# INFORMATION TO USERS

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

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

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

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

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

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600

UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

# DESIGN OF OPTIMAL SUBBAND FILTER BANKS FOR IMAGE DISCRIMINATION

A Dissertation

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

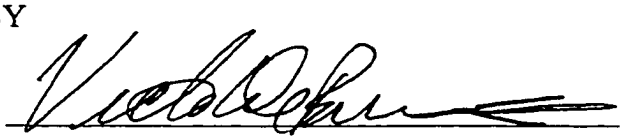
By

MADHAVI KADIYALA
Norman, Oklahoma
1999

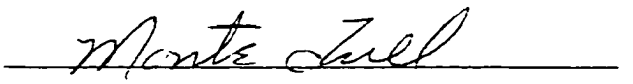
DESIGN OF OPTIMAL SUBBAND FILTER BANKS FOR IMAGE DISCRIMINATION

A Dissertation APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

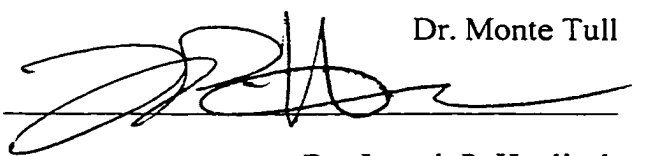
BY



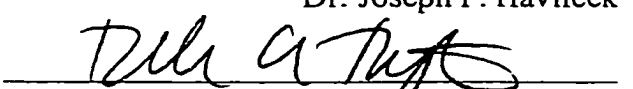
Dr. Victor E. DeBrunner (Chair)



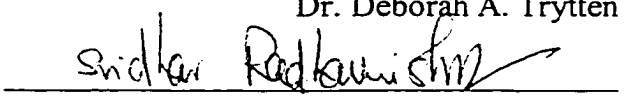
Dr. Monte Tull



Dr. Joseph P. Havlicek



Dr. Deborah A. Trytten



Dr. Sridhar Radhakrishnan

# ACKNOWLEDGEMENTS

This work would not have been possible without the help and support from my advisor Dr. Victor DeBrunner. I would like to thank him for his advice, guidance, support and assistance throughout this work. I would like to thank Dr. Monte Tull, Dr. Joseph Havlicek, Dr. Deborah Trytten and Dr. Sridhar Radhakrishnan, who served as my committee members. I also like to thank Dr. J.R.Cruz and Dr. William L. Kuriger for their help. I would also like to thank Dr. T. Prezbinda for his helpful comments and suggestions.

I would like to thank my father for his encouragement, support and the freedom he has given me in choosing the path of my life in every aspect and my mother for her patience and understanding. I would like to thank my parents-in-law for their encouragement and support throughout this work. I also like to thank my brother and sister for their encouragement and support.

I would like to thank the staff in ECE office Lynn Halverson, Marita Hardcastle and Cheryl Carney for their help. I would like to thank my sponsors McGee Eye Clinic and Lucent Technologies for their financial assistance.

I could not have completed this dissertation without the help, support, encouragement and understanding of my husband. I want to thank him for bearing me through my ups and downs and trying to make me see the silver lining of the clouds.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABSTRACT

The primary objective of this work is to improve texture classification system performance. The work is extended to improve the accuracy with which the faulty components of a printed circuit board are detected from a video sequence of infrared images generated by warming the board at power up. The direct motivation of this research is to enhance the FAULT DETECTION and IDENTIFICATION (FDI) system performance based on classification of the components in the circuit boards. The classification problem may be divided into the stages of feature extraction, dimensionality reduction and pattern recognition. Central to this work is that the signal representation plays a crucial role in the classification performance. Specifically, it is proposed that designing an optimal sub-band filterbank for fault detection and identification or texture classification improves the classification performance when the filterbank is used for that purpose.

The focus of this dissertation is on the design of subband filterbanks for feature extraction and classification of images. One of the major conclusions of the experiments is that the wavelet used for decomposing the images for classification plays a crucial role in the classification task. Furthermore, the commonly used octave band decomposition is evaluated against alternative decompositions. It is concluded that non-octave decompositions are generally superior. Also, the classification performance using various feature extraction techniques along with dimensionality reduction methods are compared. A quadrature mirror filterbank designed is tested in texture classification and fault detection, and results in superior classification performance compared to other filterbanks.

Optimal filters designed with image compression in mind do not guarantee optimality with respect to discrimination. Therefore, approaches for the design of optimal filterbanks with optimal discrimination are proposed. A simulated annealing algorithm is used to find the optimal filter coefficients by maximizing class separability. Algorithms are developed to find the optimal filterbank for a given dataset and to classify an unknown texture or to find if the given component is faulty or not.

Performance of the proposed methods is demonstrated in extensive experiments, which justify the new approaches.

# CHAPTER 1

# INTRODUCTION

## 1.1 OBJECTIVES

The primary objective of this work is to improve texture classification system performance. The design is applied to improve the accuracy with which faulty components of a printed circuit board are detected using infrared images. The direct motivation of this research is to enhance the FAULT DETECTION and IDENTIFICATION (FDI) system performance based on classification of the components in the circuit boards. Central to this work is the idea that the signal representation plays a crucial role in classification performance. Specifically, it is proposed that an optimal sub-band filterbank can be designed to improve fault detection and identification or texture classification performance.

In an effort to provide improved performance, this work seeks to generalize the Fault Detection problem. Correspondingly, some secondary objectives result:

- Demonstrate the efficacy of the optimal sub-band filter on signal classification, explain the factors that effect the performance, and specify the optimum configuration for texture classification.
- Establish a better understanding of the fault detection and identification system and how the various factors influence system performance with respect to the optimal filters in the time-frequency domain.

This work accomplishes these objectives using a complement of empirical and theoretical investigation.

## 1.2 PROBLEM DEFINITION

In industrial processes, the detection of defects in manufactured products or in the raw material can be crucially important. Manual visual inspection is often a tedious and laborious task; thus, automation is of great interest. As an example, consider an electronic industry manufacturing circuit boards. Operational costs for maintaining and repairing the circuit boards by testing each component in the circuit are expensive and time consuming. Automatic fault detection and identification is of interest in a wide variety of applications such as control systems, image analysis, analysis of radar signals, smart sensors, texture

analysis, medicine, industry *etc*. To solve this problem, one needs to find a means to reduce the time and expense involved by automation and also to improve the FDI system performance.

Most natural surfaces and naturally occurring patterns exhibit texture. A texture recognition system will therefore be a natural part of many computer vision systems. Some of the applications of the texture recognition systems are given here. In many cases of **industrial inspection**, the quality of a surface is well characterized by its texture. Hence, texture analysis plays an important role in inspection [1]. Images in **medicine** arise from non-intrusive techniques as x-ray, ultrasound, tomography etc. In several of these image types, textural properties are important diagnostics [2]. **Remote sensing** is the measurement of properties of a far distant object. Remote sensing techniques include satellite photography, seismic surveys, sonar surveys *etc*. Numerous approaches to texture recognition in remote sensing have been presented in the literature. Applications include terrain classification [3], cloud classification [4], and seismic pattern recognition [5].

## 1.3 INTRODUCTION

Pattern recognition is the study of theory and algorithms for automating the process of recognition through efficient representation of relevant information and its analysis using intelligent schemes [6]. The success of pattern recognition depends not only on the power of the data processing algorithm, but also on the proper representation of the input data so that all the salient aspects of the data for the specific task at hand are captured and utilized while all the irrelevant information is discarded. A general schematic of a classification process is shown in Figure 1.1. The first step is to find an effective and appropriate representation of the signal or image, which is based on a given criterion, representing only the most relevant information in a compact form. The fact that classification systems with small numbers of parameters have better generalization, and so are computationally cost effective, and also can be trained and adapted faster are motivations for efficient feature extraction techniques [6].

Feature extraction [7] can also be thought of as the transformation that replaces the measurements from various sources with features. One needs to combine information provided by various sources (or features) to obtain more reliable performance. In this dissertation, general methods of pattern classification through optimal sub-band filter design are investigated.


Figure 1.1. Structure of a Classifier

Recent developments have shown that wavelets are optimal or near optimal for solving a wide range of problems in compression, estimation and detection [8]. While previous contributions have shown that wavelets are a significant tool in solving a number of problems in DSP, they have not addressed the problem of how to choose the wavelet basis for a given application. In contrast to classical transforms such as Fourier, cosine, Walsh, Hartley, Hadamard, *etc.*, the wavelet transform is actually a class (or family) of transforms that are parameterized and can hence be tuned to a specific application. This freedom makes wavelets powerful, but with a limited theoretical understanding for why wavelets work; it also means that choosing the best wavelet transform or basis for a given application is difficult. Given that wavelets can be parameterized, one might ask: "What properties should the transform have?" While practical problems typically have a well-defined knowledge of optimality, translating this into choosing an optimal wavelet basis is typically non-trivial.

One method for choosing the best wavelet transform for a given application would be to choose a time varying and signal dependent wavelet basis. Clearly, this would solve the problem, if for a given application a signal dependent optimization problem could be posed and solved in real time. However, in most applications, it is not obvious what the desired cost function should be. The more interesting and desirable solution (although sub-optimal to signal dependent design) is to find a robust solution for a given problem. Conceptually the wavelet design can be obtained in two distinctly different ways:

1. Find the optimal wavelet basis for a large class of signals that are representative (statistically) for the problem to be solved.
2. Design the wavelet basis such that it is near-optimal for the given problem based on implicit properties of the basis rather than being explicitly signal dependent.

Excluding the consideration of special classes of functions and signals, neither of these two problems has been analyzed in any detail. However, problem (1) can be solved theoretically if a sufficiently rich

class (statistically) of test signals can be generated or obtained, an appropriate cost function defined, and one has the required computational resources and time.

The wavelet designs presented in this dissertation focus on situations where problem (1) can be successfully solved in that it considers the design of a wavelet basis with a proper cost function. In the following section, a short overview of the work accomplished in the dissertation is given.

## 1.4 DISSERTATION OVERVIEW

As stated above, this dissertation addresses particular problems related to the design of sub-band filters with a given set of goals. Each problem considered in this work addresses issues in wavelet theory dealing with alternative views on classical wavelet or sub-band filter design methods. Broadly speaking, the types of questions addressed here can be categorized into three different classes:

- **Influence of wavelet:** For a given classification task or fault detection and identification system, what is the importance or role of the wavelet? Can one in practice design an optimal wavelet that improves the performance of that system?
- **Design of an Optimal wavelet:** How to design the optimal wavelet based on a more appropriate cost function for a given task or application?
- **Configuration:** What are the other factors that can influence the system performance in the time-frequency domain in the context of optimal wavelet basis? Can we specify the optimum configuration for signal classification or fault detection?

## 1.5 DISSERTATION OUTLINE

The dissertation is divided into four categories (sections): *the introduction, literature review and necessary background,* and *methodology and applications*. The methodology and application parts constitute the core part of this work. The chapters are outlined according to their categories in the categorical order. That means, chapters introducing the problem are outlined first, followed by chapters that have literature review and the necessary background. Chapters and appendices with methodology are outlined and the chapters with applications follow methodology section. Finally, a chapter with conclusions and future work is outlined.

### 1.5.1 INTRODUCTION

**Chapter 1** introduces the problem of texture classification and fault detection. The objectives accomplished in the research are listed. A brief background of the general topic is presented. Also, a brief overview of the applications and approaches is presented.

### 1.5.2 BACKGROUND

**Chapter 2** presents a detailed discussion of signal representation for pattern classification. The problem is decomposed to feature extraction, dimensionality reduction and classification. It is emphasized that although an appropriate classifier is necessary, it is the signal representation that profoundly affects the classification performance of a given problem. The importance of feature extraction and dimensionality reduction is introduced. Feature selection and feature projection methodologies for dimensionality reduction are presented.

**Chapter 3** provides a mathematical background for wavelet and wavelet packet transforms. The necessity for choosing the optimal local basis for a given task (signal compression and signal classification) is presented and previous algorithms developed based on these lines are reviewed. In all these algorithms, the influence of the QMF filter used for the optimal local basis selection for the given task is ignored. This dissertation focuses on this issue.

### 1.5.3 METHODOLOGY

**Chapter 4** presents the recently developed interesting algorithms that propose criterion dependent optimal local basis for texture classification. In this chapter, various factors involved in a classification system, namely feature extraction, dimensionality reduction methods, types of classifiers, and the wavelet tree decomposition methods are analyzed. The influences of these parameters on the classification performance are studied thoroughly. A computationally simple algorithm is developed for classification and the performance is compared to that of the successful existing classification methods. This chapter sets the background for the next analysis in which we want to know the impact of the QMF filters. Some of the new studies and results in Chapter 4 are:

- Study the influence of various factors involved in classification. These factors include feature extracting measures (e.g., $L_1$-norm, F-norm, $\Delta$-norm etc.), dimensionality reduction techniques (feature selection, feature projection) and the types of classifiers (Euclidean distance, simplified Mahalanobis distance, and neural network) on classification system performance.
- Develop a computationally simple algorithm for classification.
- Compare the classification performance for various tree structures (WT, WPT etc.) with respect to feature extracting measures, dimensionality reduction techniques and types of classifiers.

**Chapter 5** studies the influence of using various QMF filters on texture classification performance. This study is conducted on several types of wavelet tree structures, i.e., the wavelet transform (octave tree), the uniform tree and the wavelet packet transform based on some criteria (e.g., energy, separability). Along with the influence of QMF filters on classification performance, other factors involved in the classification task as described in Chapter 4 are also studied. This chapter essentially emphasizes the effect of QMF filters on texture classification and proves that the influence is significant with respect to the classification and sets the background for the necessity to design optimal sub-band filters for that purpose. Some of the new studies and results in Chapter 5 are:

- Study the influence of sub-band filters on texture classification performance.
- The classification performance for various tree structures (WT, WPT etc.) with respect to various sub-band filter candidates are compared and WPT performs better than WT.

**Chapter 6** considers the problem of designing wavelets based on class separability. The relationship between classification rate and class separability is established. Simulated annealing (SA) is used to find the optimal basis by maximizing the class separability and satisfying the QMF constraints, as there are many minima and maxima in the class separability with respect to the sub-band filters. An algorithm is developed to find an optimal wavelet basis given the data samples for a classification task. Some of the new studies and results in this chapter are:

- Establish an empirical relationship between classification performance and class separability.
- Design optimal sub-band filters based on class separability for classification using SA.
- Study the various parameters (e.g., step size, initial temperature, temperature reduction etc.) involved in the SA for convergence.

- Develop an algorithm using the above mentioned design technique from the given data sets for classification.

**Chapter 7** clearly elaborates the whole process of designing the optimal filter for a given data set for classification from scratch. It details the steps involved in training and classification for the final result.

**Appendix A** provides a comparison of the classification performance of algorithms used in Chapter 4. These algorithms are energy-based tree decomposition and separability-based tree decomposition. The results reproduced in implementing these algorithms are compared with the results provided in previous work.

**Appendix B** presents the mathematical details involved in expressing the cost function, class separability in terms of sub-band filter coefficients, and the complexity involved in doing so to calculate the gradient of the class separability with respect to the low pass filter coefficients. These mathematical details are introduced in Chapter 6 and are included in this appendix.

### 1.5.4 APPLICATIONS

**Appendix C** provides an important application of signal classification. The fault detection and identification system, which is a special task requiring signal classification, is presented in this appendix. The approach of designing the optimal sub-band filter based on class separability is similar, except that the data involved is different. The performance is measured differently from the texture classification problem, and reflects either the number of times the faulty component is detected correctly or the separation between the faulty components in a bad board from the functioning ones. Some of the new results in this appendix are:

- Influence of the optimal sub-band filter on FDI system performance.
- Influence of feature extraction, dimensionality reduction, and various wavelet tree structures on FDI system performance.

**Appendix D** has more figures of the second dataset (Fault identification analysis is performed on two sets of circuit boards. Analysis and results on the first dataset is presented in detail in Appendix C and, for

the second set, briefly in Appendix D) plotted using the results obtained in the analysis performed in Appendix C.

**Chapter 8** provides the conclusions, listing the major contributions of the work and suggesting directions for future research. It is concluded that an optimal filterbank should be designed based on class discrimination rather than energy compaction when the filter is used for signal classification.

# CHAPTER 2

# SIGNAL REPRESENTATION FOR CLASSIFICATION

## 2.1. INTRODUCTION

Signal classification, signal compression, and noise removal are examples of classic signal analysis problems. Each has been widely studied and each has a wide variety of applications. For any aspect of signal analysis, the means by which the signal is represented is of vital importance. In this chapter, I investigate methods to extract features that are relevant and discard the information that is irrelevant in the context of signal classification. Often, important features for classification are characterized by local information in the dual domains of time and frequency.

The chapter is organized as follows. Section 2.2 provides a formulation of the problem of feature extraction and pattern recognition. Section 2.3 introduces pattern classification, and describes the advantages and drawbacks of established techniques. In section 2.4, the importance of feature extraction to the success of pattern classification is discussed. Dimensionality reduction is often a necessary complement to feature extraction, and this is discussed in section 2.5. Finally, section 2.6 summarizes the chapter.

## 2.2. PROBLEM FORMULATION

A pattern may be said to consist of N variables $x = [x_1, x_2, ..., x_N]^T$. This is the measurement vector, which may be the elements of a sampled signal. Each pattern $x$ may be said to belong to one of L classes, denoted by $y_l$. We may then say that $x \in X \subseteq \Re^N$ is the input signal space and $y \in Y = \{y_1, ..., y_L\}$ is the output response space, which is simply a collection of L class labels. Signal classification may be regarded as a function $d: x \rightarrow y$, which assigns a class label to each input signal $x \in$ X.

Direct application of the data in signal space is usually prohibitive due to high dimensionality of this space. Indeed, the signal space is highly redundant with respect to the response space. This implies the need

to reduce the dimensionality of the problem. One must extract the features needed to discriminate the signals and discard everything else. This can greatly improve the performance of the chosen classifier and reduce complexity. Feature space is the space in between the signal space and the response space. A feature extractor is defined as a map from signal space to feature space, and the classifier is a map from feature space to response space. The classification process consists of feature extraction followed by classification.

## 2.3. PATTERN CLASSIFICATION

The task of classifying the data is central to many applications. The act of classification is tightly bound to the proper extraction of relevant features from the unprocessed data. The basic blocks in a classification problem are depicted in Figure 2.1.

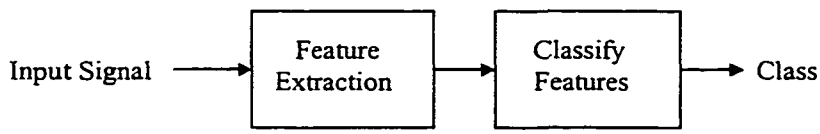



Figure 2.1. The classification problem

In this section, a brief overview of pattern classifiers is given. It is shown that feature extraction is fundamental to classifier performance; even the most adept classifier must have an appropriate and efficient representation of the input signal.

The practical methodologies that exist for pattern classification may be loosely grouped into three categories. Historically, the two classical methods are the statistical (or decision-theoretic) approach [7] and the structural (or syntactic) approach [9]. The third, and most recently established type of pattern classifier is the learning (or neural) approach. Learning algorithms have their origins in perceptrons and adaptive linear elements [9], and have matured into the diverse field of neural networks [11].

**Statistical pattern recognition** is based upon the statistical analysis of the data to be classified. The data are assigned to a particular class by compiling a probabilistic model (estimating probability density functions) of the data in N-dimensional space, and dividing the space into regions corresponding to each class according to some criterion. The major accomplishments in statistical pattern recognition include

Bayesian classifiers, distance classifiers, and classification used in regression trees. These are examined in this chapter.

**The syntactic approach** on the other hand is based on utilizing the structure of patterns and the interrelationships between the components of a pattern. Syntactic pattern recognition involves identifying meaningful components or "primitives" of the patterns, and developing a formal syntax or "grammar" describing the synthesis of the patterns from their primitives. The preference here is to discuss structural methods in the context of feature extraction. From the perspective of this work, the development of primitives and syntax is more a signal representation issue than a classification task. Section 2.4 demonstrates the importance of structural representations in feature extraction.

**Learning algorithms** almost invariably take the form of artificial neural networks. Artificial neural network approaches may also be termed deterministic as opposed to statistical because the learning algorithms assume nothing about the statistical properties of the pattern classes. It is shown, however, that statistical and neural network pattern classifiers are very similar in form and objective [11].

The intent of this section is to illustrate the major features of the most popular pattern classifiers in use today, and the differences between them that are important. Correspondingly, two representative classifiers are chosen to carry through the thesis based upon the ease with which they may be interpreted and their applicability to the texture classification problem.

### 2.3.1. BAYESIAN PATTERN CLASSIFICATION

The central problem in statistical pattern recognition is the development of decision functions from sets of finite sample patterns of different classes so that the functions partition the input space into regions, each of which contains the sample patterns belonging to each class. In general, the most information that can be known about the input space are the *a posteriori* probabilities $P(y_l|x)$ for $l = 1,\ldots, L$. This is the probability that pattern x comes from class $y_l$.

In this framework, pattern classification is posed as a statistical decision problem. One evaluates the $L$ *a posteriori* probabilities and selects the largest. In general, the *a posteriori* probabilities $P(y_l|x)$ are not

known, but may be calculated from the *a priori* probabilities $P(y_l)$ and the conditional density functions $P(x|y_l)$ using the Bayes' theorem, which is [12].

$$P(x,y_l) = P(y_l)p(x|y_l) = p(x)P(y_l|x) \tag{2.1}$$

Rearranging, we get

$$P(y_l|x) = \frac{P(y_l)p(x|y_l)}{p(x)} \tag{2.2}$$

where

$$p(x) = \sum_{j=1}^{L} P(y_j)p(x|y_j) \tag{2.3}$$

Note that $p(x)$ is the probability density function of the input space that remains constant for all $P(y_l|x)$, so it can be ignored for purposes of discrimination. When the true class distributions are not known, the *a priori* probabilities are often made equal: $P(Y_l) = 1/L$ for $l = 1,\ldots,L$.

To summarize, Bayes' decision rule is really nothing more than the implementation of the decision functions:

$$d_l(x) = p(x|y_l)P(y_l),\ l = 1,\ldots L \tag{2.4}$$

where a pattern x is assigned to class $y_i$ if for that pattern $d_i(x)>d_j(x)$ for all $j\neq i$. This Bayes' decision rule has the property that the probability of classification error is minimized, making Bayes' classifier statistically superior to any other. The Bayes classifier is illustrated in Figure 2.2.

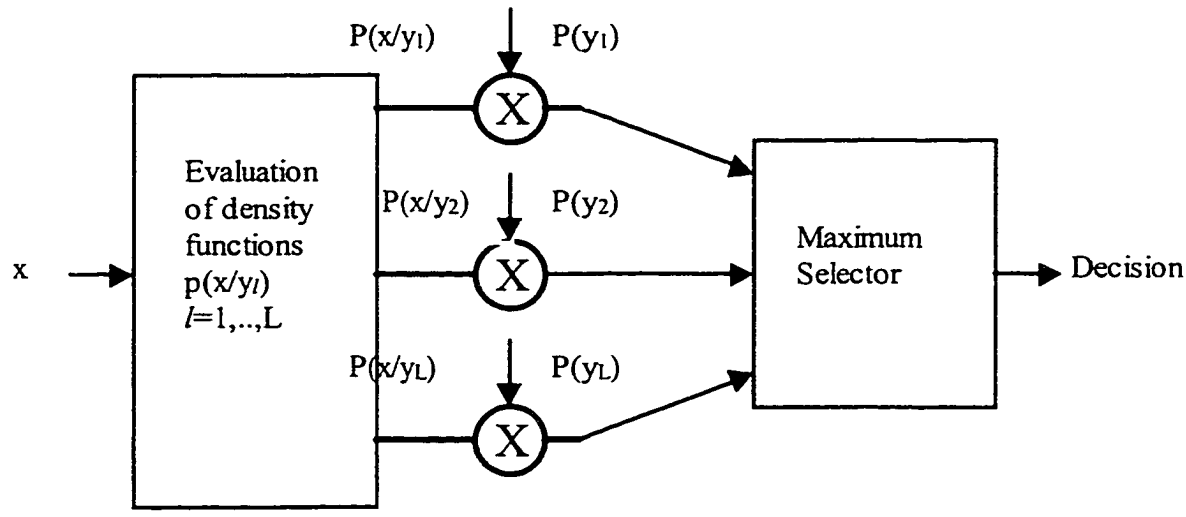



Figure 2. 2. A Bayesian Classifier

The challenge here lies in estimating the densities $p(x|y_l)$ from the training data. This is difficult, if not impossible, when the dimension of the input space N is large. The high dimensionality imposes the constraint that the number of training set examples must be much greater than N to get a reliable estimate of $p(x|y_l)$.

2.3.1.1. The Gaussian Bayes Classifier [7]

If it is reasonable to assume a parametric form of the conditional probability density function $p(x|y_l)$, then the Bayes classifier derived in the preceding section can take a more tractable form. A common assumption is that the densities $p(x|y_l)$ are multivariate normal (Gaussian). Although for some datasets it is not well suited to make this assumption, the normal distribution does represent an appropriate model for many practical applications.

Consider L classes of patterns, governed by the multivariate normal density functions:

$$p(x|y_l) = \frac{1}{(2\pi)^{N/2}|C_l|^{1/2}} \exp\left[-\frac{1}{2}(x-m_l)^T C_l^{-1}(x-m_l)\right], \; l = 1,\ldots,L \tag{2.5}$$

where each density is completely specified by its mean vector $m_l$ and its covariance matrix $C_l$, which are defined as

$$m_l = E_l[x] \tag{2.6}$$

and

$$C_l = E\left[(x - m_l)(x - m_l)^T\right] \tag{2.7}$$

where $E_l[.]$ denotes the expectation operator over the patterns of class $y_l$. Here, $|C_l|$ indicates the determinant of matrix $C_l$. Sample patterns taken from a normal distribution tend to fall in a single cluster with its center determined by the mean vector and its shape defined by the covariance matrix. The loci of points of constant density are hyper-ellipsoids with the principal axes in the directions of the eigenvectors of the covariance matrix and the lengths of these axes determined by the eigenvalues.

According to equation (2.4), the decision function for class $y_l$ may be chosen as $d_l(x) = p(x|y_l)P(y_l)$. In other words, we may use the form

$$d_k(x) = \ln[p(x|y_l).P(y_l)] = \ln p(x|y_l) + \ln P(y_l) \tag{2.8}$$

because ln is monotonic.

Substituting equation (2.5) into equation (2.8) yields

$$d_k(x) = \ln P(y_k) - \frac{N}{2}\ln 2\pi - \frac{1}{2}\ln|C_k| - \frac{1}{2}\left[(x - m_k)^T C_k^{-1}(x - m_k)\right], \; k = 1,\ldots, \mathrm{K} \tag{2.9}$$

Since the term $\frac{N}{2}\ln 2\pi$ does not depend on k, it can be eliminated, giving

$$d_k(x) = \ln P(y_k) - \frac{1}{2}\ln|C_k| - \frac{1}{2}\left[(x - m_k)^T C_k^{-1}(x - m_k)\right], \; k = 1,\ldots, \mathrm{K} \tag{2.10}$$

which is the Bayesian decision function for normally distributed patterns. These decision functions are hyper-quadratic, meaning that this Bayesian classifier can only place a quadratic discriminant function between pattern classes. If the pattern classes are truly characterized by normal densities, however, no other surfaces yield better results on an average basis. The quadratic decision functions are

$$d_k(x) = \ln P(y_k) - \frac{1}{2}\ln|C_k| - \frac{1}{2}x^T C_k^{-1}x + x^T C_k^{-1}m_k - \frac{1}{2}m_k^T C_k^{-1}m_k \;, \; k = 1,\ldots,\mathrm{K} \tag{2.11}$$

If it is assumed that all covariance matrices are equal, $C_k = C$ for $k = 1,\ldots,\mathrm{K}$, it follows that the decision functions become

$$d_k(x) = \ln P(y_k) + x^T C^{-1}m_k - \frac{1}{2}m_k^T C^{-1}m_k \;, \; k = 1,\ldots,\mathrm{K} \tag{2.12}$$

which represents a set of linear discriminant functions. In this case, the decision surface is linear with respect to the input space, describing a hyper-plane. The normal Bayesian classifier is therefore often called linear discriminant analysis (LDA). The advantages of LDA are that it is interpreted and implemented easily, that it trains quickly with reasonably-sized datasets, and that no adjustment of its architecture or training algorithm is required. Some constraints relevant to the applicability of LDA are that the assumptions of normal pattern densities must be reasonable; the data must be reasonably well clustered and linearly separable. It is sensitive to outliers and noise. The assumption of identical covariance structures for all classes is also unrealistic in many cases.

### 2.3.2. ARTIFICIAL NEURAL NETWORKS

An artificial neural network (ANN) is a computational system inspired by the learning characteristics and the structure of biological neural networks. The applications of ANNs as pattern classifiers are described in this section. Figure 2.3 shows the hierarchy of some artificial networks that have been used as pattern classifiers. The discussions of ANNs are limited to those trained using supervised learning.

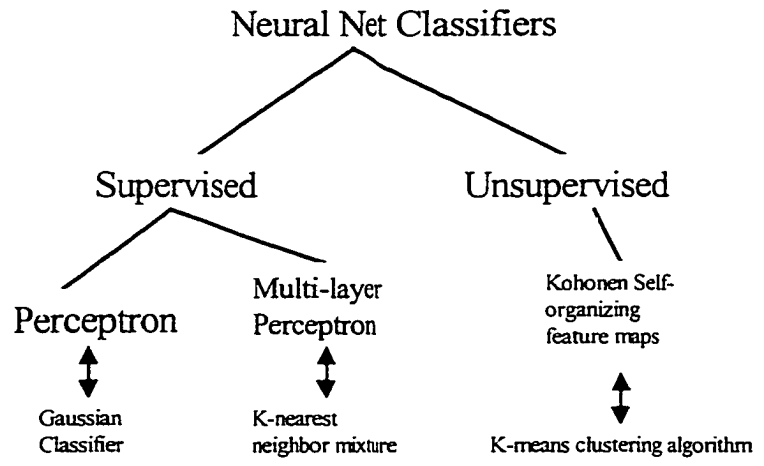



Figure 2.3. Taxonomy of neural nets that can be used as classifiers. Classical algorithms, which are most similar to the neural net models, are listed along the bottom.

Such ANN's are presented with a training set of $p$ example pairs from the input space and the response space:

$$\left\{(x^{(1)},y^{(1)}),\ldots,(x^{(p)},y^{(p)})\right\}, \tag{2.13}$$

where $p$ is finite. If class membership information is available during training, supervised methods, in general, fare better than unsupervised methods. This is due to the fact that knowledge of class membership aids the construction of appropriate discriminant boundaries.

The strength of neural network based pattern classifiers lies in its applicability to problems involving arbitrary distributions of data. Moreover, a firm understanding of the pattern recognition properties of neural networks has emerged, relating their characteristics to Bayesian decision making.

Of all the ANN architectures that have been used as pattern classifiers, the most commonly used is the multilayer perceptron (MLP). In turn, the learning algorithm that is almost always used to train MLP is the backpropagation algorithm, which is a stochastic approximation of the steepest descent algorithm. The MLP architecture and the backpropagation algorithm are the simplest and most extensively studied of all neural network paradigms. A MLP containing nonlinear activation functions is capable of constructing arbitrarily complex decision boundaries in feature space for networks of two layers or more. Some problems associated with the MLP and backpropagation are that training may be slow and that selection of the best network size may be difficult [11].

For the purpose of assessing the discriminatory nature of textures using a variety of signal representations, a MLP network trained using a standard backpropagation algorithm suffices [10]. The following sections introduce MLP, the means by which it is trained, and its capabilities.

#### 2.3.2.1. The Perceptron

The perceptron [13] is a feedforward network with one output neuron that learns a separating hyper-plane in a pattern space. As depicted in Figure 2.4, the perceptron forms a weighted sum of the n components of the input vector $x=[x_1,x_2,\ldots,x_N]^T$ and adds a bias value, $\theta$. The result is then passed through a nonlinearity $f(\bullet)$.

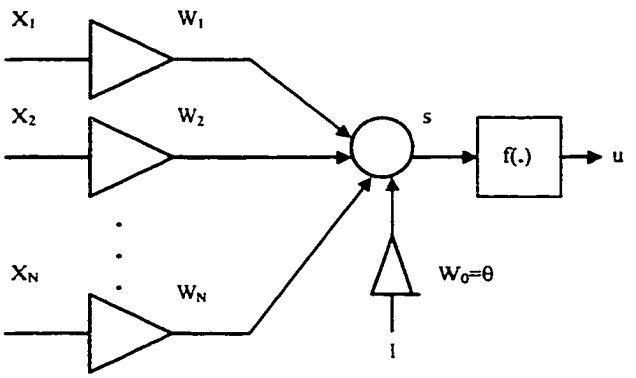



Figure 2.4. The Perceptron

Rosenblatt's original model used a hard-limiting nonlinearity:

$$f(s) = \begin{cases} 1 & s > 0 \\ 0 & s \leq 0 \end{cases} \tag{2.14}$$

which is illustrated in Figure 2.5. When perceptrons are combined together in layers, it is more common to use the logistic sigmoidal nonlinearity:

$$f_{log}(s) = \frac{1}{1 + e^{-\beta s}} \tag{2.15}$$

This function is continuous and varies monotonically from 0 to 1 as s varies from $-\infty$ to $\infty$. The gain of the sigmoid, $\beta$, determines the steepness of the transition region; this is often set to 1. The main advantage of the sigmoid nonlinearity is that it is differentiable. This property has had an historical impact because it made it possible to derive a gradient search algorithm for networks with multiple layers.

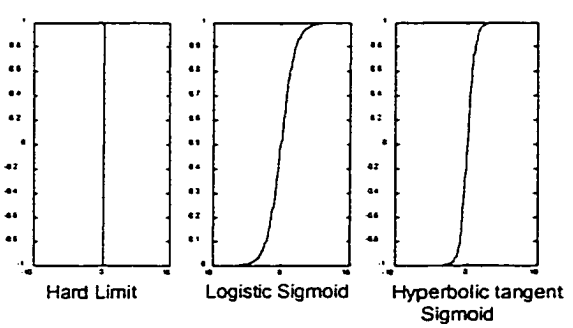



Figure 2.5. Three common types of nonlinearity used as the activation function in an artificial neuron

Another function belonging to the sigmoid family is the hyperbolic tangent sigmoid:

$$f_{\tanh}(s) = \frac{e^{\beta s} - e^{-\beta s}}{e^{\beta s} + e^{-\beta s}} \tag{2.16}$$

the outputs of which range from -1 to 1. In many cases, networks which use the hyperbolic tangent sigmoid as a nonlinearity tend to learn faster than those which use the logistic sigmoid [14].

2.3.2.2. The Multilayer Perceptron

The capabilities of single perceptrons are limited to linear decision boundaries, however, and are suitable only for problems requiring a simple linear division of the pattern space. Many problems require a nonlinear partitioning of the pattern space. This can be achieved using a multilayer perceptron network, which cascades two or more layers of perceptrons together, making it possible to partition the pattern space with arbitrarily complex decision boundaries. The individual perceptrons in the network are called neurons or nodes, and usually employ a sigmoid nonlinearity instead of a hard limiter. A typical MLP network architecture is depicted in Figure 2.6.

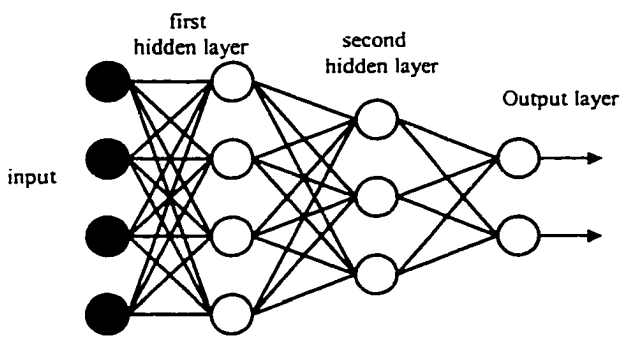



Figure 2.6. The architecture of a typical MLP network

The input vector feeds in to the first layer nodes; the outputs of this layer feed into each of the second layer nodes, and so on. Often, the nodes are fully connected between layers. The multiple nodes in the output layer correspond to multiple classes in a pattern recognition problem.

For classification problems, Lippmann [13] demonstrated that a 2-layer MLP can implement arbitrary convex decision boundaries given a sufficient number of hidden layer nodes. Essentially, each hidden layer

node provides a linear boundary in pattern space, and each of the boundaries may be nonlinearly connected in a smooth fashion with the others by the sigmoid nonlinearity.

Many algorithms have been developed which adapt the network weights so as to provide a suitable map between the set of input vectors and the set of desired responses. In general, an algorithm is either supervised, in which case the desired response is available during the learning phase, or unsupervised, in which case clusters are formed from the input patterns. A training dataset of textures includes knowledge of the actual class of movement, and therefore, our interest here is limited to supervised learning. The backpropagation algorithm is used to train the MLP in this work; it is briefly explained in the next section.

#### 2.3.2.3. The Back-Propagation Training Algorithm

Back Propagation is a generalization of the LMS algorithm [13]. It uses a gradient search technique to minimize a cost function equal to a mean square difference between the desired and the actual net outputs. The net is trained by initially selecting small random weights and internal thresholds and then presenting all training data repeatedly. Weights are adjusted after every trial using information specifying the correct class until the weights converge and the cost function is reduced to an acceptable value. An essential component of the algorithm is the iterative method described in Figure 2.7 that propagates error terms required to adapt weights back from nodes in the output layer to nodes in lower layers.

#### 2.3.2.4. Issues in MLP training

Learning Rate: The learning rates can be uniform throughout the network, or different for each layer or node. In general, it is difficult to determine the best learning rate, but a useful rule of thumb is to make the learning rate for each node inversely proportional to the average magnitude of the vector feeding the node. Many schemes that adapt the learning rate as a function of the local curvature of the error surface have been proposed [14]. The simplest approach is to add a momentum term of the form $\alpha(w(k) - w(k-1))$ to each weight update, where $0 < \alpha < 1$. This term makes the current search direction an exponentially weighted average of past directions, and helps keep the weights moving across flat portions of the error surface after they have descended from steep portions.

Stopping Criteria: The iterative process of computing the gradient and adjusting the weights is continued until a minimum is found in the error surface or a point determined to be sufficiently close. Several measures are candidates for stopping criteria. If the magnitude of the gradient falls below a chosen level, the algorithm may be terminated, as this may indicate that the minimum is being approached. Perhaps a more common stopping criterion is a lower threshold on the sum square error, J(w). This requires knowledge of the minimum value of J(w), which is not always available. One might consider stopping when a chosen number of iterations have been performed. In this situation, the number of iterations must be determined by empirical evidence gathered from previous training sessions. There is no guarantee that the best network performance with respect to the network's sum squared error reflects the set of weights yielding the best classification performance. Indeed, this is a limitation of the sum squared error cost function used by the back-propagation algorithm.

Hidden layer Nodes: The optimum number of hidden layers is difficult to establish, and is strongly dependent on the nature of the data. The optimum size is that which would enable the network to capture only the underlying structure of the data. The upper bound on the number of hidden layer nodes should be less than the number of training samples, or the network simply memorizes the training samples, resulting in poor generalization. In addition to network architecture, generalization is affected by the number of patterns and the complexity of the problem at hand.

### 2.3.3. OTHER CLASSIFIERS

Multi-layer perceptron neural networks and linear discriminant analysis are simple, yet effective pattern classifiers, and for this reason, they are widely used [10]. Many alternative approaches to pattern recognition exist however. Ultimately, the best classifier depends on the nature of the data to be classified. The next section provides a brief review of other important classifiers, which offer slightly different approaches to the classification problem.

A sigmoidal logistic nonlinearity is used where the function f(α) is

$$f(\alpha) = \frac{1}{1 + e^{-(\alpha - \theta)}}$$

Step 1. Initialize weights and offsets

Set al weights and node offsets to small random values.

Step 2. Present input and desired outputs

Present a continuous valued input vector $x_0, x_1, \ldots, x_{N-1}$ and specify the desired outputs $d_0, d_1, \ldots, d_{M-1}$. If the net is used as a classifier then all desired outputs are typically set to zero except for that corresponding to the class the input is from. That desired output is 1. The input could be new on each trial or samples from a training set could be presented cyclically until weights stabilize.

Step 3. Calculate actual outputs

Use the sigmoid nonlinearity from above and the formulas

- output layer output $$Y_l = f\left(\sum_{k=0}^{N_2-1} W_{k,l}^N x_k^N - \theta_l^N\right) \quad 0 \le l \le M-1$$
- second hidden layer output $$x_k^N = f\left(\sum_{j=0}^{N_1-1} W_{j,k}^{'} x_j^{'} - \theta_k^{'}\right) \quad 0 \le k \le N_2 - 1$$
- first hidden layer output $$x_j^{'} = f\left(\sum_{j=0}^{N-1} W_{ij} x_i - \theta_i\right) \quad 0 \le j \le N_1 - 1$$

to calculate the outputs $y_0, y_1, \ldots, y_{M-1}$.

Step 4. Adapt weights

Use a recursive algorithm starting at the output nodes and working back to the first hidden layer. Adjust weights by

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_j^{'}$$

where $w_{ij}(t)$ is the weight from the hidden node i or from an input node j at time t, $x_j^{'}$ is either the output of the node i or is an input, is the gain term, and $\delta_i$ is an error term for node j. If node j is an output node, then

$$\delta_j = y_j(1 - y_j)(d_j - y_j),$$

where $d_i$ is the desired output of node j and $y_i$ is the actual output.

If node j is an internal hidden node, then

$$\delta_j = x_j^{'}(1 - x_j^{'}) \sum_k \delta_k w_{jk},$$

where k is over all nodes in the layers above node j. Internal node thresholds are adapted in a similar manner by assuming they are connection weights on links from auxiliary constant-valued inputs. Convergence is sometimes faster if a momentum is added and weight changes are smoothed by

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_j^{'} + \alpha(W_{ij}(t) - W_{ij}(t-1)), \qquad \text{where } 0<\alpha<1.$$

Step 5. Repeat by going to step 2.

Figure 2.7. The Back-Propagation Training Algorithm

2.3.3.1. Distance classifiers

The motivation for using distance functions as a classification tool follows naturally from the notion that the similarity of pattern vectors may be measured by their proximity. Pattern classification by distance functions can be expected to yield satisfactory results only when the classes tend to be well clustered. Since the proximity of an unknown pattern to the patterns of a known class serves as a measure for its classification, these approaches are termed minimum-distance classifiers. The Euclidean distance between a given pattern vector x and the $i^{th}$ prototype vector is

$$D_i = \|x - y_i\| = \sqrt{(x - y_i)^T (x - y_i)} \tag{2.17}$$

where $y_1, y_2, ..., y_K$ are the prototypes of the K pattern classes. A minimum-distance classifier computes the distance from an unknown pattern x to the prototype of each class, which is the mean vector of the pattern vectors within each class, and assigns the pattern to the closest class.

The decision boundaries for minimum-distance classifiers are the perpendicular bisectors of the lines joining the prototypes of different classes. Therefore, minimum distance classifiers are a special case of linear classifiers, in which the decision boundaries are constrained to have this property.

Consider a set of sample patterns of known class membership $\{s_1, s_2, ..., s_p\}$, where it is assumed that each pattern belongs to one of the classes $y_1, y_2, ..., y_K$. The nearest-neighbor (NN) classification rule assigns a pattern x to the class of its nearest neighbor, where $s_i \in s_1, s_2, ..., s_p$ is a nearest neighbor to x if

$$D(s_i, x) = \min_p \{D(s_p, x)\}, \qquad p = 1, ,... P \tag{2.18}$$

where D is any distance measure defined on the pattern space.

This is called the 1-NN rule since it employs only the class membership of the nearest neighbor to x. A k-NN rule consists of determining the k nearest neighbors, and classifying x according to the most prevalent class in this group.

One of the drawbacks of the k-NN methods is that, in order to provide a sufficiently rich set of exemplars, it is necessary to store a large set of sample patterns of known classification. In addition, the distances from each pattern to be classified to all the stored samples must be computed for classification. This represents a severe computational burden for large datasets.

## 2.4. FEATURE EXTRACTION

### 2.4.1. THE IMPORTANCE OF FEATURE EXTRACTION

The preceding section explained the various techniques available for pattern classification. Before a pattern classifier can be properly designed or effectively used, it is necessary to consider the feature extraction and data reduction problems. Although feature extraction should be considered before a classifier is designed, a greater appreciation of the importance of feature extraction is gained when the order of presentation of the two topics has been reversed.

The goal is to eliminate a significant number of dimensions of the multivariate data to obtain efficient representation of the underlying structure. In the context of pattern classification, feature extraction consists of choosing those features that are most effective for preserving class separability. Feature extraction methods can be divided into two groups: statistical and structural. Statistical feature extraction methods lend themselves to direct mathematical description and machine implementation. Among the significant contributions to statistical feature extraction are the orthogonal transform methods (the FFT, WT, singular value decomposition, etc.)

### 2.4.2. FEATURE EXTRACTION FOR CLASSIFICATION

When we have two or more classes of data, the goal of feature extraction is to choose those features that are most effective for preserving class separability. This section explores the issues in selecting a feature set that accurately represents the data of interest. A feature set may be considered optimum in some sense if it maximizes or minimizes a chosen class labeling, class separability criterion.

Class separability criteria are essentially independent of the coordinate systems [7]. Furthermore, class separability depends not only on the class distributions but also on the classifier used. For example, the optimum feature set for a lineal classifier may not be the optimum set for the other classifiers for the same distributions. In order to avoid this additional complexity, let us assume that we seek the optimum feature set with reference to the Bayes classifier, which results in the minimum error for the given distributions. Then, class separability becomes equivalent to the probability of error due to the Bayes classifier, which is the best one can expect provided exact knowledge of the *a posteriori* probabilities is available..

Therefore, theoretically speaking, the Bayes error is the optimum measure for feature effectiveness. A major disadvantage of the Bayes error as a criterion is the fact that an explicit mathematical expression is not available except for a very few special cases.

The criteria to evaluate the effectiveness of features must be a measure of the overlap or class separability among the distributions, and not a measure of fit such as the mean-square error. The Bayes error is the best criterion to evaluate feature sets, and *a posteriori* probability functions are the ideal features. Unfortunately, the Bayes error is too complex mathematically, and therefore we need a simpler criterion associated with the systematic feature extraction algorithms. One such criterion is based on scatter matrices. It is simple in expression and gives systematic feature extraction algorithms.

There are two criteria types that are frequently used in practice. One type is based on a family of scatter matrices that are conceptually simple and give systematic feature extraction algorithms. The criteria used measure the class separability of the L classes but do not relate directly to the Bayes error. The other type is a family of criteria that give upper bounds to the Bayes error. The Bhattacharyya distance is one of these criteria. However, these criteria are developed only for two-class problems, and they are based on the normality assumption [7].

2.4.2.1. Scatter Matrices and Separability Criteria

In discriminant analysis of statistics, within-class, between-class and mixture scatter matrices are used to formulate criteria for class separability [7]. A within-class scatter matrix shows the scatter of samples around their respective class expected vectors, and is expressed as

$$S_w = \sum_{i=1}^{L} P_i E\left\{(X - M_i)(X - M_i)^T | \omega_i\right\} = \sum_{i=1}^{L} P_i \Sigma_i \tag{2.19}$$

On the other hand, a between-class scatter matrix is the scatter of the expected vectors around the mixture mean as

$$S_b = \sum_{i=1}^{L} P_i (M_i - M_0)(M_i - M_0)^T \tag{2.20}$$

where $M_0$ represents the expected vector of the mixture distribution and is given by

$$M_0 = E\{X\} = \sum_{i=1}^{L} P_i M_i \tag{2.21}$$

In order to formulate class separability, these matrices need to be converted to a number. The number should be larger when the between-class scatter is larger or when the within-class scatter is smaller. There are several ways to do this, and typical criteria are the following:

$$J_1 = tr(S_w^{-1} S_b) \tag{2.22}$$

$$J_2 = \ln\left|S_w^{-1} S_b\right| = \ln|S_b| - \ln|S_w| \tag{2.23}$$

$$J_3 = trS_b - \mu(trS_w - c) \tag{2.24}$$

$$J_4 = \frac{trS_b}{trS_w} \tag{2.25}$$

where $|S|$ is the determinant of $S$, $\mu$ is a Lagrange multiplier and c is a constant.

$J_1$ and $J_2$ are invariant under any nonsingular linear transformation, while $J_3$ and $J_4$ are dependent on the coordinate system. The optimization of $J_1$ and $J_2$ will result in the same linear features, i.e., the trace and determinant criteria produce the same linear features for signal representation. Furthermore, these optimal features are the same no matter which combination of $S_b$ and $S_w$ is used [7]. In this dissertation, $J_1$ is used for optimization.

A pattern recognition system consists of two parts; a feature extractor and a classifier. If we look at the feature extraction block closely, it consists of a feature extraction block and a dimensionality reduction block, which are shown in Figure 2.7.

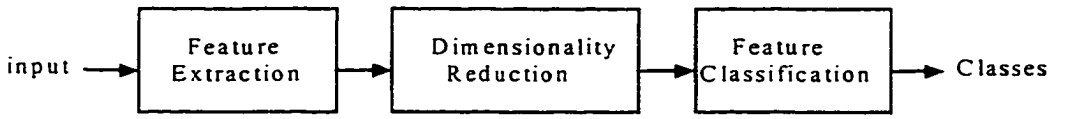



Figure 2.8. Structure of a classifier

## 2.5. DIMENSIONALITY REDUCTION

The best subset or combination of features for the purpose of classification needs to be determined. Reducing the dimensionality of the problem simplifies the task of the classifier. The main goal is to ensure that as much of the relevant information as possible is preserved in as few dimensions as possible. A classifier with fewer inputs has fewer adaptive parameters to be determined, leading to a classifier with better generalization properties. Dimensionality reduction strategies may be characterized as either feature selection or feature projection.

### 2.5.1. FEATURE SELECTION

The feature selection approach attempts to reduce the number of variables by selecting the best subset of the original feature set, according to some criterion. Feature selection necessarily consists of two components [7].

1. A criterion must be established by which it is possible to judge whether one subset of features is better than another.
2. A systematic procedure must be found for searching through candidate subsets of features.

Ideally, the selection criterion should be taken to be the probability of misclassification. In practice, evaluation of this criterion is generally too complex, and we have to resort to simpler criteria such as those based upon class separability. Similarly, in an ideal situation the search procedure should consist of an exhaustive search of all possible subsets. Exhaustive methods are often impractical due to computational complexity, and non-exhaustive searches and suboptimal searches are often used in practice [7].

### 2.5.2. FEATURE PROJECTION

As opposed to feature selection, which seeks to select the best subset of the original feature space for class separability, the goal of feature projection methods is to determine the best combination of the original features to form a feature set. For classification, the projection should map the data into separate clusters, one per class, facilitating the classification task.

2.5.2.1. Principal Components Analysis

Principal component analysis (PCA) [15] provides a linear map that minimizes the mean-square error (MMSE). PCA's effectiveness in pattern recognition is due to its ability to eliminate linear dependencies and uncorrelated noise in the data.

2.5.2.2. Separability based Dimensionality Reduction

Unlike Mean Square Error (MSE), which is the most widely used criteria for signal representation, class separability measures are typically invariant under any non-singular linear or non-linear transformation. However, any singular mapping used for dimensionality reduction results in the loss of some discriminating information. Our objective is to find the mapping that for a given reduction in space dimensionality provides the maximum class separability. In other words, we are searching among all possible singular transformations for the best subspace, which preserves class separability as much as possible in the lowest possible dimensional space, as illustrated in Figure 2.8. So we are seeking a linear transformation A from $\mathbf{R}^n$ to $\mathbf{R}^m$ with m<n such that

$$A: X \subset R^n \rightarrow Y \subset R^m \tag{2.26}$$

$$A = \arg\min_{A_o} \left\{ \left| J_X - J_{A_o^T X} \right| \right\} \tag{2.27}$$

where $J_X = tr(S^X)$ and $J_Y = tr(S^Y)$ are separabilities computed over the X and $Y = A^T X$ spaces respectively. Thus A optimizes $J_Y$, i.e. minimizes the drop in cost $\left| J_X - J_{A^T X} \right|$ incurred by the reduction in the feature space dimensionality. It can be shown that for such an optimum A

$$\left\{ \lambda_i^Y \right\} \subset \left\{ \lambda_j^X \right\} \quad i = 1,\ldots,m \quad j = 1,\ldots,n \tag{2.28}$$

where the $\lambda^X$ s and $\lambda^Y$ s are the eigenvalues of the corresponding separation matrices $S^X$ and $S^Y$ [7]. This observation and the fact that

$$J_Y = tr(S^Y) = \sum_{i=1}^{m} \lambda_i^Y \tag{2.29}$$

suggest that one can maximize (or minimize) $J_Y$ by taking the largest (or smallest) m eigenvalues of $S^X$. Note that the dimensionality m of the resulting set of feature vectors is rank(S) = min(n, L-1), where L is number of classes in our training set.

Therefore, the optimal linear transformation from the initial representation space in $\mathbf{R}^n$ to a low-dimensional feature space in $\mathbf{R}^m$ based on our selected separation measure results from projecting the input vectors x onto m eigenvectors corresponding to the m largest eigenvalues of the separation matrix $S^X$. These optimal vectors or direction can be obtained from a sufficiently rich training set and can be updated if needed.

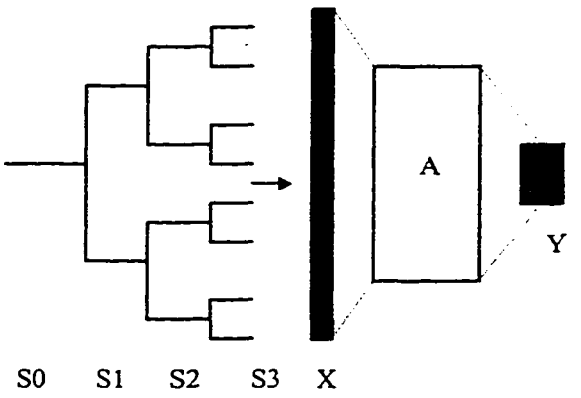



Figure 2.9. Dimensionality Reduction of the feature vectors

## 2.6. SUMMARY

The intent of this chapter has been to provide a global perspective on the problem of signal classification, and to provide some insight into the issues specific to texture classification and fault identification. The aspects of the classification problem are perhaps seen with greater clarity by partitioning the task into the stages of signal representation and classifier design. The process of signal classification consists of a feature extraction stage and a dimensionality reduction stage. It is proposed that the time-frequency domain provides a robust and versatile framework for feature extraction, with the expectation that this two-dimensional representation concentrates discriminant information more effectively than one-dimensional alternatives in either time or frequency. These time-frequency representations and the transforms with such representations are presented in the next chapter.

# CHAPTER 3

# WAVELETS AND SUB-BAND FILTER BANKS

This chapter gives a short introduction to the fundamentals of wavelets, sub-band filter banks and multiresolution analysis (MRA) [16]. The material presented in this chapter, although simplified, should be sufficient for understanding the fundamental principles of wavelets and sub-band filter banks. The chapter is only meant to serve as a building block for later chapters where individual aspects of the theory and the design of wavelets are treated more carefully as required in each chapter.

This chapter starts by expanding the signals in terms of wavelet basis functions and proceeds by representing the signals in the time-frequency plane. It reviews the fundamentals of wavelet theory and defines the scaling and wavelet functions. It then goes on to present the sub-band filter banks and their relationship with wavelets. The chapter proceeds by discussing various wavelet design techniques based on different design criteria. The fast implementation is in fact one of the primary reasons wavelets have attracted such an interest in signal processing, applied mathematics and engineering in general. Notice that in the interest of making this introduction to wavelets and MRA as simple and as clear as possible, the entire chapter has been limited to the discussion of compactly supported orthogonal dyadic (2-band) wavelets.

## 3.1. SERIES-EXPANSION OF SIGNALS

Given a signal from a space S and a set of signals $\{\varphi_i\}_{i\in Z}$ for that space so that $x$ can be written as

$$x = \sum_i \alpha_i \varphi_i \tag{3.1}$$

The set $\{\varphi_i\}$ is complete for the space S if all signals $x\in S$ can be expanded as in (3.1). In that case, there exists a dual set $\{\tilde{\varphi}_i\}_{i\in Z}$ such that the expansion coefficients can be computed as

$$\alpha_i = \sum_n \tilde{\varphi}_i[n]x[n] \tag{3.2}$$

when $x$ and $\tilde{\varphi}_i$ are real discrete-time sequences, and

$$\alpha_i = \int \tilde{\varphi}_i(t)x(t)dt \tag{3.3}$$

when they are real continuous-time functions. The above expressions are the inner products of the $\tilde{\varphi}_i$'s with the signal x, denoted as $<\tilde{\varphi}_i, x>$. When the set $\{\tilde{\varphi}_i\}$ is orthonormal and complete, it is an orthonormal basis for S and the basis and its dual are the same, that is $\varphi_i = \tilde{\varphi}_i$. Then

$$\langle \varphi_i, \varphi_j \rangle = \delta[i-j] \tag{3.4}$$

where $\delta[i]$ equal 1 if i=0, and 0 otherwise. If the set is complete and the vectors $\varphi_i$ are linearly independent but not orthonormal, the basis is a biorthogonal basis, and the basis and its dual satisfy

$$\langle \varphi_i, \tilde{\varphi}_j \rangle = \delta[i-j] \tag{3.5}$$

What is a good basis for S? The answer depends on the class of signals to be represented and on the choice of a criterion for quality. However, in general a good basis is one that allows compact representation. Desirable properties of the basis functions include computational efficiency, orthogonality, and good time-frequency localization. Expansions with some structure are of interest for complexity reasons. That is, expansions where the various basis vectors are related to each other by some elementary operations such as shifting in time, scaling, and modulation are of interest [16].

## 3.2. TIME-FREQUENCY REPRESENTATIONS

The primary goal of signal analysis is to extract information from a signal, relevant to a particular application. Time-frequency representations (TFR) combine time-domain and frequency-domain analyses to yield a potentially more revealing picture of the temporal localization of a signal's spectral characteristics. The time-frequency localization of the basis functions and the amplitude of their coefficients describe the signal's TFR.

When calculating the signal expansion, localization [17] of a given basis in time and frequency is the primary concern. The localization of a particular basis function is the spread of the function in time ($I_t$) and frequency ($I_\omega$). The intervals $I_t$ and $I_\omega$ contain 90% of the energy of the time and frequency domain

functions, and are centered around the center of gravity of $|f(t)|^2$ and $|F(\omega)|^2$. This is called a "tile" in the time-frequency domain [16]. A shift in time results in shifting the tile while modulation shifts the tile in frequency. By scaling the function, both the shape and the localization of the tile are affected. The analysis functions of the wavelet transform are defined as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}), a \in R^+ \tag{3.6}$$

where the function $\psi(t)$ is usually a band-pass filter. Thus, large a's (a>>1) correspond to long basis functions, and identify long-term trends in the signal to be analyzed. Small a's (0<a<1) lead to short term basis functions, which follow short term behavior of the signal (refer to Figure 3.1). This implies that: Scale is proportional to the duration of the basis functions used in the signal expansion or inversely proportional to the frequency. Scaling does not change the time-bandwidth product, it only exchanges one resolution for the other.

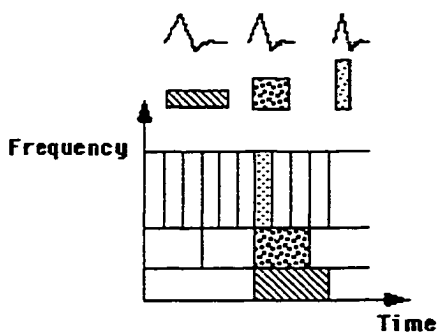



Figure 3.1 Shifts and scales of prototype Band-pass wavelet and Tilings of the Time-Frequency plane

TFRs may be divided into two groups by the nature of their transforms: linear methods (including the short-time Fourier transform and the wavelet transform) and quadratic methods (of which the Wigner-Ville distribution is fundamental). The concept central to linear methods is that of decomposing a signal into time-frequency atoms.

### 3.2.1. THE SHORT-TIME FOURIER TRANSFORM

The short-time Fourier transform (STFT) was first adopted by Gabor [18] to define a two-dimensional time-frequency representation. The Fourier transform of the windowed signal $x(\tau)g^*(\tau - t)$ yields the STFT:

$$STFT(t,f) = \int x(\tau)g^*(\tau - t)e^{-2j\pi f\tau}d\tau \tag{3.7}$$

The STFT has many useful properties, including efficient computation. The main drawback of the STFT is that even the most carefully chosen sampling grid is nonetheless constrained by the fact that each cell in the time-frequency plane must have an identical shape. The division of the frequency domain for the STFT is shown in Figure 3.2.a.

### 3.2.2. THE WAVELET TRANSFORM

A fundamental property of the wavelet transform (WT) is that the time resolution $\Delta t$ and the frequency resolution $\Delta f$ vary in the time-frequency plane. The dyadic wavelet transform is a constant relative bandwidth analysis with shifts and scales. The bandwidths of the analysis windows are spread logarithmically with respect to frequency, which is shown in Figure 3.2.b. To achieve changing time-frequency tiles because of the scaling, take a real band-pass filter with impulse response $\psi(t)$ and zero mean

$$\int_{-\infty}^{\infty}\psi(t)st = \Psi(0) = 0 \tag{3.8}$$

Thus, define the continuous wavelet transform as

$$CWT_f(a,b) = \frac{1}{\sqrt{a}}\int_R \psi^*(\frac{t-b}{a})f(t)dt \tag{3.9}$$

where $\psi(t)$ is a prototype window referred to as the mother wavelet and $a \in R^+$ and $b \in R$. So

$$CWT_f(a,b) = \langle \psi_{a,b}(t), f(t) \rangle$$

where $\psi_{a,b}(t)$ follows as in equation (3.6) and the factor $\frac{1}{\sqrt{a}}$ is used to conserve norm. The analysis determines the correlation of the signal with shifted and scaled versions of the mother wavelet, shown in Figure 3.3. This zooming in and out property makes wavelets extremely powerful for analyzing both time

and space localized phenomena as well as long term signal trends. The digital implementation of the CWT can be computed directly by convolving the signal with a scaled and dilated version of the mother wavelet, which is called Discrete Wavelet Transform (DWT).

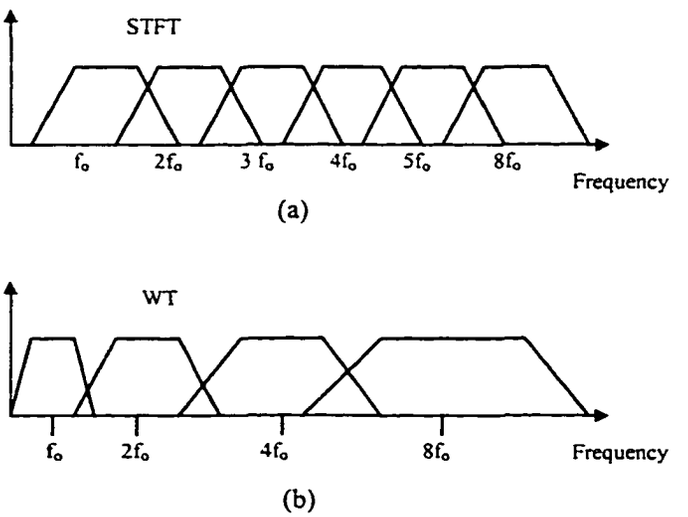



Figure 3.2. Division of the frequency domain for the STFT and the WT

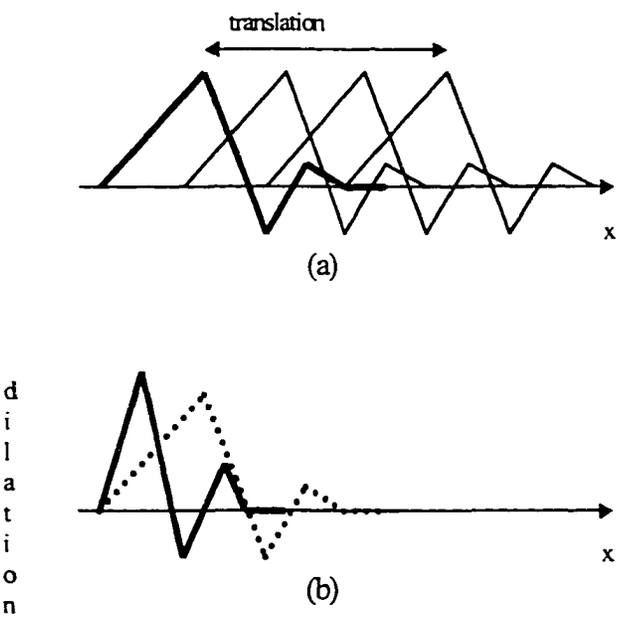



Figure 3.3. Dilation and translation

### 3.2.3. THE WAVELET PACKET TRANSFORM

The wavelet packet transform (WPT) is a generalized version of the CWT and the DWT. The transform is redundant, allowing one of many orthogonal bases to be chosen. As a result, the tiling of the time-frequency plane is configurable; the partitioning of the frequency axis may take many forms to suit the needs of the application. This is illustrated in Figure 3.4.b.

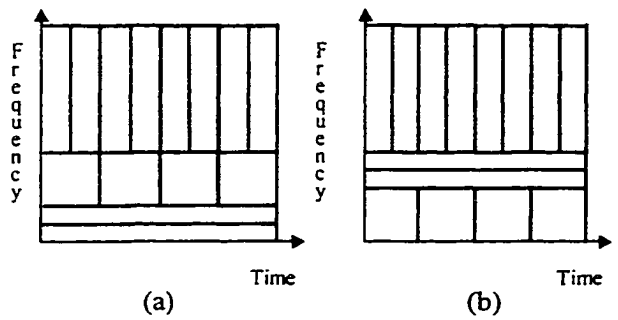



Figure 3.4. The time-frequency plane tiling of (a) wavelet basis (b) an arbitrary wavelet packet basis

### 3.2.4. TIME-FREQUENCY REPRESENTATIONS FOR SIGNAL CLASSIFICATION

The fundamental purpose of feature extraction for classification is to emphasize the important information in the data, and to de-emphasize that which is irrelevant. This implies transforming the raw data into a domain that presents the information contained in the signal more clearly: a map, which concentrates and localizes information. Time-frequency methods offer the ability to localize the energy distribution of a signal in time and frequency. The nature of the localization depends upon the method chosen.

The utility of the TFR as a feature extractor for pattern classification lies in its ability to describe important structures in the time-frequency plane. This requires an appropriate tiling of the time-frequency plane [19]. The time-frequency tilings for the transforms are:

- The STFT segments the time-frequency plane into rectangles of fixed aspect ratio
- The wavelet transform allows greater frequency resolution at lower frequencies and better time resolution at high resolutions

- The wavelet packet transform permits an arbitrary segmentation of the frequency axis. The tiling is the result of a basis selection procedure that optimizes a cost function chosen to evaluate the efficacy of the wavelet packet basis.

The time-frequency resolution of all the above are bounded by the Heisenberg[1] uncertainty. To be a good feature extractor, the TFR must cluster the information within each group class, and provide maximal discrimination of these clusters. When using the time-frequency plane as a feature space, it is imperative that the representation should provide good localization using as few TFR cells as possible to simplify the role of the pattern classifier.

### 3.3. MULTIRESOLUTION

The basic idea of multiresolution is successive approximation. A signal is written as a coarse approximation (typically a low-pass, subsampled version) plus a prediction error, which is the difference between the original signal and a prediction based on the coarse version. Reconstruction can be done by simply adding the prediction error to the prediction. This scheme can be iterated on the coarse version. The successive approximation approach is identical to wavelet decomposition since it performs a multiresolution analysis [20] on the signal. The decomposition into a coarse resolution, which gives an approximate but adequate version of the full image, plus a difference or detail image, is conceptually very important. Coarse and detail subspaces are orthogonal to each other, i.e. the detail signal is the difference between the fine and the coarse version of the signal. By applying the successive approximation recursively, the space of input signals can be spanned by spaces of successive details at all resolutions. This follows because, as the detail resolution goes to infinity, the approximation error goes to zero.

The applications reported in this work are two-dimensional, which can be extended from one-dimensional signal decompositions. Following the separable filter case, the two-dimensional decomposition can be obtained by performing a one-dimensional decomposition separately in each dimension. For easy

---

[1] If $f(t)$ vanishes faster than $\frac{1}{\sqrt{t}}$ as $t \to \pm\infty$, then $\Delta_t^2 \Delta_\omega^2 \geq \frac{\pi}{2}$, where equality holds only for Gaussian signals.

understanding, all the discussion and mathematics involved in this chapter are limited to one-dimension, which can be extended to two dimensions using separable decomposition.

### 3.3.1. SCALING FUNCTION

The principles of wavelet theory and multiresolution analysis [21] are best understood by first considering the function generating the multiresolution analysis, namely the scaling function, $\psi_0(x)$. A family of scaling functions can be generated by scaling and translating a function, possibly of compact support (on a real Euclidean distance space, compact is equivalent to closed and bounded). A family of dyadic functions, $\psi_{0,j,k}(x)$, is defined by

$$\psi_{0,j,k}(x) = 2^{\frac{j}{2}} \psi_0(2^j x - k) \quad \forall j,k \in Z \tag{3.10}$$

In the above equation, the scaling factor 2 can be replaced by any integer M. If the support of $\psi_0(x)$ is finite, then as j increases (decreases), e.g., as scale changes between fine and coarse, the translation step size decreases ( increases), and hence the scaling function $\psi_0(x)$ can be localized in both space-time and scale-frequency.

The multiresolution resolution principles can be understood clearly by introducing a sequence of successive approximation spaces, $W_{0,j}$ defined by

$$W_{0,j} = \underset{k}{span}\, \psi_{0,j,k}(x) \tag{3.11}$$

Moreover, to generate a multiresolution analysis, the closed subspaces $W_{0,j}$ should satisfy the following nesting property

$$\ldots W_{0,-2} \subset W_{0,-1} \subset W_{0,0} \subset W_{0,1} \subset W_{0,2} \subset \ldots \tag{3.12}$$

with

$$\lim_{j \to -\infty} W_{0,j} = \{0\}, \quad \lim_{j \to -\infty} \overline{W_{0,j}} = L^2(R) \tag{3.13}$$

However, conditions (3.12) and (3.13) are not sufficient to define a multiresolution analysis. In fact, in addition to the above scale space relations each of the scale spaces should be scaled versions of the central space $W_{0,0}$. That is, if $f \in L^2(R)$ the following must be true:

$$f(2^j x) \in W_{0,j} \quad \Leftrightarrow \quad f(x) \in W_{0,0} \tag{3.14}$$

then a multiresolution analysis exists. Furthermore, if $f(x) \in W_{0,0}$, then $f(x-k) \in W_{0,0}$ for all k∈Z and hence from (3.14) this implies that if $f(x) \in W_{0,j}$ then $f(x-2^{-j}k) \in W_{0,j}$ for all k∈Z. Finally, if one in addition require that $\psi_0(x)$ satisfy

$$\int_R \psi_0(x)\psi_0(x-k)dx = \delta(k) \tag{3.15}$$

then $\{\psi_{0,j,k}(x)\}_{k \in Z}$ is an orthonormal basis for $W_{0,j}$ for all j∈Z.

Clearly since $W_{0,0} \in W_{0,1}$, any function in the space $W_{0,0}$, spanned by $\psi_{0,0,k}(x)$, can uniquely be represented by a sum of basis functions in $W_{0,1}$, namely $\psi_{0,1,k}(x)$. Hence the fundamental scaling function, $\psi_0(x)= \psi_{0,0}(x)$, satisfies the two-scale dilation equation

$$\psi_0(x) = \sqrt{2}\sum_k h_0(k)\psi_0(2x-k) \tag{3.16}$$

where $h_0$ is defined to be the scaling filter relating $W_{0,0}$ and $W_{0,1}$ and determines the properties of $\psi_0$. If $h_0$ is a finite length N sequence (FIR filter) then $\psi_0(x)$ has compact support. The dyadic difference equation given by (3.16) is the most fundamental relation in building up the theory of wavelet analysis. In Figure 3.5 the relation between the multiresolution scale spaces given by (3.13)-(3.15) are pictorially illustrated.

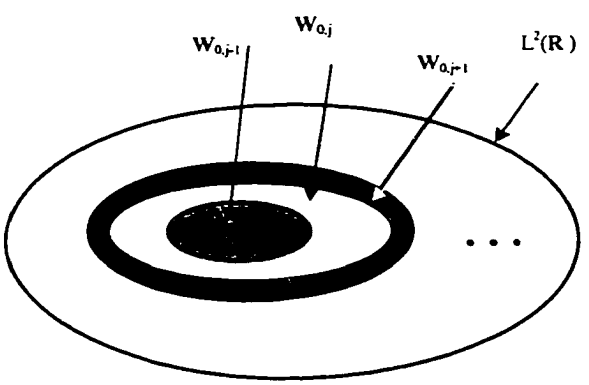



Figure 3.5. Illustration of the nesting of the scale spaces $W_{0,j}$

So far, we have only set forth the necessary and desired properties that a multiresolution decomposition should satisfy and we have not made any reference to the existence of such a sequence of

spaces. In the following, the conditions on $h_0$ for the existence of multiresolution analysis are shown. Daubechies [22] showed that if the scaling sequence $h_0$ was of finite length, N=2K, and furthermore satisfied

$$\sum_{n=0}^{N-1} h_0(n) = \sqrt{2} \tag{3.17}$$

$$\sum_{n=0}^{N-1} h_0(n) h_0(n+2l) = \delta(l) \tag{3.18}$$

then (3.16) is guaranteed to converge to a compactly supported function. Equation (3.17) is a necessary and sufficient condition for $L^2$ convergence of (3.16) [23], (3.18) is only a necessary condition for orthogonality of the scaling functions. We also observe that $h_0$ is a low-pass filter, that the frequency response of $h_0$ at $\omega=0$ is $\sqrt{2}$, and furthermore that the frequency response of $h_0$ must have a zero at $\omega=\pi$ [21].

A filter $h_0$ satisfying (3.17) is often referred in the filter bank literature as a quadrature mirror filter (QMF) [24].

### 3.3.2. WAVELET FUNCTION

While the scaling function defines a sequence of nested spaces generating an MRA, the wavelets define difference spaces (differences between the nested scale spaces). From the previous section, we recall that the scaling function $\psi_0(x)$, or equivalently the scaling filter $h_0$, uniquely specifies the multiresolution analysis. Associated with the scaling function is a family of wavelets,

$$\psi_{1,j,k}(x) = 2^{\frac{j}{k}} \psi_1(2^j x - k) \tag{3.19}$$

and if we require that the wavelets and scaling functions are orthogonal under integer translation

$$\int_R \psi_{0,i,k_1}(x) \psi_{1,j,k_1}(x) dx = 0 \quad \forall i,j,k_1,k_2 \in Z, i \le j \tag{3.20}$$

then $\left\{\psi_{1,j,k}\right\}_{j,k \in Z}$ is a basis for a subset of $L^2(R)$. Now let

$$W_{1,j} = \underset{k}{span}\, \psi_{1,j,k}(x)$$

where $W_{1,j}$ is the orthogonal complement of $W_{0,j}$ in $W_{1,j+1}$ and hence

$$W_{0,j+1} = W_{0,j} \oplus W_{1,j} \tag{3.21}$$

with

$$W_{1,j} \perp W_{1,j'} \qquad if \qquad j \neq j' .$$

Moreover, by reason of (3.12) and (3.13)

$$L^2(R) = \bigoplus_{n=-\infty}^{\infty} W_{1,n} = W_{0,\infty} \tag{3.22}$$

That is, $L^2(R)$ is decomposed into mutually orthogonal subspaces $W_{1,j}$. It then follows that if n > J then

$$\begin{aligned} W_{0,n} &= \bigoplus_{j=-\infty}^{n-1} W_{1,j} \\ &= W_{0,J} \oplus \bigoplus_{j=J}^{n-1} W_{1,j} \end{aligned} \tag{3.23}$$

which is obtained by recursively applying (3.21) as a refinement of the space $W_{0,j}$ spanned by $\left\{\psi_{0,j,k}(x)\right\}_{k\in Z}$. It is also worth noting that the wavelet spaces $W_{1,j}$ inherit the scaling property from $W_{0,j}$,

$$f(2^j x) \in W_{1,j} \quad \Leftrightarrow \quad f(x) \in W_{1,0} \tag{3.24}$$

as well as the invariance under integer translation property. Hence, if $f(x) \in W_{1,j}$ then $f(x - 2^{-j}k) \in W_{1,j}$ for all k∈Z. The relation between $W_{0,j}$ and $W_{1,j}$ is pictorially illustrated in Figure 3.6.

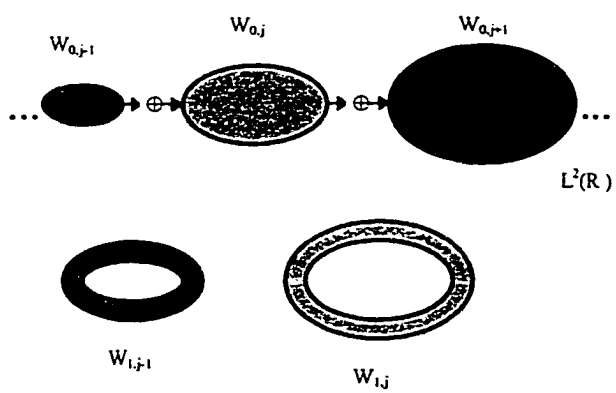



Figure 3.6. Illustration of the relationship between the scale spaces $W_{0,j}$ and the wavelet difference spaces $W_{1,j}$

Clearly from (3.21) and Figure 3.6, $W_{0,j} \subset W_{0,j+1}$ and hence any function in $W_{1,j}$ can be defined in terms of the basis functions of $W_{0,j+1}$. In particular, the wavelets $\psi_1(x) = \psi_{1,0}(x)$, are defined by the following dyadic scale recursion

$$\psi_1(x) = \sqrt{2} \sum_{k \in Z} h_1(k) \psi_0(2x - k) \tag{3.25}$$

and $h_1$ is the associated wavelet filter of real or complex coefficients relating the two spaces, $W_{1,0}$ and $W_{0,1}$. From (3.20) one can show, by applying (3.15), that necessarily

$$\sum_{k \in Z} h_0(k) h_1(k - 2l) = 0 \tag{3.26}$$

Using (3.26), a length N orthogonal filter $h_1$ can be obtained [24] from the length N scaling filter $h_0$ and is given by

$$h_1(l) = (-1)^k h_0(N - 1 - k) . \tag{3.27}$$

Also, if (3.17) and (3.18) are satisfied then (3.27) implies that

$$\sum_{k} h_1(k) = 0 \tag{3.28}$$

and hence $h_1$ is a complementary high-pass filter. Using filter bank terminology, the filter pair $h_0$ and $h_1$ are called a quadrature mirror filter bank (QMF).

A unique compactly supported 2-band wavelet basis exists when condition (3.17) holds if in addition the $\frac{N}{2}$ quadratic constraints given by (3.18) are satisfied. Satisfying (3.17) and (3.18) leaves $\frac{N}{2} - 1$ free parameters for designing the scaling filter. However, it is not always clear what properties one should ask $\psi_0(x)$ to have. Finally, since $\psi_1(x)$ is generated by finite linear combinations of $\psi_0(x)$, mathematical properties of $\psi_1(x)$ can be derived from the properties of $\psi_0(x)$.

### 3.3.3. THE SCALING AND WAVELET FUNCTIONS

Although we rarely perform computation using the wavelet and the scaling function directly, we should be able to compute and display both of them. There are two fundamentally different methods for

computing these functions [21]. The first method computes the values of the function on the dyadic rational and is based on the evaluation of (3.10) over the integers giving raise to

$$\psi_0(k) = \sqrt{2}\sum_{n=0}^{N-1} h_i(n)\psi_0(2k-n) \tag{3.29}$$

The recursion is initialized with the solution to the eigenvector associated with the eigenvalue of unity of the eigenvalue problem obtained by evaluating (3.29) for k = 0,1,...,N-1. An alternative method for generating the functions is successive approximation. This is based on solving the basic recursive equation (3.10) iteratively. That is, if $\psi_i^k(x)$ denotes the $k^{th}$ iteration then

$$\psi_0^k(x) = \sqrt{2}\sum_{n=0}^{N-1} h_i(n)\psi_0^{k-1}(2x-n) \tag{3.30}$$

which in the limit converges to $\psi_0(x)$ independent of the shape of the initializer $\psi_i^0(x)$.

With either of these algorithms the actual functions can be computed and displayed. Figure 3.7 gives several examples of classical Daubechies wavelets denoted by $D_N$ where N indicates the length of the support.

### 3.4. WAVELET SELECTION

The choice of mother wavelet depends very much on the nature of the signals and the goal of the signal processing. The wavelet basis vectors have the following important properties.

1. **Regularity**: Wavelet regularity is also known as polynomial regularity or smoothness. A K-polynomially regular wavelet system is a wavelet system for which the first K wavelet moments vanish. Daubechies wavelet bases are called maximally regular wavelet bases or K-regular wavelet bases because they have a maximal number of wavelet moments set to zero.
2. **Vanishing Moments**: The first K discrete moments of $h_1$ are zero. That is choose $h_0$ such that the obtained $h_1$ satisfies the condition

$$\mu(1,k) = \sum_n n^k h_1(n) = 0 \quad for \quad k = 0,1,..,K-1 \tag{3.31}$$

where $K = \frac{N}{2}$ . Furthermore, Daubechies showed that setting $\mu(l,k) = 0 \quad for \quad k = 0,...K-1$ improves the smoothness of the associated wavelets with increasing K [22]. By requiring that a maximum number of moments vanished (K=N/2), Daubechies was able to design wavelets of length N that could represent polynomials of degree up to K-1 exactly. Although this enabled her to get good solutions, it is not clear that the vanishing moment property has particular advantages in signal processing. Although vanishing moments and smoothness are related, maximizing the number of vanishing moments does not yield optimal smoothness for the given number of parameters [21].

3. **Compact Support:** Compactly supported functions on any real Euclidean distance space are nonzero only on a set that is both closed and bounded. This property is important for efficient and exact numerical implementation [25].

Some wavelets are better than others for specific applications, with respect to the properties listed above. In general, because of these properties, wavelet bases generate very efficient and simple representations for piecewise smooth signals and images. The manner in which vanishing moments, regularity and compact support affect the wavelet's efficacy as a basis for signal classification is not clear. One would expect that a wavelet that "looks like" the elemental components of the signals under consideration would be the most appropriate. For a given wavelet, it is reasonable to expect that the small scales would capture the local activity, while larger scales would model longer-duration trends in the signal. More important is the ability of the wavelet basis to generate a TFR that clearly distinguishes signals in different classes. This requires that the wavelet functions appropriately model the signal, and that they be well localized and well behaved in the time-frequency plane.

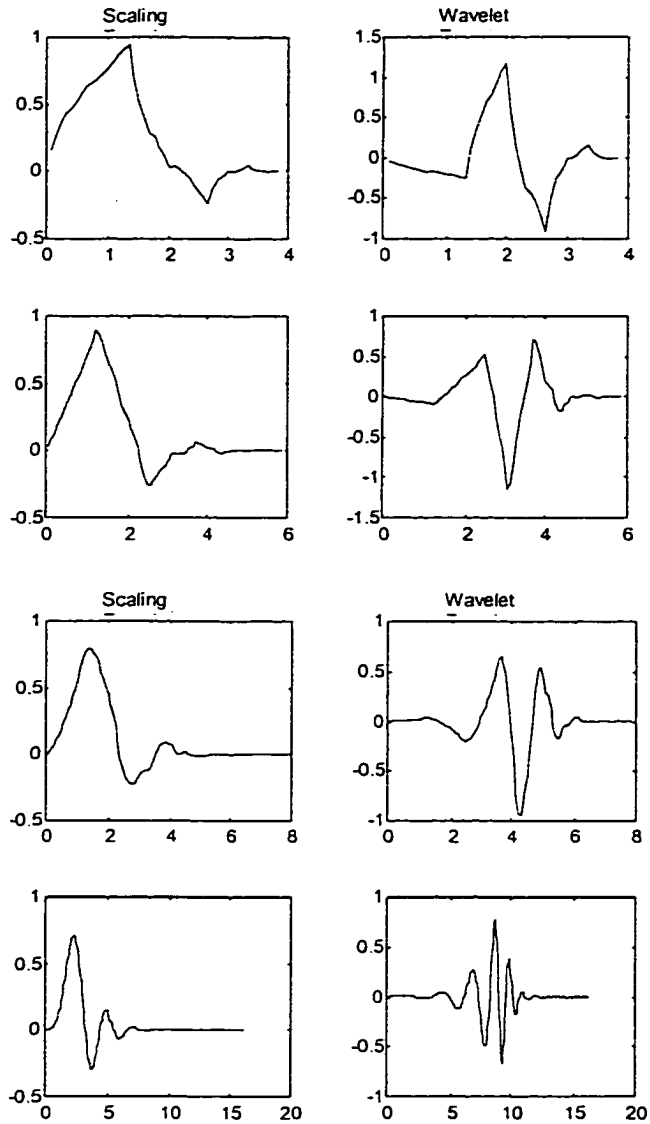



Figure 3.7. Daubechies Scaling and Wavelet functions $D_4$, $D_6$, $D_8$, and $D_{16}$

## 3.5. TWO-CHANNEL FILTER BANKS

So far, we have been concerned with the wavelet bases and their properties. In this section, we examine computation with wavelets, and in particular an efficient algorithm for obtaining the approximated wavelet coefficients from samples of the signals.

To compute the wavelet transform, one has to evaluate a nontrivial integral. However, it is well known that the wavelet transform coefficients can be approximated using a filter bank approach, giving rise to the familiar discrete wavelet transform (DWT). In fact, rather than having to evaluate the integral in (3.32), the

discrete wavelet transform can be computed using only the scaling and wavelet filters. Hence, one does not have to deal explicitly with the scaling and wavelet function for most applications.

$$d_{i,j,k} = \int_R f(x)\psi_{i,j,k}(x)dx \tag{3.32}$$

### 3.5.1. ANALYSIS FILTER BANK - DISCRETE WAVELET TRANSFORM

A relation between the transform coefficients and the scaling and wavelet filters can be derived from (3.32) using (3.16) and (3.25).

$$\begin{aligned}
d_{i,j,k} &= \int_R f(x)\psi_{i,j,k}(x)dx \\
&= \int_R f(x)2^{\frac{j}{2}}\psi_i(2^j x-k)dx \\
&= \int_R f(x)2^{\frac{j}{2}}\sum_{n\in Z} h_i(n)\psi_0(2(2^j x-k)-n)dx \\
&= \int_R f(x)2^{\frac{j}{2}}\sum_{m\in Z} h_i(m-2k)2^{\frac{1}{2}}\psi_0(2^{j+1}x-m)dx \\
&= \sum_{m\in Z} h_i(m-2k)\int_R f(x)2^{\frac{(j+1)}{2}}\psi_0(2^{j+1}x-m)dx \\
&= \sum_{m\in Z} h_i(m-2k)\int_R f(x)\psi_{0,j+1,m}(x)dx \\
&= \sum_{m\in Z} h_i(m-2k)d_{0,j+1,m} \\
&= [\downarrow 2]\{h_i(k-m)*d_{0,j+1,m}\}_m
\end{aligned} \tag{3.33}$$

Hence, the coarse resolution scaling and wavelet coefficients are obtained from the fine resolution scaling coefficients by way of convolution with the appropriate filter (time reversed) followed by time sampling or decimation by 2 ($\downarrow 2$). In Figure 3.8 the procedure for computing the discrete wavelet transform coefficients is illustrated.

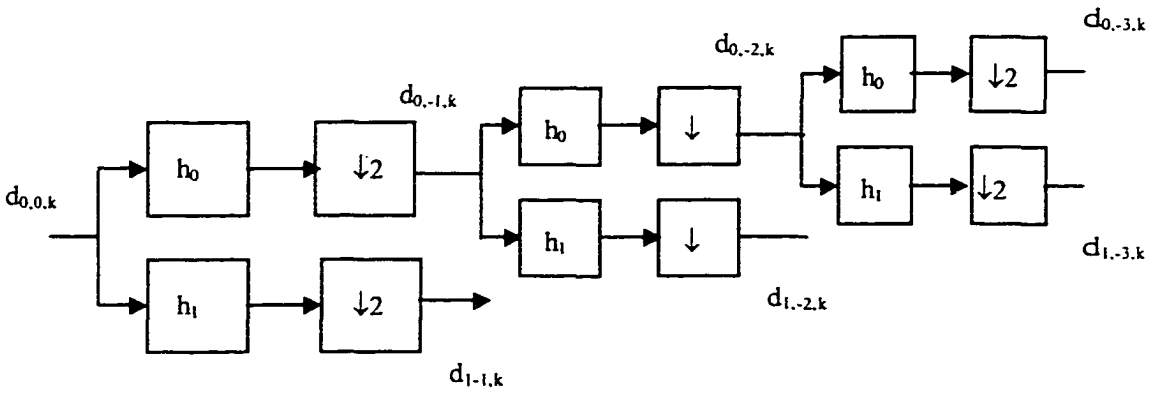



Figure 3.8. Wavelet analysis or discrete wavelet transform

From the derivation of the DWT algorithm given by (3.33) and Figure (3.8), we see that the DWT coefficients are computed without evaluating the integral.

### 3.5.2. SYNTHESIS FILTER BANK - INVERSE DISCRETE WAVELET TRANSFORM

Similar to the analysis filter bank or the DWT, the synthesis filter bank or signal reconstruction is obtained as a weighted sum of wavelet coefficients without the need to deal with the basis functions themselves. A function g(x) can be represented at the coarser scale (e.g., j-1) in terms of both the scaling and wavelet functions as follows

$$
\begin{aligned}
g(x) &= \sum_{k\in Z} d_{0,j-1,k}\,\psi_{0,j-1,k}(x) + \sum_{k\in Z} d_{1,j-1,k}\,\psi_{1,j-1,k}(x) \\
&= \sum_{k\in Z} d_{0,j-1,k}\,2^{\frac{(j-1)}{2}}\,\psi_0(2^{j-1}x-k) + \sum_{k\in Z} d_{1,j-1,k}\,2^{\frac{(j-1)}{2}}\,\psi_1(2^{j-1}x-k) \\
&= \sum_{k\in Z} d_{0,j-1,k}\,2^{\frac{j}{2}}\sum_n h_0(n)\psi_0(2^j x-(2k+n)) + \sum_{k\in Z} d_{1,j-1,k}\,2^{\frac{j}{2}}\sum_n h_1(n)\psi_0(2^j x-(2k+n)) \\
&= \sum_{i=0}^{1}\left[\sum_{k\in Z} d_{i,j-1,k}\,2^{\frac{j}{2}}\sum_l h_0(l-2k)\psi_i(2^j x-l)\right]
\end{aligned}
\tag{3.34}
$$

Multiplying (3.34) by $\psi_{0,j,m}(x)$ on the left and right and integrating over x, it can be shown using (3.15) and (3.20) that

$$d_{0,j,k} = \sum_{i=0}^{1}\left\{\sum_{m} h_i(k-2m)d_{i,j-1,m}\right\}$$
$$= \sum_{i=0}^{1}\left\{h_i(m) * \left\{[\uparrow 2]d_{i,j-1,m}\right\}\right\}_m \tag{3.35}$$

and hence the fine resolution scaling coefficients are obtained by up-sampling ($\uparrow$2)the coarse resolution scaling and wavelet coefficients by 2, convolving with the respective filters (i.e., $h_0$ and $h_1$), and adding the result. In Figure 3.9 the synthesis or IDWT is illustrated graphically.

Quadrature mirror filters (QMF) allow a signal to be split into two down-sampled sub-band signals and then reconstructed without aliasing, although non-ideal filters are used. Any input signal can be transformed using a two-channel filter bank, with filters $h_0[n]$ and $h_1[n]$, followed by down-sampling by 2.

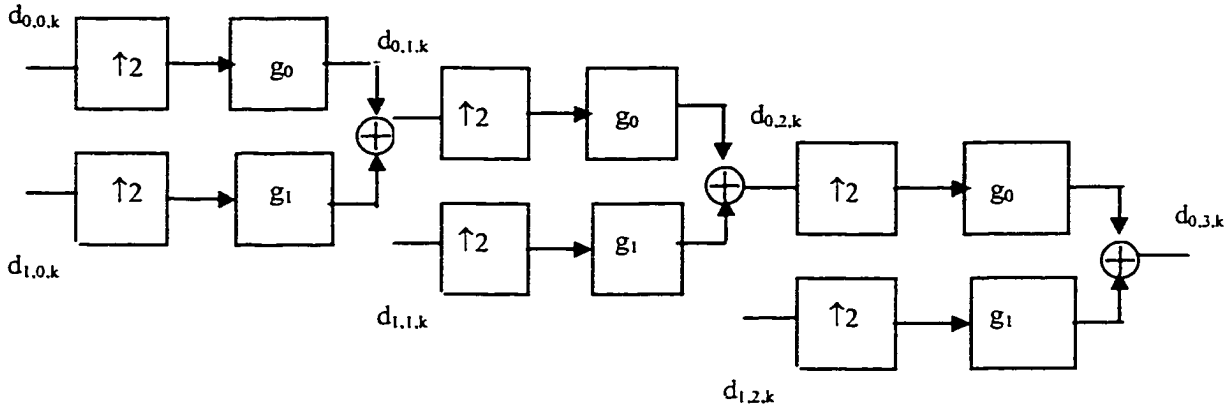



Figure 3.9. Wavelet synthesis or IDWT

## 3.6. OPTIMAL DESIGN AND PERFORMANCE

No function can be perfectly localized in both the time and frequency domains [16]. The desired features of the design depend on the application under consideration. The optimality concepts and their measures are constrained by the limits of time and frequency localization and the completeness requirements. The optimal criteria might be based on a single design measure or a set of measures. Several optimality measures are reviewed here [26], [27], [28].

### 3.6.1. ENERGY COMPACTION

The energy compaction measure is derived from rate-distortion theory and merges with the entropy minimization measure for Gaussian sources. The energy compaction measure for unitary transforms is

$$G_{TC} = \frac{\sigma_\varepsilon^2(PCM)}{\sigma_\varepsilon^2(TC)} = \frac{\frac{1}{N}\sum_{j=0}^{N-1}\sigma_j^2}{\left[\prod_{j=1}^{N-1}\sigma_j^2\right]^{1/N}} \tag{3.36}$$

This is the ratio of the arithmetic mean of the energy, $\{\sigma_j^2\}$, in each sub-band to the geometric mean. This measure is widely used in evaluating the performance of block and sub-band transforms. This measure shapes the frequency responses of the filter functions for the given input spectrum. Hence, the solution is a matched filter bank.

### 3.6.2. ALIASING ENERGY

Any realizable lossless decomposition technique performs an aliasing cancellation in order to achieve perfect reconstruction. Because of the quantization (discarding) of some of the subbands, non-canceled aliasing signal components may exist in the reconstructed signal in practice. The aliasing energy component at the reconstructed low-pass filter output is:

$$\sigma_{LA}^2 = \frac{1}{2}\int_{-\pi}^{\pi}\left|H(e^{-j\omega})\right|^2 S_{xx}(e^{j(\omega+\pi)})\left|H(e^{j(\omega+\pi)})\right|^2 d\omega \tag{3.37}$$

where $S_{xx}(e^{j\omega})$ is the input spectrum. The optimal solution based on this measure minimizes the aliasing energy component of the low-pass branch.

### 3.6.3. UNIT STEP RESPONSE

The time-local features of signals can be effectively represented with bases that consist of functions well localized in the time domain. The unit-step response of the filter is widely used as a measure of time localization and is defined as

$$a(n) = h(n) * u(n) \tag{3.38}$$

where u(n) is the unit step function. The energy difference between the unit step response a(n) and the unit step function u(n) is expressed as

$$E_s = \sum_{k=0}^{N-1}\left[\left(\sum_{n=0}^{k} h(n)\right) - 1\right]^2 \tag{3.39}$$

$E_s$ becomes zero when $h(n)=\delta(n)$. This is the best time-local solution with the worst frequency selectivity. This measure is a mathematical tool to monitor the time domain properties of the designed filter.

### 3.6.4. ZERO-MEAN BAND-PASS AND HIGH-PASS FILTERS

Many practical signal sources have a significant spectral component at zero frequency. Therefore, an efficient decomposition technique should be able to represent the DC component with only basis functions. The high-pass filter of a two-band PR-QMF banks should be constrained to have zero-mean (or a DC gain of zero) via

$$\sum_n (-1)^n h(n) = 0 \tag{3.40}$$

This requirement implies that there must be at least one zero of the low-pass prototype filter $H(e^{j\omega})$ at $\omega=\pi$. That implies some degree of regularity in the context of wavelet transform theory.

### 3.6.5. UNCORRELATED SUB-BAND COEFFICIENTS

The Karhunen-Loeve Transform (KLT) is the unique example of block transforms with perfectly uncorrelated transform coefficients for the given input statistics. The uncorrelatedness and maximum energy compaction requirements are available in the KLT solution. However, this is not the case with filter banks. The cross-correlation of the two sub-band signals $y_0(n)$, $y_1(n)$ at the same time index is

$$R_{LH}(0) = \sum_n \left[\sum_l h(l)(-1)^l h(n-l)\right] R_{xx}(n) \quad \forall n \tag{3.41}$$

In general, there is more than one filter solution that satisfies the perfect decorrelation condition, $R_{LH}(0)=0$. The one that maximizes the objective function for optimization is the desired solution.

### 3.6.6. MAXIMALLY FLAT FREQUENCY RESPONSE

The fundamental question is the degree of flatness in the frequency response. The sharpness of the transition band has been investigated since the inception of filter design in the signal processing filed. The most common tool for flatness is the number of z-plane zeros in the filter at $\omega=\pi$. This condition forces the filter bank solution to have zero-mean for the band-pass and high-pass functions. This degree of smoothness might be necessary for some signal processing applications. Wavelet regularity was proposed as a measure of wavelet basis design. The number of zeros at z = -1 in the low-pass prototype filter in a two-band PR-QMF bank was used as the design tool for wavelet and scaling bases. Daubechies proposed a wavelet basis obtained by placing, the maximum possible number of zeros of the low-pass filter at $\omega=\pi$. The following optimality measure is from [29].

### 3.6.7. TIME-FREQUENCY LOCALIZATION

The time and frequency centers and spreads are defined as

$$\sigma_n^2 = \frac{\sum_{-\infty}^{\infty} (n-\bar{n})^2 |x(n)|^2}{E} \tag{3.42}$$

and

$$\sigma_\omega^2 = \frac{\frac{1}{2\pi}\int_{-\pi}^{\pi} (\omega-\bar{\omega})^2 \left|X(e^{j\omega})\right|^2 d\omega}{E} \tag{3.43}$$

All the measures of design discussed in this section merely shape time and frequency features of the filters or basis functions. They can be directly used as the design criteria.

## 3.7. BEST BASIS SELECTION

The power of the wavelet packet transform is that a "best basis" can be chosen for a specific task if it can be properly identified from an ensemble of possible candidates. To determine the best basis, it is necessary to evaluate and compare the efficacy of many bases. To this end, the cost function must be

chosen to represent the goal of the application. The best-basis selection algorithm has its origins in signal compression [30] and the cost functions [31], [32] associated with compression entail the use of some form of entropy measure. This form of best basis algorithm is the simplest and is used to introduce the concept of the best basis selection. Subsequently, it is shown how the algorithm may be modified to suit the classification problem. The best basis algorithm operates on a binary tree of subspaces.

### 3.7.1. BEST BASIS SELECTION FOR SIGNAL COMPRESSION

The best basis selection algorithm operates on a single signal. The best-basis algorithm proposed by Coifman and Wickerhauser [30] is a divide and conquer search of a binary tree in which one begins with a fully decomposed tree, starts at the lowest level, and eliminates branches until an optimal solution is found.

The cost function associated with the pruning algorithm is based on entropy since the goal in signal compression is to maximize the information with respect to the chosen set of coordinate axes. A natural choice is the Shannon entropy:

$$H(P) = \sum_i p_i \log_2 p_i \tag{3.44}$$

where P={p} is a nonnegative sequence with $\sum_i p_i = 1$. Other entropy measures are possible [30] with varying effects on the outcome of the algorithm. A brief description of the pruning algorithm follows.

Consider a single subspace $W_{j,k}$ within a binary packet tree. Let $\psi_{j,k}$ denote a set of basis vectors belonging to the subspace $W_{j,k}$, arranged in matrix form:

$$\psi_{j,k} = \left[\psi_{j,k,0}, \psi_{j,k,1}, \cdots, \psi_{j,k,2^{n_0-j}-1}\right]^T$$

Let $A_{j,k}$ represent the best basis for the signal x restricted to the span of $\psi_{j,kl}$ and let E be the chosen information cost function. The algorithm given in Figure 3.10 "prunes" the binary tree by comparing the cost function of each parent node with its two children.

When the algorithm has completed, we have the best basis $A_{0,0}$ for the signal x restricted to the span of $\psi_{0,0} \equiv R^N$. The chosen best basis consists of a disjoint set of subspaces, and each subspace $W_{j,k}$ contains

$2^{n_0-j}$ basis vectors. The total number of basis functions is always N, where $N = 2^{n_0}$ is the length of each signal x. To make this algorithm fast, the cost function E must be additive: $E(\{x_i\}) = \sum_i E(x_i)$ so that

$$E(A_{j+1,2k}x \cup A_{j+1,2k+1}x) = E(A_{j+1,2k}x) + E(A_{j+1,2k+1}x) \tag{3.45}$$

This implies that a simple addition suffices instead of computing the cost of the union of the nodes. The proof that this algorithm yields the best basis relative to an additive form of E may be found in [30]. Given the best basis, the transform must find only those coefficients corresponding to the chosen subspaces rather than all of the coefficients in the entire binary packet tree.

Given a signal x

Step 1. Choose a time-frequency decomposition method. That is, specify a wavelet packet transform (QMF's). Specify the depth of the decomposition J, and an information cost function E.

Step 2. Decompose x into its binary packet tree, and obtain the coefficients $\{\psi_{j,k}x\}$ for $0 \le j \le J$ and $0 \le k \le 2^j - 1$.

Step 3. Begin at level J: set $A_{j,k} = \psi_{j,k}$ for $k = 0,...2^J - 1$.

Step 4. Determine the best subspaces $A_{j,k}$ for j = J-1,...,0, k = 0,...,$2^j$-1 by

$$A_{j,k} = \begin{cases} \psi_{j,k} & \text{if} \quad E(\psi_j x) \le E(A_{j+.2k}x \cup A_{j+.2k} \\ A_{j+1,2k} \oplus A_{j+1,2k+1} & \text{otherwise} \end{cases}$$

Figure 3.10. The Best Basis Algorithm

### 3.7.2. BEST BASIS SELECTION FOR CLASSIFICATION

Fundamental to the success of any classifier is the quality of the feature set with which it is provided. The desirable properties of a feature set for classification are:

- the statistical distance between classes are maximized, and
- the feature set supplies the most important features while suppressing the redundant ones.

These concepts are emphasized in Chapter 2. This previous section presented the adaptive basis selection algorithm for signal compression. This section presents an algorithm for selecting the best basis for signal classification.

3.7.2.1. Discriminant Measures

In order to determine the best basis for classification among the ensemble of redundant bases in a complete packet decomposition, it is necessary to establish a measure of discriminant power. As explained in Chapter 2, the ideal criterion would be the probability of misclassification, evaluated upon each candidate basis. In practice, evaluation of this criterion is generally too complex because of its complex mathematical expression and one must resort to mathematically simpler criteria such as class separability. Additionally, an ideal evaluation would have each of the $2^N$ possible orthonormal bases compared in terms of the discriminant power. A suboptimal technique that requires far less computation is to prune the packet tree by evaluating the individual discriminability of each sub-band. This pruning algorithm is discussed in the next section.

It can be assumed that class separability is the most practical measure of discriminant power as it has a closed form expression. An n-feature discriminant measure can be defined as D(p,q), where $p = \{p_i\}_{i=1}^n, q = \{q_i\}_{i=1}^n$ are measures used to represent the n features. If $p_i$ and $q_i$ are scalars, then the discriminant measure may take one of the following forms:

1. Relative Entropy:

$$D(p,q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} \tag{3.46}$$

Relative entropy measures the discrepancy of p from q. The drawback to this measure is that it is not symmetric in p and q: characteristics of the features in p with respect to q does not yield the same measure if the class order is reversed. This may tend to bias the relative entropy measure toward the activity in one class over that of another. This is desirable if the goal is to separate the signal from noise, but does not give a fair treatment of the classes in a pattern recognition system.

2. Symmetric Relative Entropy:

$$D(p,q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} + \sum_{i=1}^{n} q_i \log \frac{q_i}{p_i} \tag{3.47}$$

Symmetric relative entropy yields symmetric activity among the classes.

3. Euclidean Distance:

$$D(p,q) = \|p - q\| = \sum_{i=1}^{n} (p_i - q_i)^2 \tag{3.48}$$

The Euclidean distance is another symmetric measure.

In general, it is necessary to discriminate from among more than two classes. To compute the discrepancy between the distributions of K classes, one must take $\begin{bmatrix} K \\ 2 \end{bmatrix}$ pairwise combinations of D:

$$D\left(\left\{P^{(k)}\right\}_{c=1}^{K}\right) = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} D\left(P^{(i)}, P^{(j)}\right) \tag{3.49}$$

3.7.2.2. The Local Discriminant Basis Algorithm

The local discriminant basis (LDB) algorithm originally developed by Saito [33] selects the binary wavelet packet tree, which most discriminates data from a given set of classes. The measure of class separability is conveyed by the discriminant measure D. In order to optimize the classification with respect to the time-frequency localization characteristics of the wavelet packet basis, the input parameters to D are the time-frequency energy maps of each class.

Let $\left\{x_i^{(c)}\right\}_{i=1}^{N_c}$ be a set of training signals belonging to class c, where $N_c$ is the number of patterns in that class. The time-frequency energy map of class c is a table of positive real values indexed by (j,k,n):

$$\Gamma_c(j,k,n) = \frac{\sum_{i=1}^{N_c} \left(W_{j,k,n} X_i^{(c)}\right)^2}{\sum_{i=1}^{N_c} \left\|X_i^{(c)}\right\|^2} \quad \text{for j=0,...,J, k=0,...,}2^j\text{-1, n=0,...,}2^{n_0-j}-1. \tag{3.50}$$

That is, $\Gamma_c$ is computed by accumulating the squares of the transform coefficients for each entry in the binary packet tree (j,k,n), and normalizing by the total energy of the signal belonging to class c.

Since the algorithm must choose the best set of subspaces from the binary packet tree, the response from individual temporal locations from within a subspace must be added. For K classes, the overall discriminant measure for the subspace $W_{j,k}$ is thus:

$$D\left(\left\{\Gamma_c(j,k,:)\right\}_{c=1}^{K}\right) = \sum_{n=0}^{2^{n_0-j}-1} D(\Gamma_1(j,k,n),\ldots,\Gamma_K(j,k,n)) \tag{3.51}$$

Let $\psi_{j,k}$ denote a set of basis vectors belonging to the subspace $W_{j,k}$, arranged in matrix form:

$$\psi_{j,k} = \left[\psi_{j,k,0},\psi_{j,k,1},\ldots,\psi_{j,k,2^{n_0-j}-1}\right]^T$$

Let $A_{j,k}$ represent the LDB for the training set restricted to the span of $\psi_{j,k}$, and let $\Delta_{j,k}$ be a work array containing the discriminant measure of the node (j,k). The algorithm is shown in Figure 3.11. When step 3

Given a training dataset consisting of K classes of signals $\left\{\left\{X_i^{(c)}\right\}_{i=1}^{N_c}\right\}_{c=1}^{K}$,

Step1. Choose a time-frequency decomposition method. Specify the depth of the decomposition J and the discriminant measure D.

Step2. Construct the time-frequency energy maps $\Gamma_c$ for c=1,..., K.

Step3.Begin at level J: set $A_{j,k} = \psi_{j,k}$ and $\Delta_{J,k} = D\left(\left\{\Gamma_c(J,k,:)\right\}_{c=1}^{K}\right)$ for k=0,..., $2^J$ -1.

Step4. Determine the best subspaces A for j=J-1,...,0, k=0,... ,$2^j$ -1 by the following rule:

Set $\Delta_{j,k} = D\left(\left\{\Gamma_c(j,k,:)\right\}_{c=1}^{K}\right)$

if $\Delta_{j,k} \geq \Delta_{j+,2k} + \Delta_{j+,2k+1}$,

then $A_{j,k} = \psi_{j,k}$,

else $A_{j,k} = A_{j+1,2k} + A_{j+1,2k+1}$ and $\Delta_{j,k} = \Delta_{j+1,2k} + \Delta_{j+1,2k+1}$.

Step5. Order the N basis functions in the LDB by their power of discrimination

Step 6. Use the L (<<N) most discriminating basis functions in the LDB for classifier features.

Figure 3.11. The Local Discriminant Basis Algorithm

has been completed, we have the best basis $A_{0,0}$ which is the LDB restricted to the span of $\psi_{0,0} \equiv R^N$. The chosen LDB consists of a disjoint set of subspaces, and each subspace $W_{j,k}$ contains $2^{n_0-j}$ basis vectors. The total number of basis functions is always N, where $N = 2^{n_0}$ is the length of each signal $x_i^{(c)}$. Once the LDB has been selected, the N transform coefficients, each corresponding to a basis vector within the LDB, may be used as features for a classifier. It is desirable to reduce the dimensionality of the representation for a classifier feature set. In the algorithm this is done using feature selection methods. The basis functions in the LDB must be ranked to determine those that are the most important for classification. As with the discriminant measure used for selecting among subbands to determine the LDB, a measure of class separability is used to assess the discriminant power of each basis function within the LDB. The dimension of the representation is reduced from N to L by keeping only the bases, which provide the most discriminant information in terms of the time-frequency energy distributions between classes. The best value of L depends upon the problem, the nature of the data and the type of the classifier. In general, this can be determined empirically.

Feature projection methods have not been used for dimensionality reduction. Certainly the information tends to be dispersed throughout the time-frequency plane; it is difficult to retain the class separability information in a low-dimensional feature set using feature selection. Feature projection methods may prove to be superior to feature selection methods as they seek to find the best combination of all features in a low-dimensional projection.

In all the algorithms discussed, the QMF-filters used are fixed. The influence of using different filters has not been studied with respect to any wavelet packet structure for the purpose of discrimination or signal classification. The dissertation focuses on this study throughout the next chapters. Also, the influence of using measures other than the time-frequency energy maps, as well as their influence on discrimination of features needs to be studied. Initially, empirically it is shown that the role of QMF filter is important in the task of signal classification and then we proceed by designing the optimal wavelet based on class separability, which significantly improves the classification performance. We extend the work to use the same design technique for fault identification in printed circuit boards using infrared imaging.

## 3.8. SUMMARY

This chapter has provided the background necessary to develop a means of wavelet-based feature extraction for signal classification. The mathematics and concepts of wavelet theory have been introduced, and extended to include the wavelet packet transform. It is shown that the basis can be determined optimally to localize discriminant information by pruning a packet decomposition according to a class separability criterion.

In the next chapter, various wavelet packet tree structures based on different criteria are analyzed. The performance of the WT and WPT based feature sets are evaluated in the context of texture classification. This chapter also compares the performance of feature projection based dimensionality reduction to the feature selection methods presented in the previous chapters. Finally, the influence of using various feature extracting measures on classification performance is studied.

# CHAPTER 4

# CHOICE OF BEST BASIS FOR TEXTURE CLASSIFICATION

During last few years, there have been many studies of classification of signals and images [34] - [39]. A variety of descriptors based on statistical, structural, and spectral characteristics of the single or multidimensional signals are used to form the best sets of discriminant features. This chapter provides an analysis of two multi-scale basis selection algorithms that are used to generate features for classification of textures. The necessity to use criteria dependent basis algorithms for signal classification is presented in the Chapter 3. The chapter provides an investigation of the classification performance of feature extraction and dimensionality reduction strategies for the textures, which are the major blocks in a general classification system.

This chapter is organized as follows. Section 4.1 describes the textures used in analysis and motivates the use of multiresolution based methods and tree decompositions for texture classification. The criterion used for the wavelet tree decomposition should be chosen depending on the problem [40]. Energy-based tree decomposition algorithm developed by Kuo *et.al* [41], [42] is presented in Section 4.2. The optimal features are not known for a given problem. Hence, in Section 4.3 different feature sets are analyzed to extract features with appropriate information for texture classification. For signal classification purposes, the tree decomposition should be based on some signal-discrimination measure other than the energy of the signal. A separability-based tree decomposition algorithm developed by Chellappa *et.al* [43], [44] is presented in Section 4.4. This algorithm uses class-separability as a criterion to decompose the signals. In each case, the parameters of the feature extraction process are empirically optimized for the texture classification problem, based on the subject database acquired for this work. Correspondingly, the efficacy of the dimensionality reduction strategies is determined for each feature set. These feature extraction and dimensionality reduction strategies are prescribed in the context of a Euclidean distance classifier. The features are based upon the wavelet packet transform that is presented in Section 4.5. Even though we know that criteria based wavelet tree decompositions are better for any given problem, the analysis is performed with the octave tree also for better understanding and performance comparison. The octave tree

based features and the corresponding results are discussed in Section 4.6. The relative performance of each feature set for all the tree structures is compared in Section 4.7 to give a picture of the best overall signal representation for texture classification. One can expect separability based tree decomposition methods yield better performance for texture classification. However, the complexity involved is high. A simple, new methodology developed for texture classification is presented in Section 4.8. Performance is summarized in Section 4.9.

## 4.1. NATURAL TEXTURE IMAGES

Textures provide important characteristics for object identification from biomedical images, satellite photographs and many other images. Their analysis is fundamental to many applications such as remote sensing, medical diagnosis etc. A large class of natural textures can be modeled as a quasi-periodic pattern and detected by highly concentrated spatial frequencies and orientations.

Study of the human visual system indicates that spatial or frequency representation, which preserves both global and local information, is adequate for quasi-periodic signals. This observation has motivated researchers to develop multiresolution texture models. New algorithms such as methods with Gabor filters [45], [46], [47] and Wigner distribution have been proposed, and successful results have been reported. A spatial or frequency analysis known as wavelet theory has been applied to texture analysis in the last few years.

The wavelet and wavelet packet transform can be implemented efficiently with pyramid- and tree-structured algorithms and hence they are called pyramid and tree-structured wavelet transforms, respectively. The pyramid-structured wavelet transform decomposes a signal into a set of frequency channels that have narrower bandwidths in the lower frequency region. The transform is suitable for signals consisting of smooth components, where most of the information is concentrated in the low frequency regions. However, it may not be suitable for quasi-periodic signals whose dominant frequencies are in the middle frequency region. To analyze quasi-periodic signals, the concept of wavelet bases has been generalized to include a library of modulated waveform orthonormal bases, called wavelet packet bases. Wavelet transforms and their generalized form, called wavelet packets, provide signal analysis through smooth partitioning of the phase plane along the frequency axis.

The multiresolution-based, tree-structured wavelet transform helps one zoom into any desired frequency channels for further decomposition. It is usually unnecessary and expensive to decompose all subbands at each scale to achieve full decomposition (uniform decomposition). To avoid full decomposition, a criterion needs to be adopted to decide whether decomposition is needed for a particular sub-band. The tree-structured wavelet transform determines important channels dynamically according to a specific criterion and can be viewed as an adaptive multichannel method. Depending on the specific application, different criteria can be used to build the optimal or sub-optimal wavelet packet tree. Coifman and Wickerhauser [30] have used entropy as a measure of energy spread among the transform coefficients for maximum energy compaction. For signal compression applications Vetterli et al. [40] suggest the minimization of the rate-distortion function as a criterion for basis tree selection. Tree-structured wavelet transforms based on different criteria are presented in Sections 4.2 and 4.4. The algorithm for the energy-based tree-structured wavelet transform basically follows [41]. The textures used in the analysis in all the algorithms are Brodatz textures [48], which are obtained from "http://sipi.usc.edu/services/database/database.cgi". These textures are shown in Figure 4.1.

## 4.2. ENERGY-BASED TREE-STRUCTURED WAVELET TRANSFORM

An appropriate way to perform the wavelet transform for textures is to detect the significant frequency channels and then decompose them further. In the tree-structured transform, only certain frequency bands are decomposed, depending on the deciding criteria. The averaged $L_1$-norm is used as the energy function to locate dominant channels, which is defined as

$$e(x) = \frac{1}{N}\|X\|_1 = \frac{1}{N}\sum_{i=1}^{N}|x_i| \tag{4.1}$$

where $\underline{X} = (x_1,...,x_J)$.

Basic Algorithm:

1. Decompose a given textured image into four subimages using a 2-D wavelet transform. This can be viewed as the parent and children nodes in a tree.

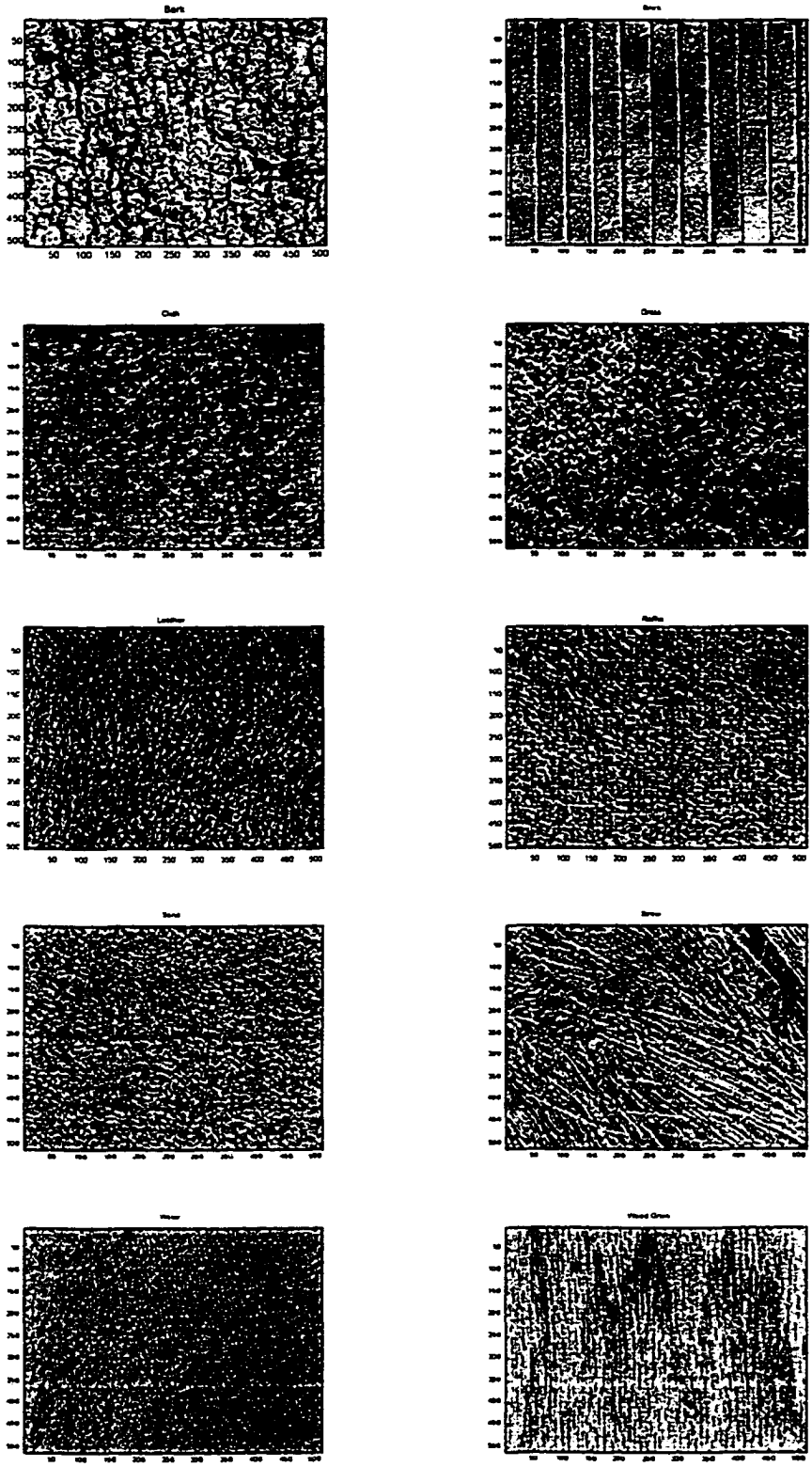


Figure 4. 1. Textures used in the analysis

2. Calculate the energy of each decomposed image using the equation (4.1). If the decomposed image is x(m,n), with $1 \leq m \leq M$ and $1 \leq n \leq N$, the energy is

$$e = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} |x(i,j)| \qquad (4.2)$$

3. If the energy in a subimage is significantly smaller than that which is present in the others, stop the decomposition in that band since it contains less information. This step can be achieved by comparing energy with the largest energy value in the same scale. That is, if $e < Ce_{max}$, stop decomposing this region, where C is a constant less than 1.
4. If the energy of a subimage is significantly larger, the above decomposition procedure is applied to the subimage.

Practically, the size of the smallest subimages should be used as a stopping criterion for decomposition. If the decomposed channel has a very small spatial size, the location and energy value of the feature may vary widely from sample to sample so that the feature may not be robust. Once the dominant channels of the textures are known, the features need to be extracted from these channels for classification.

### 4.2.1. FEATURE VECTOR

Each sample is decomposed with the tree-structured wavelet transform and the normalized energy is calculated at its leaves. Generate a representative energy map for each texture by averaging the energy maps over all the samples. Decompose the unknown texture with the tree-structured wavelet transform and construct its energy map. Pick up the first J dominant channels, which are the leaf nodes in the energy map with the largest energy values as features. Denote this feature set by $X = (x_1, ...., x_N)$. This feature set is used in classification.

### 4.2.2. CLASSIFIER

*Texture Classification Algorithm:*

A simple texture classification algorithm follows from the above algorithm using fixed number of features J.

- Learning Phase

1. Given m samples obtained from the same texture, decompose each sample with the tree-structured wavelet transform and calculate the normalized energy at its leaves.
2. Generate a representative energy map for each texture by averaging the energy maps over all m samples.
3. Repeat the process for all textures.

- Classification Phase

Decompose the unknown texture with the tree-structured wavelet transform and construct its energy map.

1. Pick up the first J dominant channels, which are the leaf nodes in the energy map with the largest energy values as features. Denote this feature set by $X = (x_1, \ldots, x_J)$.
2. For texture i in the database, pick up the energy values in the same channels and denote the energy value by $m_i = (m_{i,1}, \ldots, m_{i,J})$.
3. Calculate the discrimination function for textures in the candidate list by

$$D_i = distance(X, m_i). \tag{4.3}$$

4. Assign the unknown texture to texture i if $D_i < D_j$ for all $j \neq i$.

When the leaf node does not exist in the energy map of texture i in step 3, the texture i is discarded from the candidate list, because the energy value of the corresponding channel is too low or too high.

Several distance functions can be used in equation (4.3). Euclidean distance and simplified Mahalanobis distance are popular and are used in this study.

Euclidean distance is given by

$$D_i = \sum_{j=1}^{J} (x_j - m_{i,j})^2 \ . \qquad (4.4)$$

The Mahalanobis distance is calculated using

$$D_i = (x - m_i)^T C_i^{-1} (x - m_i) \qquad (4.5)$$

When the covariance matrix $C_i$ of the feature set is a diagonal matrix, or equivalently the features are independent of each other, the Mahalanobis distance can be reduced to the form

$$D_i = \sum_{j=1}^{J} \frac{(x_j - m_{i,j})^2}{c_{i,j}} \qquad (4.6)$$

where $c_{i,j}$ is the variance of feature j in class i. For the case when $C_i$ is not diagonal but diagonally dominant, (4.6) is still a good measure.

### 4.2.3. RESULTS

Seven textures are obtained from the Brodatz texture album [48]. Each image is of size 512 x 512 pixels with 256 gray levels. The mean of the image is removed before processing.

One hundred sample images of size 256 x 256 are chosen from the original image and used in training and classification with the leave-one-out algorithm in classification [7]. The energies of the largest 5 dominant channels of the unknown textures are used as the features and the two distance measures (4.4) and (4.6) are used for classification.

The Battle-Lemarie cubic spline wavelet basis function with 16 taps is used and the results are compared in Appendix A, in Table A-1 with the results given in reference [41] for Euclidean distance. For comparison purposes with reference [41], only seven textures are used. To see the effect of the size of window or sample size of the samples obtained from each original texture on the classification performance, three different window (w) sizes are tested using Battle-Lemarie 16 tap filter as shown in Table 4.1 and Table 4.2. The classification performance is the highest for window size 256 and poorest for w=64. This is related to the overlap area from sample to sample. For sample size equal to 256, the overlap area between samples is high compared to when the sample size is 64. The classification performance should be higher using the Mahalanobis distance compared to the Euclidean distance, as it uses the

covariance of the features along with the mean in classifying the textures. The classification performance is higher for sample sizes 128 and 64 except for the sample size 256.

**TABLE-4.1: EUCLIDEAN DISTANCE**

| Texture | Correct Classification Rate (%) | | |
|---|---|---|---|
| | w=256 | 128 | 64 |
| Brick | 94 | 87 | 69 |
| Grass | 100 | 100 | 99 |
| Sand | 85 | 90 | 72 |
| Wood Grain | 100 | 93 | 85 |
| Cloth | 96 | 90 | 89 |
| Leather | 99 | 100 | 100 |
| Raffia | 100 | 96 | 76 |
| Overall | 96.29 | 93.71 | 84.3 |

**TABLE-4.2: SIMPLIFIED MAHALANOBIS DISTANCE**

| Texture | Correct Classification Rate (%) | | |
|---|---|---|---|
| | w=256 | 128 | 64 |
| Brick | 100 | 100 | 87 |
| Grass | 100 | 100 | 100 |
| Sand | 75 | 87 | 64 |
| Wood Grain | 98 | 95 | 86 |
| Cloth | 94 | 80 | 87 |
| Leather | 100 | 100 | 100 |
| Raffia | 100 | 100 | 80 |
| Overall | 95.29 | 94.57 | 86.29 |

## 4.3. WAVELET PACKET BASED FEATURES

### 4.3.1. FEATURE EXTRACTION

Aside from the algorithm, one can argue about the appropriate choice of the feature set for each node. Without claiming optimality, as reasonable choices we have used features based on the statistics of the sub-band signals, e.g., the $L_1$-norm, F-norm and the $\Delta$-norm (maximum of the discrete wavelet transform coefficients in the subbands) of the wavelet coefficients, which are defined as [49]

$$\|X\|_1 = \max_j \sum_{i=1}^{m} |x_{ij}| \tag{4.7}$$

$$\|X\|_F = \frac{1}{mn}\left[\sum_{i=1}^{m}\sum_{j=1}^{n} |x_{ij}|^2\right]^{1/2} \tag{4.8}$$

$$\|X\|_\Delta = \max|x_{ij}| \tag{4.9}$$

The feature vector is formed by calculating the features at the nodes using the above measures. Training and classification are performed using all these measures on the data set and the results are listed in the following Sections.

### 4.3.2. $L_1$-NORM OF THE WAVELET SUB-BAND COEFFICIENTS

The $L_1$-norms[2] of the discrete wavelet transform coefficients at the nodes are calculated, which form the feature vector. The percentage of correct classification rates are computed using B-L 16-tap wavelet and 4-tap Daubechies wavelet for window sizes 256 and 64 respectively.

[2] Though the $L_1$-norm is defined differently in Section 4.2, it is defined as in equation (4.1) by the authors in reference [41]. So, it has been used by that terminology in that section only. In the rest of the sections, it has been renamed as absolute mean, and the $L_1$-norm is defined as in equation (4.7).

### 4.3.3. F-NORM OF WAVELET SUB-BAND COEFFICIENTS

The F-norm or the normalized energy of the discrete wavelet transform coefficients at the nodes is calculated. The percentage of correct classification rate is calculated using this measure for B-L 16-tap wavelet and 4-tap Daubechies wavelet. The percentage of correct classification rates are calculated for window sizes 256 and 64.

### 4.3.4. Δ-NORM OF WAVELET SUB-BAND COEFFICIENTS

The maxima of the discrete wavelet transform coefficients at the nodes form the feature vector. The performance is compared using this measure for the two wavelets. The percentage of correct classification rates are calculated for window sizes 256 and 64 respectively.

### 4.3.5. COMPARISON OF PERFORMANCE

The correct percentage of classification rates for the seven textures are listed in Table 4.3 for all the above measures in Sections 4.3.2 to 4.3.4 for a window size of 256. These results are obtained using the B-L 16 tap wavelet. The overall percentage listed in the last row shows the average performance of all the textures for the corresponding measure. These percentages are obtained using the Euclidean distance classifier. The same procedure is repeated using the above measure for a window size of 64 and results are listed in Table 4.4.

The overall percentage of correct classification rates are compared in Table 4.5 for the two wavelets, B-L 16 tap and Daubechies 4-tap for all the measures and for window sizes 64 and 256 using the Euclidean distance classifier. The performance is much better using the Daubechies wavelet for the measures and for both window sizes. The absolute-sum measure seems to perform better than other measures for the B-L wavelet and even for the Daubechies wavelet for window size 64.

**TABLE-4.3: EUCLIDEAN DISTANCE, WINDOW SIZE = 256**

| Texture | Correct Classification Rate (%) | | | |
|---|---|---|---|---|
| | Abs. mean | $L_1$-norm | F-norm | Δ-norm |
| Brick | 94 | 100 | 78 | 44 |
| Grass | 100 | 100 | 100 | 100 |
| Sand | 85 | 70 | 74 | 61 |
| Wood Grain | 100 | 100 | 90 | 85 |
| Cloth | 96 | 85 | 89 | 79 |
| Leather | 99 | 95 | 100 | 75 |
| Raffia | 100 | 100 | 99 | 69 |
| Overall | 96.29 | 92.86 | 90 | 73.29 |

**TABLE-4.4: EUCLIDEAN DISTANCE, WINDOW SIZE = 64**

| Texture | Correct Classification Rate (%) | | | |
|---|---|---|---|---|
| | Abs. mean | $L_1$-norm | F-norm | Δ-norm |
| Brick | 69 | 72 | 72 | 72 |
| Grass | 99 | 98 | 99 | 99 |
| Sand | 72 | 62 | 65 | 60 |
| Wood Grain | 85 | 51 | 88 | 89 |
| Cloth | 89 | 85 | 87 | 84 |
| Leather | 100 | 100 | 89 | 89 |
| Raffia | 76 | 57 | 61 | 53 |
| Overall | 84.3 | 75 | 80.14 | 78 |

**TABLE-4.5: OVERALL PERCENTAGE OF CORRECT CLASSIFICATION USING EUCLIDEAN DISTANCE**

| Wavelet | Window Size | Abs. mean | $L_1$-norm | F-norm | Δ-norm |
|---|---|---|---|---|---|
| B-L | 64 | 84.30 | 75.00 | 80.14 | 78.00 |
| | 256 | 96.29 | 92.86 | 90.00 | 73.29 |
| Daubechies | 64 | 90.86 | 84.00 | 87.14 | 81.43 |
| | 256 | 96.85 | 98.14 | 97.57 | 88.29 |

## 4.4. SEPARABILITY-BASED TREE-STRUCTURED WAVELET TRANSFORM

The algorithm for the separability-based tree-structured wavelet transform basically follows [43]. In Section 4.2, dominance of the energy concentration in a sub-band is used as a criterion for further decomposition. However, for classification purposes, a criterion based on the difference between patterns or signals of different classes, i.e. class separability, is preferable. One may observe relatively high energy subbands in which the desired signals are quite similar and subbands of relatively low average energies that contain significant information about the difference between the signals. The following algorithm is developed by selecting the tree basis depending on the class separability or discrimination. The next section gives a brief definition of class separability.

### 4.4.1. CLASS SEPARABILITY

To design an efficient classification system, one has to select features that are most effective in capturing the salient differences between signals so that the signal clusters are well separated in the feature space.

A simple way of formulating a criterion for class-separability is based on within- and between-class scatter matrices [7]. The within-class scatter matrix shows the scatter of sample vectors (V) of different classes around their respective mean vectors M.

$$S_w = \sum_{i=1}^{L} \Pr\{C = C_i\} \Sigma_i$$

where

$$\Sigma_i = E[(V - M_i)(V - M_i)^T | C_i] \qquad (4.10)$$

represents the spread of feature vectors in the $i^{th}$ class. In addition, one can define the between-class scatter matrix as the scatter of the conditional mean vectors $M_i$ around the overall mean vector M

$$S_b = \sum_{i=1}^{L} \Pr\{C = C_i\}(M - M_i)(M - M_i)^T \quad (4.11)$$

In order to have good separability for classification, one needs to have "large" between-class scatter and "small" within-class scatter simultaneously. The cost function measuring the combined separability is

$$J = Tr(S_w^{-1} S_b) \quad (4.12)$$

Using J defined in equation (4.12) as the class separability criterion, the algorithm for basis selection can be summarized as follows. The basis selection algorithm is based on class separability rather than energy or entropy. At each level, accumulated tree-structured class separabilities obtained from the tree that includes a parent node and the one that includes its children are compared. Decomposition of the sub-band is performed if it provides larger combined separability. The algorithm follows in the next section.

### 4.4.2. BASIS SELECTION ALGORITHM

Select an appropriate wavelet or QMF filter.

1. Perform one level of decomposition on each terminal node.
2. For each parent node and children nodes, compute the feature sets.
3. Calculate the combined class separability (CCS) using all previously selected tree nodes with the current (parent) node. Divide the parent node into children nodes. Calculate CCS using all previously selected tree nodes excluding the parent node but including all its children nodes. Compare the obtained CCS's with the parent node and with the children nodes (with out parent). Retain the tree that provides better separability. Thus, we retain the parent if it provided better CCS than its children; otherwise, we keep the children.
4. Repeat steps 2-4 for the updated tree until no further significant improvement of separation is observed by decomposing the terminal nodes. The iteration can be terminated earlier if the amount of achieved separation is larger than a pre-selected threshold.

### 4.4.2. FEATURE EXTRACTION

Once all the texture samples are decomposed based their combined class separability, features need to be extracted from the subbands of each sample for classification. The following features are used to compare the performance with the results reported in reference [43]. The features based on second order and third order central moments (n = 2 and 3) of the corresponding sub-band signals are calculated using

$$\mu_n(W) = \frac{1}{|W|}\left( \sum_{x \in W} \left( f(x) - \overline{f_W} \right)^n \right)^{1/n}$$

$$V = \left\{ v_i = \mu_2(W_i), v_i' = \mu_2(W_i) / \mu_3(W_i); i = 0,1,...N_{subbands} \right\} \quad (4.\ 13)$$

where V denotes the feature vector with the features calculated from the chosen N subbands, depending on their CCS, $W_i$ is the local window on the $i^{th}$ sub-band, and $\mu_2$ and $\mu_3$ are the second and third order central moments. On each sub-band, f(x) and $\overline{f_W}$ are defined as the intensity value at the location x and the average intensity on window W centered at x respectively. For each sub-band, $\mu_2$ shows the average energy, which is also called the F-norm in previous sections. The ratio $\mu_3/\mu_2$ roughly represents the information about the shape of the spectrum in that sub-band.

### 4.4.3. DIMENSIONALITY REDUCTION

The feature reduction approach attempts to reduce the number of features by selecting the best subset of the original feature set according to some criterion.

Feature selection necessarily consists of two parts:

1. A criterion must be established by which it is possible to judge whether one subset of the features is better than another.
2. A systematic procedure must be found for searching through the candidate subsets.

Ideally, the selection criterion should be the probability of mis-classification [7]. In practice, evaluation of this criterion is generally too computationally complex, and we must resort to simpler criteria such as those based upon class separability. Similarly, in an ideal situation, the search procedure should consist of an exhaustive search over all possible subsets. Exhaustive methods are often impractical due to

computational complexity, and non-exhaustive searches and suboptimal searches are often used in practice [7].

A simple way of formulating the class-separability criterion is based both on the within- and between-class scatter matrix. The within-class and between-class scatter matrices are as defined in equation 4.10 and 4.11. The combined class separability as defined in equation 4.12 is used as the criterion for dimensionality reduction.

Class separability measures are typically invariant under any nonsingular, linear or nonlinear transformation. However, any singular mapping used for dimensionality reduction results in the loss of some discrimination information. The objective is to find a mapping that, for a given reduction in space dimensionality, provides maximum separability [50]. Consequently, we are seeking a linear transformation A , which is non-invertible from $\mathbf{R}^n$ to $\mathbf{R}^m$ with m<n such that

$$A: X \subset R^n \rightarrow Y \subset R^m; A = \arg\min_{A_o} \{J_X - J_{A_o^T X}\} \tag{4. 14}$$

where $J_X = tr(S^X)$ and $J_Y = tr(S^Y)$ are separabilities computed over both X and $Y=A^TX$. Thus, A optimizes $J_Y$, i.e., it minimizes the drop in the cost $J_X - J_{A^T X}$ incurred by the reduction in the feature space dimensionality. It can be shown that [7] for such an optimum A

$$\{\lambda_i^Y\} \subset \{\lambda_j^X\} \qquad i=1,\ldots,m, \qquad j=1,\ldots,n \tag{4. 15}$$

where $\lambda^X$ and $\lambda^Y$ are the eigenvalues of the corresponding separation matrices $S^X$ and $S^Y$. The observation and the fact that

$$J_Y = tr(S^Y) = \sum_{i=1}^{m} \lambda_i^y \tag{4. 16}$$

suggest that one can maximize $J_Y$ by taking the largest m eigenvalues of $S^X$. Thus the corresponding eigenvectors form the transformation matrix A. The optimal linear transformation from $\mathbf{R}^n$ to $\mathbf{R}^m$ based on our selected separation measure results from projecting the feature vectors X onto m eigenvectors corresponding to the m largest eigenvalues of the separation matrix $S^X$.

### 4.4.4. CLASSIFIER

In some applications, even with the best separation achieved using the above method, clusters have some overlap. This is sometimes due to the inherent similarity between signals. In such cases, neural network classifiers are found to be efficient tools for representing the inherent uncertainties and similarities.

In classification, a simple multilayer feed-forward neural network is used as a classifier. At each neuron, a weighted sum of incoming activation levels plus a bias is passed through a sigmoidal non-linearity.

$$Y = f(W_2 f(W_1 X + \Theta_1) + \Theta_2)$$

where

$$f(x) = \frac{1}{1 + \exp(-x)}. \tag{4. 17}$$

The $W_1$ and $W_2$ are connection weight matrices, and $\Theta_1$ and $\Theta_2$ are the bias vectors corresponding to the hidden and output layers, respectively. The adaptive nonlinear mapping characteristics of neural networks are utilized to create a set of fuzzy hyperplanes in the feature space which tries to separate clusters. By combining outputs of hidden nodes, which are primary membership tests, the output nodes form the fuzzy decision boundaries. In other words, the desired decision boundaries or nonlinear membership functions are formed in the process of training (or adaptive adjustment of connection weights as well as bias vectors) of a neural network architecture. Supervised learning based on differences between an actual and a target output value for all classes is formed using a nonlinear optimization scheme that minimizes the total mean squared error.

In addition to the neural network classifier, a Euclidean distance classifier is also used to classify the textures, both for its simplicity and for comparison. The classification performance needs to be compared for separability-based tree decomposition and energy-based tree decomposition. Hence, the Euclidean distance classifier is used to classify the textures using the features extracted from the subbands with higher CCS. Also, the performance needs to be compared between the Euclidean distance classifier and Neural network classifier. If the performance improvement achieved with the neural network is not significant, then the Euclidean distance classifier would be preferred because of its lower computational complexity.

### 4.4.5. RESULTS

Using the above algorithm, textures are decomposed using two, three and four levels. A Daubechies four-tap filter is used to perform the decompositions. Features are computed using the second and third central moments of the image subbands and the dimensionality is reduced as described above. The four most important features are obtained. Using these features, a simple feed-forward neural network is designed [11]. Our database consists of ten Brodatz [48] textures of size 512x512. Each texture sample of size 64x64 is obtained from the larger texture image. Each of the training and classification sets has 100 samples. A four-input neural network with eight hidden and ten output neurons is used for the classification.

The results are provided in Table-A.2 in Appendix A and compared to the results reported in reference [43]. The textures used in the database are not the same as in the reference [43]. Also, the wavelet and the depth of the decomposition used in obtaining the results listed in [43] are unknown. Thus, exact reformulation of the experiment for direct comparison is not possible.

## 4.5. WAVELET PACKET BASED FEATURES

### 4.5.1. FEATURE EXTRACTION

Aside from the algorithm, one can make arguments concerning the appropriate choice of the feature set for each node. Without claiming optimality, as reasonable choices we have used features based on the statistics of the sub-band signals. Features are extracted based on the statistics of the sub-band signals, e.g. the $L_1$-norm and the $\Delta$-norm (maximum of the discrete wavelet transform coefficients in the subbands) of the wavelet coefficients that are defined in Section 4.3.1. The feature vector is formed by calculating the features at the nodes using the above measures. The feature vectors are dimensionally reduced. Training and classification is performed using all these measures on the data set and the results are listed in the following sections.

### 4.5.2. $L_1$-NORM OF THE WAVELET SUB-BAND COEFFICIENTS

The $L_1$-norms of the discrete wavelet transform coefficients at the nodes are calculated and used to form the feature vector. The classification rate percentage is computed using the 4-tap Daubechies wavelet to do the decomposition. The overall classification rate percentage is calculated for window size 64 using the Euclidean distance classifier described in Section 2.3.3.

### 4.5.3. Δ-NORM OF WAVELET SUB-BAND COEFFICIENTS

The maxima of the discrete wavelet transform coefficients at the nodes form the feature vector. The classification percentage is calculated using this measure for the Daubechies 4-tap wavelet.

### 4.5.4. ABSOLUTE SUM OF WAVELET SUB-BAND COEFFICIENTS

The absolute sum of the discrete wavelet transform coefficients is defined as

$$e(x) = \frac{1}{N}\|X\|_1 = \frac{1}{N}\sum_{i=1}^{N}|x_i| \tag{4.18}$$

where $X = (x_1, \ldots, x_N)$ is the coefficient vector of N DWT coefficients of each sub-band. The absolute sums at the nodes form the feature vector. The performance is measured using this measure for the Daubechies wavelet. The overall classification rate percentage is calculated for window size 64 using the Euclidean distance classifier.

### 4.5.5. COMPARISON OF PERFORMANCE

The classification rate performances of the textures are listed in Table 4.6 using $\mu_2$ (4 features) and $\mu_2$ and $\mu_3/\mu_2$ (8 features) for a window size of 64. These results are obtained using the Daubechies 4-tap wavelet. These percentages are compared using both the Euclidean distance classifier and the neural-net classifier. Extracting more features using both $\mu_2$ and $\mu_3/\mu_2$ should increase the classification percentage. The classification rate improved by extracting 8 features compared to extracting 4 features for level 2 only. But the classification rate did not improve by increasing the depth of the sub-band tree. Also, it did not improve the classification performance when compared to the 4 feature case for levels 3 and 4. Also, using

the neural network the classification performance should be better compared to using Euclidean distance classifier, as it is trained adaptively for the given data with known inputs and outputs. Using the neural network the results improved insignificantly only in the case of level 3. In all other cases the Euclidean distance classifier performed better than the neural network.

The overall classification rates are compared for the Daubechies 4-tap wavelet for all the measures using the Euclidean distance classifier in Table 4.7. The results are compared for different levels of the tree, i.e., for tree decompositions of level 2, 3, and 4. The absolute sum measure gave a better performance for levels 2 and 3, whereas the $L_1$-norm extracted better features for the level 4 decomposition that resulted in improved performance. As expected, the classification performance improved with the increase in depth of the tree except for the case of extracting 8 features using $\mu_2$ and $\mu_3/\mu_2$.

**TABLE-4.6: OVERALL CLASSIFICATION FOR ALL THE FEATURES USING EUCLIDEAN DISTANCE AND NEURAL NET CLASSIFIERS**

| Level of Decomposition | Feature | Classification (%) | |
|---|---|---|---|
| | | Euclidean Distance | Neural Net |
| 2 | $\mu_2$ | 97.3 | 96.8 |
| 3 | $\mu_2$ | 97.3 | 97.4 |
| 4 | $\mu_2$ | 97.8 | 94.9 |
| 2 | $\mu_2$ and $\mu_3/\mu_2$ | 97.8 | 95.8 |
| 3 | $\mu_2$ and $\mu_3/\mu_2$ | 97.2 | 97.2 |
| 4 | $\mu_2$ and $\mu_3/\mu_2$ | 97.1 | 95.3 |

**TABLE-4.7: OVERALL CLASSIFICATION FOR ALL THE FEATURES USING EUCLIDEAN DISTANCE**

| Level of Decomposition | Feature | Classification (%) |
|---|---|---|
| 2 | $L_1$-norm | 92.9 |
| 3 | $L_1$-norm | 96.9 |
| 4 | $L_1$-norm | 98.3 |
| 2 | $\Delta$-norm | 89.3 |
| 3 | $\Delta$-norm | 96.5 |
| 4 | $\Delta$-norm | 97.5 |
| 2 | abs. sum | 97.5 |
| 3 | abs. sum | 98.4 |
| 4 | abs. sum | 98.3 |

## 4.6. WAVELET TRANSFORM

In this section, the textures are classified using the regular wavelet transform, which has an octave tree. This is done to compare the performance between the wavelet packet transform and the wavelet transform for texture classification. These results strongly suggest that there is a considerable performance gain that can be achieved by using criterion-based wavelet packet decomposition, especially the separability based decomposition for texture classification as opposed to an octave tree wavelet transform. Table 4.8 compares the classification performance for the Daubechies 4-tap wavelet with the second level of decomposition using the Euclidean distance classifier both with and without dimensionality reduction. The classification performance is considerably improved using the dimensionality reduction technique for the octave tree structure.

## 4.7. COMPARISON AMONG THE WAVELET TREE BASES

In this section the performance of the three tree bases discussed so far are compared using the Daubechies 4-tap wavelet. The results are obtained using the F-norm for the second level decomposition and are classified using the Euclidean distance classifier. The results are listed in Table 4.9. The separability-based tree decomposition gives the best classification performance. However, the classification performance is comparable among the tree decompositions when dimensionally reduced features are used for classification. Once again, these results strongly suggest that the criterion-based wavelet packet decomposition is superior to the octave tree wavelet transform for purposes of texture classification.

TABLE-4.8: OVERALL CLASSIFICATION PERFORMANCE USING F-NORM

| Level | No. of Features | Dimensionality Reduction | Daubechies |
|---|---|---|---|
| 2 | 7 | No | 83.50 |
| 2 | 7 | Yes | 95.20 |

TABLE-4.9: OVERALL CLASSIFICATION PERFORMANCE USING F-NORM FOR TREE BASES

| *Tree* | *Feature Reduction* | *Daubechies* |
|---|---|---|
| Energy-Based Tree | Feature selection | 87.14 |
| Separability-Based tree | Feature Selection | 83.50 |
| | Using all features | 87.30 |
| | Dimensionality Reduction | 97.30 |
| Octave Tree | Using all features | 83.50 |
| | Dimensionality Reduction | 95.20 |

## 4.8. A SIMPLE ALGORITHM FOR TEXTURE CLASSIFICATION

The classification performance was compared in the previous sections for different measures using different tree structures. Two algorithms based on the tree-structured wavelet transform for classification were reviewed and analyzed. The classification performance is impressive. In this section, a simple algorithm is developed for texture classification that has significantly lower computational complexity compared to the previously reviewed algorithms. We develop our algorithm by addressing the three main issues of multichannel texture classification: 1) Feature extraction within each channel 2) Channel selection and 3) Feature combination among channels.

For signal analysis and classification problems, energy concentration in a sub-band has been used as a criterion for its further decomposition [41], i.e., at each level, subbands of maximum energy are identified and decomposed further. In these schemes, the energy map of the surviving subbands is used as the feature set.

However, for classification purposes, a criterion based on the difference between signals of different classes is preferable. The desired signals may be quite similar in the high-energy bands and may contain significant information in the low energy bands. Instead of decomposing the subbands further based on some criterion, e.g. energy concentration, only one level of the wavelet transform decomposition is used and features are obtained from these subbands. The features in these subbands give significant information for texture classification. Many features may be required from these subbands to achieve similar performance to the tree-structured decomposition methods in classification. On the other hand, the average energy of the subbands may not be the best feature set for classification.

This section investigates the effectiveness of a separability-based measure for suitable wavelet basis selection. The chosen wavelet may not be optimal for representing or approximating individual signals and may not provide good performance for some other tasks, e.g. compression, identification and modeling.

### 4.8.1. FEATURE EXTRACTION

The measures used for comparison in this work are the $L_1$-norm, F-norm and the $\Delta$-norm of the wavelet coefficients, which are defined in Section 4.3.1.

The features are calculated from the four bands, low-low (LL), low-high (LH), high-low (HL) and high-high (HH) using these measures. A feature is extracted from a sub-band by calculating the norm of the coefficients in that sub-band. Specifically, four features are extracted from the four bands, one from each sub-band. The effect of increasing the number of features on the classification performance can be compared by extracting 16 features. This is done by dividing each sub-band into four equal regions (not subbands) from which one feature is obtained. These 16 features from 16 regions (4 features from each sub-band) give significant information for texture classification. Hence, one can achieve similar or better performance using more features instead of using the tree-structured decomposition based on some criteria, which will be demonstrated in section 4.8.2.

### 4.8.2. EXPERIMENTAL RESULTS

Seven textures are obtained from the Brodatz texture album [48]. Each image is of size 512 x 512 pixels with 256 gray levels. The mean of the image is removed before processing.

One hundred sample images of size 256 x 256 are chosen from the original image and used in training and classification. Classification is performed using the leave-one-out algorithm [7]. In the training phase, the feature vector is calculated for all 100 samples. The mean of these 100 feature vectors is calculated to represent this texture class. The average feature vector is calculated for all the texture classes in the database. In the classification phase, the feature vector is calculated for the test texture and the distances between the test feature vector and the feature vectors of the known texture classes are calculated. The test texture is classified as a particular class when the Euclidean distance between the test feature vector and the feature vector of that class is less than the distance between the test feature vector and the feature vectors of the remaining texture classes. The Battle-Lemarie cubic spline wavelet basis function with 16 taps is used to calculate the discrete wavelet transform coefficients. This particular wavelet is chosen to compare the results with the performance using the Energy-based tree decomposition.

The classification performance is compared among the measures using 4 and 16 features. These results are also compared in Table 4.10 with the results using the tree-structured decomposition from [41], which uses five features. One can observe that the performances are comparable using only the four features and with much less computational complexity. The classification performance is compared using 16 features

with window size 256 for all the features in Table 4.11. The results are compared in tables 4.12 and 4.13 when the texture sample size is 64 with 4 and 16 features for all the measures.

It can be observed that performance increases with increasing numbers of features or with increasing sample size. The result implies that the classification rate is improved by incorporating the statistical information of texture features.

**TABLE-4.10: CLASSIFICATION PERFORMANCE FOR WINDOW SIZE = 256**

| Texture | Correct Classification Rate (%) | | | |
|---|---|---|---|---|
| | $L_1$-norm | F-norm | Δ-norm | Results From [2] |
| No.of features | 4 | 4 | 4 | 5 |
| Brick | 100 | 94 | 60 | 98 |
| Grass | 100 | 100 | 100 | 96 |
| Sand | 95 | 90 | 98 | 92 |
| Wood Grain | 92 | 90 | 98 | 97 |
| Cloth | 93 | 90 | 100 | 100 |
| Leather | 100 | 100 | 100 | 100 |
| Raffia | 100 | 90 | 90 | 100 |
| Overall | 97.14 | 93.43 | 92.29 | 97.57 |

**TABLE-4.11: CLASSIFICATION PERFORMANCE FOR NUMBER OF FEATURES = 16, WINDOW SIZE = 256**

| Texture | Correct Classification Rate (%) | | |
|---|---|---|---|
| | $L_1$-norm | F -norm | Δ-norm |
| Brick | 100 | 93 | 92 |
| Grass | 100 | 100 | 100 |
| Sand | 99 | 90 | 100 |
| Wood Grain | 100 | 93 | 100 |
| Cloth | 94 | 90 | 100 |
| Leather | 100 | 100 | 100 |
| Raffia | 100 | 91 | 100 |
| Overall | 99 | 93.86 | 98.86 |

TABLE-4.12: CLASSIFICATION PERFORMANCE FOR NUMBER OF FEATURES = 4, WINDOW SIZE = 64

| Texture | Correct Classification Rate (%) | | |
|---|---|---|---|
| | $L_1$-norm | F-norm | Δ-norm |
| Brick | 76 | 74 | 75 |
| Grass | 90 | 93 | 96 |
| Sand | 36 | 54 | 74 |
| Wood Grain | 54 | 79 | 90 |
| Cloth | 60 | 81 | 73 |
| Leather | 94 | 100 | 99 |
| Raffia | 67 | 86 | 88 |
| Overall | 68.14 | 81.00 | 85.00 |

TABLE-4.13: CLASSIFICATION PERFORMANCE FOR NUMBER OF FEATURES = 16, WINDOW SIZE = 64

| Texture | Correct Classification Rate (%) | | |
|---|---|---|---|
| | $L_1$-norm | F-norm | Δ-norm |
| Brick | 75 | 79 | 74 |
| Grass | 93 | 94 | 97 |
| Sand | 38 | 59 | 84 |
| Wood Grain | 66 | 83 | 98 |
| Cloth | 65 | 82 | 85 |
| Leather | 100 | 100 | 100 |
| Raffia | 79 | 89 | 90 |
| Overall | 73.71 | 83.00 | 89.71 |

## 4.9. PERFORMANCE SUMMARY

The previous sections have described the application of wavelet packet transform and wavelet transform based feature sets to the task of texture classification. Different parameters and their influence on the texture classification are studied. The parameters used in this study are the sample size of each texture sample obtained from the original texture, the feature extracting measures, the number of features extracted and the feature reduction methods. All of these parameters have considerable influence on the classification performance. Each measure may extract features, which may result in better classification performance for a particular combination of the rest of the features. Overall, the F-norm seems to perform better for all cases. The classification rate decreases with decrease in sample size since it has lower overlapped information content among the texture samples. This can be improved by extracting features that give better information or by increasing the depth of the decomposition. When using time-frequency based features, dimensionality reduction improves the performance compared to the feature selection methods.

# CHAPTER 5

# INFLUENCE OF WAVELET BASES AND WAVELET-BASED PARAMETERS ON TEXTURE CLASSIFICATION

## 5.1. OPTIMAL FILTERS FOR CLASSIFICATION

The fundamental purpose of feature extraction for classification is to emphasize the important information in the measured signal and to de-emphasize that which is irrelevant. This implies transforming the raw data into a domain that presents the information contained in the signal more clearly, i.e. mapping the raw data to concentrate and localize information. Time-frequency methods offer the ability to localize the energy distribution of a signal in both time and frequency (space). The localization depends on the method chosen.

Our preliminary work has shown that feature extraction is crucial for correctly classifying a signal. To obtain features that have significant information for classification, the wavelet basis used to transform the raw data into the "wavelet" or "time-frequency" domain is important. We know that wavelet functions can be used for function approximation and finite energy signal representations that are useful in signal processing and system identification. The wavelet basis is generated by dilating and shifting a single mother wavelet function. The wavelet design is not unique and its design is related to that of a symmetric FIR low pass filter. An appropriate selection of the wavelet for signal classification can result in maximal benefits in classification. Recently, the application of wavelets and multirate filterbanks to multiscale feature extraction [51], [52], [53] has received significant attention. Wavelet based features have been shown to be efficient representations for compression, detection, classification and segmentation of signals and images [54]-[57]. Examples of texture analysis and classification using wavelets and wavelet packets are given in [58]-[61]. Different wavelets may be suitable for classifying different sets of signals or images. So we need to find an optimal mother wavelet function to span the appropriate feature space for signal classification. Before the optimal wavelet basis is designed based on some as yet to be determined criterion,

we need to study the effect of the mother wavelet function on the classification of textures. This chapter focuses on studying the influence of varying the wavelet bases on classification performance.

Let us start with the study of the computationally simple algorithm developed in Section 4.8. In Section 5.2 we study the effect of the mother wavelet function on the classification performance using this simple algorithm. Also, the other parameters that have influence on the classification are studied along with varying the mother wavelet. The influence of wavelet bases on all the types of tree decomposition algorithms is studied. This helps us understand if a particular tree decomposition algorithm gives better performance when a suitable wavelet is used. All the algorithms analyzed in the previous chapter are analyzed again in this chapter with respect to the mother wavelet, where as in the previous chapter the mother wavelet was fixed and other parameters involved in a general classification task were studied. Section 5.3 examines the energy-based wavelet packet tree algorithm and its performance for various wavelet bases. The separability-based wavelet packet tree is studied using various wavelets in Section 5.4. Though we know that separability-based tree decomposition is better for signal classification tasks, it is not known what the best tree structure is for classification when a suitable mother wavelet is used for tree decomposition. Hence, the textures are classified using the octave tree decomposition using various wavelets. This is analyzed in Section 5.5. When an appropriate wavelet is used for decomposition, the features extracted may have more relevant information for classification. The effect of dimensionality reduction on classification performance is studied in Section 5.6.1. The effect of decomposing the wavelet tree into more depth is studied in Section 5.6.2. We have used varying database sizes (7 and 10 textures) to study the influence of increasing the database sizes on their classification performance using the same settings. This is shown in Section 5.6.3. The performance is compared and summarized for various parameters in Section 5.7. Our conclusion follows in Section 5.8.

## 5.2. WAVELET BASIS CHOICE

We are interested in finding the most suitable wavelet for texture classification and applying the identified wavelet to improve the classification performance of the textures. To achieve this, the wavelet has to be designed based on the properties of the data, i.e., the choice of the mother wavelet should be dependent on the properties of the data set.

To test the effect of using different wavelet bases on the classification performance, different 4-tap wavelets with different transition band characters are chosen satisfying the constraints of the Quadrature-Mirror Filters (QMF). These constraints are explained in detail in Chapter 3. The magnitude responses of these filters are shown in Figure 5.1 and the filter coefficients are listed in Table 5.1. This set of wavelets includes the Daubechies 4-tap wavelet. In all our plots, the last case of h, i.e. h=24, is the Daubechies 4-tap wavelet. These wavelets are used instead of the B-L 16-tap wavelet used in the previous analysis and the procedure is repeated for all of the measures using 4 and 16 features. Figures 5.2 to 5.7 are plotted for measures $L_1$-norm, F-norm, and $\Delta$-norm, for sample size 256 and for number of features equal to 4 and 16. In some of the figures through out the dissertation colors are used for better clarity and understanding. Figure 5.2 shows the classification rate of all textures for various wavelets. Four features are extracted using $L_1$-norm from 4 subbands with one-level wavelet decomposition. The classification performance varied considerably for the two textures cloth and sand across the wavelets, but is constant for the rest of the textures in the database. The performance is the highest for all the wavelets for these eight textures. There is no improvement left that can be expected. This performance may be due to high sample size used from the original texture. All the samples belonging to a class have large overlapping regions, which aids in classification, as they are tightly clustered. Similar performance can be observed using F-norm in Figure 5.3. However, the $\Delta$-norm in Figure 5.4 could not classify well, even with so much overlapped information. Figures 5.5 to 5.7 show the plots with higher numbers of features extracted. That is, 16 features are extracted from 4 subbands and are used for classification. The performance improved significantly for both the best wavelet and Daubechies wavelet. Similar performance is observed when the sample size is 64. These results are not reported here, as they have similar performance.

The Figures 5.8-5.10 show the average classification of all the textures in the database for various wavelets using $L_1$-, F- and $\Delta$-norms. The performance is compared for window sizes 64 and 256 and for 4 and 16 features. The performance variation for the different wavelet bases is considerable in some cases. Figure 5.8 shows the overall classification rate using $L_1$-norm for both the sample sizes 64 and 256 with 4 and 16 features. The performance is the best for the sample size 256 with 16 features, as it has more overlapped information among the samples and also a greater number of features, which helped extract

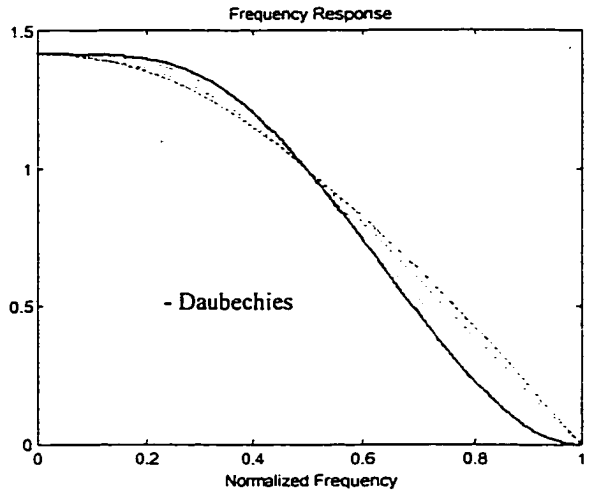



Figure 5. 1. Magnitude response of some of the filters used in the analysis.

**TABLE-5.1: FILTER COEFFICIENTS USED IN THE ANALYSIS**

| | | | |
|---|---|---|---|
| 7.8100000e-002 | 7.6910000e-001 | 6.3110000e-001 | -6.4019000e-002 |
| 1.1210000e-001 | 7.9210000e-001 | 5.9410000e-001 | -8.4002000e-002 |
| 1.4010000e-001 | 7.9210000e-001 | 5.8510000e-001 | -1.0299300e-001 |
| 1.6910000e-001 | 8.0610000e-001 | 5.5510000e-001 | -1.1606900e-001 |
| 1.9210000e-001 | 8.4610000e-001 | 4.8510000e-001 | -1.0904300e-001 |
| 2.2410000e-001 | 8.2610000e-001 | 4.9910000e-001 | -1.3504500e-001 |
| 2.5110000e-001 | 8.4410000e-001 | 4.5410000e-001 | -1.3504500e-001 |
| 2.6010000e-001 | 8.5910000e-001 | 4.2210000e-001 | -1.2699100e-001 |
| 2.8410000e-001 | 8.3210000e-001 | 4.5110000e-001 | -1.5298900e-001 |
| 3.1510000e-001 | 8.5310000e-001 | 3.9010000e-001 | -1.4406400e-001 |
| 3.3310000e-001 | 8.5310000e-001 | 3.7410000e-001 | -1.4599300e-001 |
| 3.7210000e-001 | 8.3910000e-001 | 3.6310000e-001 | -1.6003500e-001 |
| 3.9210000e-001 | 8.6310000e-001 | 2.9010000e-001 | -1.3098800e-001 |
| 4.2110000e-001 | 8.4410000e-001 | 2.9710000e-001 | -1.4799200e-001 |
| 4.4710000e-001 | 8.4510000e-001 | 2.5910000e-001 | -1.3702100e-001 |
| 4.5910000e-001 | 8.3110000e-001 | 2.7510000e-001 | -1.5106300e-001 |
| 4.8300000e-001 | 8.3650000e-001 | 2.2410000e-001 | -1.2940000e-001 |
| 4.8310000e-001 | 8.4210000e-001 | 2.0810000e-001 | -1.1906500e-001 |
| 5.1310000e-001 | 8.3310000e-001 | 1.7610000e-001 | -1.0798900e-001 |
| 5.3910000e-001 | 8.1410000e-001 | 1.8010000e-001 | -1.1906500e-001 |
| 5.6710000e-001 | 7.9910000e-001 | 1.6310000e-001 | -1.1504400e-001 |
| 5.9010000e-001 | 7.9810000e-001 | 9.8100000e-002 | -7.2075000e-002 |
| 6.2310000e-001 | 7.7110000e-001 | 1.0210000e-001 | -8.2017000e-002 |
| 6.8210000e-001 | 7.3010000e-001 | 3.0100000e-002 | -2.8064000e-002 |

more useful information for classification. The classification is poorest for sample size equal to 64 when only 4 features are extracted from the subbands. Clearly, the features did not contain enough information for classification in this case. As one can observe, the best wavelet was the first wavelet in Figure 5.8 for all the four curves. The same can be observed from the Figures 5.9 and 5.10 that the performance is best for the first case of filter on the x-axis compared to the others in almost all of the observed cases. This wavelet has low-pass filter coefficients [0.0781 0.7691 0.6311 -0.0640]. Figure 5.11 shows the frequency response of this filter and the Daubechies 4-tap filter. Table 5.2 compares the minimum, maximum and change of overall percentage of classification for the different wavelet bases for all measures, different window sizes (w=256, 64) and different number of features (4 and 16). The results are also compared to the performance using the Daubechies 4-tap wavelet. The classification performance is improved using various wavelets compared to the Daubechies wavelet even with the simple algorithm after one level of decomposition. In the next section the same experiment is conducted on the energy-based tree decomposition algorithm to study the influence of the mother wavelet on the classification.

TABLE-5.2: COMPARISON OF OVERALL CLASSIFICATION PERFORMANCE USING A SIMPLE ALGORITHM

| | Min. % | Max. % | Variation % | Daubechies % |
|---|---|---|---|---|
| **$L_1$-norm of Dwt coeff.** | | | | |
| W=64, no=4 | 77.86 | 85.57 | 7.71 | 81.29 |
| W=64, no=16 | 88.29 | 91.57 | 3.29 | 90.14 |
| W=256,no=4 | 95.71 | 99.86 | 4.14 | 98.00 |
| W=256, no=16 | 98.14 | 100 | 1.86 | 98.86 |
| **F-norm of Dwt coeff.** | | | | |
| W=64, no=4 | 86.00 | 90.43 | 4.43 | 86.29 |
| W=64, no=16 | 88.57 | 93.29 | 4.71 | 89.00 |
| W=256,no=4 | 97.00 | 98.00 | 1.00 | 97.14 |
| W=256, no=16 | 97.14 | 100.00 | 2.86 | 97.14 |
| **Δ-norm of Dwt coeff.** | | | | |
| W=64, no=4 | 71.86 | 83.29 | 11.43 | 73.57 |
| W=64, no=16 | 83.00 | 90.29 | 7.29 | 84.86 |
| W=256,no=4 | 89.43 | 95.43 | 6.00 | 91.43 |
| W=256, no=16 | 96.43 | 99.57 | 3.14 | 97.29 |

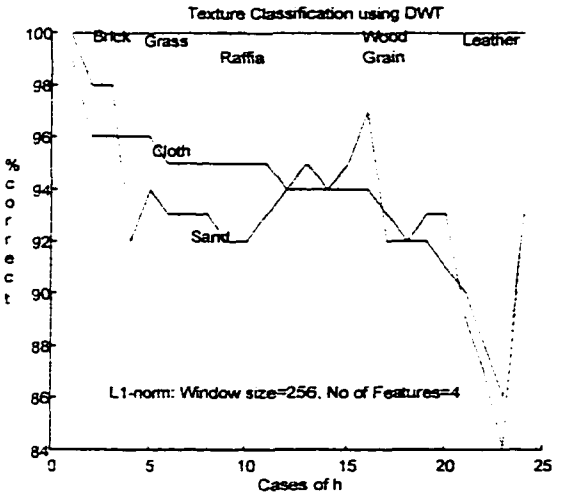



Figure 5.1. Texture Classification using the $L_1$-norm of the DWT coefficients using four features.

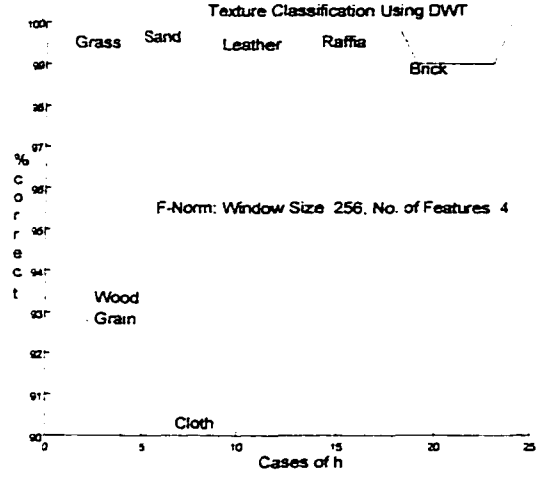



Figure 5.2. Texture Classification using the F-norm of the DWT coefficients and four features.

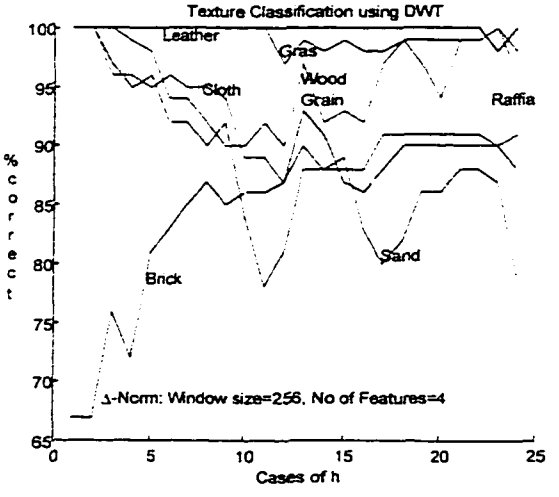



Figure 5.3. Texture Classification using the Δ-norm of the DWT coefficients and four features.

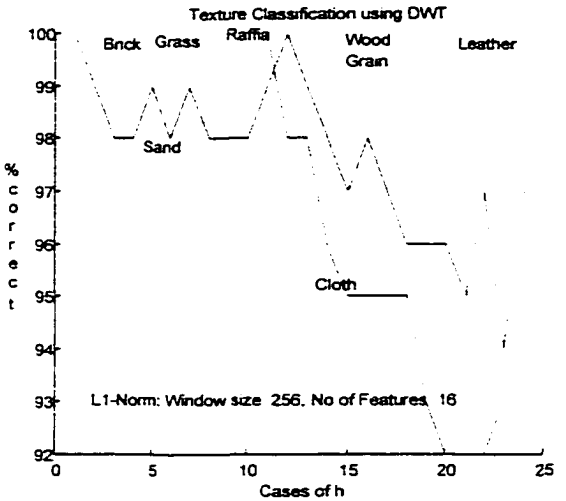



Figure 5.4. Texture Classification using the $L_1$-norm of the DWT Coefficients and 16 features.

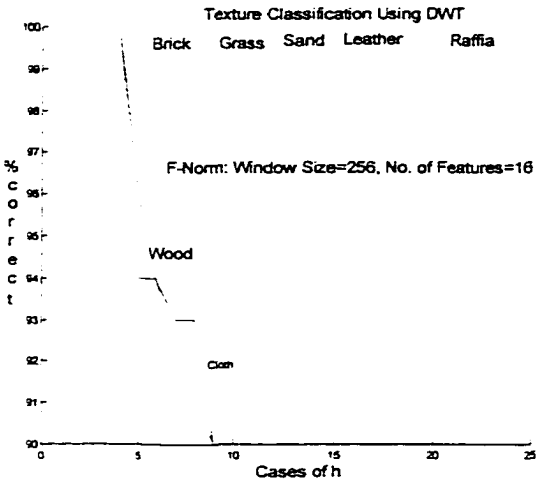



Figure 5.5. Texture Classification using the F-norm of the DWT Coefficients and 16 features.

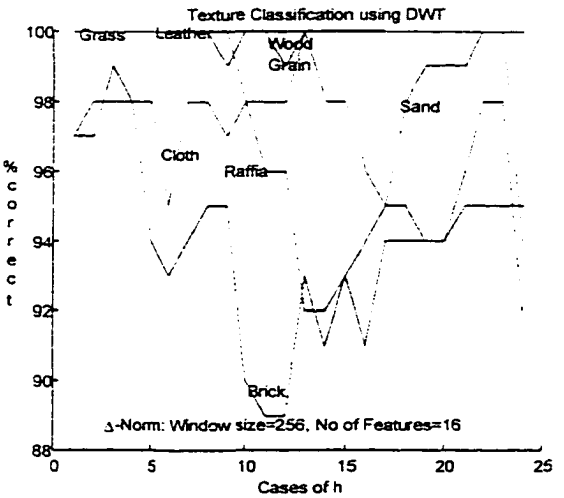



Figure 5.6. Texture Classification using the Δ-norm of the DWT Coefficients and 16 features.

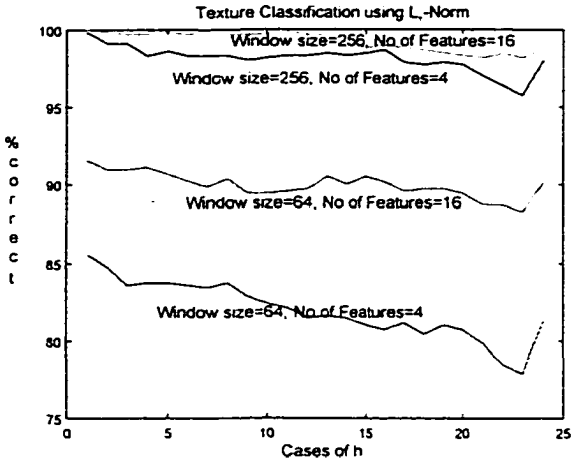



Figure 5.8. Comparison of percentage of overall correct classification rate as a function of wavelet bases for different window sizes and number of features.

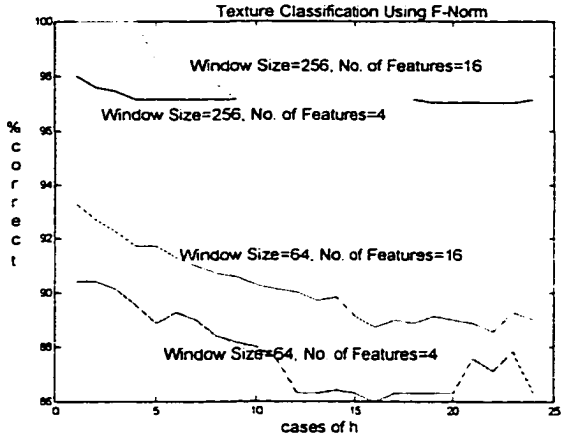



Figure 5.9. Comparison of % of overall correct classification rate as a function of wavelet bases for different window sizes and number of features.

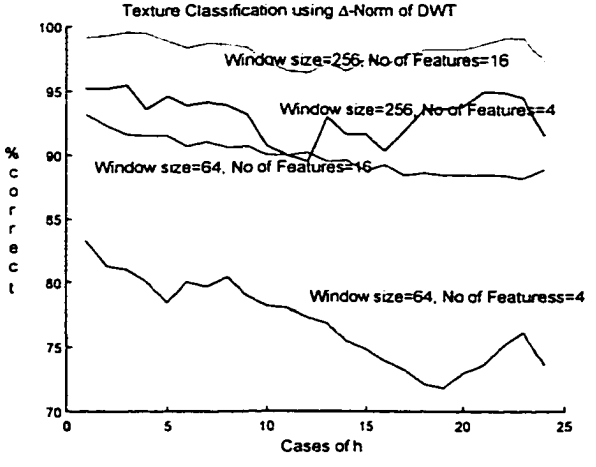



Figure 5.10. Comparison of % of overall correct classification rate as a function of wavelet bases for different window sizes and number of features.

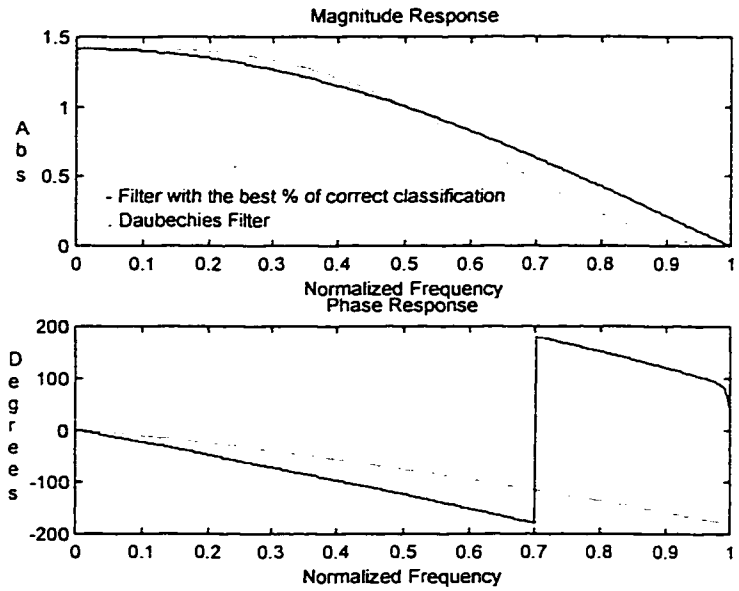



Figure 5.11. Magnitude and phase response of the filter with the best classification and the Daubechies filter.

## 5.3. EFFECT OF MOTHER WAVELET ON TEXTURE CLASSIFICATION USING ENERGY-BASED WAVELET-PACKET TREE

### 5.3.1. CHOICE OF MOTHER WAVELET FOR TEXTURE CLASSIFICATION

Using each mother wavelet, the energy based tree-structured decomposition is applied to the textures in the database and both training and classification is performed. For each wavelet, the procedure described in Section 4.2.2 is repeated and the classification performance is calculated for all textures.

### 5.3.2. CLASSIFICATION PERFORMANCE AND RESULTS

The percentage of correct classification rate and the overall correct classification rate is compared for all of the wavelets using different window sizes and either the Euclidean distance defined in equation (4.4) or the simplified Mahalanobis distance defined in equation (4.6) classifiers. The results are plotted in Figures 5.12 to 5.17. The overall classification rate is the average of the classification rates of all the textures in the database.

The overall classification rate varies from 95.57 to 97.71 for window size 256 using the Euclidean distance classifier. In other words, the classification rate increases by 2.14% from the worst wavelet basis to the best. The classification rate varies from 96.57 to 98.29, a 1.71% change using the simplified Mahalanobis distance.

The overall classification rate varies from 88.29% to 90.86% for window size 64 using the Euclidean distance classifier. The classification rate increases by 2.57% from the worst wavelet basis to the best. Using the simplified Mahalanobis distance, classification performance varies from 92.86% to 97.57%, which is a 4.71% variation.

The overall classification may not vary much with respect to the different wavelet bases, but a particular texture of interest may show a considerably improved classification rate for a particular wavelet basis. In addition, using only the four-tap wavelet basis results in better (or at least equal) overall performance as compared to the Battle-Lemarie 16 tap wavelet in classification. This suggests that a

particular wavelet may result in improved performance compared to others when used for a specific application. The question is: How much improvement is one looking for?

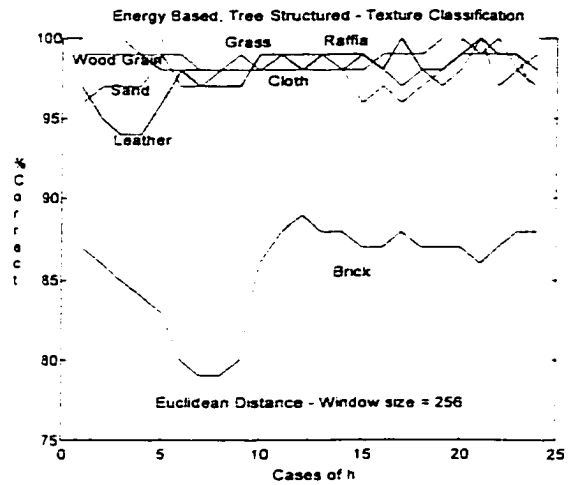



Figure 5.1. Tree-Structured classification for different wavelet bases using Euclidean distance.

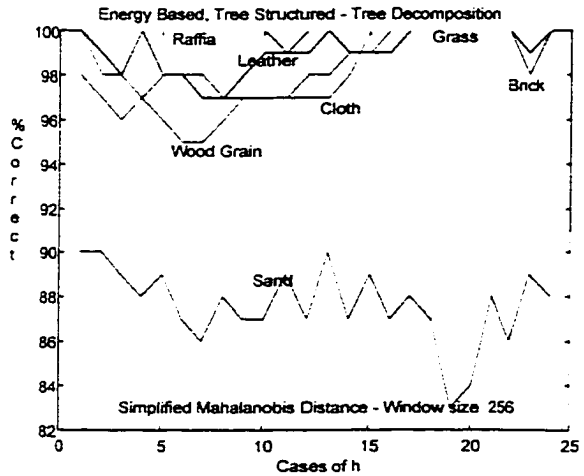



Figure 5.2. Tree Structured decomposition using simplified Mahalanobis distance.

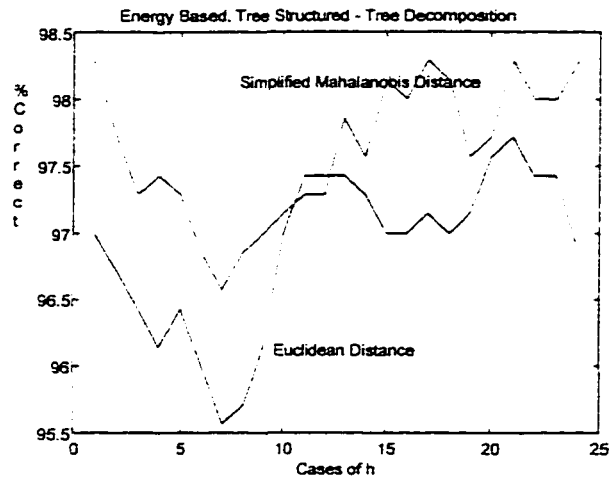



Figure 5.3. Overall classification rate.

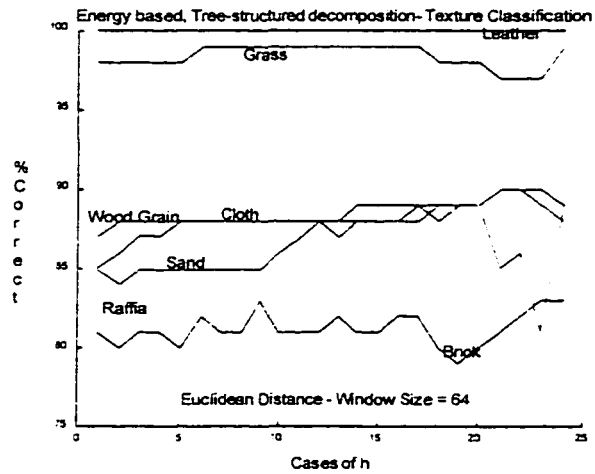



Figure 5.4. Tree-Structured Classification for different wavelet bases with window size 64 using Euclidean distance.

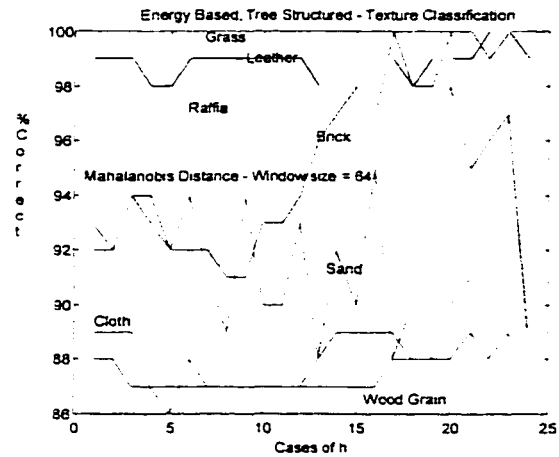



Figure 5.5. Tree-Structured Classification for different wavelet bases with window size 64 using simplified Mahalanobis distance.

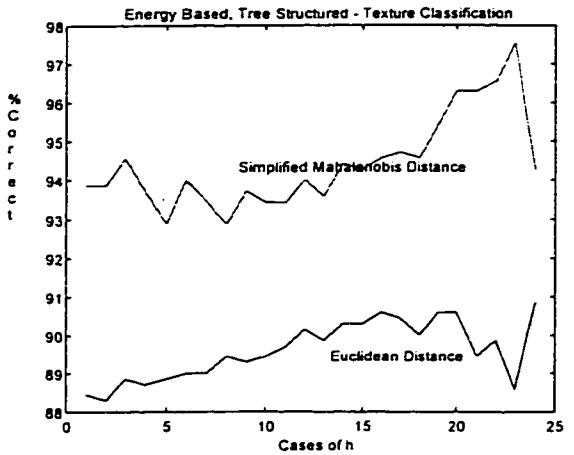



Figure 5.17. Overall classification rate for window size 64.

### 5.3.3. COMPARISON OF PERFORMANCE

The percentages of correct classification rates for the $L_1$-norm, F-norm and $\Delta$-norm measures are plotted in Figures 5.18 to 5.23 for window sizes 256 and 64 respectively. The overall percentage of correct classification rate, i.e. the average of the percentage of correct classification rates of all the textures in the database is compared for all the measures. The overall classification percentages are plotted for different wavelets using the $L_1$-norm, F-norm and $\Delta$-norm measures for window sizes 256 and 64 in Figure 5.24. Table 5.3 compares the performance for all the features using different wavelets. The minimum and maximum percentage of correct classification, percentage of variation among the wavelets and the classification rate for the Daubechies four tap wavelet are listed in this table. Using various wavelets, the classification performance improves compared to the Daubechies filter using the energy-based wavelet packet tree decomposition. Both the F-norm and $L_1$-norm perform equally well for window size 256, whereas the absolute sum seems to perform better for window size 64.

TABLE-5.3: OVERALL CLASSIFICATION PERFORMANCE USING ENERGY-BASED TREE DECOMPOSITION

| | Min. % | Max. % | Variation % | Daubechies % |
|---|---|---|---|---|
| **$L_1$-norm of Dwt coeff.** | | | | |
| W=64, no=4 | 81.7 | 85.57 | 3.86 | 84.00 |
| W=256,no=4 | 97.29 | 98.43 | 1.14 | 98.14 |
| **F-norm of Dwt coeff.** | | | | |
| W=64, no=4 | 85.57 | 87.86 | 2.29 | 87.14 |
| W=256,no=4 | 97.00 | 98.86 | 1.86 | 97.57 |
| **Δ-norm of Dwt coeff.** | | | | |
| W=64, no=4 | 80.00 | 89.71 | 9.71 | 81.43 |
| W=256,no=4 | 81.29 | 91.43 | 10.14 | 88.29 |
| **Abs. mean of Dwt coeff.** | | | | |
| W=64, no=4 | 88.29 | 90.86 | 2.57 | 90.86 |
| W=256,no=4 | 95.57 | 97.71 | 2.14 | 96.85 |

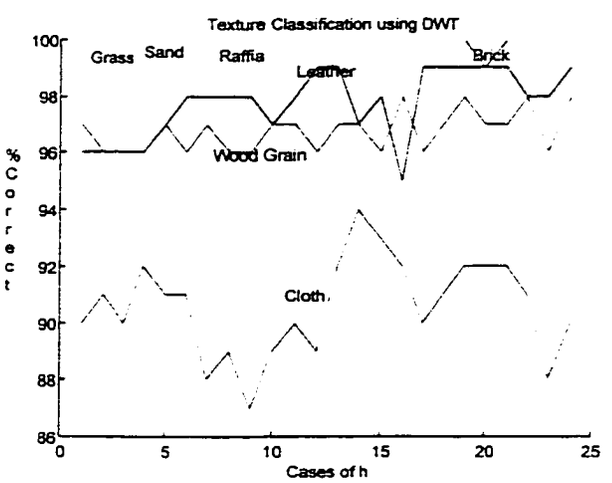



Figure 5.1. Tree-Structured classification Using $L_1$-norm for Window size 256 using Euclidean distance.

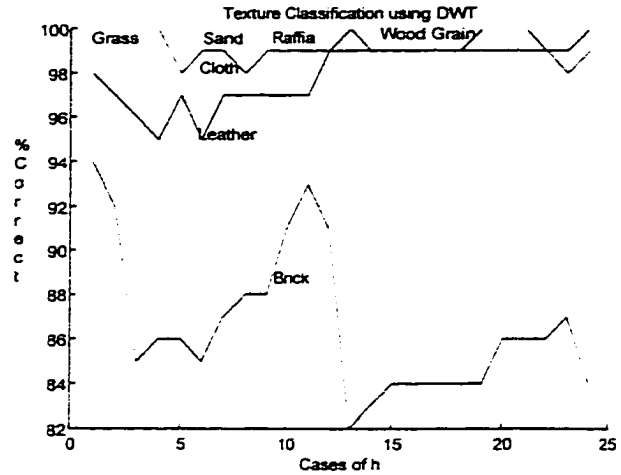



Figure 5.2. Tree-Structured classification using F-norm for window size 256 using Euclidean distance.

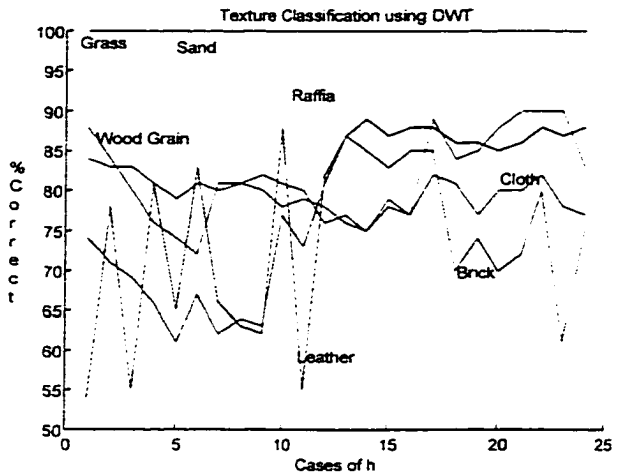



Figure 5.3. Tree-Structured classification using Δ-norm for window size 256 using Euclidean distance.

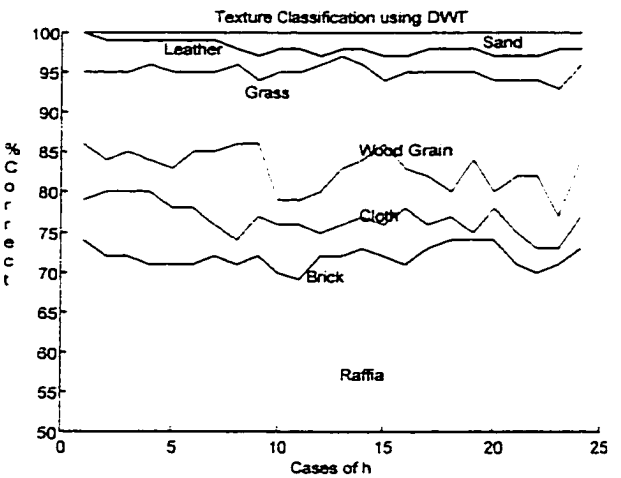



Figure 5.4. Tree-Structured classification using $L_1$-norm for window size 64 using Euclidean distance.

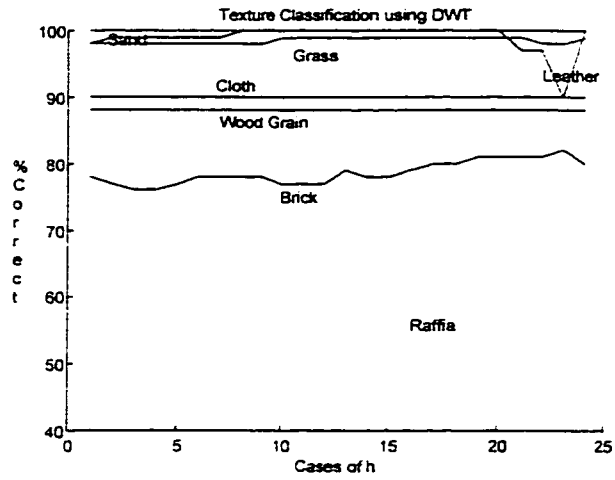



Figure 5.5. Tree-Structured classification using F-norm for window size 64 using Euclidean distance.

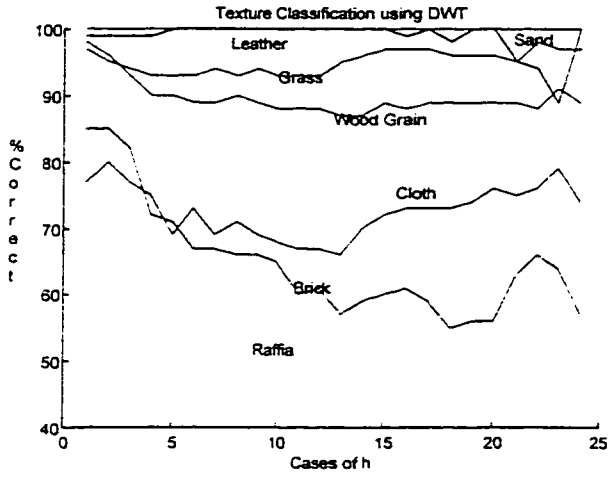



Figure 5.6. Tree-Structured classification using Δ-norm for window size 64 using Euclidean distance.

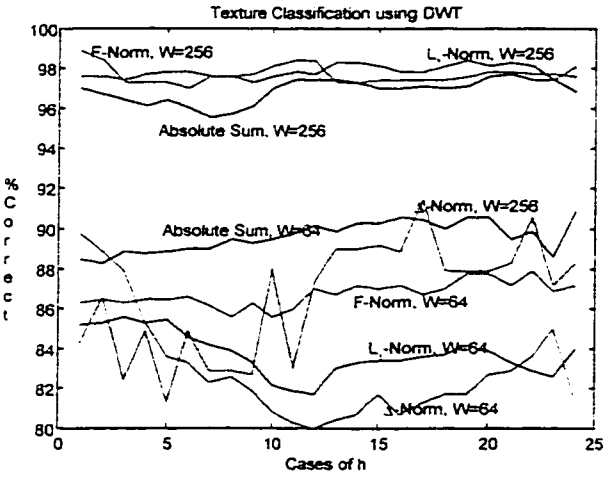



Figure 5.7. Overall classification rate for all measures and for window sizes 256 and 64.

## 5.4. EFFECT OF MOTHER WAVELET ON TEXTURE CLASSIFICATION USING SEPARABILITY-BASED WAVELET PACKET TRANSFORM

Using each mother wavelet, the separability-based tree-structured decomposition is applied to the textures in the database and both training and classification is performed. For each wavelet, the procedure described in Section 4.4.2 is repeated using the class separability-based measure. Then the features are dimensionally reduced and the classification percentage is calculated for all the textures.

### 5.4.1. CLASSIFICATION PERFORMANCE AND RESULTS

The overall classification results are listed in Table 5.4 for different wavelets using both the neural network and the Euclidean distance classifiers for features $\mu_2$ and $\mu_3/\mu_2$. Twenty-four wavelets are applied to the above algorithm for decomposition levels 2, 3 and 4. The minimum and maximum classification performances for these wavelets along with the Daubechies wavelet are compared. The classification percentage increased for the best wavelet by extracting 8 features using $\mu_2$ and $\mu_3/\mu_2$ compared to extracting 4 features using only $\mu_2$. Also, the performance did not decrease with an increase in the depth of the decomposition for the best wavelet unlike the case using the Daubechies wavelet with the Euclidean distance classifier. The classification performance using the neural network is comparable to the performance using the Euclidean distance for the best wavelet, which is not true for the Daubechies wavelet. The classification performance for the textures in the database is plotted in Figures 5.25-5.36 for various wavelets, levels and classifiers. The overall percentage of classification is plotted in Figures 5.37-5.43 for the various wavelets, levels, features and classifiers.

### 5.4.2. COMPARISON OF PERFORMANCE

The overall percentage of correct classification rate (i.e. the average of the percentage of correct classification rates of all the textures in the database) is compared for all the measures. Table 5.5 compares the performance for all the features using different wavelets. The minimum and maximum percentage of correct classification and the classification rate for the Daubechies four-tap wavelet are listed in this table. The classification performance increased considerably using the best wavelet when compared to the

Daubechies wavelet using all the measures for all the levels of decomposition. The highest performance for the best wavelet is 99.2% using the absolute sum measure at the fourth level of the tree decomposition using the Euclidean distance classifier. But, using either the F-norm or the features $\mu_2$ and $\mu_3/\mu_2$ results in a 98.9% correct classification rate using just a second level of decomposition.

**TABLE-5.4: OVERALL CLASSIFICATION PERFORMANCE FOR VARIOUS LEVELS USING EUCLIDEAN DISTANCE AND NEURAL NET CLASSIFIERS**

| Level of Decomposition | Feature | Euclidean Distance | | | Neural Net | | |
|---|---|---|---|---|---|---|---|
| | | min. % | max. % | Daubechies % | min. % | max. % | Daubechies % |
| 2 | $\mu_2$ | 96.3 | 98.4 | 97.3 | 77.6 | 98.4 | 96.8 |
| 3 | $\mu_2$ | 97.0 | 98.3 | 97.3 | 93.0 | 98.2 | 97.4 |
| 4 | $\mu_2$ | 97.8 | 98.8 | 97.8 | 78.0 | 98.0 | 94.9 |
| 2 | $\mu_2$ and $\mu_3/\mu_2$ | 96.2 | 98.9 | 97.8 | 87.6 | 97.9 | 95.8 |
| 3 | $\mu_2$ and $\mu_3/\mu_2$ | 94.1 | 98.8 | 97.2 | 87.3 | 98.1 | 97.2 |
| 4 | $\mu_2$ and $\mu_3/\mu_2$ | 96.6 | 99.0 | 97.1 | 94.0 | 97.7 | 95.3 |

**TABLE-5.5: OVERALL CLASSIFICATION PERFORMANCE USING EUCLIDEAN DISTANCE FOR VARIOUS FEATURES FOR VARIOUS LEVELS**

| Level of Decomposition | Feature | Euclidean Distance | | |
|---|---|---|---|---|
| | | min. % | max. % | Daubechies % |
| 2 | $L_1$-norm | 92.4 | 95.8 | 92.9 |
| 3 | $L_1$-norm | 93.5 | 98.2 | 96.9 |
| 4 | $L_1$-norm | 95.2 | 98.9 | 98.3 |
| 2 | $\Delta$-norm | 86.8 | 92.2 | 89.3 |
| 3 | $\Delta$-norm | 94.3 | 97.4 | 96.5 |
| 4 | $\Delta$-norm | 95.1 | 98.5 | 97.5 |
| 2 | abs. sum | 97.1 | 98.5 | 97.5 |
| 3 | abs. sum | 97.5 | 98.8 | 98.4 |
| 4 | abs. sum | 95.5 | 99.2 | 98.3 |

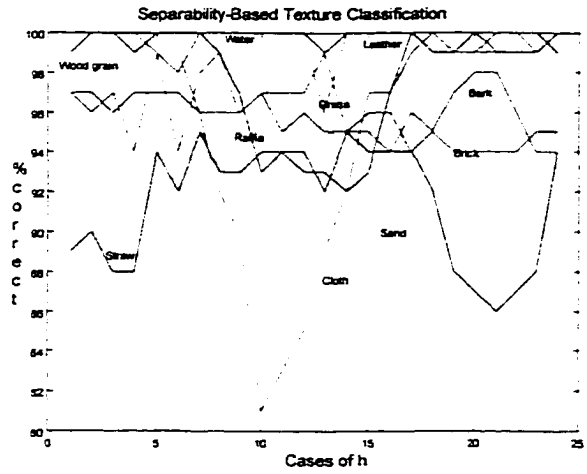



Figure 5.1. Texture classification using $\mu_2$ and $\mu_3/\mu_2$ for level 2 using Euclidean distance classifier.

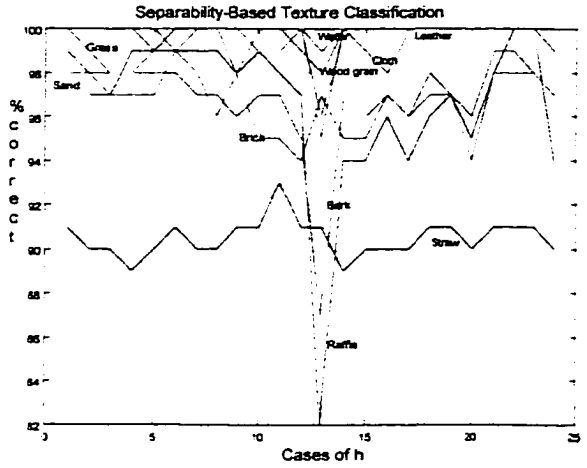



Figure 5.2. Texture classification using $\mu_2$ and $\mu_3/\mu_2$ for level 3 using Euclidean distance classifier.

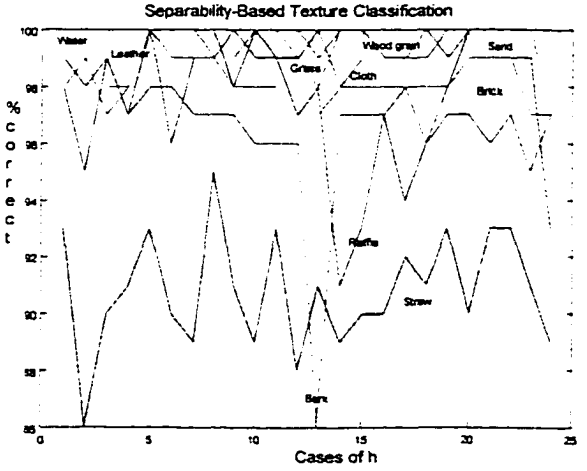



Figure 5.3. Texture classification using $\mu_2$ and $\mu_3/\mu_2$ for level 4 using Euclidean distance classifier.

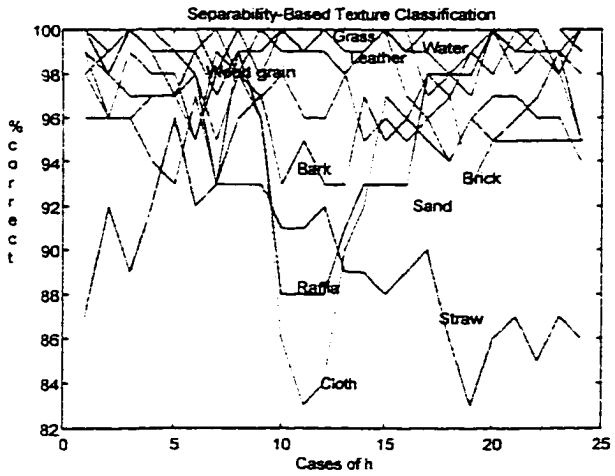



Figure 5.4. Texture classification using $\mu_2$ and $\mu_3/\mu_2$ for level 2 using Neural Net classifier.

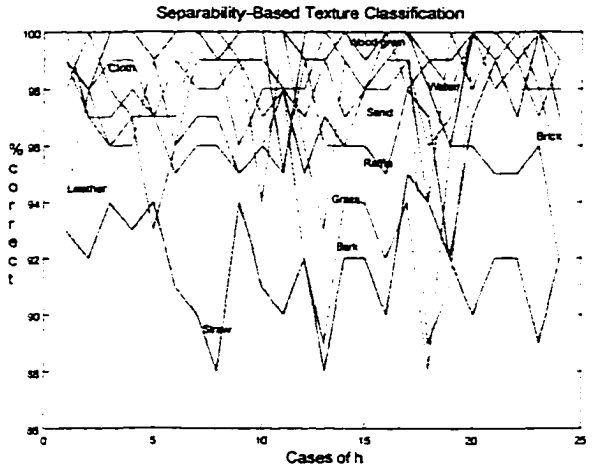



Figure 5.5. Texture classification using $\mu_2$ and $\mu_3/\mu_2$ for level 3 using Neural Net classifier.

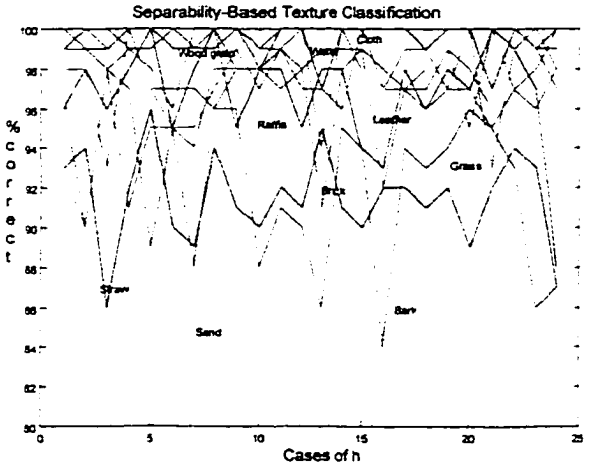



Figure 5.6. Texture classification using $\mu_2$ and $\mu_3/\mu_2$ for level 4 using Neural Net classifier.

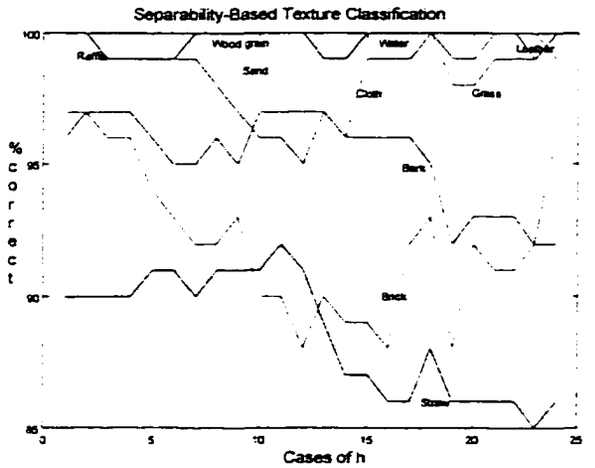



Figure 5.7. Texture classification using $\mu_2$ for level 2 using Euclidean distance classifier.

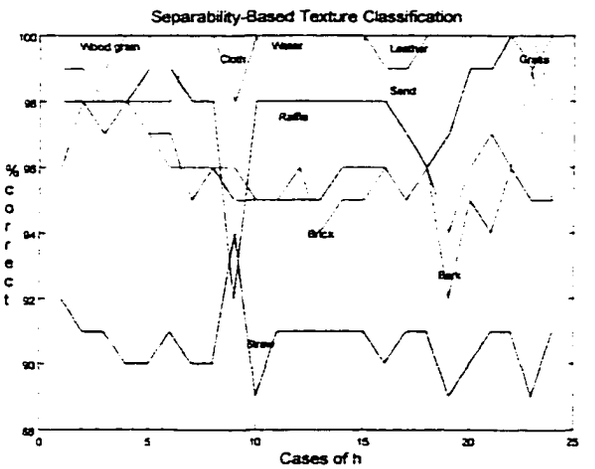



Figure 5.8. Texture classification using $\mu_2$ for level 3 using Euclidean distance classifier.

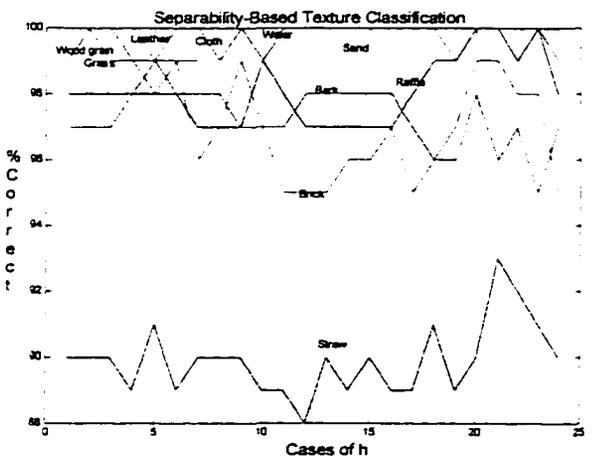



Figure 5.9. Texture classification using $\mu_2$ for level 4 using Euclidean distance classifier.

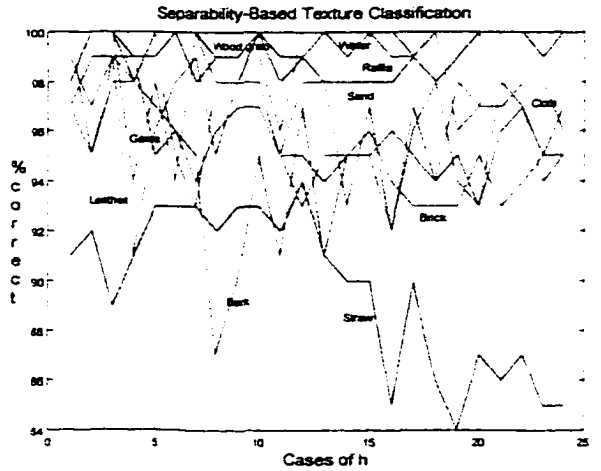



Figure 5.10. Texture classification using $\mu_2$ for level 2 using Neural Net classifier.

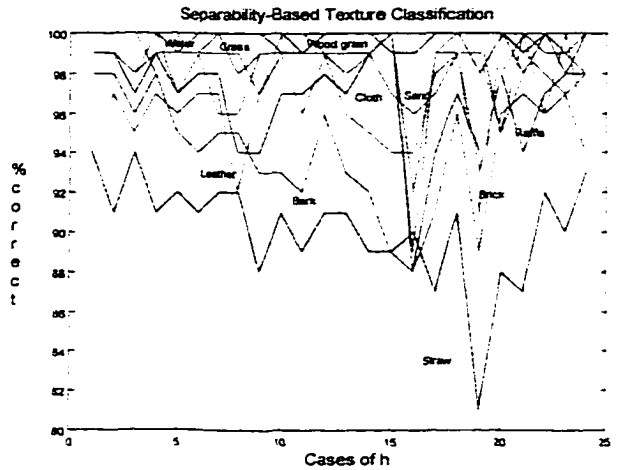



Figure 5.11. Texture classification using $\mu_2$ for level 3 using Neural Net classifier.

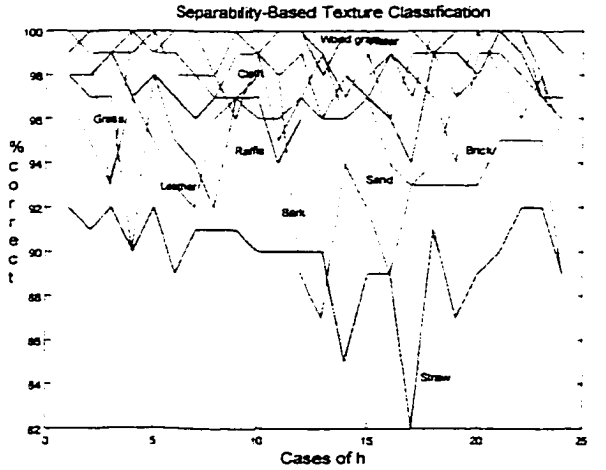



Figure 5.12. Texture classification using $\mu_2$ for level 4 using Neural Net classifier.

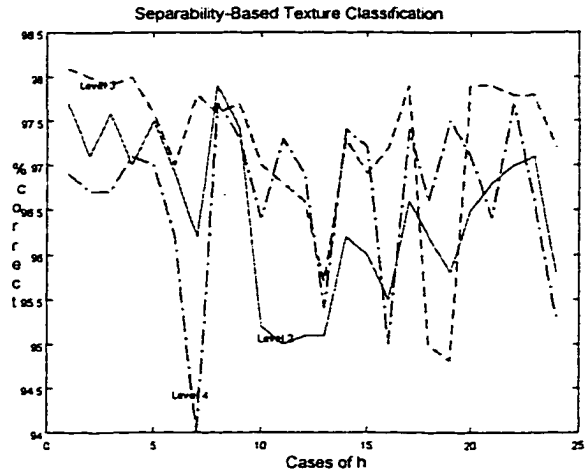



Figure 5.37. Overall percentage of classification using $\mu_2$ and $\mu_3/\mu_2$ using Neural Net classifier.

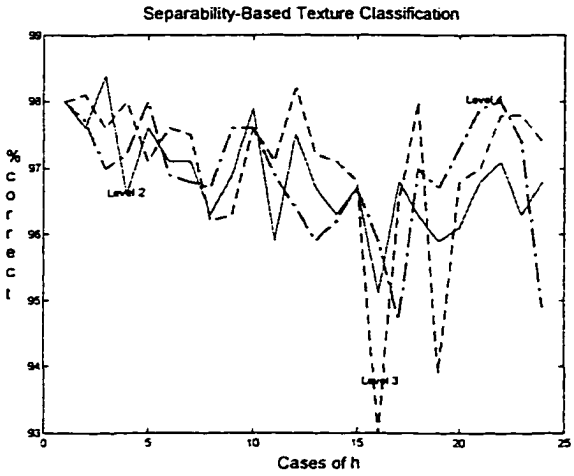



Figure 5.38. Overall percentage of classification using $\mu_2$ using Neural Net classifier.

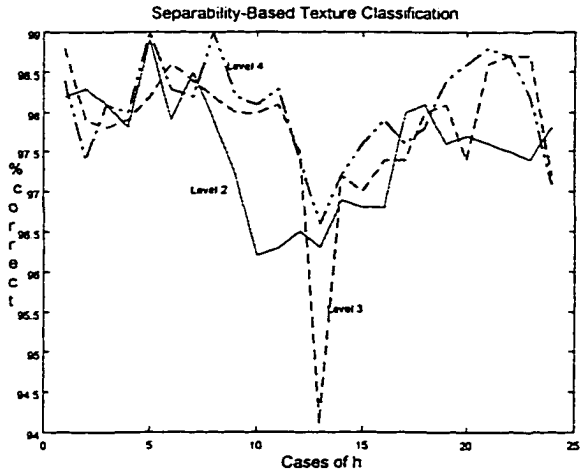



Figure 5.39. Overall percentage of classification using $\mu_2$ and $\mu_3/\mu_2$ using Euclidean distance classifier.

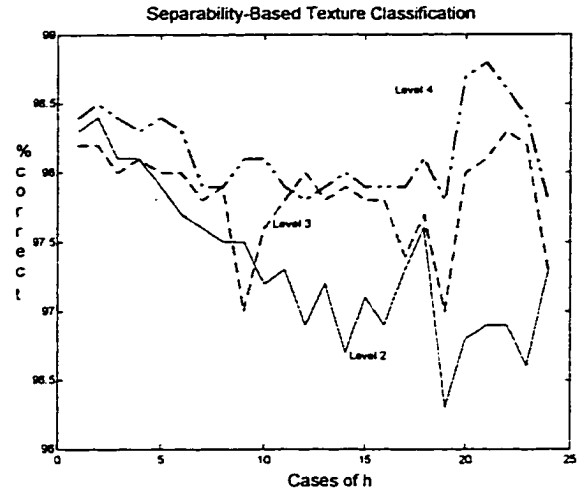



Figure 5.40. Overall percentage of classification using $\mu_2$ using Euclidean distance classifier.

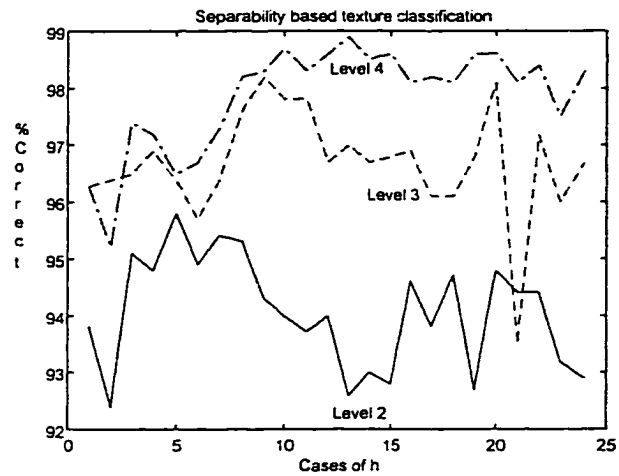



Figure 5.41. Overall percentage of classification using $L_1$-norm using Euclidean distance classifier.

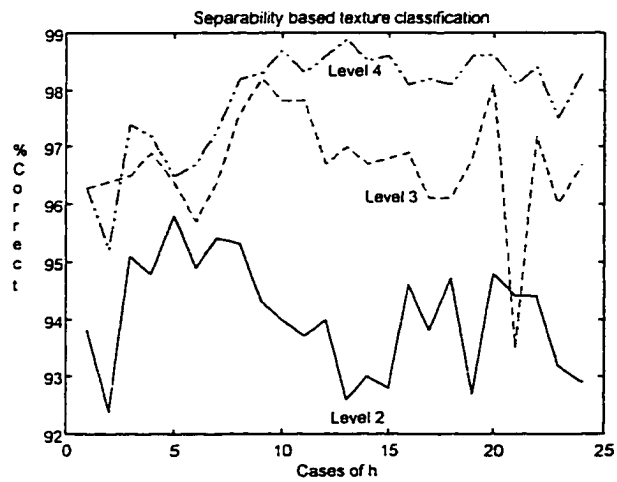



Figure 5.42. Overall percentage of classification using $\Delta$-norm using Euclidean distance classifier.

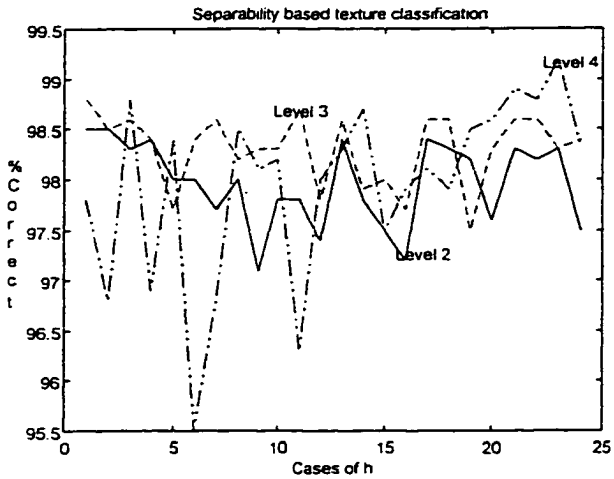



Figure 5.43. Overall percentage of classification using absolute sum using Euclidean distance classifier.

## 5.5. EFFECT OF MOTHER WAVELET ON CLASSIFICATION USING WAVELET TRANSFORM

Using each mother wavelet, the pyramid decomposition is applied to the textures in the database and both training and classification is done. Features are extracted using the F-norm and the second level decomposition. Then the features are dimensionally reduced and the classification percentage is calculated for all textures.

Table 5.6 compares the classification performance for the Daubechies 4-tap wavelet with the second level of decomposition using the Euclidean distance classifier with and without dimensionality reduction. The classification performance is considerably improved using the dimensionality reduction technique for the octave tree structure. Using various wavelets does not improve the classification performance considerably compared to that of the Daubechies wavelet for the pyramid tree.

The results are also compared by obtaining more features from each band using the procedure of the algorithm developed in Section 4.6. The performance does not improve without the dimensionality reduction, even when more features are extracted. Also, the effect of increasing the database size is observed by adding three textures to the previous database. The performance decreases considerably in all cases. In other words, the performance decreases with an increase in database size, even when more features are extracted and dimensionally reduced.

TABLE-5.6: OVERALL CLASSIFICATION PERFORMANCE USING F-NORM FOR SECOND LEVEL OF DECOMPOSITION

| Level | No. of Features | DataBase Size | Dimensionality Reduction | min. % | max. % | mean % | Wavelets median % | Daubechies % |
|---|---|---|---|---|---|---|---|---|
| 2 | 28 | 10 | No | 80.50 | 83.40 | 82.10 | 81.95 | 83.00 |
| 2 | 28 | 10 | Yes | 92.20 | 94.70 | 93.70 | 93.75 | 94.30 |
| 2 | 28 | 7 | No | 85.43 | 87.57 | 86.67 | 86.64 | 87.57 |
| 2 | 28 | 7 | Yes | 97.86 | 99.29 | 98.62 | 98.71 | 99.00 |
| 2 | 7 | 10 | No | 80.50 | 84.10 | 82.38 | 82.15 | 83.50 |
| 2 | 7 | 10 | Yes | 87.90 | 95.30 | 93.33 | 93.45 | 95.20 |
| 2 | 7 | 7 | No | 85.43 | 89.14 | 87.50 | 87.50 | 88.43 |
| 2 | 7 | 7 | Yes | 96.14 | 99.00 | 97.81 | 97.93 | 98.86 |

## 5.6. INFLUENCE OF VARIOUS PARAMETERS ON CLASSIFICATION

### 5.6.1 THE RELATIVE PERFORMANCE OF DIMENSIONALITY REDUCTION

As described in Section 2.5, an appropriate form of dimensionality reduction is crucial to the success of the wavelet transform as a basis for classification. The role of dimensionality reduction when using the wavelet transform is examined here. Three methods have been used to derive features from a wavelet decomposition that are also examined: a representation by wavelet transform local extrema, a representation by wavelet transform first order norm, and a representation by wavelet transform sub-band energy.

A reduction of the wavelet transform using feature selection has been performed here using a class separability measure. The $L_1$-norm, F-norm (sub-band energy) and $\Delta$-norm (local extrema) are used as the feature extractors. Sixteen features are obtained from one level wavelet decomposition as explained in Section 5.2.1 for each measure and these 16 features are dimensionally reduced to four features. This dimensionality reduction is repeated for all the wavelets and performance statistics are obtained. The minimum, maximum, mean and median classification performances for the wavelets are obtained for the above methods. These results are compared in Table 5.7 to the corresponding results without the

dimensionality reduction component. The percentage of correct classification rates for the three measures using the wavelets are plotted in Figures 5.44 to 5.46 and the overall percentages are plotted in Figure 5.47 for window size 64.

The feature extraction using the F-norm measure seems to give better classification performance in the case of textures. It is presumed that a substantial degree of temporal dispersion is present in a set of signals. Computing the energy in each WT sub-band smooths the effects of temporal shift, which might be the reason for the improvement in the performance.

### 5.6.2. THE DEPTH OF THE DECOMPOSITION

Aside from the selection of the mother wavelet, the only other adjustable parameter when performing the WT is the depth of the decomposition. For a signal of length N, the maximum depth of decomposition is $J = \log_2 N$. For each texture, a WT feature set is extracted from the full (uniform) WT decomposition at the second level. The test set average classification performance statistics are listed in Table 5.8 for decomposition levels 1 and 2 and for differing numbers of features with and with out dimensionality reduction using the F-norm measure. The percentage of classification rates are plotted in Figure 5.48 for the second level decomposition using the F-norm for feature reduction.

Regardless of the number of features, the classification performance improves by increasing depth of decomposition. It is clear that the frequency resolution in the subbands provided by full WT decomposition is useful for texture classification.

### 5.6.3. THE EFFECT OF INCREASING THE DATABASE SIZE

In this section, the effect of increasing the database size, i.e. increasing the number of textures (classes) in the database on the classification performance is explored. Three more textures are added to the previous database, which has 7 textures. The classification performance is computed for these 10 textures using the measures for the first and second level decompositions for all the wavelets using the Euclidean distance classifier. They are listed in Table 5.9. The classification results are plotted in Figures 5.49 and 5.50 for the second level decomposition using 10 textures for the F-norm and feature reduction. Figure 5.51 shows the

overall percentage of classification rates for a second level decomposition using 10 textures and Figure 5.52 shows the performance for the first level decomposition using 7 textures only, both with and without the dimensionality reduction.

The results show that the classification performance increases with the depth of the decomposition. The first level of decomposition may not provide enough information for classification when the textures in the database have overlapped clusters. The classification performance can be improved in the first level by extracting more features. Further improvement results when the dimensionality is reduced. In the second level of decomposition, extracting more features from the subbands does not increase the classification performance, even with dimensionality reduction. This implies that too many features are counterproductive and that obtaining the optimum number of coefficients is crucial for classification performance. The optimum number of features varies for each level of decomposition.

**TABLE-5.7: OVERALL CLASSIFICATION PERFORMANCE WITH DIMENSIONALITY REDUCTION**

| Feature | Dimensionality Reduction | Wavelets | | | | |
|---|---|---|---|---|---|---|
| | | min. % | max. % | mean % | median % | Daubechies % |
| F-norm | No | 88.57 | 93.29 | 90.20 | 89.93 | 89.00 |
| | Yes | 89.43 | 98.57 | 93.66 | 93.29 | 90.86 |
| $L_1$-norm | No | 88.29 | 91.57 | 89.99 | 89.93 | 90.1 |
| | Yes | 87.86 | 94.29 | 91.43 | 91.36 | 91.29 |
| $\Delta$-norm | No | 83.00 | 90.29 | 86.04 | 85.29 | 84.86 |
| | Yes | 81.29 | 91.86 | 84.80 | 84.07 | 81.29 |

TABLE-5.8: OVERALL CLASSIFICATION PERFORMANCE USING F-NORM FOR VARIOUS LEVELS

| Level | No. of Features | Dimensionality Reduction | Wavelets | | | | |
|---|---|---|---|---|---|---|---|
| | | | min. % | max. % | mean % | median % | Daubechies % |
| 1 | 4 | No | 86.00 | 90.43 | 87.71 | 87.57 | 86.29 |
| 1 | 16 | No | 88.57 | 93.29 | 90.20 | 89.93 | 89.00 |
| 1 | 16 | Yes | 89.43 | 98.57 | 93.66 | 93.29 | 90.86 |
| 2 | 16 | No | 90.57 | 91.86 | 91.13 | 91.14 | 91.43 |
| 2 | 16 | Yes | 97.86 | 99.57 | 98.83 | 98.86 | 99.00 |
| 2 | 64 | No | 93.43 | 93.86 | 93.68 | 93.71 | 93.86 |
| 2 | 64 | Yes | 99.57 | 99.86 | 99.72 | 99.71 | 99.71 |

TABLE-5.9: OVERALL CLASSIFICATION PERFORMANCE USING F-NORM WITH DATABASE SIZE = 10

| Level | No. of Features | Dimensionality Reduction | Wavelets | | | | |
|---|---|---|---|---|---|---|---|
| | | | min. % | max. % | mean % | median % | Daubechies % |
| 1 | 16 | No | 77.20 | 79.00 | 77.88 | 77.70 | 77.60 |
| 1 | 16 | Yes | 81.70 | 92.30 | 86.72 | 86.85 | 83.20 |
| 2 | 16 | No | 86.30 | 87.80 | 86.99 | 86.95 | 87.30 |
| 2 | 16 | Yes | 96.30 | 98.40 | 97.34 | 97.30 | 97.30 |
| 2 | 64 | No | 86.00 | 87.00 | 86.55 | 86.55 | 86.50 |
| 2 | 64 | Yes | 96.10 | 97.50 | 96.91 | 96.90 | 96.60 |

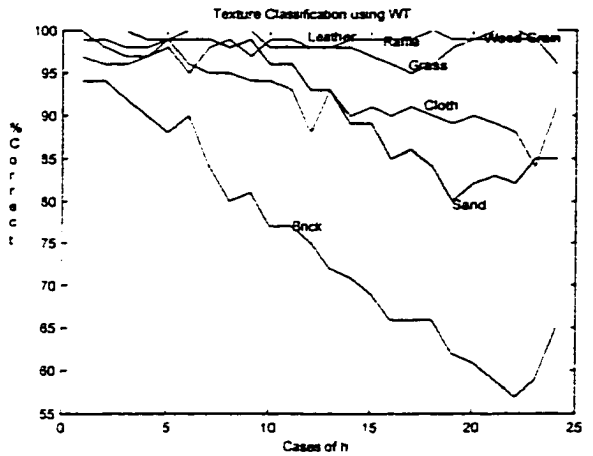



Figure 5.1. Texture Classification using the F-norm of the DWT Coefficients and dimensionally reduced four features.

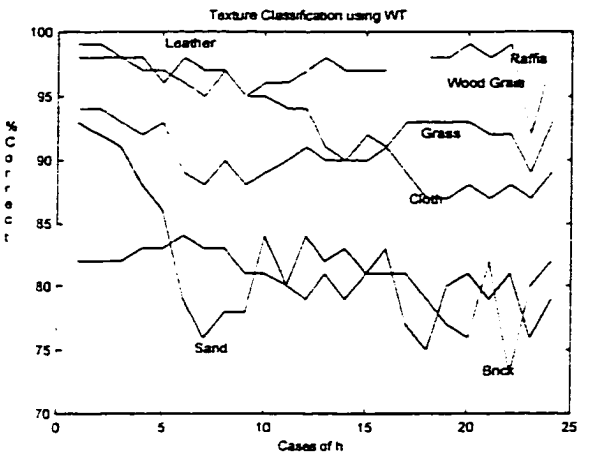



Figure 5.2. Texture Classification using the $L_1$-norm of the DWT Coefficients and dimensionally reduced four features.

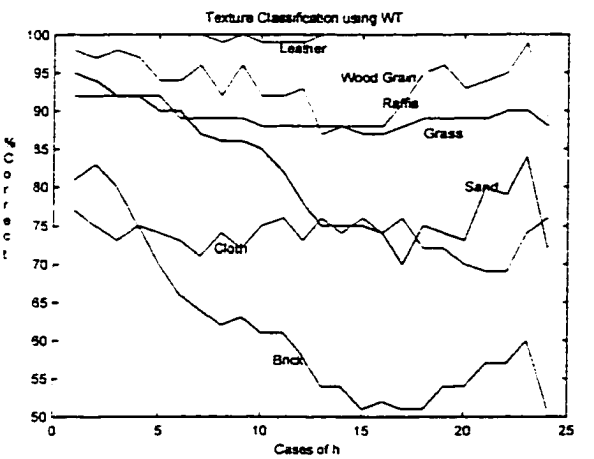



Figure 5.3. Texture Classification using the Δ-norm of the DWT Coefficients and dimensionally reduced four features.

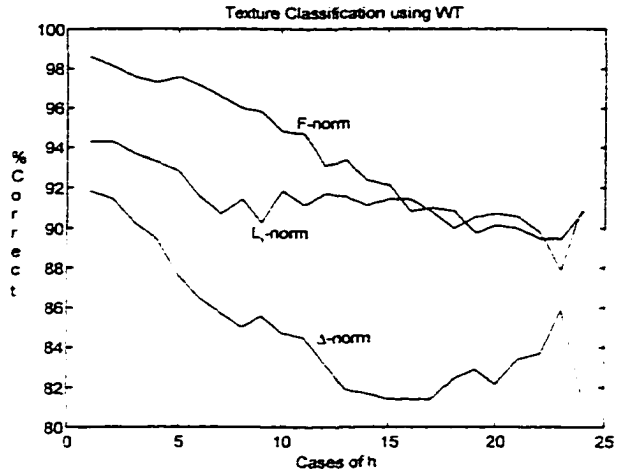



Figure 5.4. Texture Classification using feature extraction and dimensionally reduced four features.

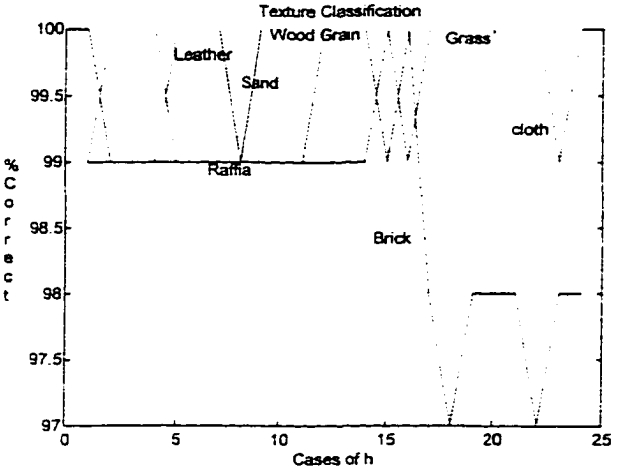



Figure 5.5. Texture Classification using the F-norm for level 2 and dimensionally reduced four features.

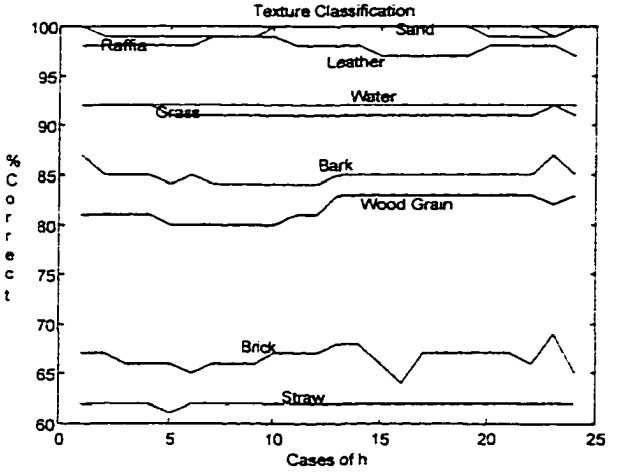



Figure 5.6. Texture Classification for 10 textures using the F-norm for level 2.

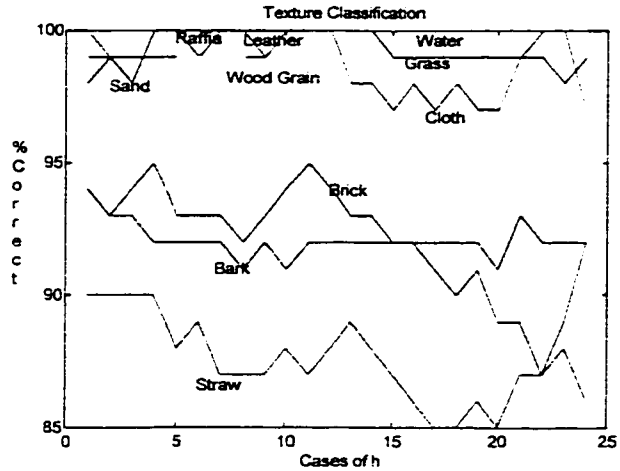



Figure 5.7. Texture Classification for 10 textures using the F-norm for level 2 and dimensionally reduced four features.

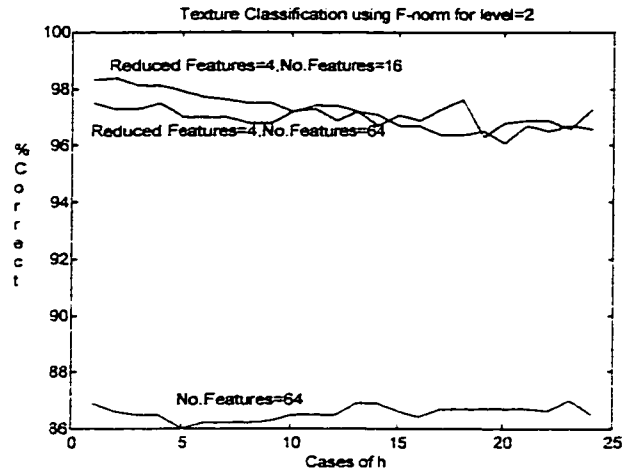



Figure 5.8. Overall Texture Classification for 10 textures using the F-norm for level 2 and dimensionally reduced four features.

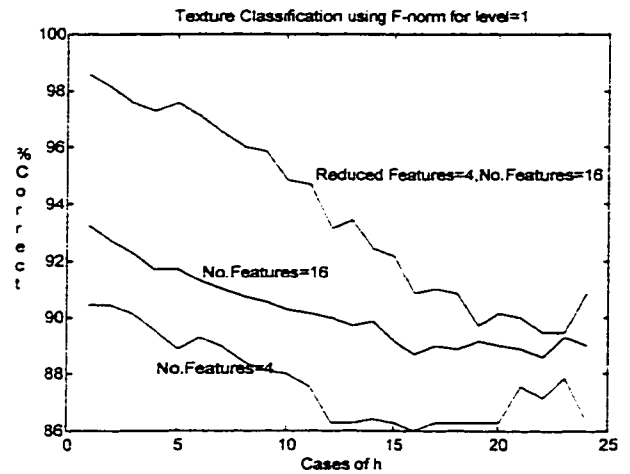



Figure 5.9. Overall Texture Classification for 7 textures using the F-norm for level 1 and dimensionally reduced four features.

## 5.7. PERFORMANCE SUMMARY

### 5.7.1. THE RELATIVE PERFORMANCE AMONG THE FEATURE SETS

The previous sections have described the applications of the WT- (Wavelet Transform) and the WPT- (Wavelet Packet Transform) based features sets to the task of texture classification. Closely coupled with the use of feature set is the means of dimensionality reduction. When using time-frequency based representations, it is crucial that an appropriate form of dimensionality reduction be performed. It has been demonstrated that separability based feature projection or dimensionality reduction significantly improves the performance

The classification performances for all the features ($L_t$-norm, F-norm and $\Delta$-norm) using the uniform or wavelet packet decomposition and wavelet transform are listed in Table 5.10 using the second level decomposition. The results show that performances using these different measures are comparable, and the F-norm, which is the average energy across the subbands, outperforms the other measures when using any wavelet based decomposition along with the dimensionality reduction.

### 5.7.2. THE RELATIVE PERFORMANCE AMONG THE TREE STRUCTURES

When using the wavelets for analysis, a few parameters can be varied. These are the choice of the mother wavelet, the depth of the decomposition and the time-frequency tiling. Any image can be decomposed into an octave-tree using wavelets, or a wavelet packet tree or a full tree according to its time-frequency tiling. In this section, the performance of these three tree structures for texture classification are compared.

The results are compared using the F-norm, as it is shown to be the best measure in the previous analysis for texture classification. Only a second level decomposition is performed on the data. For the wavelet packet decomposition, in the second level all the bands are decomposed. This results in a uniform decomposition, which is a full tree. The results are compared for the octave tree (wavelet transform) and the full tree for the second level using the F-norm with or without dimensionality reduction. Only 7 features are obtained in the octave tree because it has seven bands in the two level decomposition, whereas the

uniform tree has 16 bands resulting in 16 features. Even when the number of features extracted from each band is increased, there was no improvement in performance, which may indicate that the performance limit has been reached. The classification results are listed in Table 5.11 for the octave and full tree. The results show that the wavelet packet or the full-tree performs better when used for the texture classification.

### 5.7.3. THE RELATIVE PERFORMANCE AMONG THE CRITERIA FOR TREE DECOMPOSITION

Now, we know that the wavelet packet transform performs better than the wavelet transform for texture classification. But, to decompose any image using a wavelet packet tree, one needs to know the criterion with which the bands are decomposed. As we have shown in Chapter 4, the texture can be decomposed based on energy or separability or some other criterion. In this section, the classification performance is compared for wavelet packet decomposition based on energy and separability using an Energy-based tree decomposition algorithm [41] and a separability-based tree decomposition algorithm [43]. The results are compared in Table 5.12 for all wavelets using second level decomposition and the F-norm with and without dimensionality reduction. The feature reduction method used is feature selection, which means that the five best features are selected out of all extracted features. The best features are the highest energy-valued features. Using the feature selection method, the energy based tree decomposition seems to be performing better than the separability-based algorithm. However, when the dimensionality reduction is used instead of the feature selection method based on separability, the separability-based tree decomposition performs better than the other tree. Also, the separability-based dimensionality reduction outperforms the feature selection, so one prefers the separability-based decomposition for texture classification along with the dimensionality reduction.

### 5.7.4. IMPORTANCE OF WAVELETS

One can observe from the tables 5.11 and 5.12 that the choice of the mother wavelet is crucial irrespective of the wavelet tree chosen, the criterion for the tree, the level of decomposition, the feature extraction method, the number of features extracted, or the feature reduction method. One can improve the classification performance by choosing the better, more suitable (sub-optimal) wavelet for the dataset. All

other possible combinations of parameters can be varied as in a typical classification task. The overall classification performance improvement may vary from 1% to 10% depending on the dataset and the parameters chosen to classify the dataset. The classification for individual textures may vary more than that for some textures, which are difficult to classify. It varies less for textures, which are highly clustered and are farther removed from the rest of the classes.

## 5.8. THE NECESSITY TO DESIGN FILTER BANKS

As one can observe from the above results, the performance varies considerably using different wavelets. Classification performance is not the best for either the four-tap Daubechies wavelet or the sixteen-tap Battle-Lemarie wavelet. Improvement in performance for all measures using different tree structures (for both energy-based tree and separability-based tree) can be observed for other wavelets compared to the Daubechies wavelets. The influence of mother wavelet on classification performance is significant in all cases. This is true for all the wavelet tree decomposition algorithms, various feature extracting measures, and sample sizes. The performance improvement is significant, even for the computationally simple algorithm. This work has shown that designing a wavelet for a given application and applying that wavelet to the application improves the performance. The obtained wavelet may not give performance improvement for different sets of data in other applications.

Thus, the original work described in this chapter has established the existence of a wavelet that is both suitable for texture classification and capable of delivering an appreciable performance gain as opposed to the standard wavelets. We want to find this wavelet that gives an improved performance for our application. How do we design this wavelet? What is the criterion that needs to be minimized or maximized to design this wavelet? Is it feasible to design such a wavelet? All these questions are addressed in the next chapter and the optimal or sub-optimal wavelet is designed for texture classification. A suitable criterion for finding the optimal wavelet is presented and is used to find the optimal wavelet.

TABLE-5.10: COMPARISON OF OVERALL CLASSIFICATION PERFORMANCE FOR WT AND WPT

| | No. of Features | Reduced Features | $L_1$-Norm % | F-Norm % | Δ-Norm % |
|---|---|---|---|---|---|
| Wavelet Transform | 7 | 4 | 90.90 | 95.30 | 84.40 |
| Wavelet Packet Transform | 16 | 4 | 95.80 | 98.40 | 92.20 |

TABLE-5.11: OVERALL CLASSIFICATION PERFORMANCE USING F-NORM FOR WAVELET TREES

| | No. of Features | Dimensionality Reduction | Wavelets min. % | max. % | mean % | median % | Daubechies % |
|---|---|---|---|---|---|---|---|
| Octave Tree | 7 | No | 80.50 | 84.10 | 82.38 | 82.15 | 83.50 |
| | 7 | Yes | 87.90 | 95.30 | 93.33 | 93.45 | 95.20 |
| | 28 | No | 80.50 | 83.40 | 82.10 | 81.95 | 83.00 |
| | 28 | Yes | 92.20 | 94.70 | 93.70 | 93.75 | 94.30 |
| Full Tree | | | | | | | |
| | 16 | No | 86.30 | 87.80 | 86.99 | 86.95 | 87.30 |
| | 16 | Yes | 96.30 | 98.40 | 97.34 | 97.30 | 97.30 |
| | 64 | No | 86.00 | 87.00 | 86.55 | 86.55 | 86.50 |
| | 64 | Yes | 96.10 | 97.50 | 96.91 | 96.90 | 96.60 |

TABLE-5.12: OVERALL CLASSIFICATION PERFORMANCE USING F-NORM FOR WAVELET-PACKET TREE BASED ON SOME CRITERIA

| Tree | Feature Reduction | Wavelets min. % | max. % | Daubechies % |
|---|---|---|---|---|
| Energy-Based Tree | Feature selection | 85.57 | 87.86 | 87.14 |
| Separability-Based tree | Feature Selection | 80.60 | 84.10 | 83.50 |
| | Using all features | 86.30 | 87.80 | 87.30 |
| | Dimensionality Reduction | 96.3 | 98.4 | 97.3 |

# CHAPTER 6

# OPTIMAL FILTER BANKS

It was shown in Chapter 5 that the choice of the wavelet basis function is crucial for texture classification. The wavelet needs to be designed based on some criterion yet to be established. The chosen criterion should be dependent on the data sets in terms of their separation and classification. Recently, various wavelets have been designed using different cost functions depending on the problem that needs to be solved [62], [63], [64]. In this chapter, an approach to the design of sub-band filters with optimal separation is proposed. In Section 6.1, the approach is outlined and in Section 6.2, the cost function used to design the filters is presented. The relationship between wavelets and the cost function with respect to the classification performance is established in Section 6.3. Once the cost function is chosen, it needs to be optimized. An appropriate optimization algorithm should be chosen to find the global maximum. In Section 6.4, the optimal sub-band filters are designed and the existing optimization techniques are briefly reviewed. The optimization algorithm (Simulated Annealing) used to find the global optimal solution for the problem is described in Section 6.5. The parameters and their effect on the performance of the simulated annealing algorithm are studied empirically in Section 6.6. This is required for proper convergence of the optimization algorithm. An algorithm to find the optimal sub-band filter for a data set is developed in Section 6.7. The results are discussed in Section 6.8 and our conclusion follows in Section 6.9. Mathematical details in this chapter are presented in Appendix B.

## 6.1. OPTIMAL FILTERS BASED ON SEPARABILITY

It was shown in Chapter 5 that the choice of the wavelet basis function is crucial for signal classification irrespective of the wavelet decomposition methods, levels of decomposition, different measures used for feature extraction, and feature reduction methods. Designing sub-band filters for classifying the sources of similar nature as one group is addressed in this chapter. For good feature extraction, the wavelet must cluster the information within each class and it must provide maximal

discrimination among these clusters. When using the time-frequency plane as the feature space, it is imperative that the representation provide good localization (to prevent any overlap in the information that may provide discrimination) using as few features as possible, to simplify the classification task. This problem can be solved by designing sub-band filter-banks so that the distance between the clusters is maximized. In other words, the sub-band filters need to be designed to cluster the signals that belong to a class and provide maximal separation between the clusters. The criterion chosen for the texture classification problem is discussed in the next section.

## 6.2. SEPARABILITY

When we have two or more classes, feature extraction consists of choosing those features that are most effective for preserving the class separability. Class separability criteria are essentially independent of the coordinate systems. Furthermore, class separability depends not only on the class distributions but also on the classifier implementation. For example, the optimum feature set for a lineal classifier may not be the optimum set for other classifiers (non-linear). In order to avoid this additional complexity, let us assume that we seek the optimum feature set with respect to the Bayes classifier, which results in the minimum error for the given distributions. Then, class separability becomes equivalent to the probability of error due to the Bayes classifier, which is the best one can expect. Therefore, theoretically speaking, the Bayes error is the optimum measure for feature effectiveness. A major disadvantage of the Bayes error as a criterion is the fact that an explicit mathematical expression is not available except in a very few special cases. The criteria to evaluate the effectiveness of features must be a measure of the overlap or class separability among the distributions, and not a measure of fit such as the mean-square error. The Bayes error is the best criterion to evaluate feature sets, and *a posteriori* probability functions are the ideal features. Unfortunately, the Bayes error is too complex, and therefore we need simpler criteria associated with systematic feature extraction algorithms. One such criterion is based on scatter matrices, which is simple and gives systematic feature extraction algorithms. The criteria used measure the class separability of L classes, but do not relate to the Bayes error directly. The class separability is defined in Section 2.4.2.1. Theoretically, class separability is an appropriate criterion for signal classification. Before we proceed with this criterion, let us examine the relationship between the wavelets and class separability for texture classification

## 6.3. RELATIONSHIP BETWEEN WAVELETS AND SEPARABILITY

Using the results obtained in the previous chapter, we can see the relationship between the mother wavelet basis and the class separability. The corresponding class separability should be the highest for the wavelet basis that gives the best classification performance. This simple test should give some direct proportionality between the wavelet basis and the class separability. That means the class separability is the highest using the best wavelet basis and is the least when using the worst wavelet basis. If the results confirm these initial conjectures, then we can design a wavelet basis based on class separability using the scatter matrices.

Figure 6.1 shows the percentage of correct classification rate vs. different wavelet bases and between class separability vs. different wavelet bases. One can easily observe a definitive trend in these two curves. They may not be parallel, or the separability may not be the least at the worst wavelet, but at least it is the highest or near the highest at the best wavelet. Figure 6.2 shows the plots for the classification rate percentage, with-in class separability, between class separability and total separability as a function of the wavelet bases. Classification percentage variation with respect to between-class separability and class separability are shown in Figure 6.3.

The classification rate should be the highest for the wavelet with the highest separability. Though we expect a direct relationship between the classification rate and the total separability, that is observed only with respect to the between-class separability and the classification rate. Consequently, one can design sub-band filter banks by maximizing the between-class separability and achieve improved classification performance. These sub-band filters are designed depending on the dataset, the measure used to extract the features, and also on the classifier used.

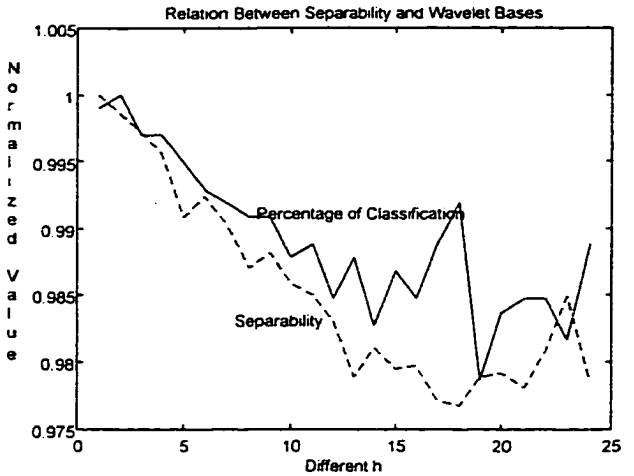



Figure 6.1. Relationship between the percentage of correct classification and between class separability for different wavelet bases.

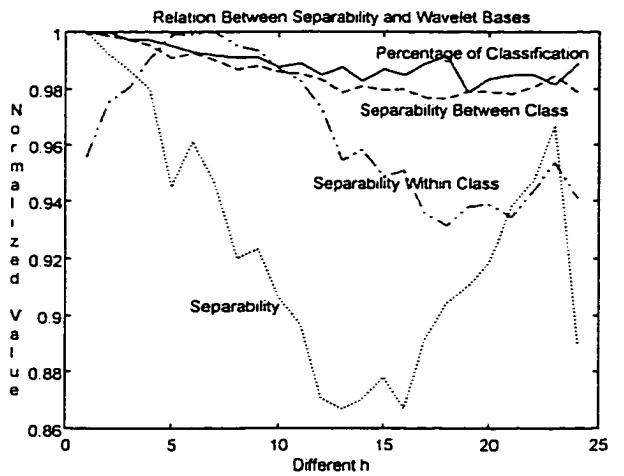



Figure 6.2. Relationship between the percentage of correct classification and class separabilities for different wavelet bases.

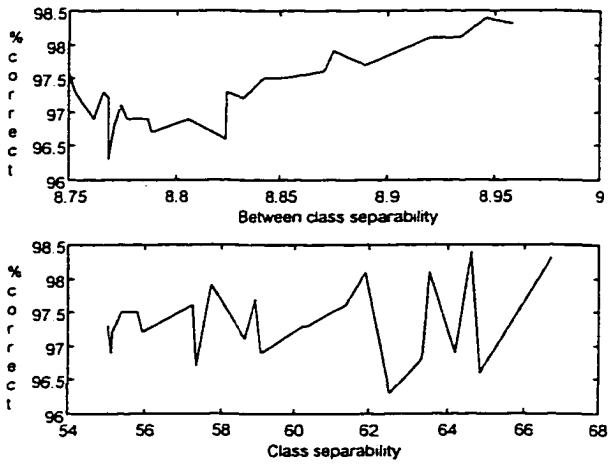



Figure 6.3. The relationship between class separability and % classification rate.

## 6.4. DESIGN OF WAVELETS

One of the main ideas of this study is to investigate the effectiveness of a separability or discrimination based criterion for wavelet basis selection. The analysis compares projections of a set of signals onto waveforms and subsequent selection of the wavelet basis corresponding to the projections that contain the most discriminatory information. This selection permits discrimination of signals to a specified accuracy with the fewest waveforms. The wavelet basis selected based on class separability may not be optimal for representing or approximating individual signals.

In this section, we present our wavelet basis selection scheme, which tries to find the best wavelet basis for classification purposes. The wavelet basis is chosen that gives the best between-class separability. The between class separability is defined as

$$J_b = trace(S_b) \tag{6. 1}$$

where $S_b$ is the between-class scatter matrix (the scatter of the conditional mean vectors $M_i$ of each class around the overall mean vector of all the classes M), i.e.,

$$S_b = \sum_{i=1}^{L} \Pr\{C = C_i\}(M - M_i)(M - M_i)^T \tag{6.2}$$

The sub-band filters need to satisfy the following constraints.

$$\sum_{n=0}^{N-1} h_0(n) = \sqrt{2} \tag{6.3}$$

$$\sum_{n=0}^{N-1} h_i(n) h_j(n+2k) = \delta(i-j)\delta(k) \quad i, j \in \{0,1\} \tag{6.4}$$

Hence, the optimal wavelet solution can be obtained by solving the following constrained optimization problem

$$h_{optimal} = \max_{h}[J_b] \tag{6.5}$$

subject to equations (6.3) and (6.4).

One can observe from Figure 6.3 that the cost function, which is the class separability, is not strictly concave or convex. It in general has a number of local maxima, some of which may lead to bad choices of the sequence $h_o$. It is very difficult to define explicitly the expressions for the gradient of the cost function in terms of the coefficients $h_o$. Before proceeding to other alternatives of optimization, let us try to express the cost function as a function of $h_o$. Please refer to Appendix-B for the approach used to calculate the gradient expressions. A general closed form solution has not been found. This prevents the use of gradient type optimization techniques. However, the major drawback with the gradient search is that it is likely to converge to a local optimum. Also, if the cost function is multimodal within the domain of interest, the number of available algorithms is reduced to very few. One can see from Figure 6.1 that the cost function in the filter domain is multimodal and it has many local maxima.

A simple and widely used technique is to generate a given number of different points inside the function domain, perform unimodal searches starting from each point, and then retain the best result. All of the techniques, including the unimodal direct minimization algorithms such as Hooke and Jeeves [65] and Neldar and Mead [66], and the algorithms which evaluate the derivatives of the cost function [67], are efficient in the case of functions with a few local maxima. However, when the problem has many variables and a large number of local maxima that are an increasing function of the number of variables, these

techniques offer low efficiency and limited reliability. Simulated annealing (SA) has been proposed in the area of combinatorial optimization where the cost function is defined in a discrete domain. This method is reported to perform well in the presence of a very high number of variables [68]. It is based on random evaluations of the cost function in such a way that transitions out of local maxima are possible. It is not guaranteed to converge to the global maximum, but if the function has many good near-optimal solutions, it generally finds one. In particular, this technique is able to discriminate between the "gross behavior" of the function and the finer "wrinkles." First, it iterates in an area of the function where a global maximum should be present, following the gross behavior irrespectively of small local maxima found on the way. It then develops finer details, finding a good, near-optimal local maximum, if not the global maximum itself. The constrained optimization problem posed by (6.5) can be solved by using this SA algorithm.

## 6.5. SIMULATED ANNEALING ALGORITHM

### 6.5.1. METHOD

Let x be a vector in $R^n$ and let $(x_1, x_2, ..., x_n)$ be its components. Let f(x) be the function to maximize and let $a_1 < x_1 < b_1, ..., a_n < x_n < b_n$ be its n variables each ranging in a finite, continuous interval. The function f does not need to be continuous, but it must be bounded. The SA algorithm is schematically shown in Figure 6.3. It proceeds iteratively through the points: $x_0$, $x_1$, ..., $x_i$, ..., tending to the global maximum of the cost function. New candidate points are generated around the current point $x_i$ by applying random moves along each coordinate direction, in turn. The new coordinate values are uniformly distributed in intervals centered around the corresponding coordinate of $x_i$. The half-widths of these intervals along each coordinate are recorded in the step vector v. If the point falls outside the definition domain of f, a new point is randomly generated until a point belonging to the definition domain is found. A candidate point x is accepted or rejected according to the Metropolis criterion [69], which is:

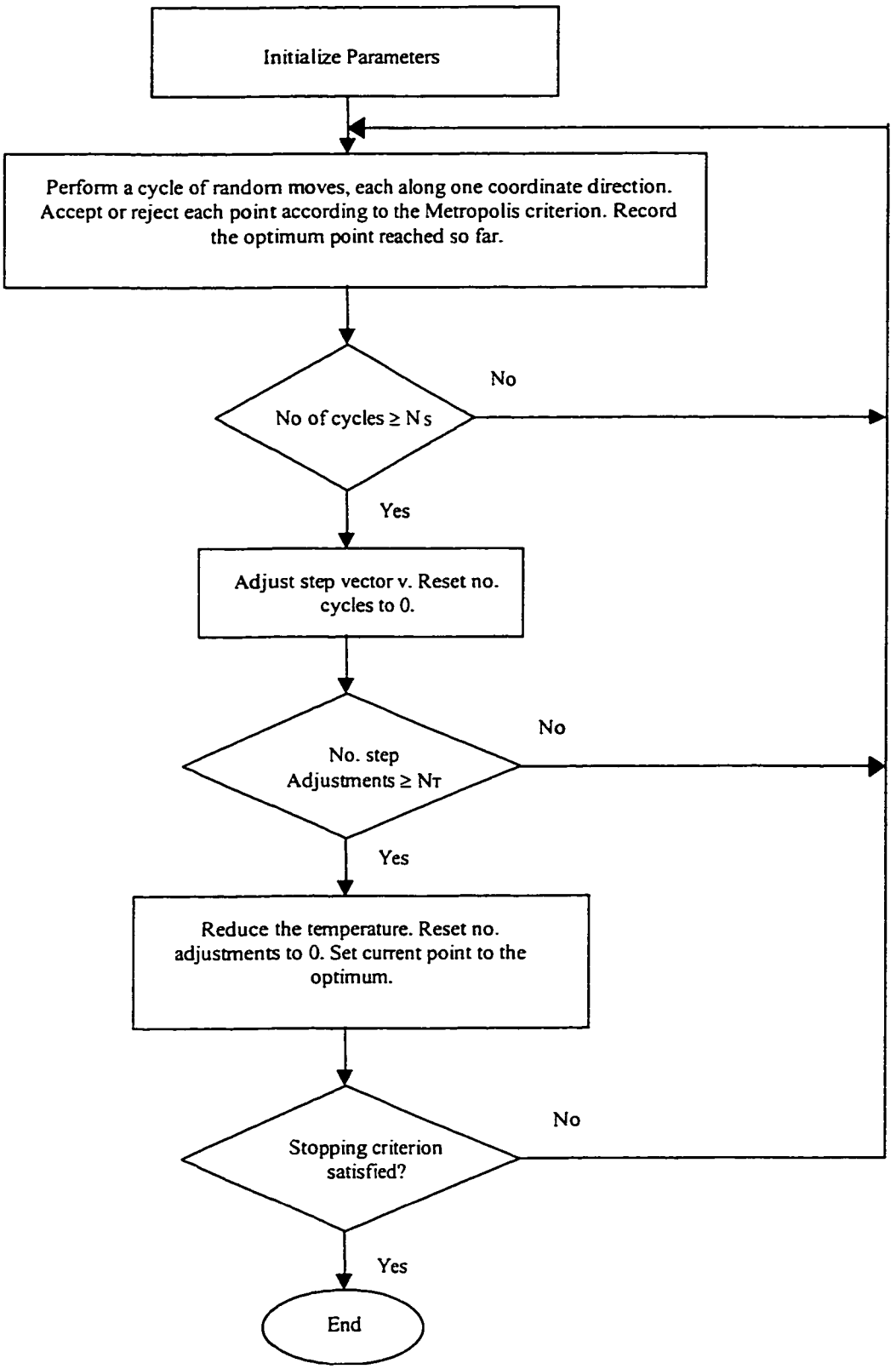



Figure 6.4. The SA minimization algorithm.

If the change in the cost function $\Delta f \leq 0$, then accept the new point: $\mathbf{x}_{i+1} = \mathbf{x}'$; else accept the new point with probability:

$$p(\Delta f) = exp(-\Delta f / T)$$

where $\Delta f = f(\mathbf{x}')-f(\mathbf{x}_i)$ and T is a parameter called temperature.

At any fixed value of T, the succession of points $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_i, ...$ is not downhill except when T=0. For values of T that are large compared to the mean value of $|f(\mathbf{x}_h)-f(\mathbf{x}_k)|$ ( $\mathbf{x}_h$ and $\mathbf{x}_k$ are points randomly chosen inside the definition domain of f) , almost all new points are accepted and the succession is a random sampling of f.

The SA algorithm starts at a high temperature T given by the user. A sequence of points is then generated until a sort of equilibrium is approached; that is, a sequence of points $\mathbf{x}_i$ whose average value of the cost function f reaches a stable value as i increases. During this phase, the step vector $\mathbf{v}_m$ is periodically adjusted to better follow the function behavior. The best point reached is recorded as $\mathbf{x}_{opt}$. After thermal equilibrium, the temperature T is reduced and a new sequence of moves is made starting from $\mathbf{x}_{opt}$, until thermal equilibrium is reached again, and so on. The process is stopped at a temperature low enough that no more useful improvement can be expected, according to a stopping criterion.

The SA optimization algorithm can be considered analogous to the physical process by which a material changes state while minimizing energy. A slow, careful cooling brings the material to a highly ordered, crystalline state of lowest energy. A rapid cooling instead yields defects and glass-like intrusions inside the material.

From an optimization point of view, an iterative search accepting only new points with lowest function values is like rapidly quenching a physical system at zero temperature. It is very likely to be stuck in a metastable, local minimum. On the contrary, SA permits uphill moves under the control of a temperature parameter. At higher temperature, only the gross behavior of the cost function is relevant to the search. As the temperature decreases, finer details can be developed yielding a quality final point. While the optimality of the final point cannot be guaranteed, the method is able to proceed toward better minima even in the presence of many local minima. The detailed algorithm is described in the next section.

6.5.2. ALGORITHM

**Step 0** (Initialization)

Choose:

- A starting point $\mathbf{x}_0$.
- A starting step vector $\mathbf{v}_0$.
- A starting temperature $\mathbf{T}_0$.
- A terminating criterion $\varepsilon$ and a number of successive temperature reductions to test for termination $N_\varepsilon$.
- A test for step variation $N_s$ and a varying criterion **c**.
- A test for temperature reduction $N_T$ and a reduction coefficient $\gamma_T$.

Set i, j, m, k to 0. i is the index denoting successive points, j denotes successive cycles along every direction, m describes successive step adjustments, and k covers successive temperature reductions.

Set h to 1. h is the index denoting the direction along which the trial point is generated, starting from the last accepted point.

Compute $f_0 = f(x_0)$.

Set $\mathbf{x}_{opt} = \mathbf{x}_0$, $f_{opt} = f_0$.

Set $n_u = 0$, u = 1,..., n.

Set $f_u^* = f_0$, u = 0, -1,..., $-N_\varepsilon+1$.

**Step 1**

Starting from the point $\mathbf{x}_i$, generate a random point **x'** along the direction h:

$$\mathbf{x}' = \mathbf{x}_i + r\, v_{mh}\, e_h$$

where r is a random number generated in the range [-1, 1] by a pseudo-random number generator; $e_h$ is a vector of the $h^{th}$ coordinate direction; and $v_{mh}$ is the component of the step vector $v_m$ along the same direction.

**Step 2**

If the $h^{th}$ coordinate of $\mathbf{x}'$ lies outside the definition of f, that is, if $x_h' < a_h$ or $x_h' > b_h$ , then return to step 1.

**Step 3**

Compute $f' = f(\mathbf{x}')$.

If $f' \leq f_i$, then accept the new point:

set $\mathbf{x}_{i+1} = \mathbf{x}'$ ,

set $f_{i+1} = f'$,

add 1 to i,

add 1 to $n_h$,

if $f' < f_{opt}$, then set

$\mathbf{x}_{opt} = \mathbf{x}'$,

$f_{opt} = f'$,

endif;

else ($f' > f_i$) accept or reject the point with acceptance probability p (Metropolis move):

$$p = exp\left(\frac{f_i - f'}{T_k}\right)$$

In practice, a pseudo-random number p' is generated in the range [0, 1] and is compared with p. If $p' < p$, the point is accepted, otherwise rejected.

In the case of acceptance:

set $\mathbf{x}_{i+1} = \mathbf{x}'$,

set $f_{i+1} = f'$,

add 1 to i,

add 1 to $n_h$.

**Step 4**

Add 1 to h.

If $h \leq n$, hen go to step 1.

else set h to 1 and add 1 to j.

**Step 5**

If $j < N_s$, then go to step 1;

else update the step vector $v_m$:

for each direction u the new step vector component $v_u'$ is

$$v_u' = v_{m_u}\left(1 + c_u \frac{n_u / N_s - 0.6}{0.4}\right) \quad \text{if } n_u > 0.6N_s,$$

$$v_u' = \frac{v_{m_u}}{\left(1 + c_u \frac{0.4 - n_u / N_s}{0.4}\right)} \quad \text{if } n_u < 0.4N_s,$$

$$v_u' = v_{m_u} \quad \text{otherwise}$$

Set $v_{m+1} = v'$,

set j to 0,

set $n_u = 0$, u = 1,..., n,

add 1 to m.

The aim of these variations in step length is to maintain the average percentage of accepted moves at about one-half of the total number of moves. The $c_u$ parameter controls the step variation along each $u^{th}$ direction.

**Step 6**

If $m < N_T$, then go to step 1;

else, it is time to reduce the temperature $T_k$:

set $T_k = \Upsilon_T.T_k$, where $\Upsilon_T$ is the reduction coefficient.

set $f_k^* = f_i$,

add 1 to k,

set m to 0.

It is worth noting that a temperature reduction occurs every $N_S.N_T$ cycles of moves along every direction and after $N_T$ step adjustments.

**Step 7** (terminating criterion)

If:

$$\left| f_k^* - f_{k-u}^* \right| \leq \varepsilon, \quad u = 1,..., N_\varepsilon$$

$$f_k^* - f_{opt} \leq \varepsilon$$

then stop the search;

else:

add 1 to i,

set $\mathbf{x}_i = \mathbf{x}_{opt}$,

set $f_i = f_{opt}$.

Go to step 1.

Reasonable values, found after some test optimizations (by Corana et.al.), of the parameters that control the simulated annealing are

$N_S = 20$.

$N_T = \max(100, 5*n)$.

$c_i = 2$, i = 1, ..., n.

$N_\varepsilon = 4$.

$\gamma_T = 0.85$.

### 6.5.3. STEP ADJUSTMENTS

In Monte Carlo simulations of fluids using the Metropolis approach, new configurations are generated trying to maintain a 1:1 rate between the accepted and rejected configurations [70]. A lower rate means that too many moves are rejected, thus wasting computational effort. A higher rate means the trial configurations are too close to the starting ones, thus having a small difference in energy compared to the temperature. This implies that the accepted configurations evolve too slowly, again wasting the

computational effort. The same criterion is used in the SA algorithm. A 1:1 rate between accepted and rejected moves means that the algorithm is following the "function behavior" well.

In the SA algorithm, the trial points are generated along each coordinate direction in turn, independently from the other directions. A step vector **v** records the maximum increments possible along each direction and is adjusted every $N_S$ th move to maintain the 1:1 ratio.

### 6.5.4. COMMENTS

This method needs many function evaluations, but it is able to find the global minimum of test functions with an extremely high number of local minima. SA can provide high reliability in the minimization of multimodal functions at high computational costs that linearly increase with the number of dimensions of the problem. In combinatorial SA, it has been suggested that $T_0$ should be of the same order of magnitude as the standard deviation of the cost function in its domain of definition [71]. A better approach could be to monitor the function behavior as the SA iteration proceeds. This can be done using the incremental ratio between the average value of the cost function and its square at the points accepted by the moves at a given temperature [69]. However, the performance of the algorithm is poor when following multimodal cost functions that have "valleys" that are not directed along the coordinate directions. This problem is due to the way new search points are generated. However, highly directional schemes might lose some of the flexibility of the random search procedure.

## 6.6 INFLUENCE OF PARAMETER SETTINGS ON PERFORMANCE OF SIMULATED ANNEALING ALGORITHM

To get a good solution, the implementation of the simulated annealing algorithm must specify the parameter set for the cooling schedule. In general, the parameter set is specified by a "one variable at a time" strategy in the literature. It is assumed that the parameters are independent of each other and that the interaction effect among parameters may be ignored. Also, the above strategy does not take into account the computational time constraint allowed for getting a good solution. The parameter set under the "one variable at a time" strategy is usually determined by how good a solution is.

To find a better way to obtain a good parameter set, many procedures have been proposed for designing the parameter set [72] - [77]. One of the techniques is to design the parameter set under the computational time constraint via application of the response surface methodology (RSM) [78].

The parameters and their effect on the performance and convergence of the simulated annealing algorithm for our application are studied empirically. The control parameter, which is the starting temperature and its decrement, is made as a function of the standard deviation of the class separability and is used in implementing the algorithm. The reduction of the control parameter is a key indicator of the stability of the annealing algorithm's operation. The control parameter convergence should possess the characteristic large decreases for the initial steps and then exhibit much smaller decrements as the algorithm converges to a solution. This convergence pattern is reminiscent of the temperature convergence used in metal annealing. The control parameter is decremented slowly so that the algorithm does not converge too quickly and to ensure that the iterations do not stick in a metastable local minimum.

Class separability convergence is another indicator of the performance of the algorithm. That means, how well the cost function converges is an indication of the performance of the algorithm. As an example, a test case is used to show the convergence of the cost function. The test case used is evaluation of the cost function, which is class separability. This is calculated for a database size of seven textures using F-norm with single level decomposition and extracting four features. Figure 6.5 shows a representative separability convergence that demonstrates the overall gradual improvement in the class separability. The control parameter $T_0$ and the initial step size used are 0.075 and 0.01, respectively. All the transitions are plotted. The convergence forces the value of the control parameter to approach zero, which then causes even fewer transitions to be accepted according to the Metropolis criterion. The convergence of the cost function is plotted in Figures 6.6 and 6.7 with different parameter settings. Figure 6.6 shows the convergence plot with $T_0$=0.75 and step size of 0.001. As we increase the initial temperature, the number of iterations needed for the cost function to converge to the global maximum is high when compared to a lower initial temperature. In addition, the number of iterations also increases with a decrease in the initial step size. This property can be observed by comparing Figures 6.6 and 6.7, which cases both have the same initial temperature but different initial step sizes.

To study the influence of varying the temperature linearly, the temperature is reduced linearly at step sizes of 0.1 from an initial temperature of 0.75. In another case, the temperature is kept constant through out the annealing process (i.e. all the random moves and the step changes) with an initial temperature of 0.75. These plots are shown in Figures 6.8 and 6.9. It can be observed from Figures 6.6 (exponential decrease in temperature) and 6.8 (constant temperature) that both the cases have the same number of iterations (approximately 1400 iterations). The cost function varies only between 47.00 and 49.00 when the temperature is reduced exponentially (Figure 6.6), where as the cost function varies between 45.00 and 49.00 when the temperature is kept constant (Figure 6.8). Both these experiments are conducted with the same initial temperature and same initial step size. The highest cost function reached is 49.00 in both cases. However, the test with exponential temperature reduction reaches the 'equilibrium' for each Ns iterations before a change in the step size is attempted. The test with the same temperature reaches the maximum value of the cost function only a few times and does not reach 'equilibrium.' The importance of reaching the maximum of the cost function is explained in the Section 6.8. The same observations can be made from Figure 6.9, which was produced with a linear temperature reduction schedule.

The next step in the study of the performance of annealing is an investigation of the reliability of the results. It has been shown that annealing has the capability of yielding impressive results. Simulated annealing, however, is a stochastic process, which implies that the output of such a system must be a random variable. The outcome of each trial corresponds to a realization of this random variable.

A figure of merit (FOM) can be defined for the final states of each annealing trial in order to verify the functionality and reliability of the annealing algorithm [79]. The FOM is defined as the ratio of the standard deviation to the mean:

$$FOM = \frac{\sigma}{\eta}.$$

Smaller values of FOM imply that the spread of the data is insignificant in comparison to the mean. The figure of merit for the final states of the trail is small and implies that the algorithm is able to converge to approximately the same value for every run. The standard deviation for the trails is 0.8893 and the mean is 47.4066, which yields FOM equal to 0.0188. This FOM is approximately equal to one fifty-third, which

is analogous to a signal to noise ratio of 53:1. A system demonstrating a signal to noise ratio of 10:1 is considered to offer an acceptable performance for most applications.

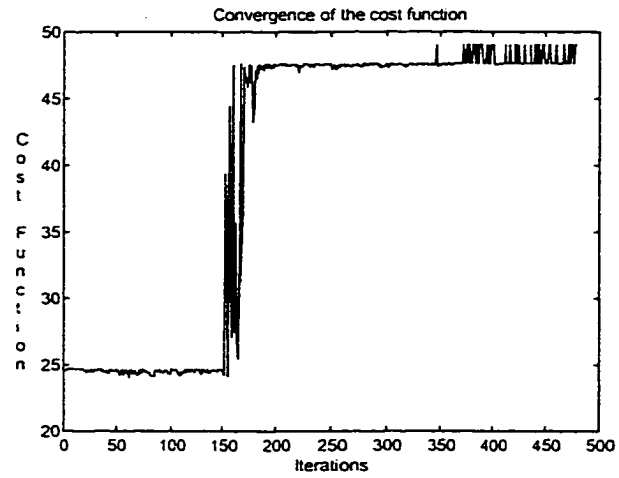



Figure 6.5. The convergence of the class separability with initial temperature 0.075 and with an initial step size of 0.01

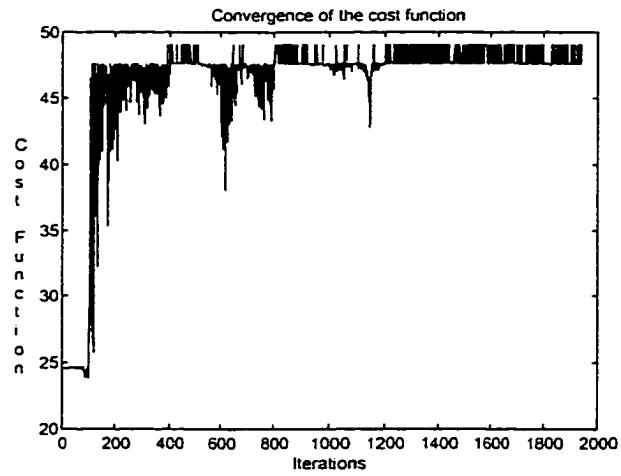



Figure 6.6. The convergence of the class separability with initial temperature 0.75 and with an initial step size of 0.001

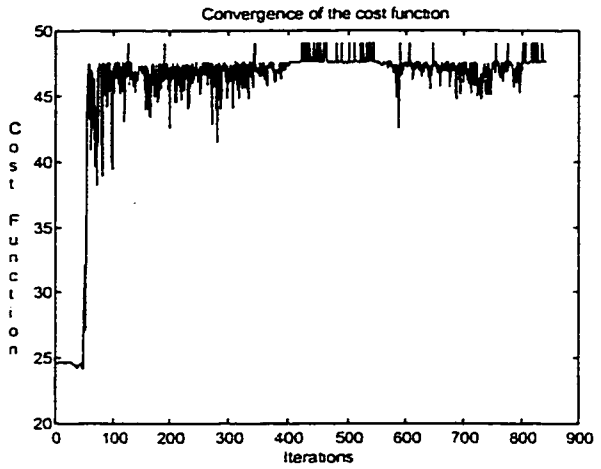



Figure 6.7. The convergence of the class separability with initial temperature 0.75 and with an initial step size of 0.01

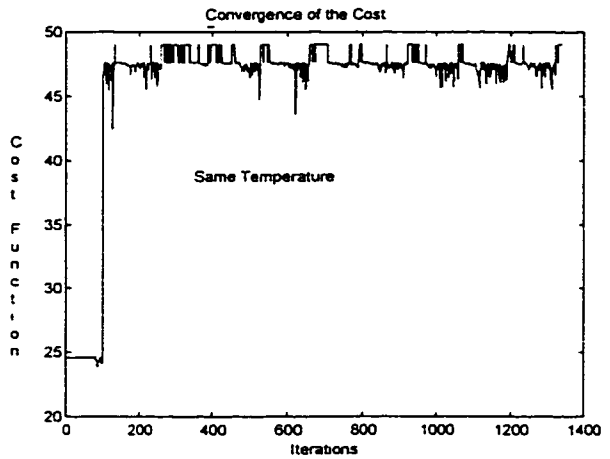



Figure 6.8. The convergence of the class separability with initial temperature of 0.75 and with an initial step size of 0.001 and the temperature is kept the same.

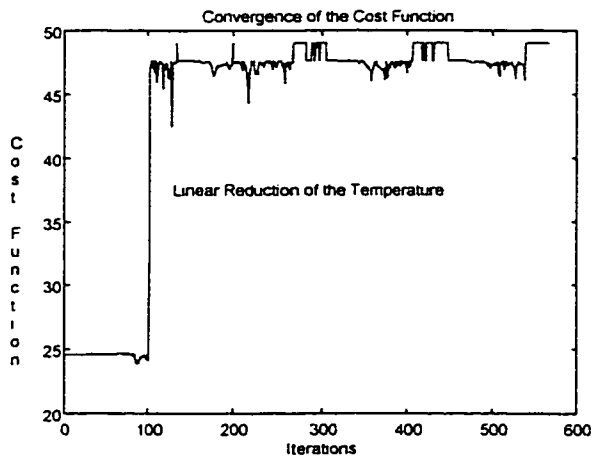



Figure 6.9. The convergence of the class separability with initial temperature of 0.75 and with an initial step size of 0.001 and the temperature is reduced linearly in steps of 0.1

## 6.7. BEST WAVELET BASES FOR DISCRIMINATION

Now, we have an optimization algorithm, which can find the globally optimal or sub-optimal wavelet. In this section, we present an algorithm to design the best wavelet basis for classification purposes.

**Algorithm**

1. Find an optimal filter-bank which maximizes the between class separability.
   1.1. Start with some initial guess for the wavelet filter-bank.
   1.2. Using SA algorithm, the between class separability is maximized. This is done according to the following steps:
      1.2.1. The required number of samples is obtained from each texture for training and classification.
      1.2.2. Using the initial guess of the wavelet, the DWT coefficients are calculated for all the texture samples in the database.
      1.2.3. The features are extracted using one of the measures, either the $L_1$-norm, F-norm, or $\Delta$–norm.
      1.2.4. The features can be reduced dimensionally by either separability-based dimensionality reduction or feature selection methods.
      1.2.5. Using these features, the separability is calculated and used to find the optimal wavelet.
2. For the unknown texture to be classified into one of the texture classes in the database, calculate the DWT coefficients of the unknown texture using the optimal wavelet.
3. Extract the features using the appropriate measure (i.e., consistent).
4. Using these features, classify the unknown texture.

When a new texture that is not in the database needs to be classified, the texture needs to be trained using the above algorithm and a new optimal wavelet is obtained for the updated database. This algorithm is independent of the wavelet tree decomposition method used to obtain the DWT coefficients, the feature extraction methods, and dimensionality reduction methods. Any of the tree decomposition methods discussed in Chapter 4 can be combined with this optimal filter-bank design algorithm. The idea of

separability-based optimal filter design can be applied regardless of the criteria used for basis selection, e.g., it can be used on the pyramid wavelet transform, balanced or unbalanced wavelet packet trees.

## 6.8. RESULTS

Using SA, the class separability is maximized for a set of textures and the global maximum is obtained. The optimization algorithm is repeated with different starting guesses for the filter coefficients for the same set of textures with the same parameters and the algorithm always converged to the same set of filter coefficients or very close to this set of coefficients. By "close", we mean that the filter coefficients are nearly identical with respect to their frequency responses and corresponding cost functions (class separability). The classification performances for the optimal filter, Daubechies filter and earlier sub-optimal filter are listed in Table 6.1. This is done for one level of decomposition using four features by extracting the features using the F-norm and without any dimensionality reduction techniques. The performance is the highest at the optimal wavelet, which is 90.86. The performance is 86.29 for Daubechies wavelet and the sub-optimal found using random search has 90.43. A considerable improvement in classification performance is achieved using the optimal wavelet compared to Daubechies wavelet. Even with the sub-optimal wavelet, the performance improvement is significant.

The classification performance using dimensionality reduction is 98.86 for the optimal wavelet, whose separability is 49.00 and the wavelet with separability 47.00 has classification performance of 94.57. This means that reaching the separability at 49.00 is very important. When the near global maximum is reached, even a small change in the cost function may yield a significant difference in the classification performance. This is shown in Table 6.2, and is due to the non-linear relationship between the class-separability and classification. Hence, it is required to use the SA algorithm to find the global maximum in our case, due to large number of local maxima. Using more iterations with exponential temperature reduction, the cost function can reach the global maximum at 49.00. The simulation with the same temperature may not yield this convergence.

The classification performance for the optimal wavelet is listed in Table 6.3 for levels 1 and 2 using the $L_1$-norm, F-norm, and $\Delta$-norm measures with feature reduction methods. The performance is the highest

using the F-norm with second level of decomposition using 16 features. A comparable performance is achieved using the $L_1$-norm and F-norm. However, the $\Delta$-norm did not perform well in all the cases. The dimensionality reduction is very important for better classification, as it brings the most relevant information from all the features into only a few features. The performance improvement using dimensionality reduction is significant compared to either using all the features or the feature selection method.

To know the influence of database size on the classification performance using the optimal wavelet, the database size is varied from 2-10 textures. That is, the optimal wavelet is found for varying databases and the corresponding classification performance is calculated. These are listed in Table 6.4 for one level using the F-norm. The same is done for two level decomposition using the F-norm and listed in Table 6.5. The performance as high as 100% is achieved when database has only two classes of textures. Even when the database size increased, the performance is consistently close to 98%-99%.

To illustrate what the optimal filters may look like, the frequency and phase responses of the optimal filter are compared with the responses of the four-tap Daubechies filter in Figure 6.10. It can be observed that the transition band characteristics for the optimal sub-band filter are different from that of the Daubechies filter.

It can be observed that, in various cases of texture classification, the optimal filter obtained is either very close to the Haar filter or a delayed Haar filter. The class separability and the corresponding classification performance achieved are very close in some cases and significantly different (optimal filter gave better classification performance than the Haar filter) in others for the Haar filter and the optimal filter. The work presented in this dissertation shows the superior performance of the Haar filterbank with respect to texture classification. Although some work has been done for using the Haar in image compression problems, its full potential in feature extraction and image analysis problems has not been determined [80]. A complete study needs to be performed on the properties of the Haar wavelet and how its properties affect the classification performance.

TABLE-6.1: THE CLASSIFICATION PERFORMANCE USING THE OPTIMAL WAVELET

| *Wavelet* | *Classification Percentage* |
|---|---|
| Optimal | 90.8571 |
| Daubechies | 86.29 |
| Previous observed best performance from among the 24 sampled wavelets | 90.43 |

TABLE-6.2: THE CLASSIFICATION PERFORMANCE USING THE OPTIMAL WAVELET WITH DIMENSIONALITY REDUCTION

| *Wavelet* | *Classification Percentage* |
|---|---|
| Optimal Wavelet with Cost function = 49.00 | 98.86 |
| Optimal Wavelet with Cost function = 47.00 | 94.57 |

TABLE-6.3: CLASSIFICATION PERFORMANCE OF THE OPTIMAL FILTER FOUND USING SA

AF - All Features; DR - Dimensionality Reduction; FS - Feature Selection

| No. of levels | $L_t$-norm (%) | | | F-norm (%) | | | Δ-norm (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | AF | DR | FS | AF | DR | FS | AF | DR | FS |
| 1 | 66.14 | 93.00 | 56.43 | 90.86 | 98.86 | 78.43 | 82.14 | 83.29 | 62.86 |
| 2 | 78.86 | 99.29 | 75.00 | 91.14 | 99.43 | 86.00 | 94.14 | 96.29 | 86.86 |

TABLE-6.4: CLASSIFICATION PERFORMANCE OF THE OPTIMAL FILTER FOR VARIOUS NUMBER OF TEXTURES IN THE DATABASE FOR LEVEL 1

| No. of textures in the database | % of overall classification |
|---|---|
| 2 | 100 |
| 4 | 96.5 |
| 7 | 98.86 |
| 10 | 91.4 |

TABLE-6.5: CLASSIFICATION PERFORMANCE OF THE OPTIMAL FILTER FOR VARIOUS NUMBER OF TEXTURES IN THE DATABASE FOR LEVEL 2

| No. of textures in the database | % of overall classification |
|---|---|
| 2 | 100 |
| 3 | 98.67 |
| 4 | 99.00 |
| 5 | 99.20 |
| 6 | 98.67 |
| 7 | 99.43 |
| 8 | 97.88 |
| 9 | 98.22 |
| 10 | 98.00 |

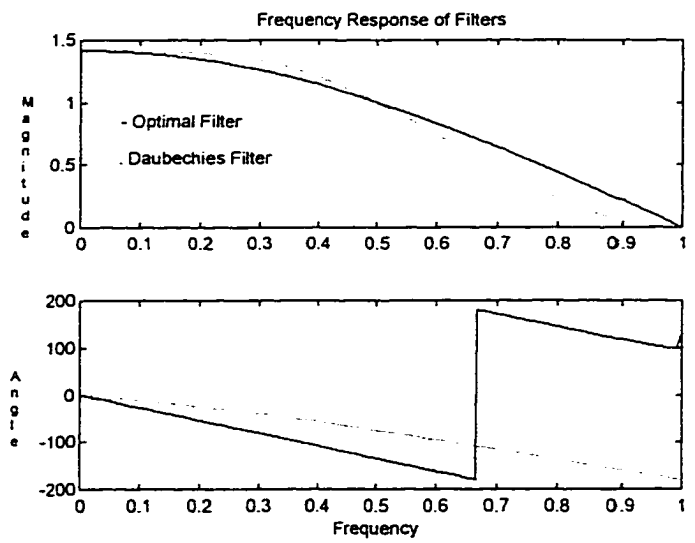



Figure 6.10. Frequency and Phase response of optimal filter and Daubechies filter.

## 6.9. CONCLUSION

Optimal sub-band filters are designed for textures with increased classification rates. The relationship between the classification rates and the class separabilities are shown. The advantage of having the class separability as the cost function is that it takes care of both the within-class and between-class scatter. The features extracted using the designed optimal sub-band filters based on the class-separability have all the information required for classification. A single optimal filter is designed for a set of textures. The set may have any number of classes. The classification performance goes down with an increase in the number of classes in a data set. The classification performances are compared for different tree structures (Octave tree, Energy based tree, and Separability based tree). In addition, the impact of extracting differing numbers of features is studied, along with variation of the feature extractor. Also, the effect of decomposing the optimal wavelet into more levels is studied and compared. The dimensionality reduction plays a major role in obtaining higher classification performance. The feature selection technique did not perform as well as the dimensionality reduction technique.

The simulated annealing algorithm is ideally suited for finding the optimal sub-band filters for classification. The simulated annealing algorithm yields maximal global class separability. The parameter settings needed in the simulated annealing algorithm are empirically chosen for the application. These parameters yield good global performance with an optimum trade off between the number of iterations required to reach the global maximum and the final stopping error. The convergence characteristics of both the class separability and the control parameter confirm the stability of the algorithm.

# CHAPTER 7

# AN ALGORITHM FOR OPTIMAL SUB-BAND FILTER DESIGN FOR SIGNAL CLASSIFICATION

This chapter provides the algorithm for optimal filter design for signal classification in detail. It elaborates the algorithm presented in Section 6.7, without involving detailed results and discussions. Given a data set, all the steps involved in the process to obtain the classification results are provided.

This is a generalized algorithm, which works for both texture classification and fault identification problems. Hence it has a wide range of applications in any signal classification, identification and detection problem. We present a fault detection and identification algorithm that automatically processes an unknown image by locating and identifying the faulty component, which is the same algorithm for classifying the textures. The heart of the algorithm is finding the optimal sub-band filters for signal classification and fault detection. These optimal filters are adapted to both the data and the pattern recognition problem. For identification or classification, the filters find the features that differentiate among the signals (e.g., textures, components). The filters are designed through a simultaneous decomposition of a training set into a two-dimensional (2-D) wavelet expansion. This yields a representation that is explicitly 2-D and encodes information locally. The design is based on class separability of the features extracted in the wavelet domain. The identification module searches the database for the identity of the unknown signal using the optimal filters to make the identification. The algorithm is demonstrated on two sets of images. The first set is textures from the Brodatz texture album. The second set contains infrared images of printed circuit boards. The algorithm follows in the next section.

## 7.1. BEST WAVELET BASES FOR DISCRIMINATION

In this section, we present an algorithm to find the best wavelet basis for classification purposes. It has two phases. First, the optimal wavelet is designed based on the given data set for the problem at hand. Once the training is done to find the optimal wavelet based on class separability, the next phase is classification or fault identification of the unknown signal (texture or component) using the optimal wavelet.

### 7.1.1. ALGORITHM FOR DESIGNING OPTIMAL FILTERS

1. The required number of samples is obtained from each class (texture or component) for training and classification.
2. Start with some initial guess of the wavelet filter-bank.
3. Using the initial guess of the wavelet, the DWT coefficients are calculated for all the samples in each class and for all the classes in the database. The tree structure and the depth of the decomposition can be chosen according to complexity and time constraints.
4. The features from the DWT coefficients for all samples are extracted using one of the measures ($L_t$-norm or F-norm or $\Delta$–norm). The required number of features is extracted.
5. These features can be reduced dimensionally by either separability-based dimensionality reduction or feature selection methods before they are used to calculate the class separability.
6. Using these features, the separability is calculated. The separability is the cost function used to find the optimal wavelet. The simulated annealing algorithm is used to find the wavelet with the maximum class separability. The wavelet with the maximum class separability is the optimal wavelet for signal classification.

This algorithm is shown in flowchart form in Figure 7.1.

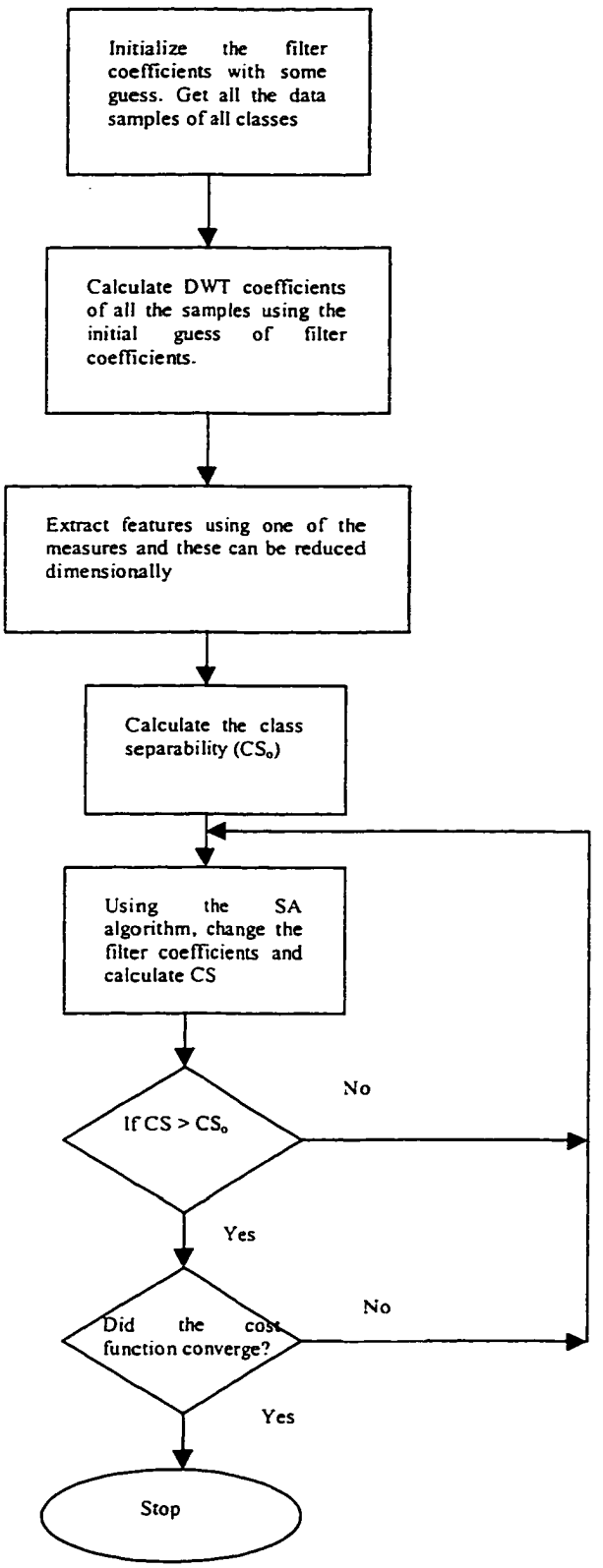



Figure 7.1. Flow chart of the algorithm to find the optimal wavelet using SA algorithm.

### 7.1.2. ALGORITHM FOR CLASSIFICATION OR FAULT DETECTION

1. Find an optimal filter-bank that maximizes the class separability.
2. For the unknown texture to be classified into one of the texture classes in the database, calculate the DWT coefficients of the unknown texture using the optimal wavelet. Use the same tree structure and the same number of levels of decomposition as in the training phase.
3. Extract the features using the same measure used to obtain the optimal wavelet. Also, use the same number of features and the same dimensionality reduction techniques as in the training phase.
4. These features are used to classify the unknown signal (texture or component). The distance or separation between the features corresponding to the unknown signal and the representative feature vector of each class in the database is calculated. The unknown signal belongs to the class with the least distance or separation.

When a new signal (texture or component), which is not in the database needs to be classified, the signal needs to be trained using the above algorithm and a new optimal wavelet is obtained for the updated database. This algorithm is independent of the wavelet tree decomposition method used to obtain the DWT coefficients, the feature extraction methods, and dimensionality reduction methods. All of the tree decomposition methods discussed in Chapter 4 can be combined with this optimal filter-bank design algorithm. The idea of a separability-based optimal filter design can be applied regardless of the criteria used for the basis selection, e.g. it can be used on the pyramid wavelet transform, balanced or unbalanced wavelet packet trees.

# CHAPTER 8

# SUMMARY AND CONCLUSIONS

This dissertation presents several new design algorithms for subband filterbank optimization. The main goal is two fold: (*i*) Develop design techniques that give better understanding of the classification and fault identification tasks and the cost functions associated with them. (*ii*) Design the optimal subband filterbanks based on the cost functions.

## 8.1. SUMMARY

A new class of filterbanks, QMF filter banks, is proposed for texture classification. Various local-basis selection algorithms such as energy-based tree decomposition, separability-based tree decomposition are reviewed and analyzed. The performances of these algorithms are also compared with the standard octave tree decomposition. The classification problem may be divided into the stages of feature extraction, dimensionality reduction and pattern recognition. Central to this work is that the signal representation plays a crucial role in classification performance.

The influence of subband filterbanks on classification performance is studied. Also, various parameters involved in the classification system are studied. This study is conducted on several types of wavelet tree structures, including the wavelet transform (octave tree), the uniform tree and the wavelet packet transform based on some criteria (e.g., energy, separability). These factors include feature extracting measures (e.g., $L_1$-norm, F-norm, $\Delta$-norm etc.), dimensionality reduction techniques (feature selection, feature projection) and the types of classifiers (Euclidean distance, simplified Mahalanobis distance, neural network) on classification system performance.

It is proposed that designing an optimal sub-band filterbank for fault detection and identification or texture classification improves the classification performance when the filterbank is used for that purpose. The optimal filterbank should be designed based on class discrimination rather than energy compaction.

The relationship between classification rate and class separability is established. Simulated annealing (SA) is used to find the optimal basis by maximizing the class separability and satisfying the QMF constraints, as there are many minima and maxima in the class separability with respect to sub-band filters. An algorithm is developed to find an optimal wavelet basis given the data samples for a classification task.

An important application to the signal classification, fault detection and identification system is presented. The approach of designing the optimal sub-band filter based on class separability is similar, except that the data involved is different. The performance is measured differently from the texture classification problem, which is either the number of times the faulty component is detected correctly or the separation between the faulty components in a bad board from the functioning ones. Influence of the optimal sub-band filter on FDI system performance is studied along with the influence of feature extraction, dimensionality reduction, and various wavelet tree structures on FDI system performance.

The performances of the proposed methods are shown in extensive experiments. The results clearly justify the new approaches.

## 8.2. MAJOR CONTRIBUTIONS OF THE WORK

- The influences of various factors involved in classification are studied. These factors include feature-extracting measures (e.g., $L_1$-norm, F-norm, $\Delta$-norm etc.), dimensionality reduction techniques (feature selection, feature projection) and the types of classifiers (Euclidean distance, simplified Mahalanobis distance, neural network) on classification system performance.
- A computationally simple algorithm for classification is developed.
- The classification performance for various tree structures (WT, WPT etc.) with respect to feature extracting measures, dimensionality reduction techniques and types of classifiers is compared.
- The influence of sub-band filters on texture classification performance is studied.
- The classification performance for various tree structures (WT, WPT etc.) with respect to various sub-band filter candidates is compared.
- An empirical relationship between the classification performance and the class separability is established.

- Optimal sub-band filters based on class separability for classification using simulated annealing are designed.
- The various parameters (e.g., step size, initial temperature, temperature reduction etc.) involved in the SA for convergence are studied.
- An algorithm using the above mentioned design technique from the given data sets for classification is developed.
- Influence of the optimal sub-band filter on FDI system performance is studied.
- Influence of feature extraction, dimensionality reduction, and various wavelet tree structures on FDI system performance is studied.

## 8.3. SUGGESTIONS FOR FURTHER RESEARCH

- A few attempts at filter optimization with respect to the class separability criterion are made in Appendix-B. An iterative or closed form solution should be targeted. Furthermore, approaches based on the alternate expressions and derivations should be examined further.
- The ultimate criterion in texture classification is the classification error rate. All optimization approaches in this dissertation have been with respect to criteria that are only indirectly related to the error rate. The optimal solutions are consequently optimal with respect to the error rate. Further effort should be put on minimum error optimization.
- Similarly, in fault detection and identification, the goal is to identify the faulty component correctly with better accuracy. However, having a better understanding of the circuit board and the design involved along with the functionality of the circuit components helps improve the system performance. A neural network can be better trained with this information for better fault identification. More effort needs to be applied with respect to incorporating the printed circuit board information into the design and fault identification.
- In various cases of texture classification, the optimal filter obtained is either the Haar filter or a delayed Haar filter. The work presented in this dissertation shows the superior performance of the Haar filterbank with respect to texture classification. We only attempt to understand why the Haar basis is

better than all other wavelet basis system for the given application. A complete study needs to be performed on the properties of the Haar wavelet and how its properties affect performance.

Addressing these types of questions using the available design techniques will possibly generate insight and help determine how the wavelet basis should be chosen for a particular application in future.

# REFERENCES


[1] A. R. Rao and B. G. Schunck, "Computing oriented texture fields," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 2, Mar. 1991, pp. 157-185.

[2] T. O. Gulsrud and S. Kjode, "Optimal filter for detection of stellate lesions and circumscribed masses in mammograms," *Proc. SPIE Visual Communications and Image Processing*, Orlando, Florida, Mar. 1996, pp. 430-440.

[3] J. S. Weszka, C. R. Dyer, and A. Rosenfield, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cyb.*, vol. 6, no. 4, Apr. 1976, pp. 269-285.

[4] N. Lamei, K. D. Hutchinson, M. M. Crawford, and N. Khazenie, "Cloud-type discrimination via multi-spectral textural analysis," *Optical Engineering*, vol. 33, no. 4, Apr. 1994, pp. 1303-1313.

[5] I. Pitas and C. Kotropoulos, "A texture-based approach to the segmentation of seismic images," *Pattern Recognition*, vol. 25, no. 9, 1992, pp. 929-945.

[6] S.-T. Bow, *Pattern Recognition: Applications to Large Data-set Problems*, Marcel Dekker Inc., New York, 1984.

[7] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Second Edition, Academic Press, 1990.

[8] A. N. Akansu and M. J. T. Smith, *Sub-band and Wavelet Transforms: Design and Applications*, Kluwer Academic Publications, 1996.

[9] R. M. Haralick, "Statistical and Structural Approaches to Texture", *Proc. IEEE*, vol. 67, May 1979, pp. 786-809.

[10] J. C. Bezdek, and S. K. Pal, *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*, IEEE press, New York, 1992.

[11] B. Kosko, *Neural Networks and Fuzzy Systems*, vol. 1, Prentice Hall, NJ, 1992.

[12] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1965.

[13] R. R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, April 1987, pp. 4-22.

[14] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural Networks*, vol. 1, no. 4, 1988, pp. 295-308.

[15] S. Y. Kung, K. I. Diamantras and J. S. Taur, "Adaptive Principal Component EXtraction (APEX) and Applications," *IEEE Trans. on Signal Processing*, vol. 42, no. 5, May 1994, pp. 1202-1216.

[16] M. Vetterli and J. Kovačević, *Wavelets and Sub-band Coding*, Prentice Hall, NJ, 1995.

[17] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. on Information Theory*, vol. 36, no. 5, Sept. 1990, pp. 961-1005.

[18] D. Gabor, "Theory of communications," *Journal. Inst. Elec. Engg.*, vol. 93, 1946, pp. 429-457.

[19] C. Herley, J. Kovačević, K. Ramchandran and M. Vetterli, "Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, Dec. 1993, pp. 3341-3359.

[20] S. Mallat, "A theory of multiscale signal decomposition: The wavelet representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, 1989, pp. 674-693.

[21] C. S. Burrus, R. A Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms*, Prentice Hall, 1996.

[22] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Applied Math.*, vol. XLI, no. 41, Nov. 1988, pp. 90-96.

[23] W. M. Lawton, "Necessary and sufficient conditions for constructing orthonormal wavelet bases," *Journal of Math. Physics*, vol. 32, no. 1, Jan. 1991, pp. 57-61.

[24] P. P. Vaidyanathan, *Multirate systems and filter banks*, Prentice Hall, Englewood Cliffs, NJ 1992.

[25] I. Daubechies, *Ten lectures on wavelets*, CBMS-NSF conference on wavelets and applications, SIAM, Philadelphia, PA, 1992.

[26] Y. Zhuang and J. S. Baras, "Image Compression using optimal wavelet basis," *Proc. of SPIE*, Orlando, FL, Apr. 1995, pp. 131-140.

[27] Y. Zhuang and J. S. Baras, "Optimal wavelet basis selection for signal representation," *Proc. of SPIE: Wavelet applications*, Orlando, FL, Apr. 1994, pp. 200-211.

[28] J. Tuqan and P. P. Vaidyanathan, "Globally optimal two channel FIR orthonormal filter banks adapted to the input signal statistics," *Proc. IEEE Int. Conf. Accoust. Speech Signal Process., ICASSP*, Seattle, WA, May 1998, pp. 1353-1356.

[29] R. A. Haddad, A. N. Akansu and A. Benyassine, "Time-frequency localization in transforms, subbands, and wavelets: A critical review," *Optical Engineering*, vol. 32, no. 7, July 1993, pp. 1411-1428.

[30] R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. on Information Theory*, vol. 38, no. 2, Mar. 1992, pp. 713-718.

[31] C. Taswell , "Near-best basis selection algorithms with non-additive information cost functions," *Proc. of IEEE SP Int. Symp. on time-frequency and time-scale analysis*, Philadelphia, PA, Oct. 1994 , pp. 13-16.

[32] K. Kreutz-Delgado and B. D. Rao, "Measures and algorithms for best basis selection," *Proc. IEEE Int. Conf. Accoust. Speech Signal Process., ICASSP*, Seattle, WA, May 1998, pp. 1881-1884.

[33] N. Saito and R. R. Coifman, "Local discriminant bases and their applications," *J. of Math. Imaging and Vision*, vol. 5, no. 4, 1995, pp. 337-358.

[34] J.-M. Boucher, G. Benie, R. Fau and S. Plehiers, "Local and Global multiscale image classification," *Proc. of SPIE*, Orlando, FL, Apr. 1994, pp. 485-493.

[35] H. H. Szu and S. Kadambe, "Neural network adaptive wavelets for signal representation and classification," *Optical Engineering*, vol. 31, no. 9, 1992, pp. 1907-1912.

[36] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, 1993, pp. 1186-1191.

[37] P. Schumacher and J. Zhang, "Texture classification using neural networks and discrete wavelet transform," *Proc. of Int. Conf. Image Process., ICIP*, Austin, TX, Nov. 1994, pp. 903-907.

[38] S. M. LoPresto, K. Ramachandran and M. Orchard, "Wavelet image coding using rate-distortion optimized backward adaptive classification," *Proc. of SPIE* , San Jose, CA, Feb. 1997, pp. 1026-1037.

[39] R. Singh, R. Vasquez and R. Singh, "Wavelet transform and SGLDM: A classification performance study using ML parameter estimation, minimum distance and k-nearest neighbor classifiers," *Proc. of SPIE*, Orlando, FL, Apr. 1997, pp. 142-150.

[40] K. Ramachandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. on Image Processing*, vol. 2, no. 2, Apr. 1993, pp. 160-175.

[41] T. Chang and C.-C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. on Image Processing*, vol. 2, no. 4, Oct. 1993, pp. 429-441.

[42] T. Chang and C.-C. J. Kuo, "Texture classification with tree-structured wavelet transform," *International Conf. on Pattern Recognition*, Aug. 1992, pp. 256-259.

[43] K. Etemad and R. Chellappa, "Separability-based multiscale basis selection and feature extraction for signal and image classification," *IEEE Trans. on Image Processing*, vol. 7, no. 10, Oct. 1998, pp. 1453-1464.

[44] K. Etemad and R. Chellappa, "Separability based tree structured local basis selection for texture classification," *Proc. of Int. Conf. Image Processing, ICIP*, Austin, TX, Nov. 1994, pp. 441-445.

[45] R. Porter and N. Canagarajah, "Robust rotation-invariant texture classification: Wavelet, Gabor filter and GMRF based schemes," *IEE Proc. Vis. Image Signal Processing*, vol. 144, no. 3, June 1997, pp. 180-188.

[46] P. Lau and N. P. Papanikolopoulos, "A Variance-based basis selection scheme for the Gabor image transformation," *Proc. IEEE Int. Conf. Accoust. Speech Signal Process., ICASSP*, Detroit, MI, May 1995, pp. 2205-2208.

[47] M. Porat and Y. Y. Zeevi, " The generalized Gabor scheme of image representation in biological and machine vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, July 1998, pp. 452-467.

[48] P. Brodatz, *Textures: A Photographic Album for Artists & Designers*, New York: Dover, New York, 1966.

[49] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore, MD, 1983.

[50] K. Etemad and R. Chellappa, "Dimensionality reduction of multiscale feature spaces using a separability criterion," *Proc. IEEE Int. Conf. Accoust. Speech Signal Process., ICASSP*, Detroit, MI, May 1995, pp. 2547-2550.

[51] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, July 1989, pp. 717-728.

[52] J.-S. Smokelin, "Wavelet feature extraction for image pattern recognition," *Proc. of SPIE*, 1996, pp. 110-121.

[53] H. M. Hajj, T. Q. Ngyuen and R. T. Chin, "On multi-scale feature detection using filter banks," *Proc. of Asilomar Conf. Signal Syst. Computers*, Pacific Grove, Oct. 1995, pp. 70-74.

[54] R. Strickland and H. Hee, "Wavelet transform methods for object detection and recovery," *IEEE Trans. on Image Processing*, vol. 6, no. 5, May 1997, pp. 724-735.

[55] A. Tewfik, D. Sinha and P. Jorgensen, "On the optimal wavelet choice of a wavelet for signal representation," *IEEE Trans. on Information Theory*, vol. 38, no. 2, Mar. 1992, pp. 747-764.

[56] A. Averbuch, D. Lazar and M. Israeli, "Image compression using wavelet transform and multiresolution decomposition," *IEEE Trans. on Image Processing*, vol. 5, no. 1, Jan. 1996, pp. 4-15.

[57] D. Park and M. H. Lee, "Image compression based on best wavelet packet bases," *IEEE Trans. on Consumer Electronics*, vol. 40, no. 3, 1994, pp. 527-537.

[58] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. on Image Processing*, vol. 4, no. 11, Nov. 1995, pp. 1549-1560.

[59] G. Loum, P. Provent, J. Lemoine and E. Petit, "A new method for texture classification based on wavelet transform," *Proc. of the IEE SP Int. Symp. on Time-Frequency and Time-Scale*, Paris, June 1996, pp. 29-32.

[60] X. Tang, W. K. Stewart, "Texture classification using wavelet packet and Fourier transforms," *IEEE Oceans*, 1995, pp. 387-396.

[61] K.S. Thyagarajan, T. Ngyune and C. E. Persons, "A maximum likelihood approach to texture classification using wavelet transform," *Proc. of Int. Conf. Image Process., ICIP*, Austin, TX, Nov. 1994, pp. 640-644.

[62] F. M. de St-Martin, E. Gautier and J-P. D'Ales de Corbet, "Signal-based design of wavelet filters," *Proc. IEEE Int. Conf. Accoust. Speech Signal Process., ICASSP*, Atlanta, GA, May 1996, pp. 1495-1498.

[63] T. Greiner, "Orthogonal and bi-orthogonal texture-matched wavelet filterbanks for hierarchical texture analysis," *Elsevier Signal Processing*, vol. 54, 1996, pp. 1-22.

[64] A. Tirakis, A. Delopoulos and S. Kollias, "Two-Dimensional filter bank design for optimal reconstruction using limited subband information," *IEEE Trans. on Image Processing*, vol. 4, no. 8, Aug. 1995, pp. 1160-1165.

[65] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *J. ACM*, vol. 7, 1960, pp. 212-220.

[66] J.A. Neldar and R. Mead, "A simplex method for function minimization," *Comput. J*, vol. 7, 1965, pp 308-313.

[67] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *Comput. J.*, vol. 7, 1964, pp. 149-154.

[68] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, 13 May 1983, pp. 671-680.

[69] A. Corona, M. Marchesi, C. Martini and S. Ridella, "Minimizing multimodal functions of continuous variables with the simulated annealing algorithm," *ACM Transactions on Mathematical Software*, vol. 13, no. 3, Sept. 1987, pp. 262-280.

[70] R. A. Rutenbar, "Simulated annealing algorithms: An overview," *IEEE Circuits and Devices Magazine*, Jan. 1989, pp. 19-26.

[71] R. H. J. M. Otten and L. P. P. P. Ginneken, *The Annealing Algorithm*, Kluwer Academic Publishers, 1989.

[72] I. Pitas, "Optimization and adaptation of discrete-valued digital filter parameters by simulated annealing," *IEEE Trans. on Signal Processing*, vol. 42, no. 4, Apr. 1994, pp. 860-865.

[73] K. H. Sahin and A. R. Ciric, "A dual temperature simulated annealing approach for solving bilevel programming problems," *Computers and Chemical Engineering*, vol. 23, 1998, pp. 11-25.

[74] V. Ravi, B. S. N. Murthy, and P. J. Reddy, "Nonequilibrium simulated annealing algorithm applied to reliability optimization," *IEEE Trans. on Reliability*, vol. 46, no. 2, June 1997, pp. 233-239.

[75] B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of Operations Research*, vol. 13, no. 2, May 1988, pp. 311-329.

[76] N. Benvenuto, M. Marchesi and A. Uncini, "Applications of simulated annealing for the design of special digital filters," *IEEE Trans. on Signal Processing*, vol. 40, no. 2, Feb. 1992, pp. 323-332.

[77] R. W. Maier, and W. B. Whiting, "The variation of parameter settings and their effects on performance for the simulated annealing algorithm," *Computers and Chemical Engineering*, vol. 23, 1998, pp. 47-62.

[78] T.-Y. Wang and K.-B. Wu, "A parameter set design procedure for the simulated annealing algorithm under the computational time constant," *Computers & Operations Research*, vol. 26, 1999, pp. 665-678.

[79] W. E. Reddick, "Automated spatial filter optimization using frequency-domain simulated annealing," *Optical Engineering*, vol. 31, no. 1, Jan. 1992, pp. 82-87.

[80] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, NJ, 1989.

[81] M. Multi and G. Vachtsevanos, "Automated fault detection and identification using a fuzzy-wavelet analysis technique," *Autotestcon*, 1995, pp. 169-175.

[82] J. E. Lopez, R. R. Tenney and J. C. Deckert, "Fault detection and identification using real-time wavelet feature extraction," *Proc. of the IEEE SP Int. Symp. on Time-Frequency, Time-Scale Analysis*, Philadelphia, PA, Aug. 1994, pp. 217-220.

[83] H. Ko, F. Berko, B. Telfer and J. Garcia, "Image enhancements using wavelet preprocessing for printed circuit board classification," *Proc. SPIE for Optical Engg.*, 1995, pp. 473-480.

[84] M. D. Pritt, "Automated subpixel image registration of remotely sensed imagery," *IBM J. Res. Develop.*, vol. 38, no. 2, Mar. 1994, pp. 157-165.

[85] M. H. Er, "Designing notch filter with controlled null width," *Signal Processing*, vol. 24, 1991, pp 319-329.

[86] K. Hirano, S. Nishimura and S. K. Mitra, "Design of digital notch filter," *IEEE Trans. on Circuits and Systems*, vol. CAS-21, no. 4, July 1974, pp. 540-546.

[87] D. B. Williams, "Counting the degrees of freedom when using AIC and MDL to detect signals," *IEEE Trans. on Signal Processing*, vol. 42, no. 11, Nov. 1994, pp. 3283-3284.

# APPENDIX A

# COMPARISON OF TEXTURE CLASSIFICATION PERFORMANCE

The Battle-Lemarie cubic spline wavelet basis function with 16 taps is used in energy based tree decomposition and the classification results are compared in Table A.1 with the results given in reference [41] for the Euclidean distance. Only seven textures are used for comparison since these are the only textures that could be obtained with the same ID as in the reference [41].

**TABLE-A.1: COMPARISON OF CLASSIFICATION USING EUCLIDEAN DISTANCE**

| *Texture* | *Correct Classification Rate (%)* | |
|---|---|---|
| | *Results* | *Results from [41]* |
| Brick | 94 | 98 |
| Grass | 100 | 96 |
| Sand | 85 | 92 |
| Wood Grain | 100 | 97 |
| Cloth | 96 | 100 |
| Leather | 99 | 100 |
| Raffia | 100 | 100 |
| *Overall* | *96.29* | *97.57* |

The classification results obtained using the separability based tree decomposition are compared in Table-A.2 with the results in reference [43]. The textures used in the database are not the same as in this reference. Also, the wavelet and the depth of the decomposition used in obtaining the results listed in this reference are unknown for exact comparison.

TABLE-A.2: CLASSIFICATION USING $\mu_2$ AND $\mu_3/\mu_2$ (8 FEATURES) AND NEURAL NET CLASSIFIER

| *Texture* | *Results From [43]* | *Our Results* |
|---|---|---|
| T1 | 100 | 92 |
| T2 | 98 | 95 |
| T3 | 99 | 98 |
| T4 | 100 | 99 |
| T5 | 100 | 99 |
| T6 | 98 | 97 |
| T7 | 97 | 100 |
| T8 | 100 | 92 |
| T9 | 100 | 100 |
| T10 | 99 | 100 |
| *Average* | 99.1 | 97.2 |

# APPENDIX B

# MATHEMATICAL DETAILS OF CLASS SEPARABILITY

A good feature extractor should not only yield a large distance between the mean vectors of the extracted feature vectors of the classes (clusters), but also should yield low variances across the feature vectors in the classes (clusters). This is exactly what is expressed by class separability:

$$J = Tr(S_w^{-1} S_b) \tag{B. 1}$$

The within-class scatter matrix shows the scatter of sample vectors ($V$) of different classes around their respective mean or expected vectors $M$:

$$S_w = \sum_{i=1}^{L} \Pr\{C = C_i\} \Sigma_i \quad \text{where } \Sigma_i = E[(V - M_i)(V - M_i)^T | C_i] \tag{B. 2}$$

represents the spread of feature vectors in the $i^{th}$ class. In addition, one can define the between-class scatter matrix as the scatter of the conditional mean vectors $M_i$ around the overall mean vector $M$:

$$S_b = \sum_{i=1}^{L} \Pr\{C = C_i\} (M - M_i)(M - M_i)^T \quad . \tag{B. 3}$$

In order to find optimal filter, the partial derivative of the criterion, $\partial J(h)/\partial h$, is equated to zero. The cost function $J$ needs to be expressed in terms of $h$ so that the gradient of the cost function can be calculated.

$$\begin{aligned} \frac{\partial J}{\partial h} &= \frac{\partial}{\partial h}\left[tr\left(S_w^{-1} S_b\right)\right] \\ &= \frac{\partial\left[tr\left(S_w^{-1} S_b\right)\right]}{\partial S_w}\frac{\partial S_w}{\partial h} + \frac{\partial\left[tr\left(S_w^{-1} S_b\right)\right]}{\partial S_b}\frac{\partial S_b}{\partial h} \\ &= -S_w^{-1} S_b S_w^{-1}\frac{\partial S_w}{\partial h} + S_w^{-1}\frac{\partial S_b}{\partial h} \end{aligned} \tag{B. 4}$$

$$\begin{aligned}\frac{\partial S_w}{\partial h} &= \frac{\partial}{\partial h}\left\{\sum_{i=1}^{L} Pr(C=C_i)E\left[(V-M_i)(V-M_i)^T\middle|C_i\right]\right\} \\ &= \sum_{i=1}^{L}\left[\frac{\partial(A_1A_2)}{\partial A_1}\frac{\partial A_1}{\partial h}+\frac{\partial(A_1A_2)}{\partial A_2}\frac{\partial A_2}{\partial h}\right] \\ &= \sum_{i=1}^{L}\left[A_1\frac{\partial A_2}{\partial h}\right] \\ &= \sum_{i=1}^{L} Pr(C=C_i)\frac{\partial}{\partial h}\left\{E\left[(V-M_i)(V-M_i)^T\middle|C_i\right]\right\}\end{aligned} \tag{B.5}$$

where $V$ is a set of feature vectors of all the texture samples that belong to a class.

$$V=\{v_i=\mu_2(W_i): i=0,1,....N_{subbands}\}$$

$$\mu_2(W)=\frac{1}{|W|}\left(\sum_{x\in W}\left(f(x)-\overline{f_W}\right)^2\right)^{1/2}$$

$W_i$ is the local window on the $i^{th}$ sub-band. On each sub-band, $f(x)$ and $\overline{f_W}$ are defined as the intensity value at the location $x$ and the average intensity on window $W$ centered at $x$ respectively. For each sub-band, $\mu_2$ shows the average energy, which is also called the F-norm. Also, $M_i$ is the mean vector of all the vectors in the $i$th class.

$$\begin{aligned}\frac{\partial S_b}{\partial h} &= \frac{\partial}{\partial h}\left\{\sum_{i=1}^{L} Pr(C=C_i)\left[(M-M_i)(M-M_i)^T\middle|C_i\right]\right\} \\ &= \sum_{i=1}^{L}\left[\frac{\partial(B_1B_2)}{\partial B_1}\frac{\partial B_1}{\partial h}+\frac{\partial(B_1B_2)}{\partial B_2}\frac{\partial B_2}{\partial h}\right] \\ &= \sum_{i=1}^{L}\left[B_1\frac{\partial B_2}{\partial h}\right] \\ &= \sum_{i=1}^{L} Pr(C=C_i)\frac{\partial}{\partial h}\left\{(M-M_i)(M-M_i)^T\middle|C_i\right\}\end{aligned} \tag{B. 6}$$

where $M$ is the mean vector of all the means of all the classes.

The complexity involved in expressing the separability and its gradient as functions of h, which are needed for obtaining the optimal filter by solving the equation $\frac{\partial J}{\partial h_o}=0$ is substantial, if not intractable. Hence, one needs to look for other ways to optimize the cost function to find the optimal wavelet.

# APPENDIX C

# FAULT DETECTION AND IDENTIFICATION

Advanced diagnostic systems can improve the safety, reliability and reduce the cost of operating sophisticated platforms such as a host of expensive commercial manufacturing systems. As an example, consider an electronic industry manufacturing circuit boards. Operational costs for maintaining and repairing the circuit boards by testing each component in the circuit are expensive and time consuming. In this chapter, we develop an algorithm and design wavelets using a data driven approach for fault detection and identification for printed circuit boards. This is a particular application of the algorithm developed in Chapter 7. Fault detection and identification is of interest in a wide variety of applications such as control systems, image analysis, analysis of radar signals, smart sensors, texture analysis, medicine, industry, *etc.*

This appendix is organized as follows. Section C.1 describes the infrared images used in the analysis for fault detection. The architecture of the FDI algorithm showing the various blocks used is presented in section C.2. The image sequences of various boards may have different initial conditions such as starting temperature. Hence, the images need to be preprocessed before they can be analyzed. This is discussed is section C.3. The necessity to adapt a data driven approach for fault identification is presented in section C.4. The influence of various parameters on the FDI system performance is studied in sections C.4.1 to C.4.4. The measure required to develop an algorithm for improving the performance is presented in section C.5. The wavelet basis is designed using this technique in section C.6. The conclusion follows in section C.7.

## C.1. INTRODUCTION

A 2-D image of a printed circuit board representing intensities as a function of position is captured by a high-resolution infrared camera. Figure C.1.a shows the infrared image of a typical printed circuit board and Figure C.1.b shows the infrared image of a component on the circuit board. Infrared thermal imaging systems operate on the basic principle that all objects above absolute zero (-273° C) radiate infrared energy,

the intensity and wavelength of which are proportional to the temperature of the object. Since infrared energy is

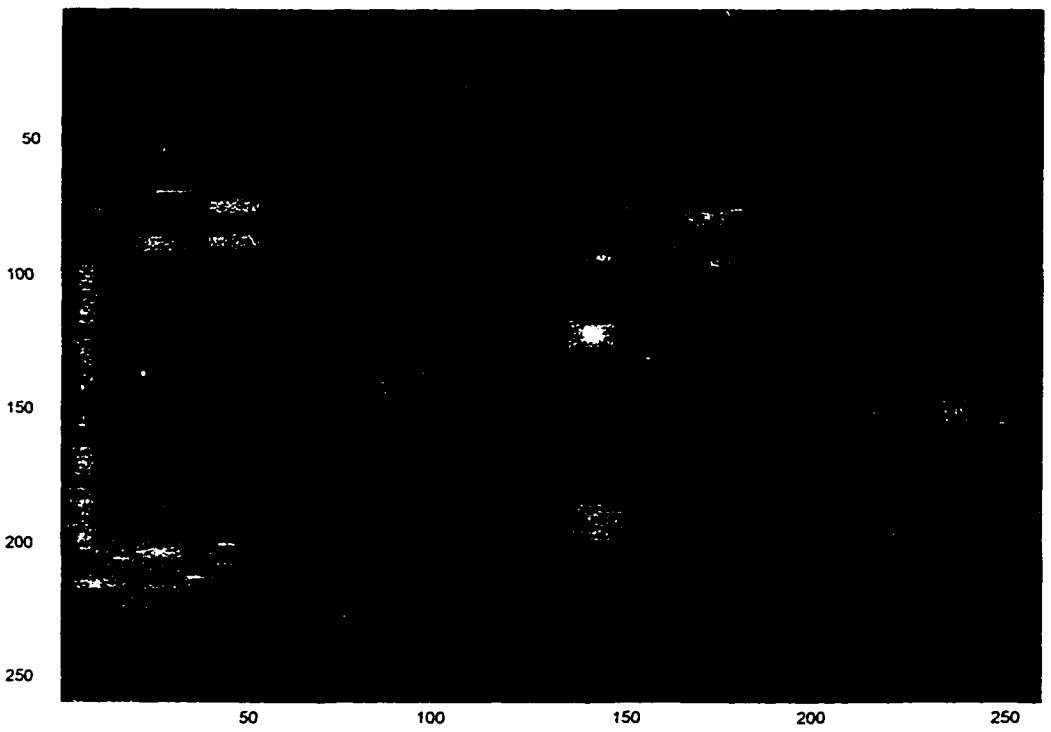



Figure C.1.a. Infrared Image of a printed circuit board

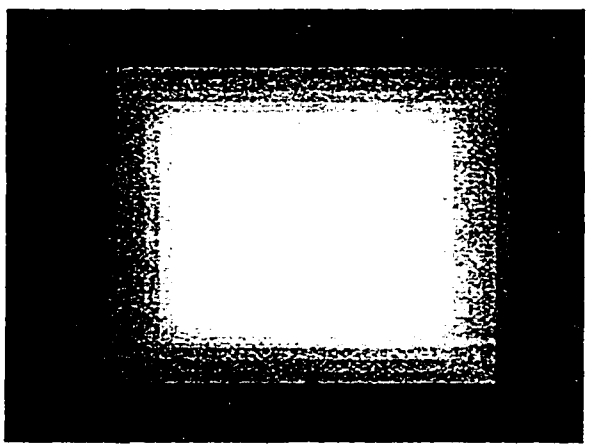


Figure C.1.b. Infrared Image of a component in the circuit board

not visible to the naked eye, some means must be provided to transform it to a visual image. Essentially, an infrared thermal imaging system consists of a camera that remotely picks up the infrared radiation being transmitted, detectors which transform the radiation to electric signals, an amplifier to boost the signal to suitable levels, and a monitor to view the visual image.

Most equipment failures in an industrial or commercial facility are accompanied by increased or decreased temperatures. Infrared thermal imaging can detect this change in temperature, and thus can be applied to a variety of areas. Since no physical contact is required between objects being scanned and the test equipment, it can be used during normal operation of a facility.

The important task of an FDI system is to identify or classify the faulty components on the circuit boards. FDI systems can assist in fault localization and isolation. They are used to maintain the functionality of the system.

With the availability of powerful computing platforms, feature processing has become an important part of many applications. Intelligent processing like fuzzy logic, neural networks and intelligent optimization techniques are aimed at accommodating a large gain in uncertainty while utilizing all of the available information about the system [81], [82], [83]. Due to the wide range of time constants, analysis of such systems in the frequency domain alone would mask the sudden high frequency bursts. Unless the frequency domain resolution is very fine, slowly varying features can be masked in the dc bias. Likewise, analysis in the time domain would not reflect the periodicity of the features. Hence, analysis in only the frequency or time domain alone is not sufficient to capture features that are spread in a wide band of frequencies. Faults of these types require analysis simultaneously in both the time and frequency domains. This can be accomplished by using Wavelet Transform (WT) techniques. The WT uses a variable window size to analyze different frequencies. Moreover, it provides a wide choice of wavelets for the best fit in different applications.

## C.2. ARCHITECTURE OF THE FDI ALGORITHM

The basic blocks involved in the fault detection and identification system are given in Figure C.2. Details of the individual components are given in the following sections. The blocks used in the analysis

section of the FDI system are given Figure C.3. The first three blocks in Figure C.2 are discussed in section C.3 and the analysis block is presented in sections C.4, C.5 and C.9.

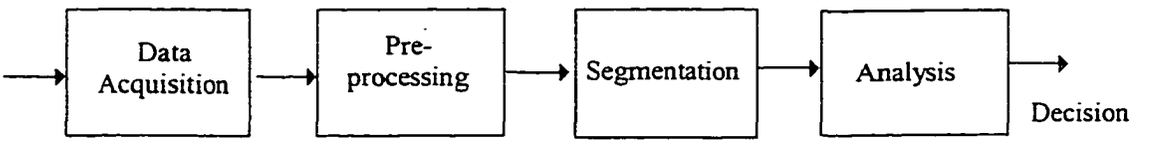



Figure C.2. Basic Block Diagram of the FDi system

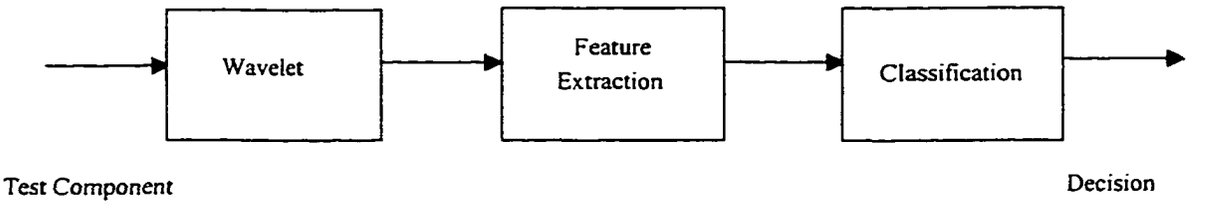



Figure C.3. The blocks used in the analysis of the Fault Identification System.

## C.3. PREPROCESSING

The images are captured using an infrared camera through time as the circuit board is powered. The circuit components brighten through time as they are powered up. The faulty components are either very bright or dark depending on whether the component is too hot or too cool. This could be due to open or short circuits. Not all of the components that are brighter or darker are necessarily faulty. Some are hot or cool as a consequence of logical function. The heat index is also dependent on the characteristics of the components and the manufacturing method. Some components can be very bright and still function. Consequently, one cannot look at the infrared images or sequences of a circuit board through time and identify the faulty component by locating the brightest or darkest component in that circuit board. Therefore, the approach is to compare the infrared images of a faulty (bad) board and a functioning (good) board, and then identify those components that are the most different from the corresponding components on the good board. To do this, one has to properly align the boards to be compared both in time and in space. To compare the components in a bad board with their corresponding components in a good board,

these components have to be extracted from the whole board. For this, the board needs to be segmented so that the components can be extracted after registration and before analysis.

### C.3.1. REGISTRATION

Most computer vision applications require the analysis of two or more images, whereby images are compared with one another to detect certain properties of the involved projects [84]. To accomplish this, a proper alignment of the images must be found, which means a transformation to relate points of one image with corresponding points of another image is required. When the infrared images of a good board and a bad board are captured through time, the boards may have different starting temperatures depending on the time the sequences are recorded. In addition, many other factors can influence the initial conditions of the two boards, such as room temperature, the time the sequences are acquired, etc. Influences of all these factors can be nullified by registering the frames through time.

To register the pixels through time, a notch filter has been used [85], [86]. The problem with the notch filter is that the transient response distorts the filter output on start-up. Typically, a notch filter with the narrower 3-dB rejection bandwidth has a longer transient response at the filter output. However, we prefer a notch filter with a narrower 3-dB rejection bandwidth to faithfully notch out the dc frequency (the mean). With the limited number of samples available (100 samples, i.e., 100 frames through time), the transient response exists until around 90 samples, which is undesirable.

To avoid this situation, a single order notch filter has been designed as follows. The transfer function of the single-order notch filter is [87]:

$$H(z) = \frac{1 - z^{-1}}{1 - rz^{-1}} = \frac{Y(z)}{X(z)} \tag{C.1}$$

The corresponding difference equation is y(n) = x(n) – x(n-1) + ry(n-1), with initial condition y(0) = 0. The parameter r can take values from 0 to 1. With r equal to one, the response y(n) would be equal to the input with the mean subtracted. The pixels are time registered with the single-order notch filter with r equal to 0.99. The warm-up sequences are plotted through time for two good boards and a bad board in Figures C.4. a, b, c and d. The Figure C.4.a shows the warm-up sequence with time registration while Figure C.4.b. shows the same without time registration. One can observe that all the components do not have the same

starting temperature. The same can be observed clearly in Figures C.4.c and d where the plots are zoomed onto the first few frames.

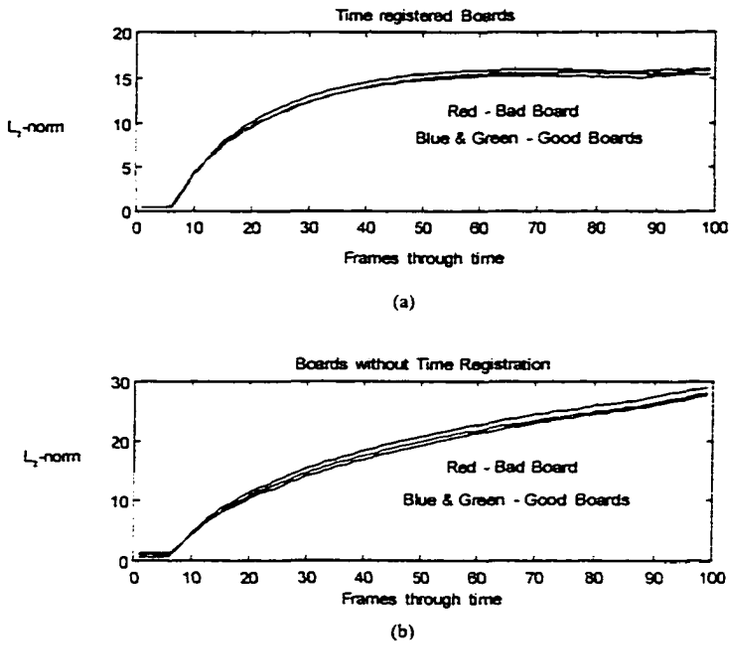



Figure C.1.a, b. Warm-up sequence of the circuit board components with and without time registration.

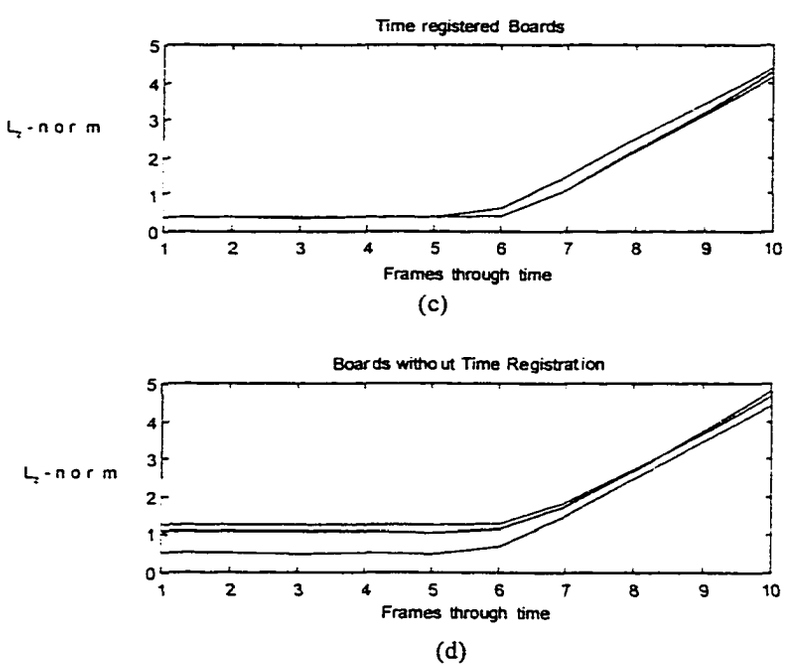



Figure C.4.c, d. Warm-up sequence of the circuit board components with and without time registration zoomed into first few frames to see the difference.

C.3.2. SEGMENTATION

Each component in the printed circuit board must be identified so that the board can be registered to a template, which is called segmentation[3]. The segmentation is difficult because of two factors: the infrared images have poor contrast, and the background color of the images varies from region to region. These factors led to the use of a connected component merging algorithm to locate the individual components. The connected component-merging algorithm proved to be quite effective, but very computationally expensive. In order to make the algorithm more efficient, a morphological pyramid is used to reduce the amount of computation. Figure C.5 shows the segmented image of a circuit board.

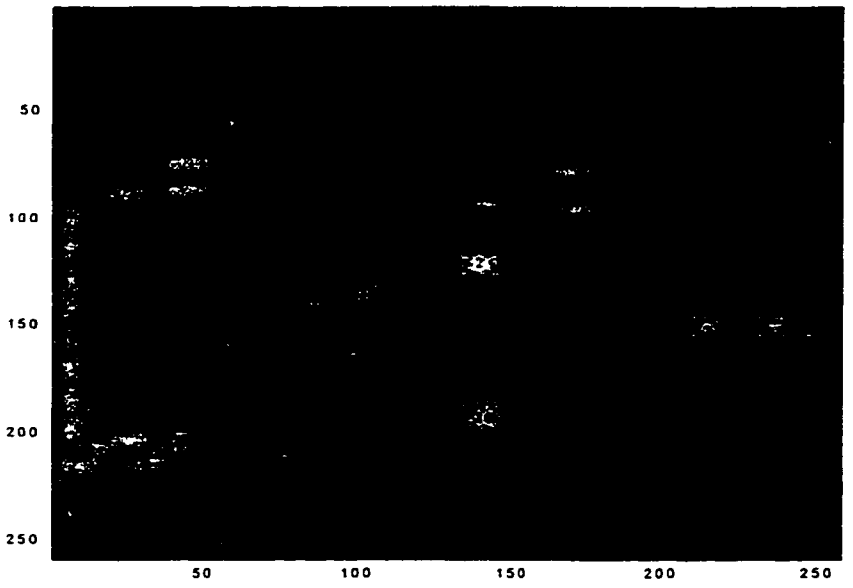



Figure C.5. Segmented image of a printed circuit board

## C.4. GENERALIZED DATA DRIVEN APPROACH

We know that analyzing the data in the wavelet domain gives better understanding and performance than looking at the data in the image domain. Also, local analysis is needed to identify the faulty component in a bad board. This means that the components that are likely to be faulty need to be segmented from the board after registration and analyzed for detection. However, the performance achieved

using the Daubechies wavelet is not satisfactory. To improve the performance, one needs to look at the various parameters involved in analyzing the data in the wavelet domain that could influence the performance in identifying and detecting the faulty component in a bad board. These factors can be roughly divided into three categories. The three major categories are the parameters involved in the wavelet transform, the factors involved in feature extraction, and the classifier used. The major parameters involved in the wavelet transform are the mother wavelet, the depth of the decomposition, and the tree structure used to decompose the image (uniform, octave, criterion-based tree decomposition). The factors in the feature extraction block that could influence the performance are the measures used to extract features, the number of features extracted, and the feature reduction methods. Finally, the type of classifier used also has some influence on the fault detection.

All these parameters and their influences on fault identification system need to be studied. This helps us understand what needs to be done to improve the performance of the FDI system. All these parameters have considerable influence on the performance of the FDI system. However, the wavelet basis function used in the FDI system is crucial as explained in Section 5.1. Feature extraction is used to extract the information that is useful for identification and fault detection. To extract useful information, the information content of the signal needs to be localized in the time and frequency domains, which is dependent on the type of wavelet transform used. The influence of the wavelet basis on the FDI system performance is studied first and the influence of the remaining factors are studied in conjunction with the wavelet used.

### C.4.1. INFLUENCE OF WAVELET-BASIS ON FDI SYSTEM PERFORMANCE

We are interested in finding the most suitable wavelet for fault detection and identification (FDI) and applying the wavelet system to improve the FDI system performance for printed circuit boards. To achieve this the wavelet has to be designed based on the properties of the data. To test the effect of different wavelet bases on FDI system performance, various 4-tap wavelets with different transition band characteristics are chosen satisfying the Quadrature Mirror Filter (QMF) constraints. This set of wavelets

[3] Thanks to Anthony Wright and Dr. Scott Acton for providing the segmentation algorithm and software.

also includes the Daubechies 4-tap wavelet. In all our plots, the last case of h, i.e. h=25, is the Daubechies 4-tap wavelet.

Using each wavelet, the image sequences are transformed into the wavelet domain after pre-processing. Not all of the frames acquired through time need to be transformed into the wavelet domain for processing. One frame halfway through the warm-up sequence has enough information for fault identification. Two data sets, each with five good boards and five bad boards are used for experiments. One data set has 100 frames through time for ten boards and the second data set has 30 frames through time. For the first data set, the $50^{th}$ frame is used for analysis while the $15^{th}$ frame is used for the second data set. All the components that are likely to be faulty are located and they are segmented from the board. All these components are transformed into the wavelet domain using each wavelet. Once, the images are transformed into the wavelet domain, all the discrete wavelet transform coefficients are not needed for fault identification. Only the information that is required for fault identification needs to be emphasized and rest of the information needs to de-emphasized. A crucial element in the data driven approach is the selection of an appropriate feature extractor. So, the features that have significant information for fault identification need to be extracted. The influence of extracting features using different measures is studied later. A typical measure used for analysis is the energy of the discrete wavelet transform coefficients in a sub-band. Let us start with a simple analysis by using one level of wavelet decomposition and extracting four features, one from each sub-band (LL, LH, HL, and HH) as shown in Figure C.6. These four features are used for fault identification. Now, the faulty component needs to be detected from all the components chosen for analysis.

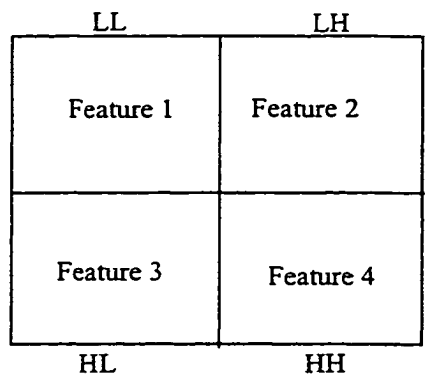



Figure C.6. One-level Sub-band Tree and Feature map.

This fault identification problem can be looked upon as a classification problem. Here, the classes are good and bad components. For each component, we have two classes. For effective fault detection, we need to possess an extensive database of good and bad components for each component under test. In such a case, each component in the bad board under test is classified as either a good or a bad component. This is done by calculating the separation between the test component and class-1 (the good component class) in the database, and the test component and class-2 (the bad component class) in the database. However, it is difficult to have a database with both good and bad components for at least a few components on the board. Hence, one needs to look at other alternatives for classification. One way to do that is to calculate the separation between each component under test in the bad board and the corresponding component in the good board. The component in the bad board with the most separation from the corresponding good component in the good board is most likely to be faulty. The Euclidean distance is used to calculate the separation between the components.

The performance of the FDI system can be measured in two ways. One way is to calculate the number of times the faulty component is detected correctly i.e. the number of correct decisions made. The second approach is to calculate the separation between the good and bad components (classes). In all our experiments, though we present the number of correct decisions made, we focus on the separation between good and bad components due to the small number of available boards in each category.

The plots in Figures C.7 to C.8 show the warm-up sequences of the faulty component and several other good components in a bad board and the corresponding good components in a good board consecutively. One can observe from the Figures C.7 and C.8 that brighter or darker components in a bad board need not be the faulty components. So, just by looking at the bad board or the faulty component alone by itself, one cannot make a decision about the faulty component. The component in the bad board that is most different from the corresponding good component in the good board is the faulty component. Figure C.9 shows the distance between the corresponding components in good and bad boards through time.

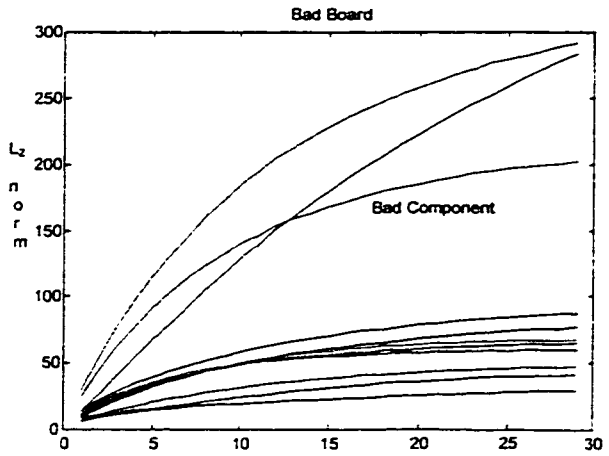



Figure C.1. Warm-up sequence of different components in a bad board

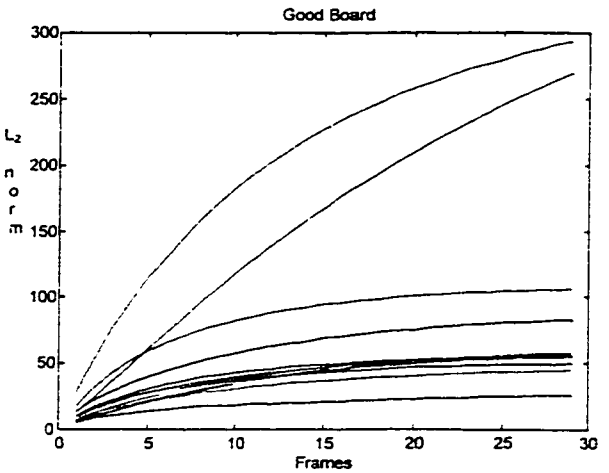



Figure C.2. Warm-up sequence of same components in a good board

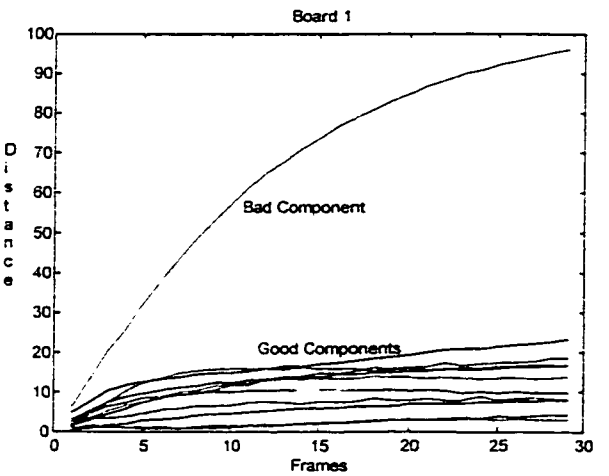



Figure C.3. Distance between corresponding components in good and bad boards

It can be observed that the distance is very high between the faulty component and the corresponding component in a good board compared to the distances between remaining components. The distance is considerable enough for fault identification at the $15^{th}$ frame, so the analysis can be done using just the $15^{th}$ frame alone. Now, these distances are used to detect the fault and make the decision. These distances are calculated for each wavelet and the distances are plotted with respect to the wavelets for different boards in Figure C.10.

It can be observed that three out of five times all the wavelets found the faulty components correctly. In board-5, only a few wavelets detected the faulty component correctly. However, in board-4, the faulty component is not correctly detected by any wavelet. The advantage of using different wavelets is not very clear from these plots except in board-5. To see the influence of wavelets on the performance of the FDI system, the separability is calculated. The separability is calculated as the separation between the faulty component and the average of the rest of the good components in the bad board. We know that the distance has to be higher for the faulty component. Hence, the separability between the bad component and the average of the good components should be positive if the faulty component has larger distance. For example, for boards-1, 2, 3 and 5, the faulty component has larger distance and so has positive separability. However, for board-4, the distance is not higher for the faulty component and so the decision made is wrong. In this case, the separability is negative. The higher the separability, the better the separation between bad and good components in the bad board and hence the better performance of the FDI system. The separability of the faulty component is plotted for different wavelets for all five boards in Figure C.11.

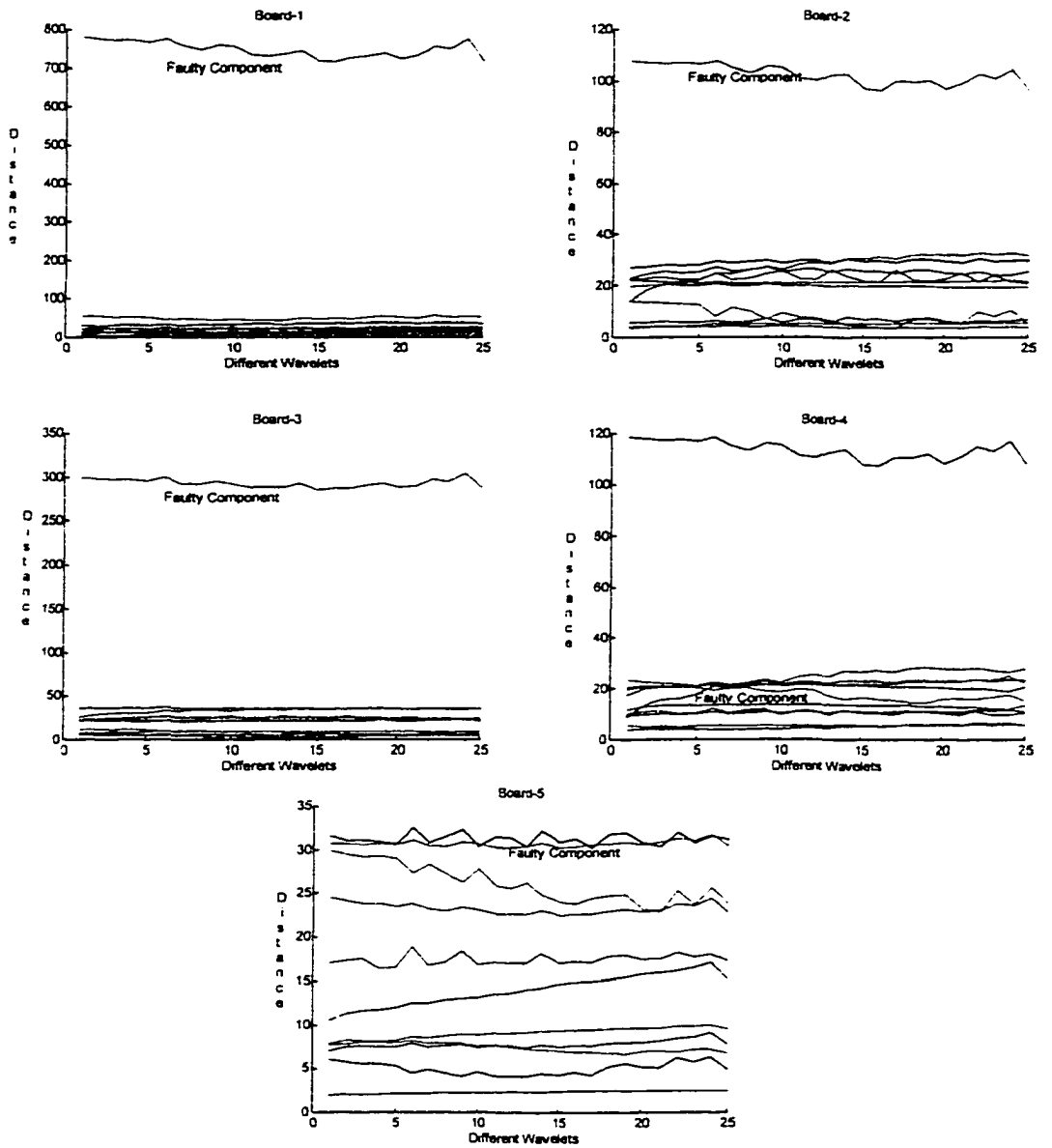



Figure C.1. Distances of various components for different wavelets for five boards

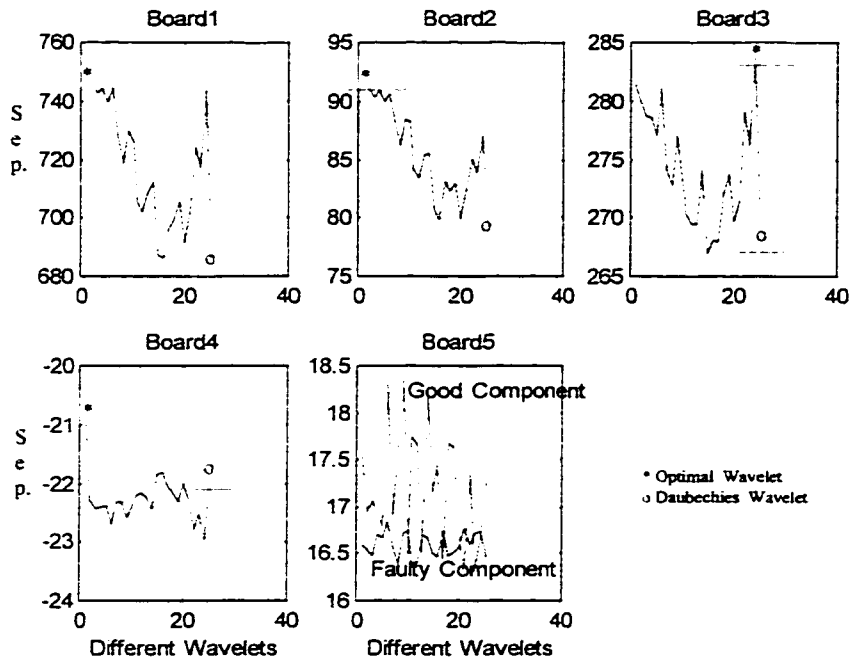



Figure C.2. Separability of the bad component for various wavelets in five boards

The wavelet that could detect the faulty component correctly the highest number of times is the best wavelet. With the limited boards available for testing, the statistics are not sufficiently significant to base the decision on the number of decisions made correctly. Hence, the wavelet with highest separability is chosen as the best for fault identification. Table-C.1 shows the statistics e.g., mean, maximum, minimum, *etc.* of various wavelets considered in the experiments. The variation in performance for different wavelets is significant for some boards and insignificant in others. As explained earlier, negative separability indicates that the faulty component could not be detected correctly.

TABLE-C.1: FDI SYSTEM PERFORMANCE FOR VARIOUS WAVELETS USING F-NORM WITH ONE-LEVEL OF DECOMPOSITION

| Boards | Max. (Optimal Wavelet) | Min. | Mean | Median | Daubechies |
|---|---|---|---|---|---|
| Board-1 | 751.81 | 686.02 | 717.58 | 716.94 | 686.02 |
| Board-2 | 92.31 | 79.51 | 85.63 | 85.41 | 79.51 |
| Board-3 | 284.62 | 266.97 | 274.22 | 273.85 | 268.61 |
| Board-4 | -20.67 | -23.02 | -22.25 | -22.31 | -21.82 |
| Board-5 | 16.83 | 16.38 | 16.60 | 16.59 | 16.45 |

In this experiment, board-5 needs separate explanation. Though the separability is positive for the faulty component in this board, there exists another good component on the same board with higher separation from its corresponding good component on the good board than the separation of the bad component from its corresponding good component on the good board. Most of the wavelets (including the Daubechies wavelet) failed to detect the faulty component correctly in this board. Only a few wavelets detected the faulty component correctly. This can be observed from subplot 5 in Figure C.11. The red curve indicates the separability of a good component and the blue curve indicates the same for the faulty component. The wavelets with the blue curve higher than the red curve detected the bad component correctly. In this case, the best wavelet is the one that detected the faulty component correctly and the one that has higher difference between the blue (faulty) and red (good) curves. This difference has to be positive, i.e. the faulty component separation should be higher than that of the good component.

Extracting four features using the F-norm with one-level of wavelet decomposition alone resulted in improved separation between the faulty component and rest of the good components in a bad board, when a suboptimal wavelet is used. However, the separation is significant only in three boards. As it is not known which measure is the best suited one for feature extraction for a particular application and data set, one needs to analyze other feature extracting measures to study the impact of these on system performance. The

next two sections give details about the influence of extracting features with different measures and the influence of extracting more features on the FDI system performance.

## C.4.2. INFLUENCE OF FEATURE EXTRACTION METHODS ON FDI SYSTEM PERFORMANCE

This section studies the influence of using different measures for feature extraction on the FDI system performance. Please refer to Section 4.3 for definitions of the various feature extraction measures used here. The measures used in the analysis are the $L_1$-norm, the $\Delta$-norm and the absolute sum. The methodology described in the above section for calculating the separability is repeated by using each measure instead of using the F-norm. The separability results are plotted in Figures C.12-C.14 using these three measures for the five boards. Features extracted using the $L_1$-norm and the $\Delta$-norm have significant information for fault detection as they could detect the faulty component correctly in all boards except in board-4. Features using absolute sum did not yield as much useful information, as it could detect faulty component only in three boards. These separability values are summarized in Table-C.2. The best wavelet (sub-optimal) has increased the separability approximately by 10% from the separability given by the Daubechies wavelet.

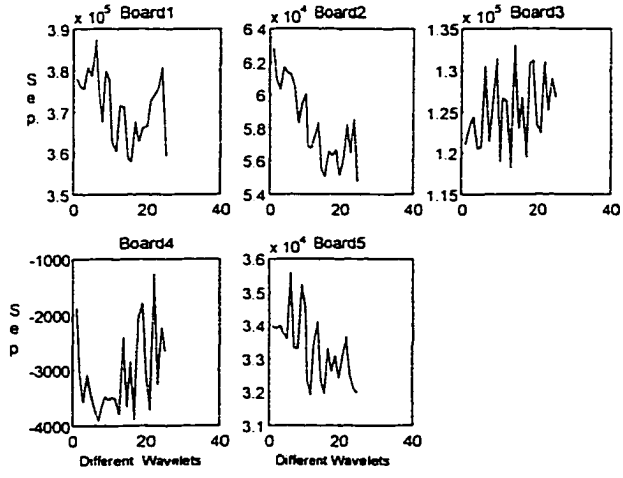



Figure C.12. Separability plots for various wavelets with one-level decomposition using $L_1$-norm

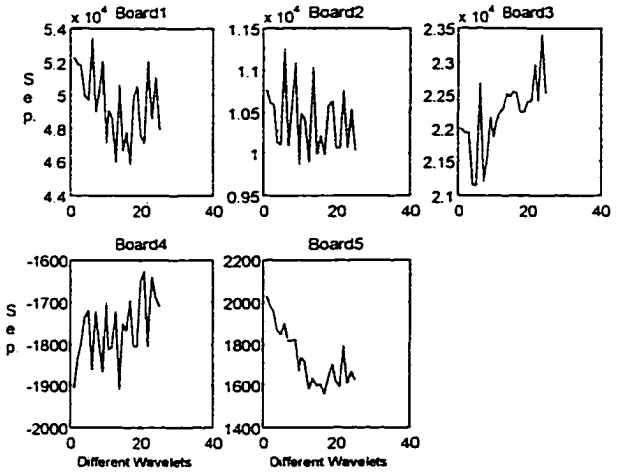



Figure C.13. Separability plots for various wavelets with one-level decomposition using Δ-norm

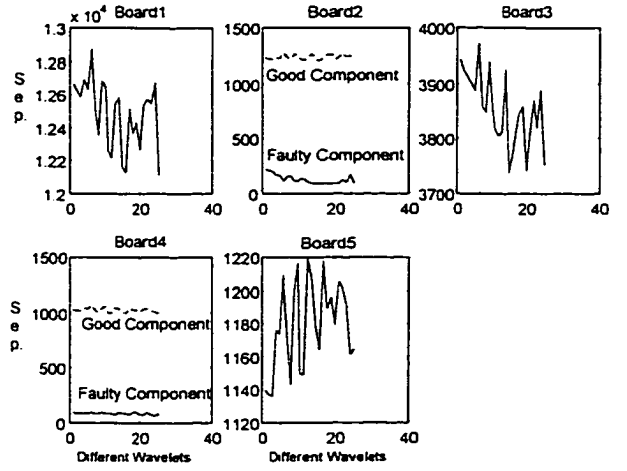



Figure C.14. Separability plots for various wavelets with one-level decomposition using absolute sum of DWT coefficients

TABLE-C.2: FDI SYSTEM PERFORMANCE FOR VARIOUS WAVELETS USING F-NORM WITH ONE-LEVEL OF DECOMPOSITION

| *Boards* | *Max.* *(Optimal Wavelet)* | *Min.* | *Mean* | *Median* | *Daubechies* |
|---|---|---|---|---|---|
| **$L_1$-norm:** | 1.0e+004* | | | | |
| Board-1 | 38.74 | 35.79 | 37.13 | 37.26 | 35.92 |
| Board-2 | 6.28 | 5.48 | 5.82 | 5.82 | 5.48 |
| Board-3 | 2.95 | 2.62 | 2.77 | 2.78 | 2.62 |
| Board-4 | -0.13 | -0.39 | -0.31 | -0.35 | -0.27 |
| Board-5 | 3.56 | 3.19 | 3.33 | 3.33 | 3.20 |
| **Δ-norm:** | | | | | |
| Board-1 | 5.34 | 4.59 | 4.95 | 4.97 | 4.79 |
| Board-2 | 1.13 | 0.99 | 1.04 | 1.04 | 1.00 |
| Board-3 | 0.21 | 0.14 | 0.17 | 0.16 | 0.15 |
| Board-4 | -0.16 | -0.19 | -0.18 | -0.18 | -0.17 |
| Board-5 | 0.20 | 0.16 | 0.17 | 0.17 | 0.16 |
| **Absolute Sum:** | | | | | |
| Board-1 | 1.29 | 1.21 | 1.25 | 1.25 | 1.21 |
| Board-2 | 0.02 | 0.0089 | 0.013 | 0.012 | 0.0089 |
| Board-3 | 0.104 | 0.096 | 0.10 | 0.10 | 0.099 |
| Board-4 | 0.0095 | 0.0066 | 0.0083 | 0.0083 | 0.0081 |
| Board-5 | 0.122 | 0.114 | 0.118 | 0.118 | 0.117 |

## C.4.3. INFLUENCE OF NUMBER OF FEATURES EXTRACTED ON FDI SYSTEM PERFORMANCE

In this section, the influence of extracting more features from each sub-band with the same wavelet tree and feature extraction methods on the FDI system performance is studied. Sixteen features are extracted instead of four features, four from each sub-band. Each sub-band is divided into four regions. Figure C.15 shows the feature extraction strategy from the image subbands and regions.

The separability values are plotted for all the measures using the 16 features for various wavelets in Figure C.16. The performance did not increase considerably either in detecting more faulty components correctly or the separability compared to the Daubechies wavelet. In fact, the measure Δ-norm could not detect the faulty component correctly for board-5. Thus, extracting more features with one level of tree decomposition did not improve the separability or faulty detection system performance over the case where only four features were used.

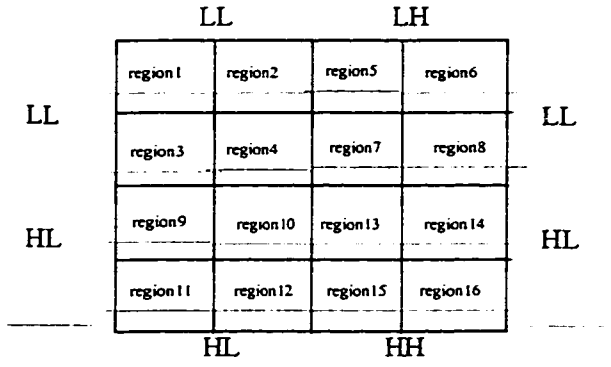
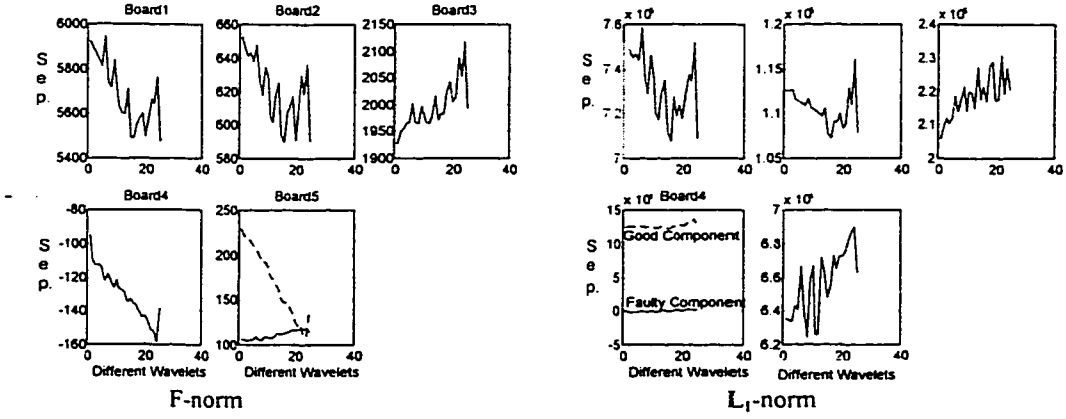



Figure C.15. Feature extraction strategy



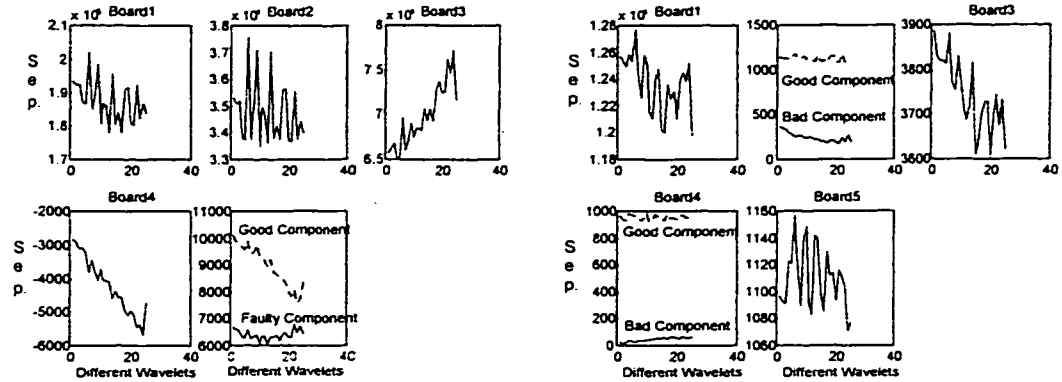



Figure C.16. Separability plots for various wavelets using F-, $L_1$-, Δ-norms and absolute sum of DWT coefficients using 16 features with one level of decomposition

## C.4.3. INFLUENCE OF STRUCTURE AND DEPTH OF THE TREE ON FDI SYSTEM PERFORMANCE

In this section, the influence of the depth of the wavelet tree or the level of decomposition on the FDI system performance is studied along with the type of tree used. With an increase in the depth of the tree, the sub-bands are decomposed further and the finer details of the image are used for identification. Uniform and octave subband tree decompositions are used here for performance comparison. Figure C.17 shows the tree structures of the uniform and octave trees for second level of decomposition. Sixteen features are extracted for the uniform tree structure and seven features for the octave band tree, one from each subband.

The variation of separability across different wavelets is shown in Figures C.18 and C.19 for the uniform and octave trees, respectively. The uniform tree decomposition performed better in terms of separability and the number of correctly made decisions than did the octave tree for the F- and $L_1$- norm measures. For the other two measures, there is not much improvement.

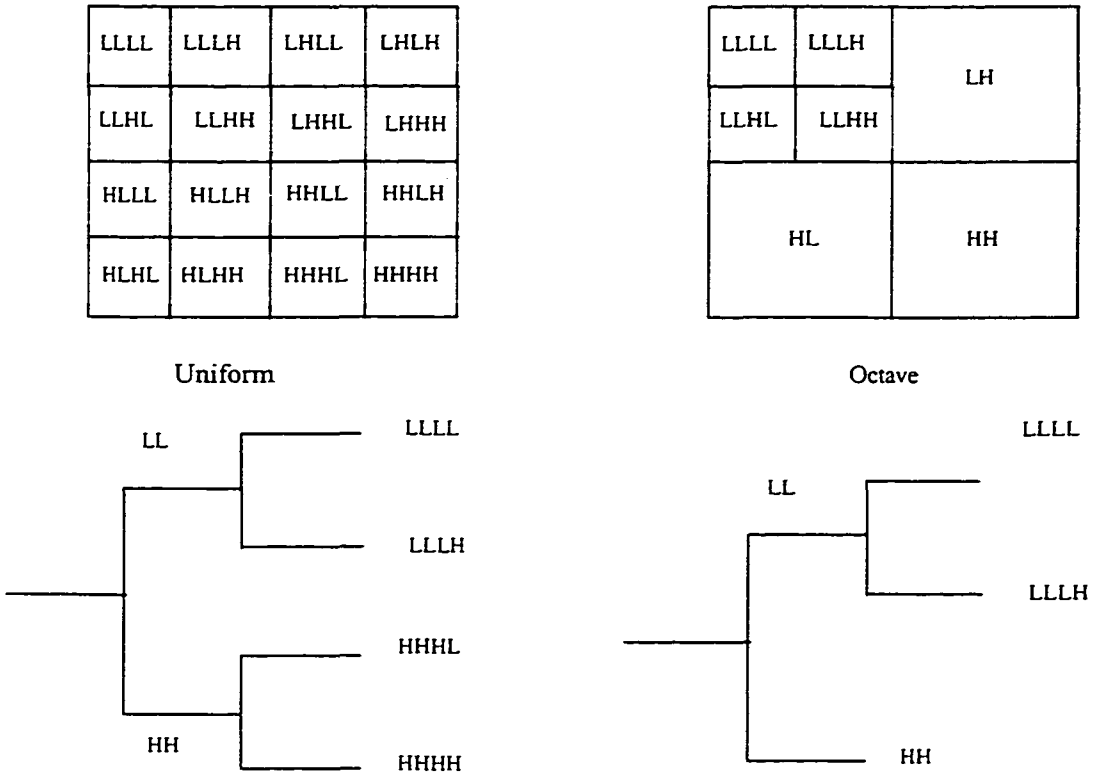



Figure C.17. Uniform and Octave trees for 2-D and 1-D

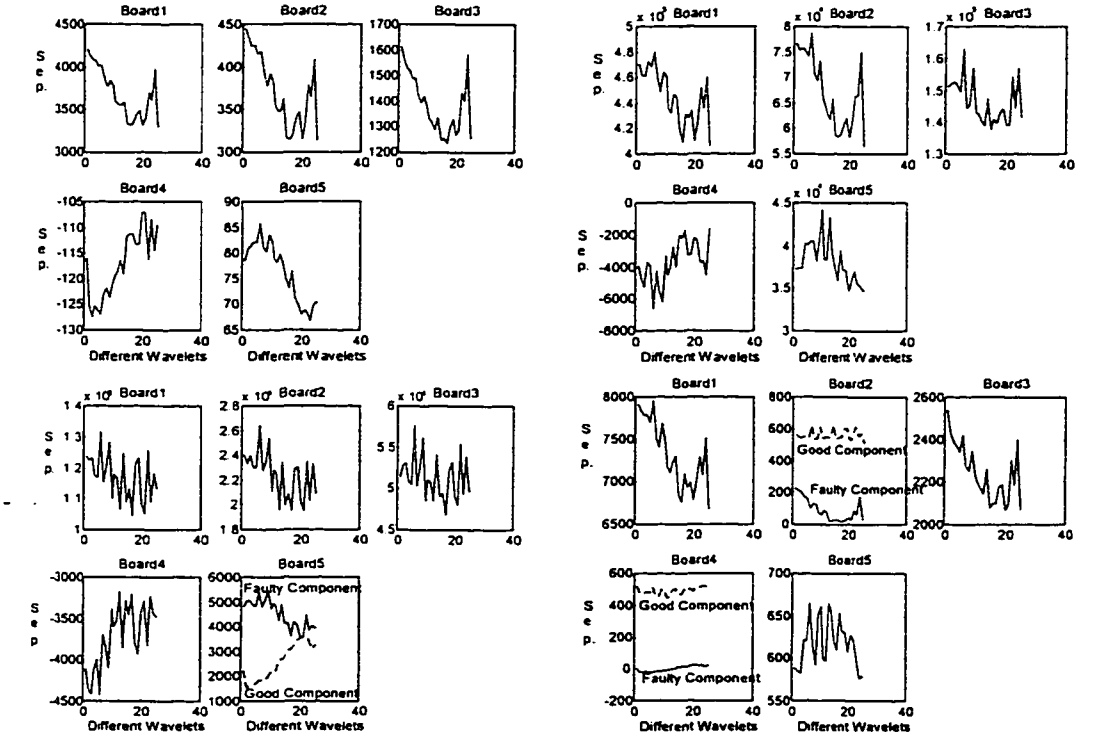



Figure C.18. Separability plots for various wavelets using uniform tree for second level and extracting 16 features.

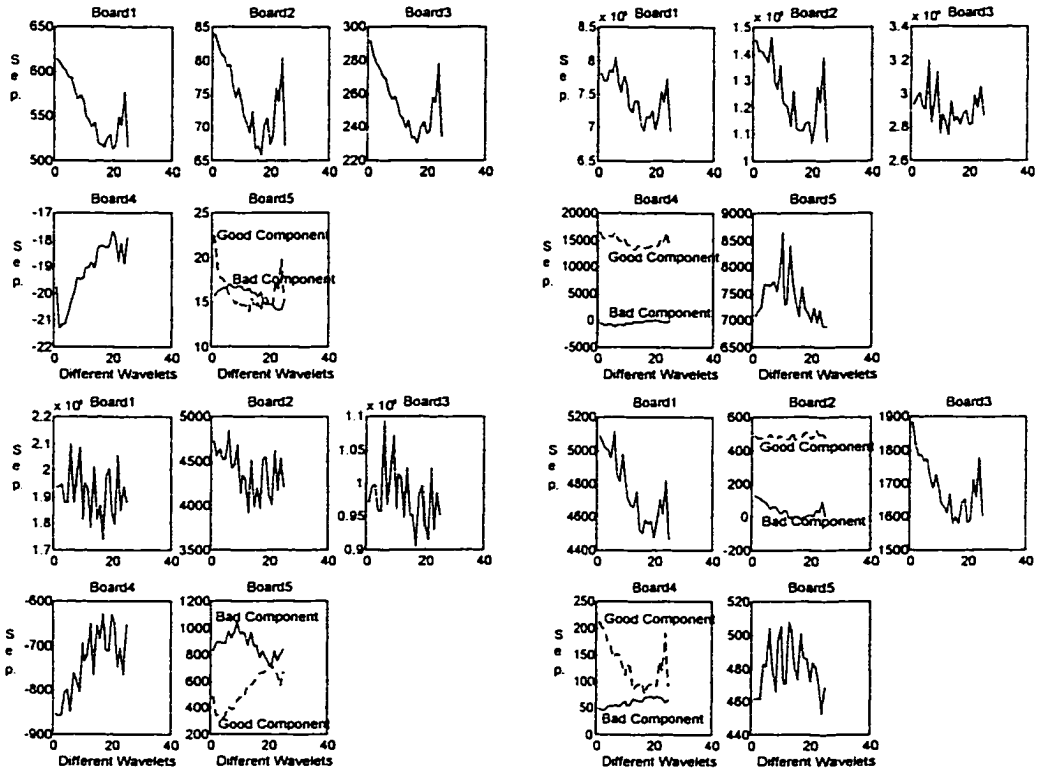



Figure C.19. Separability plots for various wavelets using octave tree for second level and extracting 7 features.

Instead of extracting just seven features, the number of features extracted is increased to 28 to see the effect of using a two-level octave tree on the FDI system performance. The 28-feature extraction strategy is shown in Figure C.20. One feature is extracted from each region. Figure C.21 shows the variation of the separability across the wavelets for all measures. As seen earlier, increasing the number of features did not improve the FDI system performance.

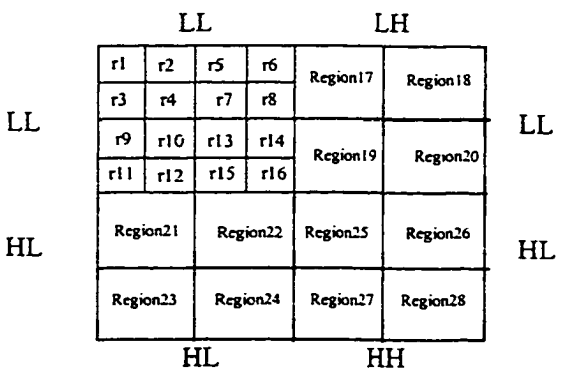



Figure C.20. A 28-Feature extraction strategy from two-level octave tree

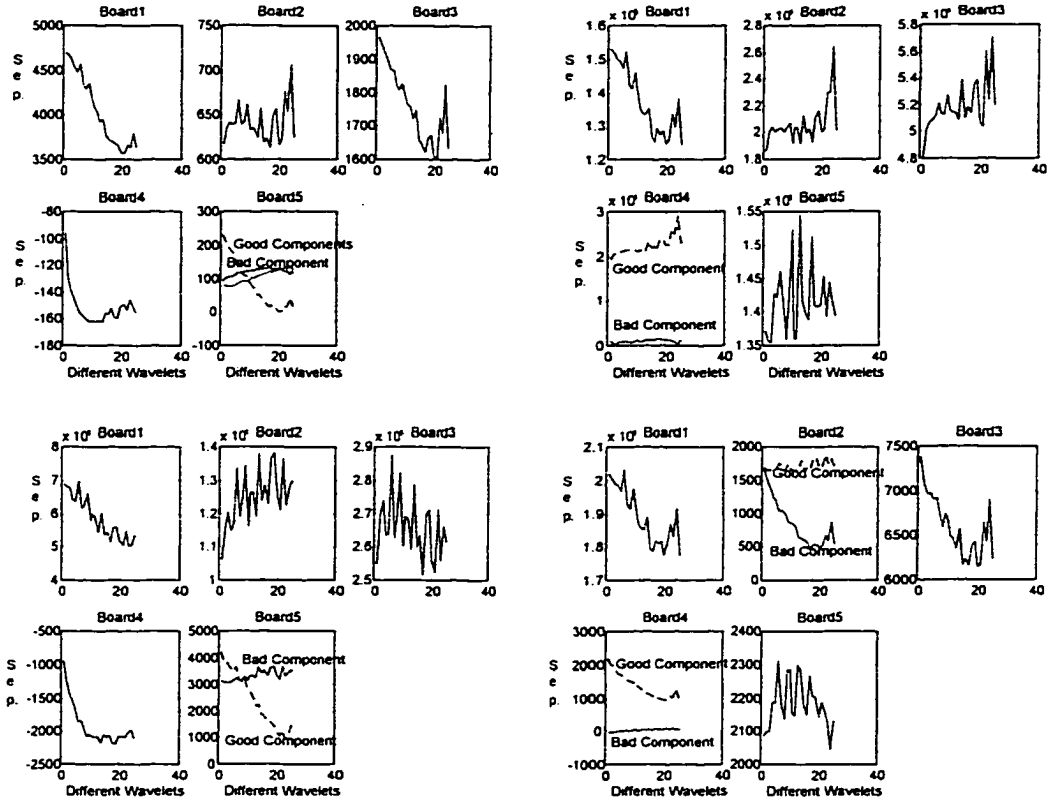



Figure C.21. Separability plots for various wavelets using octave tree for second level and extracting 28 features.

## C.4.4. INFLUENCE OF WAVELET BASES ON FDI SYSTEM PERFORMANCE USING ENERGY-BASED TREE DECOMPOSITION

The typical pyramid-type wavelet transform recursively decomposes the signals in the low frequency subbands. If the faulty components do not have most of their significant information in the low frequency region, further decomposition just in the lower frequency region like the conventional wavelet transform may not help much for the purpose of classification. To avoid a full decomposition, the tree is decomposed based on the energy of the node. This tree decomposition identifies the energy dominant subbands. The detailed energy-based tree decomposition algorithm is described in Section 4.2.

The faulty components are decomposed until the subband size is equal to 16. This results in a 3-level tree structured wavelet transform. However, all the components under test have only their low frequency subbands decomposed, i.e. the tree structured wavelet transform has become the octave tree. This is due to

the large amount of low frequency content of the circuit components. This means that all the components have most of their energy in the low frequencies.

To classify the test component, the distances between the test component and the average good component (class-1) and the average bad component (class-2) are calculated using the features at the nodes of the 3-level octave tree. Both classes (good and bad) have four components that are used in training and classification. The classification is performed using the leave-one-out algorithm. The classification experiment is conducted for all the feature extraction measures, i.e. the F-norm, $L_1$-norm, $\Delta$-norm and absolute sum. All the measures and wavelets classified all the four components correctly except when the $\Delta$ -norm was used, in which case the Daubechies wavelet could find the faulty components correctly only 3 times while the optimal wavelet correctly identified all faulty components.

The separation between the bad component and the good component in a bad board from their counter parts in a good board is used to quantify the degree to which the optimal wavelet performed better than the others. This is shown in Figure C.22. The measures $L_1$-norm and absolute sum seem to perform well compared to F-norm and $\Delta$-norm. The separability values are summarized in Table-C.3. The separability is increased by as much as 13% from the Daubechies to the optimal wavelet.

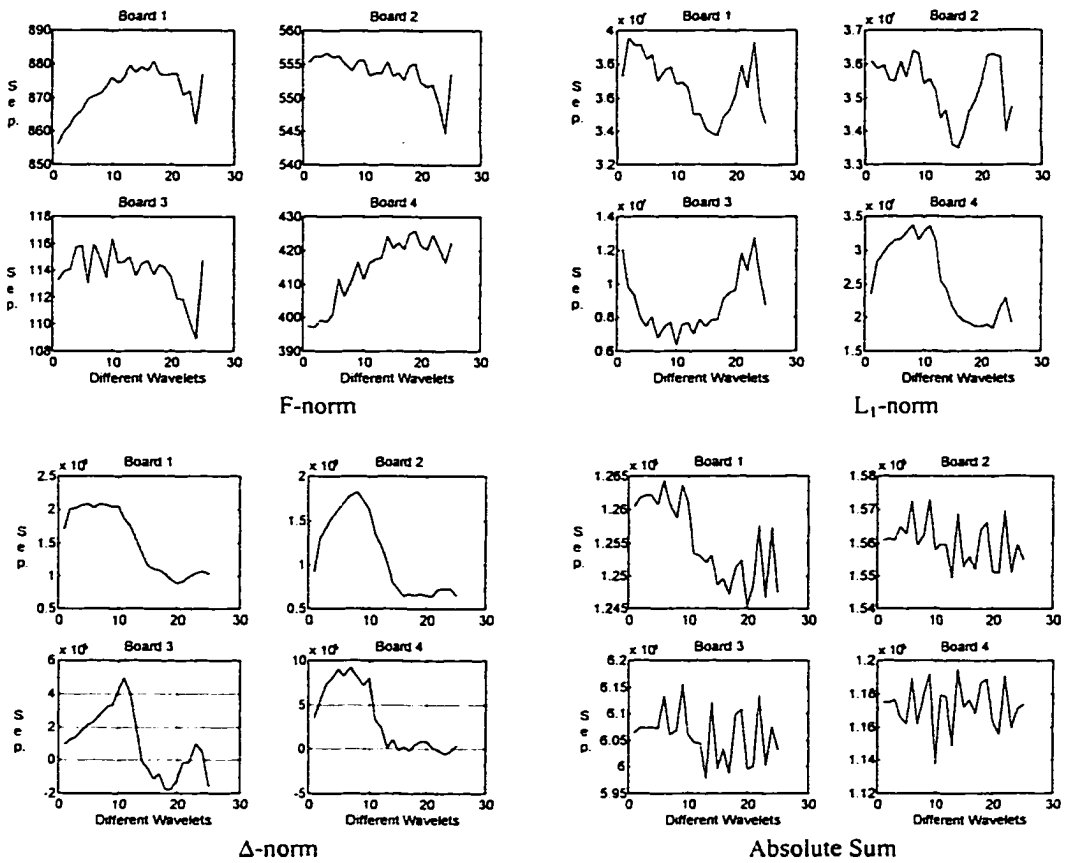



Figure C.22. Separability plots for various wavelets using 3-level energy-based tree

TABLE-C.3: SEPARABILITY STATISTICS USING 3-LEVEL ENERGY-BASED TREE FOR VARIOUS WAVELETS

| | Max. | Min. | Mean | Median | Daubechies |
|---|---|---|---|---|---|
| F-norm | | | | | |
| | 880.64 | 855.96 | 872.14 | 874.41 | 877.25 |
| | 556.68 | 544.56 | 553.77 | 554.05 | 553.57 |
| | 116.41 | 108.94 | 113.88 | 114.17 | 114.77 |
| | 425.72 | 396.99 | 414.53 | 417.52 | 422.30 |
| $L_1$-norm | 1.0e+007 * | | | | |
| | 3.96 | 3.36 | 3.66 | 3.67 | 3.44 |
| | 3.64 | 3.35 | 3.53 | 3.55 | 3.47 |
| | 1.27 | 0.63 | 0.88 | 0.80 | 0.875 |
| | 3.38 | 1.83 | 2.56 | 2.44 | 1.92 |
| Δ-norm | 1.0e+006 * | | | | |
| | 2.08 | 0.88 | 1.52 | 1.57 | 1.03 |
| | 1.83 | 0.63 | 1.09 | 0.93 | 0.64 |
| | 0.50 | -0.18 | 0.11 | 0.10 | -0.16 |
| | 0.93 | -0.06 | 0.33 | 0.11 | 0.04 |
| Absolute sum | 1.0e+005 * | | | | |
| | 1.26 | 1.25 | 1.255 | 1.25 | 1.25 |
| | 1.57 | 1.55 | 1.56 | 1.56 | 1.555 |
| | 0.62 | 0.60 | 0.61 | 0.61 | 0.60 |
| | 1.20 | 1.14 | 1.17 | 1.18 | 1.17 |

## C.4.5. INFLUENCE OF WAVELET BASES ON FDI SYSTEM PERFORMANCE USING SEPARABILITY-BASED TREE DECOMPOSITION

In Section C.5.3., dominance of the energy concentration in a sub-band is used as a criterion for further decomposition. However, for fault identification purposes, a criterion based on the differences between the patterns (signals or components) of different classes good and bad, i.e. class separability, is preferable. The components may be quite similar in the dominant energy bands and quite different in the low energy bands. Then, these low energy bands have significant information about the differences in the good and bad components that is needed for fault identification. The algorithm developed in Section 4.4.2 by selecting the tree basis depending on the class separability or discrimination is used.

To study the influence of the mother wavelet on the FDI system performance using the separability-based tree structure, each mother wavelet is used and the separability-based tree decomposition is applied to the boards in the database. The database has 4 good boards and 4 bad boards. The number of boards available is not sufficient to calculate the separability, especially when the number of features extracted is

greater than 4. So, 12 more boards are added to each class of boards (good and bad), by taking the previous and next frames of the current frame (image) as different boards belonging to the same class. For example, the image in the $50^{th}$ frame is used as the original board and the $49^{th}$, $51^{st}$, and $52^{nd}$ frames are used as different boards belonging to the same class. This is only valid when the frame acquisition rate (number of frames acquired per second) is high, since the difference in the heat index of the frames will not be significant in this case.

Using each wavelet, the separability based tree decomposition is performed for the second level. Many wavelets yielded a full tree, i.e. 16 bands, and some yielded as few as 10 bands. Only 10 bands exist due to the reason that the other 6 bands did not carry any information useful for separability. Sixteen (or ten) features are extracted from the sixteen (or ten) bands using F-norm or $L_1$-norm or $\Delta$-norm or absolute-mean. These final sixteen features are used for fault identification of the boards. For all the wavelets, the fault identification rate is 100 percent. Thus, for all the 16 bad boards, the faulty component is identified correctly. However, to know the performance improvement using various wavelets, the separation between the bad and good components of corresponding bad and good boards is compared. The higher the separation, the better the confidence with which the decision is made regarding the faulty component in the bad board. The total separation of the 16 boards is plotted in Figure C.23 as a function of various wavelets for a level-2 decomposition. Four subplots are drawn for the feature extraction measures $L_1$-norm, F-norm, $\Delta$-norm and absolute-mean, consecutively.

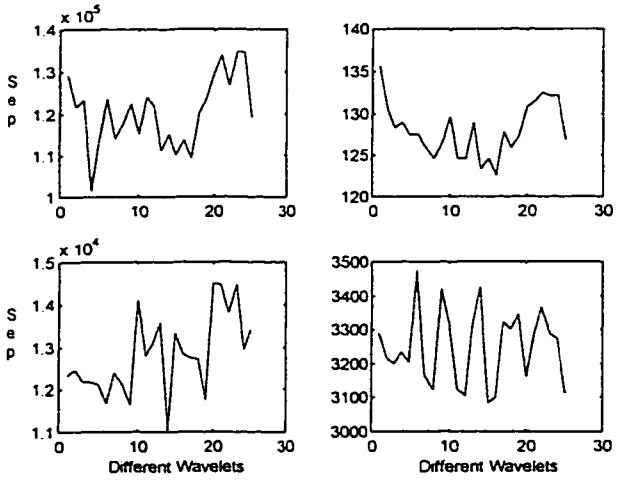



Figure C.23. The total distance between the good and bad components for all 16 boards using second level of decomposition for various wavelets

The separation or the distance between the good and bad components is high for the optimal or sub-optimal wavelet compared to the Daubechies wavelet. This is due to the increased separability between the good and bad board clusters of the sub-optimal wavelet compared to that of the Daubechies wavelet.

Figure C.24 shows the same data for a one-level decomposition for various wavelets. The same improvement in performance can be observed in this case. Figure C.25 shows the plots for a second level decomposition using various wavelets with dimensionality reduction. The 16 features extracted from the 16 bands are dimensionally reduced to one feature that is used to calculate the distance or separation between the good and bad components. The separation between the good and bad components is considerably higher for the optimal wavelet compared to the Daubechies wavelet using only one feature.

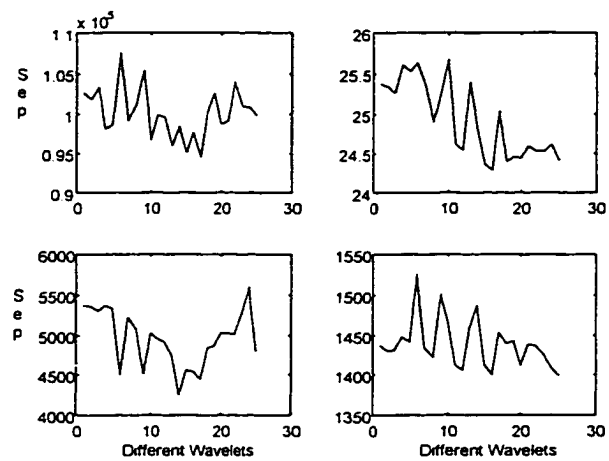



Figure C.24. The total distance between the good and bad components for all 16 boards using one level of decomposition for various wavelets

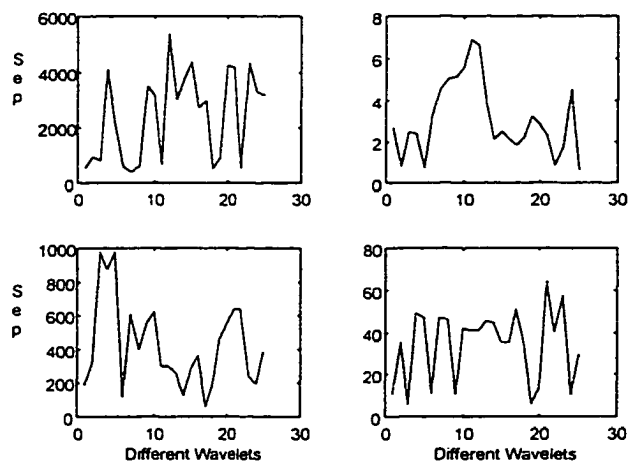



Figure C.25. The total distance between the good and bad components for all 16 boards using second level of decomposition with dimensionality reduction for various wavelets

## C.5. CLASS SEPARABILITY

From all previous experiments, one can observe and conclude that class separability is a useful measure for identifying the faulty components. We are interested in maximizing the distance or separation between the bad and good components. The wavelet with the highest distance is the optimal wavelet. If the class separability is also the highest for this wavelet, where the distance is the maximum, then the optimal wavelet can be designed by maximizing the class separability which in turn maximizes the distance. Figure C.26 shows the relation between the class separability and distance, and the between-class separability and distance. It can be observed that the relationship is nonlinear in both cases. The distance is the highest where the class separability is maximum or near maximum. So, the wavelet with the highest or near highest class separability is the optimal wavelet that results in the maximum distance between the good and bad components.

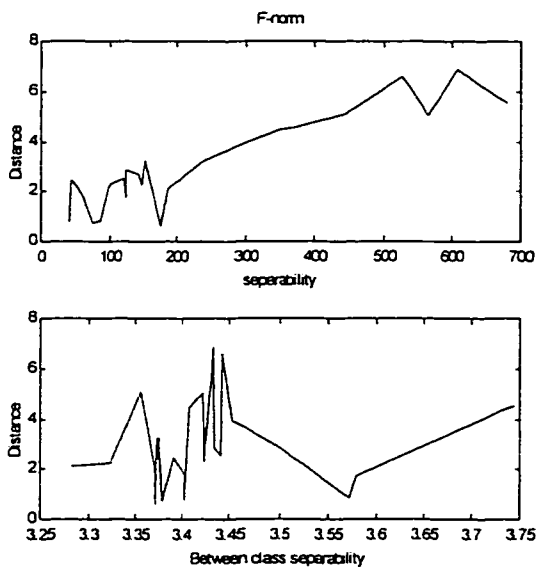



Figure C.26. The total distance between the good and bad components for all 16 boards versus the class separabilities using second level of decomposition and dimensionality reduction for various wavelets

## C.6. DESIGNING WAVELET BASIS FUNCTIONS FOR FAULT IDENTIFICATION

The optimal wavelet is obtained by maximizing the class separability using the simulated annealing algorithm. The detailed simulated annealing algorithm is described in Section 6.5. The distance between the good and bad components for the optimal wavelet is calculated. This is done for the F-norm, $L_1$-norm and $\Delta$

-norm measures, with and without dimensionality reduction, and using a second level of wavelet decomposition. These distances are listed in Table C.4. The $L_1$-norm resulted in the maximum separation between the good and bad components in a bad board compared to other measures used. The dimensionality reduction techniques did not help improve the separation between the good and bad components. This may be due to the low correlation of the features extracted from the components.

**TABLE-C.4: AVERAGE DISTANCE BETWEEN GOOD AND BAD COMPONENTS FOR ALL BOARDS FOR SECOND LEVEL OF DECOMPOSITION USING THE OPTIMAL WAVELET**

| Feature Extraction Method | Distance | Distance using dimensionally reduced features |
|---|---|---|
| F-norm | 12353.0 | 292.63 |
| $L_1$-norm | 116830.0 | 7966.1 |
| $\Delta$-norm | 125.13 | 6.40 |

## C.7. CONCLUSION

Optimal sub-band filters are designed for circuit components with increased separation between the good and bad components. This in turn increases the confidence in the decision made regarding the faulty component in a bad board. The relationship between the classification rates and the class separabilities are shown. The advantage of having the class separability as the cost function is that it takes care of both the within-class and between-class scatter. The features extracted using the designed optimal subband filters based on the class-separability have all the information required for class separation and classification. An optimal filter is designed for a set of components. The set has two classes, good and bad components. The classification performances are compared for different tree structures (octave tree, energy based tree, and separability based tree). In addition, the impact of extracting differing numbers of features is studied along with our varying of the feature extractor. Also, the effect of decomposing the optimal wavelet into more levels is studied and compared. The dimensionality reduction did not play a major role in obtaining higher classification performance or class separation, as it did in the case of texture classification

The simulated annealing algorithm is ideally suited for finding the optimal subband filters for classification. The simulated annealing algorithm yields maximal global class separability. The parameter settings needed in the simulated annealing algorithm are empirically chosen for the application. These parameters yield good global performance with an optimum trade off between the number of iterations required to reach the global maximum and the final stopping error. The convergence characteristics of both the class separability and the control parameter confirm the stability of the algorithm.

# APPENDIX D

# MORE PLOTS USING THE ANALYSIS IN APPENDIX C FOR ANOTHER SET OF BOARDS

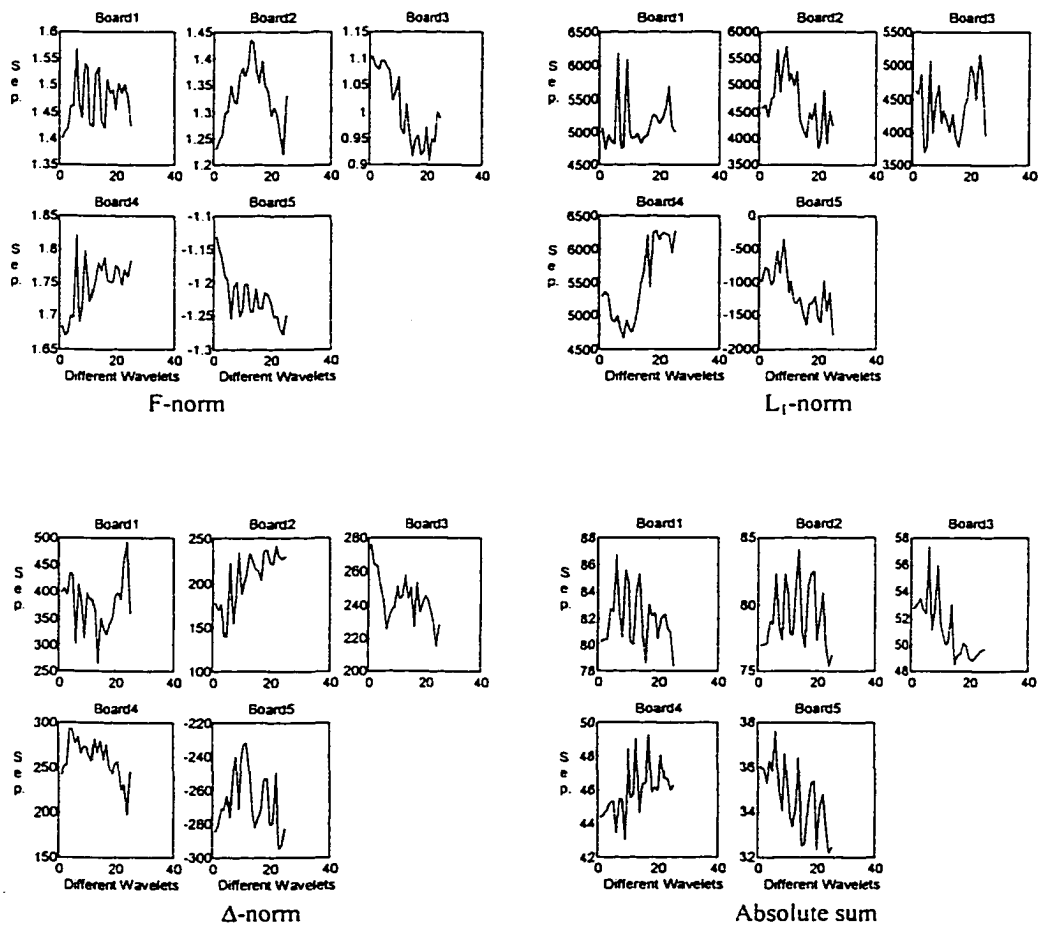



Figure D.1. Separability plots for various wavelets using F-, $L_1$-, Δ-norms and absolute sum of DWT coefficients using 4 features with one level of decomposition

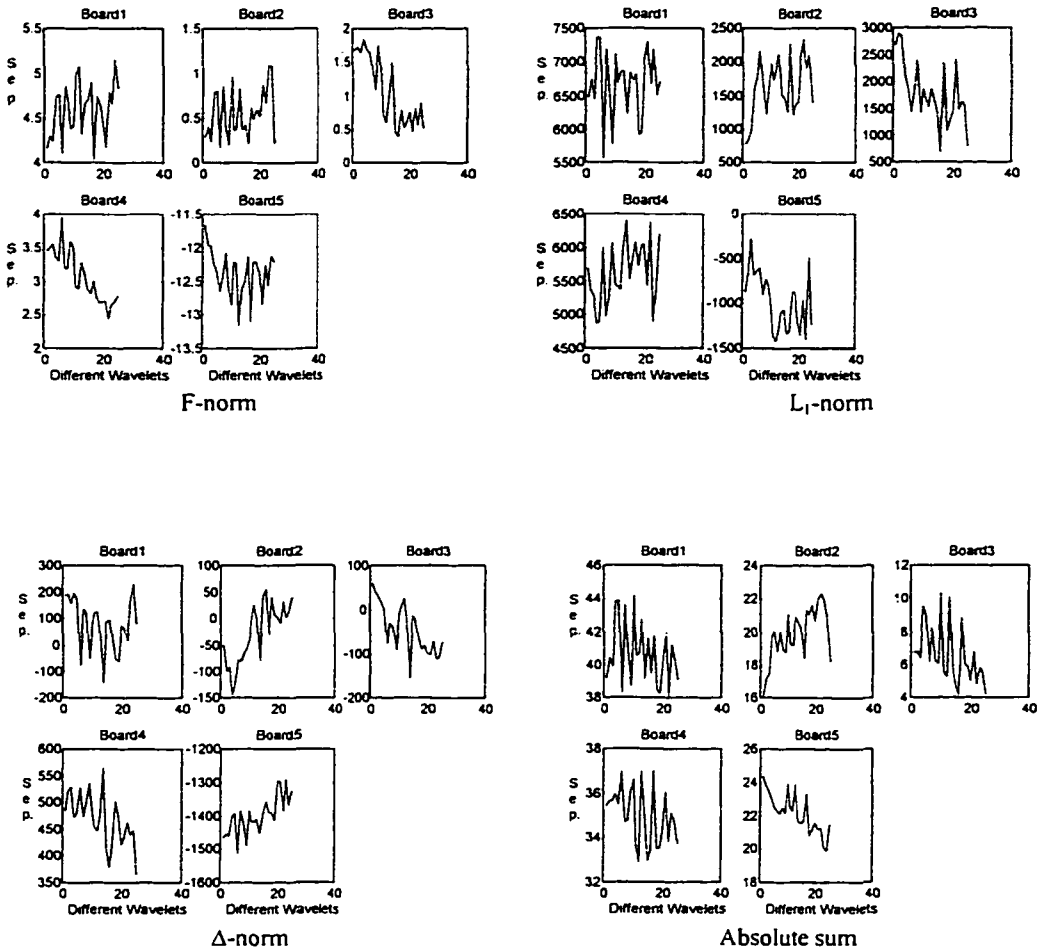



Figure D.2. Separability plots for various wavelets using F-, $L_1$-, Δ-norms and absolute sum of DWT coefficients using 16 features with one level of decomposition

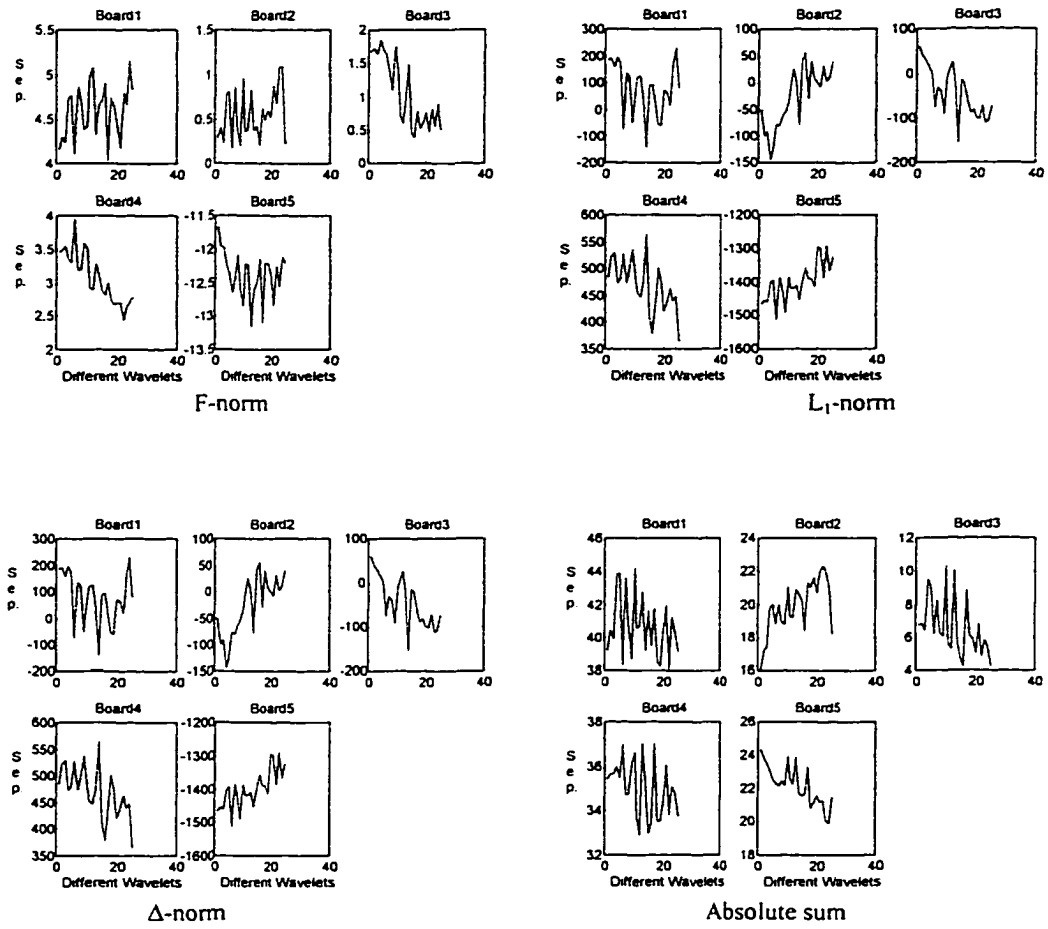



Figure D.3. Separability plots for various wavelets using F-, $L_1$-, Δ-norms and absolute sum of DWT coefficients using 16 features with second level of decomposition

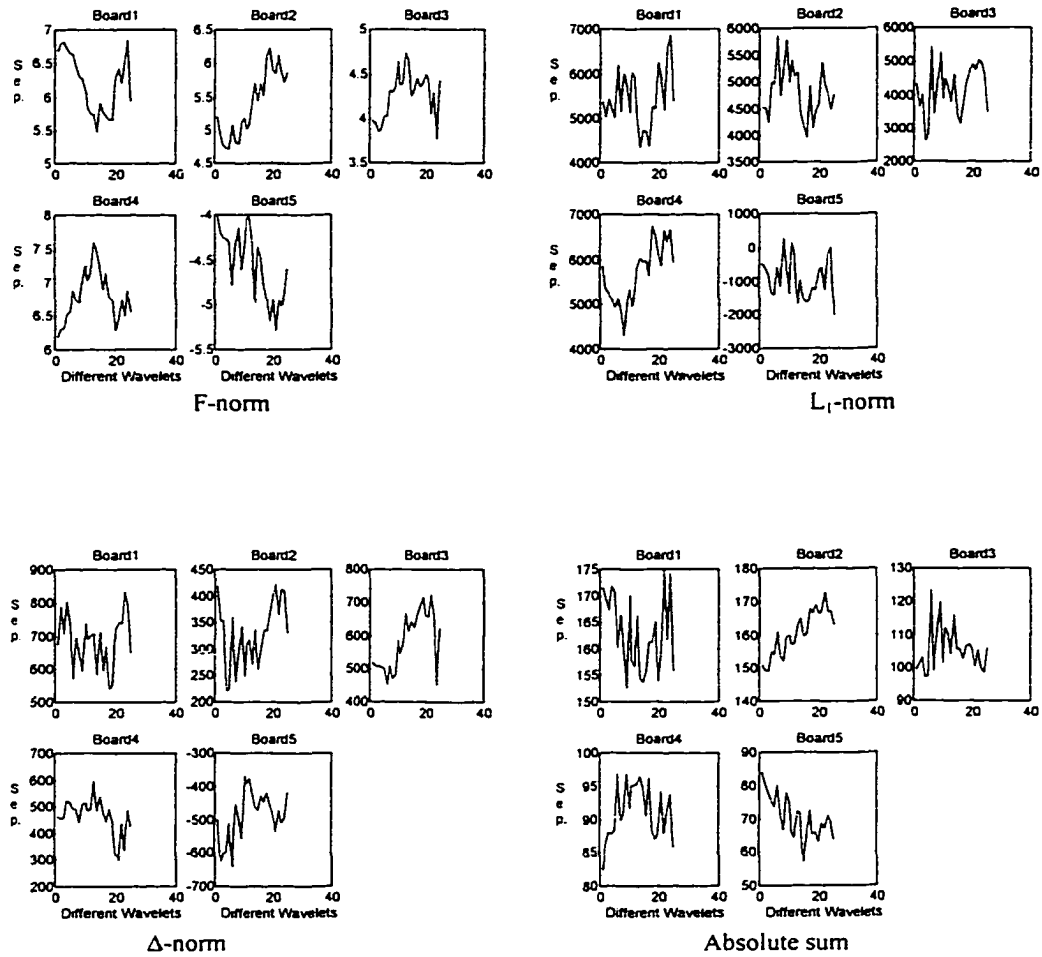



Figure D.4. Separability plots for various wavelets using F-, $L_1$-, Δ-norms and absolute sum of DWT coefficients using 7 features with second level of decomposition and octave tree structure

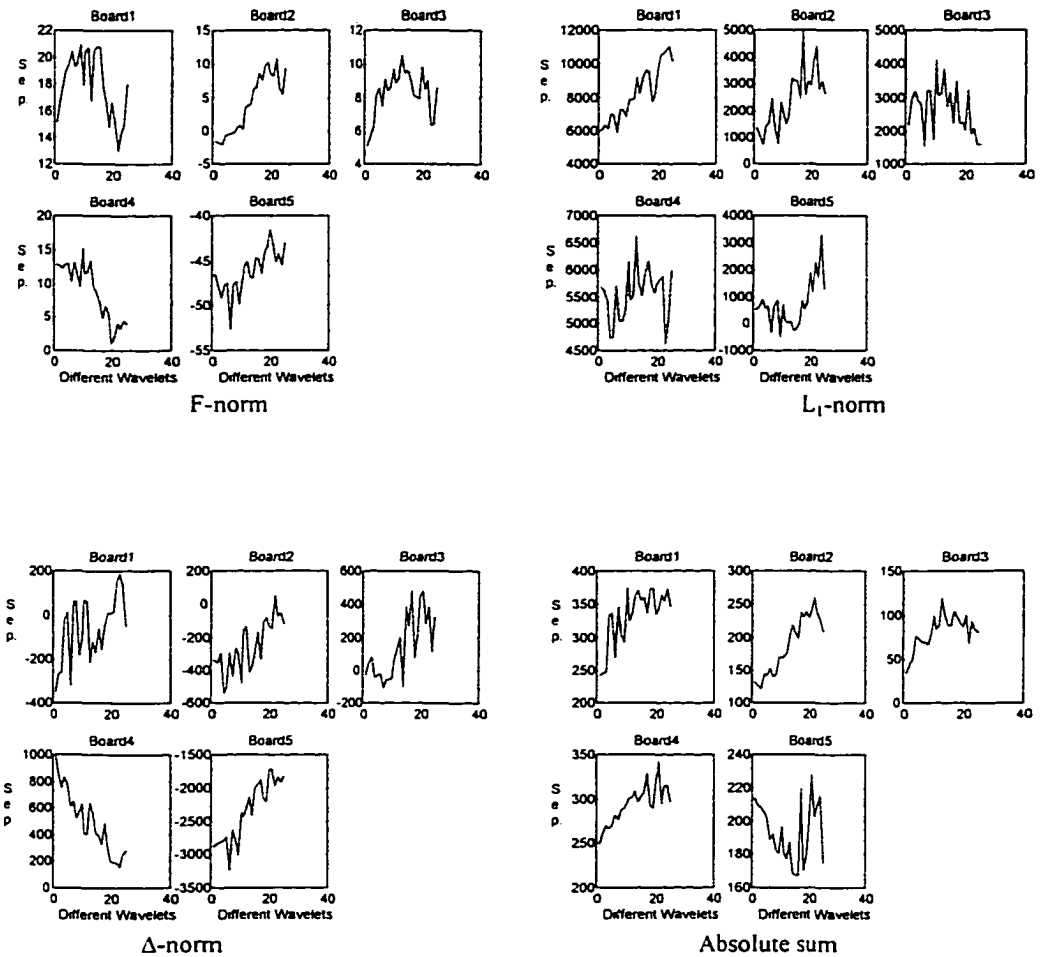



Figure D.5. Separability plots for various wavelets using F-, $L_1$-, Δ-norms and absolute sum of DWT coefficients using 28 features with second level of decomposition and octave tree structure

# VITA

Madhavi Kadiyala received the B.Tech. degree from VR Siddhartha Engineering College, Vijayawada, India in 1990 in Electronics and Communications Engineering, the M.S. degree from Birla Institute of Technology, Mesra, India in 1992 in Microwave Engineering and the Ph.D. degree from the University of Oklahoma, Norman, Oklahoma, the United States of America in 1999 in Electrical and Computer Engineering.

From 1992 to 1993, she was a senior research fellow in the Department of Electronics and Communications Engineering, VR Siddhartha Engineering College, India. From 1994 to 1999, she was a research assistant and teaching assistant in the School of Electrical and Computer Engineering, the University of Oklahoma.

Her research interests include image/video processing, digital signal processing, multimedia communications, and fault detection and identification.