving the lowpass filter with both rows and columns of the image. Since most noise is a high frequency signal which could be filtered out by lowpass filters, using a lowpass filter yields a relatively “noise free” LL subband.

3.2.4. Feature information projection onto HL, LH and HH coefficients Set

After detecting the edge regions in this algorithm, the information needs to be projected onto the remaining three subbands. It is well known the wavelet decomposition is applied separably in the horizontal and vertical directions. Therefore for a particular image, it translates to a low-frequency approximation of the original image, primarily horizontal edges, primarily vertical edges, and diagonal edges. Figure 2.5 is the two-level decomposition of a test image. We can see LH subband only contains vertical edges, HL only contains horizontal edges, and HH contains only edges in the diagonal direction. But if an image is corrupted by noise, HH, LH and HL subbands contain both edge information and noise coefficients. This procedure will discard the coefficients attributed to noise and preserve the signal energy by keeping signal coefficients constant.

The basic procedure for the proposed algorithm consists of five steps:
Figure 3.4(a): Test image, with additive $iid \ N(\mu, \sigma)$ noise

Figure 3.4(b): One level decomposition of the test image

Figure 3.5(a): The LL subband after One level decomposition of the noisy Image

Figure 3.5(b): edge information of LL subband after Sobel edge detection
1. Image decomposition: The noisy image is decomposed by DWT for the first level using Haar basis vectors.

2. Feature extraction: The Sobel boundary detection algorithm is used on LL subband to find the edges. The result is saved in the matrix $M_{edge}$ and has the same size as LL subband.

3. After determining the edge regions, a filter mask $F_{edge}$ is constructed based on the value of $M_{edge}$, for each position $(x_0, y_0)$ in $M_{edge}$. If $M_{edge}(x_0, y_0)$ is not equal to zero, then the corresponding position in $F_{edge}(x_0, y_0)$ is set to one; otherwise, it is set to zero.

4. Considering different properties of wavelet bases, it is possible to have 1-2 positions shifting between LL and the other three subbands for the same decomposition scale. An operation called erosion[32] is used on the filter mask $F_{edge}$. The dilation rule is to add (set to On) any background pixel which touches other pixels that are already part of a region. In our algorithms, we look for an On pixel $(F_{edge}(x, y)=1)$ and turns on all of its immediate neighbors.

5. HL, LH and HH subbands are filtered by the filter mask $F_{edge}$, this procedure discards most of the noise coefficients and preserves the coefficients for the edge regions to maintain the signal energy.

The result of HL, LH and HH subbands after the $F_{edge}$ filtering is shown in Figure 3.5. Compared with the HL, LH and HH subbands before filtering, most of the noise coefficients are filtered out while the edge coefficients are preserved.
Figure 3.5 (a): HL subband after $F_{edge}$ Filtering

Figure 3.5(b): LH subband after $F_{edge}$ Filtering

Figure 3.5(c): HH subband after $F_{edge}$ filtering

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.2.5 LL Subband Segmentation and Noise Deduction

3.2.5.1 Introduction to image segmentation

Image segmentation is important in many image processing applications. The goal of image segmentation is to find regions that represent objects or meaningful parts of objects. Image segmentation methods look for objects that either have some measure of homogeneity within themselves or have some measure of contrast with the objects on their border. Most image segmentation algorithms are modifications, extensions or combinations of these two basic concepts [31]. The homogeneity and contrast measure can include features such as gray level, color and texture. After some preliminary segmentation is performed, we may incorporate higher-level object properties, such as perimeter and shape, into the segmentation process.

The established segmentation algorithms for images can be generally grouped into two classes: region growing and shrinking, clustering techniques and histogram peak finding.

- Region growing and shrinking methods segment the image into regions by operating principally in the $rc$-based image space. Some of the techniques used are local, in which small areas of the image are processed at a time; others are global, which the entire image is considered during the processing. Methods that can Combine local and global techniques, such as split and merge, are referred to as state space techniques and use graph structures to represent the regions and their boundaries.

Various split and merge algorithms have been developed, but they all are most effective when heuristics applied to the domain under consideration can be
applied. This gives a starting point for the initial split. In general, the split and merge technique proceeds as followings:

1. Define a homogeneity test. This test defines the homogeneity measure, which may incorporate brightness, color, texture or other application-specific information, and determines a criterion the region must meet to pass the homogeneity test.

2. Split the image into equally sized regions and calculate the homogeneity measure for each region. If the homogeneity test is passed for a region, then a merge is attempted with its neighbors. If the criterion is split.

3. Continue this process until all regions pass the homogeneity test. The user-defined homogeneity test is largely application dependent, but the general idea is to look for features that will be similar within an object and different from the surrounding objects. For example, we could use the gray-level variance as the homogeneity and define a homogeneity test that requires to be less than some threshold. The gray-level variance is defined as:

\[
\frac{1}{N-1} \sum_{(r,c) \in \text{region}} \left[ I(r,c) - \bar{I} \right]^2
\]  

(3.26)

\[
\bar{I} = \frac{1}{N} \sum_{(r,c) \in \text{region}} I(r,c)
\]  

(3.27)

- Clustering Techniques

Clustering techniques are image segmentation methods by which individual elements are placed into groups; these groups are based on some measure of similarity within the group. The major difference between these techniques and the region growing techniques is that domain other than the \( rc \)-based image space.
may be considered as the primary domain for clustering. Some of these other domains include color spaces, histogram spaces, or complex feature spaces.

The simplest method is to divide the space of interest into regions by selecting the center or median along each dimension and splitting it there; this can be done iteratively until the space is divided into the specific number of regions needed.

Recursive region splitting is a clustering method that has become a standard technique. This method uses a thresholding histograms technique to segment the image. A set of histograms is calculated for a specific set of features, and then each of these histograms is searched for distinct peaks. This process is illustrated in Figure 3.6. The best peak is selected and the image is split into regions based on this thresholding of the histogram. One of the algorithms based on these concepts is as follows:

1. Consider the entire image as one region and compute histograms for each component of interest.
2. Apply a peak finding test to each histogram. Select the best peak and put thresholding on either side of the peak, segment the image into two regions based on this peak.
3. Smooth the binary thresholded image so that a single connected sub region is left.
4. Repeat steps 1-3 for each region until no new sub regions can be created, that is, no histograms have significant peaks.
Figure 3.6: two threshold are selected, one on each side of the best peak. The image is then split into two regions. Region 1 corresponds to those pixels with feature value between the selected thresholds. Known as those in the peak. Region 2 consists of those pixels with feature values outside the threshold.

3.2.5.2 LL subband segmentation

As we discussed in previous section, most of the noise coefficients of HH, LH and HL subbands are removed by $F_{edge}$ filtering. It is now necessary to denoise LL subband. Typical denoising methods are based on thresholding of the wavelet coefficients, most of the literature thus far has concentrated on developing threshold selection methods, very little has been done with the consideration that noise has varied visibility in different regions. The spatial information contained in LL subbands provides the motivation and possibility to develop a denoising method in a spatial manner.

As previously discussed, a drawback of uniform thresholding is the loss in image clarity as a result of noise removal, after extracting the edge coefficients from HL, LH
and HH subbands, the LL subbands can be classified as texture or smooth regions. S., Grace Chang and Martin Vetterli[1] segment the image in spatial domain by calculating a variance image, which has shown some success. In DWT domain, although LL subband still contains spatial information after Discrete Wavelet transform, the variance of these coefficients is not as intense as those in the spatial domain, only considering the variance is not sufficient; therefore, the standard derivation is also used as a feature to classify the LL subband.

In practice, features are calculated from small regions, which are obtained by sliding a window of size \( M \times M \). The following two structural features are considered for classification:

1. Standard Derivation of the region (M, M): each pixel of standard derivation[32] image is computed with the following equation:

\[
V_m(x, y) = \sqrt{\frac{1}{M^2 - 1} \sum_{i=1}^{M} \sum_{j=1}^{M} (C(i, j) - m)^2}
\]

(3.28)

Where, \( m \) is the mean feature of the region (M, M) defined:

\[
m_{r,c} = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} C(i, j)
\]

(3.29)

\( C(i, j) \) is the wavelet coefficients of the LL subband at the location \( (i, j) \). Usually, for the smooth regions, the value of this feature tends to be zero while for the text texture, the value is much larger due to the fact that many abrupt changes exist.

2. Image Variance: each pixel of the variance image[1] is defined as:
\[ V_r(x,y) = \sqrt{\frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} (C(x,y) - C(i,j))^2} \]  

(3.30)

\( C(i,j) \) are the wavelet coefficients of LL subband around pixel location \((x,y)\) in the window.

Since both features should be considered, we take the pixel value of \( V_r \) and \( V_m \) at the location \((m,n)\) as a vector of size 2 and Euclidean distance \( E_d(i,j) \), is computed between \((V_m(m,n), V_r(m,n))\) and \((0,0)\), it is defined as:

\[ E_d(i,j) = \sqrt{V_r(i,j)^2 + V_m(i,j)^2} \]  

(3.31)

After computing all the corresponding pixel pairs, we get the Euclidean distance image \( E_d \), the segmentation is done by thresholding the \( E_d \) image between certain amplitudes[12]. The threshold \( \lambda \) is found empirically. The pixels \( E_d(i,j) \) satisfying the following condition is labeled texture: \( E_d(i,j) \geq \lambda \), and the value of these pixels is then set to one. The remaining pixels are classified as smooth pixels and set to zero. Because of the existence of noise in the image, this approach is sometimes prone to error[1]. For example, among a patch of predominantly smooth regions, there are many holes with texture pixels. These holes are closed by combinations of binary image processing operations called Erosion and Dilation[12], which belong to the class of morphological operations. The classical erosion is to remove (set to OFF) any pixel touching another pixel that is part of the smooth regions (is already OFF). On the contrary, instead of removing pixels from features, dilation adds (set to ON) any smooth pixel, which touches at least one pixel that is already part of the region. Therefore the combination of
a dilation followed by an erosion is called closing, which removes small patches of OFF pixels and small patches of On pixels could be removed by erosion followed by dilation (called opening), erosion and dilation can also add or remove a layer of pixels around the periphery of all features and regions. In our experiments, the combination of erosion and dilation is used to close some bigger holes. Figure 3.7 is the classification map, which shows the result of LL subband segmentation.

Figure 3.7: the result of LL subband segmentation, the white is a smooth region, the black is an edge and the gray is texture.

After classifying the smooth and texture regions of LL subband, the pixels that belong to texture regions are represented as one in $E_d$, and the zero pixels represent the smooth regions. Hence, we use $E_d$ to remove texture regions from LL subband. After Wiener filtering the smooth regions, the texture regions are added back into LL subband. This
results in most of the noise in the LL subband being filtered out; therefore the clarity of the texture regions are preserved without blurring.

3.3. Experimental Results

The “gardener” image with additive iid Gaussian $N(0,\sigma)$ is used as the test data in the experiment. In order to reduce the computational complexity, the Haar wavelet basis is used in one level wavelet decomposition, the decomposition result is illustrated in Figure 3.4(b). We can see because of the existence of noise, the edges of the original image are not very clear. Figure 3.6 shows the $F_{edge}$ filtering result of HH, LH and HL subbands. The new algorithm is compared with uniform thresholding, Wiener filtering and Median Filtering in the spatial domain. For uniform thresholding, the threshold for each subband is selected as half of the maxima coefficient among each subband. Table 3.1 presents the SNR of the various methods for several values of noise strength $\sigma$, as mentioned in[1], according to the table, Wiener filter is superior to uniform thresholding, however, it is not necessarily visually superior.

In Figure 3.8(a), the original image is shown, noisy image with iid Gaussian $N(0,\sigma)$, $\sigma = 0.03$ is shown in Figure 3.8(b), Figure 3.8(c) displays the denoised image using the new algorithm. This figure is sharper and preserves more details than the image denoised using uniform thresholding in figure 3.8(d). Figure 3.8(e) illustrates the denoised image using Wiener filtering. Since the texture regions are removed from LL subband before the Weiner filtering and then added back later, at the expense of preserving more details, there is a little artifact around the boundary of the texture and
smooth regions. It can be reduced by adding a layer of pixels around texture regions using *dilation*. The denoising results using various methods of two sample images: gardener and cameraman are shown in Table 3.1.

<table>
<thead>
<tr>
<th>SNR/σ</th>
<th>0.015</th>
<th>0.03</th>
<th>0.045</th>
<th>0.06</th>
<th>0.075</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gardener</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noisy</td>
<td>9.3933</td>
<td>7.1551</td>
<td>6.0103</td>
<td>5.3296</td>
<td>4.8156</td>
</tr>
<tr>
<td>New method</td>
<td><strong>15.4504</strong></td>
<td><strong>13.8279</strong></td>
<td><strong>12.7022</strong></td>
<td><strong>11.8735</strong></td>
<td><strong>11.1767</strong></td>
</tr>
<tr>
<td><strong>Cameraman</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noisy</td>
<td>11.0734</td>
<td>8.5670</td>
<td>7.2105</td>
<td>6.3858</td>
<td>5.8063</td>
</tr>
<tr>
<td>New method</td>
<td><strong>16.7548</strong></td>
<td><strong>15.0305</strong></td>
<td><strong>13.9931</strong></td>
<td><strong>13.1134</strong></td>
<td><strong>12.5539</strong></td>
</tr>
<tr>
<td>Wiener</td>
<td>15.5299</td>
<td>13.6394</td>
<td>12.3557</td>
<td>11.5242</td>
<td>10.8291</td>
</tr>
</tbody>
</table>

Table 3.1: The SNR(in dB) of the noisy image and as of the denoised image using the various methods for several values of normalized variance $\sigma$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 3.8(a): Original *Gardener*

Figure 3.8(b): noisy image

Figure 3.8(c): denoising using new algorithm

Figure 3.8(d): denoising using Uniform thresholding algorithm
Figure 3.8(e): Denoising using Wiener filter
3.4 Summary

In this chapter a simple and effective algorithm for image denoising in Discrete Wavelet Transform (DWT) domain is proposed, being different from other methods for image denoising in DWT domain, the new method directly extracts the edges and texture details of an image from the spatial information contained in LL subband instead of detecting the edges across multiple scales. Hence, the computational complexity can be dramatically reduced. Both soft thresholding and hard thresholding require that the energy of the reconstructed image to be higher than the energy of the noise[5]. Our algorithm does not select the coefficients only based on their values. Therefore, the method of distinguishing the noise coefficients from image signal coefficients, which is addressed in this thesis, is more effective than thresholding.

Because of the property that compared to spectrums of the other transforms where there is no visible correlation to the image itself, spatial information is still contained in DWT domain, other methods to manipulate the coefficients in spatial manner could be investigated in the future. The possible approaches include better methods for image segmentation, texture extraction and edge detection.
4.1 Introduction

Image interpolation is a very important application in image processing, computer graphics and vision. This technique is used to smooth the surface of an object, enlarge an image, or recover a signal from the limited samples. The base of interpolation is the sampling theorem. If the image formation process meets the conditions of the sampling theorem, the digital image is a complete representation of the continuous image[22].

4.2 Interpolation in the Spatial Domain

There are many image interpolation methods used in the spatial domain such as nearest-neighbor, linear and bicubic interpolation. The easiest interpolation method is the nearest-neighbor interpolation, it is also called zero-order hold.

The nearest neighbor interpolation strategy may be formulated and analyzed in the terms of the following equation:

\[ f(x_1, x_2) = \sum \sum f^*(n, m)g(x_1 - nX_s, x_2 - mY_s) \]  

where \( X_s \) and \( Y_s \) are the \( x_1 \) and \( x_2 \) spatial sampling intervals. Simply stated, given a point \((x_1', x_2')\), which may or may not be on the sampling grid. \( g \) becomes unity in Eq. (4.1)
for the sampled image plane location to closest in distance to \((x_1, x_2)\). This choice of interpolation function corresponds to

\[
g(x_1 - nX, x_2 - mY) = \begin{cases} 
1 & \text{for } (n, m) \text{s.t. } (x_1 - nX)^2 + (x_2 - mY)^2 \text{ is min} \\
0 & \text{otherwise}
\end{cases}
\]

This formulation also allows handling of the case where there is no unique value of \((m, n)\) that satisfies Eq. (4.2), such as the case of \(P\) being equidistant from all four neighbors.

Linear interpolation is a first-order hold, this method is to find the average value between two pixels and use that as the pixel value between the two. Normally, we do this for the row first, then take the result and expand the columns in the same way, while assuming that the input image is zero outside the boundary. There are two kinds of methods for linear interpolation, according to the domains of the objects; they are linear interpolation and bi-linear interpolation. Figure 4.1 is the diagram for linear interpolation methods.

Figure 4.1: Diagram of linear interpolation methods
In the above diagram, \( g_2 = \left( \frac{x_3 - x_1}{x_2 - x_1} \right) (g_2 - g_1) + g_1 \), application to images requires a generalization to two dimensions, the resulting technique is known as bi-linear interpolation. In that Figure, A, B, C and D are the pixels in the vicinity of pixel P, whose value we need to establish. First, we interpolate along the x direction by applying the above equation between A and B to obtain the value of point Q. Then we can use points C and D to obtain the value at R. Finally we apply the equation again, this time in the y direction, between points Q and R to get the value at P.

Another method that achieves the same result requires convolution, i.e. convolving the zero interlaced image with the mask \( H \) as follows [15,23].

\[
H = \begin{bmatrix}
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\
\frac{1}{2} & 1 & \frac{1}{2} \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4}
\end{bmatrix}
\] (4.3)

Higher-order interpolation, say \( m \) order, is possible by padding each row and each column of the input image by \( m \) rows and \( m \) columns of zeros, respectively, and convolving it \( m \) times with the filter \( H \), as shown in Figure 4.2. For instance \( m = 3 \) yields a cubic spline interpolation in between the pixels[23].

![Figure 4.2: m-th order image interpolation](image_url)
4.3 Interpolation in the DWT domain

Before we state how the interpolation in the DWT domain is performed, let us first define what a modulus maxima is. Most traditional image applications determine sharp variance points by examining the first and second derivatives of the signal. This is because inflection points indicate neighborhoods of signal variation, and inflection points in the signal domain correspond to the extrema points of the first derivative and to the zero-crossings of the second derivative of the signal. Furthermore, extrema points of large magnitude in the first derivative represent points of sharp variation, while those of small magnitude imply points of slow transition. The modulus maxima is the maxima points of the absolute value of the first derivative[24,26].

Define a smoothing function \(\theta(x)\), which satisfies

\[
\lim_{x \to \pm \infty} \theta(x) = 0, \quad (4.4)
\]

and

\[
\int_{-\infty}^{\infty} \theta(x) dx = 1 \quad (4.5)
\]

We assume that \(\theta(x)\) is differentiable and we introduce a function \(\psi(x)\) as the first derivative of \(\theta(x)\):

\[
\psi(x) = \frac{d\theta(x)}{dx} \quad (4.6)
\]

We call a wavelet any function whose average is 0. Hence, \(\theta(x)\) can be considered as a wavelet since

\[
\int_{-\infty}^{\infty} \psi(x) dx = 0 \quad (4.7)
\]

now we define \(\psi_s(x)\) the detailed version of the wavelet function.
\[ \psi(x) = \frac{1}{s} \psi\left(\frac{x}{s}\right), \quad (4.8) \]

and we refer \( s \) as the scale. To take a wavelet transform of a signal, it is convolved with the wavelet function. The wavelet transform of \( f(x) \) at scale \( s \) and position \( x \) is denoted by \( W_s f(x) \), where

\[ W_s f(x) = f(x) * \psi_s(x), \quad (4.9) \]

and \( * \) is the convolution operator. From the linearity of convolution and differentiation, it could be verified easily that

\[ W_s f(x) = f(x) * (s \frac{d}{dx} \theta_s(x)) = s \frac{d}{dx} (f * \theta_s)(x) \quad (4.10) \]

where \( \theta_s(x) \) is defined similarly as \( \psi_s(x) \). Eq. (4.10) shows that taking the wavelet transform of the signal at scale \( s \) and at position \( x \) is equivalent to taking the first derivative of the smoothed signal. Hence, if we detect an extremum in the wavelet transform, we have found an inflection point in \( f * \theta_s \). For an extremum of large magnitude, or a maximum of \(|W_s f(x)|\), it has the physical meaning of being in a region of sharp transition in the signal domain, while for an extremum of small magnitude, it indicates a region of slow transition. This is the reason why spatial information is still contained in the DWT domain.

From the above discussion we see that a sharp variation point induces modulus maxima in the wavelet transform. Figure 4.3 shows the wavelet transform of a waveform consisting of a step, an impulse, and their smoothed versions. Each isolated singularity produces extrema points, which propagate across the scales[27].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
This important property of discrete wavelet transform also applies to 2-D images, as we discussed earlier, the wavelet transform breaks an image down into four subsampled, or decimated, images. These results consist of one image that has been highpass filtered in both the horizontal and vertical directions, one that has been highpass filtered in the vertical and lowpass filtered in horizontal, one that has been lowpassed in the vertical and highpassed in the horizontal, and one that has been lowpass filtered in both directions, known as "HH", "LH", "HL", and "LL". Further, a second level wavelet transform is performed in the LL corner of Figure 4.4 creating seven subimages. This process is called *multiresolution decomposition*.

Figure 4.3 A synthetic waveform and its wavelet transform, showing the propagation of extrema points across the scale
One important property of the DWT is that its high frequency subbands hold the directional edge information at different resolutions. Therefore, the edge information can be employed to find the relationship between adjacent subbands. Based on this property, Caley et al [4] proposed a regularity-preserving image interpolation method, which results in a better quality of interpolated image. Their algorithm estimates the regularity of edges by measuring the decay of wavelet transform coefficients across scales and attempts to preserve the underlying regularity. A new subband is extrapolated using edges, which are selected by measuring regularity and correlation between scales. However, Carey’s algorithm pays more attention to the image quality than to the computational complexity. It measures regularity using the entire wavelet decomposition subbands in order from coarsest scale to finest scale, which are undecimated. Therefore, it is not practical in a situation where we have only down-sampled subbands in a database. Our purpose is to utilize these limited coefficients to extrapolate the new subband, and reduce the computational complexity as well. Figure 4.4 shows the subbands of the orthogonal discrete wavelet transform (DWT), also shows the parent-child relationship. The solid line forms the decimated subbands, where \( S_1 \) is the finest subband and \( S_3 \) is the coarsest. Our purpose for interpolation in DWT domain is to extrapolate a new subband, \( S_0 \), which is illustrated by dashed lines.
In general, interpolation in the wavelet domain can be represented by Figure 4.5. For a $N \times N$ image, the wavelet transform breaks it down into four subsampled images, known as "$HH_1$", "$HL_1$", "$LH_1$", and "$LL_1$". The $n$-th row (column) of $\{LH_1, HL_n, HH_1\}$ is used to estimate rows and columns of the scale 0 row(column) of $\{LH_0, HL_0, HH_0\}$. The in-between lines are then filled in by interpolation. After $S_0$ is constructed and combined with the existing wavelet subbands, we can obtain the reconstructed and enlarged image by applying inverse wavelet transform to the new set of wavelet coefficients.
4.4 Experimental Results

In our experiments, several images are used in our experimentation. Table 4.1 includes detailed information about these images.

<table>
<thead>
<tr>
<th>Image Name</th>
<th>Category</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gardener</td>
<td>Nature image</td>
<td>512x512</td>
</tr>
<tr>
<td>Cameraman</td>
<td>Nature image</td>
<td>512x512</td>
</tr>
<tr>
<td>Bay Area</td>
<td>Bird’s eye of bay area, CA</td>
<td>512x512</td>
</tr>
<tr>
<td>Ellipt-2</td>
<td>Planetary nebulae</td>
<td>512x512</td>
</tr>
</tbody>
</table>

Table 4.1: Set of images used in the experimentation

We first down-sampled these images to the size of 256-by-256, then applied the spatial domain and discrete wavelet domain interpolation methods to these down-sampled images.
images. After interpolation, modification and image denoising which was discussed in chapter 3, the size of the image is recovered to 512-by-512. We compared these interpolated images with the original ones. The experimental data is obtained in the DWT domain using Haar, Db2, and Bior2.4. Figure 4.6 through 4.13 are reconstrated images from their down-sampled data and original images.

The quality of the remonstrated images is measured by computing Peak Signal-To-Noise Ratio (PSNR). Table 4.2 shows the PSNR difference between the interpolation methods, including nearest neighbor, bilinear, bicubic interpolation, and using different wavelet basis, including Haar, Db2, and Bior2.4.

From the experimental results, we can see the PSNR from discrete wavelet transform domain approaches are comparable with the spatial domain methods, sometimes even better. For the visual effect, the images reconstructed using DWT domain approaches are sharper and preserve more details than the image reconstructed using spatial domain methods. This is caused by two reasons: First, the HH, LH and HH subbands in DWT domain hold the directional edge information. Direct manipulation in DWT domain will not attenuate edge coefficients but tend to enhance them. Second, noise coefficient dies out swiftly with increasing scales. Therefore, the noise is reduced.
Figure 4.6: Original Gardener
512 x 512

Figure 4.7: Interpolating Gardener
Using bilinear method and Db2
Wavelet

Figure 4.8: Original Cameraman
512 x 512

Figure 4.9: Interpolating Cameraman
Using bilinear method and Bior2.4
Wavelet

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 4.10: Original Bay Area 512×512

Figure 4.11: Interpolating Bay Area Using bicubic method and Db2 Wavelet

Figure 4.12: Original Planetary Nubulae 512×512

Figure 4.13: Interpolating Planetary Nubulae Using bilinear method and Db2 Wavelet

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Linear interpolation is claimed to perform satisfactorily in most of the image processing applications[23]. From our experimental data, the linear interpolation scheme results in better image quality compared with the other two spatial domain interpolation methods. This is also mentioned in Biao Chen’s paper[28]. Among the wavelet transform domain interpolation methods, using bilinear interpolation method with Haar basis results in better interpolation quality. This is because that the DWT is a convolution process, when other mother wavelets are employed, the size of each subband is not necessity of a power of two. It is very useful to develop corresponding interpolation methods in DWT domain for other mother wavelets since they are also widely used today for image compression and analysis. In our experiments, the denoising method, which is discussed in chapter 3, is applied to the reconstructed image. The quality of the reconstructed images using other mother wavelets are very good, the visual deference between image reconstructed using Haar wavelet and other wavelets is not noticeable.
4.5 Time Analysis

As we discussed earlier, many images are compressed and stored by coefficients, which is based on the Discrete Wavelet Transform (DWT). Traditionally, to interpolate a compressed image, we usually need to fully decompress the image, do the related manipulation, then recompress it to save it back into the storage media, since our algorithm interpolate the DWT coefficients directly in DWT domain, computational complexity and processing time are reduced dramatically.

Suppose the image size is \( N \times N \) where \( N = 2^m \). Figure 4.14 compares the conventional methods and the DWT domain approaches for interpolation operation.

![Diagram](image1.png)

Figure 4.14(a): Conventional Method for Image Interpolation

![Diagram](image2.png)

Figure 4.14(b): New method for image interpolation

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In figure 4.15, $C_{j,k}$ and $C_{x,y}$ are pixel values for the different locations of an image, $W_{j,k}$ and $W_{x,y}$ are the corresponding DWT coefficient of $C_{j,k}$ and $C_{x,y}$; $O_{m,n}$ is output pixel values after interpolation operation in spatial domain; $W_{m,n}$ is the DWT coefficient of $O_{m,n}$; $IO$ and $IO^{-1}$ are operators related to interpolation operation in the spatial and DWT domain; $IDWT$ and $DWT$ are inverse and forward DWT, respectively. Intuitively, performing image interpolation in DWT domain can reduce the computational efforts.

The current implementation of the proposed algorithm for image interpolation in DWT domain requires about 0.44 CPU seconds and 2168441 times floating-point operations to interpolate a 256 x 256 image to the size of 512 x 512 using Matlab software on a Pentium III processor (700 Mhz). Under the same condition, using traditional interpolation method in spatial domain requires about 1.04 CPU seconds and 6110465 times floating-point operations to interpolate the same image to the size of 512 x 512, the processing time is reduced dramatically.

4.6 Summary

This chapter is related to the image interpolation in DWT domain, the possibility and efficient of such method were investigated. Due to the property of DWT, the computational complexity and processing time are reduced for image interpolation in DWT domain.

A fast and efficient interpolation algorithm in the DWT domain was proposed in this chapter. For any image, which is compressed by DWT, we can use the $HH$, $LH$ and $HL$ subbands of its DWT coefficients for the first scale to extrapolate
the new subbands by interpolation. The original image itself can be considered as $LL$ subband. Then, inverse DWT is implemented to get the enlarged image. In the experiments, different interpolation methods and different mother wavelets were investigated. The processing time and computational complexity were also discussed. The experimental result indicate the quality of the interpolated images is comparable and even better than those obtained by spatial interpolation methods while the processing time is reduced.

Compared to the previous DWT domain interpolation approaches, which only focus on Haar wavelet, the Interpolation algorithms for other wavelets were also addressed. The quality of the reconstructed image using other mother wavelets is very good. The visual deference between image reconstructed using Haar wavelet and other wavelets is not noticeable.
CHAPTER 5

CONCLUSIONS AND FUTURE WORK

This thesis proposed two techniques to image manipulation in the Discrete Wavelet Transform (DWT) domain: Image interpolation and denoising. The motivation comes from two important properties of DWT. First, compared to spectrums of the other transforms where there is no visible correlation to the image itself, spatial information is still contained in the DWT domain. Second, the DWT has the property of the multiresolution decomposition.

Based on the fact that textures and edges could tolerate some noise but not blurring, whereas smooth regions tolerate blurring but not noise, algorithm for image denoising in DWT domain was derived. Being different from other methods for image denoising in DWT domain, the new method directly extracts the edges and texture details of an image from the spatial information contained in LL subband instead of detecting the edges across multiple scales. Hence, the computational complexity can be dramatically reduced. The experimental results indicate that the new algorithm results in better image quality compared with the other image denoising methods such as uniform and adaptive thresholding in DWT domain, Wiener Filtering in Fourier Transform domain and Median filtering in the spatial domain.

Since the DWT has become more popular for image analysis and compression, algorithm for interpolation in DWT domain was also addressed. Through the
experimental results, we can see the image quality using DWT domain approaches is comparable with those using spatial domain interpolation methods and the visual effect is even better. This is caused by two reasons: First, the HH, LH and HH subbands in DWT domain hold the directional edge information. Direct manipulation in DWT domain will not attenuate edge coefficients but tends to enhance them. Second, noise coefficient dies out swiftly with increasing scale. Therefore, the noise is reduced.

The previous DWT domain interpolation approaches only focus on Haar wavelet, this is because the wavelet decomposition is a convolution process. The different lengths for analysis filters for the DWT make the size of the subbands not necessarily a power of 2. Therefore, the new subbands can not be simply appended to the existing subbands. This thesis extends the algorithm for interpolation in DWT domain to other wavelet bases, such as Daubecuies and Bior. Most noise introduced by size modification of new subbands was filtered out using the denoising algorithm addressed in this thesis. The quality of the reconstructed images using other wavelet bases is very good. The visual deference between images reconstructed using Haar wavelet and other wavelets is not noticeable.

Further research is needed to improve the results of our algorithms. For image denoising in DWT domain, other methods to manipulate the coefficients in spatial manner could be investigated. The possible approaches include better methods for image segmentation, texture extraction and edge detection. In addition, other useful interpolation methods for image interpolation, which combine our algorithms with statistical properties of DWT coefficients across multiple scales, should be investigated.
BIBLIOGRAPHY


66


VITA

Graduate College
University of Nevada, Las Vegas

Shaojie Wu

Local Address:
1600 East University Avenue, Apt #208
Las Vegas, NV 89119

Home Address:
#19-5-301,Yuquan Road No.19(B)
Beijing 100039, P.R. China

Degrees:
Bachelor of Science, Electrical Engineering, 1996
Beijing Union University

Publications:
Spatial Image Denoising in Discrete Wavelet Transform Domain,

Spatial Image Denoising in Discrete Wavelet Transform Domain,
Submitted to the Computer Version and Image Understanding.

Thesis Title: Image Interpolation and Denoising in Discrete Wavelet Transform Domain

Thesis Examination Committee:
Chairperson, Dr. Shahram Latifi. Ph.D.
Committee Member, Dr. Henry Selvaraj
Committee Member, Dr. Sahjendra Singh
Graduate Faculty Representative, Dr. Zhiyong Wang

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.