I, 

Swathi Nibhanupudi

hereby submit this as part of the requirements for the degree of:

Master of Science

in:

Computer Engineering

It is entitled:

Signal Denoising Using Wavelets

Approved by:

Dr. Carla Purdy

Dr. Harold W. Carter

Dr. Wen Ben Jone

Dr. Robert L. Ewing
Signal Denoising Using Wavelets

A thesis submitted to the
Division of Research and Advanced Studies
of the University of Cincinnati

in partial fulfillment of the
requirements for the degree of

Master of Science

in the Department of Electrical and Computer Engineering
and Computer Science
of the College of Engineering

2003

by

Swathi Nibhanupudi


Committee Chair: Dr. Carla Purdy
Abstract

In any type of signal processing, it has been demonstrated that it is important to remove noise from the signal before recognizing or classifying the patterns. Otherwise, the whole process may give wrong results. In this work the choice of denoising mechanisms for various types of input data and Gaussian noise is explored, to increase the signal strength.

In this thesis, denoising the input signals using a wavelet transform is discussed. It is shown that the performance of a signal classifier improves when these denoising techniques are introduced before actually applying the classifier. For our experiments, the classifier applied is a hybrid intelligent system that employs three important techniques of artificial intelligence, namely genetic algorithms, neural networks and fuzzy logic. Along with explaining the denoising algorithm clearly, this work shows the importance of selection of a suitable wavelet for the given input data and thus shows that the efficiency of a signal denoiser depends on three factors: the thresholding techniques, the kind of wavelet used in denoising, and the synchronization between the wavelet selected and the input data. This statement is justified with results from experiments on ECG data which employ different kinds of wavelets such as Haar, Daubechies, Symlet and Coiflet. The improvements in denoising after using vector quantization of wavelet coefficients before thresholding are also discussed.
Acknowledgements

I am deeply grateful to my advisor, Dr. Carla Purdy, for introducing the subject of hybrid intelligent systems to me with such clarity and insight, and for her continual and patient guidance and support. Dr. Purdy has helped to create a stimulating environment here for the study of broader topics of data and image classification. I have been inspired by her rigor and attention to detail and look forward to future collaborations with her.

For their support, comments, suggestions, and careful reading of my work, I am grateful to the members of my supervisory committee: Dr. Hal Carter, Dr. Wen Ben Jone and Dr. Robert L. Ewing.

I would like to express my appreciation to Professor Yuan Zhang for his assistance and advice with research involving the wavelet analysis and for providing latest information in the field. I would also like to thank Roshdy Youssif for constant advice and for providing the data and analytical tools and also for help with initial data analysis.

To my family, I am deeply grateful for their patient tolerance and loving support and motivation. My friends were gracious, patient, and loving throughout my studies. Finally, I would like to thank my lab mates Aravind, Jeff, Shruthi, Dennis and Vikram as well as the Computer Engineering faculty, students, and staff for supporting my education, providing excellent research facilities, and for being a great group of knowledgeable and fun people to work with.
# Table of Contents

List of Tables...........................................................................................................4

List of Figures...........................................................................................................5

1. INTRODUCTION.................................................................................................7

1.1 Introduction to Signal Processing.................................................................7

1.1.1 Basic Definitions.........................................................................................10

1.2 Organization of the Thesis............................................................................12

2. WAVELET THEORY.........................................................................................14

2.1 Introduction to Wavelets................................................................................14

2.2 Fourier Transform versus Wavelet Transform...........................................16

2.3 Significant Properties of Wavelets...............................................................19

2.4 Types of Wavelets.........................................................................................26

2.5 Wavelet Applications....................................................................................34

3. SIGNAL DENOISING.......................................................................................37

3.1 Introduction to Noise.....................................................................................37

3.2 Denoising Procedure......................................................................................41

3.2.1 Decomposition...........................................................................................41

3.2.2 Thresholding.............................................................................................42

3.2.3 Reconstruction...........................................................................................45

4. SIGNAL CLASSIFICATION AND ECG ANALYSIS.................................47

4.1 Introduction to Signal Classification............................................................47

4.2 ECG Characteristics......................................................................................54

4.3 Applications of Wavelets to ECG.................................................................58
List of Tables

2.1 Summary of Wavelet Families and Associated Properties……………………………33

5.1 Denoising Versus Levels of Decomposition for ECG Data…………………………69

5.2 Denoising Versus Wavelet Family Selected for ECG Data……………………………70

5.3 Denoising Versus Different Methods of Thresholding for ECG Data…………………71

5.4 Denoising Versus Threshold Selection Rules for ECG Data…………………………71

6.1 Mean Square Errors Versus Decomposition Levels……………………………………82

6.2: Mean Square Errors Versus Wavelet Family……………………………………………83

6.3: Mean Square Errors Versus Thresholding Methods……………………………………83

6.4: Mean Square Errors Versus Thresholding Rules………………………………………84

7.1: Classifier Performance for Different Algorithms………………………………………89
List of Figures

2.1 Diagrams of Sinusoidal Signal and a Daubechies Wavelet .......................... 14
2.2 Fourier Basis Functions, Time-Frequency Tiles, and Coverage of the
    Time-Frequency Plane ............................................................................. 17
2.3 Daubechies Wavelet Basis Functions, Time-Frequency Tiles, and
    Coverage of the Time-Frequency Plane ................................................. 17
2.4 Time Domain, Frequency Domain and Time-Frequency Domain .......... 18
2.5 Diagram for Continuous Wavelet Analysis ............................................ 20
2.6 Representation of Wavelet Shifting ....................................................... 21
2.7 Wavelet Decomposition ....................................................................... 22
2.8 Wavelet Reconstruction ....................................................................... 23
2.9 Wavelet Analysis ................................................................................. 24
2.10 Wavelet Packet Decomposition Tree .................................................... 24
2.11 Different Families of Wavelets ............................................................ 28
2.12 Daubechies Wavelets db4 on the Left and db8 on the Right ............... 29
2.13 Symlets sym4 on the Left and sym8 on the Right .................................. 30
2.14 Coiflets coif3 on the Left and coif5 on the Right ................................ 32
3.1 A Simple De-Noising Example ............................................................... 40
3.2 Demonstration of Hard thresholding and Soft thresholding ................. 43
4.1 Block Diagram of Signal Classifier ....................................................... 50
4.2 Original and Reconstructed Fingerprints .............................................. 52
4.3 Representative Signal Cycle of a Normal ECG .................................... 54
4.4 Examples of Normal and Pathological ECG signals

4.5 ECG Temporal Analysis – Characteristic Points and Calculation of Areas Above and Below Isoline

4.6 Details of Artificial Signal Obtained by Filtering by Equivalent FIR Filter of Wavelet Transform

4.7 ECG Record and its Significant Attributes

4.8 Detection of Significant Points of an ECG Signal

5.1 Wavelet Packets

5.2 Levels of Decomposition Using Wavelet Packet Trees

5.3: The Original Signal and Its Noisy Version

5.4: Denoised Signal at the First Level of Decomposition

5.5: Denoised Signal at the Second Level of Decomposition

5.6: Denoised Signal at the Third Level of Decomposition

5.7: Denoised Signal at the Fourth Level of Decomposition

6.1 One Dimensional Vector Quantization

6.2 Two Dimensional Vector Quantization

7.1 A Sample Classification Tree

7.2 Flowchart of Building the Classification Tree
Chapter 1

Introduction

1.1 introduction to Signal Processing

Digital Signal Processing, a field which has its roots in 17\textsuperscript{th} and 18\textsuperscript{th} century mathematics, has become an important tool in a multitude of diverse fields of science and technology.

The field of digital signal processing has grown enormously in the past decade to encompass and provide firm theoretical backgrounds for a large number of individual areas. [1]

The term “digital signal processing” may have a different meaning for different people. For example, a binary bit stream can be considered a “digital signal” and the various manipulations, or “signal processing”, performed at the bit level by digital hardware may be construed as “digital signal processing”. But, the viewpoint taken in this thesis is different. Implicit in the definition of digital signal processing (DSP) is the notion of an information-bearing signal that has an analog counterpart. What are manipulated are samples of this implicitly analog signal. Further, these samples are quantized, that is represented using finite precision, with each word representative of the value of the sample of an (implicitly) analog signal. These manipulations, or filters, performed on these samples are arithmetic in nature – additions and multiplications. The definition of DSP includes the processing associated with sampling, conversion between analog and digital domains, and changes in wordlength. [2]
Digital Signal Processing is concerned with the representation of signals by sequences of numbers or symbols and the processing of these sequences. The purpose of such processing may be to estimate characteristic parameters of a signal or to transform a signal into a form which is in some sense more desirable.

Signal processing, in general, has a rich history, and its importance is evident in such diverse fields as biomedical engineering, acoustics, sonar, radar, seismology, speech communication, data communication, nuclear science and many others. In many applications, as, for example, in Electrocardiogram (ECG) analysis or in systems for speech transmission and speech recognition, we may wish to remove interference, such as noise, from the signal or to modify the signal to present it in a form which is more easily interpreted. As another example, a signal transmitted over a communications channel is generally perturbed in a variety of ways, including channel distortion, fading, and the insertion of background noise. One of the objectives at the receiver is to compensate for these disturbances. In each case, processing of the signal is required.

Signal processing problems are not confined to one-dimensional signals. Many image processing applications require the use of two-dimensional signal processing techniques. This is the case, for example, in x-ray enhancement, the enhancement and analysis of aerial photographs for detection of forest fires or crop damage, the analysis of satellite weather photos, and the enhancement of television transmissions from lunar and deep space probes. Seismic data analysis as required in oil exploration, earthquake
measurements and nuclear test monitoring also utilizes multidimensional signal processing techniques. [3]

There are many reasons why digital signal processing of an analog signal may be preferable to processing the signal directly in the analog domain. First, a digital programmable system allows flexibility in reconfiguring the digital signal processing operations simply by changing the program. Reconfiguration of an analog system usually implies a redesign of the hardware followed by testing and verification to see that it operates properly.

Accuracy considerations also play an important role in determining the form of the signal processor. Tolerances in analog circuit components make it extremely difficult for the system designer to control the accuracy of an analog signal processing system. On the other hand, a digital system provides much better control of accuracy requirements.

Digital signals are easily stored on magnetic media (tape or disk) without deterioration or loss of signal fidelity beyond that introduced in the A/D conversion. As a consequence, the signals become transportable and can be processed off-line in a remote laboratory. The digital signal processing method also allows for the implementation of more sophisticated signal processing algorithms. It is usually very difficult to perform precise mathematical operations on signals in analog form, but these same operations can be routinely implemented on a digital computer using software.
The digital implementation of the signal processing system is almost always cheaper than its analog counterpart as a result of the flexibility for modifications. As a consequence of these advantages, digital signal processing has been applied in practical systems covering a broad range of disciplines.

However, digital implementation has its limitations. One practical limitation is the speed of operation of A/D converters and digital signal processors. Signals having extremely wide bandwidths require fast sampling rate A/D converters and fast digital signal processors. Hence there are analog signals with large bandwidths for which a digital processing approach is beyond the state of the art of digital hardware. [4]

The techniques and applications of digital signal processing are expanding at a tremendous rate. With the advent of large scale integration and the resulting reduction in cost and size of digital components, together with increasing speed, the class of applications of digital signal processing techniques is growing.

1.1.1 Basic Definitions

A ‘signal’ can be defined as a function that conveys information, generally about the state or behavior of a physical system. Although signals can be represented in many ways, in all cases the information is contained in a pattern of variations of some form. Signals are represented mathematically as functions of one or more independent variables.
The independent variable of the mathematical representation of a signal may be either continuous or discrete. ‘Continuous time signals’ are signals that are defined at a continuum of times and thus are represented by continuous variable functions. ‘Discrete time signals’ are defined at discrete times and thus the independent variable takes on only discrete values, i.e., discrete time signals are represented as sequences of numbers.

In addition to the fact that the independent variables can be either continuous or discrete, the signal amplitude (dependent variable) may be either continuous or discrete. ‘Digital signals’ are those for which both time and amplitude are discrete. Continuous time, continuous amplitude signals are called ‘Analog signals’.

In almost every area of science and technology, signals must be processed to facilitate the extraction of information. Thus, the development of signal processing techniques and systems is of great importance. These techniques usually take the form of a transformation of a signal into another signal that is in some sense more desirable than the original. For example, we may wish to design transformations for separating two or more signals that have been combined in some way; we may wish to enhance some component or parameter of a signal; or we may wish to estimate one or more parameters of a signal.

Signal processing systems may be classified along the same lines as signals. That is, ‘Continuous time systems’ are systems for which the both the input and output are continuous signals and ‘Discrete time systems’ are those for which the input and output
are discrete time signals. Similarly, ‘Analog systems’ are systems for which the input and output are analog signals and ‘Digital systems’ are those for which the input and output are digital signals. ‘Digital signal processing’, then, deals with transformations of signals that are discrete in both amplitude and time. [1]

Discrete time signals may arise by sampling a continuous time signal or they may be generated directly by some discrete time process. Whatever the origin of the discrete time signals, digital signal processing systems have many attractive features, as noted above. They can be realized with great flexibility using general purpose digital computers, or they can be realized with digital hardware. Thus, digital representations of signals are often desirable when sophisticated signal processing is required.

1.2 Organization of the Thesis

In this thesis, wavelet analysis is employed for denoising of a specific type of signals - ECG signals. The remainder of the thesis is structured as follows. In chapter 2, we review the fundamentals of wavelet theory, explaining significant properties of different types of wavelets. Their advantages and applications are discussed. In chapter 3, the important kinds of noise, their elimination techniques are explained. Here, the effect of thresholding methods and levels of decomposition to get wavelet coefficients are emphasized.

In chapter 4, the main structure and properties of ECG signals are explained. Chapter 5 contains the main algorithm used, the results achieved and their analysis. In chapter 6, the
The concept of vector quantization is explained. This includes the Linde-Buzo-Gray (LBG) algorithm [21] and its application in denoising. The results are explained with quantization added in the code. In chapter 7, the Genetic Neural Signal Pattern Classifier (GNSPC), a hybrid pattern classifier defined in [5] is described and the improvement in performance of this system after including denoising in its feature extraction stage is demonstrated. Chapter 8 gives the summary of this work and also gives the possible extensions of this implementation in future.
Chapter 2
Wavelet Theory

2.1 Introduction to Wavelets

Wavelet theory is the mathematics associated with building a model for a signal, system, or process with a set of “special signals”. The special signals are just little waves or “wavelets”. They must be oscillatory (waves) and have amplitudes, which quickly decay to zero in both the positive and negative directions (little). The required oscillatory condition leads to sinusoids as the building blocks. The quick decay condition is a tapering or windowing operation. These two conditions must be simultaneously satisfied for the function to be a little wave or wavelet. A wavelet is a waveform of effectively limited duration that has an average value of zero. Unlike sine waves (the basis of Fourier analysis), which are smooth and predictable, wavelets tend to be irregular and asymmetric.

![Diagram of Sinusoidal Signal and Daubechies Wavelet](Image)

Figure 2.1: Diagrams of Sinusoidal Signal and a Daubechies Wavelet in [6]

From the figure above, we can say that signals with sharp changes might be better analyzed with an irregular wavelet than with a smooth sinusoid, just as some foods are
better handled with a fork than a spoon, as quoted in [7]. Also, local features can be described better with wavelets that have local extent.

Sets of “wavelets” are employed to approximate a signal and each element in the wavelet set is constructed from the same function, the original wavelet, appropriately called the mother wavelet, by shifting (translating) and scaling (dilating or compressing) it.

Wavelet theory represents “things” by breaking them down into many interrelated component pieces. When the pieces are scaled and translated wavelets, this breaking down process is termed a wavelet decomposition or wavelet transform. Wavelet reconstructions or inverse wavelet transforms involve putting the wavelet pieces back together to retrieve the original object or process.

As a physical analogy of a wavelet transform, consider stargazing with a telescope. The analogy is that the sky (the input, f) is to be represented by multiple views (wavelet coefficients) through a lens (mother wavelet) at different scales (focuses or resolutions), and different “look” directions (translations). At each scale and translation (focus and position) a new “look” or view (wavelet coefficient) of the sky is created. [8]

Other than one-dimensional data, which encompasses most ordinary signals, wavelet analysis can be applied to two-dimensional data (images) and, in principle, to higher dimensional data. Wavelet analysis is capable of revealing aspects of data that other
signal analysis techniques miss such as trends, breakdown points, discontinuities in higher derivatives, and self-similarity.

2.2 Fourier Transform versus Wavelet Transform

The wavelet transform can be related to the more commonly used Fourier transform or Fourier series. The Fourier models represent functions as weighted sum of exponentials at different frequencies. The weight at each different frequency is called the Fourier coefficient at that frequency. Wavelet models analogously represent functions as a weighted sum of scaled and translated mother wavelets. The wavelet transform has a mother wavelet replace the exponential, scaling and translation replace frequency shifting, and a two-dimensional surface of wavelet coefficients replace the one dimensional Fourier coefficients.

One way to see the time-frequency resolution differences between the Fourier transform and the wavelet transform is to look at the basis function coverage of the time-frequency plane. The figure below shows a windowed Fourier transform (WFT), where the window is simply a square wave. The square wave window truncates the sine or cosine function to fit a window of a particular width. Because a single window is used for all frequencies in the WFT, the resolution of the analysis is the same at all locations in the time-frequency plane.
An advantage of wavelet transforms is that the windows vary. In order to isolate signal discontinuities, one would like to have some very short basis functions. At the same time, in order to obtain detailed frequency analysis, one would like to have some very long basis functions. A way to achieve this is to have short high-frequency basis functions and long low-frequency ones. This happy medium is exactly what you get with wavelet transforms. This figure shows the coverage in the time-frequency plane with one wavelet function, the Daubechies wavelet.

Figure 2.2: Fourier Basis Functions, Time-Frequency Tiles, and Coverage of the Time-Frequency Plane. [9]

Figure 2.3: Daubechies Wavelet Basis Functions, Time-Frequency Tiles, and Coverage of the Time-Frequency Plane. [9]
Wavelet transforms have an infinite set of possible basis functions. Thus wavelet analysis provides immediate access to information that can be obscured by other time-frequency methods such as Fourier analysis. Wavelets are mathematical functions that cut up data into different frequency components, and then study each component with a resolution matched to its scale.

**Fourier Transform** processes data referring to the **frequency**

**Wavelet Transform** processes data referring to the **scale**

In wavelet analysis we retain some frequency localization and some time localization, so it is a compromise between using filters or Fourier transforms.

![Figure 2.4: Time Domain, Frequency Domain and Time-Frequency Domain Representations [10]](image)

In summary, the Fourier transform provides some information about both when and at what frequencies a signal event occurs. Unfortunately the window size is fixed, hence the information is provided only with limited precision. It would be better to use windows, which are larger in the time domain in parts of the signal dominated by low frequency
information, and to use narrower windows in time during high frequency part since many signals require a more flexible approach. The Wavelet transform can do that, by using short window at high frequencies and long window at low frequencies.

Since wavelet analysis uses bases that are localized in time as well as frequency, it can represent non-stationary (transient) signals i.e., most natural and human-made signals more effectively. So, it’s more compact and easier to implement.

2.3 Wavelet Definition and Significant Properties

Continuous Wavelet Transform

The results of the Fourier transform are the Fourier coefficients \( F(\omega) \), which when multiplied by a sinusoid of frequency \( \omega \) yield the constituent sinusoidal components of the original signal. Similarly, the continuous wavelet transform (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function \( \psi \).

\[
Tf(a,b) = \langle f, \psi_{a,b} \rangle = \frac{1}{\sqrt{a}} \int f(t) \psi^* \left( \frac{t-b}{a} \right) dt
\]

If \( \Psi \) is such that

\[
C_{\psi} = \int_{-\infty}^{+\infty} \left| \frac{\Psi(\omega)}{\omega} \right|^2 d\omega < +\infty
\]
$f(t)$ can be reconstructed by an inverse wavelet transform:

$$f(t) = C^{-1} \int_{0}^{+\infty} \int_{-\infty}^{+\infty} T_C(a,b) \psi_{a,b}(t) \ db \ \frac{da}{a^2}$$

The results of the CWT are many wavelet coefficients $C$, which are a function of scale and position. Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal.

![Diagram for Continuous Wavelet Analysis](image)

Figure 2.5: Diagram for Continuous Wavelet Analysis [6]

**Scaling**

Scaling a wavelet simply means stretching (or compressing) it. We denote the scale factor by the letter $a$. The scale factor is related (inversely) to the frequency.

Low scale $a$ => Compressed wavelet => Rapidly changing details => High frequency $\omega$.

High scale $a$ => Stretched wavelet => Slowly changing, features => Low frequency $\omega$. 
Shifting

Shifting a wavelet simply means delaying (or hastening) its onset. Mathematically, delaying a function \( f(t) \) by \( k \) is represented by \( f(t-k) \).

![Wavelet function and Shifted wavelet function](image)

Figure 2.6: Representation of Wavelet Shifting [6]

The continuous wavelet transform is the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet. This process produces wavelet coefficients that are a function of scale and position.

Discrete Wavelet Transform

It turns out that, if we choose scales and positions based on powers of two—called dyadic scales and positions—then, our analysis will be much more efficient and just as accurate as the CWT. This kind of analysis is called the discrete wavelet transform (DWT).
Wavelet Decomposition (Analysis) - Approximations and Details

The approximations $A$ are defined as the high scale, low frequency components of the signal. The details $D$ are defined as the low scale, high frequency components. Suppose we start with 1000 samples. After decomposition we get 1000 $A$ components and 1000 $D$ components. These signals $A$ and $D$ are interesting, but we get 2000 values instead of the 1000 we had. To avoid this, we use downsampling in which, we keep only one point out of two in each of the two 2000-length samples to get the complete information. We produce two sequences called $cA$ and $cD$.

![Figure 2.7: Wavelet Decomposition [6]](image)

The process on the right, which includes downsampling, produces DWT coefficients. The detail coefficients $cD$ are small and consist mainly of a high-frequency noise, while the approximation coefficients $cA$ contain much less noise than does the original signal.

When we begin fitting the wavelets to time series, we can select the number of different frequencies that will be fitted and what those frequencies will be. For practical reasons the frequencies selected are always powers of two. For example, if we have a signal of 256 samples in time span of 1s, the highest frequency within the signal is 128 Hz. This is the first fitting frequency, the next are 64, 32, 16, 8, 4, 2. We have to downsample the
signal after each fitting to save CPU time. After the fit of 128 Hz we apply anti-aliasing filter and sample down by a factor of two. We now have only 128 samples on which the 64 Hz wavelet will be applied and so on. We define the number of frequencies (that are power of two) that will be fitted with the parameter called decomposition level.

**Signal Reconstruction (Synthesis) – Reconstructing Approximations and Details**

This is the process of assembling the components back into the original signal without loss of information. The mathematical manipulation to achieve this is called the inverse discrete wavelet transform (IDWT).

![Wavelet Reconstruction Diagram](image)

Figure 2.8: Wavelet Reconstruction [6]

**Wavelet Packets**

The decomposition of both approximations and details generates a wavelet packet. This results in a balanced binary tree structure. The bottom leaves contain detailed information for each frequency. The wavelet packet analysis offers much better frequency resolution than the simple wavelet analysis. We know that in simple wavelet analysis the DFT would produce 128 frequency lines from a signal with 256 samples and simple wavelet analysis would give only 8 lines (Level 1: 128Hz, Level 2: 64Hz, Level 3: 32Hz, Level 4:
16Hz, Level 5: 8Hz, Level 6: 4Hz, Level 7: 2Hz, Level 8: 1Hz). Wavelet packet analysis can produce all the frequencies as the DFT. To perform partial packet analysis, you can save one of the wavelet details to file and then perform wavelet decomposition on that detail. The second option is to simply perform FFT on the detail, when sufficient depth (level) has been reached. It should be noted that wavelets have sacrificed some frequency resolution and physical interpretation to get better time resolution.

In wavelet analysis, for an n level decomposition, there are n+1 possible ways to decompose or encode the signal.

\[
S = A_1 + D_1 = A_2 + D_2 + D_1 = A_3 + D_3 + D_2 + D_1
\]

Figure 2.9: Wavelet Analysis [6]

In wavelet packet analysis, the details as well as the approximations can be split. This yields more than \(2^{n-1}\) different ways to encode the signal.

Figure 2.10: Wavelet Packet Decomposition Tree. [6]
Wavelet packet analysis allows the signal $S$ to be represented as $A_1 + AAD_3 + DAD_3 + DD_2$. This is an example of a representation that is not possible with ordinary wavelet analysis.

**Multiresolution Analysis**

Wavelets provide multiresolution analysis. This is defined as the process of representing a signal by a finite sum of components at different resolutions so that each component can be processed adaptively based on the objectives of the application. It is the technique of representing signals compactly and in several levels of resolution to use for decomposition and reconstruction purposes.

**Selection of the Wavelet**

This is the most interesting question for most of the users. The wavelet has one or two parameters. Because wavelets have so many constraints, that are not associated with the signal, but more with math and calculation limitations, it is virtually impossible to blindly select a wavelet. Wavelets are usually chosen on the basis of “If you see what you need to see, then that's that, if not, then try something else”. The most general-purpose usable wavelet is Daubechies. The Haar wavelet is actually a differential operator. The Daubechies1 equals Haar. [12]

As mentioned, the wavelets have one primary parameter. This parameter defines two things: region of support and the number of vanishing moments. The region of support means, how long the wavelet is. This will affect the localization capabilities. The longer
the wavelet, the larger the part of the time series that will be taken into account for calculating the amplitude at any time position. And more averaging will occur, similar to that in DFT. The number of vanishing moments is always the same as the region of support level. The number of vanishing moments defines the order of the polynomial that will be ignored if present in the time series. Suppose we have added a series: \( x_1^2 + x_2^3 \) (sample by sample) to our original signal, if we define the number of vanishing moments to be 3, than the whole added series will be ignored when performing the wavelet decomposition. If the number of vanishing moments would be 2, than only the second degree term of the series added to the original signal would be ignored. This is an extra set of mathematical relationships for the coefficients that must be satisfied, and is directly related to the number of coefficients. [12]

### 2.4 Types of Wavelets

The attention of researchers gradually turned from frequency-based analysis using Fourier transforms to scale-based analysis using wavelet transforms when it started to become clear that an approach measuring average fluctuations at different scales might prove less sensitive to noise.

The first recorded mention of what we now call a "wavelet" seems to be in 1909, by Alfred Haar. The concept of wavelets in its present theoretical form was first proposed by Jean Morlet. The methods of wavelet analysis have been developed mainly by Y. Meyer and his colleagues, who have ensured the methods' dissemination. The main algorithm dates back to the work of Stephane Mallat in 1988. Since then, research on wavelets has
become international and is spearheaded by the work of scientists such as Ingrid Daubechies, Ronald Coifman, and Victor Wickerhauser. [10]

Based on experimental results, we have to choose to use any one kind of wavelets. We can go for the mother wavelet DB1 (Daubechies One) because its detail coefficients indicate sharp changes in a signal indicating transition states (acceleration or deceleration) and implement it. We can develop and implement an algorithm to segment a signal automatically using wavelets. The different wavelet families make different trade-offs between how compactly the basis functions are localized in space and how smooth they are. Wavelet transforms comprise an infinite set. Some of the wavelet bases, such as Daubechies, have fractal structure.

Within each family of wavelets are wavelet subclasses distinguished by the number of coefficients and by the level of iteration. Wavelets are classified within a family most often by the number of vanishing moments.
Different Families Of Wavelets

![Wavelet Families](image)

Figure 2.11: Different Families of Wavelets [10]

The number next to the wavelet name represents the number of vanishing moments. The qualities of these wavelet families vary according to several criteria, such as the speed of convergence of the function, the symmetry, the number of vanishing moments and the regularity.

They may also be associated with these less important properties such as the existence of an explicit expression, the ease of tabulating, and the familiarity with use. The definition equations for several wavelets are given explicitly by their time definitions, or by their frequency definitions, or by their filters.
Daubechies Wavelets: dbN

In $dbN$, $N$ is the order.

![Graphs of Daubechies Wavelets db4 and db8](image)

This family includes the Haar wavelet, written $db1$, the simplest wavelet imaginable and certainly the earliest. These are compactly supported wavelets with extreme phase and highest number of vanishing moments for a given support width. Associated scaling filters are minimum-phase filters. They are orthogonal, biorthogonal, provide compact support. Examples are db1 or haar, db4. Number of vanishing moments is $N$.

These wavelets have no explicit expression except for $db1$, which is the Haar wavelet. However, the square modulus of the transfer function of $h$ is explicit and fairly simple.

The support length of $\psi$ and $\phi$ is $2N - 1$. Most $dbN$ are not symmetrical.
Haar

Ψ(x), the wavelet function and Φ(x), the scaling function are expressed as follows.

\[ Ψ(x) = \begin{cases} 
1 & \text{if } 0 < x < 0.5; \\
-1 & \text{if } 0.5 < x < 1 \\
0 & \text{if } x \text{ is not between } 0 \text{ and } 1 
\end{cases} \]

\[ Φ(x) = \begin{cases} 
1 & \text{if } x \text{ is between } 0 \text{ and } 1 \\
0 & \text{if } x \text{ is not between } 0 \text{ and } 1 
\end{cases} \]

Haar wavelets are the oldest and the simplest wavelets. They are not continuous, but are symmetric. Number of vanishing moments is 1.

Symlet Wavelets: symN

Symlets are only near symmetric; consequently some authors do not call them symlets.

Figure 2.13: Symlets sym4 on the Left and sym8 on the Right [6]
Daubechies proposes modifications of her wavelets that say their symmetry can be increased while retaining great simplicity. The idea consists of reusing the function $m_0$ introduced in the $dbN$, considering the $|m_0(\omega)|^2$ as a function $W$ of $z = e^{i\omega}$.

Then we can factor $W$ in several different ways in the form of

$$W(z) = U(z)\left(\frac{1}{z}\right)$$

because the roots of $W$ with modulus not equal to 1 go in pairs. If one of the roots is $z_1$, then $1/z_1$ is also a root.

By selecting $U$ such that the modulus of all its roots is strictly less than 1, we build Daubechies wavelets $dbN$. The $U$ filter is a "minimum phase filter." By making another choice, we obtain more symmetrical filters; these are symlets.

The symlets have other properties similar to those of the $dbNs$. They have highest number of vanishing moments for a given support width. Associated scaling filters are near linear-phase filters. Order $N$ can be 2, 3 ... . They are orthogonal, biorthogonal and provide compact support. Number of vanishing moments is $N$. 
Coiflet Wavelets: coifN

Figure 2.14: Coiflets coif3 on the Left and coif5 on the Right [6]

Built by Daubechies at the request of Coifman, the function $\psi$ has $2N$ moments equal to 0 and, what is more unusual, the function $\phi$ has $2N-1$ moments equal to 0. The two functions have a support of length $6N-1$.

The $coifN$ $\psi$ and $\phi$ are much more symmetrical than the $dbNs$. These are compactly supported wavelets with highest number of vanishing moments for both $\phi$ and $\psi$ as $2N$ and $2N-1$ respectively. Order $N = 1, 2, ..., 5$. These are orthogonal, and biorthogonal.
<table>
<thead>
<tr>
<th>Property</th>
<th>haar</th>
<th>dbN</th>
<th>symN</th>
<th>coifN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crude</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infinitely regular</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arbitrary regularity</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Compactly supported orthogonal</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Compactly supported bioorthogonal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symmetry</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asymmetry</td>
<td></td>
<td>●</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Near symmetry</td>
<td></td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Arbitrary number of vanishing moments</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>Vanishing moments for</td>
<td></td>
<td></td>
<td></td>
<td>●</td>
</tr>
<tr>
<td>Existence of</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Orthogonal analysis</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Biorthogonal analysis</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Exact reconstruction</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>FIR filters</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Continuous transform</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Discrete transform</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Fast algorithm</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Explicit expression</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: Summary of Wavelet Families and Associated Properties [6]
2.5 Wavelet Applications

Signal processing includes the following operations: converting some physical phenomenon (i.e., reflected light, a contracting muscle, air motion) into a “signal” (or vice-versa), conditioning and manipulating the signal to extract or encode desired information, and interpreting and/or reacting to the extracted information.

Wavelet theory can be applied in many fields, such as speech analysis, image analysis, biomedical imaging, theoretical mathematics and physics, data compression, communication systems, control systems, oil exploration and seismic sensing, sonar, weather forecasting, stock market modeling, radar, air acoustics, and endless other signal processing areas. The primary, and most advantageous, application areas are those that have, generate, or process wideband signals (wideband signals are those that can simultaneously have both short term characteristics or trends and relatively long term characteristics as well – the duration of the interesting portion of the signal is not known a priori). In addition, systems involving time and or space variation (systems that are not time or space invariant as well are appropriately and efficiently represented by wavelet theory. The wavelets bring their own strong benefits: a local outlook, a multiscaled outlook, cooperation between scales, and a time-scale analysis. They demonstrate that sines and cosines are not the only useful functions and that other bases made of weird functions serve to look at new foreign signals, as strange as most fractals or some transient signals.[12]
Applications Due to Space Aspects of Wavelets

• Biology for cell membrane recognition, to distinguish the normal from the pathological membranes
• Metallurgy for the characterization of rough surfaces
• Finance, for detecting the properties of quick variation of values
• In Internet traffic description, for designing the services size

Applications Due to Time Aspects of Wavelets

• Industrial supervision of gear-wheel
• Checking undue noises in craned or dented wheels, and more generally in nondestructive control quality processes
• Detection of short pathological events as epileptic crises or normal ones as evoked potentials in EEG (medicine)
• SAR imagery
• Automatic target recognition
• Intermittence in physics

Some domains are very productive. Medicine is one of them. We can find studies on micro-potential extraction in EKGs, on time localization of electrical heart activity, in ECG noise removal. In EEGs, a quick transitory signal is drowned in the usual one. The
wavelets are able to determine if a quick signal exists, and if so, can localize it. Also, there are attempts to enhance mammograms to discriminate tumors from calcifications.
Chapter 3

Signal Denoising

3.1 Introduction to Noise

Before looking at different denoising techniques, it is essential to clarify what is meant by noise. Typically noise in a real-world data acquisition system has an unknown distribution.

We have seen that the wavelet transform is an indispensable tool for a variety of applications such as classification, compression, and estimation. One of the key properties of wavelets is the fact that they form unconditional bases for many different signal classes. Thus, most signal information in wavelet expansions is conveyed by a relatively small number of large coefficients. This property of the wavelet transform makes the use of wavelets particularly suited to signal denoising.

It has been shown that wavelets can remove noise more effectively than the previously used methods. This noise may be caused by several factors including disruptions in transmissions, imperfections in data collection, quality of audio recording, experimental error, etc. Undoubtedly, the use of wavelets will continue to impact the sciences through the practical use of noise reduction.
When models are created to either predict or classify sampled data, in many cases there may be a natural variation which occurs in a problem domain. It is impossible to create a general model which is capable of classifying or predicting all problems in the universe better than any model which has been designed for one particular problem. If a model is only required to distinguish between two different classes, then the more a priori information that can be obtained pertaining to those two classes the better.

The method for removing noise from scattered data varies with the perspective of the research. If the scattered data had no noise, then the problem would be one strictly of interpolation. The essence of the majority of the methods when there is noise is that the unobservable function we wish to approximate can be accurately approximated with a type or class of functions which have been selected in advance. In the case in which the noise process is assumed to be unknown, but there is some family of functions that has been preselected for the approximation, the method is known as ‘non-parametric regression’ [12].

Denoising always involves the implementation of a method for discovering a parameter, or a set of parameters, which control the tradeoff between bias and variance in the resulting model, where variance is roughly proportional to the number of coefficients included in the denoiser and bias is caused by the exclusion of nonzero coefficients. These parameters in some instance are called smoothing parameters. If the values of these parameters are chosen to be too small, then the noise variance will be the dominating factor. If these parameter values are selected to be too large, then the bias of the model
will dominate over the contents of the data, possibly resulting in the loss of high frequency contents of the signal.

Wavelet transforms are used to denoise data. This is accomplished by applying a wavelet transformation to the noisy data, thresholding the resulting coefficients which are below some value in magnitude, and then inverse transforming to obtain a smoother version of the original data. This process has been described by Donoho and Johnstone and called “Wavelet Shrinkage”. [11]

In many systems, the concept of Additive White Gaussian Noise (AWGN) is used. This simply means a noise, which has a Gaussian probability density function and white power spectral density function (noise distributed over the entire frequency spectrum) and is linearly added to whatever signal we are analyzing.

Here, we discuss the problem of signal recovery from noisy data. This problem is easy to understand looking at the following simple example where a slow sine is corrupted by a white noise.
The underlying one-dimensional model for the noisy signal is basically of the following form:

\[ s(n) = f(n) + \sigma e(n) \]

where time \( n \) is equally spaced.

In the simplest model we suppose that \( e(n) \) is a Gaussian white noise \( N(0,1) \) and the noise level \( \sigma \) is supposed to be equal to 1. The de-noising objective is to suppress the noise part of the signal \( s \) and to recover \( f \). The method is efficient for families of functions \( f \) that have only a few nonzero wavelet coefficients. These functions have a sparse wavelet representation. From a statistical viewpoint, the model is a regression
model over time and the method can be viewed as a nonparametric estimation of the function \( f \) using orthogonal basis.

### 3.2 Denoising Procedure

The general de-noising procedure involves three steps. The basic version of the procedure follows the steps described below.[11]

1. **Decompose** - Choose a wavelet, choose a level \( N \). Compute the wavelet decomposition of the signal \( s \) at level \( N \).
2. **Threshold detail coefficients** - For each level from 1 to \( N \), select a threshold and apply soft or hard thresholding to the detail coefficients.
3. **Reconstruct** - Compute wavelet reconstruction using the original approximation coefficients of level \( N \) and the modified detail coefficients of levels from 1 to \( N \).

#### 3.2.1 Decomposition

Given a signal to be denoised, we need to select a suitable wavelet and level of decomposition. Based on the different types of wavelets and their correlation to different signals as explained in the previous chapter, we select the appropriate wavelet.

Unlike conventional techniques, wavelet decomposition produces a family of hierarchically organized decompositions. The selection of a suitable level for the
hierarchy will depend on the signal and experience. Often the level is chosen based on a desired low-pass cutoff frequency.

At each level $j$, we build the $j$-level approximation $A_j$, or approximation at level $j$, and a deviation signal called the $j$-level detail $D_j$, or detail at level $j$. We can consider the original signal as the approximation at level 0, denoted by $A_0$.

One way of understanding this decomposition consists of using an optical comparison. Successive images $A_1$, $A_2$, $A_3$ of a given object are built. We use the same type of photographic devices, but with increasingly poor resolution. The images are successive approximations; one detail is the discrepancy between two successive images. Image $A_2$ is, therefore, the sum of image $A_4$ and intermediate details $D_4, D_3$:

$$A_2 = A_3 + D_3 = A_4 + D_4 + D_3$$

### 3.2.2 Thresholding

Two points to be addressed for thresholding are how to perform the thresholding, and how to choose the threshold.
Soft or Hard Thresholding

Hard thresholding is the simplest method. Soft thresholding has nice mathematical properties and the corresponding theoretical results are available.

Let $t$ denote the threshold.

The hard threshold signal $x$ is $x$ if $|x| > t$,

is $0$ if $|x| \leq t$.

The soft threshold signal $x$ is $\text{sign}(x)(|x| - t)$ if $|x| > t$

is $0$ if $|x| \leq t$.

Hard thresholding can be described as the usual process of setting to zero the elements whose absolute values are lower than the threshold. Soft thresholding is an extension of hard thresholding, first setting to zero the elements whose absolute values are lower than
the threshold, and then shrinking the nonzero coefficients towards 0. The hard procedure creates discontinuities at \( x = \pm t \), while the soft procedure does not.

**Threshold Selection Rules**

We have mainly four threshold selection rules.[11]

- **Rigsure** – threshold is selected using the principle of Stein’s Unbiased Risk Estimate (SURE): quadrature loss function. We get an estimate of the risk for a particular threshold value \( t \). Minimizing the risks in \( t \) gives a selection of the threshold value.

- **Sqtwolog** – Fixed form threshold yielding minimax performance multiplied by a small factor proportional to \( \log (\text{length}(s)) \). It is usually equal to \( \sqrt{2 \cdot \log (\text{length}(s))} \)

- **Heursure** – Threshold is selected using a mixture of first two methods. As a result, if the signal-to-noise ratio is very small, the SURE estimate is very noisy. So if such a situation is detected, the fixed form threshold is used.

- **Minimaxi** – Selected using the minimax principle. Uses a fixed threshold chosen to yield minimax performance for mean square error against an ideal procedure. The minimax principle is used in statistics to design estimators. Since the de-noised signal can be assimilated to the estimator of the unknown regression function, the minimax estimator is
the option that realizes the minimum, over a given set of functions, of the maximum mean square error.

Because $y$ is a standard Gaussian white noise, we expect that each method kills roughly all the coefficients and returns the result $f(x) = 0$. For Stein's Unbiased Risk Estimate and minimax thresholds, roughly 3% of coefficients are saved. For other selection rules, all the coefficients are set to 0.

We know that the detail coefficients vector is the superposition of the coefficients of $f$ and the coefficients of $e$, and that the decomposition of $e$ leads to detail coefficients, which are standard Gaussian white noises. So minimax and SURE threshold selection rules are more conservative and would be more convenient when small details of function $f$ lie near the noise range.

Level dependent thresholds $T(j)$ can be defined by $a \ast \max(|d(j)|)$

Where $a$ is a sparsity parameter $0.2 \leq a \leq 1$, typically $a=0.6$ by default and $d(j)$ is the detail coefficients at $j^{th}$ level of decomposition.

3.2.3 Reconstruction

Finally, we must perform a multilevel one-dimensional wavelet reconstruction using either a specific wavelet or specific reconstruction filters $Lo_R$ and $Hi_R$. Reconstruction is the inverse function of decomposition.
Since the thresholded details and approximations are given to this reconstructing stage as inputs, we get the signal with the noise eliminated.

Thus, signal denoising is achieved using wavelet coefficients.
4.1 Introduction to Signal Classification

Soft Computing is a heuristic methodology which has attracted significant interest in recent areas and is successful in many areas such as pattern recognition, fault diagnosis, modeling and control. It is based on the implementation of different approaches such as fuzzy logic, neural networks, genetic algorithms and others. Each of these techniques is suited for solving specific types of problems. [13] In this respect,

*Fuzzy logic* is efficient for approximate modeling and reasoning.

*Neural networks* are well suited for learning based adaptation.

*Genetic algorithms* are powerful for evolutionary based optimization.

These techniques are used in combination rather than using each of them separately to build artificially intelligent systems using wavelet analysis. The reason for using the term ‘intelligent’ was the similarity with some heuristic capabilities of the intelligent human beings, e.g., approximate reasoning, self learning, classifying etc. The latest development in research involves making artificial systems for different actions like logical analysis, diagnosis, decision making etc and in this category comes the interesting field “Pattern recognition and classification“.
The representation of patterns is feature oriented. The process of pattern classification is divided into several modules in order to achieve maximum flexibility without loss of any functionality.

1. Image processing module
2. Feature extraction module
3. Feature adaptation module
4. Classification module

The main functionality of each module depends on the type of input we are applying to the system. Here is a description of some of the intelligent systems in which our work can be employed in the feature extraction stage.

**System 1: Tool Wear Classifier**

Cutting tool condition is a major factor relating to the state of a machine tool. Monitoring tool condition by using an integrated system composed of multi-sensors, signal processing devices and intelligent decision making plans is a necessary requirement for modern automatic manufacturing processes. A unique fuzzy driven neural network based pattern recognition algorithm has been developed in [14]. It can fuse the information from multiple sensors and has strong learning and noise suppression ability. This leads to successful tool wear classification under a range of machining conditions. This can ensure machining accuracy and reduce the production costs. Coupling various
transducers with intelligent data processing techniques to deliver improved information relating to tool condition makes optimization and control of the machining process possible. [14]

Most existing tool wear sensing methods can be classified into two major categories. In direct sensing, optical, radioactive and distance transducers are used to measure the actual tool wear directly. For indirect sensing, some parameters correlated with tool wear are measured by using transducers, e.g., force, vibration, power, temperature and roughness. In the milling process, the cutting operation is intermittent and the cutting tool is always rotating. Hence, the signals are noisy and it is not convenient to mount sensors on the tool. So, sensors must be chosen accordingly before using the recognition system.

The tool wear monitoring system is composed of four kinds of sensors; signal amplifying and collecting devices and the microcomputer. Features extracted from the time domain and frequency domain for pattern recognition are as follows:

1. Power consumption signal : mean value
2. Acoustic emission – RMS signal(root mean square signal) : mean value, skew and kurtosis
3. Cutting force and vibration: mean value, standard deviation and the mean power in 10 frequency ranges.

Most features are found relevant with the development of tool wear values. Tool condition monitoring is a pattern recognition process in which the characteristics of the
tool to be monitored are compared with those of the standard models. The process is composed of the following parts: determination of the membership functions of signal features, calculation of fuzzy distances, learning and tool wear classification.[14]

In the practical tool condition monitoring process, the tool with unknown wear value is the object and it will be recognized as “new tool”, “normal tool” or “worn tool”. The approaching degrees between the corresponding features of the object and different models can be the inquiry input of the artificial neural network.

Thus, the combination of neural networks and fuzzy logic techniques integrates the strong learning and classification ability of the former and the superb flexibility of the latter to express the distribution characteristics of signal features with vague boundaries and the fuzzy distances between them. This methodology indirectly solves the weight assignment problem of the conventional fuzzy pattern recognition system and lets it have greater representative power and be more robust.

**System 2: Handwritten digits classifier [12]**

In this case, the block diagram can be explained as below.

![Block Diagram of Signal Classifier](image)

**Figure 4.1: Block Diagram of Signal Classifier from [12]**
1. The first module is the image preprocessing module. Its input is an arbitrary sized image bitmap and this module is responsible for separating one digit from another, scaling the input image for each digit to a 32x32 image bitmap, reducing noise and smoothing the edges.

2. Feature extraction module consists of one or more filters. Filters are routines that extract relevant features from the image.

3. Then the feature adaptation module transforms the data generated by the feature extraction module into data that are more suitable for the specific recognition module.

4. The recognition module is the core of the whole process, recognizing digits (0 to 9) from the adapted features. It outputs a set of 10 probabilities \( P_d \), \( d=0, 1... 9 \), where \( P_d \) is the probability that the input image corresponds to the digit \( d \).

The results vary for different kinds of recognition modules employed.

**System 3: Fingerprints recognition [15]**

Between 1924 and today, the US Federal Bureau of Investigation has collected about 30 million sets of fingerprints. Let's put the data storage problem in perspective. Fingerprint images are digitized at a resolution of 500 pixels per inch with 256 levels of gray-scale information per pixel. A single fingerprint is about 700,000 pixels and needs about 0.6 Mbytes to store. A pair of hands, then, requires about 6 Mbytes of storage. So digitizing
the FBI's current archive would result in about 200 terabytes of data. Obviously, data compression is important to bring these numbers down. The following figure shows an FBI-digitized left thumb fingerprint. The image on the left is the original; the one on the right is reconstructed from a 26:1 compression using the wavelet transforms.

![Original and Reconstructed Fingerprints](image)

Figure 4.2: Original and Reconstructed Fingerprints from [15]

**System 4: Mammograms and Electrocardiographs**

There is a great work going on to recognize microcalcifications in mammograms and heart diseases in electrocardiographs. These graphs can be taken as input and classified into different classes according to the degree of seriousness of the problem. Similar cases can be grouped together and treatment can be given at a time. Even these signals need wavelets for their representation.[16]
Microcalcifications are small deposits of calcium in the breast tissue. They appear in mammograms as small spots which are brighter than the background. In many cases, they are not quite visible because of small size and low contrast. The variability of the objects in terms of size and shape requires flexibility of the detection process. For this, they use fuzzy detection.

Fuzzy clustering is commonly applied for classification of such data. Clustering is the process of grouping similar data. This can be applied to perform segmentation of an image. The pixels can be clustered with respect to attributes such as color and location, and the resulting clusters will represent the segments of the image. Segments can be clustered further with other segments to perform a higher level interpretation of the image, as explained in [16] Thus the microcalcifications are intelligently detected.

**System 5: Speech or Odor Signals**

We can take the speech signals and classify using the pitch, frequency and acoustics as the features for extraction. Also, different odors can be classified using Vapor prints which are based on the concentration of different components of gases present in an odor.
4.2 ECG Characteristics

An electrocardiogram (ECG) is a recording of the electrical activity of the heart in dependence on time. The mechanical activity of the heart is linked with its electrical activity. An ECG is therefore an important diagnostic tool for assessing heart function. An ECG as an electrical manifestation of a human activity is composed of heartbeats that repeat periodically. In each heart beat several waves and interwave sections can be recognized. The shape and length of these waves and interwave sections characterize cardiovascular diseases, arrhythmia, ischemia and further heart diseases. Basic waves in ECG are denoted by P, Q, R, S, T, U, as shown in the figure. From these, the denotation (and length) of the intervals and segments is derived. The time axis uses the order of milliseconds, while the potential axis uses the order of mV.

![Figure 4.3: Representative Signal Cycle of a Normal ECG](image)

There exist classification systems that are able to localize pathological changes in ECG records, detect the QRS complex and the ST segment, find the RR interval, the appearance of individual waves, and many other values. The figure below illustrates four examples of normal and pathological ECG signals. An ECG of a healthy heart is regular
and may vary slightly in frequency, amplitude and length of the individual parts of the P-QRS-T complex.

Figure 4.4: Examples of Normal and Pathological ECG signals from [17]

When interpreting an ECG, physicians first locate the P waves, QRS complexes, T complexes and U waves. Then they interpret the shapes (morphology) of these waves and complexes; in addition they calculate the heights and the interval of each wave, such as the RR interval, PP interval, PR interval, QT interval, and ST segment. From the technical point of view, the assumption for ECG analysis is the existence of perfect ECG signals (i.e. signals with sufficient dynamics and a minimum of artefacts).
It follows from the physicians' approach that three related problems are usually distinguished and must be considered when designing a system for ECG analysis and classification:

**Recognition of several primary characteristic ECG components:** In a selected ECG segment, the elements are divided into two groups, namely those belonging to isolines and those representing waves, complexes and other graph elements whose specific diagnostic significance is known or supposed.

**Graph element quantification:** The curvature of the arcs, and the length of the waves and complexes are calculated (or visually evaluated); their amplitudes are measured, etc. Measurement of individual cardio intervals is also important. In the simplest case, a certain number of characteristic points are determined on a single ECG period. The characteristic points are selected in such a way that they enable the co-ordinates to be determined for calculation of all important graph elements of the corresponding ECG segment. Undesirable effects such as breathing waves (which have period $T$ equal to 3 - 5 s) and slow waves must be sufficiently removed for successful separation of significant diagnostic ECG features. For automatic analysis, it is necessary to define the amplitudes of the individual graph elements with a maximum error of $\pm 5$ per cent. For time intervals an error of 10 per cent is usually acceptable. When analyzing the QRS complex and P and T waves the corresponding areas above and below the isoline are calculated and the velocity of the potential change in the QRS complex is determined.
Classification into certain diagnostic classes: This is performed on the basis of appropriately defined features.

Figure 4.5: ECG Temporal Analysis – Characteristic Points and Calculation of Areas Above and Below Isoline. [17]

The most frequently used (or cited) ECG classification method is neural network classification. It is followed by expert systems, machine learning methods, fuzzy systems and, more recently, the wavelet transform and genetic algorithms. Morlet used the wavelet transform to acquire contour maps that allow the localization of short, low energy transient signals, even within the QRS complex. Bortolan combined two techniques of pattern recognition, namely cluster analysis and neural networks, in the problem of diagnostic classification of 12-lead electrocardiograms. Wavelet analysis, neural networks or digital filtering may bring good results in direct detection and
classification of events. However, they do not express the relations between signals and
process states, even if these exist.

### 4.3 Applications of Wavelets to ECG

The wavelet transform (WT) is a new and promising method for time and frequency
signal analysis. A signal is decomposed into building blocks that are well represented in
time and frequency. In the search for significant features of the ECG signal we used
filtering of the signal using wavelet filtering based on the wavelet transform.

While the set of decomposition functions of the Fourier transform are the functions \( \sin (k \omega_0 t) \) and \( \cos (k \omega_0 t) \) only, the set of decomposition functions of the wavelet transform is
wider and different sets of decomposition functions are used. This property of the wavelet
transform allows us to select the most suitable set of decomposition functions according
to the specific signal and its properties, and to achieve optimal results for specific types
of signals. The coefficients can be calculated efficiently. The transformation can be
calculated in \( O (N \log (N)) \) steps in general, the same as the Fast Fourier Transform.
Many wavelet transforms can be calculated in \( O (n) \) steps.

Virtually all wavelet systems have these very general characteristics. Where the Fourier
transform maps a one-dimensional signal to a one-dimensional sequence of coefficients,
the wavelet transform maps it into two dimensional arrays of coefficients. This allows us
to localize the signal in both time and frequency. The concept of the wavelet transform is usually introduced by the resolution concept to define the effect of changing scale.

Dyadic WT is used for extracting ECG characteristic points. Our approach is based on the work of Mallat. The basic idea stems from the properties of the function derivatives that represent useful information about the analyzed signal in the time domain, i.e. the morphology of the signal. We can think of an ECG signal as a composition of basic elements that we show in the top row of figure below. In the following rows of the figure, we show the first derivatives of the original signal smoothed at different scales with a smoothing function $\theta$; we detect significant points of a signal by using zero-crossing points of the first derivatives of the signal smoothed with $\theta$.

Figure 4.6: Details of Artificial Signal Obtained by Filtering by Equivalent FIR Filter of Wavelet Transform. [17]
Extraction of Values for Further Classification

For the task of learning and classification it is important to select a set of significant parameters of the ECG signal that allows us to successfully learn and classify the data. Initial selection of the attributes was based on consultations with medical doctors.

Figure 4.7: ECG Record and its Significant Attributes. [17]
There are a number of algorithms that may be employed to classify unknown ECG or EEG signals. Basically, they can be divided into two groups. The first group is based on rules defined by human experts. Since there is no exact algorithm or gold standard specifying ECG signals of healthy persons and patients with different diagnoses, the rules are biased towards human expertise. The same situation arises when a single ECG recording is evaluated by several human experts. The second group utilizes various forms of learning, thus avoiding human biasing. The experiments showed that a knowledge base originally created from expert rules refined with generated rules gives better results than the original knowledge base and decision tree as separate systems. One of the most important aspects of the ECG and EEG classification systems is reliable analysis of ECG and EEG records respectively, which enables significant values to be identified on the measured signal. This analysis is a necessary condition for correct classification.
The wavelet transform has proven to be a good tool for ECG signal analysis. It achieves a sufficiently high level of reliability – about 80 per cent. It enables to detect required values of selected attributes in few steps which is important from the point of view of time required for processing. The extracted values are then used as input values for a classification system.

4.4 Input Data

We used the physionet database of biological signals as a source of ECG records, namely the MIT-BIH ECG Database. This database is freely accessible on the internet on http://www.physionet.org/ [18]. An excellent advantage of this data source is that it is widely used in experimental works on classification of ECG signals and biological signals in general. Each data file consists of time stamps and values recorded from two electrodes; the length of each record is 60 seconds. The data is available in the form of samples, annotations, and waveforms as shown in the attached figures. We have the time on x-axis and voltage on y-axis.

We can convert the signals into text and see the “samples“ as follows.

<table>
<thead>
<tr>
<th>Time (Sec)</th>
<th>ECG (mV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>-0.060</td>
</tr>
<tr>
<td>0.020</td>
<td>-0.060</td>
</tr>
<tr>
<td>0.040</td>
<td>-0.065</td>
</tr>
</tbody>
</table>
We can convert the signals to “annotations“ and view only those as follows.

```plaintext
0:00.000  0  N  0  0  0
1:00.000  6000  N  0  0  0
2:00.000  12000  N  0  0  0
3:00.000  18000  N  0  0  0
4:00.000  24000  N  0  0  0
5:00.000  30000  N  0  0  0
6:00.000  36000  N  0  0  0
7:00.000  42000  N  0  0  0
8:00.000  48000  N  0  0  0
9:00.000  54000  N  0  0  0
10:00.000  60000  N  0  0  0
11:00.000  66000  N  0  0  0
12:00.000  72000  N  0  0  0
13:00.000  78000  A  0  0  0
14:00.000  84000  A  0  0  0
15:00.000  90000  A  0  0  0
```
Analysis of ECG records is thus performed by the wavelet transform, which allows good localization of QRS complexes, P and T waves in time and amplitude. The average accuracy of detection of all events is above 87 per cent and this extracted data are used as inputs for learning methods.
Chapter 5

Results Of Denoising

5.1 Denoising with Wavelet Packets

Wavelet analysis corresponds to windowing frequency space in "octave" windows [19]. A natural extension is provided by allowing all dyadic windows in frequency space and adapted window choice. This sort of analysis is equivalent to wavelet packet analysis.

The actual fast wavelet packet analysis algorithms permit us to perform an adapted Fourier windowing directly in time domain by successive filtering of a function into different regions in frequency. The dual version of the window selection provides an adapted subband coding algorithm. The wavelet packet library is constructed by iterating the wavelet algorithm as shown in Figure 5.1.

These waveforms are mutually orthogonal. Moreover, each of them is orthogonal to all of its integer translates and dyadic rescaled versions. The full collection of these wavelet packets (including translates and rescaled versions) provides us with a library of "templates" or "notes" which are matched "efficiently" to signals for analysis and synthesis, Wavelet packet expansions correspond algorithmically to subband coding schemes and are numerically as fast as the Fast Fourier Transform (FFT). [20]
5.2 Algorithm

The following is the algorithm we implemented using wavelet packets and thresholding techniques. This is more accurate and provides additional options to the user to select the kind of thresholding and other parameters, than the direct function provided in MATLAB.

Step 1: The signal is decomposed at depth ‘lev’, with ‘wname’ wavelet packets, where
lev = level of decomposition of the signal

wname = type of wavelet selected.

The result is a wavelet packet tree.

Step 2: The threshold value is selected based on the methods explained in section 3.3.

Step 3: Now, these wavelet packet tree coefficients are individually thresholded based on the threshold value selected in second step. This step returns a new wavelet packet tree by coefficient thresholding. Here, we provided options to threshold both the approximations and the details or only the details. Also, we can decide whether to perform soft thresholding or hard thresholding.

Step 4: From this coefficient thresholded wavelet packet tree, the corresponding signal is reconstructed.

5.3 Experiments and Results

For our experiments, we considered four levels of decomposition, four kinds of wavelet families and four thresholding rules for every ECG signal. We worked on three ECG datasets taking 1024 samples at a time and adding a random noise of 10% and 20%. The noise is randomly generated and added linearly to the original ECG signal. The decomposition levels selected are level 1, level 2, level 3 and level 4. The four wavelet
families used are Daubechies, haar, symlet and coiflet. And the four thresholding rules are minimaxi, heursure, rigsure and sqtwolog. The algorithm is provided with an option for the user to select the noise level to be added and the number of samples to be selected from the input signal.

The experiments are done varying three parameters

1. Levels of decomposition of wavelet packet tree
2. Wavelet family
3. Thresholding method

The mean square errors calculated after conducting these experiments for the random noise added to the ECG signal are tabulated as follows. Less error indicates efficient denoising of the original signal.

These are the results obtained for the experiments conducted using symlet wavelet with soft thresholding using heuristic sure method and 20% random noise. The experiments for 10% noise yielded similar results. We can see that there is an optimum level of decomposition for every input data after which further decomposition will not show much improvement in denoising. The confidence intervals are calculated by running the algorithm thirty times adding random noise each time separately.
<table>
<thead>
<tr>
<th>Decomposition level</th>
<th>Error[5]</th>
<th>Mean</th>
<th>Std.dev</th>
<th>Confi.Interval(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>0.3122</td>
<td>0.0215</td>
<td>0.02</td>
<td>0.019-0.026</td>
</tr>
<tr>
<td>Level 2</td>
<td>0.2832</td>
<td>0.0209</td>
<td>0.018</td>
<td>0.017-0.021</td>
</tr>
<tr>
<td>Level 3</td>
<td>0.1436</td>
<td>0.0119</td>
<td>0.018</td>
<td>0.010-0.013</td>
</tr>
<tr>
<td>Level 4</td>
<td>0.1389</td>
<td>0.0117</td>
<td>0.018</td>
<td>0.010-0.012</td>
</tr>
</tbody>
</table>

Table 5.1: Denoising Versus Levels of Decomposition

![Tree Decomposition Diagram]
These are the results obtained at third level of decomposition. For the various experiments conducted, our algorithm works best with symlet wavelet.

![Wavelet Packet Trees](image)

**Table 5.2: Denoising Versus Wavelet Family Selected**

<table>
<thead>
<tr>
<th>Wavelet family</th>
<th>Error[5]</th>
<th>Mean</th>
<th>Std.dev</th>
<th>Confi.Interval(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haar</td>
<td>0.1244</td>
<td>0.0845</td>
<td>0.01</td>
<td>0.081-0.086</td>
</tr>
<tr>
<td>Daubechies</td>
<td>0.1876</td>
<td>0.0663</td>
<td>0.012</td>
<td>0.058-0.067</td>
</tr>
<tr>
<td>Symmlet</td>
<td>0.1779</td>
<td>0.0430</td>
<td>0.012</td>
<td>0.041-0.048</td>
</tr>
<tr>
<td>Coiflet</td>
<td>0.2943</td>
<td>0.1124</td>
<td>0.011</td>
<td>0.110-0.113</td>
</tr>
</tbody>
</table>
Since Soft thresholding thresholds all the values uniformly, we get a reconstructed signal with less mean square error.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error[5]</th>
<th>Mean</th>
<th>Stddev</th>
<th>Confi.interval(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft</td>
<td>0.0682</td>
<td>0.0212</td>
<td>0.01</td>
<td>0.02-0.024</td>
</tr>
<tr>
<td>Hard</td>
<td>0.0946</td>
<td>0.0434</td>
<td>0.012</td>
<td>0.042-0.0438</td>
</tr>
</tbody>
</table>

Table 5.3: Denoising Versus Different Methods of Thresholding

The minimaxi method is more conservative and more convenient since small details of the signal lie near the noise range. Hence, it gives best results.

<table>
<thead>
<tr>
<th>rule</th>
<th>Error[5]</th>
<th>Mean</th>
<th>Std. dev</th>
<th>Confi.interval(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimaxi</td>
<td>0.1956</td>
<td>0.0032</td>
<td>0.005</td>
<td>0.003-0.0035</td>
</tr>
<tr>
<td>Heursure</td>
<td>0.1759</td>
<td>0.0553</td>
<td>0.032</td>
<td>0.055-0.0555</td>
</tr>
<tr>
<td>Rigrsure</td>
<td>0.2016</td>
<td>0.0421</td>
<td>0.038</td>
<td>0.0419-0.0424</td>
</tr>
<tr>
<td>sqtwolog</td>
<td>0.1812</td>
<td>0.0417</td>
<td>0.038</td>
<td>0.0416-0.0422</td>
</tr>
</tbody>
</table>

Table 5.4: Denoising Versus Threshold Selection Rules
Figure 5.3: The Solid Line Shows Original Signal and the Dotted Line Shows its Noisy Version

Figure 5.4: Denoised Signal at the First Level of Decomposition
Figure 5.5: Denoised Signal at the Second Level of Decomposition

Figure 5.6: Denoised Signal at the Third Level of Decomposition
Figure 5.7: Denoised Signal at the Fourth Level of Decomposition

5.4 Conclusion

From the results obtained, we can show that the denoising of a signal depends on the optimum value of level of decomposition, suitable forms of wavelet family and thresholding techniques. This varies for different kinds of input signals.
Chapter 6

Vector Quantization

6.1 Introduction

Vector Quantization (VQ) is a processing technique that has been used extensively in speech and image compression. An extension of scalar quantization, VQ exploits the memory or correlation that exists between neighboring samples of a signal by quantizing them together rather than individually. [21]

A VQ is nothing more than an approximator. The idea is similar to that of "rounding-off" (say to the nearest integer). An example of a 1-dimensional VQ is shown below:

![Figure 6.1: One Dimensional Vector Quantization in [21]](image)

Here, every number less than -2 is approximated by -3. Every number between -2 and 0 is approximated by -1. Every number between 0 and 2 is approximated by +1. Every number greater than 2 is approximated by +3. Note that the approximate values are uniquely represented by 2 bits. This is a 1-dimensional, 2-bit VQ. It has a rate of 2 bits/dimension.
An example of a 2-dimensional VQ is shown below:

Figure 6.2: Two Dimensional Vector Quantization in [21]

Here, every pair of numbers falling in a particular region is approximated by a star associated with that region. Note that there are 16 regions and 16 stars -- each of which can be uniquely represented by 4 bits. Thus, this is a 2-dimensional, 4-bit VQ. Its rate is also 2 bits/dimension.

In the above two examples, the stars are called codevectors and the regions defined by the borders are called encoding regions. The set of all codevectors is called the codebook and the set of all encoding regions is called the partition of the space.
The VQ design problem can be stated as follows. Given a vector source with its statistical properties known, given a distortion measure and given the number of codevectors, find a codebook and a partition which result in the smallest average distortion.

We assume that there is a *training sequence* consisting of M source vectors:

\[ T = \{x_1, x_2, \ldots, x_M\}. \]

This training sequence can be obtained from some large database. For example, if the source is a speech signal, then the training sequence can be obtained by recording several long telephone conversations. M is assumed to be sufficiently large so that all the statistical properties of the source are captured by the training sequence. We assume that the source vectors are k-dimensional, e.g.,

\[ X_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,k}), \quad m = 1, 2, \ldots, M. \]

Let N be the number of codevectors and let \( C = \{c_1, c_2, \ldots, c_N\} \) represent the codebook. Each codevector is k-dimensional, e.g., \( c_n = (c_{n,1}, c_{n,2}, \ldots, c_{n,k}), \quad n = 1, 2, \ldots, N. \)

Let \( S_n \) be the encoding region associated with codevector \( C_n \) and let \( P = \{S_1, S_2, \ldots, S_N\} \) denote the partition of the space. If the source vector \( x_m \) is in the encoding region \( S_n \), then its approximation is given by \( Q(x_m) = c_n \), if \( x_m \in S_n \).

Assuming a squared-error distortion measure, the average distortion is given by:

\[ D_{\text{ave}} = \frac{1}{M_k} \sum_{m=1}^{M} \| x_m - Q(x_m) \|^2, \quad \text{where} \quad \| e \|^2 = e_1^2 + e_2^2 + \ldots + e_k^2. \]
The design problem can be succinctly stated as follows: Given T and N, find C and P such that $D_{\text{ave}}$ is minimized.

**Optimality Criteria**

If C and P are a solution to the above minimization problem, then it must satisfy the following two criteria.

**Nearest Neighbor Condition**

This condition says that the encoding region $S_n$ should consist of all vectors that are closer to $c_n$ than any of the other codevectors. For those vectors lying on the boundary, any tie-breaking procedure will do.

**Centroid Condition**

This condition says that the codevector $c_n$ should be average of all those training vectors that are in encoding region $S_n$. In implementation, one should ensure that at least one training vector belongs to each encoding region.

**6.2 LBG Design Algorithm [21]**

The LBG VQ design algorithm is an iterative algorithm which alternatively solves the
above two optimality criteria. The algorithm requires an initial codebook $C^{(0)}$. This initial codebook is obtained by the *splitting* method. In this method, an initial codevector is set as the average of the entire training sequence. This codevector is then split into two. The iterative algorithm is run with these two vectors as the initial codebook. The final two codevectors are splitted into four and the process is repeated until the desired number of codevectors is obtained.

The algorithm is summarized below.

1. Given $T$, fix $\varepsilon > 0$ to be a “small” number.

2. Let $N=1$ and $c_i^* = \frac{1}{M} \sum_{m=1}^{M} x_m$; Calculate $D_{ave}^* = \frac{1}{M} \sum_{m=1}^{M} \|x_m - c_i^*\|^2$

3. **Splitting:** For $i = 1, 2, \ldots, N$, set $c_i^{(0)} = (1 + \varepsilon)c_i^*$, $c_{N+i}^{(0)} = (1 - \varepsilon)c_i^*$

   Set $N=2N$.

4. **Iteration:** Let $D_{ave}^{(i)} = D_{ave}^*$. Set the iteration index $i = 0$.

   i. For $m = 1, 2, \ldots, M$, find the minimum value of $\|x_m - c_n^{(i)}\|^2$ over all $n = 1, 2, \ldots, N$. Let $n^*$ be the index which achieves the minimum. Set $Q(x_m) = c_n^{(i)}$

   ii. For $n = 1, 2, \ldots, N$, update the codevector

   $$c_n^{(i+1)} = \frac{\sum_{m} x_m}{\sum_{m} I_{Q(x_m) = c_n^{(i)}}}$$
iii. Set $i = i+1$.

iv. Calculate $D_{ave}^{(i)} = \frac{1}{M_k} \sum_{m=1}^{M} \| x_m - Q(x_m) \|^2$

v. If $(D_{ave}^{(i-1)} - D_{ave}^{(i)}) / D_{ave}^{(i-1)} > \varepsilon$, go back to Step (i).

vi. Set $D_{ave}^* = D_{ave}^{(i)}$. For $n=1,2,\ldots,N$, set $c_n^* = c_n^{(i)}$

as the final codevectors.

5. Repeat Steps 3 and 4 until the desired number of codevectors is obtained.

### 6.3 Application in Denoising

After decomposing into wavelet coefficients, they are quantized first and then thresholded. Due to this, the noise gets uniformly distributed and thus, can be easily eliminated.

Algorithm with Vector Quantization

Step 1: The original signal is taken as input and a random noise is added.

Step 2: This signal is decomposed at depth ‘lev’, with ‘wname’ wavelet packets, where

\[
\text{lev} = \text{level of decomposition of the signal} \\
\text{wname} = \text{type of wavelet selected.}
\]
The result is a wavelet packet tree.

Step 3: The threshold value is selected based on the methods explained in chapter 3.

Step 4: Now, these wavelet packet tree coefficients are quantized using LBG Algorithm to obtain a uniform distribution.

Step 5: These quantized wavelet packet tree coefficients are now thresholded based on the threshold value selected in above step. This step returns a new wavelet packet tree by coefficient thresholding. Here, we provided options to threshold both the approximations and the details or only the details. Also, we can decide whether to perform soft thresholding or hard thresholding.

Step 5: From this coefficient thresholded wavelet packet tree, the corresponding signal is reconstructed.

Step 6: The difference between the original signal and this reconstructed signal is noted and the mean error is calculated.

6.4 Experiments and Results

These experiments are conducted using symlet wavelet with soft thresholding using heuristic sure method. We can see that there is an optimum level of decomposition for
every input data after which further decomposition will not show much improvement in
denoising. The mean square error, standard deviation and confidence interval after
including quantization are represented by the MeanQ, std.devQ and Confi.intervalQ
respectively. These are calculated by running the algorithm thirty times, adding random
noise each time, separately. The noise levels chosen were 10% and 20%. The tables
shown below are for 20% noise. The results for 10% are similar.

<table>
<thead>
<tr>
<th>Level of Decomposition</th>
<th>Error[5] Mean</th>
<th>MeanQ</th>
<th>Std.devQ</th>
<th>Confi.IntervalQ(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>0.3122</td>
<td>0.0215</td>
<td>0.018</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.017-0.019</td>
</tr>
<tr>
<td>Level 2</td>
<td>0.2832</td>
<td>0.0209</td>
<td>0.013</td>
<td>0.018</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.012-0.016</td>
</tr>
<tr>
<td>Level 3</td>
<td>0.1436</td>
<td>0.0119</td>
<td>0.010</td>
<td>0.018</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.010-0.013</td>
</tr>
<tr>
<td>Level 4</td>
<td>0.1389</td>
<td>0.0117</td>
<td>0.010</td>
<td>0.017</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.010-0.012</td>
</tr>
</tbody>
</table>

Table 6.1: Mean Square Errors Versus Decomposition Levels.

These are obtained at third level of decomposition. For the various experiments
carried out, our algorithm works best with symlet wavelet.
<table>
<thead>
<tr>
<th>Wavelet family</th>
<th>Error[5]</th>
<th>Mean</th>
<th>MeanQ</th>
<th>Std.devQ</th>
<th>Confi.IntervalQ(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haar</td>
<td>0.1244</td>
<td>0.0845</td>
<td>0.062</td>
<td>0.01</td>
<td>0.061-0.066</td>
</tr>
<tr>
<td>Daubechies</td>
<td>0.1876</td>
<td>0.0663</td>
<td>0.051</td>
<td>0.012</td>
<td>0.050-0.056</td>
</tr>
<tr>
<td>Symmlet</td>
<td>0.1779</td>
<td>0.0430</td>
<td>0.039</td>
<td>0.012</td>
<td>0.037-0.040</td>
</tr>
<tr>
<td>Coiflet</td>
<td>0.2943</td>
<td>0.1124</td>
<td>0.089</td>
<td>0.011</td>
<td>0.087-0.090</td>
</tr>
</tbody>
</table>

Table 6.2: Mean Square Errors Versus Wavelet Family.

Since Soft thresholding thresholds all the values uniformly, we get a reconstructed signal with less mean square error.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error[5]</th>
<th>Mean</th>
<th>MeanQ</th>
<th>StddevQ</th>
<th>Confi.intervalQ(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft</td>
<td>0.0682</td>
<td>0.0212</td>
<td>0.018</td>
<td>0.01</td>
<td>0.017-0.019</td>
</tr>
<tr>
<td>Hard</td>
<td>0.0946</td>
<td>0.0434</td>
<td>0.037</td>
<td>0.012</td>
<td>0.036-0.039</td>
</tr>
</tbody>
</table>

Table 6.3: Mean Square Errors Versus Thresholding Methods.

For different thresholding rules, the results obtained are as follows. Again, we can observe that minimaxi method gives best results as explained in section 5.3.
<table>
<thead>
<tr>
<th>rule</th>
<th>Error[5]</th>
<th>Mean</th>
<th>MeanQ</th>
<th>StddevQ</th>
<th>Confri.intervalQ(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimaxi</td>
<td>0.1956</td>
<td>0.0032</td>
<td>0.0028</td>
<td>0.005</td>
<td>0.0026-0.0029</td>
</tr>
<tr>
<td>Heursure</td>
<td>0.1759</td>
<td>0.0553</td>
<td>0.0547</td>
<td>0.032</td>
<td>0.0542-0.0548</td>
</tr>
<tr>
<td>Rigrsure</td>
<td>0.2016</td>
<td>0.0421</td>
<td>0.0420</td>
<td>0.038</td>
<td>0.0419-0.0422</td>
</tr>
<tr>
<td>sqtwolog</td>
<td>0.1812</td>
<td>0.0417</td>
<td>0.0415</td>
<td>0.038</td>
<td>0.0414-0.0417</td>
</tr>
</tbody>
</table>

Table 6.4: Mean Square Error Versus Thresholding Rules.

Thus, we can see a drastic improvement in the denoising after including quantization before thresholding the wavelet coefficients.
Chapter 7

Improvement in Hybrid Intelligent System

7.1 Working of Genetic Neural Signal Pattern Classifier (GNSPC)

GNSPC is a decision tree classifier with similarity classification rules. This is built with pure pattern data and tested against noisy data. The goal of GNSPC is to be able to differentiate among large number of classes, with high performance and minimum classification cost. GNSPC combines genetic algorithms, neural networks and fuzzy logic for finding the rule patterns and the similarity measures. GNSPC is basically a binary decision tree. Each node in the tree has a similarity rule SR while each leaf has a class C associated to it. Before building the tree, each signal pattern SP, in the training set is preprocessed to extract its important features. Starting with a root node, the algorithm recursively adds nodes to the tree while dividing the SP set to pass and fail subsets.

Figure 7.1: A Sample Classification Tree Mentioned in [5]
Figure 7.2: Flowchart of Building the Classification Tree as in [5].

**Feature extraction**

Feature points include local optima and other locations where the SP derivative exhibits a significant change. Points with the wavelet coefficients greater than threshold features are
constructed as feature points (spikes of the signals). The GNSPC feature has four parameters: start location, end location, amplitude and area.

**Pattern Similarity Rule**

If (Sr, P>t) visit pass child node, else visit fail child node. Genetic algorithm generates the similarity rule’s feature pattern and evolves the best rules while the neural network finds the similarity cutoff for the rule and evaluates the rule fitness. All the features are encoded as real numbers. Five features are coded as 21 real numbers, i.e., number of features * (number of feature attributes + cutoff value).

**Finding the Similarity Cutoff**

For all SP in the rule set, the similarity value is computed. The gaps between similarity values are sorted by the widest. For the widest K gaps, the fitness of the middle point of the gap is computed. And, the middle point with the highest fitness value is picked as the similarity cutoff.

**7.2 Inclusion of Wavelets in Feature Extraction Stage**

The Classifier takes the input signal from the training set. Before this is applied to the decision tree, we are incorporating this denoising mechanism on the input signals so that the signal strength increases and the main feature points can be easily extracted.
7.3 System Performance

The performance of the GNSPC system is observed with and without our denoising system included.

7.3.1 Original Classifier

Without our denoising system, the classifier gives a mean probability of 0.9. This means, that 90% of the original input signals are classified correctly.

7.3.2 Classifier with Denoising

After adding the denoising system at the feature extraction stage, for 1024 samples of ECG signal, the classification probability has increased to a mean value of 0.91. We considered only first 1024 samples because, the signal is almost periodic and classification of the first set implies the classification of the entire signal. After selecting the first 1024 samples, a randomly generated white Gaussian noise is added to these samples and the output errors are noted by running the same algorithm 30 times before calculating the mean square errors, standard deviation and confidence interval. After this, the denoised signal is given as input to the classifier and the output of the classifier is noted. An improvement of the classifier performance is recorded.
7.3.3 Classifier with Quantization and Denoising

There is a tremendous improvement in the performance of the hybrid intelligent system in classifying ECG signals after introducing the concept of vector quantization. The mean percentage of classification increased to 0.925. These are the results obtained after giving the quantized and thresholded input samples to the classifier system. The error remarkably decreases after quantizing the signals. The confidence intervals are calculated as explained in section 5.3. The results are recorded as follows.

<table>
<thead>
<tr>
<th>System</th>
<th>Mean</th>
<th>Stddev</th>
<th>Confi.interval(95%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifier</td>
<td>0.90</td>
<td>0.005</td>
<td>0.90-0.905</td>
</tr>
<tr>
<td>Classifier+denoiser</td>
<td>0.91</td>
<td>0.012</td>
<td>0.905-0.912</td>
</tr>
<tr>
<td>Classifier+denoiser+Quantizer</td>
<td>0.925</td>
<td>0.018</td>
<td>0.920-0.926</td>
</tr>
</tbody>
</table>

Table 7.1: Classifier Performance for Different Algorithms.
Chapter 8
Conclusions and Future Work

8.1 Conclusions

The important conclusions of this thesis are as follows.

8.1.1 Wavelet Study

As the first step, different types of wavelets are studied thoroughly and the properties of these wavelets are explained. This enables us to select a suitable wavelet for any application. This adds details to the documentation available in Matlab, which will enable users of the Matlab toolbox to work more productively.

8.1.2 Denoising ECG

Secondly, the improvement in the signal to noise ratio of the ECG signals is achieved, through which any kind of further processing of these signals becomes more accurate. Unless the noise and its effects are eliminated, we cannot be sure of the signal processing. Therefore, the denoising importance is emphasized, and the process is implemented using thresholding techniques.
8.1.3 Denoising with Quantization

After denoising the signal using simple thresholding techniques, another concept called quantization is applied to it. The improvement in denoising and in the speed of the computation is observed.

8.1.4 Performance Improvement in Hybrid Intelligent Systems

By applying this system to the classifier before the feature extraction stage, it is clearly observed that the classifier performance improved for ECG data. This shows the effect of denoising and quantizing the signals.

8.2 Future Work

The future work is possible mainly in three directions as described below.

8.2.1 LPC Analysis [22]

Linear Predictive Coding (LPC) is a method of digitally encoding analog signals, which uses a single-level or multilevel sampling system in which the value of the signal at each sample time is predicted to be a linear function of the past values of the quantized signal. LPC is related to Adaptive Predictive Coding (APC) in that both use adaptive predictors.
However, LPC uses more prediction coefficients to permit use of a lower information bit rate than APC, and thus requires a more complex processor.

LPC is another method of separating out the effects of source and filter from the signal, similar in intention to spectral analysis but using quite different methods. One way of thinking about LPC is as a coding method -- a way of encoding the information in a signal into a smaller space for transmission over a restricted channel. LPC encodes a signal by finding a set of weights on earlier signal values that can predict the next signal value:


If values for a [1..3] can be found such that e [n] is very small for a stretch of signal (say one analysis window), then we can transmit only a [1..3] instead of the signal values in the window. The frame can be reconstructed at the other end by using a default e[n] signal and predicting subsequent values from earlier ones. Clearly this relies on being able to find these values of a[1..k] but there are a couple of algorithms which can do this (one is covered in [22]). The result of LPC analysis then is a set of coefficients a[1..k] and an error signal e[n]. This error signal will be as small as possible and represents the difference between the predicted signal and the original.

There is an obvious parallel relationship between the LPC equation and that of a recursive filter (\( y*a = x \)):

where the terms are rearranged. The LPC coefficients correspond to those of a recursive filter and the error signal corresponds to a source signal. Moreover, the conditions under which the error signal is minimized in LPC analysis mean that the error signal will have a flat spectrum and hence that the error signal will approximate either an impulse train or a white noise signal. So, LPC analysis has the wonderful property of finding the coefficients of a filter. For example, this method can be used to convert either noise or an impulse train into the original frame of speech for speech recognition. [22]

In [22] the LPC coefficients are used to make up a model of the vocal tract shape that produced the original speech signal. A spectrum generated from these coefficients would show us the properties of the vocal tract shape without the interference of the source spectrum. We know that we can take the spectrum of the filter in various ways, for example by passing an impulse through the filter and taking its DFT, or by substituting for $z=\exp(i\omega_k)$ in the $z$ transform of the signal. Either way, the result can be quite useful in signal analysis.

Looking at an LPC smoothed spectrum of voiced speech we can clearly see the formant peaks; they tend to be much more well defined than in a spectrally smoothed spectrum. We can use the $z$-transform notation to find the locations of these formant peaks for a given set of LPC coefficients, corresponding to the points at which $A(z)$ is zero. This is the key to automatic formant tracking of speech signals -- derive the LPC coefficients, solve the $z$-transform equation and record the resulting formant positions. Unfortunately since the LPC model isn't a perfect fit to real speech production (it assumes a lossless, all
pole model, for example) this method will derive spurious formants; most of the work in a good formant tracking program is working out which of the candidate formants is the real thing.

The poor and synthetic quality of speech vocoders has led to what is known as the residual excitation approach to speech coding, a concept whereby not all the speech is synthesized, but a small part is transmitted as a coded waveform part of the original envelop, hence the name hybrid. The penalty is the higher bit rate of transmission required, but a very much improved speech quality is realized.

Vector Sum Excited Linear Predictive (VSELP) coding is the same as regular pulse excited speech coding but the details of the excitation analysis and bit assignments differ. With a source rate of 7.95kbps, the North American code is more efficient than the 13kbps code of GSM. On the other hand, the signal processing hardware of the coder is more complex. [22]

In the system we are working on, LPC analysis could be applied after vector quantization to give similar improvements for our applications to those obtained for speech recognition in [22].
8.2.2 Hardware Implementation

The system that has been developed to classify ECG signals is in software using MATLAB. Today, the importance of developing the large scale integrated chips is growing tremendously. Hence, we can implement the whole system in hardware using Verilog or VHDL. We can construct each block of the denoising algorithm in hardware by transforming the signal thresholding stage and quantizing stage into hardware. Also, the main hybrid classifier system can be realized in hardware by converting the codes for genetic algorithm, neural networks and fuzzy logic into hardware architectures. Due to this, the system portability increases and the cost of manufacturing decreases drastically. Thus, this can be applied to a wider range of applications.

8.2.3 Extending to Other Data

The same concept can be extended to other datasets, as mentioned in chapter 2. We can classify the mechanical tool wear and decide whether the tool is worn out, used or new. This can make the manufacturing processes simple and avoid huge damages. Also, this system can be used to classify fingerprints, which can be applied to identify a person and authenticate. Other than ECG signals, we have Electroencephalographs (EEG) and Electromyographs (EMG). These are the signals that demonstrate the activities of brain and body muscles respectively. So far Fourier transforms are used in the processing of these signals. Wavelets show a great improvement in ECG because of the signal characteristics, spikes and short discontinuities for which wavelets are most suitable. But,
the same can be applied to other kinds of signals too. Similarly, mammograms can be studied using wavelet analysis because a very fine resolution is required to identify the microcalcifications from the original spectrum. Multiresolution analysis of wavelets can be applied to process these kinds of signals. [16]

Finally, another application is deciding the strength of bridge based on the vehicle movements over it. We can collect data of the bridge beam pressures and tensions at various frequencies of the vehicles running on it. These can be processed using our system to decide the vulnerability of the bridge and take appropriate measures for its improvement.
Bibliography


Appendix

This code can be executed along with the code mentioned in [5] before giving input of the ECG samples to the classifier system. The inputs are the signal $s$, the level of decomposition $lev$ and wavelet family $wname$. The outputs obtained are a denoised version of the noisy version of the input signal and the mean value of the difference between the two signals. This code is executed in Matlab version 6.5.

% code for denoising using wavelet packets

% function [xd] = sden(x)

h = daubcqf(6);

[s,N] = makesig('Gaussian');

n = randn(1,N);

x = s + n/10;    % (approximately 10dB SNR)

plot(x,':');hold on;% plot(s,'-');

%Denoise x with the default method based on the DWT

% [xd,xn,opt1] = denoise(x,h);

% figure;plot(xd);hold on;plot(s,'g');

% 11111 original method

%Denoise x using the undecimated (LSI) wavelet transform

[yd,yn,opt2] = denoise(x,h,1);

% figure;plot(yd,'r');hold on;
% figure; plot(s,'k');
% hold on; % plot(x,'g');
err1=abs(norm(s)-norm(yd));
% 22222our method
lev= 3;

[xd]=wden(x,'heursure','s','sln',lev,'sym8');
% plot(xd,'b');hold on;
err2=abs(norm(s)-norm(xd));
% k=black=original signal
% g=green=noise added signal
% r=red=roshdy's function
% b=blue=our function
% 333333our second method

% The current extension mode is zero-padding (see dwtmode).
% Decompose x at depth 3 with db1 wavelet packets
% using Shannon entropy.
% threshold select
thr = thselect(x,'sqtwolog');

wpt = wpdec(x,2,'sym8','shannon');
% The result is the wavelet packet tree wpt.
% Plot wavelet packet tree (binary tree, or tree of order 2).
% plot(wpt);
% Read packet (2,1) coefficients.
% cfs = wpcoef(wpt,[2 1]);

% plot(cfs); title('Packet (2,1) coefficients');

% wpthcoef is a one-dimensional de-noising.

% NT = wpthcoef(T,KEEPAPP,SORH,THR) returns a new wavelet packet tree NT
% obtained from the wavelet packet tree T by coefficients thresholding.
% If KEEPAPP = 1, approximation coefficients are not thresholded;
% otherwise, they can be thresholded.
% If SORH = 's', soft thresholding; if SORH = 'h', hard thresholding
% THR is the threshold value.

w = wpthcoef(wpt,1,'h',thr);

% Reconstruct packet (2,1).

a = wprec(w);

% rcf = wprcoef(w,[1]);

figure; plot(a,'m'); % title('Reconstructed packet (2,1)');

% returns the reconstructed vector a corresponding to a wavelet packet tree wpt.
% wprec is the inverse function of wpdec in the sense that the abstract statement
% wprec(wpdec(X,wname')) would give back X

err3=abs(norm(s)-norm(a));
% code with quantization

% Name this m-file LBG.m

% u = training sequence

% Length of u must be a multiple of the dimension of quantizer

% oldcodebook = matrix whose rows are the codebook at the
% start of the iteration

% ----------------- random signal generation, adding noise

h = daubcqs(6);

[s,N] = makesig('Gaussian');

n = randn(1,N);

x = s + n/10;     % (approximately 10dB SNR)

figure; plot(s,'b');% hold on;

plot(x,'g');

% ----------------- threshold selection

thr = thselect(x,'minimaxi');

% ----- wavelet packet representation/decomposition of noisy signal

wpt = wpdec(x,2,'sym8','shannon');

% The result is the wavelet packet tree wpt.

% Plot wavelet packet tree (binary tree, or tree of order 2).

plot(wpt); hold on;

% Read packet (2,1) coefficients.
% cfs = wpcoef(wpt,[2 1]);

% plot(cfs); title('Packet (2,1) coefficients');

% wpthcoef is a one-dimensional de-noising.

% NT = wpthcoef(T,KEEPAPP,SORH,THR) returns a new wavelet packet tree NT
% obtained from the wavelet packet tree T by coefficients thresholding.
% If KEEPAPP = 1, approximation coefficients are not thresholded;
% otherwise, they can be thresholded.
% If SORH = 's', soft thresholding; if SORH = 'h', hard thresholding
% THR is the threshold value.

% Reconstruct packet (2,1).

% ----------- wavelet coefficients from the packet form
rcfs = wprcoef(wpt,[1]);

% rcfs = wprcoef(wpt,[2,1]);

% plot(rcfs); title('Reconstructed packet (2,1)');

% ---- quantizing these wavelet coefficients
% [c,l]=wavedec(x,2,'sym8');
% u=l;
% u=rcfs;u=x;

u=rcfs;

len=length(u);

% u
% len

% oldcodebook=avg(l);
% [wpt1,ca,cd] = wpsplt(wpt,[2 0]);
% %with cA = approximation and cD = detail of node N
%
% % oldcodebook= [ca,cd];
oldcodebook=rcfs;
%newcodebook = matrix whose rows are the codebook at the
%end of the iteration
% function newcodebook = LBG(oldcodebook,u)
p=oldcodebook;
% plot(oldcodebook,'k');
[N,k]=size(p);
%N is the size of the codebook
%k is the dimension of the quantizer
m=length(u)/k;
%m is number of blocks in training sequence
for i=1:m;
  datablock=u((i-1)*k+1:i*k);
  M(i,:)= datablock;
  dist=[ ];
  for j=1:N;
    v=p(j,:);
    S=sum((datablock-v).^2);
    dist=[dist S];
  end
end
end

[y, ind] = min(dist);
Q(i) = ind;
end

for j1 = 1:N;
    r = (Q == j1);
    if length(r) > 0
        p(j1, :) = r * M / sum(r);
    else
        end
    end
end

newcodebook = p;

% figure; plot(newcodebook, 'g');
l = newcodebook;

% ----------- wavelet packet with these quantized coefficients
wpt = wpdec(l, 2, 'sym8', 'shannon');

% --------- thresholding these quantized wavelet coefficients
w = wpthcoef(wpt, 1, 's', thr);
    % Reconstruct packet (2,1).
a = wprec(w);
    % rcf = wprcoef(w, [1]);
plot(a, 'm'); % title('Reconstructed packet (2,1)');
    % returns the reconstructed vector a corresponding to a wavelet packet tree wpt.
% wprec is the inverse function of wpdec in the sense that the abstract statement
% wprec(wpdec(X,wname')) would give back X

err3 = abs(norm(s) - norm(a));

err3

% a = waverec(p,l,'sym8');
% err = norm(x-a)
% plot(a,'r');