

September, 1992

LIDS-TH 2138

Research Supported By:

AFOSR grant F49620-92-J-0002
Draper CSR Corporate Sponsored Research Project

Wavelet Packet Based Transient Signal Classification

Learned, R.E.

September 1992

LIDS-TH-2138

WAVELET PACKET BASED TRANSIENT SIGNAL
CLASSIFICATION

by

Rachel E. Learned

This report is based on the unaltered thesis of Rachel E. Learned, submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering at the Massachusetts Institute of Technology in September 1992. This research was conducted at the Laboratory for Information and Decision Systems with support provided in part by the Air Force Office of Scientific Research under grant F49620-92-J-0002 and the Draper Laboratory CSR corporate sponsored research project C-38.

Massachusetts Institute of Technology
Laboratory for Information and Decision Systems
Cambridge, MA 02139

Wavelet Packet Based Transient Signal Classification

by

Rachel E. Learned

B.S., Tufts University (1990)

Submitted to the Department of Electrical Engineering and
Computer Science

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1992

© Rachel E. Learned, MCMXCII. All rights reserved.

The author hereby grants to MIT and to C. S. Draper Laboratory
permission to reproduce and to distribute copies
of this thesis document in whole or in part, and to grant others the
right to do so.

Author *Rachel Learned*

Department of Electrical Engineering and Computer Science

September 10, 1992

Certified by *Alan S. Willsky*

Alan S. Willsky

Professor of Electrical Engineering

Thesis Supervisor

Accepted by

Campbell L. Searle

Chair, Departmental Committee on Graduate Students

Wavelet Packet Based Transient Signal Classification

by

Rachel E. Learned

Submitted to the Department of Electrical Engineering and Computer Science
on September 10, 1992, in partial fulfillment of the
requirements for the degree of
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Abstract

This thesis explores the feasibility of applying the wavelet packet transform to the detection and classification of transient signals in background noise. A wavelet packet-based algorithm, specific to the detection and classification of underwater acoustic signals generated by snapping shrimp and sperm whale clicks, is proposed. The algorithm is based on the exploitation of differences in the wavelet packet transform coefficients specific to each signal class. The Charles Stark Draper Laboratory and the Naval Underwater Systems Center furnished an extensive collection of biologically generated acoustic signals in background noise which allowed for an empirical study of some typical occurrences of snapping shrimp and whale clicks. Testing of several proposed wavelet-packet-based transient signal detection/classification algorithms on various classes of biologically generated underwater acoustic signals showed excellent results.

Thesis Supervisor: Alan S. Willsky
Title: Professor of Electrical Engineering

Acknowledgements

First and foremost, I wish to thank my thesis supervisor and mentor Professor Alan Willsky. His insight, enthusiasm and encouragement proved to be most valuable. His contribution to the development of this thesis is more than greatly appreciated. I am also grateful for the ideas that Professor George Verghese readily offered.

I especially wish to thank my Draper supervisors Paul Rosenstrach and John Irza and their division leader John Furze for allowing me the freedom to pursue this topic. I am grateful to Draper Laboratory for the fellowship which supported my work and study at MIT. I would like to thank Makund Desai and Dov Shazeer for making the underwater recordings available for use in this thesis and Mark Dzwonczyk for allowing me the use of Pichhu and Neuralware. Other Draper colleagues deserved of my thanks include: Joe Scola for his help with formatting the data, Rami Mangoubi for his help translating a paper written in French, and Michael Le Blanc for his help with Neuralware.

I would like to thank Clem Karl for his help with various problems related to this work. I wish to thank Eric Miller and Ilya Polyak for the wavelet packet Matlab software and Dave Mackovjak for his Navy stories (some of which actually made it into this thesis). I am especially thankful for the comraderie of my friend Seema Jaggi.

This report is part of the Draper Laboratory CSR corporate sponsored research project C-38.

Publication of this report does not constitute approval by the Draper Laboratory or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

I hereby consign my copyright of this thesis to the Charles Stark Draper Laboratory, Inc. Cambridge, Massachusetts.

Rachel Learned

Permission is hereby granted by the Charles Stark Draper Laboratory, Inc. to the Massachusetts Institute of Technology to reproduce any or all of this thesis.

Contents

1	Introduction	10
1.1	Purpose	10
1.2	Motivation	10
1.3	Current Work In This Area	12
1.4	Thesis Outline	13
2	The Wavelet Packet Transform and its Energy Mapping	14
2.1	Wavelet Packet Decomposition	15
2.1.1	Paley Ordering of the Wavelet Packet Decomposition Tree . .	18
2.1.2	Indexing the Wavelet Packet Decomposition Tree	22
2.1.3	A few Examples of Wavelet Packet Decomposition	22
2.2	Energy Mapping of the Wavelet Packet Decomposition Tree.	26
3	Application of the Wavelet Packet Transform to the Data	29
3.1	Description of the Data	29
3.2	Wavelet Packet Transform of the Data	36
3.3	Energy Mapping of the Wavelet Packet Transform of the Data	38
4	Choice of an Optimum Reduced Parameter Feature Set	40
4.1	Vector Representation of the Energy Maps	40
4.2	Examination of 63 Bins of the Energy Maps	41
4.3	Compensating for the Noise	44
4.4	Reducing the Search for a Feature Set to Eleven Bins of the Energy Maps	47

4.5	Searching the Noise Normalized Energy Maps for a Reduced Feature Set	49
5	Decision Rules	54
5.1	A Statistical Approach	54
5.2	The Nearest Neighbor Rule	56
5.3	Neural Networks	57
6	Test Results	60
6.1	Test Results using Data Set I	62
6.2	Test Results using Data Set II	66
6.3	Test Results Using Data Set III	76
7	Conclusion and Future Work	79
7.1	Conclusion	79
7.2	Suggestions for Future Work	81
A	Caclulation of the Error Bounds on the Nearest Neighbor Rule	83
B	Matlab Code	87

List of Figures

2-1	The Daubechies 14-point filters.	16
2-2	F_0 and F_1	17
2-3	The fully decomposed wavelet packet tree for a signal of length eight.	17
2-4	An example of aliasing.	20
2-5	An example of aliasing. Another level down the tree.	21
2-6	The WPD tree displayed in Paley order.	22
2-7	The WPD tree with index label at each bin in the first four levels.	23
2-8	Color mapping used in the image displays.	23
2-9	The WPD of an impulse.	24
2-10	The WPD of two sinusoids.	25
2-11	The WPD of a time and frequency localized function.	27
2-12	Energy mapping of the WPD tree from Figure 2-3.	28
3-1	Plots of the transients of interest.	30
3-2	Plots of some data.	33
3-3	Short term Fourier transforms of some transients.	34
3-4	Short term Fourier transforms of the data.	35
3-5	Wavelet packet transforms of the data.	37
3-6	Energy maps of the data.	39
4-1	The first six levels of an energy map.	42
4-2	The 63-element singular vectors.	45
4-3	The 63-element noise normalized singular vectors.	48
4-4	Eleven bins of the energy map.	49

4-5	The 11-element noise normalized singular vectors.	50
4-6	Dominant bins of the energy maps of the transient signal data set. . .	51
4-7	Element 3 vs element 4 of the 11-element singular vectors.	52
4-8	Energy map showing the two bins that make up the feature set. . . .	52
4-9	Cluster plot of the two dominant bin energies.	53
6-1	Cluster plot of the two dominant bin energies for the test set of <i>data set I</i>	64
6-2	Short term Fourier transforms of both shrimp data.	68
6-3	Energy maps of both shrimp data.	69
6-4	Singular vectors (63 elements) for both shrimp data.	70
6-5	Singular vectors (11 elements) for both shrimp data.	70
6-6	Cluster plot of the two dominant bin energies for some other data. . .	71

List of Tables

3.1	The data sets.	32
4.1	Energy vector set.	41
4.2	Singular values for the $63 \times M_t$ element energy matrix.	44
6.1	Three data sets used for testing.	61
6.2	Results obtained from the nearest neighbor rule using <i>data set I</i>	63
6.3	Results obtained from the neural network using <i>data set I</i>	65
6.4	Results obtained from the nearest neighbor rule using <i>data set II</i>	73
6.5	Results obtained from the neural network using <i>data set II</i>	75
6.6	Results obtained from the nearest neighbor rule using <i>data set III</i> . . .	76
6.7	Results obtained from the neural network using <i>data set III</i>	78

Chapter 1

Introduction

1.1 Purpose

Signals possessing non-stationary information are not suited for detection and classification by traditional Fourier methods. An alternate means of analysis needs to be employed so that valuable time-frequency information is not lost. The wavelet packet transform [1] is one such time-frequency analysis tool. This thesis examines the applicability of using the wavelet packet transform in automatic transient signal classification through the development of various classification algorithms for biologically generated underwater acoustic transient signals in ocean noise.

1.2 Motivation

The ability to classify underwater acoustic signals is of great importance to the Navy. Today, detection and classification, tailored for stationary signals, is done by the sonar officer who monitors incoming signals and determines their origins. He does this by listening to the reception of incoming sounds while viewing computer generated displays such as time vs. angle-of-arrival and time vs. frequency. He tracks the signal of interest and observes the primary frequencies contained in the signal so that he can make an initial guess as to the source. In efforts to confirm or contradict his guess, he will, perhaps repeatedly, consult tables which contain the frequency information

on a large range of possible signals.

Transient signals, lasting only a fraction of a second, are of particular concern because they will typically appear as broadband energy on the frequency display. Thus, the sonar officer cannot rely on any visual displays to assist him in the classification process. He must be able to detect and classify transient signals by only listening for them. These brief signals, such as the single acoustic transmission due to the closing of a door within a ship, may be missed by the sonar officer. An automatic method of classification for transient signals would greatly aid in the detection/classification process.

The frequency display, using standard spectral analysis methods, is useful for the process of stationary signal classification; transient signals are not well matched to these methods. In particular, Fourier-based methods are ideally suited to the extraction of narrow band signals whose duration exceeds or is at least on the order of the Fourier analysis window length. That is, Fourier analysis, particularly the short-term Fourier transform (STFT), does an excellent job of *focusing* the information for sources of this type, thus, providing features (spectral amplitudes) perfectly suited to detection and discrimination. The STFT does allow for some temporal as well as frequency resolution, but it is not well suited for the analysis of many transient signals and, in particular, to the generation of features for detection and discrimination.

The STFT may be viewed as a uniform division of the time-frequency space. It is calculated for consecutive segments of time using a predetermined window length. The accuracy of the STFT for extracting localized time/frequency information is limited by the length of this window relative to the duration of the signal. If the window is long in comparison with the signal duration there will be time averaging of the spectral information in that window. On the other hand, the window must be long enough so that there is not excessive frequency distortion of the signal spectrum. The STFT with its non-varying window is not readily adaptable for capturing signal-specific characteristics. Additionally, all time resolution is lost within each window. We look to the wavelet packet transform for a bit more freedom in dealing with this time-frequency trade-off.

1.3 Current Work In This Area

Over the last decade much work has been done in applying time-frequency transforms to the problem of signal representation and classification. The recent development of wavelet theory by Grossman and Morlet [2] for application to seismic signal analysis has motivated a considerable amount of research in transient and non-stationary signal analysis.

Current work in the area of underwater acoustic transient classification using wavelet related concepts has been done by Nicolas [3] and, more recently, Desai and Shazeer [4]. Nicolas offers two methods of automatic identification with a primary focus on pattern recognition using two neural networks, an adaptive linear network and a multi-layered perceptron. The training feature set for the neural network was derived from autoregressive signal models, histograms, and node energies of the wavelet transforms of the training signal set. The wavelet transform is a specific case of wavelet packet decomposition. Nicolas uses the energies at all nodes of the wavelet transform down to the fifth level regardless of possible class-specific frequency characteristics. In addition to ignoring the redundancy between parent and children nodes of the transform, this method, by prohibiting signal-specific division of the time-frequency space, prevents the exploitation of any class dependent frequency variations. Successful classification ranged from 84% to 100% when the system was tested with lobster clicks, snapping shrimp, porpoise clicks, and whale clicks.

Desai and Shazeer also use two classifiers, a quadratic Bayesian classifier and a multi-layered perceptron, for the recognition of transient patterns. They employ the wavelet packet transform as a means of generating class-dependent features from four classes of underwater acoustic transients. Using the eight frequency channels represented by the eight nodes of the wavelet packet transform at the third iteration of decomposition, four features were derived from the normalized discrete time Fourier transform and curve length at each node. Again, exploitation of class dependent frequency characteristics are overlooked by choosing a uniform division of the frequency space. The method was tested using underwater recordings of the closing and open-

ing of two doors inside a ship. Perfect classification (100%) was obtained using both classifiers.

The above mentioned works do not pay particular attention to the choice of the wavelet packet decomposition that offers maximum feature separability due to class-specific characteristics. A logical expansion of their works would be to address the issue of finding a wavelet packet based feature set for best classification performance given the signal characteristics. This thesis explores the feasibility of applying the wavelet packet transform to the detection and classification of transient signals in background noise focusing on the exploitation of class-specific differences through careful examination of the feature separation attainable from the wavelet packet transform.

1.4 Thesis Outline

This thesis investigates the use of the wavelet packet transform in the detection and classification of transient signals in background noise. Chapter 2 summarizes wavelet packets, and establishes the energy mapping of the wavelet packet transform used in this thesis. The Charles Stark Draper Laboratory and the Naval Underwater Systems Center furnished an extensive collection of acoustic signals in background noise which allowed for an empirical study of some typical occurrences of snapping shrimp and whale clicks. These data are discussed in Chapter 3.

A wavelet-packet-based feature set specific to the classification of snapping shrimp and whale clicks is formulated in Chapter 4 by focusing on the exploitation of class-specific differences obtained through careful examination of the feature separation attainable from the wavelet packet decomposition of the transients. Chapter 5 discusses several detection algorithms used with this feature set. Test results are given in Chapter 6. Conclusions and future work are discussed in Chapter 7.

Chapter 2

The Wavelet Packet Transform and its Energy Mapping

The subject of wavelets is considered a major breakthrough in mathematical sciences and provides a common link between mathematicians and engineers. In the past ten years interest in wavelets has experienced explosive growth due to formal development by Morlet and Grossman [2] [5], Meyer [6], Daubechies [7], and Mallat [8]. Many of the basic ideas comprising wavelet theory are not new. The wavelet and wavelet packet transforms are a unification of ideas that originated during the last twenty or thirty years in fields ranging from engineering to pure mathematics. They are the formalized framework for spectrograms, octave-band filter banks, quadrature mirror filtering, and multiresolution analysis. Some current applications of this theory are in areas such as subband coding, image compression, time-frequency localization, decomposition and reconstruction, and information storage and retrieval. Because of their interdisciplinary origins, wavelets appeal to scientists and engineers of many different backgrounds.

Over the last decade much work has been done in applying time-frequency transforms to the problem of signal representation and classification. Most recently, the emergence of wavelet theory has motivated a considerable amount of research in transient and non-stationary signal analysis.

This chapter is not meant to be a tutorial on wavelet theory and wavelet packets. A

basic understanding of the discrete time wavelet transform is helpful but not necessary. Rioul and Vetterli [9], Mallat [8] and Strang [10] are all excellent references. In this chapter the wavelet packet transform is described as a general extension of the wavelet transform. The transformation of the wavelet packet decomposition tree into an energy map which is used for the remainder of this thesis is established.

2.1 Wavelet Packet Decomposition

The wavelet packet decomposition (WPD) of a signal can be viewed as a step by step transformation of the signal from the time domain to the frequency domain. The top level of the WPD is the time representation of the signal. As each level of the decomposition is calculated there is an increase in the trade-off between time and frequency resolution. The bottom level of a fully decomposed signal is a frequency representation.

This section presents the ideas developed by Wickerhauser [1] extending wavelet concepts to wavelet packets. Using Wickerhauser's notation, let $h(n)$ and $g(n)$ be the finite impulse response lowpass and highpass filters used for the decomposition. The Daubechies 14-point filters shown in Figure 2-1 are used for all the wavelet packet decompositions in this work. These filters are derived from the wavelet and scaling functions as discussed in [8].

Let \mathbf{x} be the vector having elements $x_n = x(n)$, where $x(n)$ is the original discrete-time sequence that we wish to decompose via the wavelet packet method. The original signal, $x(n)$, is of finite length, 2^η , where η is an integer. Let F_0 and F_1 be the operators which perform the convolution of $x(n)$ with $h(n)$ and $g(n)$, respectively, followed by a decimation by two. The wavelet decomposition may be calculated using a recursion of bandpass linear filtering and decimation as shown in Figure 2-2.¹

The full WPD can be displayed as a tree with a discrete sequence at every branch resulting from the processing of the original signal. The sequences at each level

¹The discrete convolution and decimation is done with one matrix multiplication of F_0 or F_1 . Each row of these matrices is simply the discrete sequence, $h(n)$ or $g(n)$, circularly shifted by a multiple of the necessary factor of two.

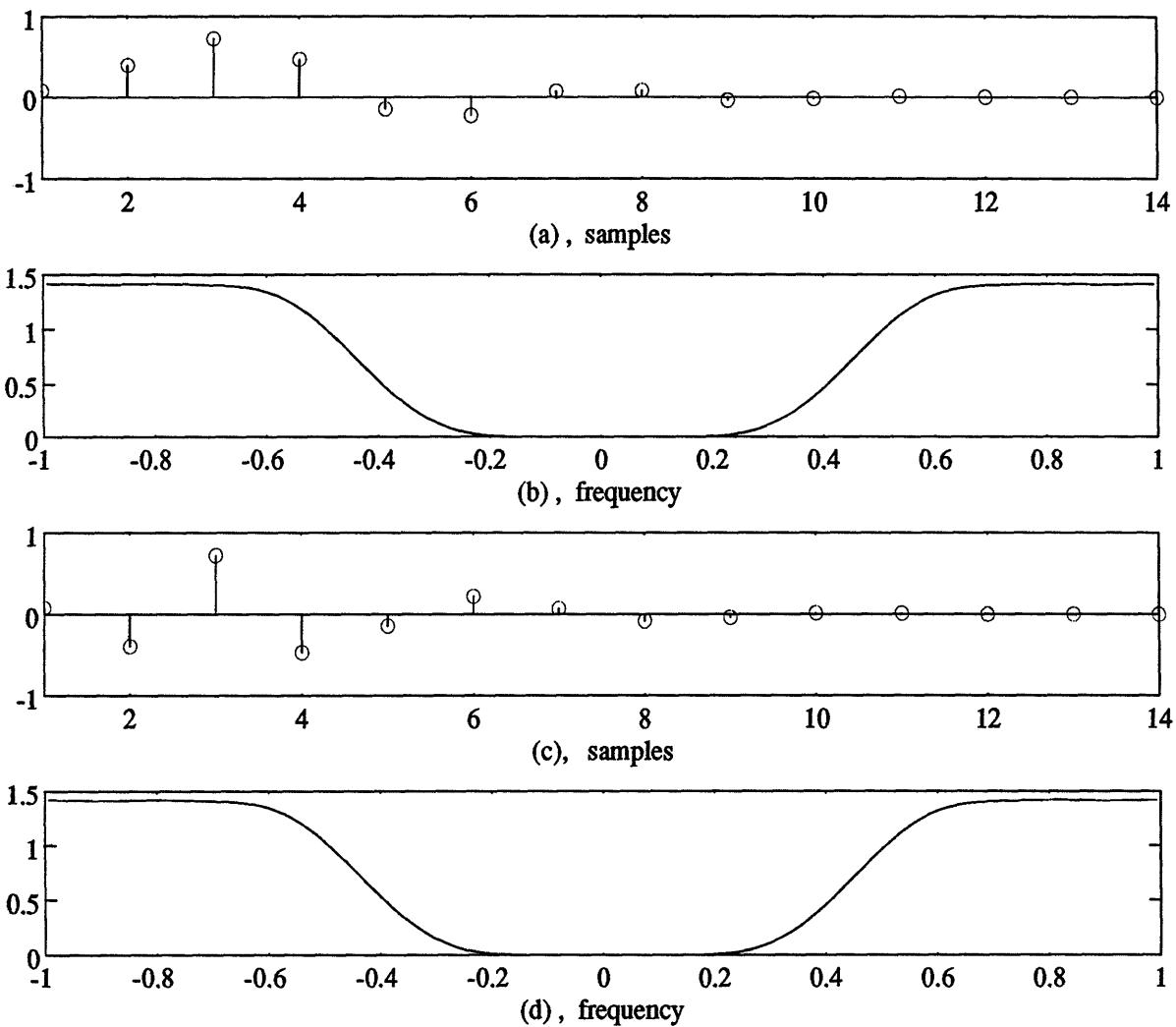
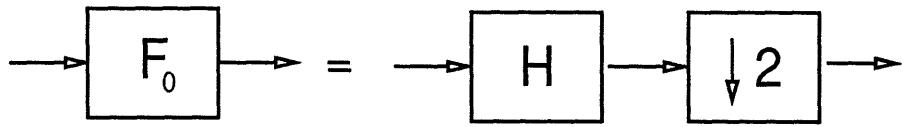
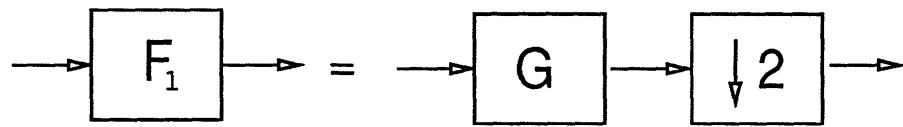


Figure 2-1: The Daubechies 14-point filters in time, (a) $h[n]$ and (c) $g[n]$, and the magnitude of their Fourier transforms, (b) $H(\omega)$ and (d) $G(\omega)$ with the horizontal axes being frequency in units of $\pi \times$ radians.



Lowpass filtering followed by a decimation.



Highpass filtering followed by a decimation.

Figure 2-2: The convolution and decimation steps in the WPD can be interpreted as a discrete time filtering and downsampling.

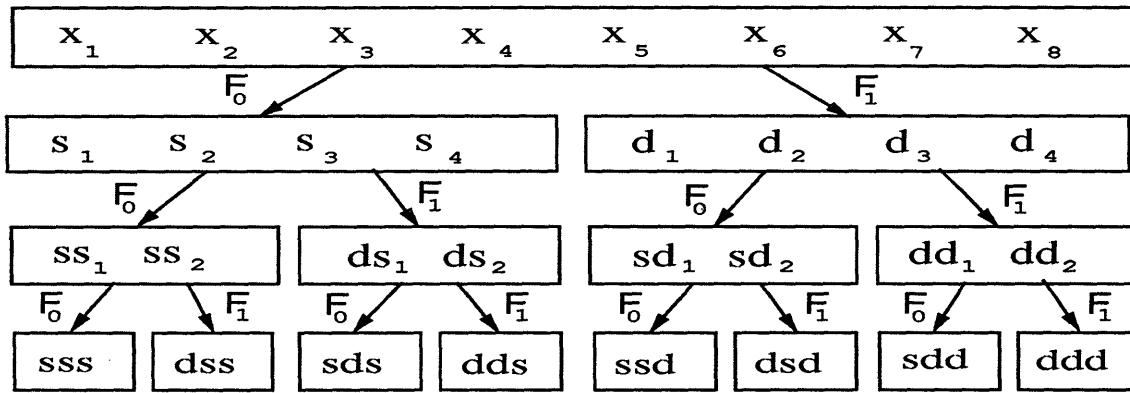


Figure 2-3: The fully decomposed wavelet packet tree for a signal of length eight.

are calculated by applying F_0 and F_1 to the sequences of the previous level. The sequences at each branch are the results of linear operations on the original sequence. The operation that results in any given sequence of the tree from the original sequence is equivalent to successive applications of the filter-decimation operations that fall in the path from the original sequence to that branch. Each sequence is referred to as a bin vector for the remainder of this thesis. The original discrete signal is at the first level. Figure 2-3 shows a WPD tree for a signal of length eight.

Wickerhauser uses the notation \mathbf{s} and \mathbf{d} to represent the sequences resulting from the applications of F_0 and F_1 to \mathbf{x} .

$$F_0 \mathbf{x} = \mathbf{s} \quad \& \quad F_1 \mathbf{x} = \mathbf{d}$$

In Figure 2-3, s_i and d_i are the i^{th} components of \mathbf{s} and \mathbf{d} . The \mathbf{s} and \mathbf{d} are used as prefixes for the bin vector symbols throughout the tree because the lowpass filter-decimation operation can be compared to a sum and the highpass filter-decimation can be compared to a difference. For example, at the third level of the tree, \mathbf{ss} , \mathbf{ds} , \mathbf{sd} , and \mathbf{dd} are the vectors resulting from the filter-decimation operations

$$F_0 \mathbf{s} = \mathbf{ss} \quad \& \quad F_1 \mathbf{s} = \mathbf{ds}$$

and

$$F_0 \mathbf{d} = \mathbf{sd} \quad \& \quad F_1 \mathbf{d} = \mathbf{dd}.$$

Due to the decimation, each bin vector contains half as many elements as its parent bin vector. The decomposition can be carried down to the final level where there is only one element in each bin vector.

2.1.1 Paley Ordering of the Wavelet Packet Decomposition Tree

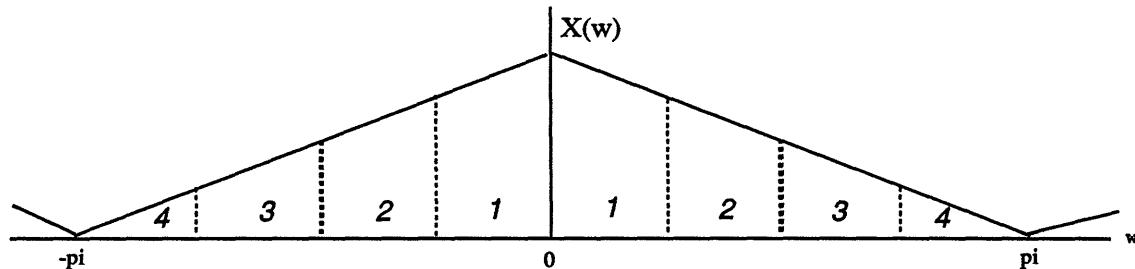
An intuitively pleasing way to view the wavelet packet decomposition tree is by displaying the bins at a given level so that they occur in increasing frequency order

from left to right. For example, the bin to the far left should represent the lowest frequency portion of the signal and the bin to the far right should represent the highest frequency portion of the signal. The method of decomposition described above, using successive lowpass and highpass filters followed by a downsampling, does not result in a WPD tree displayed in this intuitively pleasing manner. Aliasing occurs which exchanges the frequency ordering of some branches of the tree. A simple swapping of the appropriate branches results in this increasing frequency ordering which Wickerhauser refers to as the Paley ordering of the tree.

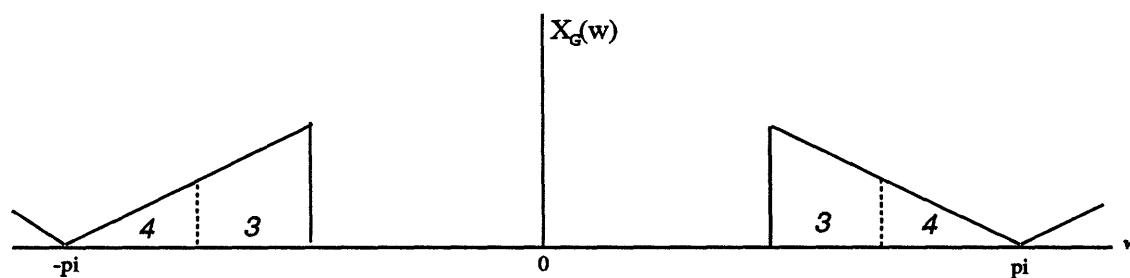
To illustrate the aliasing that occurs we track the wavelet packet decomposition of a discrete signal in the frequency domain. Let $X(\omega)$ be the Fourier transform of the signal we wish to decompose. Section the spectrum of the signal into four portions as shown in Figure 2-4. Let $X_G(\omega)$ be the result of the highpass filtering of $X(\omega)$. Consistent with Wickerhauser's notation for the bin vectors, denote the result of the downsampling of $X_G(\omega)$ by $D(\omega)$ also shown in the figure. Figure 2-4 shows that the sections of $D(\omega)$ labeled 3 and 4 are swapped in comparison with their original ordering in $X(\omega)$. Portion 3 is now of higher frequency than portion 4. This is a result of the aliasing which occurs when the signal is downsampled.

The next step in the decomposition would be to lowpass and highpass $D(\omega)$. The results are denoted by $D_H(\omega)$ and $D_G(\omega)$ and are shown in Figure 2-5. We continue by downsampling, denoting the results by $SD(\omega)$ and $DD(\omega)$, also shown in the figure. The ordering of the bins from left to right is not in intuitive frequency order. The bins at level three of the WPD tree, shown in Figure 2-3, are $SS(\omega)-DS(\omega)-SD(\omega)-DD(\omega)$ from left to right. As indicated by Figure 2-5, these bins hold sections 1–2–4–3 of $X(\omega)$.

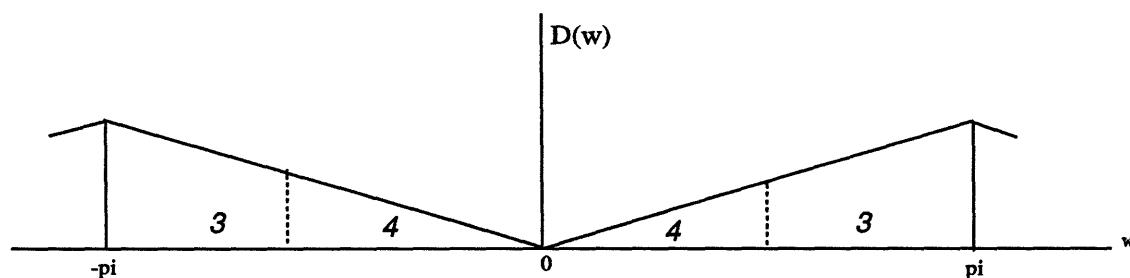
Only the first occurrence of aliasing is shown from the above example. Aliasing of this sort occurs in an easily identified pattern throughout the tree. This aliasing may be compensated for by a simple swapping of the affected bins, thus, re-ordering the branches to display the bins in frequency increasing order. A Paley ordering of the WPD tree is shown in Figure 2-6. The wavelet packet decomposition trees and other trees derived from the wavelet packet tree are displayed in the Paley order for



The Fourier transform of the signal to be decomposed.

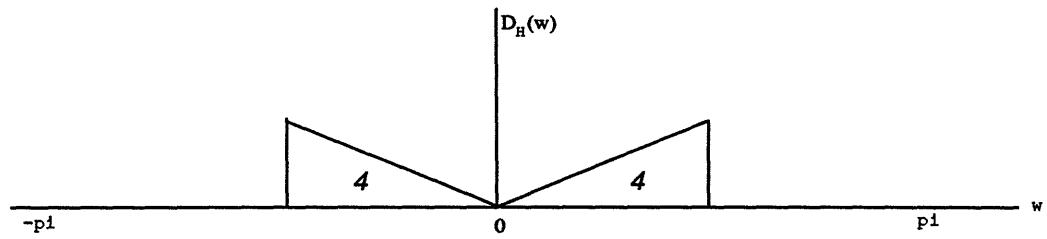


The result of highpass filtering.

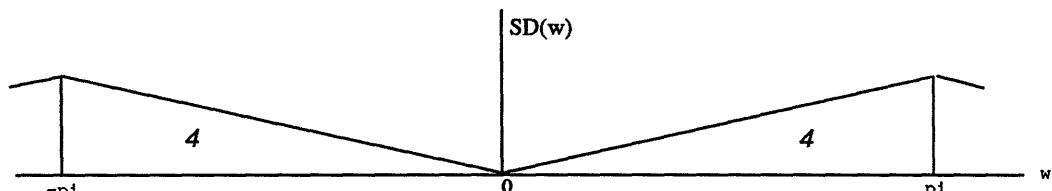


The result of downsampling.

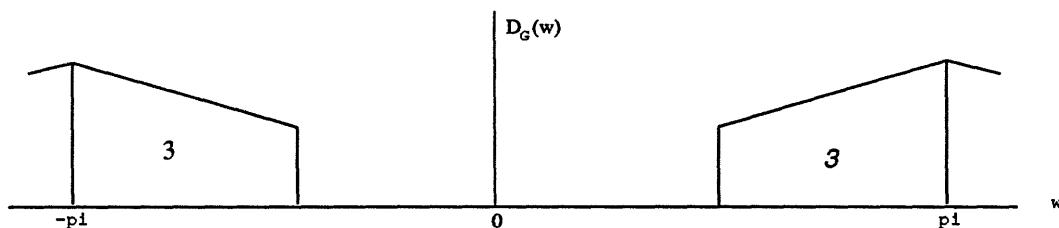
Figure 2-4: An example of aliasing while decomposing a signal via the wavelet packet transform. The signals are displayed in the frequency domain.



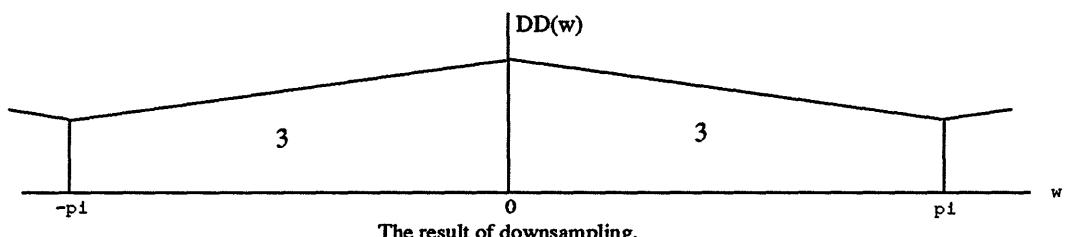
The result of lowpass filtering $D(w)$.



The result of downsampling.



The result of highpass filtering $D(w)$.



The result of downsampling.

Figure 2-5: The aliasing example continued another level down the tree.

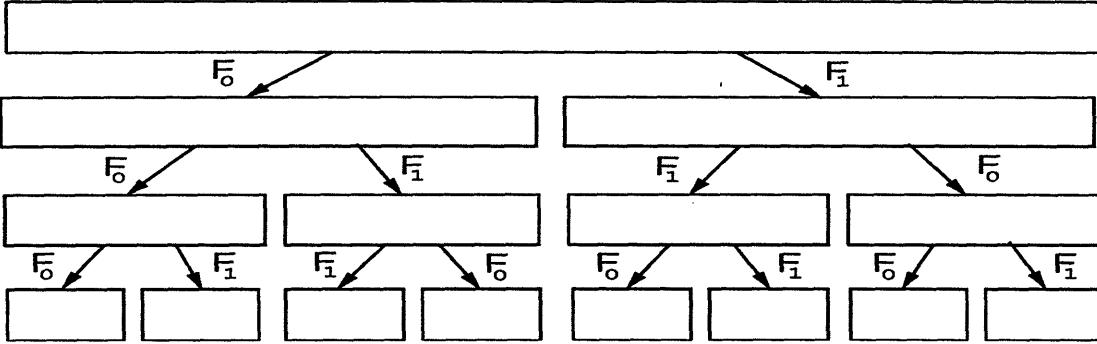


Figure 2-6: The WPD tree displayed in Paley order.

the remainder of this thesis. The Matlab [11] program that was used for the Paley ordering is included in Appendix B.

2.1.2 Indexing the Wavelet Packet Decomposition Tree

In this thesis the bin locations within a tree will be represented by the notation $b(l,c)$ where a bin is indexed by two parameters, level, l , and column, c . For example, the notation $b(1,1)$ refers to the bin at the top level containing the time domain signal. This bin is at level 1 and, since there is only one column at the top level, column 1. Figure 2-7 shows each bin of a WPD tree labeled with the appropriate bin position notation.

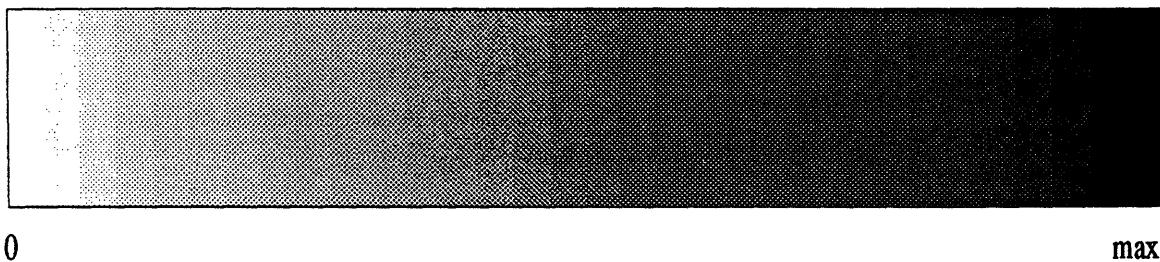
2.1.3 A few Examples of Wavelet Packet Decomposition

A few examples will illustrate the time-frequency trade-off inherent in the WPD. Each example is illustrated by a figure having three plots. The first plot of each figure shows a 128-sample signal, the second plot is an image display of the WPD tree, and the third plot shows the discrete Fourier transform of the signal. Each bin vector of the WPD tree is displayed as a rectangular intensity plot at its appropriate position in the tree. The magnitude of each element of a bin vector is displayed from left to right within a rectangular intensity plot. The shading scheme is shown in Figure 2-8. Black corresponds to the maximum value in the tree and white corresponds to zero.

We begin with the discrete impulse localized in time but not in frequency shown

$b(1,1)$							
$b(2,1)$				$b(2,2)$			
$b(3,1)$		$b(3,2)$		$b(3,3)$		$b(3,4)$	
$b(4,1)$	$b(4,2)$	$b(4,3)$	$b(4,4)$	$b(4,5)$	$b(4,6)$	$b(4,7)$	$b(4,8)$

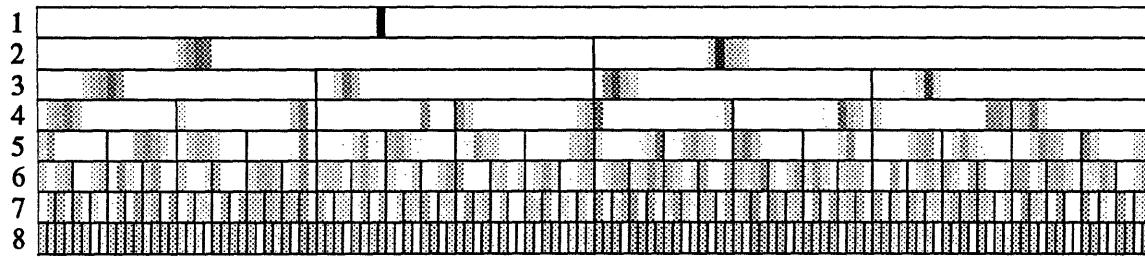
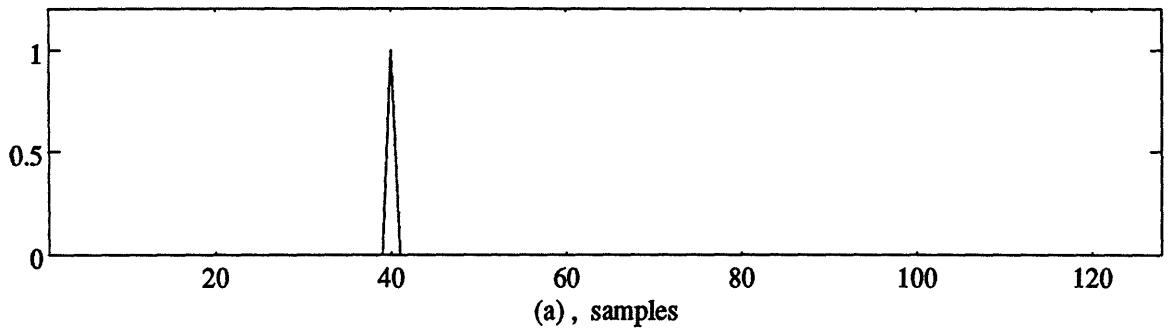
Figure 2-7: The WPD tree with index label at each bin in the first four levels.



0

max

Figure 2-8: Color mapping used in the image displays.



(b)

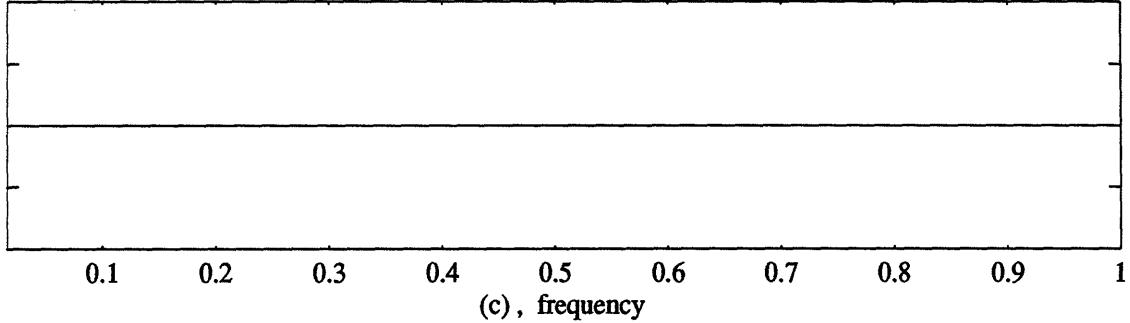


Figure 2-9: The WPD of an impulse. (a) A discrete impulse (b) The WPD of the impulse (c) The magnitude of the discrete Fourier transform of the impulse, where the horizontal axis is frequency in units of $\pi \times$ radians.

in Figure 2-9. The top level of the WPD tree is an intensity plot of the 128-sample signal containing one impulse and is focused at one sample. The bin vectors become more diffused as the tree is traversed down each level. The values at the lowest level of the tree show a similar distribution to the discrete Fourier transform of the impulse.

The next example, shown in Figure 2-10, is the WPD of a signal made from two sinusoids,

$$\sin(2\pi f_1 t) + \sin(2\pi f_2 t),$$

where $f_1 = \frac{1}{4} f_{sample}$ and $f_2 = \frac{1}{8} f_{sample}$ and f_{sample} is the frequency at which the

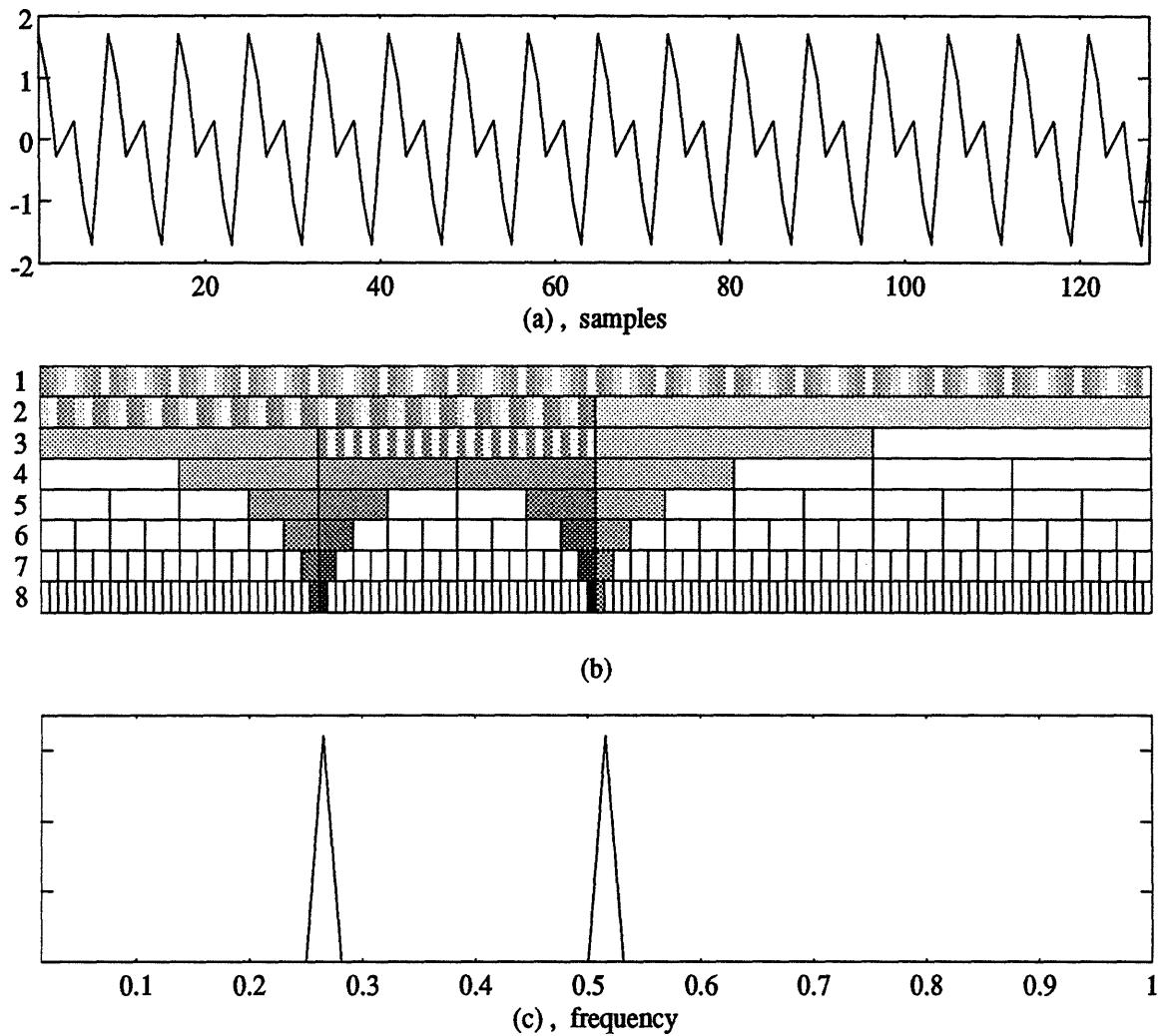


Figure 2-10: The WPD of two sinusoids. (a) A frequency localized signal (b) The WPD of the signal (c) The magnitude of the discrete Fourier transform of the signal, where the horizontal axis is frequency in units of π times radians.

signal is sampled. From Figure 2-10 we see that as the levels of the WPD tree are traversed, the information becomes more focused. The lowest level of the tree agrees with the discrete Fourier transform of the signal.

In the last example, shown in Figure 2-11, signal is derived from a wavelet packet expansion having only one non-zero wavelet coefficient. The resulting signal is localized in both time and frequency. Figure 2-11 shows a focusing of information at level 5 of the tree. The information is less focused at the top and bottom of the tree. We also see that the Fourier transform does not simplify the representation of this signal

and that the simplest representation would be at level 5 of the WPD tree.

2.2 Energy Mapping of the Wavelet Packet Decomposition Tree.

An intuitively pleasing representation of the WPD tree is one that highlights the energy distribution of the signal as it is decomposed down the tree. Such an energy map would calculate one bin energy, $e_{\mathbf{y}_b}$, from each bin vector, \mathbf{y}_b . A simple energy calculation is the average energy in each bin

$$e_{\mathbf{y}_b} = \frac{1}{2^{\eta_b}} \sum_{j=1}^{2^{\eta_b}} \mathbf{y}_b[j]^2, \quad (2.1)$$

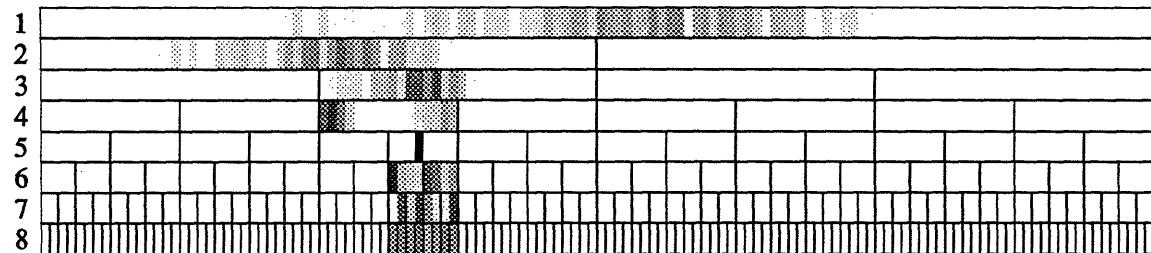
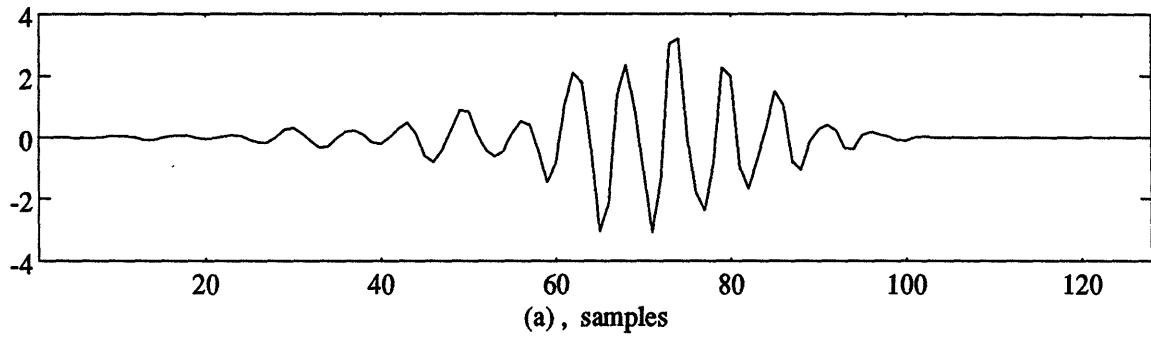
where 2^{η_b} is the number of elements in \mathbf{y}_b .

The energies are displayed in the same manner as the WPD tree. An example energy map is shown in Figure 2-12. The Matlab program used to calculate the energy map of a WPD tree using (2.1) is included in Appendix B.

The average energy used for energy tracking of the WPD may not be suited for all signals. An energy map which calculates the average energy per bin may lead to the oversight of strong time-dependent features that should be used in the classification process. There is, however, freedom in the choice of the calculation of the bin energy. For example, due to the short duration of a shrimp snap relative to a whale click (see Chapter 3) it may be beneficial to calculate a windowed energy in each bin. This can be done by calculating the energy in adjacent or overlapping segments of the bin vector and choosing a segment length small enough to encompass one snap at a time. This leads to defining an energy function of the following form:

$$\mathbf{e}_{\mathbf{y}_b}[i] = \frac{1}{L} \sum_{j=k}^{k+L-1} \mathbf{y}_b[j]^2 \quad (2.2)$$

Calculating each bin energy by sliding a window across the elements of the bin vectors of the WPD tree would give one bin-energy-vector, $\mathbf{e}_{\mathbf{y}_b}$, for each bin vector,



(b)

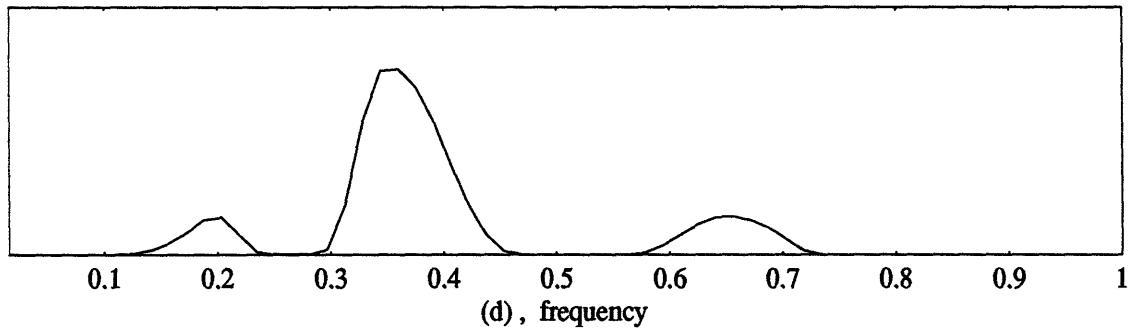


Figure 2-11: The WPD of a time and frequency localized function. (a) A frequency localized function (b) The WPD of the signal (c) The magnitude of the discrete Fourier transform of the signal, where the horizontal axis is frequency in units of $\pi \times$ radians.

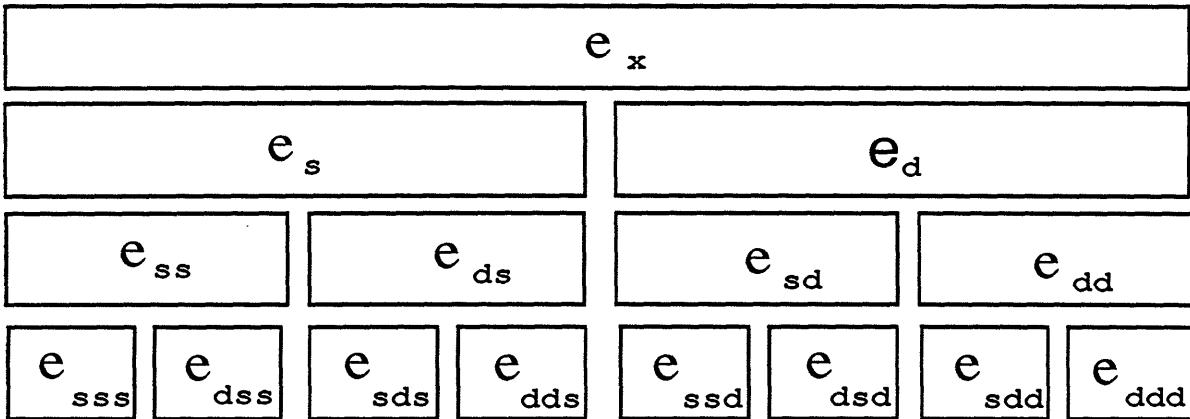


Figure 2-12: Energy mapping of the WPD tree from Figure 2-3.

y_b , of the WPD tree. One element of the bin-energy-vector, $\mathbf{e}_{y_b}[i]$, can be calculated for each position, i , of the window. The window of length L should be no larger than the length of the bin vector, 2^{n_b} . The amount to shift the window as it is moved across the bin-vector is denoted by s . In the above equation, the position of the window is determined by k , where $k = 1, (1 + s), (1 + 2s), \dots, (1 + i s), \dots, (1 + 2^{n_b} - L)$. For example, if the window length is equal to the vector length, $L = 2^{n_b}$, (2.2) would reduce to the average energy in (2.1). The other extreme would be to let the window length be one, then each bin-energy-vector, \mathbf{e}_{y_b} , would be the bin vector, y_b . The Matlab program used to calculate the energy map of a WPD tree using an arbitrary window length and overlap is included in Appendix B.

Chapter 3

Application of the Wavelet Packet Transform to the Data

An empirical analysis of a database containing samples of the transient signals that were to be automatically classified was the impetus for applying the wavelet packet transform to the problem of transient signal classification. This chapter provides the reasons that led to the application of the wavelet packet transform to the classification algorithm for these signals and describes the signals and their wavelet packet transforms and energy maps.

3.1 Description of the Data

This thesis uses a collection of ocean recordings made available by the Charles Stark Draper Laboratory and the Naval Underwater Systems Center (NUSC). The data consists of several hundred digitized records. Each record contains 12 to 20 seconds of biologically generated underwater sounds in ambient ocean noise. The recordings have been lowpass filtered with a cutoff frequency of 5KHz and, subsequently, sampled at 25KHz. The Nyquist sampling rate is 10KHz.

The biologically generated sounds are sperm whale clicks, porpoise whistles, whale cries, and snapping shrimp. Snapping shrimp and whale clicks, the shortest in duration, are used for the formulation of the classification algorithm presented in this

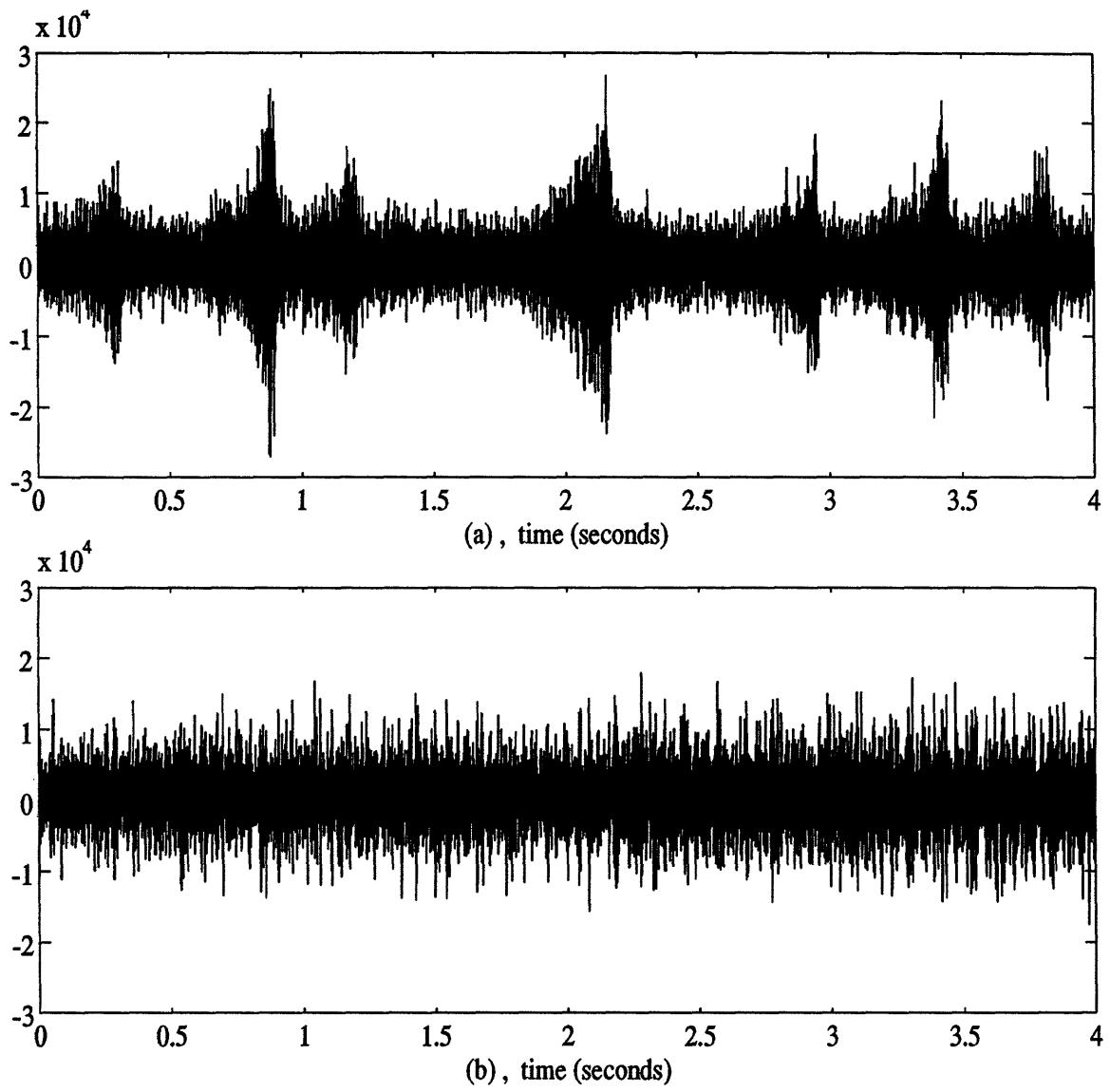


Figure 3-1: A four second interval of (a) whale clicks and (b) snapping shrimp.

thesis. A typical whale click will have a duration of approximately 80 to 120 milliseconds and a single snap of a shrimp will have a duration on the order of 1 millisecond. In comparison, a minimum duration for whale cries and porpoise whistles is approximately 2 seconds. In addition to the signals, each record contains portions of background noise alone. Figure 3-1 shows excerpts of whale clicks and snapping shrimp in ambient ocean noise. These excerpts are four seconds in duration.

A single whale click can be encompassed by a 163.8 millisecond or 4096-sample window. A power of 2 window size simplifies calculations of the wavelet packet transform. The 4096-sample window holds one to an uncountably large number of snaps.

A sample data set and a test data set, summarized in Table 3.1, were made using excerpts from 9 of the NUSC data records. The sample data set is used for the feature set derivation discussed in Chapter 4. The test data set is used to run simulations of the classification algorithms discussed in Chapters 5 and 6. The sample signal data set comprises 29 isolated whale clicks from six click records, 26 noise excerpts from the same six click records, and 20 excerpts from three records of snapping shrimp. There are 75 signals in the sample signal set. The test signal set comprises 94 isolated whale clicks from six click records, 73 noise excerpts from the same six click records, and 73 excerpts from three records of snapping shrimp. There are 240 excerpts in the test signal set. The notation $x_{t,k}(n)$ is used to represent the k^{th} signal of class t in the sample signal data set where $t = c$ (click), n (noise), and s (shrimp). The notation M_t is used in Table 3.1 to represent the total number of signals from class t in the sample data set. Figure 3-2 shows three data excerpts from the sample signal data set.

To get another perspective of these three signal classes we look at their short-term Fourier transforms (STFT). Figure 3-3 shows the STFT of four-second excerpts of whale clicks, snapping shrimp, and ambient ocean noise from Figure 3-1. The STFT was calculated using an FFT window length of 1024 samples with no overlap. The STFTs suggest that in addition to being quite brief in time duration, both the whale clicks and the snapping shrimp have a fairly broad frequency distribution.

Figure 3-4 shows a 256-no-overlapped STFT of the 4096-sample signals in Figure 3-

Table 3.1: The sample signal data set and the test signal data set.

Signal Class	Record Number	Number of Excerpts Taken		Variable Names for the Sample Signal Set
		sample set	test set	
Whale Click	NUSC2153	5	16	$x_{c,1}(n) - x_{c,5}(n)$
	NUSC2155	5	13	$x_{c,6}(n) - x_{c,10}(n)$
	NUSC2167	4	12	$x_{c,11}(n) - x_{c,14}(n)$
	NUSC2172	4	11	$x_{c,15}(n) - x_{c,18}(n)$
	NUSC2175	6	27	$x_{c,19}(n) - x_{c,24}(n)$
	NUSC2695	5	15	$x_{c,25}(n) - x_{c,29}(n)$
	total	$M_c = 29$	94	
Snapping Shrimp	NUSC2424	7	27	$x_{s,1}(n) - x_{s,7}(n)$
	NUSC2412	7	28	$x_{s,8}(n) - x_{s,14}(n)$
	NUSC2424	6	18	$x_{s,15}(n) - x_{s,20}(n)$
	total	$M_s = 20$	73	
Background Noise	NUSC2153	5	14	$x_{n,1}(n) - x_{n,5}(n)$
	NUSC2155	4	11	$x_{n,6}(n) - x_{n,9}(n)$
	NUSC2167	4	12	$x_{n,10}(n) - x_{n,13}(n)$
	NUSC2172	4	10	$x_{n,14}(n) - x_{n,17}(n)$
	NUSC2175	4	12	$x_{n,18}(n) - x_{n,21}(n)$
	NUSC2695	5	14	$x_{n,22}(n) - x_{n,26}(n)$
	Total	$M_n = 26$	73	

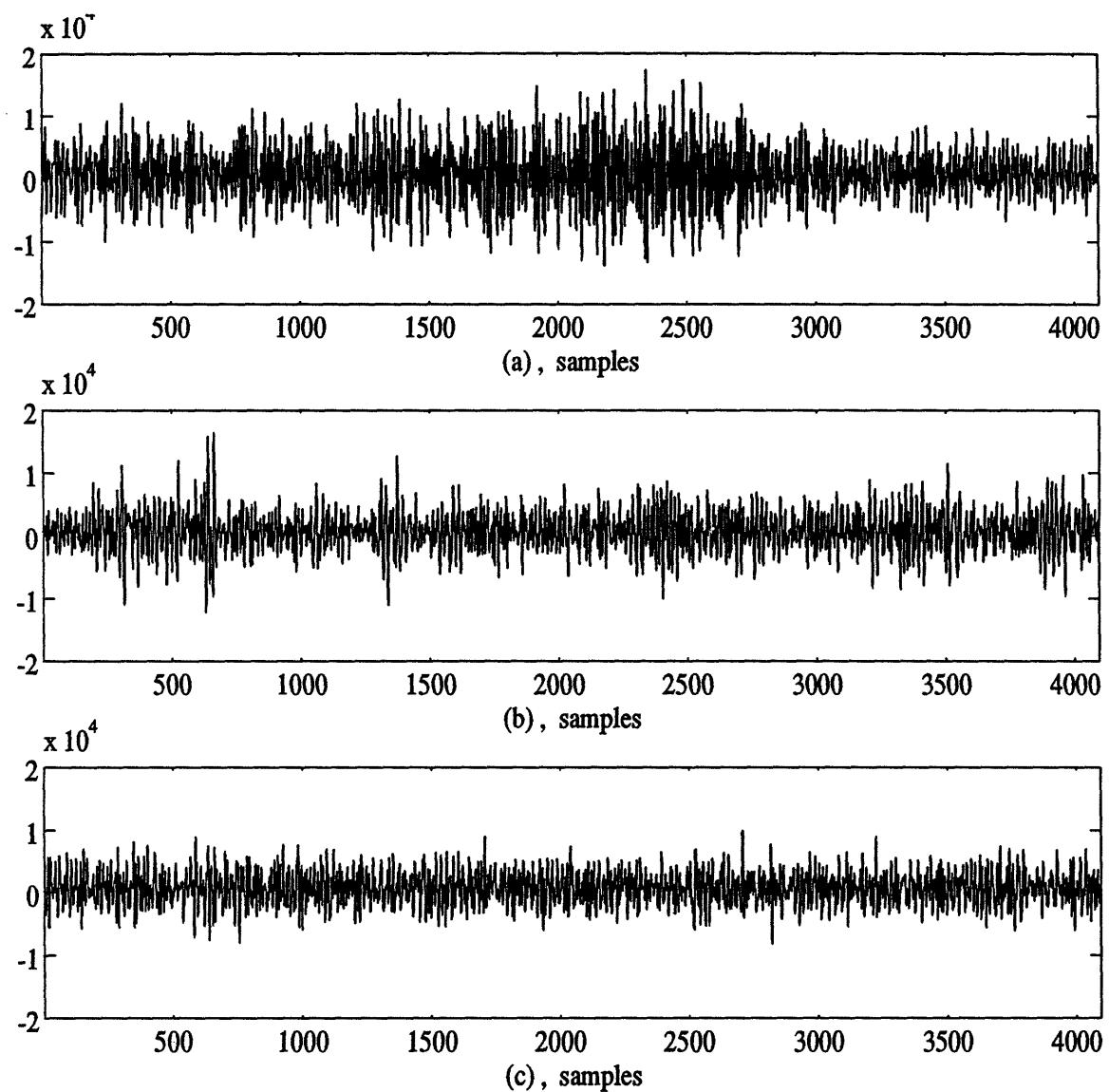


Figure 3-2: Some 4096-sample plots from the sample signal data set. (a) Whale Clicks. (b) Snapping Shrimp. (c) Background Noise.

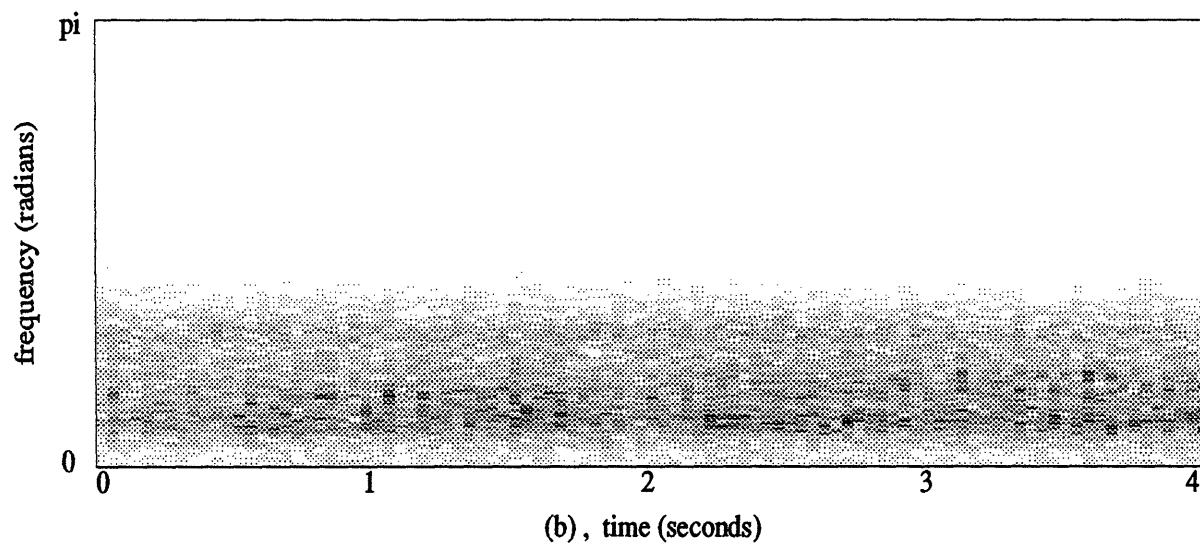
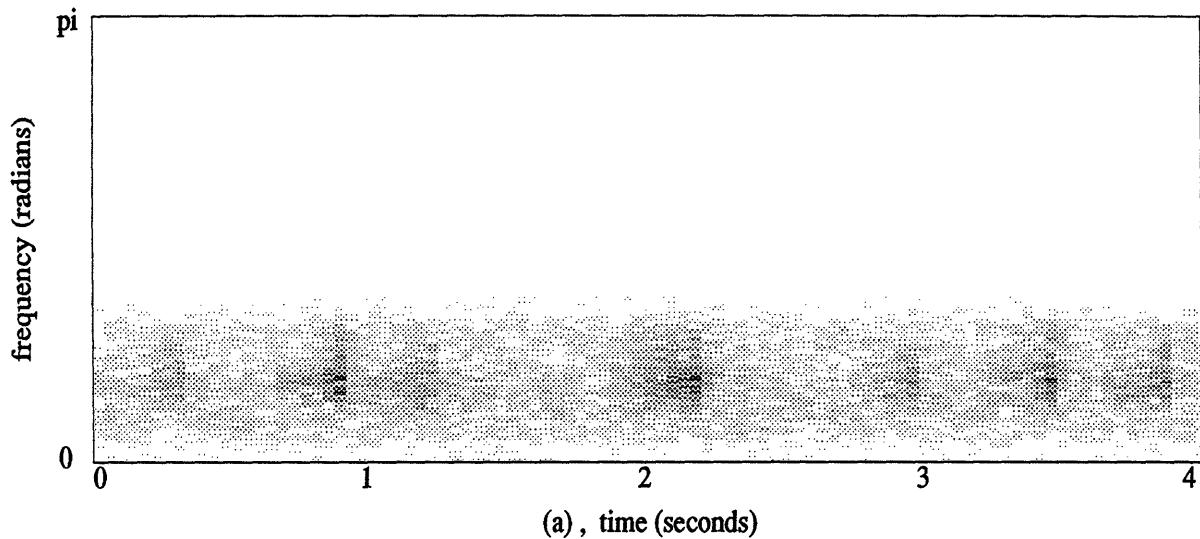


Figure 3-3: The short term Fourier transforms of the whale clicks and snapping shrimp from Figure 3-1. (a) Whale Clicks. (b) Snapping Shrimp.

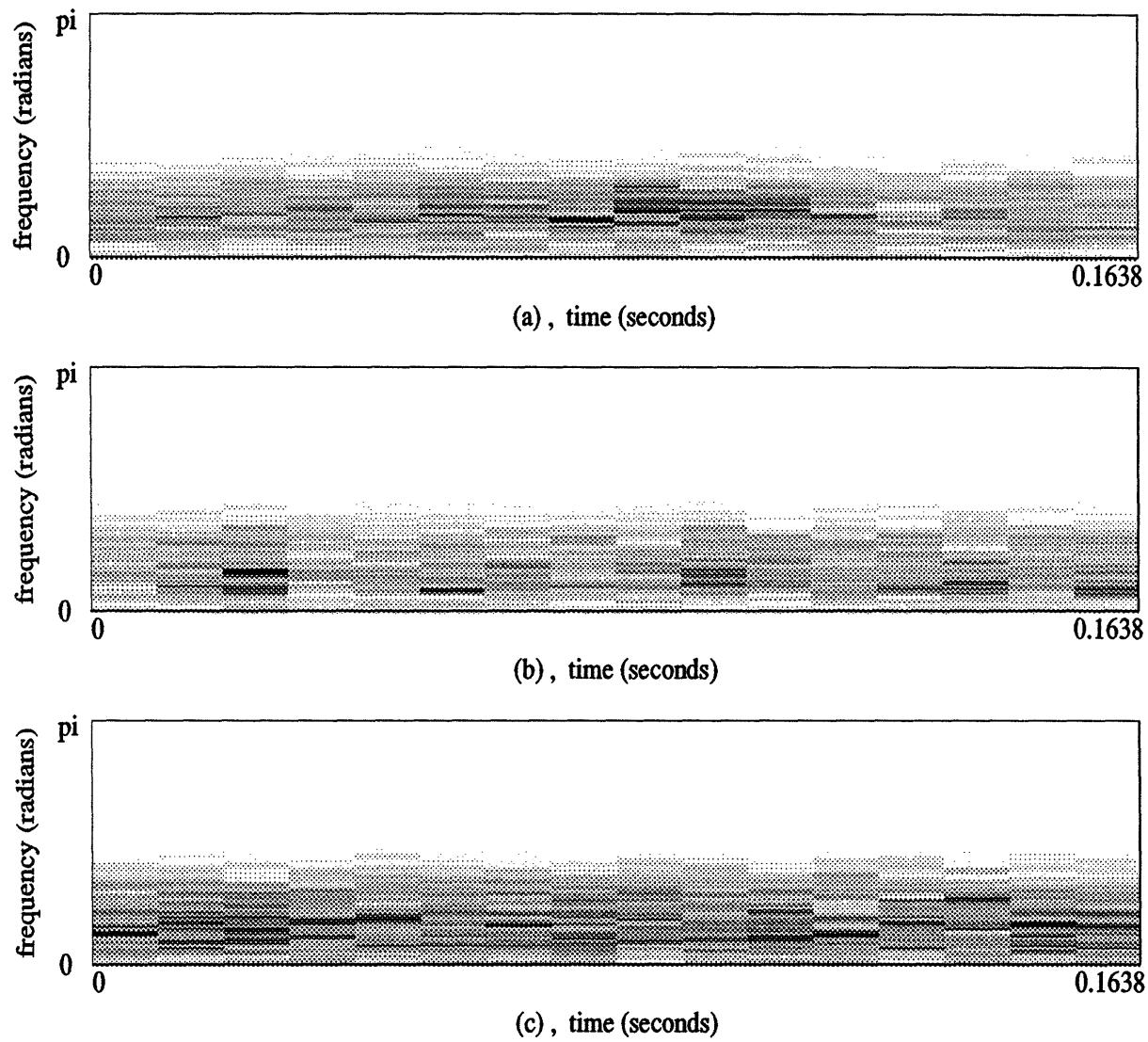


Figure 3-4: The STFT of the 4096-sample signals from Figure 3-2. A window length of 256 samples was used. (a) Whale Clicks. (b) Snapping Shrimp. (c) Background Noise.

2. In order for classification to be possible, features which dominate one signal class must be distinct from features which dominate another signal class. As can be seen from Figure 3-4, the STFT transform being well suited for narrow band, long duration, stationary signals, does not do well in separating features for many transient signals such as the whale click and shrimp snaps of interest. This difficulty was predicted in the discussion in Section 1.2. We look to the wavelet packet transform for some class-specific adaptability to aid in feature extraction.

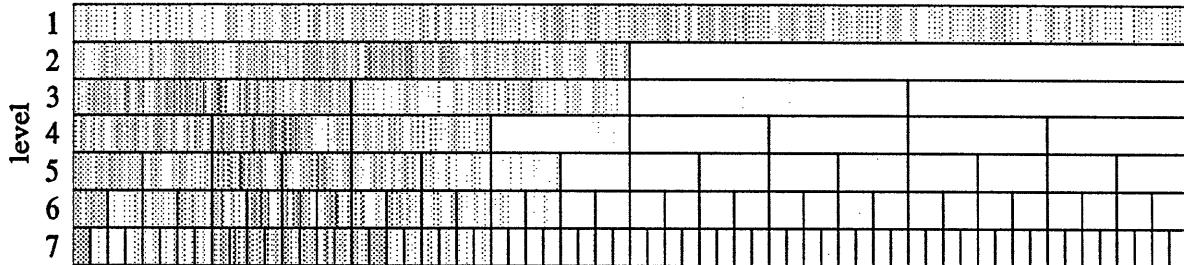
3.2 Wavelet Packet Transform of the Data

The wavelet packet transform, with its variable degree of localization in both time and frequency, is a promising tool for the detection and classification of transient signals and signals whose energy patterns may not simply be localized in frequency. Using our Matlab implementation of the algorithm presented in [1] with the Daubechies 14 point wavelet [7], the wavelet packet transforms of the 75 data excerpts from Table 3.1 were calculated.

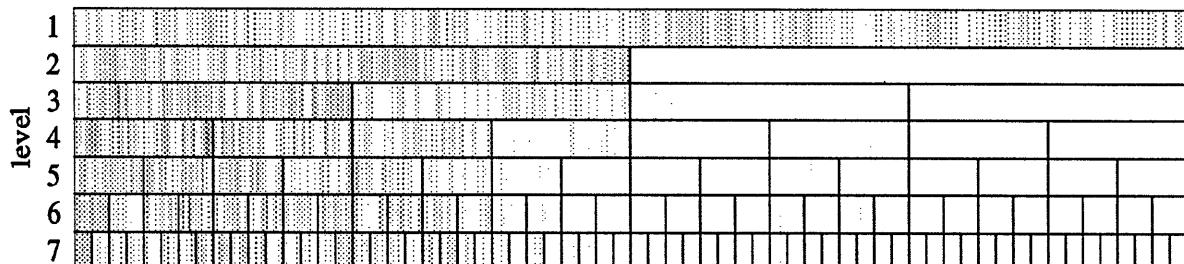
Shown in Figure 3-5 are the wavelet packet transforms of the three excerpts that appear in Figure 3-2. As detailed in Section 2.1.3 the magnitude of the wavelet packet transform is displayed as an intensity image tree with the wavelet packet transform coefficients displayed as an intensity vector at each bin.

Due to the sampling of these signals at over twice the Nyquist rate, all of the information in the wavelet packet transform of these excerpts is contained in the portions corresponding to the lower half of the frequency spectrum. The upper half of the frequency spectrum correspond to the right most bin vectors which are essentially zero. Beyond this simple observation, any other information that may be present is not apparent at this stage.

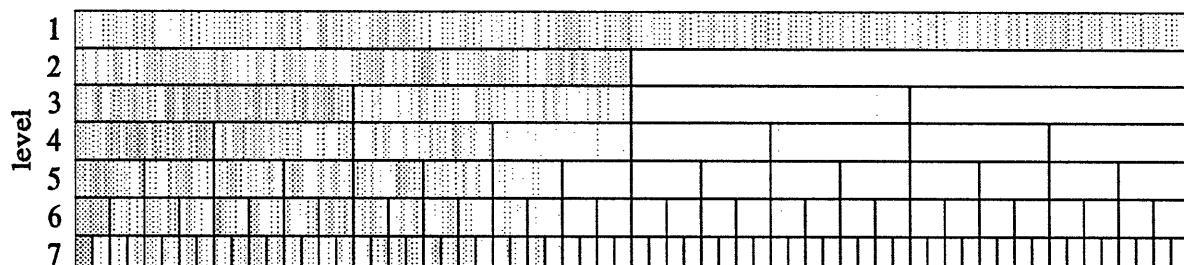
The human eye has a difficult, if not impossible, time assimilating the information contained in the wavelet packet transform. The quality of the intensity plot is degraded when using a gray scale. In addition, it is difficult to display the wavelet packet transform coefficients to preserve both time and frequency characteristics. The next



(a)



(b)



(c)

Figure 3-5: The first seven levels of wavelet packet transforms of the 4096-sample excerpts from Figure 3-2. (a) Whale Click. (b) Snapping Shrimp. (c) Background Noise.

section discusses a way to simplify the information embedded in the wavelet packet transform by reducing the number of parameters in order to make the information easier to manipulate.

3.3 Energy Mapping of the Wavelet Packet Transform of the Data

The wavelet packet transforms shown in Section 3.2 need further manipulation so that characteristics unique to each class may be found. A mapping of the wavelet packet transform which highlights the energy distribution of the signal through the wavelet packet decomposition is intuitively pleasing. See Section 2.2 for a detailed explanation of the energy mapping of the WPD tree.

An energy map was calculated for each of the 75 WPD trees of the sample data set. The average bin energy, $\epsilon_{\mathbf{y}_b}$, from (2.1) (repeated below) was used to calculate the energy for each bin vector of the WPD tree.

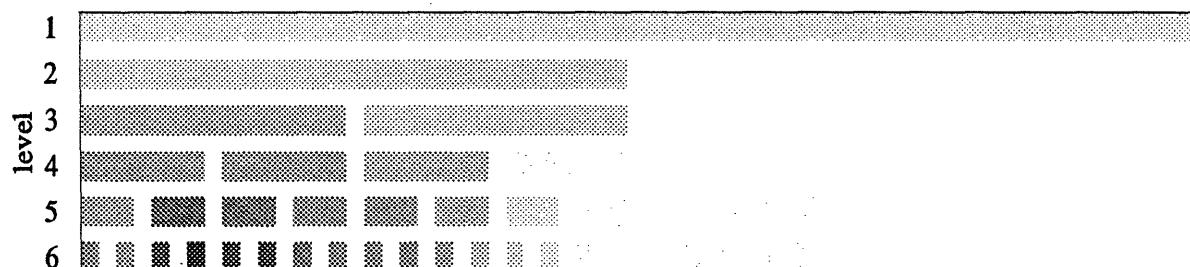
$$\epsilon_{\mathbf{y}_b} = \frac{1}{2^{n_b}} \sum_{j=1}^{2^{n_b}} \mathbf{y}_b[j]^2 \quad (3.1)$$

Figure 3-6 shows the energy maps of the wavelet packet transforms from Figure 3-5.

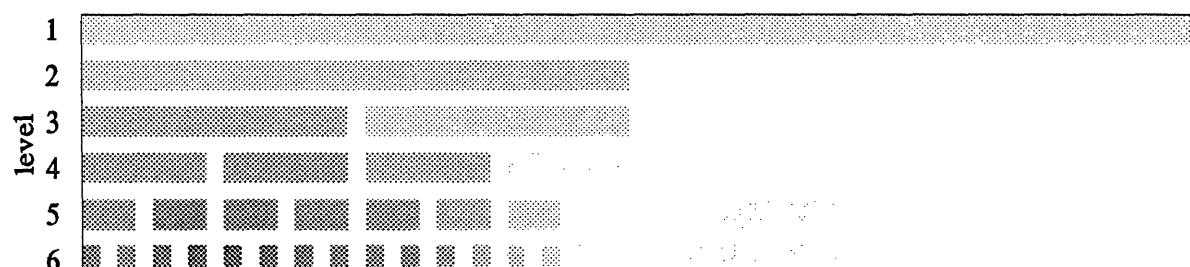
This mapping of the WPD trees shows promising clarification of information. At a glance, one can see a definite difference in the intensity distributions between the three trees shown in the figure. A brief qualitative analysis of a few energy maps showed similar energy distributions within each class. A quantitative analysis of the patterns exhibited by the energy maps is discussed in the next chapter.



(a)



(b)



(c)

Figure 3-6: Energy maps of the first 6 levels of the wavelet packet transforms shown in Figure 3-5. (a) Whale Click. (b) Snapping Shrimp. (c) Background Noise.

Chapter 4

Choice of an Optimum Reduced Parameter Feature Set

In the formulation of a decision rule, it is desirable to find a feature set which represents each class of signals. Typically, the feature set uses a greatly reduced number of parameters in comparison with the number of samples used to represent the signal. In this chapter, a feature set which best separates characteristics specific to each class is derived from the wavelet packet transforms of the three classes of signals.

4.1 Vector Representation of the Energy Maps

Further investigation of the energy maps for each class in the sample signal data set is needed. Assembling each of the 75 energy maps into column vectors allows for easier manipulation of the information. Let each energy vector be denoted by $\mathbf{e}_{t,k}$ for each class $t = c, s, n$ and signal number $k = 1, \dots, M_t$, where M_t represents the number of signals in the sample signal data set for the given class, t . The elements of this energy vector are the bin energies of interest and may include all or some bin energies at any level of the energy map.

The transformation of the sample signal data set to an energy vector set is summarized in Table 4.1. An energy map was found from the wavelet packet decomposition of each of the 75 signals in the sample signal data set as discussed in Section 3.3. One

Table 4.1: An energy vector set may be made from the energy maps of the sample signal data set.

<i>Signal Class</i>	<i>Signals in the Sample Signals Data Set</i>	<i>Corresponding Energy Vectors</i>
whale clicks	$x_{c,1}(n)$ through $x_{c,29}(n)$	$\mathbf{e}_{c,1}$ through $\mathbf{e}_{c,29}$
snapping shrimp	$x_{s,1}(n)$ through $x_{s,20}(n)$	$\mathbf{e}_{s,1}$ through $\mathbf{e}_{s,20}$
background noise	$x_{n,1}(n)$ through $x_{n,26}(n)$	$\mathbf{e}_{n,1}$ through $\mathbf{e}_{n,26}$

energy vector can be found from each of these 75 energy maps.

An energy matrix, E_t , for each signal class, $t = c, s$, and n , may be constructed by aligning the M_t energy vectors from Table 4.1 into columns as shown below.

$$E_t = \begin{bmatrix} & & \\ \mathbf{e}_{t,1} & \mathbf{e}_{t,2} & \cdots & \mathbf{e}_{t,M_t} \\ & & \end{bmatrix} \quad (4.1)$$

4.2 Examination of 63 Bins of the Energy Maps

A first step in the analysis of the sample data set is to quantitatively test that all energy maps from a given class exhibit similar behavior. This can be done by looking at the singular value decomposition (SVD) of the matrices, E_t , from (4.1). Singular value decomposition is a way to find, among other things, the effective rank and the column space for rectangular matrices. We are interested in finding the rank of each E_t matrix because a rank equal to one would mean that the energy intensity distribution within energy maps of a given class exhibit identical behavior. The column space tells us the range of each matrix, E_t . If the rank is one, the column space is represented by

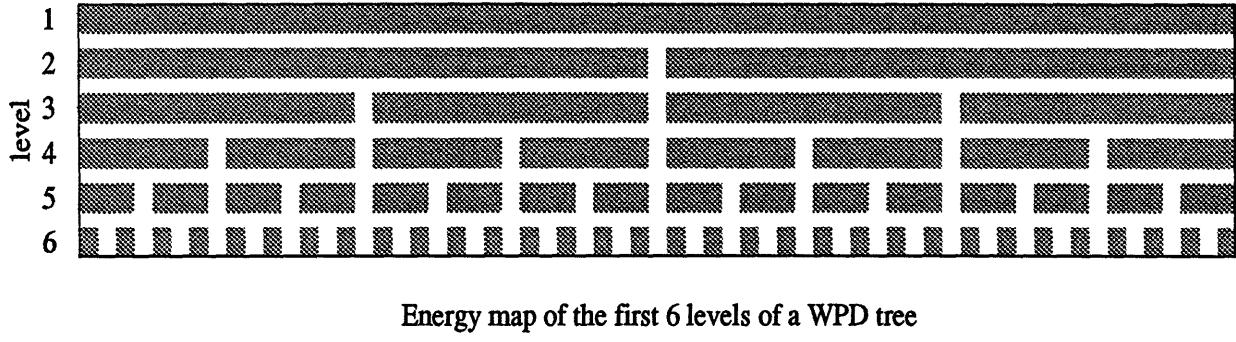


Figure 4-1: The energies in the 63 bins of the first 6 levels of an energy map are arranged into an energy vector, $\mathbf{e}_{t,k}$.

only one singular vector which can be thought of as the representative energy vector for that class. See [12] for a detailed tutorial on singular value decomposition.

Initially, the degree of agreement among the energy distributions within energy maps of a given class is determined. An energy vector, $\mathbf{e}_{t,k}$, which includes all bin energies from the first six levels of the energy map is created for each signal in the sample signal data set. The first six levels of an energy map are shown in Figure 4-1. The 63 elements of each energy vector, $\mathbf{e}_{t,k}$, correspond to the 63 bin energies of the energy map.

A $63 \times M_t$ element energy matrix, E_t , may be constructed for each class, $t = c$, s , and n , as in (4.1). Singular value decomposition of E_t gives

$$E_t = U \Sigma V^T. \quad (4.2)$$

The 63-element singular vectors, \mathbf{u}_k , make up the columns of the 63×63 orthogonal matrix U . The first M_t columns of U define the column space or range of E_t .

$$U = \begin{bmatrix} & & & \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{63} \\ & & & \end{bmatrix} \quad (4.3)$$

The $63 \times M_t$ singular value matrix, Σ , displays the rank in the first M_t diagonal

elements. The rank (or effective rank) of E_t is equal to the number of non-zero (or non-negligible) singular values.

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_{M_t} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (4.4)$$

The row space and nullspace of E_t are defined in the $M_t \times M_t$ matrix, V^T . The information in V^T is not used in the analysis of the energy maps.

$$V^T = \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_{M_t}^T \end{bmatrix} \quad (4.5)$$

To determine the effective rank of each of the three energy matrices, a difference ratio, $d_{t,\sigma}$, between the largest and second largest singular values for each of the three classes was calculated by

$$d_{t,\sigma} = \frac{\sigma_{t,1} - \sigma_{t,2}}{\sigma_{t,1}} \quad (4.6)$$

and is displayed in Table 4.2.

The difference ratio between the largest singular value and the second largest singular value is quite large in all three cases. This tells us that each of the three E_t matrices has an effective rank of one, confirming that the first six levels of the energy maps of signals within a given class do, indeed, exhibit similar behavior.

The elements of the singular vectors that correspond to the maximum singular value for each class are displayed in Figure 4-2. Notice that high valued elements for the noise roughly coincide with both the high valued elements for the snapping shrimp

Table 4.2: Difference ratio between the largest and second largest singular values for each $63 \times M_t$ element energy matrix, E_t .

Class t	$\sigma_{t,1} \times 10^6$	$\sigma_{t,2} \times 10^6$	d_{σ_t}
whale clicks	2221	285	0.87
snapping shrimp	762	79	0.90
background noise	412	43	0.89

and the whale clicks. The figure also reveals that the highest valued elements for whale clicks differ from the highest valued elements for snapping shrimp. If the noise were not present, a feature set may be chosen to include bin energies that correspond to significant elements for snapping shrimp and whale clicks, but our problem is to derive a feature set from the data set of these sounds in background noise. Before continuing the search for a reduced parameter feature vector from the energy maps of the sample signal data set, the influence of noise should be compensated for.

4.3 Compensating for the Noise

Each bin energy contains both signal and noise energies, as does each component of the energy vector, $\mathbf{e}_{t,k}$. The 26 energy maps of background noise displayed consistent energy distribution patterns. An example is seen in the sample energy map of background noise shown in Figure 3-6. This distribution of background noise energy within the energy maps may mask dominant features that are specific to shrimp or clicks. Once the noise is normalized out of each bin energy, features that best separate the two classes may be chosen without the influence of noise.

Let \mathbf{x}_s denote the portion of the received signal that is due to the source alone.

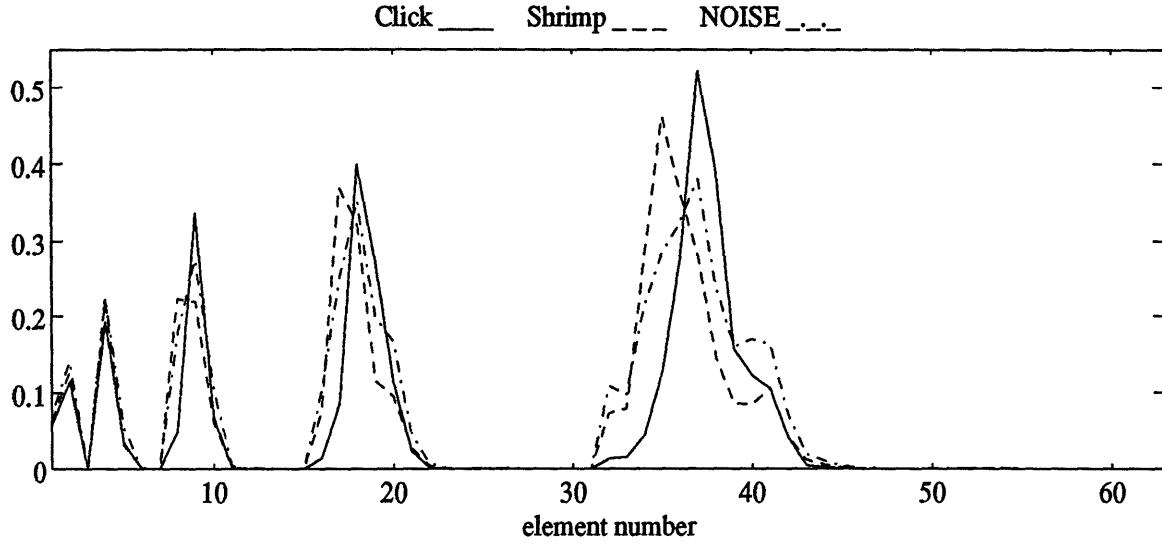


Figure 4-2: Components of the 63-element singular vectors obtained from the SVD of three energy matrices made from he noise normalized energy maps.

Let \mathbf{x}_n denote the portion of the received signal that is due to background noise. The received signal, \mathbf{x}_y , may be written as a linear combination of the source signal and the background noise.

$$\mathbf{x}_y = \mathbf{x}_s + \mathbf{x}_n \quad (4.7)$$

Let the vector of the WPD tree of \mathbf{x}_y which appears in bin b be denoted by \mathbf{y}_b . Using the same notation, the vector at bin b of the WPD of \mathbf{x}_s is denoted by \mathbf{s}_b . Similarly, the bin vector at bin b due to the WPD of \mathbf{x}_n is \mathbf{n}_b . Because the wavelet packet decomposition is a linear transform, the bin vector at each bin of the WPD tree, \mathbf{y}_b , can be written as a linear combination of the bin vector due to the source, \mathbf{s}_b , and the bin vector due to the noise, \mathbf{n}_b .

$$\mathbf{y}_b = \mathbf{s}_b + \mathbf{n}_b \quad (4.8)$$

In agreement with (2.1) from Section 2.2, let the energy due to the bin vector \mathbf{y}_b

be defined by $\epsilon_{\mathbf{y}_b}$. Equation (2.1) is repeated here for convenience.

$$\epsilon_{\mathbf{y}_b} = \frac{1}{2^{n_b}} \sum_{j=1}^{2^{n_b}} \mathbf{y}_b[j]^2 = \frac{1}{2^{n_b}} \langle \mathbf{y}_b, \mathbf{y}_b \rangle \quad (4.9)$$

Recall, 2^{n_b} is the number of elements in \mathbf{y}_b . The notation $\langle \mathbf{a}, \mathbf{b} \rangle$ is used to denote the inner product, $\mathbf{a}^T \mathbf{b}$, of the two vectors, \mathbf{a} and \mathbf{b} .

Then the energy at bin b due to the source, denoted by S_b , is

$$S_b = \frac{1}{2^{n_b}} \sum_{j=1}^{2^{n_b}} s_b[j]^2 = \frac{1}{2^{n_b}} \langle \mathbf{s}_b, \mathbf{s}_b \rangle \quad (4.10)$$

and the energy due to the noise, denoted by N_b , is

$$N_b = \frac{1}{2^{n_b}} \sum_{j=1}^{2^{n_b}} n_b[j]^2 = \frac{1}{2^{n_b}} \langle \mathbf{n}_b, \mathbf{n}_b \rangle \quad (4.11)$$

Using (4.9) and (4.8) above to calculate the energy in the bin vector \mathbf{y}_b in terms of \mathbf{s}_b and \mathbf{n}_b to get

$$\epsilon_{\mathbf{y}_b} = \frac{1}{2^{n_b}} \langle (\mathbf{s}_b + \mathbf{n}_b), (\mathbf{s}_b + \mathbf{n}_b) \rangle. \quad (4.12)$$

Carrying out the multiplication gives

$$\epsilon_{\mathbf{y}_b} = \frac{1}{2^{n_b}} [\langle \mathbf{s}_b, \mathbf{s}_b \rangle + 2 \langle \mathbf{s}_b, \mathbf{n}_b \rangle + \langle \mathbf{n}_b, \mathbf{n}_b \rangle]. \quad (4.13)$$

Assume that the signal due to the sources (whale, shrimp), \mathbf{x}_s , are uncorrelated with the background noise, \mathbf{x}_n . Then, for data records that are sufficiently long, denoting length by 2^n , we may assume that

$$\frac{1}{2^n} \langle \mathbf{x}_s, \mathbf{x}_n \rangle = 0. \quad (4.14)$$

Any linear transformation of the source signal will then be uncorrelated with any linear transformation of the noise. As previously stated, each bin vector of the wavelet packet decomposition tree, \mathbf{y}_b , is a linear transformation of the initial vector that appears at the top level of the WPD tree, \mathbf{x}_y . Therefore, the bin vector, \mathbf{s}_b , due

to the WPD of the source signal is uncorrelated with the vector, \mathbf{n}_b , due to the WPD of noise alone. That is,

$$\frac{1}{2\eta_b} \langle \mathbf{s}_b, \mathbf{n}_b \rangle = 0 \quad (4.15)$$

at any bin, b , of the tree.

Using (4.10), (4.11), and (4.15), Equation (4.13) reduces to

$$e_{\mathbf{y}_b} = S_b + N_b \quad (4.16)$$

The energies at each bin are, thus, comprised of an energy due to the source and an energy due to noise.

Normalization of the bin energy, $e_{\mathbf{y}_b}$, by the energy due to noise alone would give $\hat{e}_{\mathbf{y}_b}$.

$$\hat{e}_{\mathbf{y}_b} = \frac{e_{\mathbf{y}_b}}{N_b} = \frac{S_b}{N_b} + 1 \quad (4.17)$$

Performing the normalization described in the above paragraphs allows for a source-signal-energy to noise-energy ratio analysis of the patterns exhibited by the sample energy maps. Figure 4-3 shows the elements of the singular vectors found from SVD of the noise normalized energy matrices. Comparing this figure with Figure 4-2 we see that the high valued elements of the shrimp singular vector differ from the high valued elements of the whale click singular vector and that there is no longer high valued elements for noise.

4.4 Reducing the Search for a Feature Set to Eleven Bins of the Energy Maps

In forming a reduced parameter feature set, we look for the dominant bin energies. Discarding bins that carry no information simplifies the feature selection process. There is no need to keep all 63 bin energies because more than half of them are zero or approximately zero. The bin energy values that correspond to the high frequency portion of the signals are quite low, this is due to the sampling of the signals at

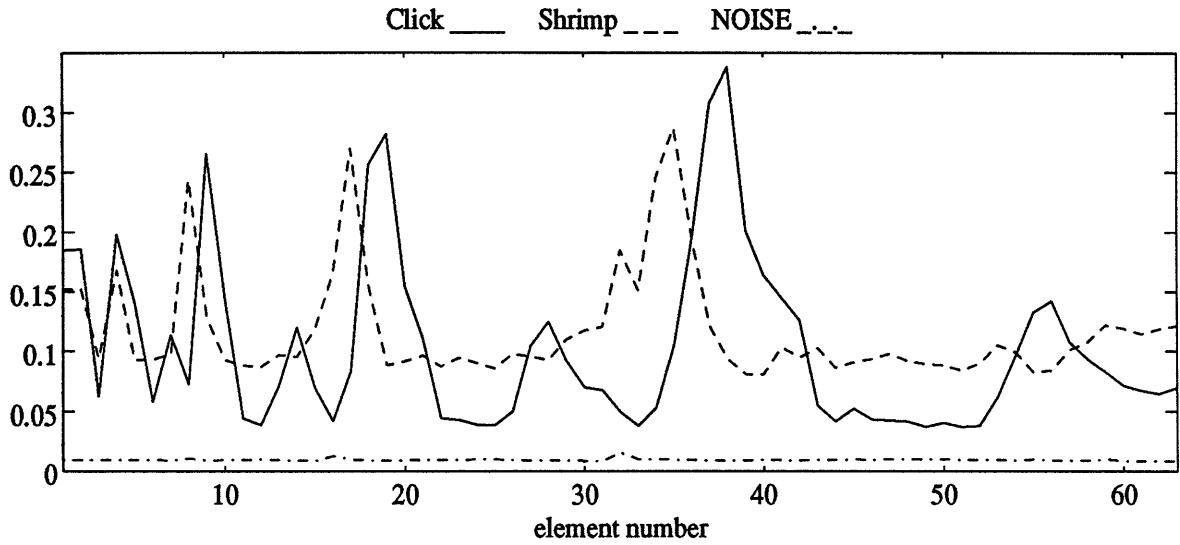


Figure 4-3: Components of the 63-element singular vectors, $\hat{\mathbf{u}}_{c,1}$, $\hat{\mathbf{u}}_{s,1}$, $\hat{\mathbf{u}}_{n,1}$ obtained from the SVD of \hat{E}_c , \hat{E}_s , and \hat{E}_n .

over twice the Nyquist rate. As Figure 3-6 of Section 3.3 reveals, all three maps exhibit negligible values in their right most bins at every level. These bins may be excluded from the energy vector, thus, simplifying further analysis. With all these bins discarded, the resulting energy vector consists of the energies corresponding to the eleven shaded bins shown in Figure 4-4. Now, let $\mathbf{e}_{t,k}$ denote this 11-element energy vector.

Proceeding with the search for a reduced parameter feature set, the normalization of the eleven bins of interest is done. An average noise energy for bin b_i of the noise energy maps is used for the noise energy, N_{b_i} , from Equation 4.17. Let $\mathbf{e}_{n,ave}$ denote the vector with the i^{th} element being this average noise energy at bin b_i . Element by element (or bin by bin) calculation of the average noise energy is done by

$$\mathbf{e}_{n,ave}[i] = \frac{1}{26} \sum_{k=1}^{26} \mathbf{e}_{n,k}[i]. \quad (4.18)$$

Each of the 75 energy vectors, $\mathbf{e}_{t,k}$, discussed in this section, corresponds to the eleven bin energies from each of the 75 sample energy maps. From these we find 75

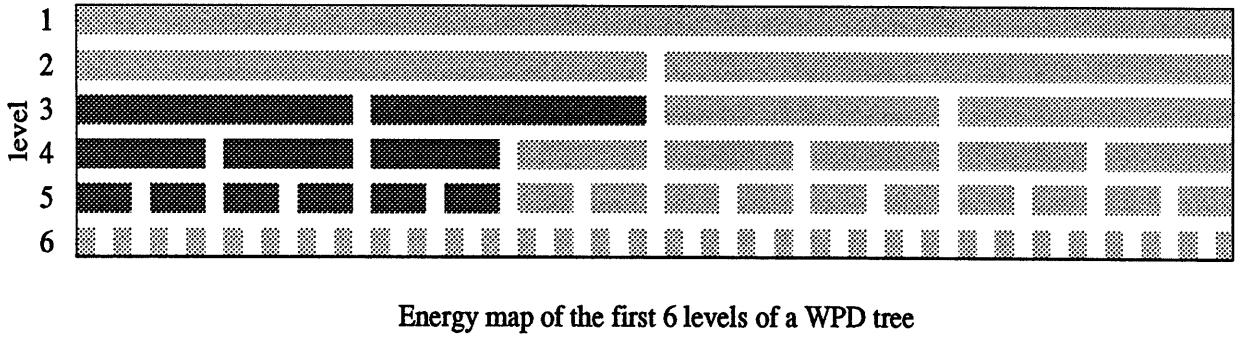


Figure 4-4: The eleven shaded bins correspond to the energies that make up the 11-component energy vector representation of the energy maps.

normalized energy vectors, $\hat{\mathbf{e}}_{t,k}$, each containing 11 normalized bin energies. Element by element normalization of $\mathbf{e}_{t,k}$ by the corresponding noise energies is done by

$$\hat{\mathbf{e}}_{t,k}[j] = \frac{\mathbf{e}_{t,k}[j]}{\mathbf{e}_{n,ave}[j]}, \quad (4.19)$$

where the element index is $j = 1, \dots, 11$, the signal number is $k = 1, \dots, M_t$, and each class is denoted by $t = c, s, n$.

4.5 Searching the Noise Normalized Energy Maps for a Reduced Feature Set

Now that the information in the energy vectors has been normalized by the noise energy, we may proceed with the search for a reduced parameter feature set which best separates the signal classes. A quantitative analysis may be done by grouping the 75 normalized energy vectors into three classes and arranging them into three matrices as shown in (4.20). Denote the $11 \times M_t$ element normalized energy matrix by \hat{E}_t .

$$\hat{E}_t = \begin{bmatrix} \hat{\mathbf{e}}_{t,1} & \hat{\mathbf{e}}_{t,2} & \cdots & \hat{\mathbf{e}}_{t,M_t} \end{bmatrix} \quad (4.20)$$

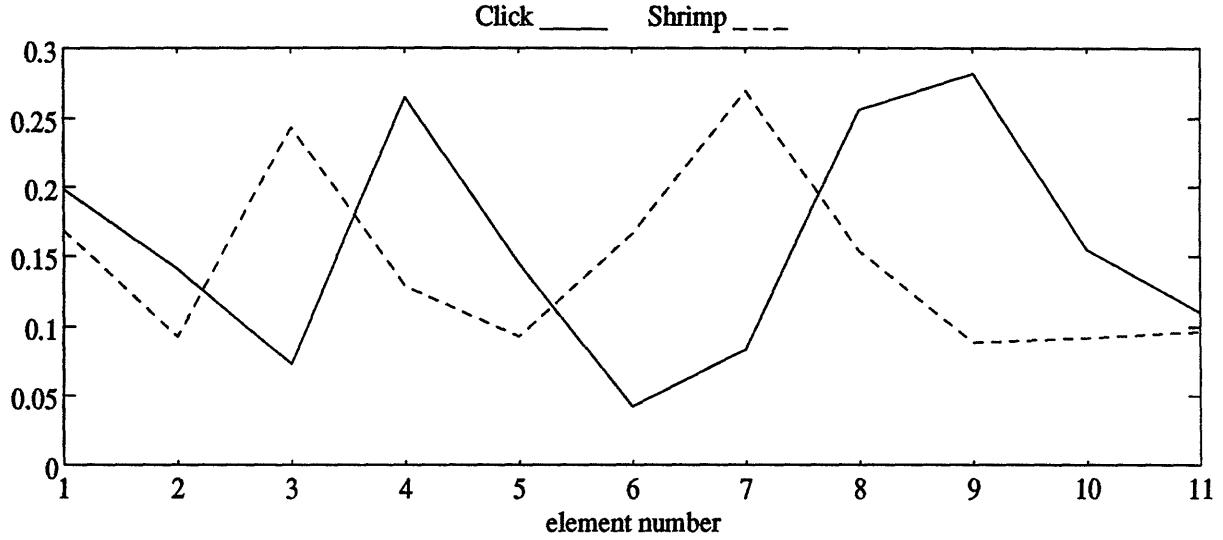


Figure 4-5: Components of the 11-element singular vectors, $\hat{\mathbf{u}}_{c,1}$, $\hat{\mathbf{u}}_{s,1}$, $\hat{\mathbf{u}}_{n,1}$ obtained from the SVD of \hat{E}_c , \hat{E}_s , and \hat{E}_n .

Again, SVD of \hat{E}_t for each class t gives one significant singular value, $\hat{\sigma}_t$, and singular vector, $\hat{\mathbf{u}}_{t,1}$. Figure 4-5 shows the components of the three singular vectors, $\hat{\mathbf{u}}_{c,1}$, $\hat{\mathbf{u}}_{s,1}$ and $\hat{\mathbf{u}}_{n,1}$. We begin by finding a collection of bins that contain significant information by examination of the components of the singular vectors, $\hat{\mathbf{u}}_{c,1}$ and $\hat{\mathbf{u}}_{s,1}$ for whale clicks and snapping shrimp. Initially, a bin is considered significant if the value of its corresponding element of the singular vector lies within 20 percent of the maximum component of that singular vector. The significant components of $\hat{\mathbf{u}}_{c,1}$ correspond to elements 4, 8 and 9. The significant components of $\hat{\mathbf{u}}_{s,1}$ correspond to elements 3 and 7. The bins corresponding to elements 3, 4, 7, 8 and 9 containing the dominant information are shaded in Figure 4-6. If the number of bins included in this collection needs to be increased, the threshold may be lowered, step by step, until the desired number of bins is chosen.

Reduction of the feature vector is desirable for the simplification of the decision rule, therefore, including superfluous information should be avoided. A feature set which contains a parent bin energy and all of its descendant bin energies may be redundant because any parent bin vector of the WPD tree can be constructed from a

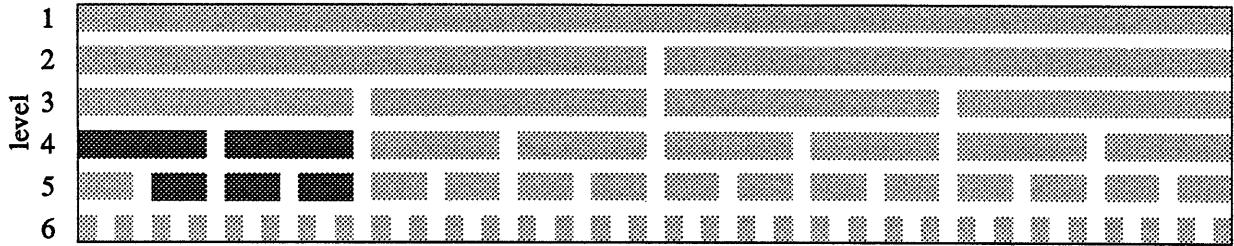


Figure 4-6: The shaded bins of the energy map correspond to the dominant elements of the 11-element singular vectors $\hat{u}_{c,1}$ and $\hat{u}_{s,1}$.

linear combination of its children bin vectors. The feature set need not include both parent and child energy bins of the energy map. Reducing the number of features used for classification will also minimize the computational complexity of the algorithm because many bins of the WPD tree will not be used and will, therefore, not be calculated.

Looking at Figure 4-6 and using the tree indexing method discussed in Section 2.1, we see that the dominant bins $b(4,1)$ and $b(4,2)$ are parents to the rest of the dominant bins $b(5,2)$, $b(5,3)$, and $b(5,4)$. Noting the parent child redundancy, it is best to see if there is enough feature separation using the energies from only the dominant parent bins at the fourth level, $b(4,1)$ and $b(4,2)$. Figure 4-7 shows elements 3 and 4 which correspond to the two dominant bins, $b(4,1)$ and $b(4,2)$, for each of the three singular vectors, $\hat{u}_{c,1}$ and $\hat{u}_{s,1}$.

The dominant ancestor bins $b(4,1)$ and $b(4,2)$ are shaded in Figure 4-8. Figure 4-9 plots the normalized energies from bins $b(4,1)$ and $b(4,2)$ of the 75 sample energy maps. There is good separation between the click and shrimp features.

Now that a reduced parameter feature set has been found, a method for detection and classification must be formulated. The next section discusses pattern recognition techniques that lend themselves to the classification of signals using features of a limited number of sample signals as a training set.

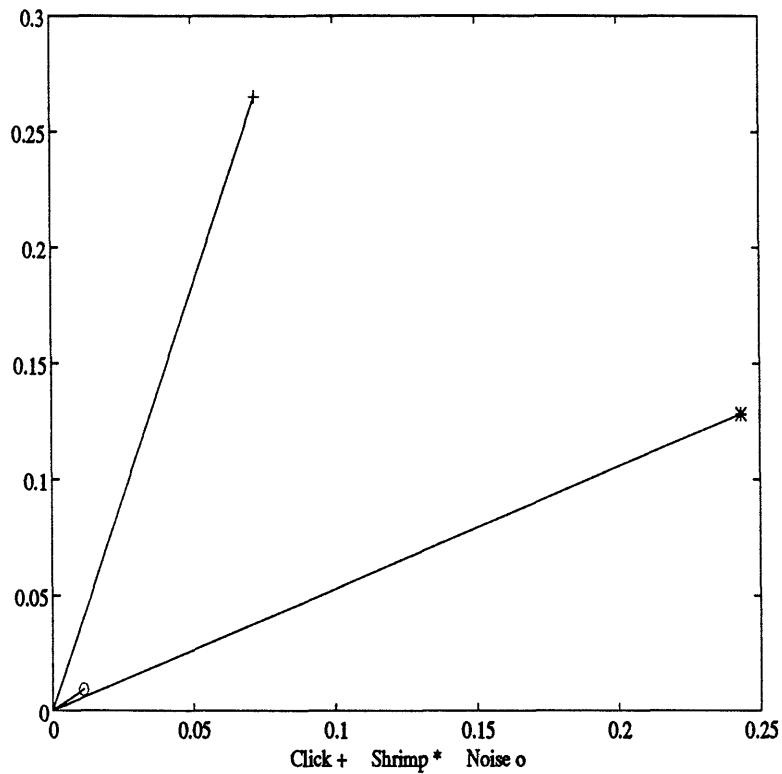


Figure 4-7: Element 3 vs element 4 of the 11-element singular vectors $\hat{u}_{c,1}$ and $\hat{u}_{s,1}$, and $\hat{u}_{n,1}$.

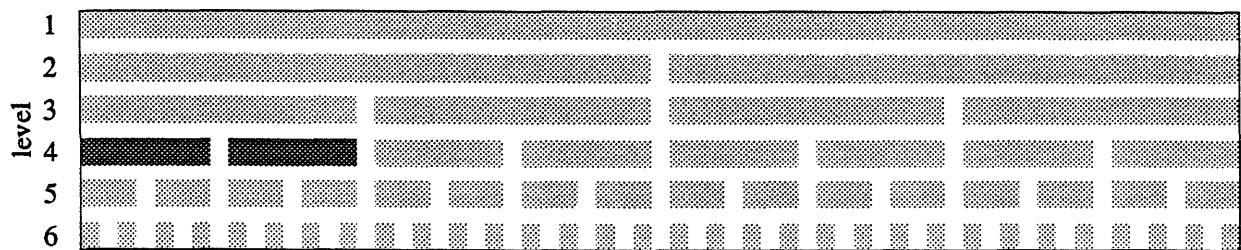


Figure 4-8: The feature set is composed of the normalized energies from bins $b(4,1)$ and $b(4,2)$ of the energy map.

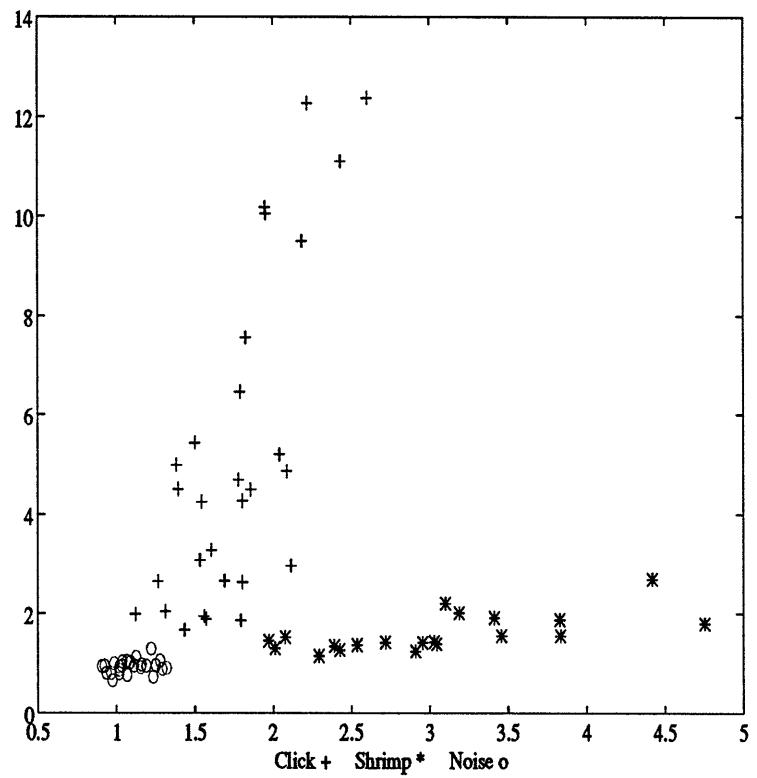


Figure 4-9: Noise normalized energies from bins b(4,1) and b(4,2) of the sample energy maps for snapping shrimp, whale clicks and background noise.

Chapter 5

Decision Rules

In Chapter 4 a feature set which offers favorable separation of classes was found. Some current classification techniques may now be employed to develop a decision rule. In this chapter two classification techniques are discussed. A simple pattern recognition technique, the nearest neighbor rule, is described and its performance is compared with that of the maximum posterior decision rule. Neural networks are also discussed as a means for classification.

5.1 A Statistical Approach

A commonly used method of detection and classification is the hypothesis test. Let \mathbf{x} denote a feature vector extracted from the signal, $x[n]$. Let t_j be a possible classification label for the signal. Let the cost of choosing label t_i when t_j is correct be C_{ij} . Let the posterior probability of the label t_i being correct, given the parameter vector, be denoted by $P(t_i|\mathbf{x})$. The expected risk of choosing label t_i , given the parameter vector, is denoted by $R(t_i|\mathbf{x})$ and is calculated by

$$R(t_i|\mathbf{x}) = \sum_{j=0}^2 C_{ij} P(t_j|\mathbf{x}). \quad (5.1)$$

To minimize the risk, simply choose the label which gives the smallest expected risk.

Allowing no penalization for a correct decision and letting all possible errors be

penalized with the same cost gives

$$C_{ij} = \delta_{ij},$$

where

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}.$$

This reduces Equation (5.1) to

$$R(t_i|\mathbf{x}) = \sum_{j \neq i} P(t_j|\mathbf{x}). \quad (5.2)$$

The decision rule becomes: choose label t_k if, for all $t_i \neq t_k$,

$$\sum_{j \neq k} P(t_j|\mathbf{x}) \leq \sum_{j \neq i} P(t_j|\mathbf{x}). \quad (5.3)$$

Noting that each side of the above equation has all but one term in common simplifies the decision rule to: choose label t_k if, for all $t_i \neq t_k$,

$$P(t_k|\mathbf{x}) \geq P(t_i|\mathbf{x}). \quad (5.4)$$

The multi-hypothesis test which chooses the label to minimize the expected risk is simply choosing the label with the maximum posterior probability when all errors are penalized equally. This classifier is termed the MAP (maximum posterior) decision rule. See [13] for a detailed discussion of multi-hypothesis testing.

For the problem of this thesis, \mathbf{x} is the vector containing the dominant bin energies discussed in Chapter 4. The three choices ($c = 3$) for labels are $j = 0, 1, 2$ where $t_0 = \text{background noise}$, $t_1 = \text{whale click}$, and $t_2 = \text{snapping shrimp}$. In order to use the MAP decision rule with the biological transients discussed in Chapter 3, we would need to know the three posterior probabilities, $P(t_j|\mathbf{x})$, for each $j = 0, 1$ and 2. If these probabilities are not known, as is the case for this thesis, the posterior probabilities used in the MAP decision rule may be estimated from the sample data

set. One option is to assume the form of the underlying probability densities, $p(\mathbf{x}|t_j)$, so that only a few parameters need to be estimated from the data. In our case, the underlying densities are also unknown. Another option is to estimate the posterior probabilities at each possible value of \mathbf{x} from many samples of the data.

Using the MAP decision rule as a classifier for the biological transients would require that the posterior probabilities be estimated for each possible \mathbf{x} . Estimation of these probabilities from a sample signal set of the biological transient recordings can be carried out, but a simpler procedure that does not require knowledge of the probability distributions of the data exists. The nearest neighbor rule [14], a pattern recognition method of classification, closely compares to the MAP decision rule while avoiding explicit estimation of these probabilities. A nearest neighbor classifier is developed for the biological transients. The next section describes this rule and calculates the expected error bounds.

5.2 The Nearest Neighbor Rule

The nearest neighbor rule relies on the existence of a set of n classified samples. Denote this sample set by $X^n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, where each sample signal is represented by a parameter vector, \mathbf{x}_j , having k elements. The unclassified signal, represented by the parameter vector \mathbf{x} , is classified with the same label corresponding to its nearest neighbor in the sample set. Denoting the sample closest to \mathbf{x} by $\hat{\mathbf{x}}$ and its label by \hat{t} , the nearest neighbor rule classifies \mathbf{x} with the label \hat{t} .

The notion of “nearest”, in terms of smallest distance, leaves some choice for the determination of distance. For the nearest neighbor classifier in this thesis, we use the Euclidean distance, d_j , between the parameter vector from the unknown signal, \mathbf{x} , and each sample vector, \mathbf{x}_j .

$$d_j = \sum_{i=1}^k (\mathbf{x}[i] - \mathbf{x}_j[i])^2 \quad (5.5)$$

The following qualitative discussion offers an explanation for the MAP-like per-

formance of the nearest neighbor rule. The probability that $\hat{\mathbf{x}}$'s label, \hat{t} , is equal to any given label, t_j , is just the posterior probability $P(t_j|\hat{\mathbf{x}})$. Therefore, \mathbf{x} is classified by the label t_j with probability $P(t_j|\hat{\mathbf{x}})$. It is fair to assume that the distance between \mathbf{x} and its nearest neighbor is sufficiently small when the number of sample vectors, n , is very large. Additionally, assume that $P(t_j|\mathbf{x})$ is a smooth function of \mathbf{x} around each possible \mathbf{x} . In this case, $P(t_j|\hat{\mathbf{x}}) \approx P(t_j|\mathbf{x})$, thus, \mathbf{x} is classified by the label t_j with probability $P(t_j|\mathbf{x})$.

The MAP decision rule selects label t_j such that $P(t_j|\mathbf{x}) = \max_i P(t_i|\mathbf{x})$. For \mathbf{x} 's where the posterior probability is high for a given label, the nearest neighbor rule decision will often agree with the MAP decision. This agreement is due to the high likelihood of the sample points in the region around \mathbf{x} being labeled in agreement with the maximum posterior probability for that region. For \mathbf{x} 's where the posterior probabilities are similar, ($P(t_j|\mathbf{x}) \approx 1/c$, where c is the number of possible labels), the nearest neighbor rule and the MAP will typically not make the same choice, but the probability of error, $P(\epsilon)$, will be approximately the same ($P(\epsilon) \approx 1 - 1/c$).

Let $P_{NN}(\epsilon)$ be the probability of error of the nearest neighbor rule for very large n . Let $P_{MAP}(\epsilon)$ be the probability of error of the MAP decision rule. Derived in [14] and summarized in Appendix A, the probability of error of the nearest neighbor rule with c possible labels using a very large number of sample vectors in X^n is bounded by

$$P_{MAP}(\epsilon) \leq P_{NN}(\epsilon) \leq P_{MAP}(\epsilon) \left(2 - \frac{c}{c-1} P_{MAP}(\epsilon) \right). \quad (5.6)$$

5.3 Neural Networks

This section gives a rough overview of neural network theory. See [15] for a detailed tutorial on this subject. The neural network, like the nearest neighbor rule, does not need to make assumptions or find estimates for the posterior probabilities; it only depends upon the existence of a sample data set which has previously been classified. Using a layered neural network for pattern recognition begins by training the network on a portion of the previously classified sample data. Training is achieved by feeding

the neural network a training set comprised of these representative samples along with their correct output. The network automatically adjusts to minimize the error between the known outputs and the outputs that it is producing. Once the network performs satisfactorily on the training set, a different portion of the sample data, called the test set, is run through it to see how well it performs. If the network does not perform well enough on the test set, the training process may continue with the training set until better performance is obtained, if possible. This process is called supervised learning.

A layered neural network is made from connecting together many adaptive linear neurons (ALN). Each ALN performs a weighted linear combination of its inputs followed, typically, by a sigmoid or tanh function to produce a single output. A simple description of the training process in the case of the network containing only one ALN can easily be extended to the general case of the network containing many ALN's. A single ALN is capable of giving good classification for linearly separable input vectors having two possible labels from which to choose.

During training, the sample patterns (in our case, the feature vectors from Chapter 4) and their corresponding classification labels are fed to the ALN. Each label is typically represented by a binary number. The weights used when linearly combining the input are automatically adjusted according to an adaptive algorithm. This algorithm seeks to decrease the error between the resulting output and the desired output for each input by adjusting these weights. A commonly used algorithm which was chosen for the neural networks in this thesis is the least squares adaptation algorithm, sometimes called the Widrow-Hoff-Delta Rule [16]. This algorithm minimizes the sum of the squares of the errors for each input/output pair. The error is simply the difference between the actual output and the desired output. Many iterations with the same training set will be performed before the lowest possible error is achieved. Care should be taken not to over train a network. Once the performance levels off, further training may be hazardous to the network's ability to classify new data. If the training set is not an exhaustive representation of the unclassified data, an over trained network will not be robust to slight variations in the unclassified data. A

better error rate may also be obtained by changing the transform function at each ALN.¹

To attain better separability between the various classes of data, the ALNs are connected so that there is typically one to three layers, with each layer containing several ALNs in parallel. Each ALN in a single layer receives all the same inputs. The ALNs at the first layer receive the parameter set as input, the ALNs at the second layer receive the first layer outputs as input, etc.. This arrangement accommodates nonlinearly separable input vectors as well as linearly separable input vectors and allows for an unbounded number of outputs.

The exact neural networks used for classification of the biological transients are discussed in the next Chapter.

¹The topic of training networks is more thoroughly examined in [17].

Chapter 6

Test Results

Two different classification techniques, the nearest neighbor rule and neural networks, were tested on the biological transient data sets using the wavelet-packet-based feature set discussed in Chapter 4. These classification methods were discussed in Chapter 5. The nearest neighbor rule was implemented via a simple matlab program that has been included in Appendix B. The neural network tests were done using the Neuralware software package [17] for building, training, and analyzing a layered neural network. A back propagation network with a tanh nonlinearity and the Widrow-Hoff-Delta Rule adaptive weighting algorithm was used in all tests. The number of levels and ALNs used for the neural networks is detailed along with the results for each test.

This chapter discusses results of tests done on three versions of the data set. Each section reports on tests using one of these data sets. All three data sets consist of excerpts from underwater recordings of whale clicks, snapping shrimp and, in two cases, background ocean noise. The three data sets are summarized in Table 6.1. The original recordings were made available for this thesis by the Charles Stark Draper Laboratory and the Naval Underwater Systems Center (NUSC).

The first set of data, *data set I*, consists of 315 excerpts of underwater recordings of shrimp and whale clicks as summarized in Table 3.1. The shrimp recordings were done from roughly the same location and at the same time of day. Information concerning the time and location of the whale click recordings was not available. A training

Table 6.1: Three data sets used for testing.

	<i>Comments</i>	<i>Training set size</i>	<i>Test set size</i>
<i>data set I</i>	From Table 3.1	75	240
<i>data set II</i>	<i>data set I</i> appended with more shrimp	91	302
<i>data set III</i>	<i>data set II</i> without the noise excerpts	65	229

data set, made from 75 of these excerpts, was used for the training set of the nearest neighbor rule and the training set for the neural network. The test data set, made from the remaining 240 excerpts in the database, was used to test the performance of both methods of classification.

Two more data records for snapping shrimp from the NUSC database, NUSC2552 and NUSC2576, were included in the tests of Section 6.2. These data were recorded in a different region of the same ocean, many hours apart from the snapping shrimp data records from Table 3.1. Beyond this, the exact circumstances under which the data were taken are not known. *Data set II* was made by adding this new shrimp data to *data set I*. The training set for *data set II* was made by appending the training set for *data set I* with 16 of the new snapping shrimp excerpts. The test set for *data set II* was made by appending the test set for *data set I* with the remaining 62 snapping shrimp excerpts. Thus, the second set consists of a 91-excerpt training set and a 302-excerpt test set.

The third set of data, *data set III*, was made from *data set II* by omitting the samples that correspond to noise. Thus, *data set III* is comprised of only two classes of signals, whale clicks and snapping shrimp. The training set has 65 excerpts and the test set has 229 excerpts.

The tests use the wavelet-packet-based feature vectors discussed in Chapter 4 as input. Each excerpt of the biological transient recordings was transformed into a wavelet-packet-based normalized energy vector as described in Chapter 4. The steps are summarized below:

1. A six-level wavelet packet decomposition tree, discussed in Chapter 2, was found for each excerpt in the database.
2. An energy map, discussed in Section 2.2, was found for each wavelet packet decomposition tree from step 1.
3. As detailed in Section 4.2, each energy map was arranged into an energy vector having 63 elements with each element corresponding to a bin of the decomposition tree.
4. As discussed in Section 4.3, the 26 energy vectors (from *data set I*) for background noise were averaged to find an average noise energy vector.
5. Each energy vector was normalized by this average noise energy vector as done by (4.19).

6.1 Test Results using Data Set I

The nearest neighbor rule was run twice on *data set I*. The first run used the two-parameter feature set determined in Section 4.5 and resulted in correct classification for 97.92% of the test signals. The second run used the eleven-parameter feature set determined in Section 4.4 also resulted in correct classification for the same 97.92% of the test signals. These results are summarized in Table 6.2.

Both nearest neighbor rule tests resulted in the same performance, making the same errors. Gaining nothing by adding more features is not surprising because the analysis done in Chapter 4 determined that the energies from bins b(4,1) and b(4,2) of *data set I* were the dominant features necessary for distinguishing the three classes.

Table 6.2: Results obtained from the nearest neighbor rule using *data set I*.

Number of Features	2	11
Overall Classification (%) Number of Errors Made	97.92% 5	97.92% 5
Click Classification (%) Number of Errors Made	97.87% 2	97.87% 2
Shrimp Classification (%) Number of Errors Made	97.26% 2	97.26% 2
Noise Classification (%) Number of Errors Made	98.63% 1	98.63% 1

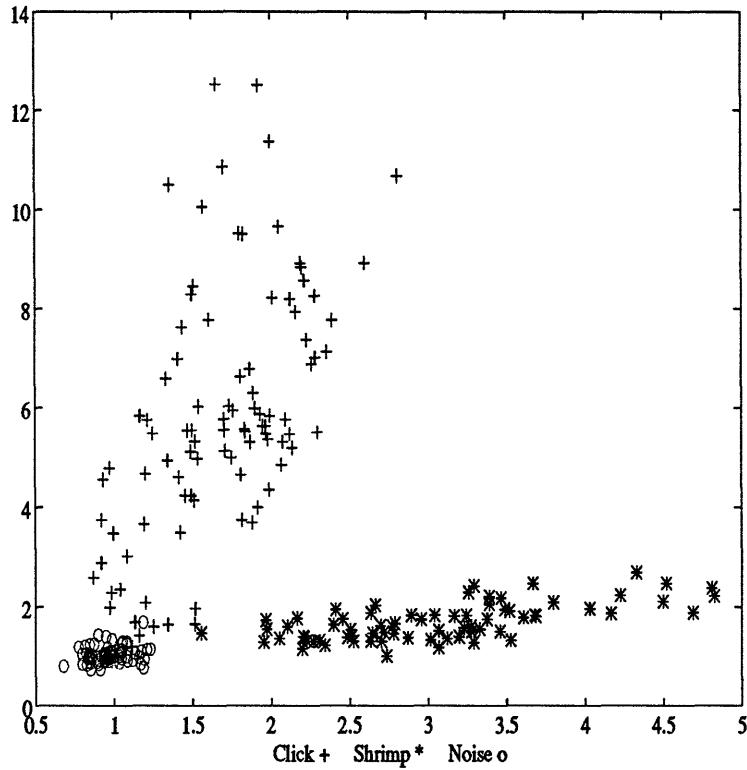


Figure 6-1: Noise normalized energies from bins $b(4,1)$ and $b(4,2)$ of the test set of energy maps for the excerpts of snapping shrimp, whale clicks, and background noise from *data set I*.

As shown by Figure 4-9 for the *data set I* training set and here in Figure 6-1 for the test set for *data set I*, adequate separation is obtained by using only two features.

Three neural networks were constructed for tests using two and eleven features. Excellent results were obtained for all tests. The networks (the number of ALNs in each layer) and their results are summarized in Table 6.3. The neural networks did an excellent job classifying the data from *data set I*. Classification ranged from 98.33% to 99.17%.

The signals that were incorrectly classified by the neural networks were a subset of the errors made by the nearest neighbor rules. The 5 signals that were incorrectly classified by the nearest neighbor rules were reduced to 4 of these 5 for the 2-input neural net, then reduced to 3 of these 4 by the one-layer 11-input neural net, then to

Table 6.3: Results obtained from the neural network using *data set I*.

Number of Inputs	2	11	11
Number of ALNs in Layer 1	3	7	7
Number of ALNs in Layer 2	0	0	3
Overall Classification (%)	98.33%	98.75%	99.17%
Number of Errors Made	4	3	2
Click Classification (%)	97.87%	98.44%	98.44%
Number of Errors Made	2	1	1
Shrimp Classification (%)	98.63%	98.63%	100%
Number of Errors Made	1	1	0
Noise Classification (%)	98.63%	98.63%	98.63%
Number of Errors Made	1	1	1

2 of these 3 by the two-layer 11-input neural net.

After training a neural network, the weights for the inputs to each ALN may be examined to see which network inputs were found to be most important in the classification process. For these neural networks large weights consistently appeared at the same five inputs of each ALN in the first layer. The weights assigned to the eleven inputs of the two neural networks discussed above showed that bins b(4,1), b(4,2), b(5,2), b(5,3), and b(5,4) were heavily stressed. Indexing into the WPD tree was detailed in Section 2.1.2. These are the same five bins that were determined significant by the analysis done in Section 4.5 and shown in Figure 4-6. Bins b(4,1) and b(4,2) are the bins that make up the two-parameter feature vector used in the 2-input neural network and the 2-feature nearest neighbor rule. Recall from Section 4.5, that the reason for choosing two out of five of the dominant bins was the recognition of parent-child redundancy in the wavelet packet transform. The goal was to represent the characteristics of the signals as simply as possible. Here, we see that only a slight gain in performance results from the addition of the child bins to the 2-parameter feature set.

The relative weighting of these inputs correspond to what the network found to be the relative importance of these bins. Ordering the bins from high to low gives: (1) b(5,2), (2) bins b(4,1) and b(5,3), (3) bin b(4,2), and (4) bin b(5,4). The neural network found one of the children bins to be most dominant, but also chose to stress all of the bins that were found to be dominant in Section 4.5.

6.2 Test Results using Data Set II

The second data set, *data set II*, includes snapping shrimp from a different time of day and region of ocean than the shrimp in *data set I*. Shown in Figure 6-2 are the short term Fourier transforms for two ten-second excerpts, one from recordings of the original snapping shrimp and one from this new snapping shrimp. From this figure we see that the primary frequency concentration for the new snapping shrimp data is higher than the primary frequency concentration for the original snapping shrimp

data.

The energy maps for a 4096-sample excerpt from each of these recordings along with a whale click are shown in Figure 6-3. Here, we note that the two snapping shrimp do not have similar energy distributions. For the original snapping shrimp, the energy begins to focus at the fourth level equally in bins $b(4,1)$ and $b(4,2)$ then focuses further at the fifth level in bin $b(5,2)$ with some in $b(5,3)$. For the new snapping shrimp, the energy also begins to focus at the fourth level equally, but this time in bins $b(4,2)$ and $b(4,3)$ then further concentrating in bins in $b(5,4)$ and $b(5,5)$, with some in $b(5,3)$. The whale click also begins to focus at the fourth level, only in bin $b(4,2)$ and then does not significantly focus further at the next level.

Repeating the SVD analysis done in Section 4.2 on both shrimp data may give a better view of which bins to consider as possible features. The 63-elements of the normalized singular vectors for whale clicks, the original shrimp, and the new shrimp are shown in Figure 6-4 and the 11-element singular vectors are shown in Figure 6-5. For the most part, each singular vector has its own significant elements in both figures. Comparing the distributions of the energy maps and the singular vectors that correspond to these energy maps indicates that a more in depth examination of the wavelet packet transforms would be beneficial. Classification to encompass these two types of shrimp may be better suited to observing the levels at which the first significant focusing of energy occurs in the energy maps. Perhaps the exploitation of time characteristics may be done if a windowed energy map (discussed in Section 2.2) is used. The incorporation of these and possibly other techniques is left for future work.

The cluster distributions for the two features (energies from bins $b(4,1)$ and $b(4,2)$) for the training data set of *data set II* are shown in Figure 6-6. Notice that the excerpts taken from the new shrimp data records forms a cluster to the left of the noise cluster. The two-feature training set for *data set II* forms two distinct clusters for shrimp, one on each side of the noise cluster. As should be expected by looking at the figure, this data was not classified correctly by both the two-feature and eleven-feature nearest neighbor rules for *data set I*. Additionally, the two-input and eleven-input neural

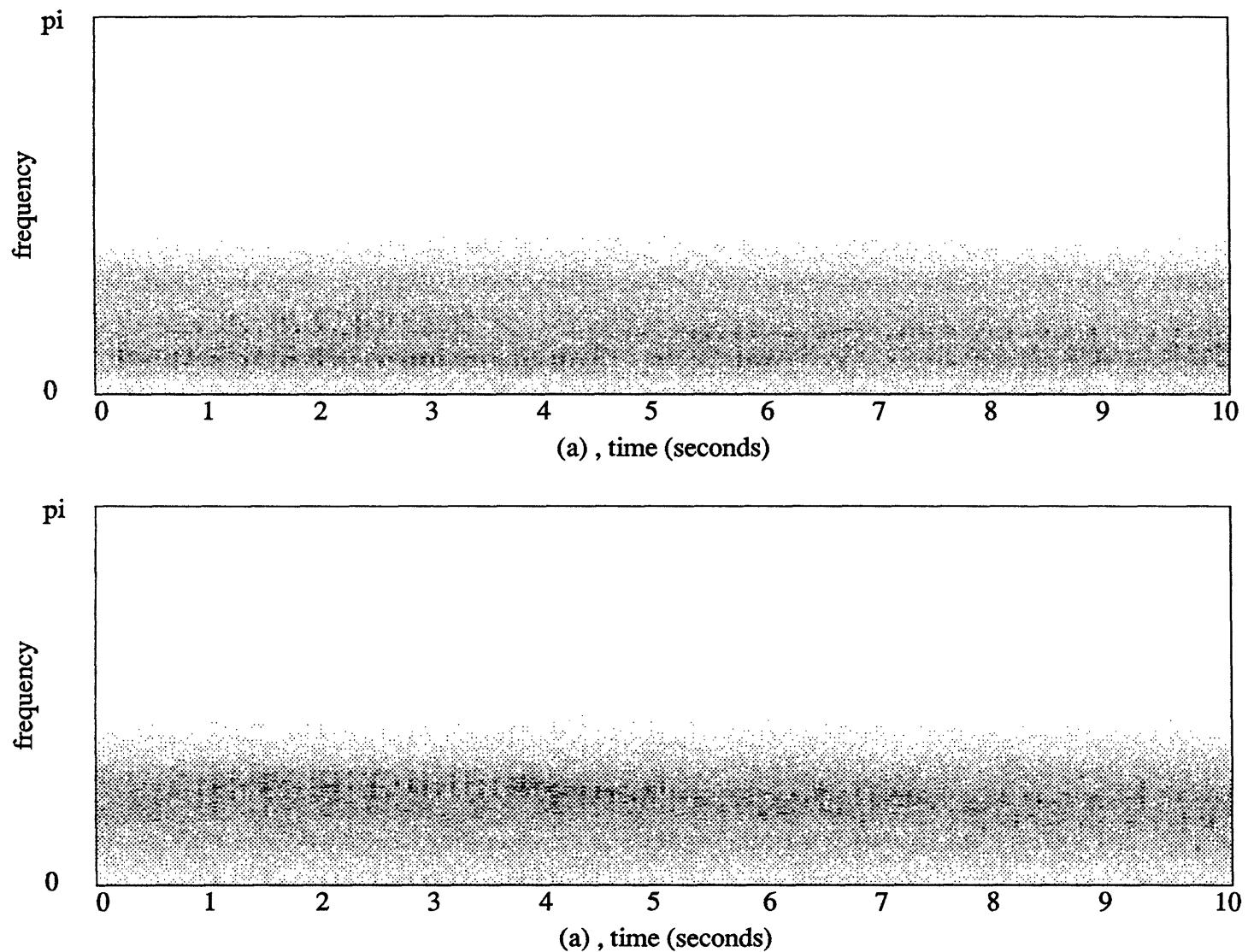
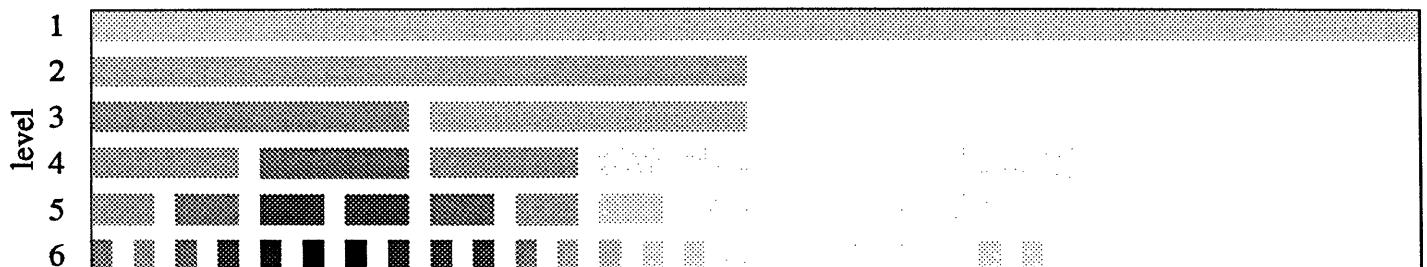
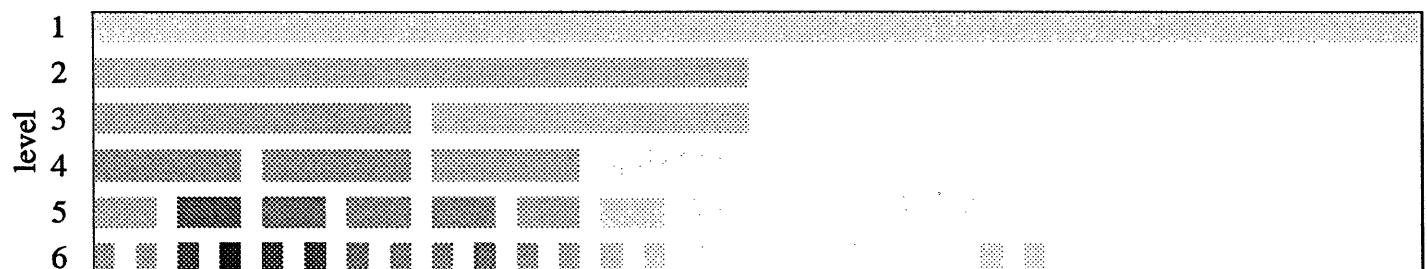


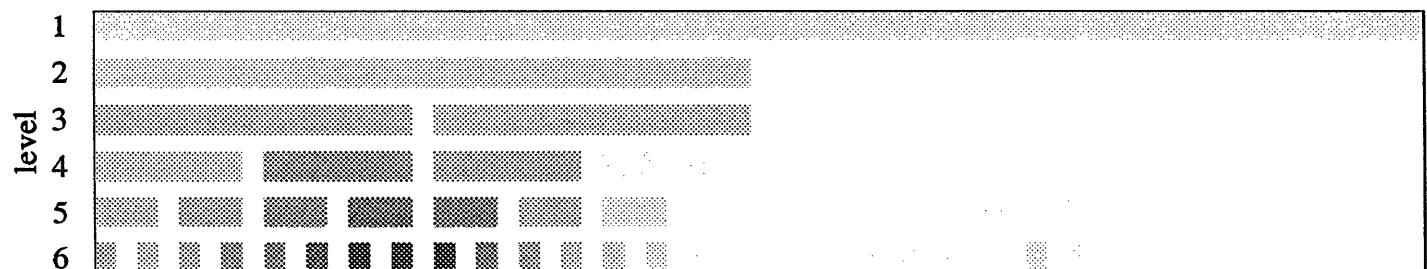
Figure 6-2: Short term Fourier transforms of both shrimp data. (a) Original Snapping Shrimp. (b) New Snapping Shrimp.



(a)



(b)



(b)

Figure 6-3: Energy maps of (a) whale clicks, (b) original snapping shrimp, and (c) the new snapping shrimp.

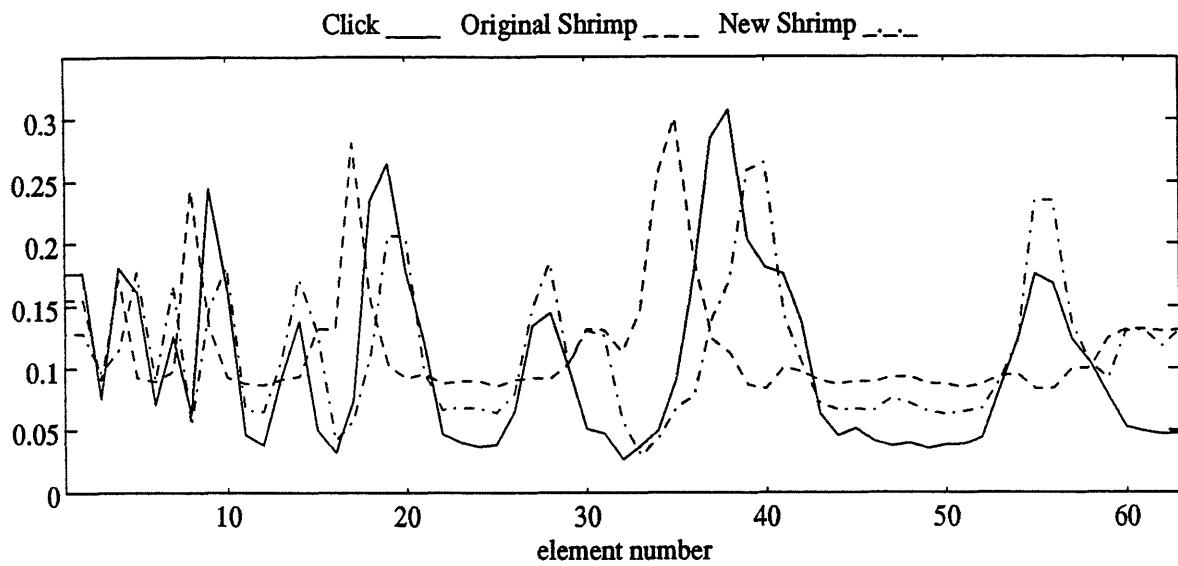


Figure 6-4: Singular vectors (63 elements) for both shrimp data.

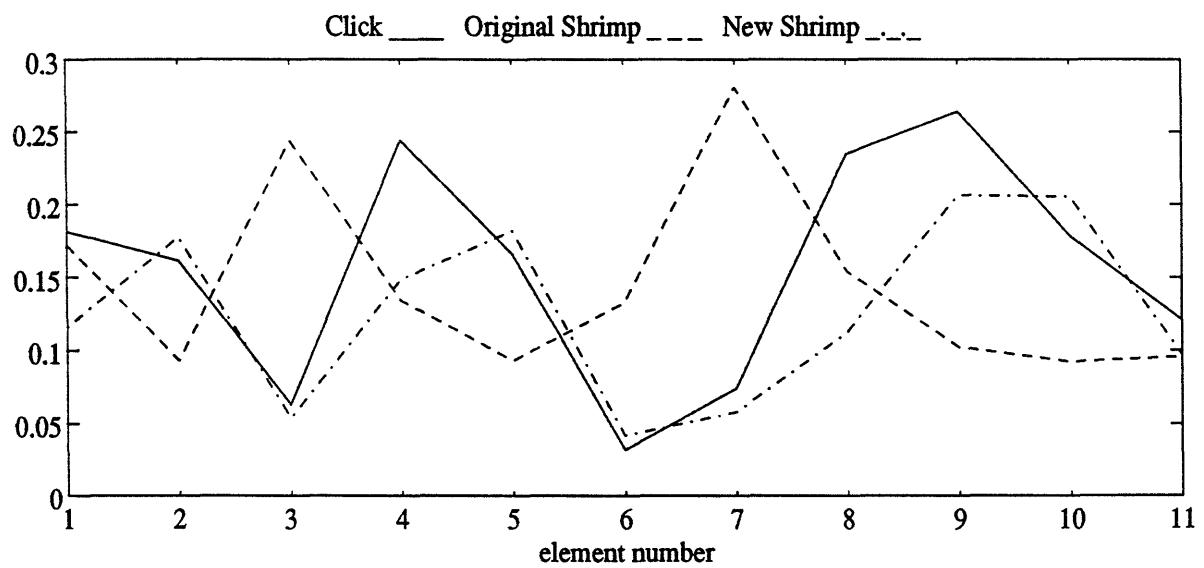


Figure 6-5: Singular vectors (11 elements) for both shrimp data.

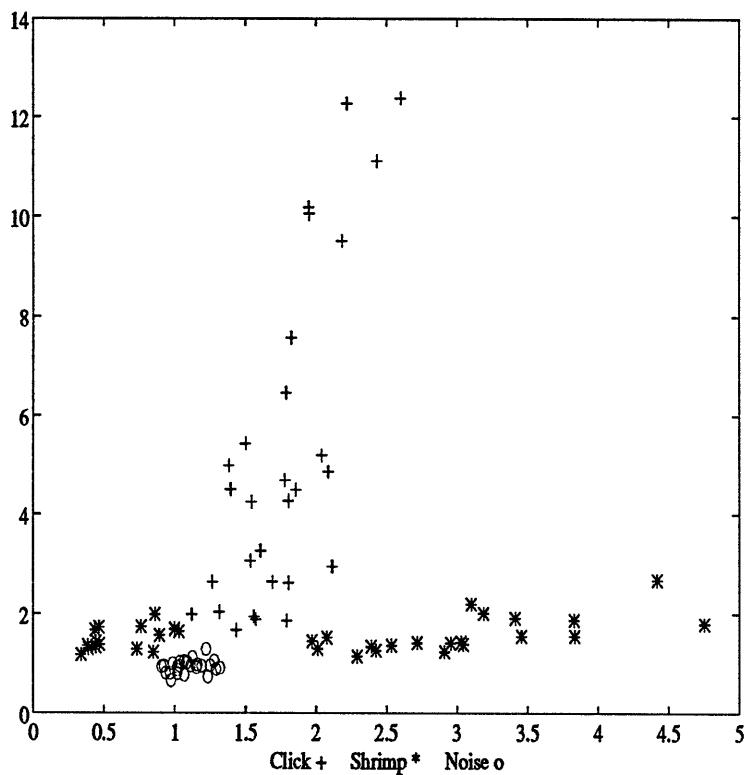


Figure 6-6: Noise normalized energies from bins $b(4,1)$ and $b(4,2)$ of the energy maps for the original data set of snapping shrimp, whale clicks, background noise and the new set of snapping shrimp.

networks from the previous section, trained without samples of this new shrimp data, were not capable of correct classification.

Both the nearest neighbor rule and the neural network tests were repeated using the training and test data sets for *data set II* that included the new shrimp data. When including these new data in the training and test sets, we expect that the same two energy bins used for the tests of the original data may not give the best performance and that adding more bin vectors may give better classification.

Results of the nearest neighbor tests are summarized in Table 6.4. The two-feature nearest neighbor rule classified 93.05% of the 302 test vectors correctly. Specifically, 91.94%, or all but 5, of the 62 samples of new shrimp data were classified correctly. The eleven-feature nearest neighbor rule classified 94.37% of the 302 test vectors correctly, where 88.71%, or all but 7, of the 62 samples of the new shrimp were correctly classified. The nearest neighbor algorithm gave a slightly better overall performance on the same data when more bin energies were used, but classified 3.23% fewer of the new shrimp correctly. The gain in performance by including additional features was in the classification of clicks and noise. The nearest neighbor rule did a better job separating these three classes by using the eleven-parameter feature vectors than by using the two-parameter feature vectors. As predicted, the two features that were found to be best for classification of *data set I* did not serve as the best features for classification of the more diverse *data set II*.

Comparing the performance of the nearest neighbor rule tests from this and the previous sections on the original data from *data set I*, we find that the tests on *data set II* resulted in a higher percentage of error for the whale clicks and background noise. Not surprisingly, the same 5 signals that were incorrectly classified by the nearest neighbor rules in the previous section were also incorrectly classified by this test. The new cluster of shrimp presented some additional confusion in whale click and noise classification, but no additional confusion occurred for the shrimp from *data set I*. The potential for additional click and noise errors is suggested by the two-feature cluster plot in Figure 6-6 because the new shrimp cluster is adjacent to both the click and noise clusters.

Table 6.4: Results obtained from the nearest neighbor rule using *data set II*.

Number of Features	2	11
Overall Classification (%) Number of Errors Made	93.05% 21	94.37% 17
Click Classification (%) Number of Errors Made	94.68% 5	96.80% 3
Shrimp Classification (%) Number of Errors Made old shrimp + new shrimp	94.81% 2 + 5	93.33% 2 + 7
Noise Classification (%) Number of Errors Made	87.67% 9	93.15% 5

The results of the neural network tests are summarized in Table 6.5. The two-input neural networks resulted in good overall performance of 91.39%–91.72%, with perfect classification (100%) of noise, good classification (90.43%) of whale clicks, and fair classification (87.14%–88.15%) of shrimp. When comparing the neural networks for *data set I* shown in Table 6.3 and *data set II* shown in Table 6.5, the degradation of performance is largest for the classification of snapping shrimp which went from 98% for the two-input networks of the previous section to 87% for these two-input networks.

The eleven-input neural network tests resulted in a much lower overall performance ranging from 77.81% to 79.14%. The eleven-input neural networks are more complex than the two-input networks; the weight adjustments during training appeared to result in the network finding a local minimum of error between the desired output and the network output. These networks were able to correctly classify 98.63% of the noise, around 86% of the clicks, and only 62% of the shrimp. They did a poor job classifying both the original shrimp and the new shrimp.

As found with all the tests up to this point, many of the same signals were incorrectly classified by all the tests. The sets of noise and click signals that were incorrectly classified by the neural networks from Table 6.5 are subsets of one another. The sets of incorrectly classified shrimp exhibit a great deal of overlap. The same signals that were incorrectly classified by the neural networks in Section 6.1 were also incorrectly classified by these neural networks.

Unlike the nearest neighbor rules for *data set II*, the neural networks were unable to sufficiently separate shrimp from noise. The two shrimp clusters on either side of the noise cluster created confusion. Perhaps this problem would be solved by separating the shrimp into two classes, allowing for the neural network to distinguish between the two.

The first layer of these 11-input neural networks was examined to find that the same five bins that were weighted heavily by the networks in Section 6.1 were also stressed in this case. In the previous section all ALNs at the first level showed high weights for all five of these inputs, but for the *data set II* neural networks there was

Table 6.5: Results obtained from the neural network using *data set II*.

Number of Inputs	2	2	11	11	11
Number of ALNs in Layer 1	3	3	7	7	7
Number of ALNs in Layer 2	0	3	0	3	7
Overall Classification (%)	91.39%	91.72%	77.81%	79.14%	78.48%
Number of Errors Made	26	25	67	63	65
Click Classification (%)	90.43%	90.43%	85.11%	87.23%	87.23%
Number of Errors Made	9	9	14	12	12
Shrimp Classification (%)	87.14%	88.15%	61.48%	62.96%	61.48%
Number of Errors Made	10 + 7	9 + 7	12 + 40	26 + 24	20 + 32
old shrimp + new shrimp					
Noise Classification (%)	100%	100%	98.63%	98.63%	98.63%
Number of Errors Made	0	0	1	1	1

Table 6.6: Results obtained from the nearest neighbor rule using *data set III*.

Number of Features	2	11
Overall Classification (%) Number of Errors Made	96.07% 9	96.51% 8
Click Classification (%) Number of Errors Made	94.68% 5	96.81% 3
Shrimp Classification (%) Number of Errors Made old shrimp + new shrimp	97.04% 2 + 2	96.30% 2 + 3

less consistency; the majority of the ALNs stressed these five inputs, but not every ALN stressed all five.

6.3 Test Results Using Data Set III

It is reasonable to assume that another method, such as total energy detection, may be employed to detect the presence of the transients. In this case, it would be necessary to distinguish between only two classes, whale clicks and snapping shrimp. The nearest neighbor rule and the neural network tests were repeated for the two-class *data set III*.

Table 6.6 summarizes the result of the nearest neighbor tests done on *data set III*. The set of signals that caused errors for the the 11-feature rule is contained in the set of signals that caused errors for the 2-feature rule. This has been the case for all nearest neighbor rule tests in this chapter.

When comparing these nearest neighbor tests with the previous ones, we find that the whale clicks and the new snapping shrimp that were incorrectly classified by the

tests in this section were also incorrectly classified by the tests of the first two sections. The errors that the tests of this section made on the original snapping shrimp signals were different from the errors made in the previous two sections.

Ordering the performances of nearest neighbor rules from the three sections find the best performance for *data set I*, a slightly diminished performance was for *data set III*, and a further diminished performance was for *data set II*. This ordering from best to worst could have been predicted by looking at the cluster plots of Figures 6-1 and 6-6. From these figures we observe that there is very good class separation for *data set I* and the principle source of confusion in *data set II* is due to the close proximity of cluster borders. Much of the potential for confusion is removed by eliminating the noise cluster in *data set II* to get *data set III*. As can be seen from Figure 6-6, the noise cluster that was omitted was bordering the new shrimp cluster. Thus, it is not surprising that the nearest neighbor rule for *data set III* does a better job when classifying the signals that previously bordered the noise cluster (the new shrimp), but does not improve classification for whale clicks and the original shrimp signals.

The results of the neural network tests are summarized in Table 6.7. Again, the errors made by these five neural networks were consistent. In fact, the three 11-input tests incorrectly classified the same signals. The 11-input neural networks overall did a slightly better job than the 2-input neural networks for *data set III*. There was some trade-off in classification errors when going from two inputs to eleven; the 11-input networks incorrectly classified 3–5 more clicks and 6–15 less shrimp than the 2-input neural network had. The overall performance of these neural networks on *data set III* ranged from 89.52% to 94.32%.

The nearest neighbor rule tests gave a better overall performance than the neural networks for *data set III*. The nearest neighbor rules did much better in click classification but slightly worse in shrimp classification.

The neural networks in this section did not have to distinguish shrimp from noise and, as should be expected, do a much better job than the neural networks of the previous section that were confused by the arrangement of noise and shrimp clusters. The overall performances of the neural networks from each of the three sections was

Table 6.7: Results obtained from the neural network using *data set III*.

Number of Inputs	2	2	11	11	11
Number of ALNs in Layer 1	3	3	7	7	7
Number of ALNs in Layer 2	0	3	0	3	7
Overall Classification (%)	89.52%	92.58%	94.32%	94.32%	94.32%
Number of Errors Made	24	17	13	13	13
Click Classification (%)	92.55%	90.43%	88.30%	88.30%	88.30%
Number of Errors Made	7	9	11	11	11
Shrimp Classification (%)	87.41%	94.07%	98.52%	98.52%	98.52%
Number of Errors Made old shrimp + new shrimp	3 + 14	0 + 8	0 + 2	0 + 2	0 + 2

compared and the ordering was the same as for nearest neighbor rule. The neural networks from Section 6.1 for *data set I* performed the best, Section 6.3 for *data set III* second best, and Section 6.2 for *data set II* the worst. As was suggested in the previous section, the neural network may make less errors if the two shrimp clusters were given two different classification labels.

When examining the weights assigned to the inputs of each ALN in the first layer of the 11-input networks, only two weights were consistently high. These high weights correspond to bins b(4,2) and b(5,3). This result is noteworthy because it doesn't coincide with the findings of Section 4.5 as did the weight examinations in the previous two sections. Although this result is interesting, it is not surprising because the addition of new shrimp to the original data set is expected to alter the results of the analysis done in Section 4.5.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This work has explored the feasibility of applying the wavelet packet transform to the detection and classification of transient signals in background noise. This examination was done through the development of various automatic classifiers using wavelet-packet-based features for biologically generated underwater acoustic transient signals in ocean noise. Class-specific differences in the wavelet packet transform coefficients were exploited by the choice of features extracted from the wavelet packet transforms of many sample signals. Using these features with several pattern recognition techniques showed excellent results.

A database of many occurrences of snapping shrimp and whale clicks in their natural habitat, discussed in Chapter 3, was furnished by the Charles Stark Draper Laboratory and the Naval Underwater Systems Center. These transients ranged from 1 millisecond to 150 milliseconds in duration. From this database, a sample set and a test set containing excerpts of whale clicks, snapping shrimp, and background noise were made, and the wavelet packet decomposition of each signal was performed. Analysis of the wavelet packet decompositions was simplified by transforming them into energy maps, discussed in Chapters 2 and 3.

The formulation of a wavelet-packet-based feature set, specific to the classification of these snapping shrimp and whale clicks, focused on the exploitation of class-specific

differences obtained through careful examination of the feature separation attainable from the wavelet packet decomposition of the transients. From singular value decomposition of matrices made from the energy maps of these data, it was determined that only two wavelet-packet-based features were necessary to distinguish among snapping shrimp, whale clicks, and background noise. This work is detailed in Chapter 4.

Two methods of classification, the nearest neighbor rule and neural networks, were used with the wavelet-packet-based features. Both the nearest neighbor rule and the neural network tests done in this thesis resulted in excellent classification of the biological transient signals using the wavelet-packet-based features developed in Chapter 4. The best performance (99.17%) was obtained when using a sample and test set that were comprised of background noise, whale clicks, and snapping shrimp, each collected in one region. A few tests were done with some snapping shrimp data that were collected in a different region at a different time of day from the shrimp of the original data set. The nearest neighbor rule and neural networks developed for the original data set were unable to correctly classify this new shrimp data. From the Fourier transforms of these new shrimp data a slight shift in dominant frequencies was apparent. A project for future work would be to extract wavelet-packet-based features that would unify signals for this case where similar characteristics exist for signals within a class, but the median frequencies differ among occurrences. Including portions of the new shrimp data in the sample set and the test set resulted in 94.37% correct classification. When distinguishing between two classes, snapping shrimp and whale clicks (no background noise), the overall performance improved from 94.37% to 96.57%. As seen from running each test with two features, and again with eleven, most of the information necessary for classification of these transients is found in the two energy bins that were derived in Chapter 4. In some cases, a slightly better performance was obtained by adding nine more prominent wavelet-packet-based features to the feature set.

The work done in this thesis has shown the wavelet packet transform to be a useful tool in detecting and classifying transient signals through the formulation of a wavelet-packet-based feature set used for classification of biologically generated

underwater acoustic transient signals.

7.2 Suggestions for Future Work

The scope of this thesis was restricted to the classification of whale clicks and snapping shrimp by analysis of a limited data set. A more inclusive analysis could be done using a well documented database that was taken from many parts of the world during many seasons that includes other underwater transient signals such as lobster clicks and porpoise clicks. This may lead to an algorithm that uses some external parameters such as season, time of day, depth, and climate, to determine which wavelet-packet-based features to use for classification. The energy map used for feature extraction in this thesis may need to be altered to retain more time information if more classes are added to the classification problem. The use of a windowed energy map was briefly discussed in Chapter 2.

An automated metric for the determination of wavelet-packet-based features would be useful. This would allow the steps taken in this thesis to be performed without human interaction; one could simply input a sample data set that represents the signals of interest and receive as output the dominant features, the most suitable energy map, and the classification algorithm.

The above suggested future work directly builds on the direction of this thesis, but a few issues were left unaddressed. The effect of time shifting a transient signal may be quite significant when using the discrete wavelet packet transform as a tool in classification. The motivation for such an investigation was mentioned in the previous section; the features extracted for classification separated the shrimp into two distinct groups and presented some confusion for the decision rules. The wavelet packet transform, being a time-frequency transform, is not invariant to time or frequency shifts. The same signal shifted by an amount that is not resolvable by the discrete wavelet packet transform does not simply shift the coefficients of the transform – it changes them. Some signals of a single class may possess time-frequency information that may be easily exploited if the effects of time and frequency shifts

on the wavelet packet transform are better understood. In addition, the choice or development of filters used to calculate the transforms deserves some investigation. Once the characteristics of the signals of interest are determined, the filters may be chosen or developed not to obstruct the enhancement of class-separable information.

Appendix A

Caclulation of the Error Bounds on the Nearest Neighbor Rule

This appendix offers a derivation of the bounds on the probability of error, $P_{NN}(\epsilon)$, for the nearest neighbor rule. A bit more detailed derivation appears in [14].

Before searching for the bounds on $P_{NN}(\epsilon)$, it is helpful to derive the expression for it. The performance of the nearest neighbor rule depends upon the particular sample set used because different sample sets will produce different nearest neighbors. We must, therefore, begin with the joint conditional probability $P(t, \hat{t}|\mathbf{x}, \hat{\mathbf{x}})$, where t is the label associated with \mathbf{x} and \hat{t} is the label associated with its nearest neighbor, $\hat{\mathbf{x}}$.

It is reasonable to assume that each parameter vector in the sample set is independent from the signal parameter. This allows us to write

$$P(t, \hat{t}|\mathbf{x}, \hat{\mathbf{x}}) = P(t|\mathbf{x}) P(\hat{t}|\hat{\mathbf{x}}). \quad (\text{A.1})$$

The nearest neighbor rule makes an error whenever $t \neq \hat{t}$. Thus, the average conditional probability of error of the nearest neighbor rule is

$$P_{NN}(\epsilon|\mathbf{x}, \hat{\mathbf{x}}) = 1 - \sum_{i=1}^c P(t = t_i|\mathbf{x}) P(\hat{t} = t_i|\hat{\mathbf{x}}). \quad (\text{A.2})$$

In finding the unconditional probability of error, $P_{NN}(\epsilon)$, the first step is to cal-

culate $P_{NN}(\epsilon|\mathbf{x})$, by integrating out the dependence on $\hat{\mathbf{x}}$ from (A.2).

$$P_{NN}(\epsilon|\mathbf{x}) = \int P_{NN}(\epsilon|\mathbf{x}, \hat{\mathbf{x}}) p(\hat{\mathbf{x}}|\mathbf{x}) d\hat{\mathbf{x}}. \quad (\text{A.3})$$

If the number of sample points is large, $\hat{\mathbf{x}}$ and \mathbf{x} are assumed to be very close. We expect that the probability of the nearest neighbor, $\hat{\mathbf{x}}$, being very close in vicinity to \mathbf{x} is approximately one and that the probability of it being outside of this vicinity is approximately zero. This allows us to approximate $p(\hat{\mathbf{x}}|\mathbf{x})$ by a delta function, $\delta(\mathbf{x} - \hat{\mathbf{x}})$. Substituting (A.2) into (A.3) and replacing $p(\hat{\mathbf{x}}|\mathbf{x})$ with $\delta(\mathbf{x} - \hat{\mathbf{x}})$ gives

$$P_{NN}(\epsilon|\mathbf{x}) = 1 - \sum_{i=1}^c P^2(t_i|\mathbf{x}). \quad (\text{A.4})$$

One more integration gives us the unconditional error probability for the nearest neighbor rule.

$$P_{NN}(\epsilon) = \int P_{NN}(\epsilon|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \int \left[1 - \sum_{i=1}^c P^2(t_i|\mathbf{x}) \right] d\mathbf{x} \quad (\text{A.5})$$

The lower bound is easily found. The minimum possible value of $P_{NN}(\epsilon)$ would occur when $P_{NN}(\epsilon|\mathbf{x})$ is a minimum for each \mathbf{x} . As (A.4) reveals, $P_{NN}(\epsilon|\mathbf{x})$ is minimized when $P(t_i|\mathbf{x})$ is a maximum for each \mathbf{x} . The MAP decision rule does exactly this, it chooses the label for the vector \mathbf{x} to maximize $P(t_i|\mathbf{x})$. Thus, the lower bound on $P_{NN}(\epsilon)$ is simply the error rate of the MAP, $P_{MAP}(\epsilon)$.

The upper bound requires a bit more work. Again, from (A.4), $P_{NN}(\epsilon|\mathbf{x})$ is maximized when $\sum_{i=1}^c P^2(t_i|\mathbf{x})$ is minimized. Rewrite $\sum_{i=1}^c P^2(t_i|\mathbf{x})$ as

$$\sum_{i=1}^c P^2(t_i|\mathbf{x}) = P^2(t_m|\mathbf{x}) + \sum_{i \neq m} P^2(t_i|\mathbf{x}), \quad (\text{A.6})$$

where t_m is the label for \mathbf{x} having the highest probability. This allows us to make comparisons with the MAP decision rule because the conditional probability of error of the MAP decision rule is $P_{MAP}(\epsilon|\mathbf{x}) = 1 - P(t_m|\mathbf{x})$. The summation in A.6 is minimized when each of the terms being summed is equal, that is, $P(t_i|\mathbf{x}) = P(t_j|\mathbf{x})$

for all $i, j \neq m$. The above observations give the following values of $P(t_i|\mathbf{x})$ for minimizing A.6:

$$P(t_i|\mathbf{x}) = \begin{cases} 1 - P_{MAP}(\epsilon|\mathbf{x}), & i = m \\ \frac{P_{MAP}(\epsilon|\mathbf{x})}{c-1}, & i \neq m \end{cases} \quad (\text{A.7})$$

Substituting (A.7) into (A.6) gives the following inequality:

$$\sum_{i=1}^c P^2(t_i|\mathbf{x}) \geq (1 - P_{MAP}(\epsilon|\mathbf{x}))^2 + \sum_{i \neq m} \frac{P_{MAP}^2(\epsilon|\mathbf{x})}{(c-1)^2} \quad (\text{A.8})$$

Carrying out the multiplication gives

$$\sum_{i=1}^c P^2(t_i|\mathbf{x}) \geq 1 - 2P_{MAP}(\epsilon|\mathbf{x}) + P_{MAP}^2(\epsilon|\mathbf{x}) + \frac{P_{MAP}^2(\epsilon|\mathbf{x})}{(c-1)}, \quad (\text{A.9})$$

and combining like terms gives

$$\sum_{i=1}^c P^2(t_i|\mathbf{x}) \geq 1 - 2P_{MAP}(\epsilon|\mathbf{x}) + \frac{c}{c-1} P_{MAP}^2(\epsilon|\mathbf{x}). \quad (\text{A.10})$$

Substituting this into (A.4) we get

$$P_{NN}(\epsilon|\mathbf{x}) \leq 2P_{MAP}(\epsilon|\mathbf{x}) - \frac{c}{c-1} P_{MAP}^2(\epsilon|\mathbf{x}). \quad (\text{A.11})$$

We integrate out the dependence on \mathbf{x} in (A.11) to get the error bound on the unconditional probability of error.

$$P_{NN}(\epsilon) \leq 2P_{MAP}(\epsilon) - \frac{c}{c-1} \int P_{MAP}^2(\epsilon|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}. \quad (\text{A.12})$$

Using

$$\text{Variance}(P_{MAP}(\epsilon|\mathbf{x})) = \int [P_{MAP}(\epsilon|\mathbf{x}) - P_{MAP}(\epsilon)]^2 p(\mathbf{x}) d\mathbf{x} \geq 0 \quad (\text{A.13})$$

to get

$$\int P_{MAP}^2(\epsilon|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \geq P_{MAP}^2(\epsilon) \quad (\text{A.14})$$

allows us to show that

$$P_{MAP}(\epsilon) \leq P_{NN}(\epsilon) \leq 2P_{MAP}(\epsilon) - \frac{c}{c-1}P_{MAP}^2(\epsilon). \quad (\text{A.15})$$

Appendix B

Matlab Code

This appendix includes many of the Matlab functions that were written and used for this thesis.

This Matlab program performs transformation from a wavelet packet decomposition tree into an energy map as described in Section 2.2 and (2.2).

```
function [et]=emapN(tree,n>windowlength)
%function [et]=emapN(tree,n>windowlength)
%calculates the windowed mean square 1/N(2normsquare(nodevector)
%of each node in the binary tree (by overlapping window by 1/2)
%down to the level which has 2^n samples per vector
>windowlength needs to be a power of 2

[ht wdth]=size(tree);

if(ht < (log2(wdth) + 1 - n) ), %use the smaller of the two
    eheight = ht;
else,
    eheight = log2(wdth) + 1 - n;
end
```

```

if (n==0),
    eheight = log2(wdth) ; %can't go to bottom level, no overlap
end

numwindows = wdth/windowlength *2 -1; %number of windows per node vector
                                    %inlcucing overlappped ones

ewdth = wdth/(2^(n)) *(numwindows);

et = zeros(eheight,ewdth);

%initialiaze num of elements devoted to one eval
elength = ewdth/numwindows ;

%for each level in tree that will be reduced into an energy
for i=1:(eheight),
    estrt = 1; %starting position in et
    eend = elength;
    %starting element of kth node vector in tree at level i
    windowstrt = 1;
    windowend = windowlength;
    for k=1:2^(i-1), %for each node in tree at level i
        for q=1:numwindows, %for each window in node (including overlapped ones)
            et(i,estrt:eend)=...
                norm(tree(i,windowstrt:windowend))^2/windowlength *ones(1,elength);
            estrt = estrt + elength;
            eend = eend + elength;
            windowstrt = windowstrt + windowlength/2;
            windowend = windowend + windowlength/2;
        end
        windowstrt = windowstrt + windowlength/2;
    end
end

```

```

windowend = windowend + windowlength/2;
end
windowlength = windowlength/(2); %window length is cut in half at next level
elength = elength/2;
end

```

This Matlab program reads in the wavelet packet decomposition tree and puts it in Paley order as discussed in Section 2.1.1.

```

function swaptree=wptswap(tree);
%function swaptree=wptswap(tree)
%Swaps children of every highpassed-downsampled node to
%reverse the effect of aliasing.

[ht wd]=size(tree);
flag=[0 1];
parentnodeLength=wd;
swaptree=tree;
i=0;
for(i=3:ht), %for every level needing swaps (1st and second need none)

parentnodeLength=parentnodeLength/2;
nodeLength = parentnodeLength/2;
n=2^(i-2); %number of node-pairs at level i
swapList=[1:wd];

% find starting positions for node-pairs to be swapped
% (left node of pair)
x = flag .* (parentnodeLength .* [0:n-1] + 1);

```

```

%put left children node positions in a temp var swapright
%put right children node positions to left temp variable swapleft
for (k=1:n),
    if x(k)>0,
        swapright = swaplist(x(k):x(k)+nodelength-1);
        swapleft = swaplist(x(k)+nodelength:x(k)+parentnodelength-1);
        swaplist(x(k):x(k)+nodelength-1)=swapleft;
        swaplist(x(k)+nodelength:x(k)+parentnodelength-1) = swapright;
    end
end

%D0 the swap from this level down
swaptree(i:ht,:)=swaptree(i:ht,swaplist);

flag=[flag,flag];

end

```

This Matlab program implements the nearest neighbor rule described in Section 5.2.

```

function [success,Ce,Se,Ne,error_index] = nnrule(test,train)
%function [success,Ce,Se,Ne,error_index] = nnrule(test,train)
% a row r of train or test might be train(r,:) = [0.5 8.9 4.2 100]
%where the last element is the label

[m,n]=size(train);
[i,j]=size(test);

%find inner product between each sample vector and each test vector

```

```

errorvectors=...

[kron(ones(i,1),train(:,1:n-1))-kron(test(:,1:j-1),(ones(m,1)))];
innerproducts=sum(errorvectors' .* errorvectors')';

squaredistances=reshape(innerproducts,m,i);

% for each test vector (each column) find the index of the
% min element of each row
[temp,index]=min(squaredistances);

%pick off the label for each test point
decision=train(index,n);

%put a one in every position that there is a disagreement
errors = (decision~=test(:,j));

pbe=sum(errors)/i;

%find out how many of each kind of errors were made
kind= errors .* test(:,j);
count=[1:i];
error_index= count(errors);
Ce=sum(kind==1);
Se=sum(kind==10);
Ne=sum(kind==100);
success =1-pbe;

```

Bibliography

- [1] M. V. Wickerhauser. “Lectures on Wavelet Packet Algorithms”. Technical report, Washington University, Department of Mathematics, 1992.
- [2] A. Grossman and J. Morlet. “Decompositions of Hardy functions into square integrable wavelets of constant shape”. *SIAM J. Math.*, 15:723–736, 1984.
- [3] A. Lemer J. Nicolas and D. Legitimus. “Identification automatique de bruits impulsifs en acoustique sous-marine par reseaux multi-couches”. In *International workshop on neural networks and their applications*, Nimes, France, Nov. 1989.
- [4] M. Desai and D. Shazeer. “Acoustic transient analysis using wavlet decomposition”. In *Proceedings of the IEEE conference on neural networks for ocean engineering*, Wash. DC, Aug. 1991.
- [5] A. Grossman P. Goupillaiud and J. Morlet. “Cycle-ocatve and related transforms in seismic signal analysis”. *Geoexploration*, 23:85–102, 1985.
- [6] Y. Meyer. “Ondelettes et functions splines”. In *Sem. Equations aux Dérivées Partielles.*, Ecole Polytechnique, Paris, France, Dec 1986.
- [7] I. Daubechies. “Orthonormal bases of compactly supported wavelets”. *Commun. Pure Appl. Math.*, 41:909–996, Nov. 1988.
- [8] S. Mallat. “A theory for multiresolution signal decomposition: The wavelet representation”. *IEEE Trans. Pattern Anal. Machine Intell.*, 11:674–693, July 1989.

- [9] Olivier Rioul and Martin Vitterli. Wavelets and signal processing. *IEEE-SP Magazine*, pages 14–38, Oct. 1991.
- [10] Gilbert Strang. Wavelets and dilation equations: A brief introduction. *SIAM Review*, 31(4):614–627, Dec. 1989.
- [11] The Math Works Inc., Cochituate Place, 24 Prime Park Way, South Natick, MA 01760. *Matlab*, 1991.
- [12] Gilbert Strang. *Linear algebra and its applications*. Harcourt Biaci Jovanovich, 4 edition, 1988.
- [13] Harry L. van Trees. *Detection, estimation, and modulation theory, Part I*. John Wiley and Sons, 1968.
- [14] Richard O. Duda and Peter E. Hart. *Pattern classification and scene analysis*. John Wiley and Sons, 1973.
- [15] Bernard Widrow, Rodney G. Winter, and Robert A. Baxter. “Layered Neural Nets for Pattern Recognition”. *IEEE Trans. on ASSP*, 36(7):1109–1118, 1988.
- [16] B. Widrow and Jr. M. E. Hoff. “Adaptive switching circuits”. In *IRE WESCON conv. Rec.*, 1960.
- [17] NeuralWare Inc., Penn. Center West, Pittsburgh, PA 15276. *Neural Works Professional II PLUS and Neural Works Explorer*, 1990.