# Object Recognition in Compressed Imagery

Thesis by

Cynthia Evors Daniell

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy


California Institute of Technology

Pasadena, California

2000

(Defended 5 May 2000)

# Acknowledgements

This work has been anything but a solitary endeavor. My sincere gratitude goes out to Abhijit Mahalanobis and Roy Matic for their generosity, guidance, and many technical discussions. I would also like to thank Hayit Greenspan for introducing me to the Microsystems Lab at Caltech, my advisor, Professor Rod Goodman, and the professors who served on my committee: Professors Yaser Abu-Mostafa, Christof Koch, Demetri Psaltis, and P.P. Vaidyanathan. To my parents, I am deeply appreciative of their continuous support and encouragement through the years. Finally, I am forever grateful to my dear husband, Michael, for his unlimited love and support.

This work was supported by a fellowship from Hughes Research Laboratories.

# Abstract

It is often necessary to search for objects in large databases of compressed imagery. In the past, object recognition and image compression have generally been treated as separate problems, resulting in inefficient suboptimal performance. Moreover, computational and storage issues make it fundamentally prohibitive to uncompress large images prior to object recognition. We provide two complementary solutions to the problem of object recognition in compressed imagery, each of which integrates subband and correlation filtering in a unique manner.

One key benefit of correlation filters is that, as linear systems, they are highly compatible with the subband filtering process. This enables us to provide a seamless operation in which object recognition and data compression are viewed as continuations of the same process. The public MSTAR data set illustrates our results on a three class problem of 79 Synthetic Aperture Radar images at one foot resolution.

Our general framework, the Pattern Recognition Subband Coder (PRSC), provides simultaneous synthesis and recognition at full resolution in a computationally efficient architecture. Its parallelism enables a result 1.6 times faster, in the limit, than correlation on uncompressed imagery. Furthermore, by jointly optimizing the synthesis and recognition filters, the PRSC achieves 100% recognition accuracy on our compressed

data set, improving performance over that produced from the original (uncompressed) data set, by 3.7%. We maintain this success for compression ratios up to 6:1.

Addressing the issue of reduced resolution recognition, our Subband Domain Correlation Filters operate directly on the subband coefficients at multiple resolution levels. For compression ratios of at least 20:1, we achieve recognition performance of at least 90%, 85%, and 75%, respectively, on two, four, and eight foot resolution data.

Thus, through our solutions with compressed imagery, we outperform correlation results on the equivalent original imagery in terms of both speed and accuracy, as well as provide success at reduced resolutions of the data.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Multimedia, high speed data transmission, and the world wide web have made visual imagery a large part of the computing experience. Images are now ubiquitous in many consumer, industrial and military applications which store and catalog them in various databases. Due to the large storage requirements of typical image data, most applications store them in a compressed format. That is, the application transforms the image data into a new representation which requires less space in a computer's memory bank and thus allows for more compact storage. The new compressed format is typically a mathematical representation with little similarity to the visual characteristics of the original image. Many applications, however, require the computer to search through a large image database quickly selecting images containing specific objects of interest to the user. Therein lies the question addressed by this thesis, "*Given a set of compressed images, how can we quickly and accurately find the ones we want?*"

The material presented in this thesis is motivated by the goal of jointly optimizing the three system components of recognition, compression, and implementation, thus providing an integrated approach to the problem of object recognition in compressed

imagery. The driving factors for system performance are 1) accurate object recognition, 2) high compression rates, and 3) low orders of computation.

## 1.1 Statement of the Problem

Our trilateral approach elicits the following specific research questions, which are addressed by this thesis.

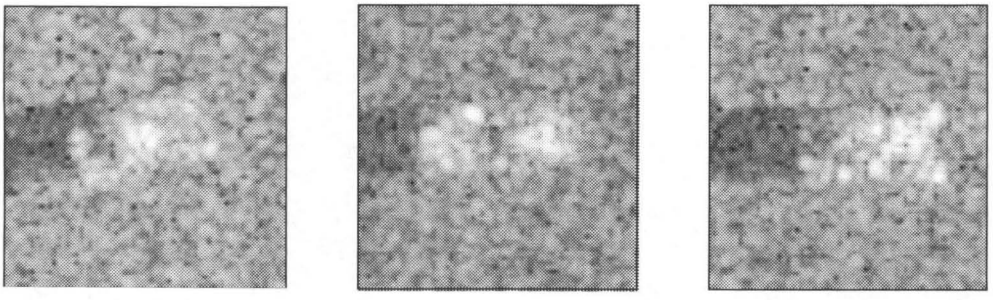
- "*Can we merge an object recognition architecture with a compression architecture in a computationally efficient manner?*"
- "*Can we perform object recognition in compressed imagery with the same accuracy as in uncompressed imagery?*"
- "*How does the accuracy of object recognition in compressed imagery degrade with the amount of compression?*"
- "*Can we perform object recognition at reduced resolutions of the compressed imagery?*"
- "*Can we provide a system which allows for tuning between the two objectives of image compression and object recognition?*"

### 1.1.1 The Problem of Object Recognition

Finding and recognizing a particular object or pattern in an image may seem like an easy task to a human, however, a robust and universal computer solution has proven elusive to numerous scientists for many years. The task involves using a computer to recognize a particular pattern of bits within a sea of distracting bits. The difficulty lies in the physical variability of the three major components of an image scene: the objects themselves, the sensors, and the surrounding environment. Several books and hundreds

of journal papers [Pra78, Mar82, Hor86, Jai89, GW93], discuss these issues in great detail. In the interest of brevity, only a few major points are reiterated here.

Addressing variations among the physical objects themselves is a difficult problem in itself. First, an object recognition algorithm must account for variations in the object's location within an image as well as both in-plane and out-of-plane rotations of an object. The object may not always appear at the same position within a scene or at the same perspective. An object may also appear in multiple types of scenes and an object recognizer must be able to handle the different scene backgrounds. Additionally, the object recognition system must be able to reject scenes that do not contain any object that it has been trained or programmed to recognize. Furthermore, the distinguishing characteristics which separate one class of objects from another may not easily be programmed into a computer algorithm. Figure 1.1 provides an example of each object class used in this thesis. Three military vehicles make up the object classes: a BMP Bradley fighting vehicle, a BTR armored personnel carrier, and a T72 tank.



(a) BMP (b) BTR (c) T72

Figure 1.1 **Images of the Object Classes: BMP, BTR, and T72**

The image data employed in this thesis is from a Synthetic Aperture Radar (SAR), and is shown in the above figure in its original uncompressed state. Figure 1.1 underscores the difficulty in differentiating one type of object from another, even for

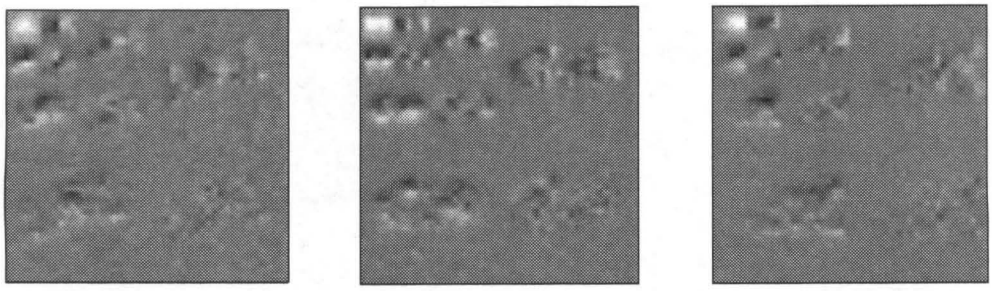
human observers, and even with visually recognizable uncompressed images. Out-of-plane rotations and translations by the objects under investigation further exacerbate this difficulty. These issues, as well as a low rejection rate, are the main focus of the recognition part of this thesis investigation. Section 2.5.1 further elucidates the characteristics of the imagery used in the thesis.

### 1.1.2 The Problem of Object Recognition in Compressed Imagery

The main focus of this thesis is not to make advances in the state of the art in object recognition per se, but to perform current levels of object recognition in compressed imagery. Why would one want to perform object recognition in compressed imagery? As mentioned earlier, memory and computational speed requirements drive the need for such a system. Images take an enormous amount of memory, hence they are typically stored in a compressed format, yet many applications need very quick searches of large image databases.

Figure 1.2 provides a compressed representation of each object class used in this thesis. We exhibit their uncompressed counterparts in Figure 1.1. The compressed representation shown is a subband decomposition. It is a mathematical representation of the original image. Figure 1.2 illustrates the much greater difficulty in discriminating between object classes once the objects are stored in the compressed domain.

Finally, we note that image compression has typically served the purpose of visual reconstruction. That is, compression quality is currently scored with measures like the Peak Signal to Noise Ratio (PSNR) and mean squared error (MSE) of the reconstructed image. Researchers derived both PSNR and MSE to reflect the visual quality of the with



(a) BMP (b) BTR (c) T72

Figure 1.2 **Compressed Images of the Object Classes: BMP, BTR, and T72**

reproduction image. One must remember that the goal of image compression for object recognition is not necessarily the same as the goal of image compression for visual reproduction quality. This thesis is interested in image data compression for content manipulation rather than content visualization.

## 1.2 Prior Developments in Object Recognition and Image Compression

Typically, object recognition and image compression are treated as separate problems. Thus, we provide a little background on the history of both fields. There is such a large body of literature on both object recognition and image compression, that, out of necessity, we have restricted the discussion to the most effective and relevant models in Section 1.2.1 and 1.2.2.

The purpose of this thesis, however, is not to advance either specialty, but to join the two, so that object recognition is now performed directly in the compressed image domain. Researchers have only recently addressed this issue; we provide the references and background to the prior work in Section 1.2.3.

### 1.2.1 Object Recognition

Object recognition techniques generally fall into two categories: 1) those that operate directly on the data, and 2) those that extract features from the data. Classical computer vision techniques began with the former approaches in the 1970s, with both filtering and statistical models. Today, the field is characterized by a shift towards feature extraction paradigms. The feature extraction methods form two branches - those that use learning algorithms to classify the feature vectors, and those that develop semantic models of the features. The model based approaches then try to match the acquired model with several representative models stored apriori.

**Template Matching**

Some of the earliest work in object recognition grew out of the matched filter originating in the field of communications [Nor43, Tur60]. It is well known that a matched spatial filter is optimum for the detection of an object in a single image in the presence of additive white noise [Nor43]. In such a system, a cross-correlation is performed between the stored filter and the image in question. If the correlation exceeds a predetermined threshold, then the image is declared to contain the object of interest.

Matched spatial filters were first applied to the problem of character recognition in [MED56, Hig61], and to the more general problem of object recognition in [CL+60]. Their use grew throughout the 1960s and 70s, with a good discussion and synopsis given in [And70, DH73, Pra78]. Then Casasent and Psaltis [CP84] showed how to make correlation filters invariant to position, rotation, and scale through a log polar transform. This procedure, however, is difficult in practice because it requires registration of the input image. Thus, unfortunately, wide variability in an object's characteristics make the

matched spatial filter a very brittle method of object recognition. Matched spatial filters, however, provided the basis for the evolution of the more sophisticated correlation filters.

**Correlation Filters**

Correlation filters are specially formulated generalizations of the matched spatial filter which are able to recognize many different views of an object in the presence of signature variations and in clutter. They were born out of Synthetic Discriminant Functions (SDFs), which themselves evolved from linear combinations of matched spatial filters [HC80]. A good introduction to correlation filter theory is given in Vijaya Kumar, Mahalanobis and Juday [VKMJ00] and in Vijaya Kumar [VKu92]. The two most important steps in the evolution of correlation filters came when Vijaya Kumar [VKu86] introduced minimum variance SDFs to maximize SDF noise tolerance, and when Mahalanobis, et. al., [Mah87, MVK+87] introduced Minimum Average Correlation Energy (MACE) filters capable of producing sharp correlation peaks at the same location as the shifted input.

In general, correlation filters are composites of several training images representative of a particular object, thus, mitigating the brittle characteristics of a matched spatial filter. Moreover, correlation filter theory suggests that the Minimum Average Correlation Height (MACH) filter is statistically optimum for detecting objects in additive noise, when Gaussian assumptions hold [MVK97b]. However, correlation filters are still not robust enough to perform recognition over wide variations of an object. In general, multiple correlation filters are formed for different aspects of an object.

**Statistical Models**

Measures of statistical information in an image can also provide signatures useful in object classification. Typical measures are derived from image histograms and from spatial frequency spectra. Examples of these measures are the mean, variance, skewness, kurtosis, energy, and entropy of an image histogram. If segmentation is performed on the objects within the image, then shape statistics, such as moments, eccentricity, major axis orientation, area, and curvature, may also be generated. The most recent and noteworthy example of using a statistical model (employing histogram statistics) with SAR imagery is given in [Sim99]. SAR imagery, however, is notorious for spectral shadows and missing spectral information. These traits are difficult to overcome with statistical models.

**Principal Components Analysis**

The method of Principal Components Analysis (PCA) was introduced in the statistics literature by Hotelling in the 1930s [Hot33]. PCA provides a way to describe the directions of maximum variation in multidimensional data. In addition, only a small number of principal components are necessary to form a well-defined representation of an image. In essence, the PCA approach specifies a basis set which forms a model of object classes. The PCA approach has been applied to several problems in the image domain [SK87, TP91, BP93, BFP+94, MN95].

**Deformable Models**

Finally, we come to a more recent method of object recognition. To represent a latitude of variations within an object class, features of a particular object class can be arranged in a deformable spatial configuration (a.k.a. model) to form a syntactic

representation of an object class. Objects under consideration in an image are then matched to the appropriate parts of the stored model. Features appropriate for a syntactic model vary depending on the application. One application of deformable models is face recognition which may use the eyes, nose, or mouth corners for a face model. Recent work which uses deformable models for object recognition can be found in [Yui91, L+93, WvdM93, PL95, Sha95, Bur97]. The main detriment of deformable models is that they are only as good as the feature detectors that provide the inputs. A missing or erroneous feature can cause the model to hallucinate. In addition, with a large number of features, the search space for a correct model match can be computationally prohibitive.

**Object Recognition in SAR Imagery**

Synthetic Aperture Radar (SAR) imagery forms a very specific domain of object recognition. To date, successful object recognition in SAR imagery primarily involves template matching and correlation filters. Matched spatial filters (a.k.a. template matching) were first applied to SAR imagery in the mid 1980s [LM85, BK88]. Currently, both Novak [NO+97, NO+98, NO+99] and Hofstetler [Hos99] perform cross-correlation with template filters. The template that has the lowest mean squared error with the object in question determines the classification of the object. These templates are formed from a mean image, over 5° spans of each object within each object class. Thus, for 360° viewing aspects of each object class, the template method requires 72 templates for each class.

Correlation filters, while similar to template matching, form more global descriptions of the objects under consideration. They were first used on SAR imagery in [MF+94], while recently, Mahalanobis, et. al., [MO+99, MO+00] have shown good results with

SAR imagery by forming correlation filters over 45° spans of the object classes. This method requires only eight correlation filters per object class.

In addition, Simonson has done some work with statistical models [Sim99]. In this work, she forms a probability distribution of the dominant scatterers in the SAR imagery. The sensed probability distributions are then matched to a set of distributions stored apriori to determine the object classification.

Finally, Diemunsch and Wissinger, et. al., [DW98, WR+99] have applied a deformable model approach to SAR imagery. They form a syntactic relationship of the dominant scatterers from the radar return and try to match that to a group of stored models for object classification. A novelty of this approach is that through their Predict Extract Match Search (PEMS) technique, they can iteratively try to elicit an optimum classification from the object in question. In follow up work by Ettinger [Ett00], however, it is suggested that a better way of performing the match may be to use methods similar to cross-correlation to match an image domain representation of the apriori models with the sensed imagery.

### 1.2.2 Image Compression

The objective of image compression is to reduce the number of bits of information necessary to represent a given image by eliminating redundancy in the image or by introducing distortion into the image in a manner that is acceptable to the viewer, or in our case, to the object recognizer. Early work in image compression began with *predictive* techniques formulated by Oliver and Elias [Oli52, Eli55] and the Differential Pulse Code Modulation (DPCM) predictive method introduced by Culter [Cul52]. Transform coding, introduced by Andrews and Pratt [AP68, AP69, And70], was the next

big step in compression of two-dimensional data. Both Jain [Jai81] and Rabbani and Jones [RJ91] provide a good review of the early work in image compression.

The first popular image compression technique was the Graphics Interchange Format (GIF), developed by CompuServe Information Service to encode graphical images. It is a variant of the Lempel Ziv Welch (LZW) algorithm [Wel84, ZL77, ZL78], which is an adaptive dictionary technique similar to the 'compress' command in UNIX. While GIF works well with graphical images, it is generally not the most efficient method to compress natural images.

A current popular image compression technique, the Joint Photographic Experts Group (JPEG) method [Wal91], combines transform and predictive coding. It uses a block based transform coding procedure which consists of segmenting an image into typically square blocks, applying an orthogonal transform to each block, and then quantizing and coding the transform coefficients. Coding within a JPEG scheme can use one of eight different linear predictive techniques (including no prediction). The main detriment of JPEG coding is that it produces blocking artifacts at low bit rates.

Finally, the most recent image compression technique is subband coding. Detailed information about subband coding can be found in Vaidyanathan [Vai93], Vetterli and Kovacevic [VK95], and also in Strang and Nguyen [SN96]. The technique, first introduced for speech coding in the 1970s, centers around splitting an image into multiple frequency bands, and then coding the resulting coefficients. In that regard, it is similar to transform coding. Subband coding differs, however, in that it is not block based, but applies the filtering operation to the entire image. In addition, most subband coders perform the filtering operation recursively to each resulting low frequency subband,

producing a multiresolution decomposition [Mal89] of the image data. Subband coders became more popular and viable with the advent of perfect reconstruction, made independently by Vaidyanathan and Vetterli [Vai87a, Vai87b, Vet87].

A good summary of image compression techniques is given in Sayood [Say96]. In addition, Baxter and Siebert [BS98] provide a survey of several image compression techniques specifically applied to SAR imagery. Their preferred compression system for the highest perceived SAR image quality consists of a wavelet packet transform that uses a Gabor-like tree structure with smooth biorthogonal wavelet filters, followed by a universal trellis coded quantizer and an arithmetic entropy coder.

### 1.2.3 Combining Object Recognition with Image Compression

We uncovered very little previous work combining object recognition with compressed imagery. Our compilation of the literature references divides them into five categories: 1) full reconstruction of the compressed image, 2) multiresolution domain processing, 3) vector quantization domain processing, 4) unconventional compression techniques, and 5) database query.

#### Full Reconstruction of the Compressed Image

Current systems typically perform the reconstruction (i.e., the inverse of compression) and recognition tasks serially. They fully uncompress the stored images, then apply an object recognizer to the uncompressed data. Walls and Mahalanobis [WM99] use this method on SAR imagery; Liu and Mitra [LM96] follow the technique with fingerprint data; Shin and Kil [SK97] employ the paradigm with sonar images; visual images of tanks undergo the same approach by Miller [Mil95]. If the recognizer does not find the object of interest, then typically, the system transforms the image back

into its compressed state and moves on to the next image, repeating this process through the entire database finding all instances of the object. This is the most straightforward way of dealing with compressed imagery, and what we will call the *baseline compression recognition system* throughout this thesis. It is not computationally efficient, and limits the speed of the search.

In the first three instances of the previous work, however, they demonstrate high recognition rates for compression ratios between 50:1 and 200:1. In addition, the Walls and Mahalanobis paper [WM99] provides us with a baseline for comparison with our own experiments.

**Multiresolution Domain Processing**

A literature search revealed only two publications which acquire features from the subband coefficients of a multiresolution decomposition. Irving, et. al., [INW97] construct an autoregression model in scale, and use that as a feature vector for object detection. That is, the algorithm was not used for object classification, but rather to denote whether or not a region of an image possibly contained an object as opposed to background clutter. In addition, this algorithm does not account for the downsampling effects encountered in a subband decomposition and discussed in detail in Section 5.1.

The approach by Nahm and Smith [NS95] is also only used for object detection, not to discriminate between multiple objects. In this case, they vector quantize the subband coefficients, and use the resulting codewords to distinguish between background clutter and possible objects. This approach is akin to the Chaddha and Perlmutter, et. al., [CPG96, P+96], approach of processing directly in the vector quantization domain.

**Vector Quantization Domain Processing**

Vector quantization of an image provides a compression technique that is typically not as powerful as JPEG or subband coding, but is still quite popular. Chaddha and Perlmutter, et. al., [CPG96, P+96] devise a method of using the codewords resulting from vector quantization of an image to provide classification. It is a nice technique because only the indices of the codebook are needed for the classification label. This technique, however, has only been used to dichotomize an image with two classification labels, such as man-made and natural, or tumor and non-tumor. The method has not been extended to multiclass discrimination of objects.

**Unconventional Compression Techniques**

A few recent publications do combine unconventional compression techniques with object recognition models. These approaches, however, do not use the standard compression models used by most applications today. Principal Components Analysis is employed by [LW00] to perform both compression and recognition. [TJ97] uses an ad hoc compression technique based on image singularities which then form the basis for recognition. Multiresolution Principal Components Analysis is used by [BP98] to provide both compression and recognition. Fractal features form the basis for both compression and recognition in the work of both [Bon95] and [BBC98].

**Database Query**

A few systems, known as database query models, exploit compressed imagery to find objects *similar* to the true object of interest. This thesis addresses a different problem - that of exact object recognition. Additionally, database query models typically use unconventional compression techniques which do not apply to many image databases.

Pentland, et. al., [PPS95] use an independent semantics preserving image compression technique for their Photobook system. Vasconcelos and Lippman [VL97] use an independent library based coding and compression scheme for their image retrieval system. A group of linear basis sets is learned in Keaton and Goodman [KG99] which facilitates both compression and database query. Finally, Vellaikal and Kuo [VK96] use quadtree structure information along with JPEG coefficients for effective database queries. [KG99] can require partial decoding of the compressed data, while [VK96] requires full reconstruction of the image to compute the quadtree structure.

## 1.3 Thesis Approach

Central to the approach of this thesis is the joint selection of the compression and recognition components to provide an integrated system and facilitate an efficient implementation. The overarching purpose of the two models remains complementary. They must work in concert to form a synergistic system. As the image compression model reduces the number of bits used to represent a given image, it must introduce distortion into the image in a manner that is acceptable to the performance of the object recognizer.

Toward this end, our approach follows these objectives:

- To obtain the result directly in the compressed domain. That is, to avoid reconstruction of the image.
- To jointly optimize the performance trinity of recognition, compression, and computation.
- To provide results at both full and reduced resolutions of the data.

- To make accurate object recognition our primary objective, high compression rates secondary, and faithful reconstruction of the image ternary.

### 1.3.1 Algorithm Selection

Currently, most image compression applications use one of two techniques: JPEG coding or subband coding. With JPEG, the physical nature of separating the image into blocks breaks up any object under consideration in an image. This did not lend itself well to choosing an object recognition model, which would then have to find discriminating characteristics within each block or as some amalgamation across blocks. Furthermore, JPEG coding produces blocking artifacts at low bit rates, which are not conducive to object recognition.

We selected subband coding as the compression model for this thesis investigation. Subband coders provide a computationally efficient filtering system which results in a multiresolution representation of the image data. Multiresolution image representations provide a compact image representation as well as an efficient basis for image analysis [Mal89]. Furthermore, subband coders allow image reproduction to degrade gracefully at low bit rates making them appealing for integration with object recognition.

The image processing literature abounds with object recognition techniques, some of which are discussed in Section 1.2. Most importantly, the object recognizer used in this investigation must work cooperatively or invariantly with the downsampling intrinsic to a subband coder. For this thesis research, we selected correlation filters as the object recognition model. The primary reason is that, as linear systems, correlation filters are highly compatible with a subband coder architecture. This enables a *seamless* operation

in which object recognition and data compression are viewed as continuations of the same process.

It is important to utilize object recognition techniques that operate directly on the compressed data, and equally important to employ compression techniques that facilitate object recognition as well as a homogenous implementation. This framework facilitates the joint optimization of the performance of a subband architecture and the correlation filters.

### 1.3.2 Two Novel Solutions

Our quest to combine subband and correlation filters resulted in the development of two distinct yet complementary solutions to the problem of object recognition in compressed imagery. The underlying duality between the approaches of these two systems lies in the domain which forms the correlation filters for the object recognition. In our first solution, the Pattern Recognition Subband Coder (PRSC), the correlation filters are synthesized from the original image, and integrated seamlessly with the subband synthesis filters to form a new filtering architecture. The Subband Domain Correlation Filters (SDCFs) form our second approach by producing correlation filters from the subband coefficients and then operating directly on the subband information.

An additional duality lies in the objectives of these two systems. The PRSC processes compressed image data to provide an object recognition response equivalent to that from its full resolution uncompressed counterpart. In contrast, the SDCFs provide object recognition responses at lower resolutions of the data, which while accurate, leave more ambiguity in object location.

## 1.4 Thesis Outline and Contributions

This thesis introduces two novel systems for performing object recognition in compressed imagery. Chapters 3 and 4 discuss the PRSC model, while Chapter 5 presents the SDCF model. The following is a chapter by chapter review of this thesis indicating contributions made by the author.

**Chapter 2 Background**

We give a brief review of the basic subband coding and correlation filter models. Essential notation and nomenclature unique to each area is provided. In addition, we establish the baseline results for comparison with experiments in later chapters.

**Chapter 3 A New Pattern Recognition Subband Coder Architecture**

This chapter reveals a method for combining the correlation and subband filters into a new compression/recognition architecture. The material addresses the computation leg of our three prong approach. We present a full analysis of the computational efficiency of the new architecture.

Novel contributions from this chapter include:

- A new signal processing architecture which *integrates* correlation and subband filters into a *homogeneous* operation, and facilitates effective object recognition directly in the compressed domain. It possesses the following characteristics.
  - It is a *universal architecture* for operating any linear filter directly on the multiresolution data of a subband coder.
  - It produces a response *identical* to that produced by the *reconstructed* image.
  - It results in a response *identical* to that effected by the *original* image, within the fidelity of the subband transform.

- It is *more computationally efficient* than the separate implementation of the two filters.
- It provides a *faster response* than that resulting from the *original* image, by a *factor of F' = 1.6*, in the limit, as image size goes to infinity.

- A matrix vector formulation for operating linear systems directly in a transform domain. It has the following attributes:
  - It is *universal* for a linear operator and unitary transform.
  - It results in a response *identical* to that produced by the *untransformed* input.

**Chapter 4 Joint Optimization of the Compression and Recognition Filters**

We address the remaining two legs of our three prong approach: optimizing the joint performance of the compression and recognition tasks within the PRSC. First we review the standard design of a quadrature mirror filter (QMF) for a subband coder. Then we introduce the new terms which incorporate recognition performance into the design of the QMF, as well as describe the technique for including the QMF response in the design of the correlation filter. In addition, we present rate error curves which can be used to tune the system between varying rates of compression and recognition performance. Finally, we give suggestions for future work.

Novel contributions from this chapter include:

- A method of *jointly optimizing* the *compression capability* of a QMF subband filter and the *recognition performance* of a correlation filter. This method provides:
  - *superior* performance than operating the two filters *independently*
  - *superior* performance to operating on the *uncompressed* data

- A suggestion for *jointly optimizing* the compression and recognition performance to *compensate for quantization*.

**Chapter 5 Subband Domain Correlation Filters**

We introduce our second solution to the problem of object recognition in compressed imagery in this chapter. The approach synthesizes correlation filters that operate directly on subband coefficients at multiple resolution levels. We present a novel training technique as well as an additional QMF design method, both of which enhance system performance. Again, we provide rate error curves which can be used to trade off the compression and recognition performance. Lastly, we make suggestions for future work.

Novel contributions from this chapter include:

- A new pattern recognition algorithm which utilizes correlation filters synthesized in the *subband coefficient domain*, and allows for object recognition at the *multiple resolutions* of a subband coder.
- A new *training* methodology to *accommodate the downsampling* of the subband coefficients.
- A novel method of *jointly optimizing* the compression capability of a QMF subband filter and the recognition performance of a correlation filter to *compensate for downsampling* in the subband coefficients.

**Chapter 6 Concluding Remarks**

In this final chapter, we summarize and compare the two models introduced in this thesis as well as recount the contributions made.

# Chapter 2

# Introduction to Correlation Filters and Subband Coders

As discussed in the previous chapter, the system presented in this thesis uses subband coding as its image compression component, and correlation filtering for object recognition. The current chapter provides background material on these two fields as well as describes important procedures of our experiments and additionally, recent results of related prior work.

An outline of the chapter is as follows. Sections 2.1 and 2.2 define the notation and terminology we use throughout this thesis. We provide a basic review of correlation filters and subband coders along with our specific implementations in Sections 2.3 and 2.4, respectively. Section 2.5 then lays the groundwork for the experiments performed in later chapters. We include all of the background material regarding the data, methodology, and performance measures for the experiments. Finally, in Section 2.6 we present current state-of-the-art results on both compressed and uncompressed data as a baseline for comparison. These results are useful for comparison with the performance achieved in this thesis and reported in chapters 4 and 5.

## 2.1 Notation

Images in the space domain are denoted in lower case italics while upper case italics represent their counterpart in the frequency domain. Thus, a two-dimensional image $x(m,n)$ has Fourier transform $X(k,l)$. Vectors are expressed by lower case bold characters while matrices are denoted by upper case bold characters. Either $x(m,n)$ or $X(k,l)$ can be expressed as a column vector $\mathbf{x}$ by lexicographical scanning. We define lexicographical scanning to proceed columnwise, unless otherwise stated. The quantities $\mathbf{X}^T$, $\mathbf{X}^*$, $\mathbf{X}^t$ and $\mathbf{X}^{-1}$ signify, respectively, the transpose, conjugate, transpose conjugate, and inverse of $\mathbf{X}$. Correlation and convolution are designated by the symbols $\otimes$ and $*$, respectively. The quantity $\hat{x}(n)$ refers to a reconstructed version of the original signal $x(n)$.

## 2.2 Terminology: A Quick Reference Guide

To bridge the two disparate fields brought together in this work, we provide some nuances of the terminology used throughout the thesis.

- The term *correlation filter* is synonymous with *object recognition filter* and *MACH filter*.
- When we *reconstruct* the image with a subband coder, we use *synthesis* filters, therefore the terms *reconstructed signal* and *synthesized signal* represent the same thing. Yet when we *create* the MACH filter, we *synthesize* it from its training input.
- A *subband coder* is really a *compression system*. The term *coding* is sometimes used to mean *compressing*; sometimes, however, we *encode symbols* with their representative *codewords*.

- A *multiresolution* decomposition, *pyramid* decomposition, *subband* decomposition, and *hierarchical* decomposition are all the same entity.
- *Analysis* filters form the *forward* compression transform, while the *inverse* compression transform uses *synthesis* filters (a.k.a. *interpolation* filters).
- The subband literature denotes the *first level* of decomposition as the *highest level*, because it contains the *highest resolution* data. The *last level* of decomposition is known as the *lowest level*.
- The *high frequency subbands* in a subband decomposition are known collectively as the *upper subbands*.
- Throughout the thesis, we often refer to a *baseline compression recognition system*. In this case, the system fully reconstructs the image prior to performing object recognition.

## 2.3 Review of Correlation Filter Theory for Object Recognition

It is well known a matched spatial filter is optimum for the detection of a single image in the presence of additive white noise. Correlation filters are specially formulated generalizations of the matched spatial filter which are able to recognize many different views of an object in the presence of signature variations and in clutter. In general, correlation filters are composites of several training images representative of a particular object. Numerous references [VKu86, Mah87, MVK+87, Vku92, VKM+92, MVK+94, MVK96, MVK97b] address the details of correlation filter theory at length, with the most current and complete examination in [VKMJ00], however, we give a brief review of the algorithm here.

In the simplest case, the output of a correlation filter $c(m,n)$ in response to an input image $x(m,n)$ is described by

$$y(m,n) = c(m,n) \otimes x(m,n) \qquad (2.1)$$

where $y(m,n)$ is referred to as the correlation surface. Typically, however, the processing is done in the frequency domain for the purpose of computational efficiency, as shown in Figure 2.1. In this case, the correlation output is given as

$$y(m,n) = \Im^{-1}\left\{C^*(k,l) \cdot X(k,l)\right\} \qquad (2.2)$$

where $C(k,l) = \Im\{c(m,n)\}$, $X(k,l) = \Im\{x(m,n)\}$. $\Im\{\}$ and $\Im^{-1}\{\}$ represent the forward and inverse discrete Fourier Transform operation, respectively.

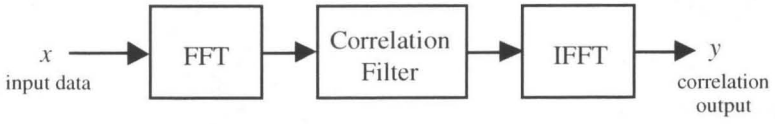



Figure 2.1 **Correlation System for Processing Uncompressed Imagery**

As stated in [VKMJ00], correlation filter theory deals with the design of $C(k,l)$ such that the filter output can achieve the following three objectives:

- recognition of distorted versions of the reference pattern
- robust behavior in the presence of clutter and noise
- high probability of correct recognition while maintaining a low error rate

Many researchers have investigated numerous design techniques for building correlation filters to meet the three objectives listed above. Our goal in this thesis is not to develop better underlying design techniques for correlation filters, but rather to effectively use them with compressed data. To this end, we employ one of the current

state of the art correlation filters, the Maximum Average Correlation Height (MACH) filter [MVK97b].

We now provide a brief description of the formulation of the MACH filter. Consider a set of $N$ training images whose discrete Fourier Transforms are given by $X_i(k,l)$, $1 \le i \le N$. The average training image is then given by

$$M(k,l) = \frac{1}{N}\sum_{i=1}^{N} X_i(k,l) \qquad (2.3)$$

We also define the power spectrum of the training set as

$$D(k,l) = \frac{1}{N}\sum_{i=1}^{N} \left|X_i(k,l)\right|^2 \qquad (2.4)$$

and a spectral variance term as

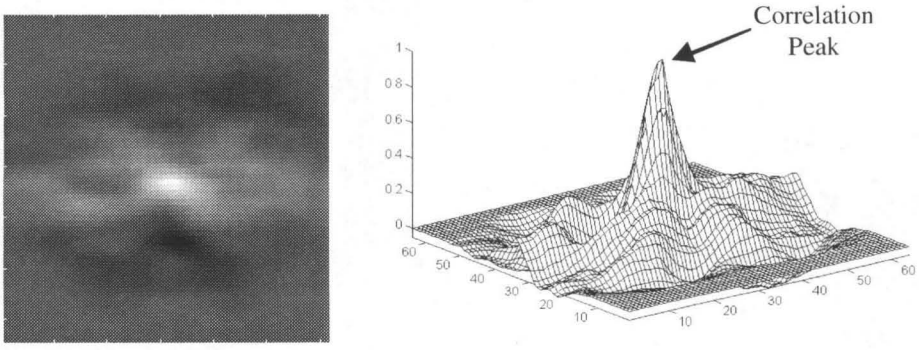
$$S(k,l) = D(k,l) - \left|M(k,l)\right|^2 \qquad (2.5)$$

The MACH filter is then defined in the frequency domain as

$$C(k,l) = \frac{M(k,l)}{\alpha S(k,l) + \beta D(k,l) + \gamma \mu} \qquad (2.6)$$

where the parameter, $\mu$, is simply the mean value of spectral variance, and the constants $\alpha$, $\beta$, and $\gamma$ are used for optimally trading between distortion tolerance, noise tolerance and correlation energy minimization. The interested reader is referred to [Car96] for further details about the use of these parameters in design of the MACH filter.

Figure 2.2 displays a typical correlation output of the MACH filter. Generation of the correlation output is only the first step; a subsequent postprocessing algorithm then interprets the peaks of the correlation surface. The height of the peak corresponds with the likelihood of a pattern match, while the peak location corresponds with the location of the pattern match. Numerous researchers have also developed many techniques to




(a) Intensity Plot (b) Surface Plot

Figure 2.2 **Plots of a Typical Correlation Output**
To obtain a robust correlation plane, maximize the peak, reduce the variations due signal distortions, and reduce any sidelobes produced by noise and clutter.

interpret the correlation output and its relationship to recognition of the object in question [SS88, WG90, MVKS96]. We selected the simplest technique for this thesis investigation.

For each correlation surface generated, we compute the peak-to-sidelobe ratio (PSR)

$$PSR = \frac{y(m,n) - \mu_y}{\sigma_y} \qquad (2.7)$$

at the location of the maximum correlation response (also know as the *correlation peak*). In Equation 2.7, $y(m,n)$ is the value of a particular location of the correlation surface; $\mu_y$ and $\sigma_y$ are the mean value and standard deviation of the correlation surface over a bounded region in its neighborhood. In fact, it can be shown that the MACH filter is an optimum detector in the sense that it maximizes PSR for the target class. The architecture of the classifier based on the MACH filter is simple. Given a test image at the input, the system assigns it the label of the class corresponding to the filter which yields the highest

$$class = \arg\left\{\max\left(PSR_i\right)\right\}, \quad i \in \left\{1,...N_c\right\} \qquad (2.8)$$

where $N_c$ is the number of object classes under consideration.

## 2.4 Subband Coding for Compression

The objective of image compression is to reduce the number of bits of information necessary to represent a given image by eliminating redundancy in the image or by introducing distortion into the image in a manner that is acceptable to the viewer, or in our case, to the object recognizer. Sayood [Say96] relates a good summary of image compression techniques. Figure 2.3 depicts a standard compression system.

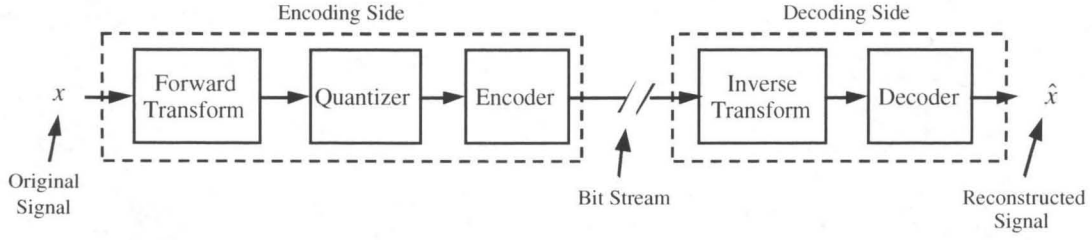



Figure 2.3 **Standard Compression System**

Within a compression system, a subband coder performs the forward and inverse transform operations, with analysis and synthesis filters, respectively. A two band, one-dimensional, subband coder is shown in Figure 2.4. The analysis filters, $H_0(z)$ and $H_1(z)$, decompose an input signal, $x(n)$, into lowpass and highpass filtered bands of

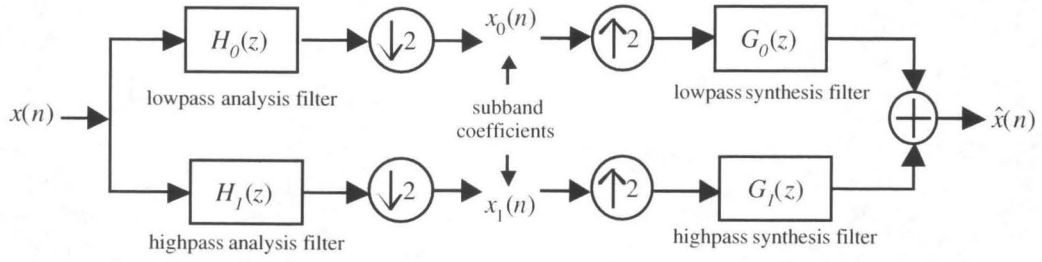



Figure 2.4 **Diagram of a One-Dimensional, Two Band Subband Coder**

information, which are then downsampled. Resulting subband coefficients are then coded and transmitted. The system accounts for the loss of information from downsampling by cleverly designed synthesis filters, $G_0(z)$ and $G_1(z)$, applied at the decoder to reconstruct the original signal. Several references provide detailed introductions to subband coding and its variants: Mallat [Mal89], Vaidyanathan [Vai93], Vetterli and Kovacevic [VK95], and Strang and Nguyen [SN96].

The overall transfer function of an analysis and synthesis system for a one-dimensional, two band subband coder like in Figure 2.4 follows the equation below.

$$\begin{aligned}\hat{X}(z) = &\frac{1}{2} X(z)\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right] \\ &+\frac{1}{2} X(-z)\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right]\end{aligned} \tag{2.9}$$

where $\hat{X}(z)$ is the reconstructed signal and $X(z)$ represents the z transform of the time domain signal $x(n)$. $H_0(z)$, $H_1(z)$, $G_0(z)$, and $G_1(z)$, are similarly defined. Signal reconstruction consists of the frequency response (first term) and the aliasing component (second term).

Subband coding is easily extended to two dimensions by analyzing the input signal independently in the horizontal and vertical directions. The analysis transform can be applied recursively to the low frequency subband, resulting in a multiresolution, or pyramid decomposition, as displayed in Figures 2.5 and 2.6.

A typical level of a multiresolution decomposition will have four subbands of information, one composed of only low frequency components, the LL subband, and three containing high frequency components, the HL, LH, and HH subbands. Often, subband literature refers to the high frequency subbands collectively as the upper subbands.

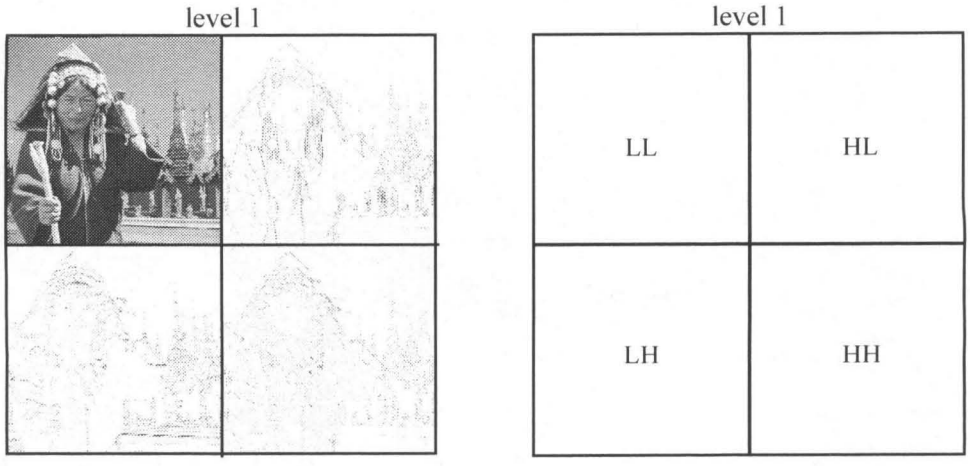



Figure 2.5 **A One Level Subband Decomposition of an Image**
The image of (a) exhibits a one level subband decomposition, while (b) displays the corresponding names commonly associated with each subband.

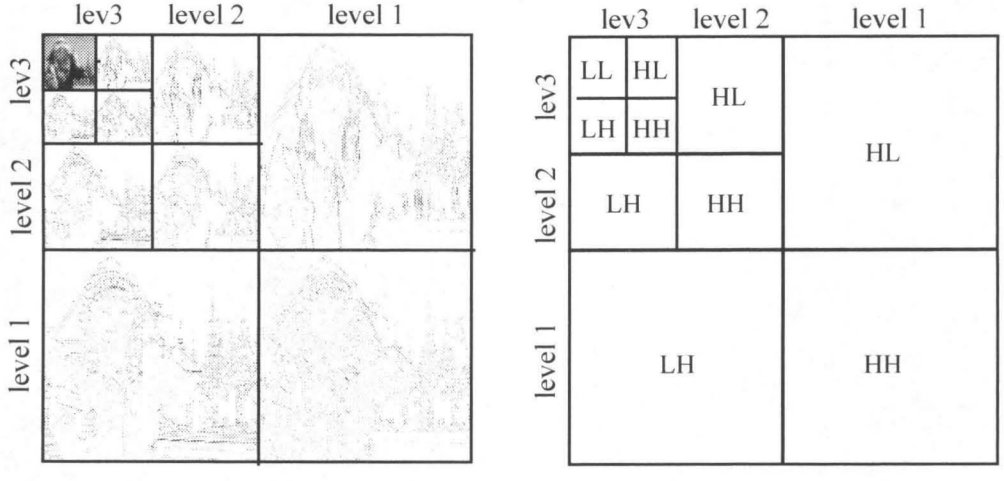



Figure 2.6 **A Three Level Subband Decomposition of an Image**
The image of (a) exhibits a one level subband decomposition, while (b) displays the corresponding names commonly associated with each subband. The subband literature denotes the first level of decomposition as the highest level, because it contains the highest resolution data. In this figure, level three is the lowest decomposition level.

In the absence of quantization and ignoring machine round-off error, subband coders can provide lossless compression. The compact distribution of the subband coefficients accounts for the reduction in data rate during lossless compression. Subband coders are

most often used in lossy compression, however, in which case the compression rate relates directly to the degree of quantization.

Additionally, with a properly designed subband filter, subband coders can provide perfect reconstruction, in the absence of quantization. Compression applications typically use perfect reconstruction filters [Vai87a, Vai87b, Vet87, Vai90] in the case of both lossless and lossy compression. A popular set of perfect reconstruction filters are the QMFs, [CEG76, Vai87b, Vai90] which we use in this thesis research. Section 4.2 discusses the details of our QMF design at great length. Again, our purpose is not to improve the compression power and quality of QMFs, but to optimize their effectiveness with an object recognition model.

## 2.5 How to Perform an Object Recognition Experiment with Correlation Filters

Four issues are key to the quality of any object recognition experiment: 1) the image data, 2) the preprocessing of the image data, 3) the methodology of the experiments, and 4) the performance measures. We address all of these issues here.

### 2.5.1 Image Data

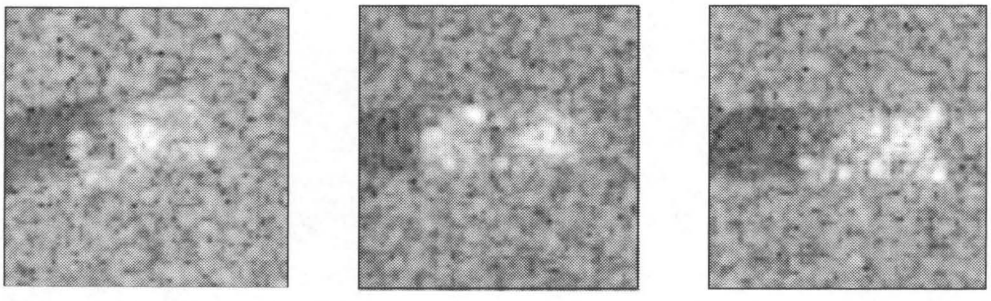
Essential to the validity of any experiment are the quality and characteristics of the data. With this in mind, we selected the MSTAR Synthetic Aperture Radar (SAR) image database publicly released by Defense Advanced Research Projects Agency (DARPA) which is easily available and widely used by many researchers. Another reason for selecting the one foot resolution MSTAR data is that its ground truth is well documented, an important feature for synthesizing the correlation filters. The main issue

involving object recognition of SAR data is that changes in the object orientation and the grazing angle of the radar sensor create strong variations in the resulting image.

We construct a three class problem to distinguish three military vehicles: a BMP Bradley fighting vehicle, a BTR armored personnel carrier, and a T72 tank. Figure 2.7 shows photographs of each vehicle, while Figure 2.8 displays the vehicles as depicted in the SAR imagery.

(a) BMP (b) BTR (c) T72

Figure 2.7 **Photos of Object Classes**



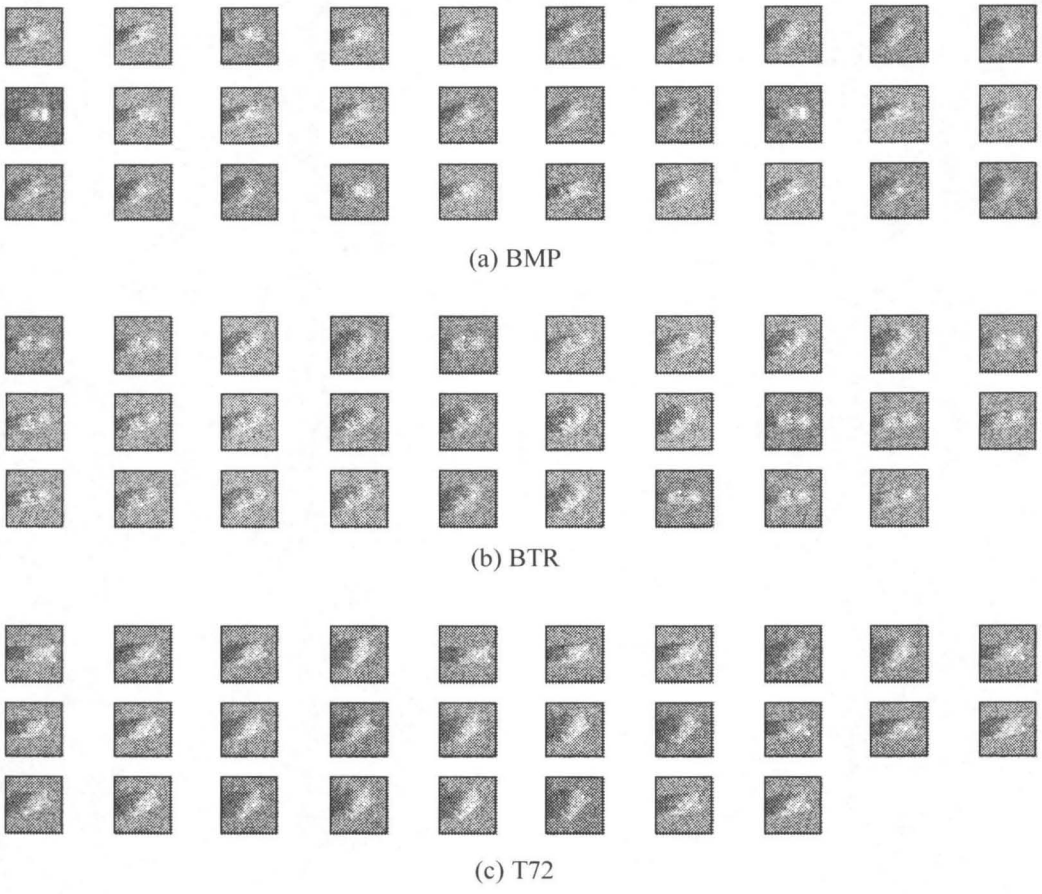
(a) BMP (b) BTR (c) T72

Figure 2.8 **SAR Images of Object Classes**

The photographs may not be displaying the exact models used in the SAR imagery. All of the vehicles lie between 0 and 45 degrees viewing angle to the sensor, with 0 degrees corresponding to a head-on view. All of the images are cropped to size 64 x 64 pixels for the experiments in this thesis. It was verified that the vehicle's dimension fit within this image size.

**Training Set**

Correlation filter theory requires a set of training images to synthesize a set of correlation filters of the objects to be recognized. To this end, we use the MSTAR data generated from radar at 17° elevation for the training set, which Figure 2.9 shows in its entirety.



(a) BMP

(b) BTR

(c) T72

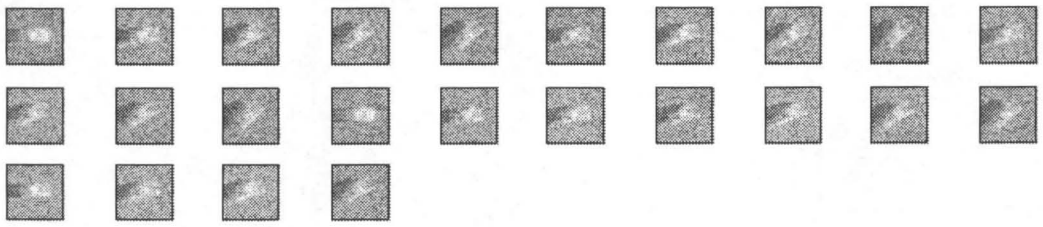
Figure 2.9 **Full Training Set of Object Classes**
The training set consists of 0° - 45° views of the objects taken at a 17° elevation.

Because the ground truth is given, the vehicles are centered within the input image prior to training. Object registration is an important step in synthesizing the correlation filters for object recognition. The training set consists of 30 BMP images, 29 BTR images, and 28 T72 images.

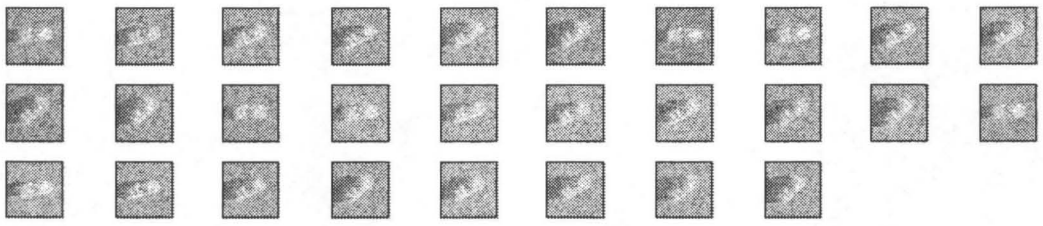
We perform training for the PRSC with unquantized reconstructed image data. That is, we process the images with the subband filters, but do not quantize the resulting coefficients. This is to incorporate any artifacts of the compression process in the correlation filter. Training data for the SDCF system comes from subbands themselves. The subbands are formed from the original images and again no quantization is applied to the subband coefficients.

**Testing Set**

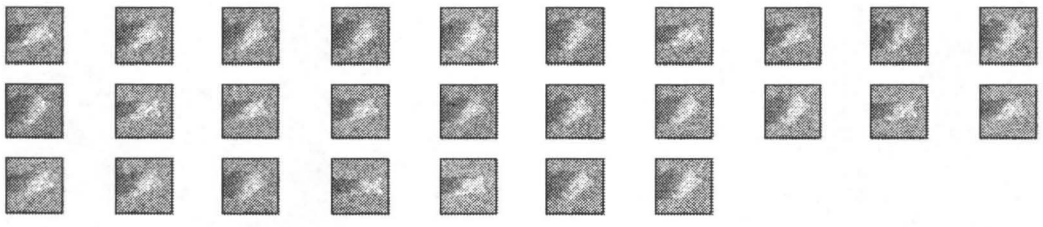
The test set of our experiments, shown in Figure 2.10, uses the MSTAR data generated from radar at 15° elevation. We do not perform any registration on the test



(a) BMP



(b) BTR



(c) T72

Figure 2.10 **Full Test Set of Object Classes**
The test set consists of 0° - 45° views of the objects taken at a 15° elevation.

data. Images of 24 BMP examples, 28 BTR examples, and 27 T72 examples comprise the test set.

### 2.5.2 Preprocessing

Correlation filter performance in SAR data improves by applying simple thresholds to remove some of the background pixels. To our benefit, prior work [VKMJ00] developed a excellent thresholding method for this particular set of SAR data which we used in all of our experiments and will review here. Prior to any thresholding, however, we take the log magnitude of the SAR data and then convert it to byte. In the case of the training images, we retain the integrity of the object by creating a mask from the ground truth data. Thresholding the test images, however, is strictly a mathematical process, with no knowledge of the object's presence or location in the input image.

The preprocessing involves two different thresholds and is a two step process. In the case of the first threshold, its value replaces all image pixels with a value less than the computed threshold.

$$x(m,n) = \max\{x(m,n),\ t_1\},\ \text{with } t_1 = \mu_x - 2.2\sigma_x, \qquad (2.10)$$

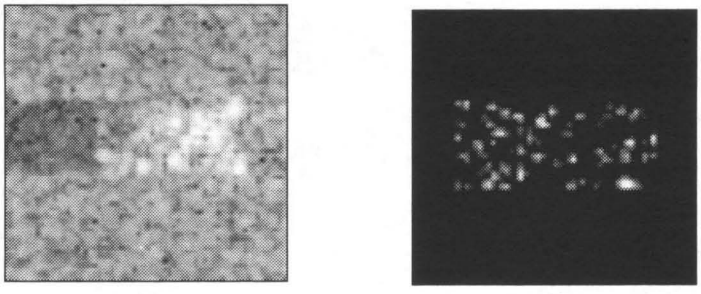
where $x(m,n)$ is a particular pixel value of the input image and $\mu_x$ and $\sigma_x$ are the mean and standard deviation, respectively, of the input image. The second threshold is computed from these new image values and the image is thresholded again according to the following rule.

$$x(m,n) = x(m,n) - t_2,\ \text{if } x(m,n) > t_2 \qquad (2.11)$$

$$\text{otherwise, } x(i,j) = 0$$

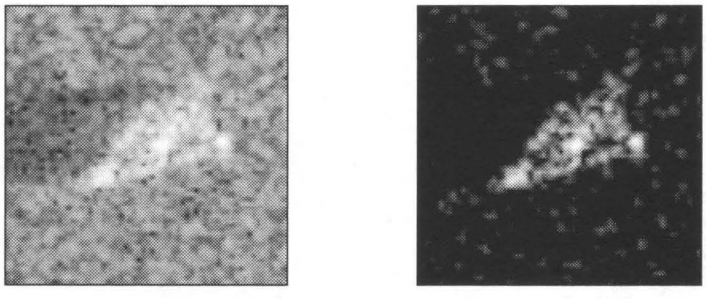
$$\text{where } t_2 = \mu_x - 0.5\sigma_x \qquad (2.12)$$

Notice that both $t_1$ and $t_2$ are image dependent, and that $\mu_x$ and $\sigma_x$ are the new mean and standard deviation values. In Figures 2.11 and 2.12 we show both the original and preprocessed images of a BMP from the train and test set, respectively.



(a) Original Image (b) Preprocessed Image

Figure 2.11 **Preprocessing for the Training Images**
Image (a) displays an original training image of a T72, (b) exhibits its preprocessed counterpart. Because we know the true location and orientation of the training data, the preprocessing can remove much of the extraneous clutter.



(a) Original Image (b) Preprocessed Image

Figure 2.12 **Preprocessing for the Test Images**
We show an original T72 test image in (a) while (b) portrays its preprocessed version. Notice that the preprocessing does not remove all of the clutter, but does remove part of the target, particularly if the target lies in a shadow.

It is important to note that we assume the preprocessing is done on the original image data, prior to the subband decomposition; that is, it is stored as a preprocessed image in the compressed domain. There are two motivations for this approach. First, it is reasonable to assume that many applications can and will preprocess the image prior to

compression and storage. Second, the same thresholding techniques used on the original image data apply to the subband coefficient data. We consider this task as secondary to the main course of this research.

### 2.5.3 Methodology

An object recognition experiment with correlation filters consists of two phases. In the training phase, a MACH recognition filter is built for each object class. Thus, three filters were built in our case; one each for the BMP, BTR and T72.

During the test phase, the system processes the unknown object with each of the stored MACH filters and computes a PSR value for each possible class according to Equation 2.10. If there are $N$ object classes, then there are $N$ PSR values. As discussed in Section 2.3, the object is classified according to Equation 2.8. Sometimes, however, the image will not contain any of the known object classes and the system must account for this. Therefore, we construct a PSR threshold that the output must exceed to register a classification. If none of the PSR values are greater than the threshold, then the system classifies the image as a 'rejection' which means that the image does not contain any object that belongs to one of the known object classes. The rejection category is an important tool in high quality object recognition.

### 2.5.4 Performance Measures

Finally, we assess the power and capability of our model for object recognition in compressed imagery with standard performance measures from each of the three foci of our thesis research: 1) object recognition, 2) compression, and 3) computational

efficiency. Note that the visual quality of the reconstructed image is secondary to other factors in this thesis investigation.

**Recognition Metrics**

To gauge the capability of the object recognizer, we compute the probability of correct classification, error, and rejection, referred to as Pc, Pe, and Pr, respectively, and always stated in percentages. A further detail of the performance is known as the confusion matrix, an example of which is shown in Table 2.1. The confusion matrix

Table 2.1 **Hypothetical Confusion Matrix**

| | | Computed Object Class | | | |
|---|---|---|---|---|---|
| | | **BMP** | **BTR** | **T72** | **Reject** |
| True | **BMP** | 21 | 2 | 1 | 0 |
| Object | **BTR** | 0 | 26 | 0 | 2 |
| Class | **T72** | 2 | 0 | 24 | 1 |

allows us to determine which objects are being confused with others and is sometimes helpful in performance assessment. In the case of Table 2.1, we see that the system misjudged three BMPs; twice it misclassified a BMP as a BTR, and once as a T72. The object recognizer did not misclassify any BTR images, however, it rejected two of them as not having any known object within the image. Furthermore, the Pc is 89.9%, the Pe is 6.3%, and the Pr is 3.8% in this example.

**Compression Metrics**

The rate of data transmission is measured in bits per pixel (bpp).

$$\text{rate} = \frac{B}{MN} \qquad (2.13)$$

where $B$ is the total number of bits required to transmit the data for the reproduction image, and $M$ and $N$ are the row and column dimensions of the reconstructed image.

Sometimes rate is reported as a compression ratio. e.g., A bit rate of two, would elicit a compression ratio of 4:1 for eight bit original data.

As stated previously, we are less interested in the reconstruction fidelity than we are in the compression aspects of a subband transform. For this reason, we evaluate the compression power solely on the basis of the bit rate. We refer to two typical compression metrics, the PSNR and MSE, as reconstruction metrics throughout the thesis.

**Reconstruction Metrics**

To quantify the quality of the compression module we use the typical measures of mean squared error (MSE) per pixel and Peak Signal to Noise Ratio (PSNR) between the original and reconstructed images defined below.

$$\mathrm{MSE} = \frac{1}{MN}\sum_{m=1}^{I}\sum_{n=1}^{J}\left(x(m,n) - \hat{x}(m,n)\right)^2 \qquad (2.14)$$

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right) \qquad (2.15)$$

where $M$ and $N$ are the row and column dimensions of the original image. The MSE has no units, whereas the PSNR is measured in decibels (db). In this thesis, the MSE is always reported for a three level subband decomposition, unless otherwise stated. Both of these values measure how accurately the reproduced image matches the original image, however, they are only meaningful when the context of a specific bit rate.

**Computation Metrics**

Lastly, to assess the computational efficiency of our model we calculate the order of the computation by totaling the number of multiply operations necessary for the system. to process one image.

## 2.6 State of the Art Object Recognition Results using Correlation Filters

In this section, we provide a performance baseline with which to compare the results of this thesis investigation. We focus on the object recognition performance and the computational order for comparison with our new system. With this baseline, our thesis can then address the following questions:

- *"How will compression degrade the recognition performance?"*
- *"What is the computational cost of the new system?"*

### 2.6.1 State of the Art Results on Uncompressed Imagery

Using the system diagrammed in Figure 2.1 and the MACH filter and postprocessing technique discussed in Section 2.3, we performed the three class experiment defined in Section 2.5.1. The experiment utilized the original uncompressed SAR test data, and resulted in the confusion matrix exhibited in Table 2.2.

Table 2.2 **Confusion Matrix Resulting from Uncompressed Imagery**

| | | Computed Object Class | | | |
|---|---|---|---|---|---|
| | | **BMP** | **BTR** | **T72** | **Reject** |
| True Object Class | **BMP** | 23 | 0 | 1 | 0 |
| | **BTR** | 1 | 26 | 1 | 0 |
| | **T72** | 0 | 0 | 27 | 0 |

Table 2.3 identifies the performance percentages. They are commensurate with results published in the open literature [VKMJ00] for much larger experiments on similar data.

Table 2.3 **Recognition Results for Uncompressed Imagery**

| | **Pc** | **Pe** | **Pr** |
|---|---|---|---|
| **Baseline Experiment** | 96.2% | 3.8% | 0% |

### 2.6.2 State of the Art Results on Compressed Imagery

The current baseline method of working with compressed imagery is to fully reconstruct the image and then process it through an object recognition model. Figure 2.13 illustrates such a system.

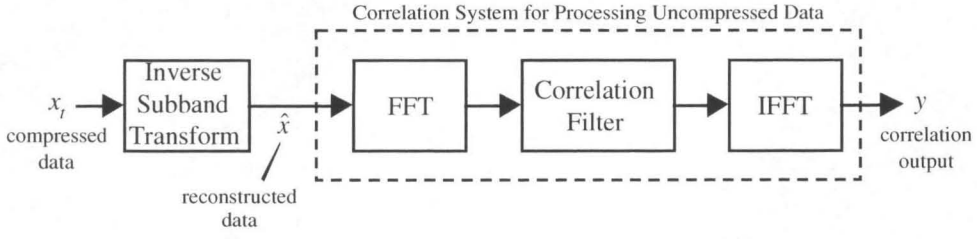



Figure 2.13 **Baseline Compression Recognition System**
Notice that the above system is simply an augmentation of the correlation system used with uncompressed imagery (shown in Figure 2.1).

Thus, for lossless compression, the recognition performance for the baseline compression recognition system is exactly the same as that resulting from uncompressed data. In the case of lossy compression with SAR imagery, previous work [WM99] shows that the recognition performance is very robust. Using the Embedded Zerotree Wavelet algorithm [Sha93], Walls and Mahalanobis report the following correct classification results on a ten class problem from the MSTAR data.

Table 2.4 **Recognition Results of the Baseline Compression Recognition System**

| | Compression Ratio | | | | | | |
|---|---|---|---|---|---|---|---|
| | **10:1** | **20:1** | **100:1** | **186:1** | **350:1** | **675:1** | **1000:1** |
| Pc | 97% | 95% | 94% | 91% | 79% | 35% | 14% |

# Chapter 3

# A New Pattern Recognition Subband Coder Architecture

Motivating the design of this new architecture, which addresses the implementation branch of our trifurcate approach, was the drive to integrate the correlation and subband filters into a single synergistic component. The key issues of the design parallel two questions extended at the outset of this thesis.

- "*Can we perform correlation in compressed image data and achieve an output identical to that resulting from the original image data?*"
- "*Is such a process computationally efficient?*"

These two objectives lead to the question, "*How can one synthesize a correlation MACH filter from original image data, and then pull it back into the individual subbands?*" Such a procedure forms the basis for *image domain processing* and is the heart of our Pattern Recognition Subband Coder (PRSC) architecture. With the PRSC architecture we provide computationally efficient correlation filtering in the compressed domain (i.e., image reconstruction is not performed) while we preserve the subband QMF

properties and circumvent entanglements with the downsampling effects encountered in subband coder architectures.

Section 3.1 presents a mathematical formulation of a linear system that results in an identical response for both compressed and uncompressed data. In Section 3.2 we elucidate the methodology for merging the subband and correlation filters. It is here that we define the PRSC architecture and explain its construction. We discuss the computational efficiency of the PRSC in Section 3.3, as well as its parallel structure, which improves the processing time over that of uncompressed images. Finally, we summarize the effectiveness and limitations of the PRSC in Section 3.4.

## 3.1 One-Dimensional Formulation

Consider the linear system

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{3.1}$$

Equation 3.1 represents a one-dimensional correlation process written in matrix vector notation, where $\mathbf{x}$ is a one-dimensional input signal, $\mathbf{C}$ is a Toeplitz matrix defined by a one-dimensional correlation filter, and $\mathbf{y}$ is the one-dimensional correlation output. Matrix vector notation provides a convenient mathematical and analytical tool for the ensuing discussion.

Our goal is to express the response $\mathbf{y}$ in terms of the transformed input, $\mathbf{x}_t$. That is, with a linear system, to achieve an output from a transformed input that it is identical to the one produced from its original counterpart. The following equations hold for any linear system and any general transform that can each be written as a matrix operation. They are not limited to the correlation process and compression transform.

### 3.1.1 A General Transform Representation

Given the matrix construction of a unitary transform operator, **T**, note that Equation 3.1 can be rewritten as

$$\mathbf{y} = \mathbf{C}\mathbf{T}^{t}\mathbf{T}\mathbf{x} \qquad (3.2)$$

We define $\mathbf{x}_t$ as the transformed input,

$$\mathbf{x}_t = \mathbf{T}\mathbf{x} \qquad (3.3)$$

$\mathbf{C}_t$ as the 'inverse transformed' correlation matrix,

$$\mathbf{C}_t = \mathbf{C}\mathbf{T}^{t} \qquad (3.4)$$

and **T** is the transformation matrix. With these definitions, Equation 3.2 can be restated.

$$\mathbf{y} = \mathbf{C}_t\mathbf{x}_t \qquad (3.5)$$

If **T** is not unitary, the above set of equations remain intact with $\mathbf{T}^t$ being replaced by $\mathbf{T}^{-1}$, as long as **T** is nonsingular.

Immediately, we see that if **T** is a unitary compression transform, then Equation 3.5 provides an elegant method of achieving the sought after correlation surface, **y**, directly from the compressed image without any reconstruction.

**Implementation**

The implementation for a compression and correlation system is simple and straightforward. We assume:

1) we are given the compressed images, $\mathbf{x}_t$,
2) we know the compression transform matrix, **T**, and
3) we can compute the transformed correlation matrix, $\mathbf{C}_t$, a priori, and store it off-line.

Now to formulate $\mathbf{C}_t$ we must first produce the standard correlation filter in the uncompressed domain (and its matrix counterpart **C**), which does require uncompressed

images. A typical recognition system completes the training phase a priori, however, when time and processing power are not as limited as during system operation.

If the system specifically calls for a subband transform, the transformation matrix, **T**, is created directly form the subband analysis filter pair. We discuss the construction of **T** in Section 4.1. In brief, the transformation matrix, **T**, combines the filtering and downsampling of a subband analysis transform over multiple levels in a matrix structure. Our informal treatment of the **T** structure in the following discussion is a preface to the detailed material of Chapter 4.

**Parallelization**

For recursive unitary transforms (like the subband transform), we define a unique mathematical formulation of Equation 3.5 that lends itself to a parallel implementation. In the case of a recursive subband transform, the first level analysis and synthesis is given by

$$\mathbf{x}_1 = \mathbf{T}\mathbf{x} \quad \text{and} \quad \mathbf{x} = \mathbf{T}^t\mathbf{x}_1 \tag{3.6}$$

The transformed signal can be separated into its low and high frequency components: $\mathbf{x}_1 = [\mathbf{x}_{1L}\ \mathbf{x}_{1H}]$. Or, with proper zero padding of the individual vector components,

$$\mathbf{x}_1 = \mathbf{x}_{1L} + \mathbf{x}_{1H} \tag{3.7}$$

Now, adapting **T** for decreasing vector size, and continuing to split the low frequency band further, the lower levels become

$$\mathbf{x}_2 = \mathbf{T}\mathbf{x}_{1L} \quad \text{and} \quad \mathbf{x}_3 = \mathbf{T}\mathbf{x}_{2L} \tag{3.8}$$

Analogous to Equation 3.7, and with the appropriate zero padding, the lower levels can also be written as a sum of their low and high frequency components.

$$\mathbf{x}_2 = \mathbf{x}_{2L} + \mathbf{x}_{2H} \quad \text{and} \quad \mathbf{x}_3 = \mathbf{x}_{3L} + \mathbf{x}_{3H} \tag{3.9}$$

Because the **T** matrix must decrease in size at every level, however, we append a subscript, $k$, denoting the matrix's accommodation of the size of the signal at the $k^{th}$ level. Thus, the appropriate analysis and synthesis equations become

$$\mathbf{x}_1 = \mathbf{T}_1\mathbf{x} \quad \mathbf{x}_2 = \mathbf{T}_2\mathbf{x}_{1L} \quad \mathbf{x}_3 = \mathbf{T}_3\mathbf{x}_{2L} \tag{3.10}$$

$$\mathbf{x} = \mathbf{T}_1^t\mathbf{x}_1 \quad \mathbf{x}_{1L} = \mathbf{T}_2^t\mathbf{x}_2 \quad \mathbf{x}_{2L} = \mathbf{T}_3^t\mathbf{x}_3 \tag{3.11}$$

(Note that in the more formal treatment of **T** in Section 4.1, the matrix will no longer require subscripts, as we will more formally define **T** over multiple levels of recursion.)

Finally, by writing each level as a sum of its low and high frequency components, and using the appropriate zero padding on the signals at levels one and two, the synthesis of the original signal, **x**, is as follows.

$$\begin{aligned}\mathbf{x} &= \mathbf{T}_1^t\left[\mathbf{T}_2^t\left[\mathbf{T}_3^t\mathbf{x}_3 + \mathbf{x}_{2H}\right] + \mathbf{x}_{1H}\right] \\ &= \mathbf{T}_1^t\mathbf{T}_2^t\mathbf{T}_3^t\mathbf{x}_3 + \mathbf{T}_1^t\mathbf{T}_2^t\mathbf{x}_{2H} + \mathbf{T}_1^t\mathbf{x}_{1H}\end{aligned} \tag{3.12}$$

Now, by again treating the correlation filter as a linear system, Equation 3.5 can be computed in parallel, i.e.,

$$\mathbf{y} = \mathbf{C}\mathbf{x} = \mathbf{C}\mathbf{T}_1^t\mathbf{T}_2^t\mathbf{T}_3^t\mathbf{x}_3 + \mathbf{C}\mathbf{T}_1^t\mathbf{T}_2^t\mathbf{x}_{2H} + \mathbf{C}\mathbf{T}_1^t\mathbf{x}_{1H} \tag{3.13}$$

After combining all the matrix terms, the above equation becomes

$$\mathbf{y} = \mathbf{C}_{t3}\mathbf{x}_3 + \mathbf{C}_{t2}\mathbf{x}_{2H} + \mathbf{C}_{t1}\mathbf{x}_{1H} \tag{3.14}$$

where $\mathbf{C}_{tk}$ is the appropriate combination of correlation and transform matrices for level $k$, and the signals at levels one and two still carry the appropriate zero padding. Equation 3.14 can also be viewed as

$$\mathbf{y} = \mathbf{y}_3 + \mathbf{y}_2 + \mathbf{y}_1 \tag{3.15}$$

where $\mathbf{y}_k$ is the correlation output surface for level $k$.

By nesting the recursion (Equation 3.12), we manifest a parallel implementation as clearly elucidated in Equation 3.14. This equation can be helpful in the implementation of one-dimensional linear systems in the transformed domain. The formal treatment here is one possible mathematical expression of the new signal processing architecture introduced in Section 3.2.

### 3.1.2 Extension to Two Dimensions

While the implementation of Equation 3.2 is straightforward in one dimension, this is not always the case for two dimensions. It is only uncomplicated in two dimensions when both the transform, **T**, and the linear operator, **C**, are separable. The subband compression transform that we use in this thesis research is separable, and therefore can be written as $\mathbf{T}_{row}$ for the row operations and $\mathbf{T}_{col}$ for the column processing.

The correlation operation, however, is not separable. It becomes quite cumbersome to use Equation 3.2 when searching for two-dimensional patterns in the input signal. It is natural to use a Toeplitz matrix (**C** in Equation 3.2) to implement convolution of a one-dimensional signal, however, the equivalent structure necessary to perform convolution of a two-dimensional signal is not so obvious. While tractable mathematically and algorithmically, the resulting matrix is impractical and computationally expensive. When processing uncompressed imagery, correlation filters avoid the issue completely by not utilizing a matrix implementation, but rather operating in the frequency domain, as explained in Section 2.3. Thus, we must find another structure to operate two-dimensionally in the compressed domain and execute an 'inverse transformed' correlation filter.

## 3.2 Merging the Interpolation and Correlation Filters

To provide an alternative method of implementing an 'inverse transformed' correlation filter, we turn to polyphase structures. These allow us to manipulate the correlation filter and operate it within the inverse transform, i.e., the interpolation filters, of the individual subbands within a subband coder.

### 3.2.1 Polyphase Structures

Fundamental to many signal processing applications is the polyphase decomposition introduced by Bellanger, et. al., [BBC76]. While many sources discuss the concept at great length [CR81, CR83, Vai88, Vai90, Vai93, SN96], we review the basic equations here.

Consider a transfer function, $H(z)$, which represents a digital filter.

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-n} \qquad (3.16)$$

A polyphase decomposition is simply a way of splitting a filter into its even and odd components as Equation 3.17 illustrates.

$$H(z) = H_e(z^2) + z^{-1} H_o(z^2) \qquad (3.17)$$

$$\text{with } H_e(z) = \sum_{n=-\infty}^{\infty} h(2n) z^{-n}, \text{ and } H_o(z) = \sum_{n=-\infty}^{\infty} h(2n+1) z^{-n} \qquad (3.18)$$

Also useful to our cause is the noble identity for interpolators [CR81, CR83, Vai88, Vai90, Vai93, SN96], defined in Figure 3.1


Figure 3.1 **Noble Identity for an Interpolator**

Combining the polyphase decomposition with the above noble identity, results in a new construction known as an efficient interpolator [CR81, CR83, Vai88, Vai90, Vai93, SN96], and exhibited for one-dimension in Figure 3.2.

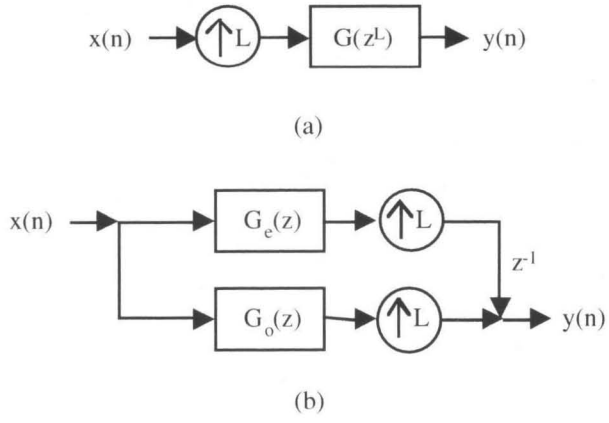


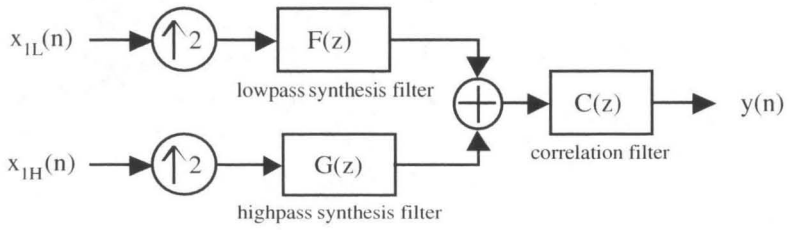

Figure 3.2 **An Efficient Interpolator 'Identity'**
Figure (a) portrays a standard interpolation filter, while (b) illustrates its efficient interpolator counterpart. It is called an efficient interpolator because the filters in (b) operate at half the sampling rate as in (a). Structures (a) and (b) are equivalent. Either one may replace the other.

Note that the structure of Figure 3.2(a) can replace that in Figure 3.2(b) and vice-versa. Thus, we say that Figure 3.2 illustrates an efficient interpolator 'identity'. The efficient interpolator becomes the architectural building block used to pull the correlation filter from the uncompressed domain into the compressed space, thus dissolving the boundary between the recognition and compression components.

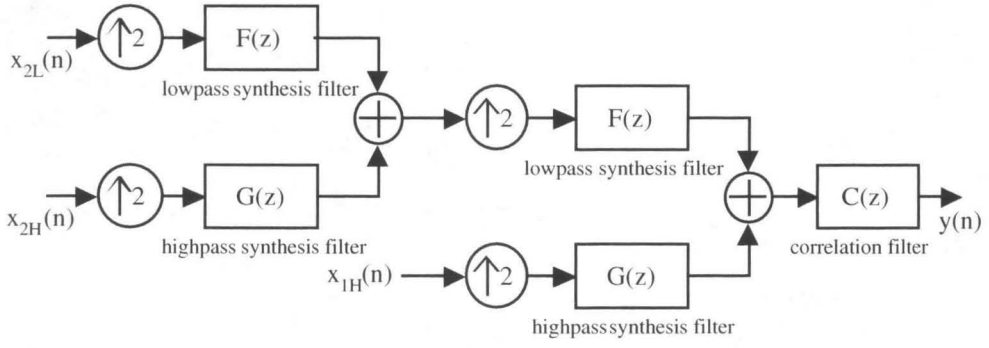
### 3.2.2 Rearranging the Architecture

We initiate this discussion with a review of the *baseline compression recognition system* (defined in Section 2.2 and diagrammed in Figure 2.13). In this case, the system fully reconstructs the image prior to performing recognition with the correlation filter.

Figure 3.3 diagrams the reconstruction and recognition portions of such a system for a one-dimensional signal with both a one level and two level decomposition.




(a)




(b)

Figure 3.3 **Synthesis and Correlation of a Baseline Compression Recognition System** The signal is fully reconstructed prior to object recognition with the correlation filter. Diagram (a) depicts a one level decomposition and (b) a two level.

Throughout the ensuing discussion, we represent all filters in their $z$ Transform notation. For simplicity of notation, we represent lowpass and highpass synthesis filters as $F(z)$ and $G(z)$, respectively, although traditionally, these two filters are denoted as $G_0(z)$ and $G_1(z)$, respectively.

Now we will describe a method of merging the correlation filter, $C(z)$, with the subband synthesis filters. In the interest of clarity, we will limit the discussion to a two level, one-dimensional system. In this case, the rearrangement of the architecture is a four part process which we illustrate in Figures 3.4 - 3.7.

*Step One:* The correlation filter, $C(z)$, of Figure 3.3(b) can migrate behind the sum because addition and correlation are both linear operations. Now it is simple to combine the correlation filter with the subband synthesis filters of level one. We define the new filters in Equation 3.19. The new architecture is shown in Figure 3.4.

$$CF(z) = C(z)F(z) \qquad CG(z) = C(z)G(z) \tag{3.19}$$

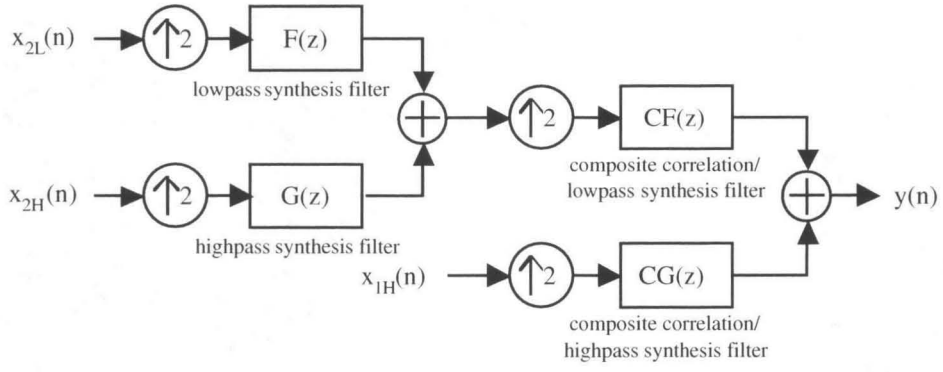



Figure 3.4 **Merging the Correlation and Subband Synthesis Filters: Step One**
The correlation filter moves back into level one of the subband decomposition.

*Step Two:* Before we can continue the migration of the recognition filter, we must employ the efficient interpolator of Figure 3.2. It is necessary to use this polyphase structure because the subband synthesis system is a linear time varying system due to the interpolators. We use the construction of Figure 3.2(b) to replace the low frequency leg of level one in Figure 3.4, resulting in the diagram of Figure 3.5. The symbols $CF_o(z)$ and $CF_e(z)$ represent the odd and even branches of the composite synthesis/recognition filter, $CF(z)$.

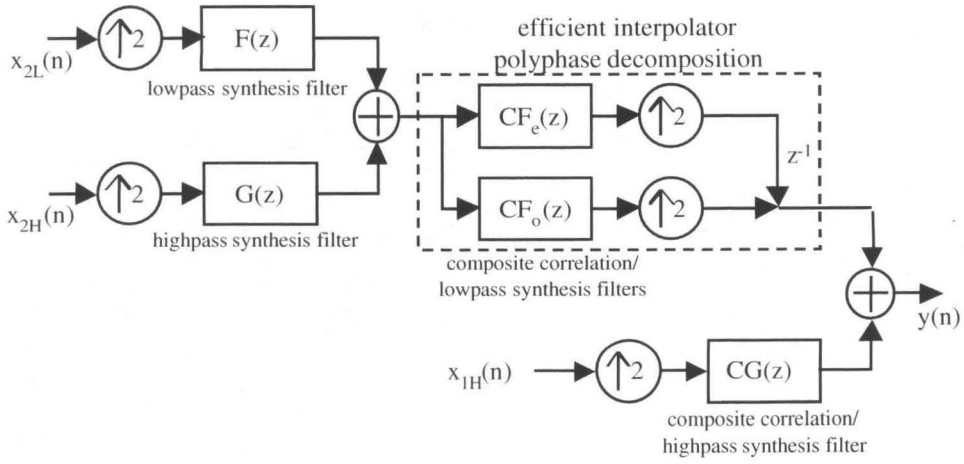



Figure 3.5 **Merging the Correlation and Subband Synthesis Filters: Step Two**
Substitute an efficient interpolator in the low frequency channel of level one.

*Step Three:* The above substitution allows us to push the recognition filter further back into the subband architecture. We move the polyphase structure on level one behind the summation at level two, and then combine the participating filters. Equation 3.20 describes the newly merged filters in a manner similar to Equation 3.12.

$$\begin{aligned} CF_oF(z) &= CF_o(z)\cdot F(z) \qquad CF_eF(z) = CF_e(z)\cdot F(z) \\ CF_oG(z) &= CF_o(z)\cdot G(z) \qquad CF_eG(z) = CF_e(z)\cdot G(z) \end{aligned} \tag{3.20}$$

The new filters, $CF_oF(z)$, $CF_eF(z)$, $CF_oG(z)$, and $CF_eG(z)$, retain the odd and even structures of the underlying filters $CF_o(z)$ and $CF_e(z)$. Now the architecture consists of two efficient interpolators, one in each branch of the decomposition's second level, as displayed in Figure 3.6.

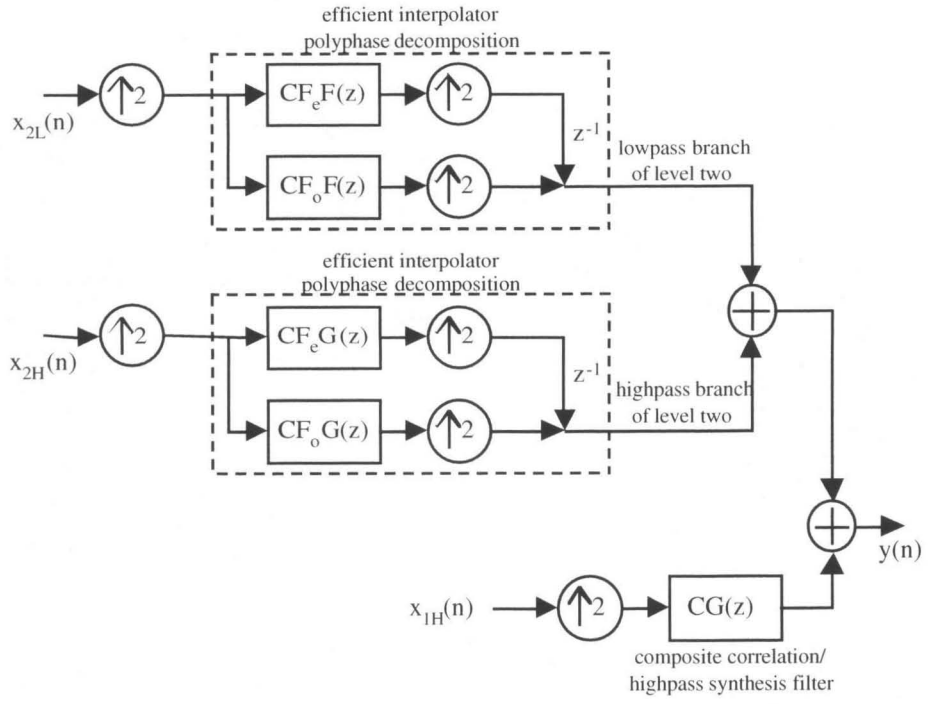



Figure 3.6 **Merging Correlation and Subband Synthesis Filters: Step Three** Move the efficient interpolators of level one back into level two, and merge the filter operations.

*Step Four:* Finally, to complete the new architecture, we once again make use of the efficient interpolator 'identity'. As illustrated in Figure 3.2, the efficient interpolator of diagram (b) is equivalent to its single branch counterpart in (a). In this final step we make just such an 'inverse' replacement. Additionally, we combine the two summations of the first and second level resulting in the parallel construction of Figure 3.7. Because the four filters, $CF_oF(z)$, $CF_eF(z)$, $CF_oG(z)$, and $CF_eG(z)$, possess odd and even structures, we can combine them via Equation 3.17 to form $CFF(z)$ and $CFG(z)$. Figure 3.7 presents the PRSC architecture for a one-dimensional signal and two decomposition levels.

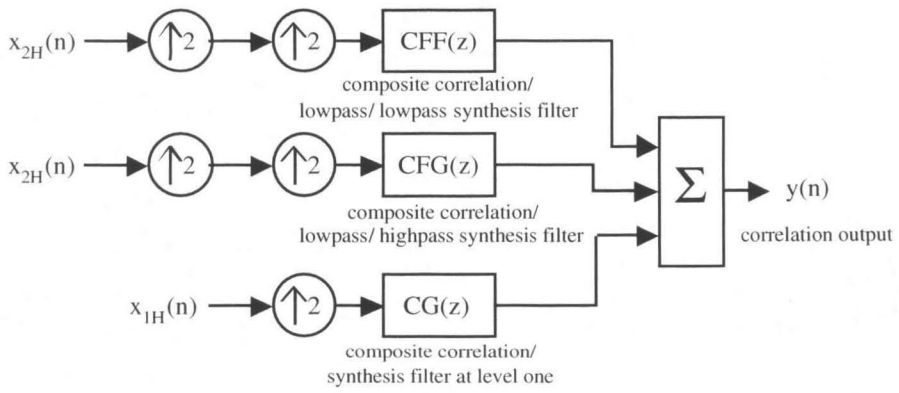



Figure 3.7 **Merging Correlation and Subband Synthesis Filters: Step Four** Substitute a single channel interpolator and filter for the efficient interpolators of level two. The final architecture consists of only parallel branches of computation. In each branch the subband synthesis and recognition filters have been combined into one composite synthesis recognition filter.

We see in Figure 3.7 that only parallel branches of computation comprise the PRSC. Each subband of the multiresolution decomposition forms its own distinct branch. Moreover, the composite filters within the PRSC embody a complete union of the compression and recognition domains.

Finally, correlation filters are typically implemented in the frequency domain, as expressed in Section 2.3. In the frequency domain, the correlation filter operation takes the form of a point-to-point multiplication rather than a full convolution. This method greatly speeds up the computation. We can also take advantage of this computational simplicity in the PRSC. Toward this end, we include a Fast Fourier Transform (FFT) of each subband's input signal in Figure 3.8. The upsampling is done in the frequency domain (as a signal replication), and the composite synthesis/recognition subband filters are stored in their frequency domain. Lastly, we must perform an Inverse Fast Fourier Transform (IFFT) of the summation output to produce a correlation surface in the time domain. We display these details of the implementation in Figure 3.8 to aid our assessment of the computational efficiency, as reviewed in the Section 3.3.

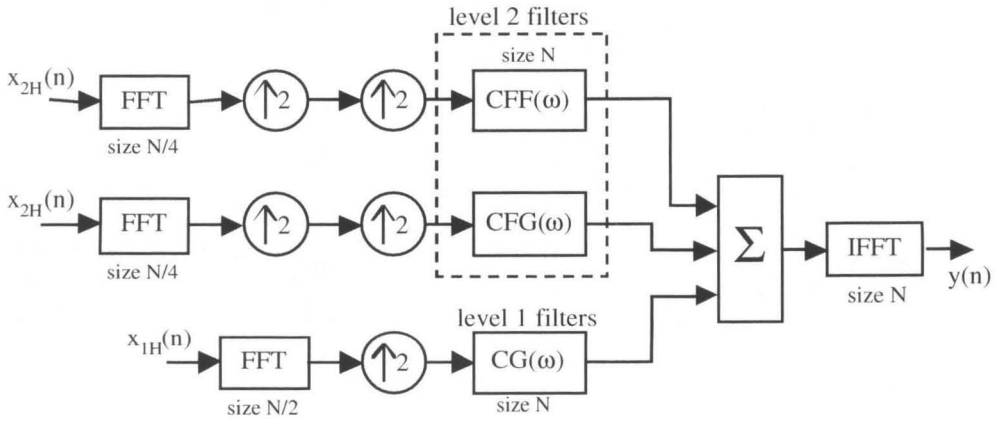



Figure 3.8 **Computationally Efficient Implementation of the PRSC**
As is typical of signal processing architectures, the PRSC is implemented in the frequency domain. The size (on one side) of the data arrays at each step is given in the above figure. Notice that all the correlation filters are of size $N$ by $N$, because the signal is upsampled (replication in the frequency domain) prior to correlation. The size of each FFT, however, is commensurate with the size of its subband input signal.

### 3.2.3 Extension to Two Dimensions

By continuing the above process outlined in Section 3.2.2 recursively, we can employ the new PRSC architecture of Figure 3.8 over any number of decomposition levels. Multiple polyphase branches have to be made for multiple numbers of interpolators, but this process occurs only in the design stage. Operation of the PRSC is a purely parallel process with its filters designed apriori. Figure 3.9 illustrates the PRSC architecture for a three level decomposition of a two-dimensional signal. In Figure 3.9, the naming convention of the composite synthesis/recognition filters follows that outlined with Equations 3.19 and 3.20, except that we use Fourier notation rather than $z$ Transform notation.

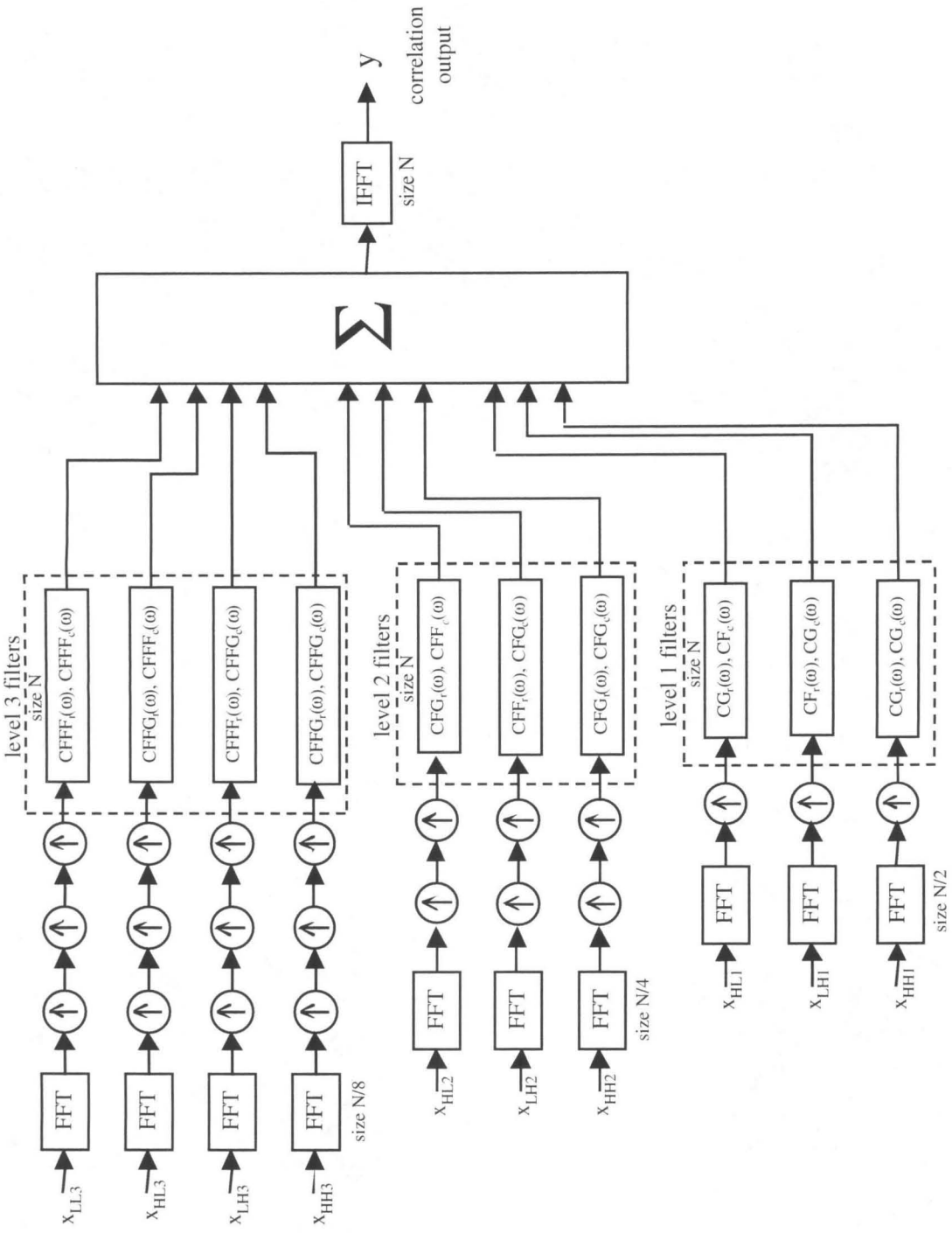



Figure 3.9 **The PRSC Architecture for a Three Level Image Decomposition** The PRSC dissolves the boundary between the compression and recognition domains. Within each branch all the subband synthesis filters and the recognition filter are combined into one composite synthesis/recognition filter. The result is ten parallel branches of computation. The FFTs are necessary only for computational efficiency.

The result, $y(m,n)$, is the correlation surface which exactly duplicates the one achieved by the baseline compression recognition system. (Such a system, albeit for a two level decomposition of a one-dimensional signal, is shown in diagram (b) of Figure 3.3.) That is, the process exhibited in Figure 3.9 is equivalent to fully reconstructing the two-dimensional image and then performing the correlation filtering operation on the reconstructed signal, $\hat{x}(m,n)$. The PRSC output is also equivalent to the correlation response achieved by operating on the original signal, $x(m,n)$, limited only by the veracity of the subband transform.

In Figure 3.9, we again display the sizes of the operations so as to facilitate our assessment of the computational expense, as reviewed in the following section.

## 3.3 Computational Complexity

In this section we compare the computational requirements for three systems: 1) the novel PRSC architecture, 2) the baseline compression recognition system (defined in Section 2.2), and 3) correlation on uncompressed images. We consider only multiplication operations in our assessment of the computational complexity of these systems. Some of the operations in the first two systems can be parallelized. Thus, we examine not only the total computation required, but also the maximum number of computes necessary to arrive at an output. The latter we call the *effective computation*.

### 3.3.1 Total Computation

#### Baseline System and Uncompressed Case

First, we review the computation involved for performing correlation on uncompressed images, as well the computation involved in the baseline compression

recognition system. In both cases, we implement the correlation filter in the most computationally efficient way, i.e., in the frequency domain. Figure 3.10 diagrams the major processing steps of the baseline system.

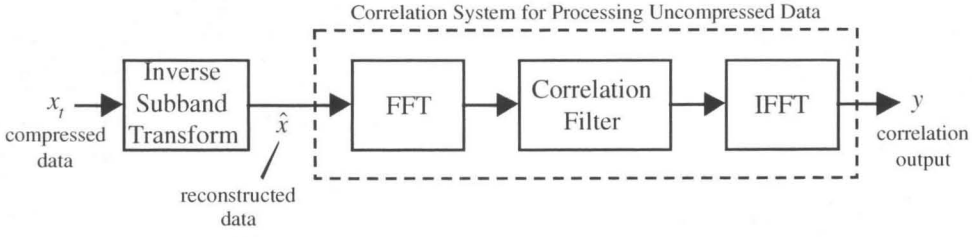



Figure 3.10 **Baseline Compression Recognition System**

Notice that the baseline compression recognition system is merely an augmentation of the system for correlation on uncompressed images.

We now quantify the computation for each step shown in Figure 3.10, assuming an original image of size $N$ by $N$, an inverse (synthesis) subband filter of length $M$, and a decomposition of $K$ levels. In such a system, the dimension (on one side) of a subband on level k is given by $\frac{N}{2^k}$. First, we examine the inverse subband transform (IST) computation. The computation required for the synthesis filters to interpolate one subband is $6M\left(\frac{N}{2^k}\right)^2$. Therefore, the computation required to reconstruct an image from its compressed state is

$$\begin{aligned} \text{IST Computation} &= 4\sum_{k=1}^{K} 6M\left(\frac{N}{2^k}\right)^2 \\ &= 6MN^2\sum_{k=1}^{K} 4^{1-k} \end{aligned} \qquad (3.21)$$

Both the FFT and IFFT operations take the same number of computes, as specified in Equation 3.22.

$$\text{FFT Computation} = \text{IFFT Computation} = N^2 \log_2 N \qquad (3.22)$$

Because the filter is implemented in the frequency domain, its operation takes the form of a point-to-point multiplication rather than a full convolution.

$$\text{Correlation Computation} = N^2 \qquad (3.23)$$

Equations 3.21 - 3.23 provide the building blocks for our computation calculations. By adding the appropriate constructs we arrive at the following calculations of total computation, *P*, for the two systems diagrammed in Figure 3.10.

$$\text{No Compression}: \quad P = N^2\left(2\log_2 N + 1\right) \qquad (3.24)$$

$$\text{Baseline System}: \quad P = N^2\left(2\log_2 N + 1 + 6M\sum_{k=1}^{K} 4^{1-k}\right) \qquad (3.25)$$

**PRSC Architecture**

Now we can turn our attention to the PRSC. The process in Figure 3.9 displays multiple parallel branches. Within each branch on level *k*, there is an FFT of size $\frac{N}{2^k}$.

$$\text{Branch FFT Computation} = \left(\frac{N}{2^k}\right)^2 \log_2\left(\frac{N}{2^k}\right) \qquad (3.26)$$

There are three FFTs on all levels but the lowest, in which case there are four.

$$\text{Total FFT Computation} = 4\left(\frac{N}{2^K}\right)^2 \log_2\left(\frac{N}{2^K}\right) + 3\sum_{k=1}^{K-1}\left(\frac{N}{2^k}\right)^2 \log_2\left(\frac{N}{2^k}\right) \qquad (3.27)$$

The IFFT computation occurs only at the end, and is of the reconstructed image size, *N*.

$$\text{IFFT Computation} = N^2 \log_2 N \qquad (3.28)$$

Each parallel branch contains a composite synthesis/correlation filter of size $N$. There are, however, $[4 + 3(K\text{-}1)]$ branches in a PRSC with $K$ decomposition levels. Again, this filter is implemented in the frequency domain, so its operation takes the form of a point-to-point multiplication rather than a full convolution. This filter operation is of size $N$ because it occurs after the upsampling operations.

$$\text{Correlation Computation} = \left[4 + 3(K-1)\right]N^2 \qquad (3.29)$$

By adding Equations 3.27 - 3.29, we can write the total computation necessary for the PRSC operation.

$$P = 4\left(\frac{N}{2^K}\right)^2 \log_2\left(\frac{N}{2^K}\right) + 3\sum_{k=1}^{K-1}\left(\frac{N}{2^k}\right)^2 \log_2\left(\frac{N}{2^k}\right) + \left[4 + 3(K-1)\right]N + N^2 \log_2 N \qquad (3.30)$$

After algebraic manipulation and combining of terms, this value becomes

$$P = \left[1 + 4^{1-K}\right]N^2 \log_2 N + \left[1 + K\left(3 - 4^{1-K}\right)\right]N^2 + 3N^2\sum_{k=1}^{K-1} 4^{-k}\left(\log_2 N - k\right) \qquad (3.31)$$

**Summary and Comparisons**

We summarize the *total computation* necessary for each system's operation in the list below.

- No Compression: $P = N^2\left(2\log_2 N + 1\right)$
- Baseline System: $P = N^2\left(2\log_2 N + 1 + 6M\sum_{k=1}^{K} 4^{1-k}\right)$
- PRSC: $P = \left[1 + 4^{1-K}\right]N^2 \log_2 N + \left[1 + K\left(3 - 4^{1-K}\right)\right]N^2 + 3N^2\sum_{k=1}^{K-1} 4^{-k}\left(\log_2 N - k\right)$

An important observation is that the PRSC computation is not dependent on the value of $M$, i.e., the length of the subband synthesis filter. This is because the subband synthesis filter is incorporated into the correlation filter prior to the system operation.

Figure 3.11 compares the total computation expended by each of the three systems for varying values of $N$. We select one value for $M$ and show the baseline computation over several values of $K$; likewise we depict the computation executed by the PRSC for several values of $K$ and $M$ is not a factor in this case.

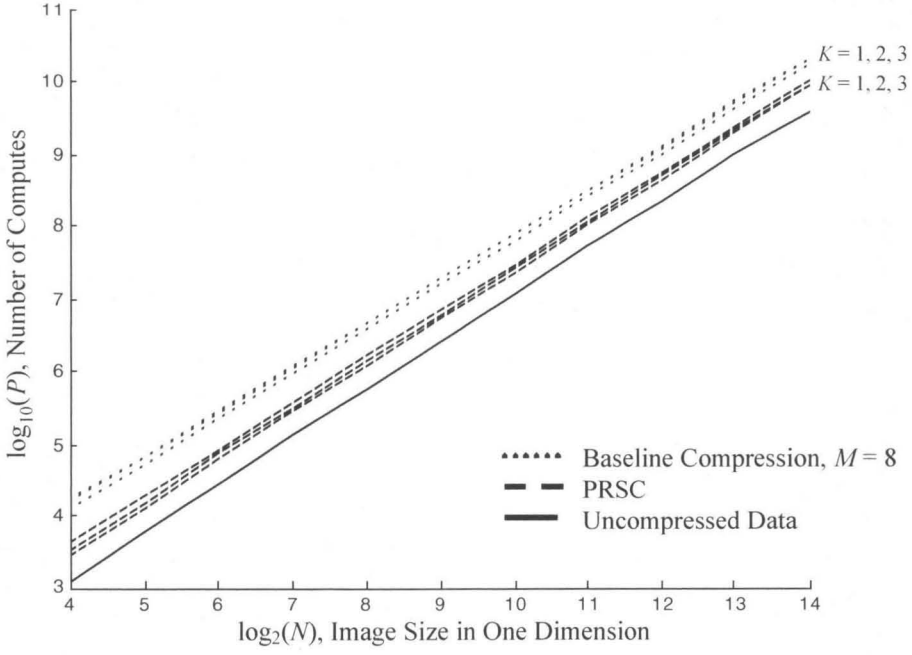



Figure 3.11 **Comparison of Total Computation for Three Systems**
The PRSC fares much better than the baseline compression recognition system. Note also that the PRSC computation is not dependent on the length of the subband synthesis filter, $M$, as the baseline system is. For larger values of $M$, the baseline computation increases. Both methods of processing compressed images consume more total computation than when working with uncompressed images.

### 3.3.2 Effective Computation

Recall that this term is defined to represent the total number of computations required to arrive at a result. By parallelizing some operations, the *effective computation* can be much lower than the *total computation* of a system.

**PRSC Architecture**

For many applications, the speed of computation is more important than the total computation involved. In this regard, the parallelism inherent in the PRSC serves us well. Operation of the PRSC can be viewed as ten parallel processing steps, with the total processing time dictated by the longest leg (i.e., one of the largest subbands). The largest subband requires an FFT of size $\frac{N}{2}$, and a filter operation of size $N$. Lastly, the final stream of data requires an IFFT of size $N$. Thus, the effective computation of the PRSC is reflected in the following equation.

$$\begin{aligned} P' &= \left(\frac{N}{2}\right)^2 \log_2\left(\frac{N}{2}\right) + N^2 + N^2 \log_2 N \\ &= N^2\left(\frac{1}{4}\log_2 N - \frac{1}{4} + \log_2 N + 1\right) \\ &= N^2\left(\frac{5}{4}\log_2 N + \frac{3}{4}\right) \end{aligned} \qquad (3.32)$$

**Baseline System and Uncompressed Case**

The inverse subband transform can also be performed in parallel in a manner analogous to the PRSC. Thus, the effective computation of the baseline compression recognition system is contingent upon only one of its largest subbands. The effective computation required to fully reconstruct the image is derived from Equation 3.21 and given below.

$$\text{Effective IST Computation} = \frac{3}{2} MN^2 \qquad (3.33)$$

We refer to Figure 3.10 to determine the effective computation of the baseline system. After adding the effective IST value with the number of computations required

by the FFT, correlation filter, and IFFT operations (all of size $N$), we arrive at the result below.

$$\begin{aligned} P' &= \frac{3}{2} MN^2 + N^2 \log_2 N + N^2 + N^2 \log_2 N \\ &= N^2 \left( 2 \log_2 N + 1 + \frac{3}{2} M \right) \end{aligned} \tag{3.34}$$

Finally, the standard method of working with uncompressed data is completely serial. Therefore, $P = P'$ and is given below.

$$P' = N^2 \left( 2 \log_2 N + 1 \right) \tag{3.35}$$

**Summary and Comparisons**

We summarize the *effective computation* necessary for each system's operation in the list below.

- No Compression: $P' = N^2 \left( 2 \log_2 N + 1 \right)$ (3.36)
- Baseline System: $P' = N^2 \left( 2 \log_2 N + 1 + \frac{3}{2} M \right)$ (3.37)
- PRSC: $P' = N^2 \left( \frac{5}{4} \log_2 N + \frac{3}{4} \right)$ (3.38)

We immediately see that the baseline system provides the slowest output response. The most important observation, however, is that the PRSC provides a faster system response than that provided by working with uncompressed imagery. As we see by subtracting Equation 3.38 from 3.36, we can arrive at a result faster with compressed images than with uncompressed images, by the following number of computations:

$$D' = N^2 \left( \frac{3}{4} \log_2 N + \frac{1}{4} \right) \tag{3.39}$$

For large values of $N$, this difference can become quite significant as exhibited in Figure 3.12, where we characterize the effective computation of all three systems. We plot the computation of the baseline system for only one value of $M$; for larger values the computation increases.

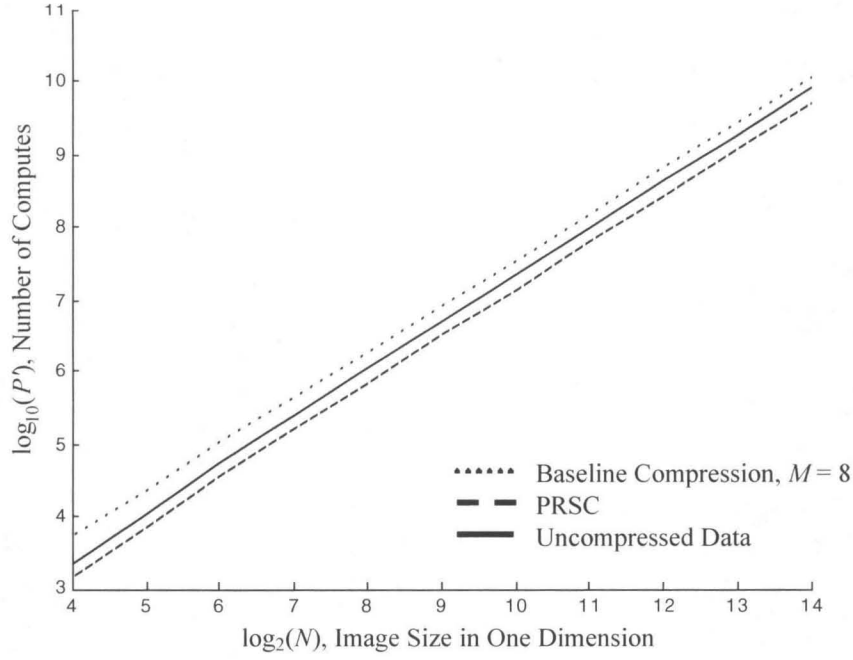



Figure 3.12 **Comparison of Effective Computation for Three Systems**
It is most important to note that the PRSC proves faster than even the case of working with uncompressed data. We display the effective computation of the baseline system for a single value of $M$; it only increases for larger values.

Moreover, by taking a ratio of Equation 3.36 and 3.38, we can estimate the savings *factor*, $F'$, achieved in the effective computation by using compressed data. In the limit, the factor goes to the value of 1.6, and is very close to this value even for small image sizes, as Figure 3.13 shows.

Effective Computation Savings Factor:
$$F' = \frac{8\log_2 N + 4}{5\log_2 N + 3} \qquad (3.40)$$
$$F'_{N\to\infty} = 1.6$$

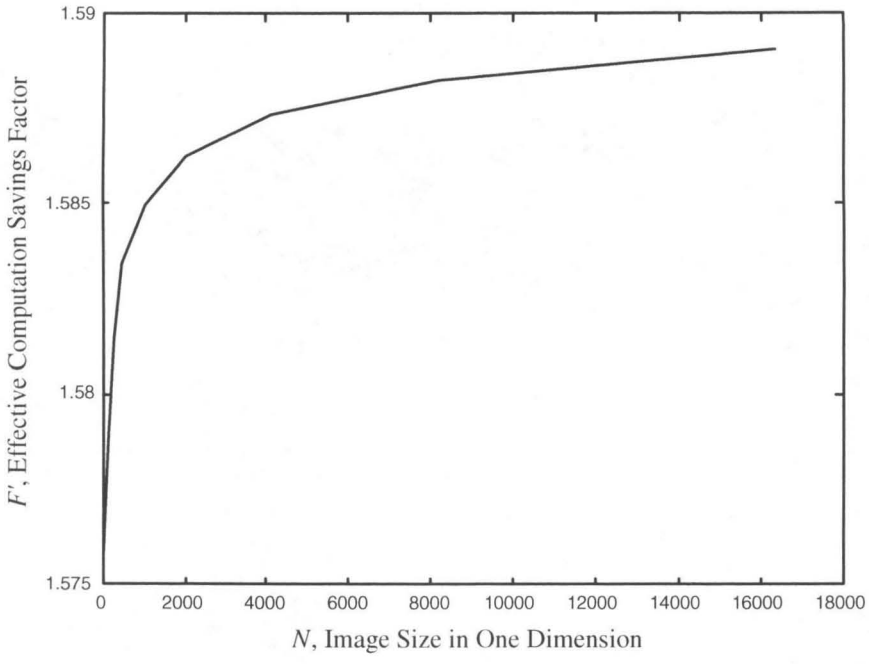



Figure 3.13 **Effective Computation Savings Factor with the PRSC**
The computational savings for operating the PRSC on compressed imagery, as opposed to standard correlation on uncompressed imagery, goes to 1.6 in the limit. It is very close to that value even for small image sizes.

## 3.4 Summary

The PRSC is a new architecture for performing object recognition on compressed image data. It is equivalent to correlation filtering on the reconstructed image and has several novel benefits.

- It is a *universal architecture* for operating any linear filter directly on the multiresolution data of a subband coder.
- It produces a response *identical* to that produced by the reconstructed image.
- It results in a response *identical* to the equivalent filtering on the *original* image, within the fidelity of the subband transform.

- It provides a *faster response* than the equivalent filtering operation on uncompressed data, by a *factor of* $F' = 1.6$ as $N$ (dimension on one side of a square image) goes to infinity. The factor is very close to 1.6 even for very small $N$.
- It requires *less total computation* than the baseline compression recognition system. The difference between the two systems grows larger with increasing image size.
- It combines the global correlation filter and each subband interpolation filter *into one composite filter* resulting in a *new correlation filtering technique* for recognizing patterns in compressed imagery.

Furthermore, this chapter introduces a new mathematical formulation to operate linear systems directly in a transform domain.

- It is *universal* for a unitary transform and a linear operator.
- It produces a response *identical* to that produced by the untransformed input.

In conclusion, the PRSC dissolves the boundary between the two domains of image compression and object recognition and combines them into one homogenous operation. Consequently, this chapter addresses two of the questions submitted for this thesis investigation.

- *We can perform object recognition in compressed imagery with the same accuracy as in uncompressed imagery, limited only by the veracity of the subband transform.*
- *A correlation filter can be adapted to operate directly in the subband space in a computationally efficient manner.*

To summarize: By using losslessly transformed data, the novel PRSC architecture performs object recognition as if on the original data, and does so with faster execution.

# Chapter 4

# Joint Optimization of the Compression and Recognition Filters

Subband and correlation filters work in concert with each other in our novel architecture, the Pattern Recognition Subband Coder, presented in the previous chapter. The subband filters provide the forward and inverse compression transform, while the correlation filters perform object recognition within the individual subbands. This chapter addresses the ensuing question, "*How do we jointly optimize the performance of the recognition and compression filters?*" Our approach dictates that system performance is best served if:

1) the training phase tunes the recognition filters to the compressed data, and

2) the data compression elicits a maximal response from the recognition filters.

We provide one algorithm for meeting these two objectives via a modified QMF design for subband decomposition.

The chapter is organized as follows. First, in Section 4.1, we present a matrix formulation of the subband filters, which, as Section 4.2 explains, forms the basis for our unique approach to QMF design. Section 4.3 then defines the new recognition criterion

that we incorporate in the QMF design. Additionally, Section 4.3 details the interaction of the correlation filter and QMF design, which the optimization algorithm interweaves. A quantizer and encoding scheme necessary to assess system performance at varying bit rates is described in Section 4.4. Finally, we present results of the Pattern Recognition Subband Coder performance for both optimized and unoptimized conditions in Section 4.5, as well as quantization effects on optimized performance. Future extensions to the optimization design technique are suggested in Section 4.6. Lastly, Section 4.7 summarizes the chapter and discusses the robust performance of the Pattern Recognition Subband Coder, its benefits and limitations.

## 4.1 Matrix Formulation of a Subband Decomposition

A review of Equations 3.2 - 3.4 suggests that it would be helpful to formulate the subband transforms in matrix representations. Section 3.1 briefly introduced this construction; we present a more formal treatment here. In Section 3.1, we discussed the **T** matrix for the purpose of a single recursion level. We present the more general case of multiple recursion levels here. The formulation presented here evolved from the work of Mahalanobis, et. al., [MS+93] which introduced a matrix representation of subband decomposition based on the Hadamard transform.

The **T** matrix representation is useful not only for analytical purposes, but also for the construction of the QMF banks as we explain in Section 4.2. Central to the following discussion is the fact that the matrix representation we develop combines the filtering and downsampling of a subband analysis transform over multiple levels in a matrix structure. In the interest of clarity, we limit the discussion to the one-dimensional case.

Let $\mathbf{x} = [x(0)\, x(1) \ldots x(L\text{-}1)]$ be a vector of length $L$ which contains the samples of the signal $x(n)$. For simplicity, we shall assume that $L = 2^M$, where $M$ is any positive integer. Let us further assume that the subband analysis filters $h_0(n)$ and $h_1(n)$ of length $N$, decompose $x(n)$ into two new subsequences $x_0(n)$ and $x_1(n)$. In matrix-vector notation, the decomposition of $\mathbf{x}$ into two subsequences $\mathbf{x}_0$ and $\mathbf{x}_1$ of length $(L+N\text{-}2)/2$ can be succinctly expressed as

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{A}_1\mathbf{x} \\ \mathbf{x}_2 &= \mathbf{A}_2\mathbf{x} \end{aligned} \tag{4.1}$$

and

$$\mathbf{x} = \mathbf{A}_1^t\mathbf{x}_1 + \mathbf{A}_2^t\mathbf{x}_2 \tag{4.2}$$

where $\mathbf{A}_0$ and $\mathbf{A}_1$ are unitary transform matrices of size $(L+N\text{-}2)/2$ by $L$. $\mathbf{A}_0$ and $\mathbf{A}_1$ satisfy the conditions $\mathbf{A}_0^t\mathbf{A}_0 = \mathbf{I}$, $\mathbf{A}_1^t\mathbf{A}_1 = \mathbf{I}$, and $\mathbf{A}_0^t\mathbf{A}_1 = \mathbf{0}$. We construct these matrices from the subband analysis filters as follows:

$$\mathbf{A}_0 = \begin{bmatrix} h_0(1) & h_0(0) & 0 & 0 & \cdots & \cdots & 0 & 0 \\ h_0(4) & h_0(3) & h_0(1) & h_0(0) & \cdots & \cdots & 0 & 0 \\ \vdots & & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & h_0(N-1) & h_0(N-2) & h_0(N-3) & h_0(N-4) \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & h_0(N-1) & h_0(N-2) \end{bmatrix} \tag{4.3}$$

$$\mathbf{A}_1 = \begin{bmatrix} h_1(1) & h_1(0) & 0 & 0 & \cdots & \cdots & 0 & 0 \\ h_1(4) & h_1(3) & h_1(1) & h_1(0) & \cdots & \cdots & 0 & 0 \\ \vdots & & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & h_1(N-1) & h_1(N-2) & h_1(N-3) & h_1(N-4) \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & h_1(N-1) & h_1(N-2) \end{bmatrix}$$

The rows of $\mathbf{A}_0$ and $\mathbf{A}_1$ are shifted versions of the subband filters, appropriately padded with zeros to implement the required filtering and decimation. As a result, the

matrix-vector multiplication of Equation 4.1 yields the desired filtered and downsampled subband sequences. The relation between the subband sequences and the original signal can also be expressed as

$$\begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \end{bmatrix} \mathbf{x} = \mathbf{T}\mathbf{x} \qquad (4.4)$$

where $\mathbf{T} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \end{bmatrix}$ may be considered to be a one level decomposition matrix for splitting a signal into two subbands. Because the subband decomposition matrices are unitary, it follows that $\mathbf{T}^t\mathbf{T} = \mathbf{I}$, and

$$\mathbf{x} = \mathbf{T}^t \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} \qquad (4.5)$$

which is the inverse decomposition (i.e., synthesis) equation.

Furthermore, we can generalize the structure of $\mathbf{T}$ to represent any *one to M band* transformation including partial decompositions and the maximally decimated case. We focus our attention on the structure of $\mathbf{T}$ for the case of a dyadic hierarchical decomposition in which only the low frequency signal is recursively decomposed. Figure 4.1 diagrams this process for a three level decomposition of the signal $\mathbf{x}$.

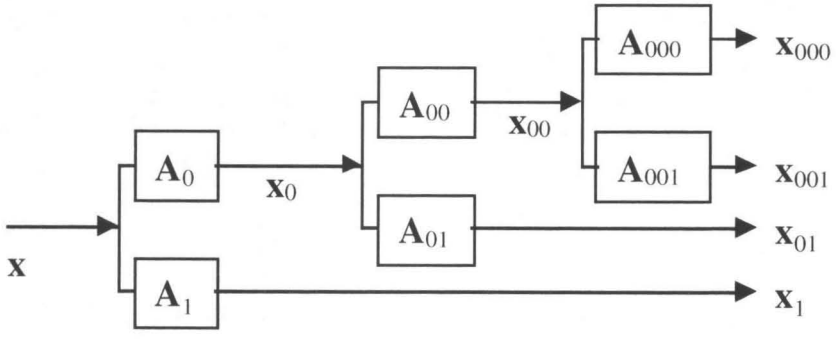



Figure 4.1 **Three Level Dyadic Decomposition of a One-Dimensional Signal** The low frequency signal is recursively decomposed, to form a one-dimensional subband decomposition.

It is easy to see that the process in Figure 4.1 leads to the following set of equations.

$$\begin{aligned}
\mathbf{x}_{000} &= \mathbf{A}_{000}\mathbf{x}_{00} = \mathbf{A}_{000}\mathbf{A}_{00}\mathbf{x}_{0} = \mathbf{A}_{000}\mathbf{A}_{00}\mathbf{A}_{0}\mathbf{x} \\
\mathbf{x}_{001} &= \mathbf{A}_{001}\mathbf{x}_{00} = \mathbf{A}_{001}\mathbf{A}_{00}\mathbf{x}_{0} = \mathbf{A}_{001}\mathbf{A}_{00}\mathbf{A}_{0}\mathbf{x} \\
\mathbf{x}_{01} &= \mathbf{A}_{01}\mathbf{x}_{0} = \mathbf{A}_{01}\mathbf{A}_{0}\mathbf{x} \\
\mathbf{x}_{1} &= \mathbf{A}_{1}\mathbf{x}
\end{aligned} \qquad (4.6)$$

In the above equations, the symbols '0' and '1' refer to the branch of the dyadic tree being traversed. The number of symbols in the subscripts indicates the stage of the decomposition. Thus, the subscript '001' refers to quantities in the second subband at the third stage of the decomposition process. All the subband decomposition matrices are of the type in Equation 4.3, but appropriately dimensioned to match the lengths of the input and output signals at each stage. Based on the above formulation, it is now easy to see that the relation between the input signal and the subband output is given by

$$\begin{bmatrix} \mathbf{x}_{000} \\ \mathbf{x}_{001} \\ \mathbf{x}_{01} \\ \mathbf{x}_{1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{000}\mathbf{A}_{00}\mathbf{A}_{0} \\ \mathbf{A}_{001}\mathbf{A}_{00}\mathbf{A}_{0} \\ \mathbf{A}_{01}\mathbf{A}_{0} \\ \mathbf{A}_{1} \end{bmatrix} \mathbf{x} = \mathbf{T}\mathbf{x} \qquad (4.7)$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{A}_{000}\mathbf{A}_{00}\mathbf{A}_{0} \\ \mathbf{A}_{001}\mathbf{A}_{00}\mathbf{A}_{0} \\ \mathbf{A}_{01}\mathbf{A}_{0} \\ \mathbf{A}_{1} \end{bmatrix} \qquad (4.8)$$

is the three level decomposition matrix that implements the process shown in Figure 4.1. Again, because each of the subband decomposition matrices which comprise $\mathbf{T}$ are unitary, it follows that $\mathbf{T}^{t}\mathbf{T} = \mathbf{I}$, and that the expression for synthesis is given by

$$\mathbf{x} = \mathbf{T}^{t} \begin{bmatrix} \mathbf{x}_{000} \\ \mathbf{x}_{001} \\ \mathbf{x}_{01} \\ \mathbf{x}_{1} \end{bmatrix} \qquad (4.9)$$

Now, consider the case where the analysis and synthesis filter banks are not unitary or orthogonal, but rather biorthogonal, as is the case with many wavelets. The above equations and discussion still hold, with the exception that the biorthogonal matrix **U** replaces $\mathbf{T}^t$, and $\mathbf{TU} = \mathbf{I}$. The matrix **U** is formed in a manner analogous to the way **T** is formed, except that the submatrices, $\mathbf{A}_0$ and $\mathbf{A}_1$, are composed from the subband synthesis filters, rather than from the subband analysis filters. In fact, as long as the forward transform is invertible, the matrix formulation **T** is applicable. (**T** has to be nonsingular, and thus invertible.) In which case $\mathbf{T}^{-1}$ is formed from the inverse transform filters, rather than the forward transform filters.

For illustration purposes, Figure 4.2 portrays a one, two, and three level **T** matrix

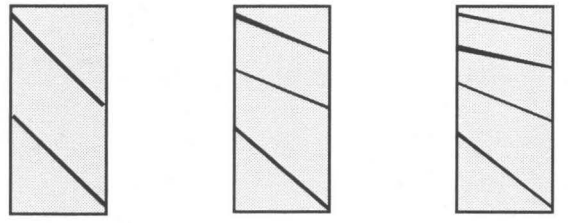



Figure 4.2 **T Matrix Images for each of Three Decomposition Levels**
Diagram (a), (b), and (c) depict a one, two, and three level **T** matrix, respectively, created from a four tap filter. In (a) the upper and lower diagonal bands represent the low and highpass subband filters, respectively. The upper two diagonals of (b) signify the low and highpass filtering process for level two, while the lowest diagonal designates the highpass operation of level one. Finally, in (c) the upper two diagonals accomplish the low and highpass filtering operations of level three; the next diagonal band depicts the highpass level two operation; the lowest diagonal band effects the level one highpass filtering process. Notice the increased decimation of the diagonal structures for the higher decomposition levels (lower levels of resolution).

constructed from a four tap filter. Matrix multiplication of the one level **T** matrix of Figure 4.2(a) with an image in both the row and column direction will result in a four subband decomposition akin to the one in Figure 2.5. Likewise, matrix multiplication of an image in both row and column directions with the three level **T** matrix of Figure 4.2(c) will result in a three level subband decomposition like the one illustrated in Figure 2.6.

In summary, the matrix construct **T** collapses the complete subband hierarchy into an aggregate structure which provides a direct channel from input to output and vice-versa. Multiplying an input image (in both the row and column direction) with an $M$ level **T** results in an $M$ level decomposition with $(3M + 1)$ bands. Thus, the matrix **T** provides a single to multiple band relationship. In addition, formulating the subband transform as a matrix construct greatly benefits an effective QMF design. All the requirements of perfect reconstruction are succinctly captured by the condition $\mathbf{T}^t\mathbf{T} = \mathbf{I}$, as elucidated in the next section.

## 4.2 Design of a Perfect Reconstruction Quadrature Mirror Filter

A quadrature mirror filter (QMF) bank consists of a lowpass and highpass filter, $h_0(n)$ and $h_1(n)$, whose purpose is to split a one-dimensional signal into two frequency bands. Such a frequency response is shown in Figure 4.3.

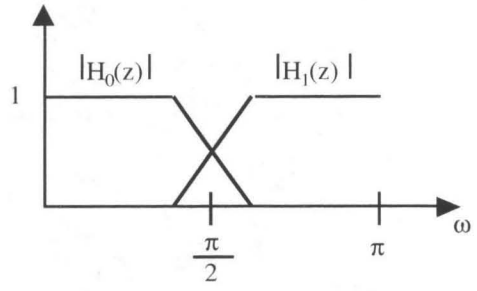



Figure 4.3 **Idealized QMF Frequency Response**

The name, quadrature mirror filter, is an apt nomenclature because its design specifies that the magnitude of the frequency response $H_0(z)$ be a mirror image of that of $H_1(z)$ with respect to the quadrature frequency $\pi/2$, as illustrated in Figure 4.3. QMFs were first introduced by Croisier, Esteban, and Galand [CEG76]. Since then, much work has been devoted to their development, with the most important step being the advance of perfect reconstruction, made independently by Vaidyanathan and Vetterli [Vai87a, Vai87b, Vet87]. Very readable texts on QMFs can be found in [Vai87b, Vai90].

A subband coder uses both a QMF analysis bank and a QMF synthesis bank, denoted by the variables $h$ and $g$, respectively. The two lowpass filters (one analysis and one synthesis) append the subscript '0', whereas the two highpass filters use the subscript '1'. To achieve a subband decomposition with multiple levels, the analysis filters are applied recursively to each succeeding lowpass output, also called a hierarchical decomposition. Likewise, the synthesis filters work in a recursive manner to recompose the signal without error. Section 2.4 discusses construction of a subband decomposition at greater length.

### 4.2.1 Our Approach Based on a Matrix Formulation of the Subband Decomposition

We apply the **T** matrix from Section 4.1 in a novel twist on a time domain design of perfect reconstruction QMF banks. As in most QMF design strategies, we design only the lowpass analysis filter, $h_0(n)$, and derive the remaining three filters from this single filter. Perfect reconstruction specifies that we design the filter $h_0(n)$ such that Equations 4.7 and 4.9 hold, allowing the subband data to recompose the original signal without any errors. In the absence of filter design constraints, three sources of error will impact the reconstructed signal: aliasing distortion (ALD), amplitude distortion (AMD), and phase distortion (PHD). Eliminating all three errors simultaneously results in our goal of a perfect reconstruction QMF bank.

Elegantly captured in the condition

$$\mathbf{T}^{\mathrm{T}}\mathbf{T} = \mathbf{I} \qquad (4.10)$$

are all the requirements of perfect reconstruction. (In Equation 4.10 and during the QMF design, we assume that our subband filters are not unitary, but orthogonal.) Essentially, this condition guarantees that the forward and inverse transforms cancel and that the system forms an all pass filter response. Hence, meeting the condition of Equation 4.10 ensures that both ALD and AMD are eliminated. Although not strictly necessary, we also assume that $h_0(n)$ is a linear phase filter which guarantees no PHD. The elimination of each of these three errors is examined in more detail in the following sections.

Thus, our approach to designing a perfect reconstruction QMF bank is to optimize the impulse response of $h_0(n)$ until the **T** matrix defined in Equation 4.8 becomes an orthogonal transformation. Three observations are in order. First, this approach is a generalized form of other time-domain techniques reported by Jain and Crochiere [JC84]

and Hemami [HG95, Hem95], and captures the ripple-balance conditions by forcing the structure of $\mathbf{T}^T\mathbf{T}$ to be identity (i.e., it is an allpass linear system with a net impulse response of only a delay). Second, the generalized structure of $\mathbf{T}$ facilitates orthogonality at multiple levels of decomposition so that the resulting QMF pair will yield perfect reconstruction up to the number of stages included in the definition of $\mathbf{T}$. Third, we exclude any explicit desired response specifications, such as passband, transition band, or stopband criterion, but allow the cutoff frequency to fall where it may.

**Initial Conditions**

We define the initial coefficients of the lowpass analysis filter, $h_0(n)$, with the matlab function, 'fir1'. We specify a lowpass linear phase finite impulse response (FIR) filter of length 16 with a cutoff frequency of $0.99(\pi/2)$. Note that because the filter is even symmetric, it has only eight degrees of freedom. Selecting the initial cutoff frequency close to the quadrature frequency of $(\pi/2)$, as well as using a FIR initialization facilitates a speedy and stable convergence.

In addition, we normalize the filter coefficients of $h_0(n)$ to sum to $\sqrt{2}$. This effectively achieves an even power distribution over all levels of the subband decomposition. A filter design which sums the $h_0(n)$ coefficients to one should have no effect on the performance of the recognition filter, however, the intrinsic effect on compression rate will alter the characteristics of the rate versus recognition performance curve.

**ALD Elimination**

To achieve aliasing cancellation we first follow the standard filter selection rules as given in [Vai87b, Vai90, Vai93].

$$h_1(n) = (-1)^n h_0(n), \quad g_0(n) = h_0(n), \quad g_1(n) = -h_1(n) \qquad (4.11)$$

The terms $h_0(n)$ and $h_1(n)$ are the lowpass and highpass analysis filters, respectively; likewise $g_0(n)$ and $g_1(n)$ are the lowpass and highpass synthesis filters, respectively. The analysis and synthesis lowpass filters are identical; the highpass filters are modulated versions of the lowpass filter; the synthesis filters are scaled versions of the analysis filters. (In our case, the scaling factor is one.)

Cancellation of the aliasing distortion, is also known as the orthogonality constraint because orthogonality between the analysis and synthesis filter banks ensures no aliasing effects. For only one level of decomposition, Equation 4.11 is enough. Recall though, that a pyramid decomposition downsamples between levels of recursion, providing a different sampling rate at each level. We need to ensure that the dual filter banks maintain orthogonality at all sampling rates present in the decomposition.

Thus, we turn to the condition of Equation 4.10 in a novel solution to this problem. We ensure this condition through two separate criteria. First, if **T** is truly an orthogonal transform, then all off-diagonal elements of its outer product will be zero. Thus, we include the sum of the squared off-diagonal elements as the first error term in the function which we numerically optimize.

$$\varepsilon_{\text{ald}} = \sum_{i \neq j} \left[ \left( \mathbf{T}^{\mathrm{T}} \mathbf{T} \right)_{ij} \right]^2 \qquad (4.12)$$

Second, the condition given in Equation 4.10, dictates that every element of the main diagonal of $\mathbf{T}^{\mathrm{T}}\mathbf{T}$ must be equal to one. To ensure this is the case, we employ Equation 4.13 as the second error term contributing to our numerical optimization function. It is denoted as $\varepsilon_{\text{amd}}$ because this criterion also guarantees AMD elimination.

$$\varepsilon_{amd} = \left(\mathbf{h}_0^T \mathbf{h}_0 - 1\right)^2 \qquad (4.13)$$

Using the T matrices to impose orthogonality conditions at all sampling rates is a time domain approach akin to the frequency domain solution provided in [Vai90].

**AMD Elimination**

As alluded to in the previous paragraph, we eliminate amplitude distortion by means of the criterion in Equation 4.13. It is also known as the allpass or ripple balance criterion. Equation 4.13 dictates that the autocorrelation response of $h_0(n)$ be a delta function of magnitude one. This, in turn, means that $h_0(n)$ will pass all frequencies with equal magnitude, thus eliminating any magnitude distortion. Additionally, Equation 4.13 represents constraining the total power of the $h_0(n)$ filter to be one.

This criterion is equivalent to allpass equalization which is sometimes used for frequency domain solutions. Such equalization is defined in the frequency domain by the following equation,

$$\left|H_0(z)\right|^2 + \left|H_1(z)\right|^2 = 1 \qquad (4.14)$$

Physically, equation 4.14 says that the power of $H_0(z)$ and $H_1(z)$ have a symmetry with respect to $\pi/2$, analogous to the symmetry of the $H_0(z)$ and $H_1(z)$ frequency response illustrated in Figure 4.3. Also, as suggested by the name ripple balance condition, this criterion tries to split the ripple energy equally between the lowpass and highpass filter responses. Again, the frequency domain criterion given in Equation 4.14 is equivalent to the time domain condition defined by Equation 4.13.

**PHD Elimination**

A linear phase filter is guaranteed to have no phase distortion and it is well known that a symmetric filter provides linear phase. We ensure linear phase by solving for only one half of the $h_0(n)$ coefficients at each iteration of our numerical optimization routine, and producing the entire set of $h_0(n)$ filter coefficients through the following definition.

$$h_0(n) = h_0(N_f - 1 - n) \quad \text{for } n = 0, \ldots, N_f - 1 \tag{4.15}$$

Consequently, the PHD elimination criterion does not add an error term to our numerical optimization function.

**DC Response Constraint**

We include a third design constraint which helps to stabilize the numerical optimization routine. We require the filter coefficients of $h_0(n)$ to sum to $\sqrt{2}$ with each iteration.

$$\varepsilon_{dc} = \left( \sum_{n=0}^{N_f - 1} h_0(n) - \sqrt{2} \right)^2 \tag{4.16}$$

where $N_f$ is the filter length of $h_0(n)$. This is also the minimum passband constraint for a lowpass filter. It basically ensures the magnitude of the filter response at dc. We expound more on the effects of the value selected for the sum in the "Initial Conditions" portion of this section.

**Optimization Equation**

Finally, a sum of the error terms derived from our design constraints form a function, $f$, which we optimize numerically, via the matlab routine, 'fminu'.

$$f = \varepsilon_{\text{ald}} + \varepsilon_{\text{amd}} + \varepsilon_{\text{dc}}$$

$$f = \sum_{i \neq j} \left[ \left( \mathbf{T}^{\mathrm{T}} \mathbf{T} \right)_{ij} \right]^2 + \left( \mathbf{h}_0^{\mathrm{T}} \mathbf{h}_0 - 1 \right)^2 + \left( \sum_{n=1}^{N_f} h_0(n) - \sqrt{2} \right)^2 \qquad (4.17)$$

The routine finds the coefficients of $h_0(n)$ that minimize the function $f$. In our experiments, the final solution produces a total error of the order $10^{-6}$, thus asymptotically reaching a perfect reconstruction QMF, limited only by machine precision.

### 4.2.2 Results

Using a filter length, $N_f$ = 16, our QMF design resulted in the time domain plot of $h_0(n)$ depicted in Figure 4.4. The magnitude of the frequency response, $H_0(z)$, of this filter is displayed in Figure 4.5. We refer to the QMF developed in this section as our baseline QMF throughout the rest of the thesis.

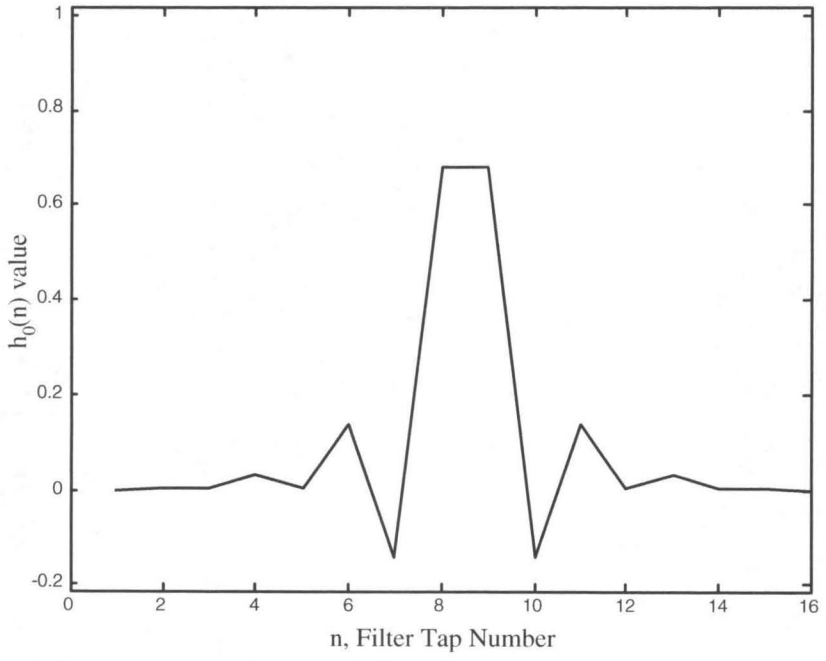



Figure 4.4 **Baseline QMF Time Domain Response**

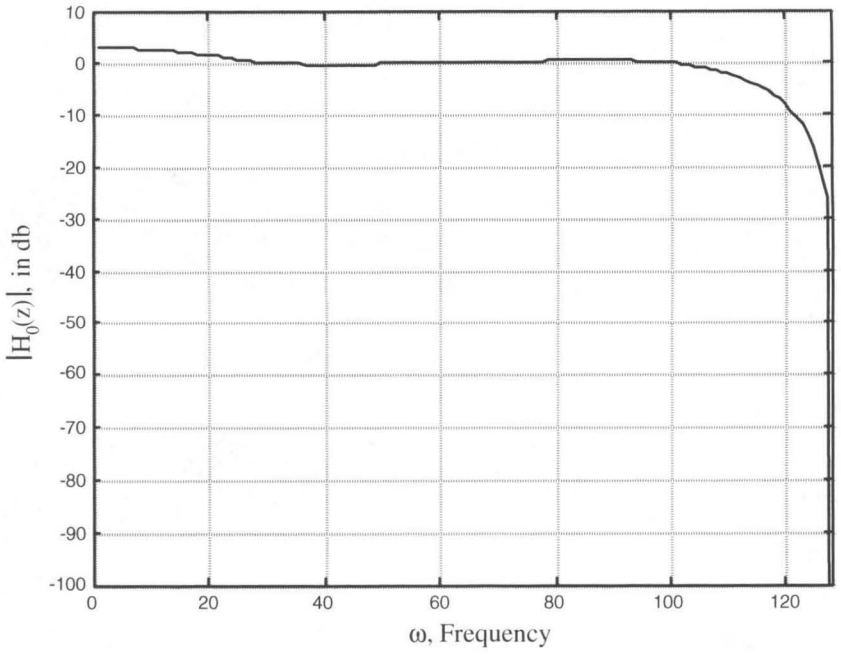



Figure 4.5 **Baseline QMF Frequency Response**

## 4.3 New Joint Optimization Design Criterion

To provide intelligent image compression and tailor it for object recognition we modify the QMF bank of the subband coder. The premise is that we can tune the subband filters to impact the information content of the subbands in a manner that is useful to object recognition. Toward that end, we introduce a new term which incorporates recognition performance into the design of the QMF; likewise we include the QMF response at every iteration in the design of the correlation filter. Figure 4.6 illustrates the design approach to jointly optimizing the performance of the two systems.

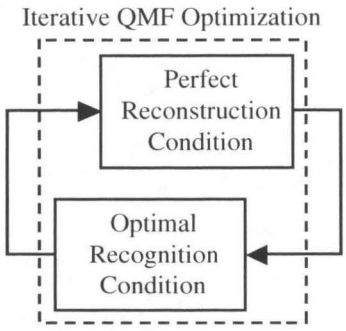



Figure 4.6 **Design Approach for Joint Compression Recognition Optimization**

Before visiting the QMF modification, let us explore a performance metric for the correlation design process. First, recall the definition of the MACH filter for one class of objects in Equation 2.6. By collecting the terms in the denominator, and using matrix vector notation, it can be rewritten as

$$\mathbf{c} = \mathbf{m}\mathbf{B}^{-1} \qquad (4.18)$$

where $\mathbf{m}$ is the average training image within one class of objects, and $\mathbf{B}$ is a spectral term defined by the denominator of Equation 2.6. Both terms are defined in the frequency domain. Recall that a different MACH filter is designed for each class, and that training is done with the reconstructed images. The latter technique integrates the subband transform into the recognition training process.

Now, from correlation filter theory [VKMJ00], we know that correlation filter design is a constrained optimization problem where the quadratic $\mathbf{c}^{\mathrm{T}}\mathbf{B}\mathbf{c}$ must be minimized subject to the linear constraint that $\mathbf{m}^{\mathrm{T}}\mathbf{c}$ achieve optimal peak values at the origin of the correlation surface. This is a Lagrange multiplier problem where the objective function that we wish to maximize is given as

$$J(\mathbf{c}) = \frac{\mathbf{c}^{\mathrm{T}}\mathbf{m}\mathbf{m}^{\mathrm{T}}\mathbf{c}}{\mathbf{c}^{\mathrm{T}}\mathbf{B}\mathbf{c}} \qquad (4.19)$$

where **c** is defined in Equation 4.18. We know that the MACH filter definition of Equation 4.18 is the optimal solution to Equation 4.19. With this substitution, Equation 4.19 can be rewritten as

$$\begin{aligned} J(\mathbf{c}) &= \frac{(\mathbf{B}^{-1}\mathbf{m})^{\mathrm{T}}\,\mathbf{m}\mathbf{m}^{\mathrm{T}}(\mathbf{B}^{-1}\mathbf{m})}{(\mathbf{B}^{-1}\mathbf{m})^{\mathrm{T}}\mathbf{B}(\mathbf{B}^{-1}\mathbf{m})} \\ &= \mathbf{m}^{\mathrm{T}}(\mathbf{B}^{-1}\mathbf{m}) \\ &= \mathbf{m}^{\mathrm{T}}\mathbf{c} \end{aligned} \qquad (4.20)$$

Because **m** and **c** are defined in the frequency domain, we take only the real part of the final quantity as our performance measure.

$$J(\mathbf{c}) = \mathrm{real}(\mathbf{m}^{\mathrm{T}}\mathbf{c}) \qquad (4.21)$$

As the Lagrangian objective function directly relates to the success of the correlation process, we use the quantity $J = \mathrm{real}(\mathbf{m}^{\mathrm{T}}\mathbf{c})$ to gauge the performance of the filter **c**. Yet, the quantity **m** (being the mean of all the reconstructed images) is directly dependent on the subband transform, and hence the QMF. Furthermore, **c** is dependent on **m**. Thus, we would like to optimize the QMF to maximize the value of *J*. Herein, lies the crux of the joint optimization process.

To jointly optimize the compression and recognition filter solutions, we add a fourth term to the baseline QMF design of Equation 4.17. Recall that the QMF design is an iterative optimization procedure which involves the minimization of several errors. Thus, we treat the inverse of the Lagrangian as an error term, provoking the QMF design to increase (for each object class) the *J* term of Equation 4.21 with every iteration.

$$\varepsilon_{\mathrm{rec}} = \sum_{k=1}^{K} \frac{1}{J_k}, \quad \text{where } J_k = \mathrm{real}\left(\mathbf{m}_k^{\mathrm{T}} * \mathbf{c}_k\right) \qquad (4.22)$$

Accordingly, the full-fledged joint optimization equation follows from Equation 4.17.

$$f = \varepsilon_{ald} + \varepsilon_{amd} + \varepsilon_{dc} + \varepsilon_{rec}$$

$$f = \sum_{i \neq j} \left[ \left( \mathbf{T}^{T} \mathbf{T} \right)_{ij} \right]^2 + \left( \mathbf{h}_0^{T} \mathbf{h}_0 - 1 \right)^2 + \left( \sum_{n=1}^{N_f} h_0(n) - \sqrt{2} \right)^2 + 17 \sum_{k=1}^{K} \frac{1}{J_k} \qquad (4.23)$$

We determined the constant applied to the new term empirically. The value of 17 achieved the highest PSNR value of the reconstructed image, while providing the maximum recognition performance. By reducing this constant, we can increase the PSNR measure, but only at the expense of recognition success.

**First Order Implementation**

The only downside of the above formulation is that a full implementation requires decomposition and reconstruction of every image in the training set with every iteration of the QMF design. This is due to the fact that the **B** term of the MACH filter equation (Equation 4.18) is dependent on second order image statistics. With a first order implementation, however, we alter only the **m** term of the MACH filter equation. This case requires that we decompose and reconstruct only the mean training image, **m**, with every QMF design iteration. We derived the **B** term from the original training data, with no reconstruction. The first order training method is much quicker than the second order technique described below. Selection of the training design technique depends on the amount of time and computer resources available for training.

**Second Order Implementation**

In this case, we decomposed and reconstructed every image in the training set with every iteration of the QMF design. This allowed us to construct both a new **m** and **B** term with each iteration. Again, our first design objective was to achieve the highest recognition performance, Pc, and secondly to produce the largest PSNR possible for that

Pc value. Hence, the fourth constant of Equation 4.23 was empirically determined to be 29, rather than 17.

**Results**

Our joint optimization QMF design (with a first order implementation) produced the time domain plot of $h_0(n)$ shown in Figure 4.7, with a QMF filter length of 16. We exhibit the magnitude of the frequency response of this filter, $H_0(z)$, in Figure 4.8. Comparing these plots with their baseline QMF counterparts (Figures 4.4 and 4.5) reveals a few differences between the baseline QMF and the jointly optimized QMF. The latter has a later and steeper rolloff, emphasizing more high frequencies, while attenuating the low frequencies earlier than the baseline QMF. This suggests that the jointly optimized

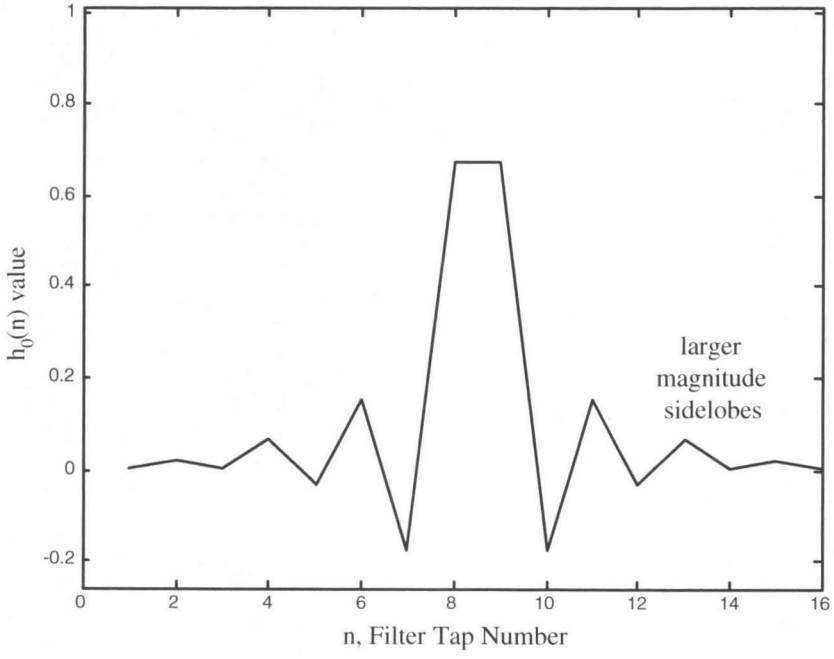



Figure 4.7 **Jointly Optimized QMF Time Domain Response**

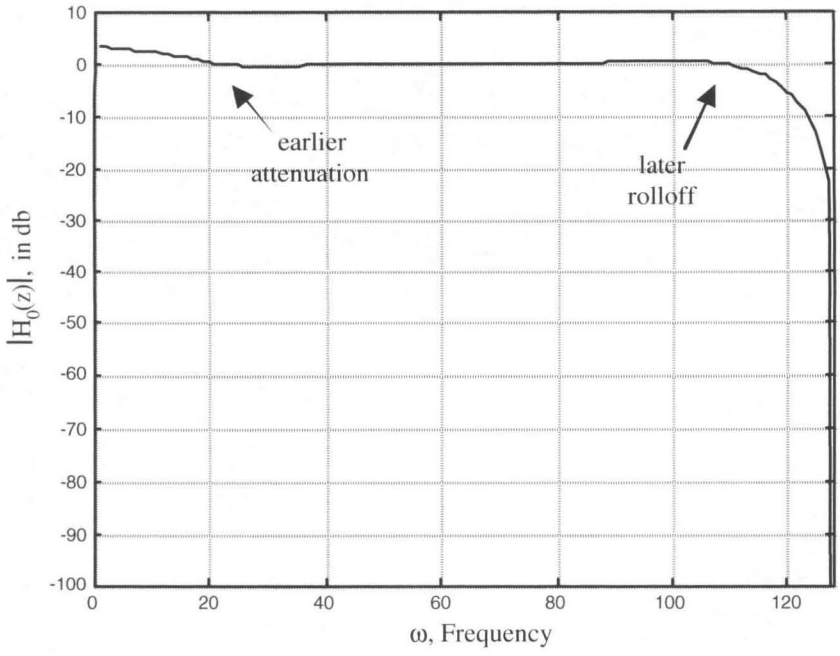



Figure 4.8 **Jointly Optimized QMF Frequency Response**

QMF tries to boost the capability of the recognition filter through intelligent frequency response. Observe also that the time domain response of the jointly optimized filter has magnified sidelobes.

**Discussion**

Finally, we should remark that if the system is given a fixed compression filter, the MACH filter can still be designed with the given subband reconstruction, although this is a non-iterative procedure. Likewise, a QMF can also be designed around a fixed MACH filter, which would use an iterative technique.

## 4.4 Compression System Design

A standard compression system is shown in Figure 2.3. In this section, we discuss the design of our quantizer and encoder for a simple compression scheme. In summary, we

use a uniform step size to scalar quantize the subband coefficients, and then assume the data will be transmitted by an entropy coder with variable length codes and a finite length codebook.

Throughout all of our experiments, quantization is applied only to the test data. The training phase utilizes reconstructed but unquantized image data. Prior to quantization, we optimally allocate the bits among the subbands with the well known expression to minimize distortion [Say96, SN96] by using the method of Lagrange. For a decomposition with $M$ equal subbands, the number of bits allocated to the $k^{th}$ subband is given by

$$R_k = R + \tfrac{1}{2}\log_2 \frac{\sigma_{y_k}^2}{\prod_{k=1}^{M}\left(\sigma_{y_k}^2\right)^{\frac{1}{M}}} \qquad (4.24)$$

where $R$ is the average number of bits per pixel to be used in the reconstructed image (a.k.a. the target bit rate), and $\sigma_{y_k}^2$ is the variance of the coefficients in the $k^{th}$ subband. Because we assume variable length codewords, $R_k$ can be a fractional value. Also, because our design is based on the set of training images, we use the maximum variance of each subband over the entire set of training images to define $R_k$.

We also assume that the encoder uses a finite length codebook so that the codewords can be stored apriori, i.e., transmission of codewords is not necessary. Thus, we define the number of quantization bins for each subband to be constant. Our design is based on the set of training images. That is, we construct an average probability distribution function for each subband from the training set of images to determine the minimum quantization step size possible for the bit allocation $R_k$.

Because we assume an entropy coder, we use entropy to determine the quantization step size. Entropy is computed by using the average probability distribution with the following equation [Sha48]

$$H = -\sum_{i=1}^{N} p_i \log_2 p_i \qquad (4.25)$$

where $N$ is the number of bins in the probability distribution. The probability distribution is quantized by a uniform quantization step size, $\Delta_q$. We begin with a very small quantization step size, $\Delta_q$, which is incremented until $H_k$ is less than or equal to the specified $R_k$.

The test images are then scalar quantized according to Equation 4.26, which holds for positive or negative coefficient values.

$$q = \left\{ \Delta_q \cdot \mathrm{rnd}\left( \frac{x}{\Delta_q} + 0.5 \right) \right\} - \left\{ \mathrm{sgn}(x) \cdot \frac{\Delta_q}{2} \right\} \qquad (4.26)$$

In a operative system, each quantized value would be transmitted as a single codeword. Again, the codebook would be stored apriori at both the encoder and decoder, and we assume a finite length codebook. If the training set is representative of the test images, then the number of quantization bins resulting from any test image should not exceed the number of codewords stored at the decoder. This was the case in all of our experiments. In practice, accommodations would have to be made for images which are exceptions.

Note that we did not actually construct a codebook for the experiments conducted in this thesis investigation. Once the quantization of the test images was complete, we computed the average entropy (Equation 4.25) over the entire test set. The actual bit rate never met the target bit rate exactly.

## 4.5 Performance

Finally, we are able to evaluate the system performance of the PRSC. The design techniques described in this chapter form the filters used in the PRSC of Chapter 3. In the first set of experiments we compare the performance of the PRSC to the results achieved with uncompressed image data (Section 2.6.1). Assessment of the PRSC is done with the baseline QMF and MACH filter, as well as the jointly optimized filter pair. Quantization is not applied to the subband coefficients during this first set of experiments.

The second set of experiments then characterizes the PRSC performance over increased quantization (i.e., decreasing bit rate). Both sets of experiments followed the conventions outlined in Section 2.5, but we review a few points here.

1) We use the training and test sets defined in Section 2.5.1.
2) We train with unquantized reconstructed image data.
3) We assume the images are preprocessed according to Section 2.5.2 prior to compression.

**Parameter Settings**

We use the following parameter settings for all experiments in this chapter:

- MACH filter design: $\alpha = 0.2$, $\beta = 0.8$, $\gamma = 0.05$ (Equation 2.6)
- Baseline QMF design: $N_f = 16$, where $N_f$ is the filter length (Equation 4.17)
- Recognition optimized QMF design: $N_f = 16$ (Equation 4.23)
- MACH filter operation: psr_thresh = 5 (Equation 2.8 and Section 2.5.3)

### 4.5.1 Joint Optimization Effects

We compare the recognition performance of four distinct systems in the first set of experiments.

Table 4.1 **System Specifications for PRSC System Performance Evaluation**

| | **Input Test Data** | **MACH Design Data** | **QMF** | **MACH Design** |
|---|---|---|---|---|
| **System for Uncompressed Data** | original | original | not applicable | standard |
| **PRSC - Baseline QMF/MACH** | subband transformed | reconstructed | baseline | standard |
| **PRSC - First Order Jointly Optimized QMF/ MACH** | subband transformed | reconstructed | jointly optimized w/ MACH | jointly optimized w/ QMF |
| **PRSC - Second Order Jointly Optimized QMF/ MACH** | subband transformed | reconstructed | jointly optimized w/ MACH | jointly optimized w/ QMF |

We compare the above systems for the following purposes.

- "*Can the baseline PRSC (which uses a transformed image) perform object recognition as well as using a correlation filter on the original image?*"
- "*What effect does joint optimization of the QMF and MACH filters have on the performance of the PRSC, in terms of both compression and recognition?*"

**System Performance**

A summary of the system performance metrics is shown below. Table 4.2 reports the object recognition measures Pc, Pe, and Pr, as well as the reconstruction metrics PSNR and MSE. Definitions of these measures appear in Section 2.5.4.

Table 4.2 **PRSC Performance Evaluation**
We show both object recognition and reconstruction metrics for the four systems outlined in Table 4.1. The PSNR and MSE were observed from one of the BMP test images.

| | Recognition Metrics | | | Reconstruction Metrics | |
|---|---|---|---|---|---|
| | **Pc** | **Pe** | **Pr** | **PSNR** | **MSE** |
| **System for Uncompressed Data** | 96.2% | 3.8% | 0% | not applicable | not applicable |
| **PRSC - Baseline QMF/MACH** | 96.2% | 3.8% | 0% | 105 db | $2\times10^{-6}$ |
| **PRSC - First Order Jointly Optimized QMF/ MACH** | 100% | 0% | 0% | 43.4 db | 2.97 |
| **PRSC - Second Order Jointly Optimized QMF/ MACH** | 100% | 0% | 0% | 46.5 db | 1.46 |

Two observations are in order:

- The baseline PRSC system does as well as the one operating on the original (i.e., uncompressed) image data.
- Jointly optimizing the QMF and MACH filters improves the recognition performance of the PRSC, although it reduces the reconstruction veracity of the compression transform.

**Correlation Performance**

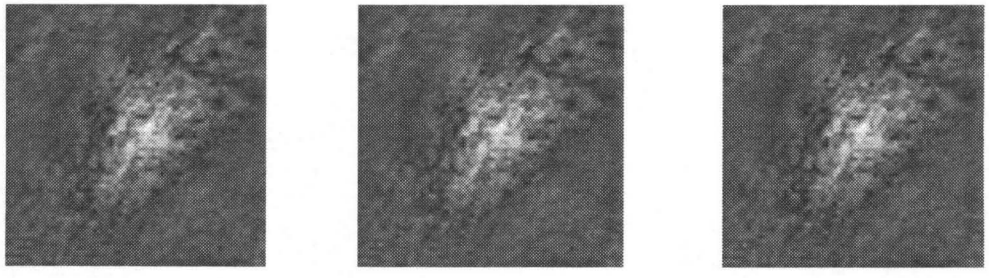
In Chapter 3, we stated two output characteristics of the baseline PRSC (i.e., without joint optimization of the QMF and MACH filters).

- It yields the identical response as correlation on the reconstructed image data.
- It is equivalent to the correlation response achieved by operating on the original image, limited only by the veracity of the subband transform.

To assess this claim, we examine the output correlation surfaces from the three appropriate systems.

Table 4.3 **System Specifications for PRSC Correlation Surface Evaluation**

| | **Input Test Data** | **MACH Design Data** | **QMF** | **MACH Design** |
|---|---|---|---|---|
| **System for Uncompressed Data** | original | original | not applicable | standard |
| **Baseline Compression System** | reconstructed | reconstructed | baseline | standard |
| **PRSC - Baseline QMF/MACH** | subband transformed | reconstructed | baseline | standard |



(a) Original Image Correlation Output

(b) Reconstructed Image Correlation Output

(c) PRSC Correlation Output

Figure 4.9 **Comparison of the PRSC Correlation Output**
In (a) we see the output of the System for Uncompressed Data, (b) the output of the Baseline Compression System, and in (c) the output of the baseline PRSC.

Visually, they all look very similar. For a better evaluation, we provide the MSE between the different correlation output surfaces.

Table 4.4 **Correlation Output MSE**
The correlation output from the baseline PRSC and the baseline compression system are identical, while the MSE between the baseline PRSC output and the original image correlation output is negligible. The latter value is less than the MSE of an image reconstructed with the baseline QMF, which is what the baseline PRSC uses as its subband transform.

| | |
|---|---|
| MSE between the PRSC and uncompressed output | $6\times 10^{-7}$ |
| MSE between the output of the PRSC and the baseline compression | 0 |

Table 4.4 verifies that our above claim is correct:

- the PRSC is equivalent to operating on the reconstructed image, and
- its output is identical to that attained from the original image, limited only by the precision of the subband transform.

Finally, we examine the improvement that joint optimization makes on the Lagrangian values ($J$ of Equation 4.21) of the MACH filter for each object class. The Lagrangian is measured with the training data.

Table 4.5 **Lagrangian Values for each System**
The jointly optimized systems yield much higher Lagrangians, as would be expected from the improved recognition performance of Table 4.2.

| | Lagrangian Values | | |
|---|---|---|---|
| | **BMP** | **BTR** | **T72** |
| **PRSC - Baseline QMF/MACH** | 279 | 324 | 316 |
| **PRSC - First Order Jointly Optimized QMF/ MACH** | 401 | 441 | 445 |
| **PRSC - Second Order Jointly Optimized QMF/ MACH** | 401 | 441 | 445 |

The jointly optimized systems provide significantly increased Lagrangian values for each object class, resulting in the improved performance seen in Table 4.2. Notice that using the second order technique of performing the joint optimization does not improve the Lagrangian values.

**Discussion**

The main goal of our compression recognition system is for content manipulation, not content visualization. Therefore, while we are interested in high compression rates, we are more concerned with tuning the compression model for object recognition rather than for visualization by a human observer. Accordingly, we are willing to sacrifice perfect reconstruction for recognition, as happens with the joint optimization of the filters. Moreover, the reconstruction of the image data is still quite good as exemplified in Figure 4.10.

(a)
Original Image

(b)
Reconstructed Image

Figure 4.10 **Image Reconstruction with our Jointly Optimized QMF**
The original image is shown in (a) while (b) exhibits the reconstructed version. The reconstructed image has a PSNR of 26.4 and an MSE of 149.

Finally, although the second order joint optimization design does slightly better, we selected the first order implementation for our next set of experiments. Selection of the training design technique depends on the amount of time and computer resources available for training. The first order technique is much quicker to implement.

**Conclusions**

We relate the main points found in this set of experiments here, to parallel the initial questions.

- "*The baseline PRSC (which uses transformed data) performs object recognition as well as using a correlation filter on the original image.*"
- "*Jointly optimizing the QMF and MACH filters improves the recognition performance of the PRSC beyond that of correlation on the original image, while reducing the faithfulness of its reconstruction ability.*"

Jointly optimizing the compression and recognition filters improves recognition performance via the following two 'simultaneous' mechanisms.

- It prompts the subband filter to attenuate frequencies which do not aid discrimination, while emphasizing frequencies that do.
- It allows the MACH filter to adapt to the frequencies present in the subbands.

Finally, we note that the second order (and slower) technique of implementing the joint optimization design does not improve the recognition performance of the PRSC, but does meliorate its reconstruction metrics.

### 4.5.2 Quantization Effects

In this set of experiments, we characterize the performance of the PRSC over increased quantization (i.e., decreasing bit rate). That is, "*How will compression degrade*

*the recognition performance of the PRSC?*" Section 4.4 describes the quantizer design used in these experiments.

**Results**

The following data summarizes the performance results of the PRSC with the joint optimization technique (as a first order implementation) in place. We express the object recognition measures Pc, Pe, and Pr as a function of bit rate, in Figure 4.11, 4.12, and 4.13, respectively. In Figure 4.11, we augment the PRSC performance with results from the baseline compression recognition system (full image reconstruction prior to correlation) for comparison, as well as the performance of a correlation system operating on the original (uncompressed) data. The latter is shown with only one data point, at eight bpp. Reduced bit rates are not applicable to the uncompressed data.

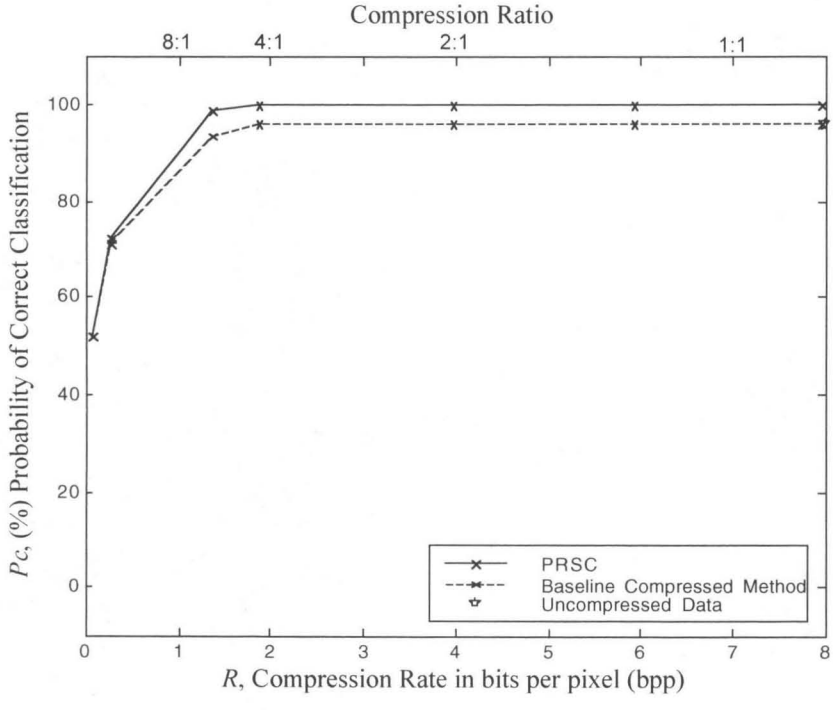



Figure 4.11 **Probability of Correct Classification vs. Bit Rate**

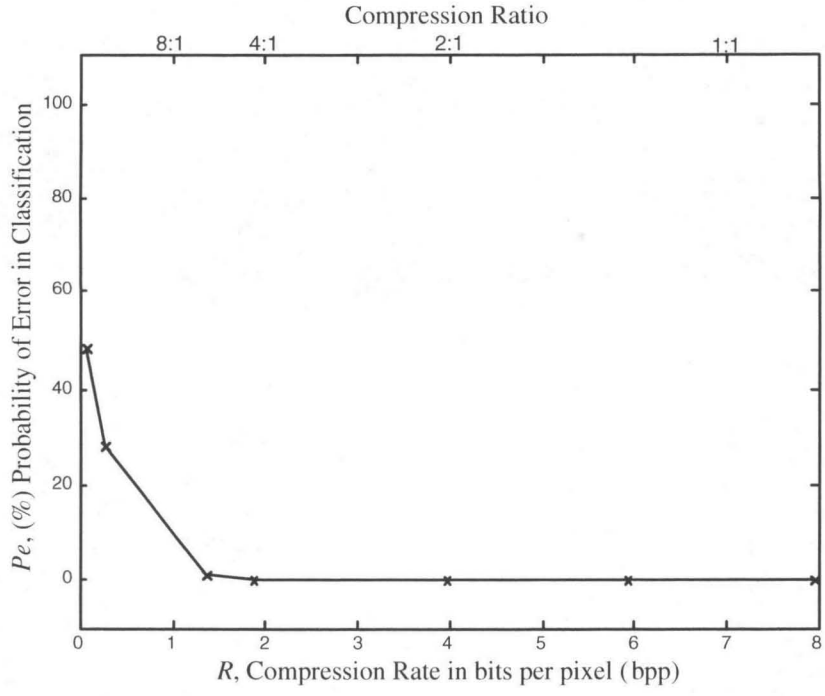



Figure 4.12 **Probability of Error in Classification vs. Bit Rate**

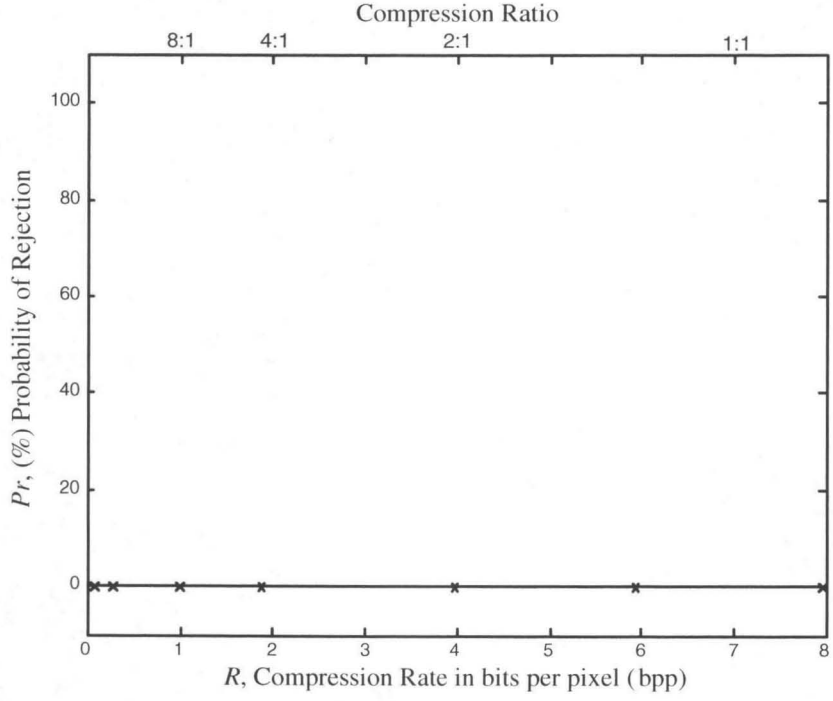



Figure 4.13 **Probability of Rejection vs. Bit Rate**

In this experiment, we used the jointly optimized QMF and MACH filter with the PRSC, whereas the baseline compression recognition system possessed the baseline QMF. In this case, the QMF and MACH filter were designed independently, with the MACH filter designed with the reconstructed but unquantized training data. The reconstruction metrics PSNR and MSE for the jointly optimized QMF are also on display in Figure 4.14 and 4.15, respectively. For completion, Table 4.6 gathers all of the results from Figures 4.11 - 4.15.

Observe that the Rate Error curve of Figure 4.12 resembles a Rate Distortion curve, while all three curves in Figures 4.11 - 4.13 possess similar characteristics. That is, they maintain good performance down to very low bit rates and then drop off quickly. We see a slight drop off in performance at 1.33 bpp, while a more significant drop off occurs at 0.28 bpp. The recognition performance curves (Pc, Pe, and Pr) allow tuning between the two ends of compression and recognition. An operating point can be selected depending on the application. The reconstruction metrics PSNR and MSE were computed from one of the BMP test images and are exhibited for the jointly optimized QMF in the above figures.

**Discussion**

Three primary observations are in order. First, the PRSC does very well. In fact, it always performs better than the baseline system, and with *one-fourth* of the information, the PRSC performs better than correlation on the original image data. This is primarily due to the joint optimization of the compression and recognition filters. The joint optimization tunes the MACH filter over all the frequency bands present in the subband coder, and vice-versa, it adapts the subband information in the best interests of the

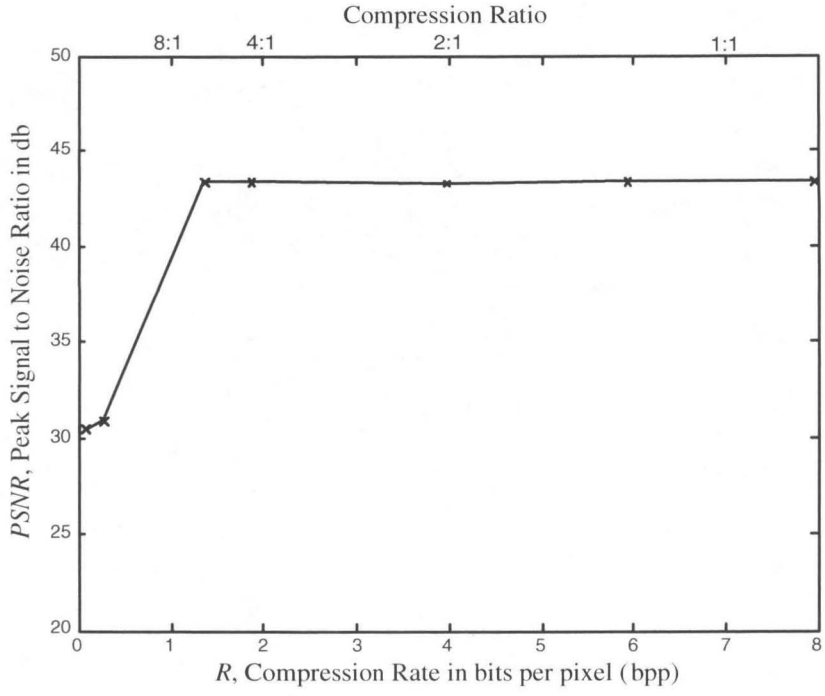



Figure 4.14 **Peak Signal to Noise Ratio vs. Bit Rate**

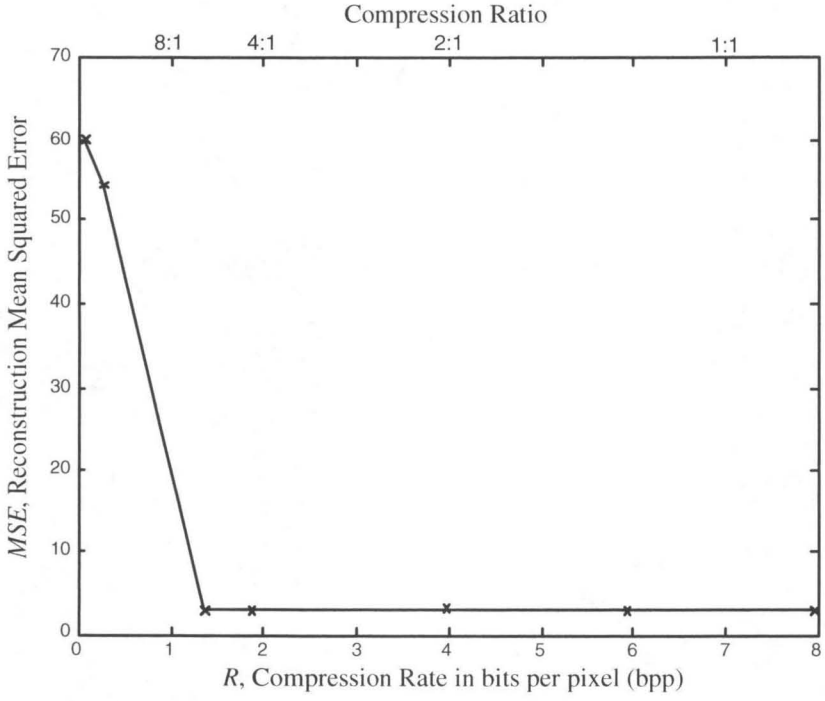



Figure 4.15 **Reconstructed Image MSE vs. Bit Rate**

Table 4.6 **PRSC Results as a Function of Bit Rate**

| Compression Rate | | Recognition Metrics | | | Reconstruction Metrics | |
|---|---|---|---|---|---|---|
| **Target Rate** | **Actual Rate** | **Pc** | **Pe** | **Pr** | **PSNR** | **MSE** |
| 8 bpp | 7.96 bpp | 100% | 0% | 0% | 43.4 db | 2.97 |
| 6 bpp | 5.94 bpp | 100% | 0% | 0% | 43.4 db | 2.97 |
| 4 bpp | 3.98 bpp | 100% | 0% | 0% | 43.3 db | 2.97 |
| 2 bpp | 1.87 bpp | 100% | 0% | 0% | 43.3 db | 2.97 |
| 1 bpp | 1.33 bpp | 98.7% | 1.3% | 0% | 43.3 db | 2.97 |
| 0.5 bpp | 0.28 bpp | 72.2% | 27.8% | 0% | 30.8 db | 54.3 |
| 0.25 bpp | 0.07 bpp | 51.9% | 48.1% | 0% | 30.4 db | 59.9 |

MACH filter.

Second, we note that these results are not quite as good as in [WM99] which report recognition performance of over 90% for compression ratios up to 200:1, as shown in Table 2.1. In this previous work, they fully reconstruct the image from a quantized subband decomposition, and perform recognition on the full resolution (albeit reconstructed) image. The most obvious reason for our reduced performance is due to the simplicity of our encoding scheme.

At low bit rates, our encoding scheme begins to drop subbands. That is, at low bit rates, the quantizer does not have enough bits to effectively code all of the subbands on all levels. Hence, some of them are encoded with zero bits (effectively 'dropping' these subbands). The table below displays the bit rates which require empty subbands. As a result, the correlation output of the PRSC contains aliasing at compression ratios below 6:1 (corresponding with bit rates below 1.33 bpp). The aliasing in the correlation surface prohibits successful recognition.

Table 4.7 **Subbands Dropped at Low Bit Rates**

| Actual Rate | Subbands Dropped |
|---|---|
| 1.01 bpp | on level 1: HL, LH, HH |
| 0.28 bpp | on level 1: HL, LH, HH<br>on level 2: HH<br>on level 3: HH |
| 0.07 bpp | on level 1: HL, LH, HH<br>on level 2: HL, LH, HH<br>on level 3: HH |

In [WM99], they employ a gainful coding technique, the Embedded Zerotree Wavelet coder (EZW) [Sha93], which helps them achieve such success. In a nutshell, the EZW coder progressively transmits the subband coefficients in order of their magnitude regardless of their subband affiliation. The EZW is well known to produce some of the best compression rates currently available for image data.

By employing a superior encoding technique the PRSC could improve its already successful performance. It is really a matter of the MSE at the output. Notice that the reconstruction MSE increases from 9.6 to 39.4 between the two and one bpp rate in Figure 4.15. A better encoding technique could certainly decrease this MSE for even high compression ratios (low bit rates).

Lastly, we must remark on the training data. These experiments used the reconstructed images as training data, and the reconstruction was 'lossless' in that it had no quantization. The test data, however, was quantized. Performance should improve by training on the quantized image data. This remains an avenue for future study.

**Conclusions**

We review the effects of quantization on PRSC performance below.

- For compression ratios as high as 6:1, the PRSC achieves recognition performance superior to that of correlation on the original image.
- Over all compression ratios, the PRSC performs equally or better than the baseline compression recognition system.
- All of the recognition performance curves (Pc, Pe, and Pr) allow a user to select an operating point and tune the system between the two opposing ends of compression rate and recognition accuracy.
- The PRSC would perform more successfully at higher compression ratios with a more sophisticated encoding scheme like the EZW coder.
- The PRSC may also perform more successfully if the training data was quantized at the same rate as the test data.

## 4.6 Future Extensions

We briefly touch on two possible areas of improvement in the future. The first suggestion involves the optimization of the MACH filter. We discuss the joint optimization of the QMF and MACH filter in Section 4.3. In the current implementation, we maximize the Lagrangian of the filter for the object class under consideration, i.e. the in-class Lagrangian. It may further improve performance to minimize an out-of-class Lagrangian term; such a term would be included in the equation for joint optimization, Equation 4.23.

Secondly, note that our current training methodology uses reconstructed, but unquantized data. By including quantization in the training loop, the system could achieve optimal quantization parameters, for an 'ultimate' optimization.

## 4.7 Summary

In the past, image compression and object recognition have been treated as separate problems. For our purpose, we have shown that it is better to jointly optimizing the performance of the two systems, rather than optimize each component individually. The main purpose of our compression recognition system is for automated object recognition, not human visualization. Therefore, we are interested primarily in recognition accuracy at high compression rates, and secondarily, in the reconstruction fidelity of the compression transform.

To facilitate the design of our compression filter, we introduce a matrix formulation of a multiple level subband transform in the time domain. The inner product of this matrix embodies all the requirements of a perfect reconstruction QMF design. We then augment this design to jointly optimize the performance of the QMF and MACH filters. This joint design provokes the QMF to adjust the subband frequency content to aid object discrimination, while simultaneously, it allows the MACH filter to adapt to the frequencies present in the subbands.

The systems investigated in this chapter enjoy the following characteristics, which in turn respond to three of the questions extended at the outset of the thesis.

- The *baseline* PRSC uses a transformed image to perform object recognition at rates *equal* to that achieved with a correlation filter on the *original* image.
- *Jointly optimizing* the QMF and MACH filters provides the PRSC with recognition accuracy *superior* to that of correlation on the *original* image, for *compression ratios as high as 6:1*.

- Moreover, *joint optimization* of the QMF and MACH filters provides the PRSC with recognition accuracy *equal to or better* than that with their *separate* implementation *at all bit rates*.
- The *rate recognition curves* provided by the PRSC allow a user to select an *operating point* and *tune* the system between the two opposing objectives of compression rate and recognition accuracy.
- At compression ratios higher than 6:1, the PRSC can perform more successfully than shown in this investigation by using a more sophisticated encoding scheme like the EZW coder.

# Chapter 5

# Subband Domain Correlation Filters

To complement the Pattern Recognition Subband Coder, we introduce Subband Domain Correlation Filters (SDCFs) as a second and sister solution to the problem of object recognition in compressed imagery. As we show in Chapters 3 and 4, the PRSC provides suitable full resolution object recognition from compressed data, but does not conveniently benefit multiresolution analysis. When working with compressed images, however, the objective is often to do a fast cursory search across the data, even if that entails a reduction in accuracy. Therefore, this chapter addresses the parallel questions, *"Can we perform object recognition at lower resolutions with reasonable accuracy?*" and "*How does the accuracy degrade with decreasing bit rate?*" The approach manifested in SDCFs is twofold: 1) We perform both the training and operating phases of object recognition directly in the subband coefficient domain, and 2) We combine the multiple subband responses with Polynomial Correlation Filters (PCFs), a powerful architecture introduced by Mahalanobis and Vijaya Kumar [MVK97a] for fusing multiple inputs into a single correlation output.

We present the concept of coefficient domain processing in Section 5.1. In addition, this first section discusses the downsampling issue inherent in coefficient domain processing and suggests two methods to mitigate the effect. Section 5.2 then reviews PCF theory and introduces its function in the fusion of subband information. Next, we examine the two methods proposed in Section 5.1: a training technique and a QMF design technique, in Sections 5.3 and 5.4, respectively. Section 5.5 reports the performance of the SDCF system along with the effects of decreasing bit rates. Equally important is the computational complexity of the SDCFs, which we review in Section 5.6. For future modifications, Section 5.7 proposes possible extensions to the SDCF algorithm presented here. Finally, Section 5.8 concludes the chapter with a review of the SDCF algorithm, its performance, benefits, and limitations.

## 5.1 Coefficient Domain Processing

Performing a quick search of images in the compressed space suggests that we exploit the multiple resolutions available in a subband decomposition. Toward this end, we pursue the intuitive idea of synthesizing a correlation MACH filter from each of the individual subbands in a subband decomposition, and then processing each subband with its corresponding filter to achieve a pattern match for object classification. We call this approach *coefficient domain processing*. Aside from the issue of fusing the multiple outputs, a larger issue impacts this approach: variations in the subband patterns due to the downsampling properties of subband coders. *Coefficient domain processing* contrasts with the *image domain processing* technique used in the Pattern Recognition Subband Coder of Chapters 3 and 4.

### 5.1.1 Downsampling Effects

Subband coders downsample by two between levels of decomposition, as discussed in Section 2.4. This operation produces different subband responses to single pixel shifted copies of the original input image. Herein lies the crux of the issue. For example, assume an object at some reference point in the original image has produced a one level subband decomposition. (We call this the reference decomposition.) Shifting the original image by one pixel in any direction will result in a similar yet distinctly different pattern in a second subband decomposition. This is because after applying the lowpass and highpass filters to the original image, only every other pixel remains in the image representation at level one. The issue is more of a problem on the upper subbands because of the high frequency nature of the data, however, it is still present in the LL subband (defined on p. 28). We illustrate the issue for the LL subband in Figure 5.1.

Now from the reference point, a two pixel shift of the original image in any direction will produce the exact same decomposition pattern (ignoring boundary effects) at level one as the reference decomposition. However, the output will be translated by one pixel. This type of displacement is not an issue for correlation filters. As noted in Section 2.3, they are well suited for translations of the pattern they are trying to match. Correlation filters are well known their *shift invariant* properties [VKMJ00]. They are not, however, *downsampling invariant*, nor are they very downsampling tolerant.

Furthermore, the downsampling issues are present and multiply on every level of the decomposition. Although the first level has only four possible patterns, the second level will produce four variations of each of them. Recursive downsampling leads to a tetradic tree of pattern variations.

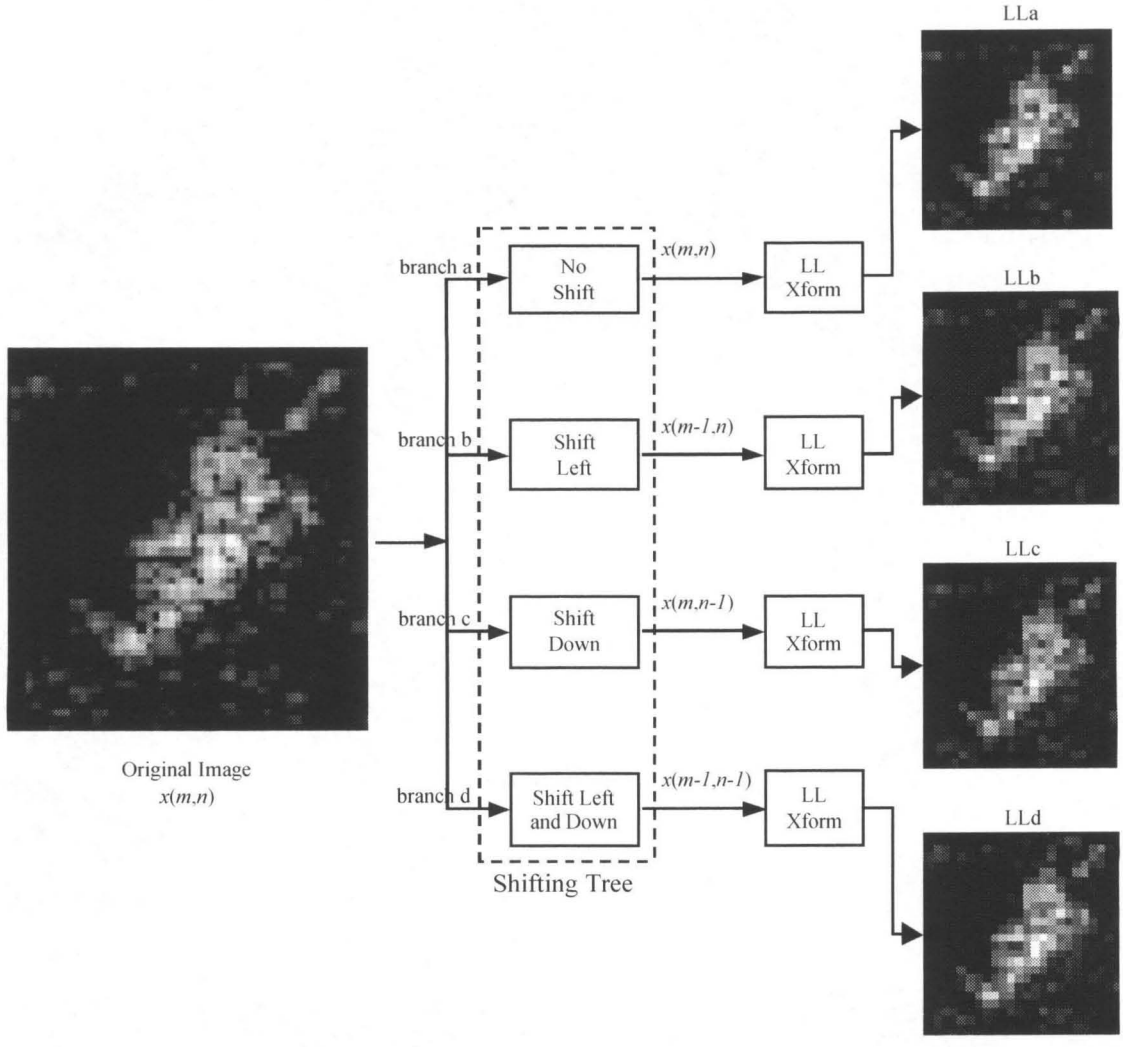



Figure 5.1 **Downsampling Effect on the LL Subband Patterns at Level One** The subband downsampling process is not shift invariant. We illustrate the issue for the LL subband only, although the upper band coefficients are most affected. Note that the four LL images shown are *not* simply shifted versions of each other. Rather, they result from shifts of the *original* image. On close inspection, one can see that the LL subband of each branch exhibits a different pattern. Furthermore, the problem explodes at the lower decomposition levels, as each shifted version on level $n$ spawns four shift variations on level $n$-1.

Moreover, when processed with identical correlation filters, similar yet distinctly different patterns produce dissimilar correlation surfaces. The existence of differing patterns for the same object consequently undermines the performance of the object recognizer.

### 5.1.2 Providing Downsampling Tolerance

To mitigate the downsampling effects of a subband coder, we propose two complementary solutions. The first requires synthesizing the correlation filters over multiple shifts of the training data. The second modifies the QMF properties to account for correlation between neighboring pixels in the training data. We designed the two solutions to work in concert with one another; the first was developed with the highpass information in mind, while the second was constructed primarily for the lowpass information. SDCFs may use these two solutions separately or together to substantiate *downsampling tolerance*.

## 5.2 Subband Fusion with Polynomial Correlation Filters

While many of the fusion techniques residing in the information processing literature may be applicable, we selected PCFs because they are a direct extension of standard correlation filters. A detailed description of PCFs is given in [MVK97a], where it was first suggested to use PCFs with a multiresolution data. PCFs embody a multiple input single output system. Given multiple inputs, a PCF system calls for a like number of correlation filters, each corresponding to one input. Yet, the system produces a single integrated correlation surface output.

In our application, the number of inputs is equal to the number of subbands in the pyramid decomposition. The architecture diagrammed below illustrates our use of PCFs with a one level subband decomposition. Maintaining a simple approach for the proof of principle availed in this thesis, we incorporate only the subbands from a single level in our experiments. In future work, however, a system may integrate any subset of a subband decomposition, as discussed in Section 5.7.

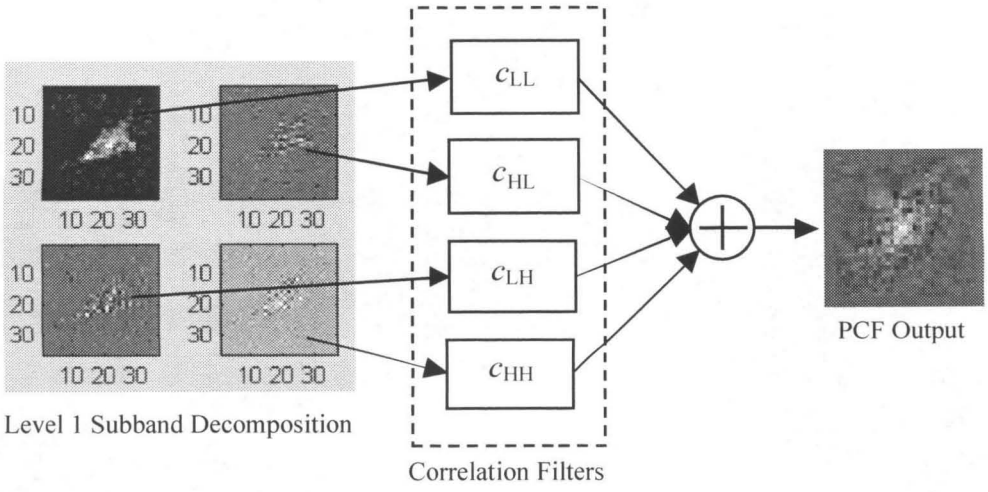



Figure 5.2 **Subband Domain Correlation Filter Architecture**
The PCF provides a way to simultaneously correlate all subbands. Each subband takes its own input channel, and thus its own polynomial correlation filter. We use an analogous structure for each level of the subband decomposition.

Mathematically the output of a general PCF is expressed as

$$y(m,n) = \sum_{p=1}^{P} c_p(m,n) \otimes x_p(m,n) \qquad (5.1)$$

where $P$ is the number of input channels, $x_p(m,n)$ is the data present at the input of each channel and $c_p(m,n)$ is the filter function for each channel. These filters are formulated to jointly maximize the peak value of $y(m,n)$ while minimizing the effects of distortions and noise. Equation 5.1 gives the appearance that PCF theory augments standard correlation by only a simple summation. The real power behind PCF theory, however, lies in the simultaneous optimization of the filters for multiple channels. We now provide a brief review of the PCF theory and the solution for the multichannel filters. As in Section 2.3, all variables can be written in the frequency domain. In that case, $C_p(k,l) = \Im\{c_p(m,n)\}$, $X_p(k,l) = \Im\{x_p(m,n)\}$. $\Im\{\cdot\}$ and $\Im^{-1}\{\cdot\}$ represent the forward and inverse discrete Fourier Transform operation, respectively.

For a set of $N$ training images, the average training image for the $p^{th}$ channel is defined by

$$M_p(k,l) = \frac{1}{N}\sum_{i=1}^{N} X_p^i(k,l) \qquad (5.2)$$

where $X_p^i(k,l)$ represents the $i^{th}$ training image of the $p^{th}$ input channel. We define the cross-power spectrum of the training set for the $p^{th}$ and $q^{th}$ input channels as

$$D_{pq}(k,l) = \frac{1}{N}\sum_{i=1}^{N} X_p^i(k,l) \cdot X_q^{i^*}(k,l) \qquad (5.3)$$

and a spectral variance term as

$$S_{pq}(k,l) = D_{pq}(k,l) - M_p(k,l) \cdot M_q^*(k,l) \qquad (5.4)$$

Using these quantities, we define the cross-channel spectral statistics,

$$B_{pq}(k,l) = \alpha S_{pq}(k,l) + \beta D_{pq}(k,l) + \gamma \mu_{pq} \qquad (5.5)$$

The term $\mu_{pq}$ in Equation 5.5 is simply the mean value of the spectral variance term, $S_{pq}$.

As in Equation 2.6, the constants $\alpha$, $\beta$, and $\gamma$ are used for optimally trading between distortion tolerance, noise tolerance and correlation energy minimization. Central to the strength of PCF theory is the fact that each filter is a conglomerate influenced by the spectral content of all the input channels. Specifically, each filter is influenced by the power spectral density of its own inputs as well as the cross-spectral density between its input channel and all the others. The term $B_{pq}(k,l)$ of Equation 5.5 is the collective spectral terms used in the derivation of the MACH filter (Equation 2.3 - 2.6), with modifications made to reflect multiple input channels. Consequently, the PCF filter solution analytically optimizes the same performance criterion as the MACH filter, but with respect to the cross-channel statistics.

Using matrix vector notation, we denote the cross-spectral statistics in Equation 5.5 by a block diagonal matrix $\mathbf{B}_{pq}$. Further let $\mathbf{m}_p$ be the mean image of the $p^{th}$ channel also expressed as a vector. The two channel PCF solution is then given by

$$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix} \qquad (5.6)$$

where $\mathbf{c}_p$ is the optimum correlation filter vector for the $p^{th}$ input channel. Thus the PCF amalgamates information from all input channels to obtain the best set of correlation filters that simultaneously process all inputs and combine them into a single result. Essential to the filter synthesis is that the system estimate the $\mathbf{m}_p$ and $\mathbf{B}_{pq}$ terms over a large number of inputs representative of the objects under study (i.e., a training set). Certainly, Equation 5.2 is easily extendable to $N$ inputs channels. Detailed information regarding the solution to the inverse of the generalized form of the matrix in Equation 5.6 is provided in [MVK97a].

## 5.3 Incorporating Downsampling Tolerance through Training

As elucidated in Section 5.1.1, single pixel variations in the placement of the original image produce differing subband images (excluding edge effects). The algorithm described here frames the following question, "*Is it possible for a correlation filter to accommodate these variations by training over multiple shifts of each training image?*"

**Partial Tetradic Tree - Our Method**

In reference to a one level decomposition, the training algorithm we use incorporates each input image four times: once in its original state, once after the image undergoes a single pixel shift up, again after we translate the image by one pixel to the left, and yet again after we shift the image up and to the left one pixel in each direction. We utilize a

circular shift technique so that the first column of $x(m,n)$ becomes the last column of the shifted image. Figure 5.3 depicts such a procedure.

We then employ this procedure at each level. That is, at level two, the LL subband from level one acts as the input image; it is shifted the requisite number of times, and the four subband images on level two make up the PCF channel inputs. All experiments in this thesis employ the training tree structure of Figure 5.3. Section 5.5 reports the performance results achieved by this methodology.

**Full Tetradic Tree**

The above procedure is but a subset of the set of shifts necessary to cover all the downsampling variations. To cover all possibilities, level two would have to incorporate four shifts for each of the four shifts in level one, and so on at each level, spawning a tetradic tree of training images, as diagrammed in Figure 5.4.

**Minimal Tetradic Tree**

Although downsampling induced variations are more apparent in the upper subbands due to their high frequency content, they still exist in the LL subband. Because the subband changes are slight in the LL subband, however, it may be possible to eliminate the training tree structure and shift only the three upper subbands on each level during the training phase. This would greatly simplify and hasten the training process. If this idea is successful, then the HL subband should only have to be shifted horizontally, the LH subband vertically, while the HH subband would still need a shift both horizontally, vertically, and diagonally. That is, the shifts required for training may only be necessary in the direction of the highpass filtering operation, as shown in Figure 5.5. Note that the training tree structure has become linear.

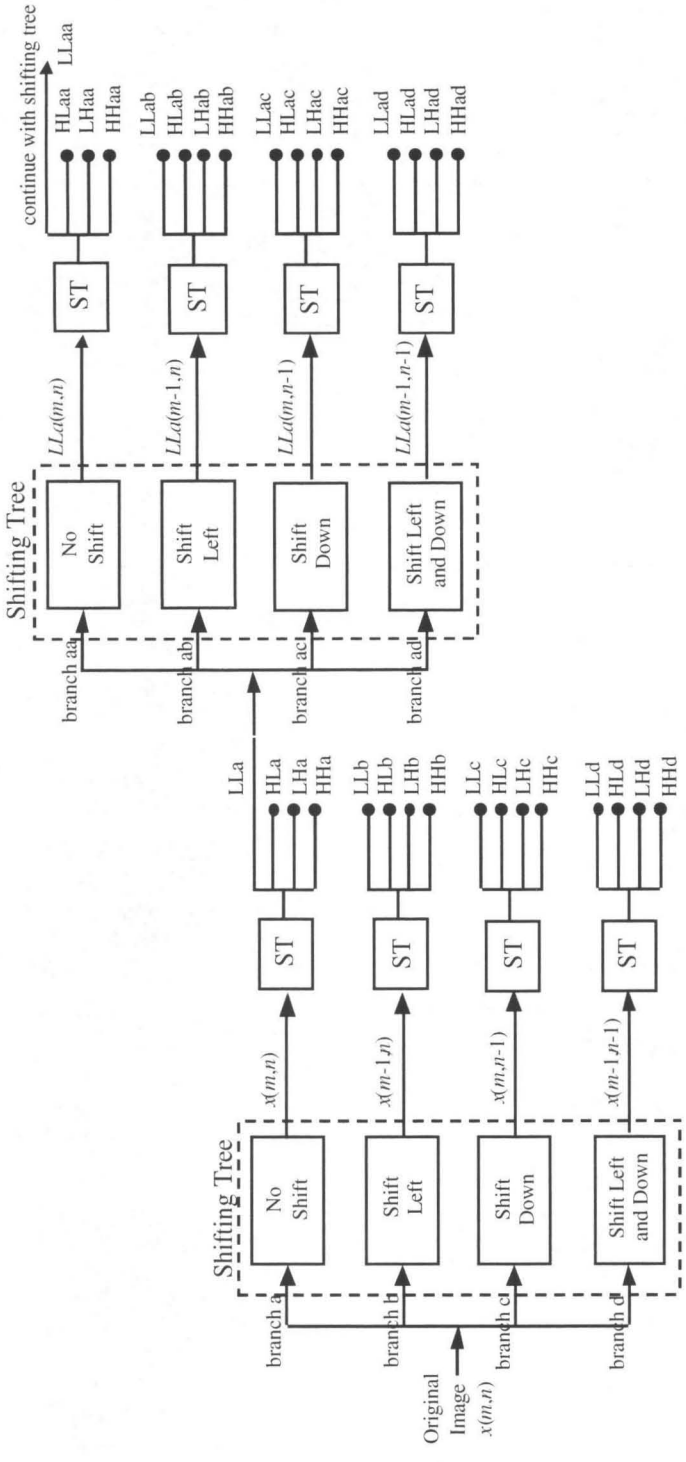



Figure 5.3 **Partial Tetradic Training Tree for Training over Image Shifts** Only the LL subband of the original placement spawns four shifting branches. Thus, there are $4N$ training images per subband of each level, given $N$ images in the original training set. This is the training method we employ in this thesis.

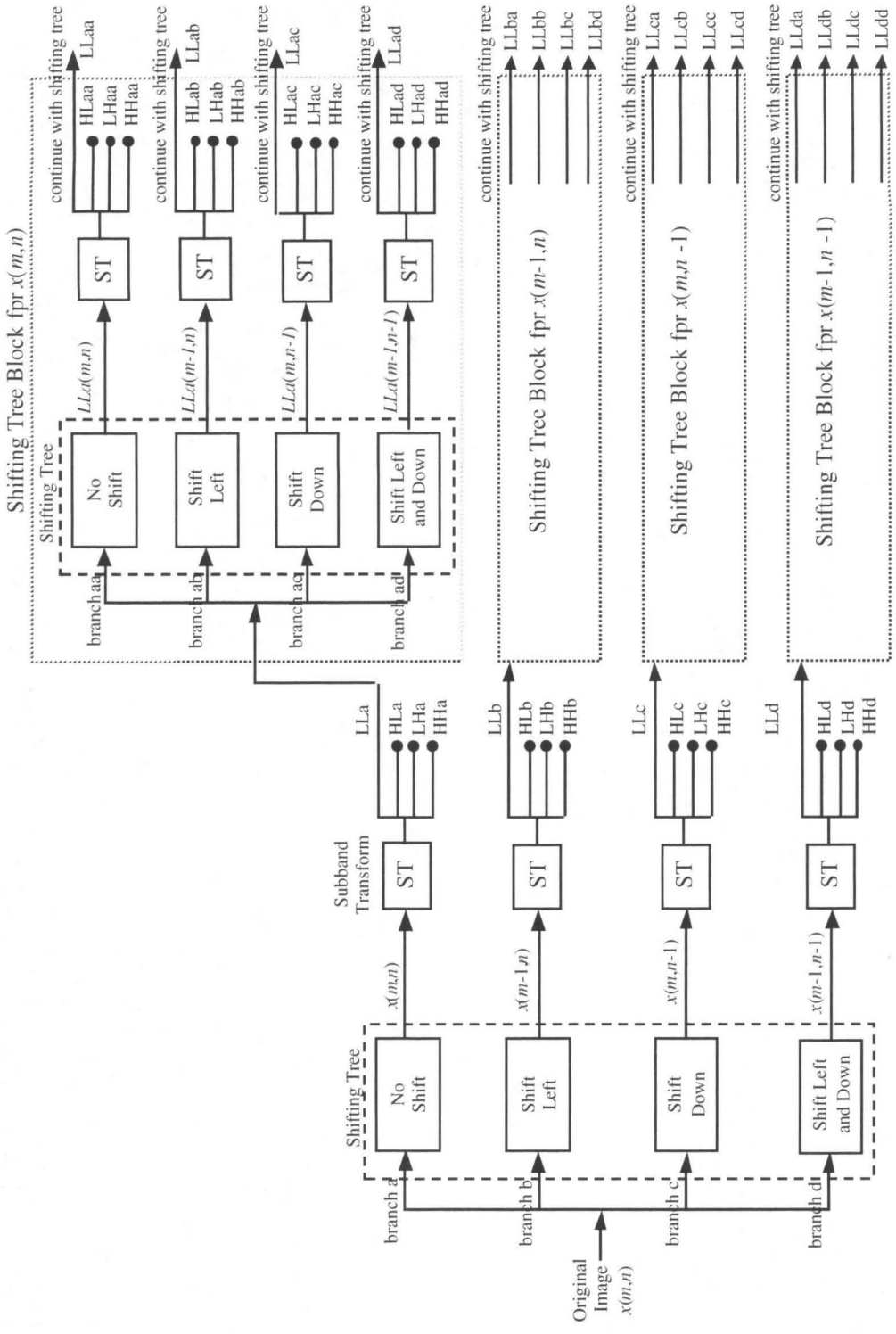



Figure 5.4 **Full Tetradic Training Tree for Training over Image Shifts** The LL subband of all image shifts spawns four shifting branches. Thus, there are $(4^k)N$ training images per subband on level $k$, given $N$ original training images.

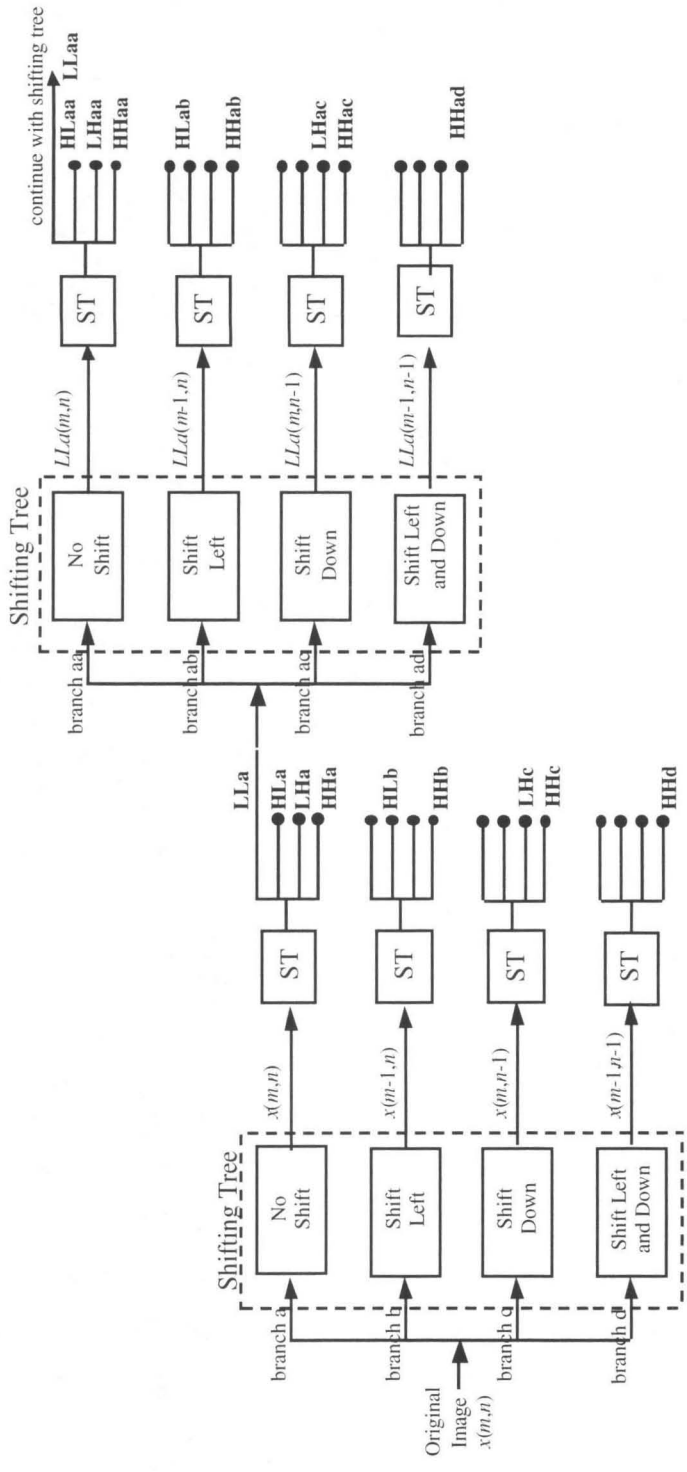



Figure 5.5 **Minimal Tetradic Training Tree for Training over Image Shifts** Only the LL subband of the original placement spawns four shifting branches. Also, only one shift required by LL, two by HL and LH, and four by HH, as shown in bold. Results in $N$, $2N$, $2N$, and $4N$ training images, respectively.

To reduce the downsampling variations within the LL subband, we consider a modification of our baseline QMF. The new QMF is primarily intended to benefit the low frequencies, whereas the training methodology introduced above is most effective over the high frequency paths. Together they provide a complete solution.

## 5.4 Incorporating Downsampling Tolerance through QMF Design

To avoid an exploding tree structure for training, we propose to modify the QMF bank of the subband coder as a second and complementary method of instilling downsampling tolerance in the system. If the LL subband response is sufficiently robust, the tree used for shift training becomes linear, as discussed in Section 5.3. This idea gives rise to the following question, "*Can we tune the QMF used for image compression to compensate for the downsampling within the subband coder?*"

By its very nature, the QMF compacts the energy in the upper subbands, forming a sparse representation. Coupled with downsampling between levels this sparseness impairs object recognition, as detailed in Section 5.1. In contrast, if one could stabilize the QMF response between adjacent coefficients, then the system could better tolerate the downsampling effects. With this objective in mind, we added another term, the downsampling tolerance criterion, to the standard QMF design (Equation 4.17). The QMF and our approach to its design is discussed in detail in Section 4.2.

Too much dispersion, however, counteracts the compression capability of the subband coder. Energy compaction is particularly critical in the upper subbands. Therefore, we tune the system to compensate for downsampling over only the low frequency channels of the subband decomposition. Even with this limitation,

compression can be compromised. Thus, to provide balance, the downsampling tolerance constraint is a composite of two terms: a shift sensitivity term and an energy compaction term.

### 5.4.1 Equivocating the QMF Response to Input Shifts

Physically, the shift sensitivity term refers to the difference of the QMF response to an image and its single pixel diagonally shifted copy. We need to make the two responses as similar as possible to compensate for the downsampling of the subband coder in both directions (vertical and horizontal). Let $h_h(m, n)$ and $h_v(m, n)$ (or equivalently $\mathbf{h}_h$ and $\mathbf{h}_v$ in vector notation) represent the subband (QMF) filters used in the horizontal and vertical directions, respectively. The subband response to an image, $x(m, n)$, (prior to downsampling) is then given by

$$y(m,n) = h_h(m,n) * x(m,n) * h_v(m,n) \tag{5.7}$$

Thus, we need to minimize the amplitude of the difference between the response, $y(m, n)$, and its shifted counterpart, i.e.,

$$\begin{aligned} \Delta &= E\{y(m,n) - y(m-1,n-1)\} \\ &= E\{h_h(m,n) * [x(m,n) - x(m-1,n-1)] * h_v(m,n)\} \end{aligned} \tag{5.8}$$

Here $E\{\cdot\}$ refers to the expectation operation. It may also behoove the system to minimize the energy of the filtered difference image, i.e.,

$$\Delta^2 = E\{|y(m,n) - y(m-1,n-1)|^2\} \tag{5.9}$$

Also, we utilize a circular shift technique so that the first column of $x(i,j)$ becomes the last column of the shifted image.

We now define the autocorrelation matrix.

$$\mathbf{R}_x = E\left\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\right\} \tag{5.10}$$

Some clarifications are in order regarding the assumptions made in estimating the autocorrelation matrix. The matrix $\mathbf{R}_x$ has Toeplitz structure and is obtained by windowing the autocorrelation function of $x(m,n)$ to the same length as the subband filters. We further assume that the rows and columns have separable and independent autocorrelation functions so that $\mathbf{R}_x$ can be estimated along each dimension separately and averaged. Admittedly, there are several alternatives to modeling $\mathbf{R}_x$; we use the stated model for the sake of simplicity. It follows that the autocorrelation matrix of the difference image,

$$d(m,n) = x(m,n) - x(m-1,n-1) \quad \forall\ (m,n) \tag{5.11}$$

is given by

$$\mathbf{R}_d = E\left\{\mathbf{d}\mathbf{d}^{\mathrm{T}}\right\} \tag{5.12}$$

It can easily be shown that the matrix $\mathbf{R}_d$ is the difference of $\mathbf{R}_x$ and its diagonally shifted version (i.e., all matrix elements moved down by one row and to the right by one column). Now, by rewriting Equation 5.8, the sum of the amplitudes of the pixels in the filtered difference image is given by

$$\Delta = \mathbf{h}_{\mathrm{h}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}\mathbf{h}_{\mathrm{v}} \tag{5.13}$$

We can now define the shift sensitivity measure in the LL subband as a filtering operation. The LL subband requires a lowpass filter in both directions.

$$\Phi = \mathbf{h}_0^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}\mathbf{h}_0 \tag{5.14}$$

In effect, the shift sensitivity term, $\Phi$, is a scalar measure of the difference between two lowpass QMF responses: 1) the QMF response elicited from an original image and 2) that resulting from a single pixel diagonally shifted version of the image.

In addition, we are interested in measuring a unidirectional shift sensitivity on the LH and HL subbands. Recall that the LH and HL subband require lowpass filtering in one direction and highpass in the other. Hence, the equation for $\Phi$ given below incorporates the effective response for the HL and LH subbands along with the LL response.

$$\Phi = \mathbf{h}_0^T \mathbf{R}_d (\mathbf{h}_0 + \mathbf{h}_1) \qquad (5.15)$$

Although we attempt to constrain the shifted response of the QMF along only the low frequency paths of the subband decomposition, the resulting QMF still loses some of its energy compaction. Therefore, we balance the shift sensitivity term with an energy compaction term. Together they form the downsampling tolerance constraint.

### 5.4.2 Energy Compaction

Variance of a subband distribution is one measure of its energy compaction, or compression capability. Smaller variance corresponds with more compression. We seek to provide a similar measure for our energy compaction term.

We define the energy compaction measure as a derivative of only the HH subband by performing a highpass filtering operation in both directions on the autocorrelation matrix of a training image. The particulars of the autocorrelation matrix, $\mathbf{R}_x$, are discussed after its definition in Equation 5.10.

$$\Theta = \mathbf{h}_1^T \mathbf{R}_x \mathbf{h}_1 \qquad (5.16)$$

Thus, the energy compaction term, $\Theta$, is a scalar measure of the variance in the HH subband resulting from the training image. Its purpose is to counterbalance the shift sensitivity term and ensure that the QMF bank maintains powerful compression performance.

### 5.4.3 A Shift Tolerant QMF Solution

Recall the baseline QMF design in Equation 4.17. To achieve a shift tolerant solution, we add a fourth term to Equation 4.17, which we call the downsampling tolerance criterion, $\varepsilon_{ds}$.

$$\varepsilon_{ds} = \lambda\Phi + (1-\lambda)\Theta \qquad (5.17)$$

Together the error shift term, $\Phi$, and the energy compaction term, $\Theta$, form the downsampling tolerance constraint, where $\lambda$ is a parameter meant to balance the two opposing functions. Thus, the full-fledged optimization equation for a shift tolerant QMF design follows from Equation 4.17 as below. We scale the downsampling tolerance constraint by 0.2 so as not to overshadow the important QMF properties.

$$f = \varepsilon_{ald} + \varepsilon_{amd} + \varepsilon_{dc} + 0.2\varepsilon_{ds} \qquad (5.18)$$

$$\begin{aligned} f = \sum_{i\neq j}\left[\left(\mathbf{T}\mathbf{T}^{\mathrm{T}}\right)_{ij}\right]^2 + \left(\mathbf{h}_0^{\mathrm{T}}\mathbf{h}_0 - 1\right)^2 + \left(\sum_{n=1}^{N_f} h_0(n) - \sqrt{2}\right)^2 \\ + \; 0.2\left[\lambda\mathbf{h}_0^{\mathrm{T}}\mathbf{R}_d\left(\mathbf{h}_0 + \mathbf{h}_1\right) + (1-\lambda)\mathbf{h}_1^{\mathrm{T}}\mathbf{R}_x\mathbf{h}_1\right] \end{aligned} \qquad (5.19)$$

where $h_1(n) = (-1)^n h_0(n)$ as given in Equation 4.11, $\mathbf{T}$ is derived from $h_0(n)$ as explained in Section 4.1, and the autocorrelation matrices $\mathbf{R}_x$ and $\mathbf{R}_d$ are defined above in Equations 5.10 and 5.12. Again, as in the design of the standard QMF, we use numerical optimization to find the coefficients of $h_0(n)$ that minimize the function $f$. Note that the $\mathbf{R}$ matrices are computed a priori and remain unchanged during the iterative optimization;

$\lambda$ is also a constant parameter. The **T** matrix and highpass filter $h_1(n)$, however, derive from the lowpass filter, $h_0(n)$, and thus are computed at every iteration. The image we use to compute the **R** matrices is one of the preprocessed BMP training images, which is exemplary of the image statistics of the entire data set.

**Shift Sensitivity versus Compression Trade-Offs**

It is worthwhile to study the effects of $\lambda$. First, we examine the compression power of the QMF analysis bank as a function of $\lambda$. As mentioned earlier, variance (or standard deviation) of a subband distribution is one measure of its compression capability. Accordingly, a ratio of the standard deviation of the LL subband to the standard deviation of the HH subband defines our metric of compression power.

$$\text{compression power} = \frac{\sigma_{LL}}{\sigma_{HH}} \qquad (5.20)$$

We compute the compression power empirically as a mean value over the set of preprocessed BMP training images, which are 30 in number. Figure 5.6 plots this metric against varying values of $\lambda$.

As expected, the compression power of the QMF decreases as $\lambda$ increases. More importantly, one can tune the system over only a small range of $\lambda$ from about 0.5 to 0.8. We cannot increase the compression power beyond its point of saturation where $\lambda = 0.5$; likewise it reaches a minimum when $\lambda = 0.8$ and cannot be reduced further. The compression power of our baseline QMF (derived from Equation 4.17) lies somewhere in the middle these two extremes.

Next, we studied the trade-off between the shift sensitivity and energy compaction terms which comprise the downsampling tolerance constraint. We use Equation 5.20

above to reflect the effect of the energy compaction term. To measure the shift sensitivity achieved by the QMF, we compute the squared error between the LL subband

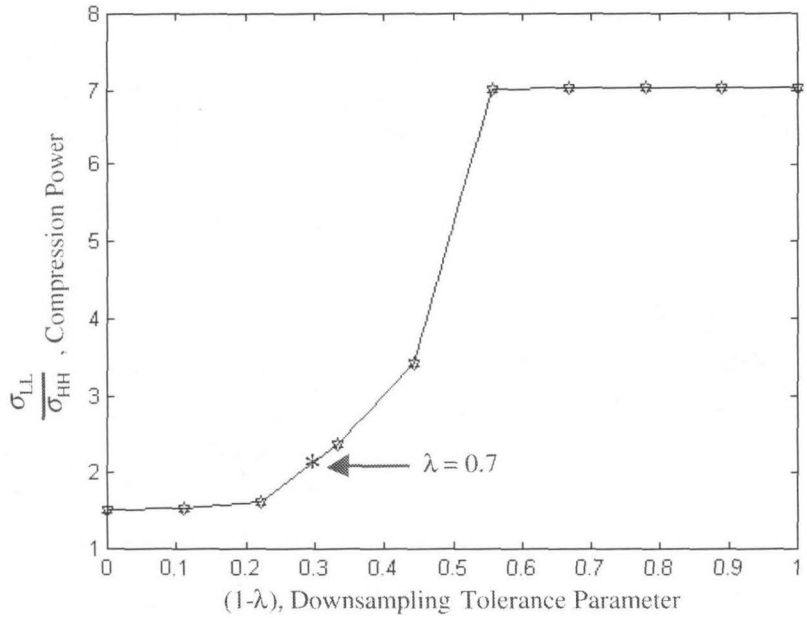



Figure 5.6 **Shift Tolerant QMF Compression Power vs. Lambda**
We vary λ and empirically calculate the mean compression power over a set of training images processed with each resulting QMF bank. Clearly, the smaller λ is, the more the resulting QMF will compress the data, with saturation at both ends. The compression power of our baseline QMF falls in the middle of the small tuning range of lambda.

resulting from a reference image and the LL subband resulting from its circularly shifted copy.

$$\text{shift sensitivity} = \left(x1_{LL}(m,n)\ x2_{LL}(m,n)\right)^2 \quad \forall\, m,n \qquad (5.21)$$

$$\text{where}\quad x2(m,n) = x1(m-1,n-1) \quad \forall\, m,n$$

and $x1_{LL}(m,n)$ represents the LL subband in a decomposition of the image $x1(m,n)$; $x2_{LL}(m,n)$ is analogous. Again, we calculate the shift sensitivity as a mean value over the aforementioned set of imagery. Figure 5.7 plots our empirical metric of compression

power against our empirical metric of shift sensitivity for $\lambda$ from zero to one in steps of 0.1.

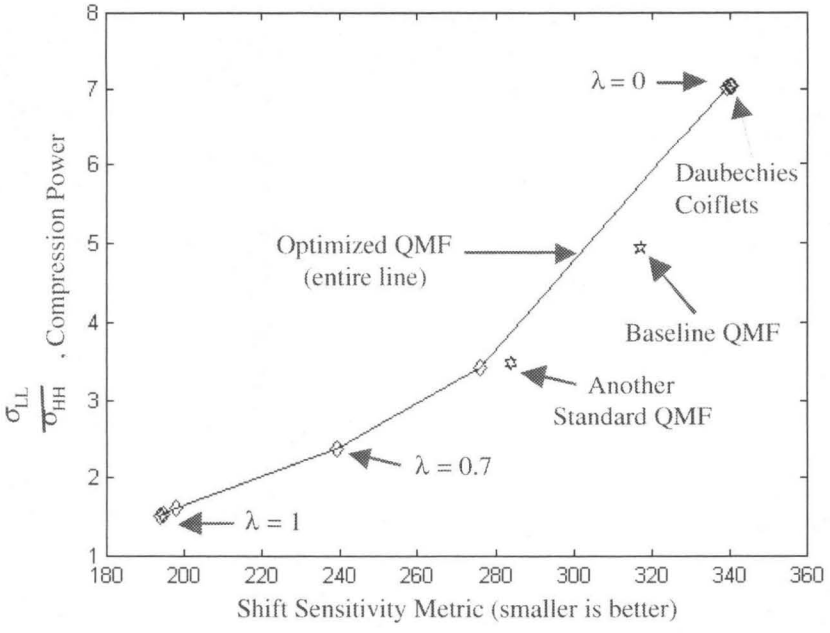



Figure 5.7 **Shift Tolerant QMF Compression Power vs. Shift Sensitivity** We plot mean values empirically calculated over a set of training images processed with various QMF and wavelet filter banks. The points on the optimized QMF line reflect eleven values of $\lambda$ from zero to one in steps of 0.1. As we increase $\lambda$ to improve shift tolerance, we lose compression power. For reference, the plot also indicates the performance of various other QMFs and wavelets.

Clearly, increasing $\lambda$ improves the shift tolerance of a QMF at the cost of compression power, while decreasing $\lambda$ works conversely. Using a $\lambda$ value less than 0.2 allows our QMF design to achieve the same compression power as a sixteen tap Daubechies or Coiflets wavelet. Our baseline QMF (derived from Equation 4.17) lies close to the operating curve with shift tolerance approximating that achieved by the optimized QMF with $\lambda = 0.55$. A different set of initial filter values, allows our baseline QMF design more shift tolerance. The second standard QMF denoted in Figure 5.7 also

uses Equation 4.17, but with a different initial setting of the $h_0(n)$ coefficients. In this case, we initialize $h_0(n)$ to the following values: $h_0\left(0:\frac{N-1}{2}-1\right)=h_0\left(\frac{N+1}{2}+1:N-1\right)=0$, $h_0\left(\frac{N-1}{2}\right)=h_0\left(\frac{N+1}{2}\right)=\sqrt{2}$.)

Again, the operating curve in Figure 5.7 illustrates the small tuning range of $\lambda$. The question is, "*Where to set $\lambda$ so as to provide sufficient shift tolerance for the recognition while not compromising compression power?*" We selected a value of $\lambda = 0.7$ because it provided significantly more shift tolerance than our baseline QMF without a large drop in reconstruction performance.

**Filter Optimization Results**

Table 5.1 reports the performance metrics achieved by the baseline QMF and our optimized QMF. For the remainder of this chapter, all results reported for the optimized QMF use $\lambda = 0.7$.

Table 5.1 **Performance Metrics of the Baseline and Shift Tolerant QMF** PSNR and MSE are reported from a three level decomposition of one of the preprocessed BMP training image. The filter has better performance when the Shift Tolerant Metric is smaller, but when the Compression Metric is larger.

| | PSNR | MSE | Shift Sensitivity Metric | Compression Metric |
|---|---|---|---|---|
| **Baseline QMF** | 105 db | $2\times10^{-6}$ | 317 | 4.9 |
| **Shift Tolerant QMF ($\lambda = 0.7$)** | 67 db | $1\times10^{-2}$ | 239 | 2.4 |

Figure 5.8 diagrams the coefficients of $h_0(n)$ for the shift tolerant optimized QMF with the baseline QMF coefficients shown for comparison. The shift tolerant QMF deviates dramatically from the baseline filter.

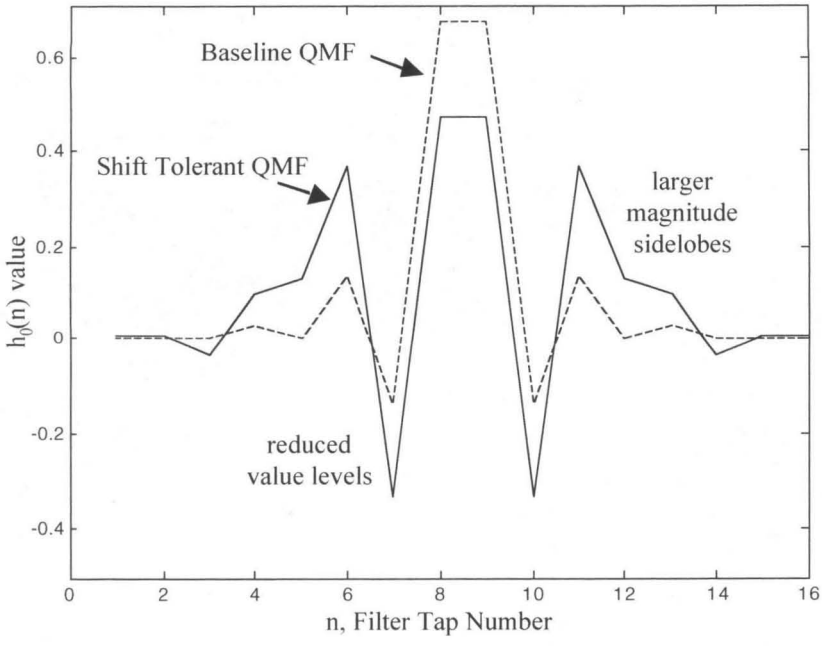



Figure 5.8 **Shift Tolerant QMF Time Domain Response**
The shift tolerant QMF (solid line) values are quite different from our baseline QMF (dashed line). Its center three values are reduced, while the remaining sidelobes increase in magnitude.

We show in Figure 5.9 that the lowpass frequency response of the shift tolerant QMF becomes somewhat of a bandstop filter. That is, it tries to attenuate the frequencies which most affect the shifted response. These frequencies are probably related to the size of the objects under investigation. Very small objects would elicit an attenuation at higher frequencies. In addition to this frequency selectiveness, the shift tolerant QMF unexpectedly magnifies some of the higher frequencies. For comparison, we display the lowpass frequency response of our baseline QMF in the same figure.

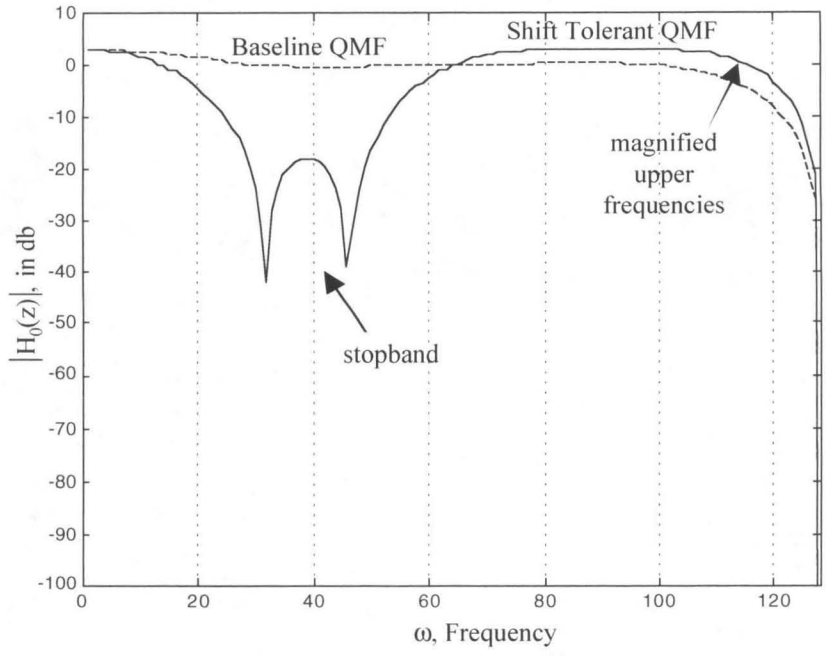



Figure 5.9 **Shift Tolerant QMF Frequency Response**
The shift tolerant QMF (solid line) forms a stopband, as it tries to attenuate the frequencies which most affect the shifted response. The baseline QMF frequency is shown as a dashed line for comparison.

## 5.5 Performance

Finally, we may report on the performance of the SDCFs. The first set of experiments examines the SDCFs ability to overcome the downsampling effects of the subband coder. We assess the two proposed solutions (incorporating downsampling tolerance through training, and designing a downsampling tolerant QMF) both separately and together. We perform this assessment at only level one of the subband decomposition over unquantized image data. This set of data allows us to select one method for SDCF algorithm definition.

Using the algorithm selected in the first set of experiments, we then evaluate the multiresolution performance of SDCFs in a second set of experiments. In this case, we

investigate system results at each level of the subband decomposition. Additionally, we characterize the performance over increased quantization (i.e., decreasing bit rate).

Both sets of experiments followed the conventions outlined in Section 2.5, but we review a few points here.

1) We use the training and test sets defined in Section 2.5.1.
2) We train with the original (i.e., unreconstructed and unquantized) image data.
3) We assume the images are preprocessed according to Section 2.5.2 prior to compression.
4) The SAR imagery has two, four, and eight foot resolution at a one, two, and three level decomposition, respectively.

**Parameter Settings**

We use the following parameter settings for all experiments in this chapter:

- PCF design: $\alpha = 0.2$, $\beta = 0.8$, $\gamma = 0.05$ (Equation 5.5)
- Baseline QMF design: $N_f = 16$, where $N_f$ is the filter length
- Shift tolerant optimized QMF design: $N_f = 16$, $\gamma = 0.7$ (Equation 5.19)
- PCF operation: psr_thresh(1) = 6, psr_thresh(2) = 5, psr_thresh(3) = 4, where the number in parentheses represents the level of decomposition. (Equation 2.8 and Section 2.5.3)

### 5.5.1 Downsampling Tolerance

We have proposed two possible solutions to the downsampling effects inherent in the coefficient domain processing of SDCFs. First, the filters may incorporate downsampling tolerance through training. Secondly, the system may benefit from a shift tolerant QMF design. We examine their effects with the following question, "*Do the*

*proposed solutions, either separately or together, provide enough downsampling tolerance for reasonable system performance?*" To that end, we test four variations of the SDCF algorithm. The models tested are:

1) Plain vanilla SDCFs using the baseline QMF and traditional training techniques
2) SDCFs which incorporates the new training method of training over shifts
3) A system which decomposes the data with a shift tolerant QMF.
4) A system that employs both the new training methodology and a shift tolerant QMF.

As this is a proof of principle experiment, we test with unquantized image data and analyze results at only the first level of the subband decomposition.

Now, to evaluate a system's tolerance to downsampling, we must effectively process a set of test data twice. That is, we must test with the images in one position, then shift the data diagonally by one pixel and reprocess the data, and finally compare the results. Consequently, we report two sets of results for each of the four SDCF models tested. Furthermore, to remove another operating variable, we test on the training set of images. This is because the training images have an established reference point for the image translation. That is, we know the system should perform optimally to the training set in its original position. We do not know what the reference position is with the test images.

**System Performance**

A summary of the system performance metrics is shown below. Table 5.2 reports the object recognition measures Pc, Pe, and Pr, while Table 5.3 provides the reconstruction metrics PSNR and MSE. All these measures are defined in Section 2.5.4.

Table 5.2 **Recognition Metrics for the Four SDCF Systems**
We show the probability of correct classification, error, and rejection for both the original training data and then for the same set after a diagonal shift of one pixel has been applied to all the images. Equivalent results between the two data sets indicate good downsampling tolerance.

| | Reference Test Data | | | Shifted Test Data | | |
|---|---|---|---|---|---|---|
| | **Pc** | **Pe** | **Pr** | **Pc** | **Pe** | **Pr** |
| **Baseline QMF/ Traditional Training** | 97.7% | 2.3% | 0% | 95.4% | 4.6% | 0% |
| **Baseline QMF/ Shift Training** | 98.9% | 0% | 1.1% | 97.7% | 1.15% | 1.15% |
| **Shift Tolerant QMF/ Traditional Training** | 96.6% | 3.4% | 0% | 94.3% | 3.4% | 2.3% |
| **Shift Tolerant QMF/ Shift Training** | 98.9% | 0% | 1.1% | 98.9% | 0% | 1.1% |

Table 5.3 **Reconstruction Metrics for the Four SDCF Systems**
Reconstruction measures are only due to the QMF of the system. Thus, systems with the same QMF achieve the same performance. Both PSNR and MSE are given from the reconstruction of one of the BMP test images.

| | PSNR | MSE |
|---|---|---|
| **Baseline QMF/ Traditional Training** | 105 db | $2\times 10^{-6}$ |
| **Baseline QMF/ Shift Training** | 105 db | $2\times 10^{-6}$ |
| **Shift Tolerant QMF/ Traditional Training** | 67 db | $1\times 10^{-2}$ |
| **Shift Tolerant QMF/ Shift Training** | 67 db | $1\times 10^{-2}$ |

Note that downsampling tolerance is most exemplified by stability across the two data sets. For completion, we provide the confusion matrices for the four systems in the Appendix.

Overall, the system does fairly well on the data set, which we expected because we are testing on the training set. It is unclear why the results on the unshifted data (i.e., the exact versions we trained on) were not 100% correct when using the baseline QMF. Although the decomposition contains enough information to reconstruct the original image, it is not the same information as in the original image. Perhaps the information at a one level decomposition is not linearly separable. In comparison, a quick test of the training data in the uncompressed domain renders 100% correct classification. The baseline QMF does, however, give reasonable reconstruction performance with a PSNR of 105.

**Correlation Performance**

While system performance is informative, the output correlation surfaces give more detail about variations in the output responses. Toward this end, we measured the standard deviation of the Peak to Sidelobe Ratio (PSR_flux) of the correlation output. (Recall the Peak to Sidelobe Ratio was discussed in Section 2.3.)

Given that $y1_i(m,n)$ is the correlation surface resulting from an image $x_i(m,n)$ in the reference data set, and $y2_i(m,n)$ is from its shifted counterpart $x_i(m\text{-}1,n\text{-}1)$ the above PSR_flux metric is defined as

$$\text{PSR\_flux} = \frac{1}{K}\sum_{k=1}^{K}\sigma_{\text{PSR}}(k) \qquad (5.22)$$

$$\text{where } \sigma_{\text{PSR}}(k) = \text{std\_dev}\left(\Delta_{\text{PSR}}(i)\right) \quad \forall\, n \in \text{ object class k,}$$
$$\Delta_{\text{PSR}}(i) = PSR\left(y1_i(m,n)\right) - PSR\left(y2_i(m,n)\right),$$

$K$ is the number of object classes, and PSR is defined in Equation 2.7. The PSR_flux metric has no units. Table 5.4 contains the value of this metric for each of the four SDCF systems we tested. Smaller values reflect a more stable system.

Table 5.4 **Correlation Stability Metric for the Four SDCF Systems**

| | **PSR_flux** |
|---|---|
| Baseline QMF/ Traditional Training | 1.89 |
| Baseline QMF/ Shift Training | 0.98 |
| Shift Tolerant QMF/ Traditional Training | 1.81 |
| Shift Tolerant QMF/ Shift Training | 0.93 |

**Discussion**

The *shift tolerant QMF/ shift training* system appears to be the most stable and thus the most tolerant to the downsampling effects of the subband coder. It also provides the best recognition results. The *baseline QMF/ shift training* system is also fairly stable, provides near the optimum recognition results, but in addition achieves a 57% improvement in the PSNR reconstruction metric.

From this small experiment, it appears that incorporating shifted versions of the input images in the training set has the most impact on downsampling tolerance. Incremental improvements are made by optimizing the QMF for shift tolerance, but at a larger cost in image reconstruction performance. We recommend that the shift training be used alone or in conjunction with the shift tolerant QMF, but that the shift tolerant QMF is not effective when used by itself.

In hindsight, a smaller value of $\lambda$ (Equation 5.17) may have more benefit. From the above experiments, we see that the baseline QMF already has substantial shift tolerance. A value near $\lambda = 0.55$ might have allowed the additional small improvement for recognition stability without compromising the reconstruction performance of the QMF.

**Conclusions**

To summarize the main points found in this set of experiments:

- Both the *baseline QMF/ shift training* system and the *shift tolerant QMF/ shift training* system embody reasonable downsampling tolerance.
- The new training methodology of including shifted versions of the input images in the training set has the most impact on downsampling tolerance.
- Incorporating shift tolerance in the design of the QMF is not by itself enough to substantiate downsampling tolerance, but it does incrementally improve performance of the shift training method.
- The baseline QMF employed in this thesis already exhibits reasonable downsampling tolerance.

### 5.5.2 Multiresolution Analysis and Quantization Effects

In this set of experiments, we assess the performance of SDCFs at lower resolutions in the subband decomposition. In this case, we are no longer concerned with downsampling tolerance, but with performance over the test set of images and the implications of declining bit rate. This set of experiments was designed with the following two questions in mind.

- "*Which of the SDCF systems from Section 5.5.1 proves capable at lower levels of resolution?*"
- "*How will compression degrade the recognition performance of SDCFs at each level of resolution?*"

Corresponding with the architecture of SDCFs outlined in Section 5.2, we decompose the image only to the level required for each experiment below. This is because only the

four subbands from an individual level are fused in the PCF structure. Thus, the quantization tables will be different for the different levels. That is, quantization of a level two decomposition has to quantize the data at both levels one and two. A level one decomposition, however, only quantizes the information at level one, and does not use the quantized data from the level two decomposition. Section 4.4 describes the quantizer design. Recall that the SAR imagery has two, four, and eight foot resolution at a one, two, and three level decomposition, respectively.

**System Selection**

We considered both the *shift tolerant QMF/ shift training* system and the *baseline QMF/ shift training* system for the multiresolution experiments. We performed an initial review with unquantized test data.

Table 5.5 **Multiresolution Performance of Baseline and Shift Tolerant QMF**
One of the BMP test images provides the PSNR and MSE values given below.

| | **level 1 Pc** | **level 2 Pc** | **level 3 Pc** | **PSNR** | **MSE** |
|---|---|---|---|---|---|
| **Baseline QMF/ Shift Training** | 96% | 85% | 75% | 105 db | $2\times10^{-6}$ |
| **Shift Tolerant QMF/ Shift Training** | 93% | 64% | 54% | 67 db | $1\times10^{-2}$ |

Observe that the shift tolerant QMF does not duplicate the success of the baseline QMF system on levels two and three. This is probably due to the lower PSNR and increased MSE of the shift tolerant QMF. PSNR is important not only for visualizing reconstructed data, but also for optimal representations at the lower levels within a subband decomposition. Again, a smaller choice of $\lambda$ (Equation 5.17) may provide a better performing shift tolerant QMF. We selected the *baseline QMF/ shift training* system for the remaining experiments.

Also note that for both systems, results decline as we go further into the decomposition. This is an expected consequence of multiresolution. As you decrease resolution, patterns become more ambiguous and thus harder to discriminate. In addition, any errors resulting from a QMF operation, only multiply with its recursion, adding to the pattern degradation at the lower levels.

**Results**

The following data summarizes the performance results from a set of SDCFs which use the baseline QMF and incorporate the new training methodology of training over shifts. Moreover, we characterize the performance over increased quantization (i.e., decreasing bit rate). Figures 5.10, 5.11, and 5.12 illustrate the object recognition measures Pc, Pe, and Pr, respectively, for a SDCF system at each of the three levels of

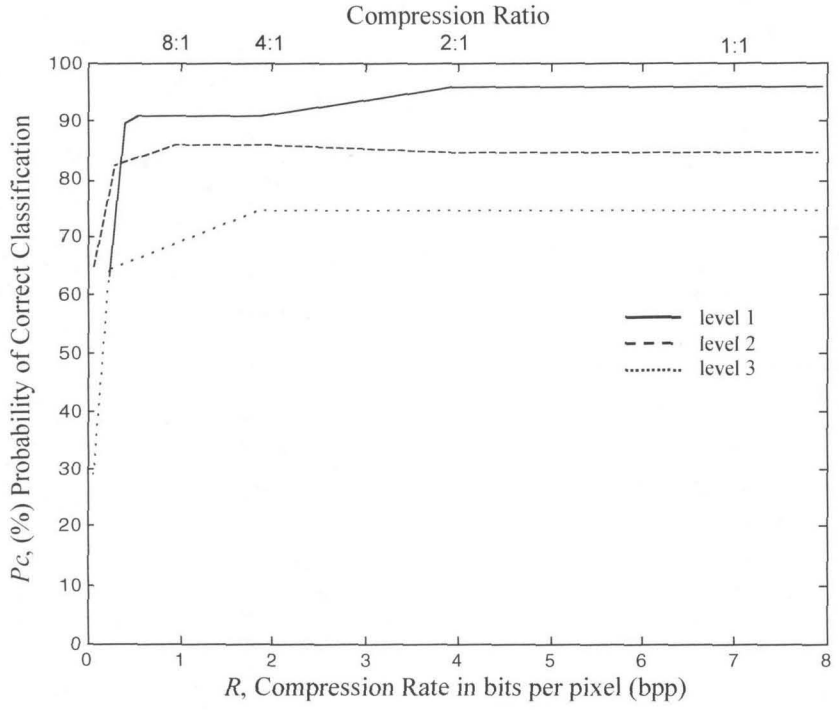



Figure 5.10 **Probability of Correct Classification vs. Bit Rate**

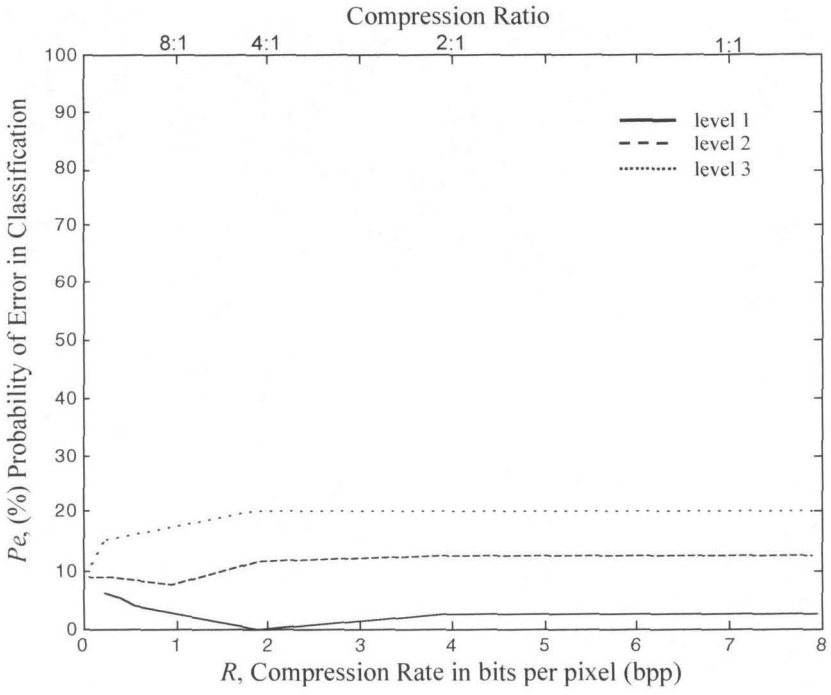



Figure 5.11 **Probability of Error in Classification vs. Bit Rate**

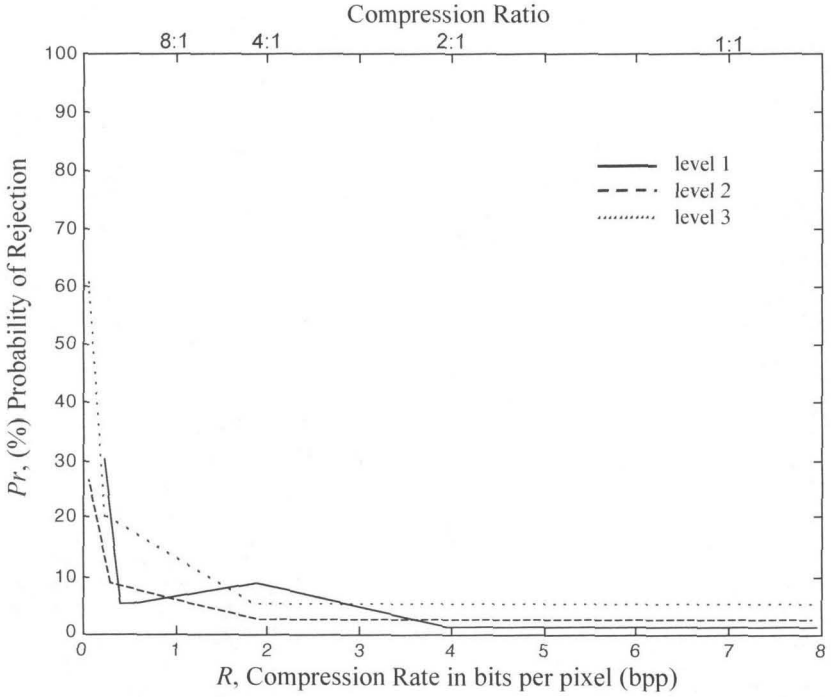



Figure 5.12 **Probability of Rejection vs. Bit Rate**

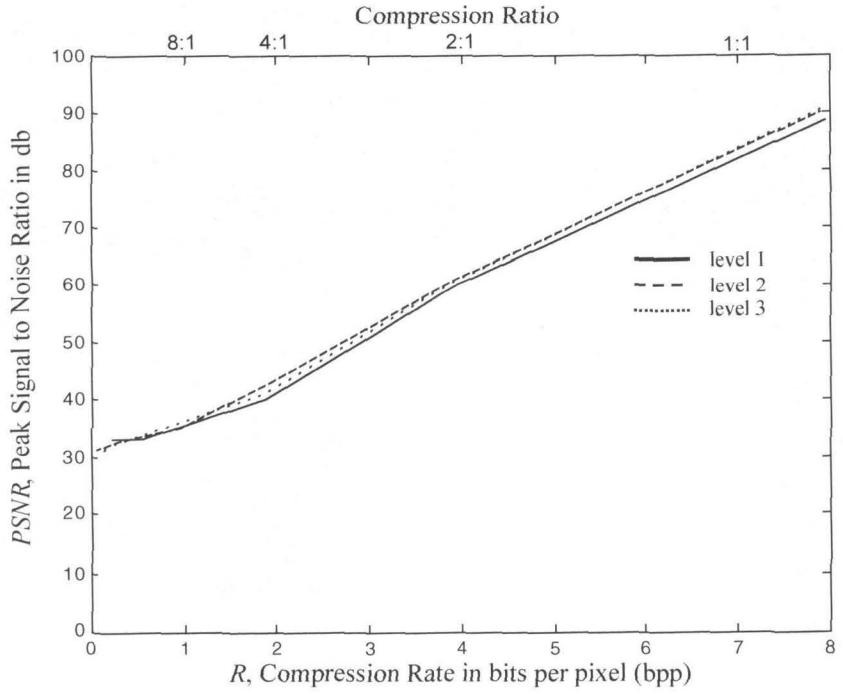



Figure 5.13 **Peak Signal to Noise Ratio vs. Bit Rate**

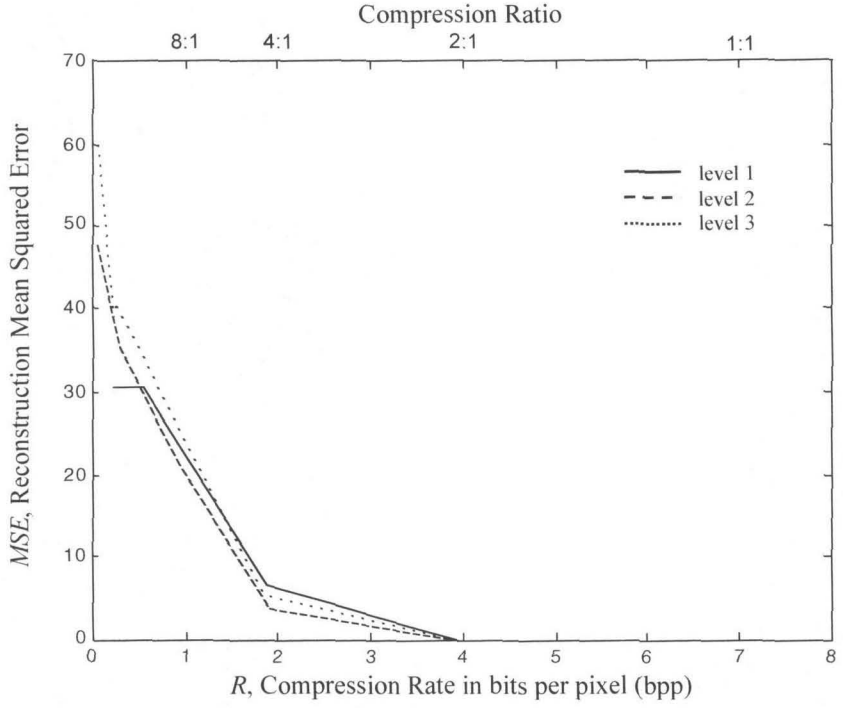



Figure 5.14 **Reconstructed Image MSE vs. Bit Rate**

decomposition as a function of bit rate. We also display the corresponding reconstruction metrics PSNR and MSE as we vary the bit rate in Figures 5.13 and 5.14, respectively.

Observe that all three of the recognition performance curves (Figures 5.10 - 5.12) possess characteristics similar to that of a Rate Distortion curve. That is, they maintain good performance down to very low bit rates and then drop off quickly. The Pe curve (Figure 5.11) displays some deviations, probably due to the small sample size. We see a slight drop off in performance (in the Pc curve) at about two bpp for the first level, and around 0.3 bpp on the second and third level. The significant drop off occurs at 0.25 bpp on the first level, at 0.0625 bpp for the second level, and at 0.05 bpp on the third level. These bit rates correspond to compression ratios of 32:1, 128:1, and 160:1, respectively. For clarity, we tabulate the SDCF performance for all three levels of operation in Tables 5.6 - 5.8.

Recall that the purpose of performing object recognition at lower resolutions is simply to execute a fast, cursory search for the object . Thus, high recognition accuracy, while desirable, is not critical. A more detailed search at full resolution can be done later on the areas that the low resolution search deems interesting. At lower levels of resolution, it is more important to have a low rejection rate. Even an error in classification can flag the application to search more thoroughly at a higher resolution of the data. The SDCFs exhibit this trait down to compression ratios of at least 4:1 as shown in Figure 5.12.

Table 5.6 **Level One SDCF Results as a Function of Bit Rate**

| Compression Rate | | Recognition Metrics | | | Reconstruction Metrics | |
|---|---|---|---|---|---|---|
| **Target Rate** | **Actual Rate** | **Pc** | **Pe** | **Pr** | **PSNR** | **MSE** |
| 8 bpp | 7.96 bpp | 96.2% | 2.5% | 1.3% | 89.9 db | $8\times 10^{-5}$ |
| 6 bpp | 5.98 bpp | 96.2% | 2.5% | 1.3% | 74.5 db | $2.3\times 10^{-3}$ |
| 4 bpp | 3.97 bpp | 96.2% | 2.5% | 1.3% | 59.7 db | $7\times 10^{-2}$ |
| 2 bpp | 1.88 bpp | 91.1% | 0% | 8.9% | 40.1 db | 6.4 |
| 1 bpp | 0.54 bpp | 91.1% | 3.8% | 5.1% | 33.3 db | 30.6 |
| 0.5 bpp | 0.41 bpp | 89.9% | 5.1% | 5.1% | 33.3 db | 30.6 |
| 0.25 bpp | 0.25 bpp | 63.3% | 6.3% | 30.4% | 33.3 db | 30.6 |

Table 5.7 **Level Two SDCF Results as a Function of Bit Rate**

| Compression Rate | | Recognition Metrics | | | Reconstruction Metrics | |
|---|---|---|---|---|---|---|
| **Target Rate** | **Actual Rate** | **Pc** | **Pe** | **Pr** | **PSNR** | **MSE** |
| 8 bpp | 7.89 bpp | 84.8% | 12.7% | 2.5% | 90.1 db | $6\times 10^{-5}$ |
| 6 bpp | 5.95 bpp | 84.8% | 12.7% | 2.5% | 75.2 db | $1.8\times 10^{-3}$ |
| 4 bpp | 3.92 bpp | 84.8% | 12.7% | 2.5% | 60.3 db | $5.6\times 10^{-2}$ |
| 2 bpp | 1.91 bpp | 86.1% | 11.4% | 2.5% | 42.1 db | 3.7 |
| 1 bpp | 0.95 bpp | 86.1% | 7.6% | 6.3% | 34.9 db | 20.7 |
| 0.5 bpp | 0.31 bpp | 82.3% | 8.9% | 8.9% | 32.7 db | 35.3 |
| 0.25 bpp | 0.06 bpp | 64.6% | 8.9% | 26.6% | 31.3 db | 47.7 |

Table 5.8 **Level Three SDCF Results as a Function of Bit Rate**

| Compression Rate | | Recognition Metrics | | | Reconstruction Metrics | |
|---|---|---|---|---|---|---|
| **Target Rate** | **Actual Rate** | **Pc** | **Pe** | **Pr** | **PSNR** | **MSE** |
| 8 bpp | 7.96 bpp | 74.7% | 20.2% | 5.1% | 90.1 db | 2.97 |
| 6 bpp | 5.95 bpp | 74.7% | 20.2% | 5.1% | 75.1 db | 2.97 |
| 4 bpp | 3.94 bpp | 74.7% | 20.2% | 5.1% | 60.2 db | 3.04 |
| 2 bpp | 1.86 bpp | 74.7% | 20.2% | 5.1% | 40.8 db | 9.67 |
| 1 bpp | 0.98 bpp | 74.7% | 20.2% | 5.1% | 35.7 db | 25.4 |
| 0.5 bpp | 0.24 bpp | 64.6% | 15.2% | 20.3% | 32.1 db | 54.3 |
| 0.25 bpp | 0.05 bpp | 29.1% | 10.1% | 60.8% | 30.3 db | 59.9 |

**Discussion**

Several observations are in order. First, we note that these results are slightly worse than the Walls, Mahalanobis results [WM99] which report good performance results up to compression ratios of 200:1, as shown in Table 2.1. This previous work, however, is much different than that presented in this thesis. In [WM99], they fully reconstruct the image from a quantized subband decomposition, and perform recognition on the full resolution (albeit reconstructed) image. Moreover, they employ a very sophisticated coding technique, the Embedded Zerotree Wavelet (EZW) coder [Sha93], whereas our results use the very simplistic coding method outlined in Section 4.4. The EZW is well known to produce some of the best compression rates currently available for image data.

Second, we observe that SDCFs at levels two and three maintain their initial performance at lower bit rates than level one SDCFs. This is primarily due to the fact that the higher levels require more bits to encode each subband, causing 'subband dropping' to occur earlier. At the bit rate of 0.5 bpp, the quantizer does not have enough

bits to effectively code all of the subbands on level one, and encodes the HL, LH, and HH subbands with zero bits (effectively 'dropping' these subbands). The LL subband is then encoded at two bpp and we maintain performance until 0.25 bpp when the quantization of the LL subband becomes too great. The table below displays the bit rates which require an empty subband for each level of decomposition.

Table 5.9 **Subbands Dropped at Low Bit Rates**

| | | Subbands Dropped | | |
|---|---|---|---|---|
| | | **level 1** | **level 2** | **level 3** |
| Approximate Rate | 1 bpp | HL, LH, HH | HH | HH |
| | 0.5 bpp | HL, LH, HH | HH | HH |
| | 0.25 bpp | HL, LH, HH | HL, LH, HH | LH, HH |
| | 0.125 bpp | HL, LH, HH | HL, LH, HH | HL, LH, HH |

We see from the above table that levels two and three retain more subbands at lower bit rates. This is expected, as their smaller size requires fewer bits for full encoding.

Finally, we note that at low bit rates, the output correlation surfaces of the SDCFs contain no aliasing components, in contrast to the PRSC architecture of Chapters 3 and 4. Even when the bit allocation must zero out certain subbands, the PCF architecture still renders a complete correlation surface.

**Conclusions**

The following conclusions address the two questions presented at the beginning of this set of experiments.

- A lower PSNR hurts the performance of the *shift tolerant QMF/ shift training* system at lower levels of resolution. We selected *baseline QMF/ shift training* system for our multiresolution analysis.
- Performance declines on the lower levels of decomposition.

- We achieve reasonable performance at compression ratios of 32:1, 128:1, and 128:1, at decomposition levels one, two and three, respectively.
- Lower levels of resolution maintain their initial performance for longer with decreasing bit rate.
- The SDCF system does not produce aliasing at the correlation output, even at low bit rates.

## 5.6 Computation

Lower resolution input data reduces both recognition accuracy and object location accuracy, but it also greatly reduces the computational requirements of the system. From Figure 5.2, we see that the SDCF system requires the following computational blocks.

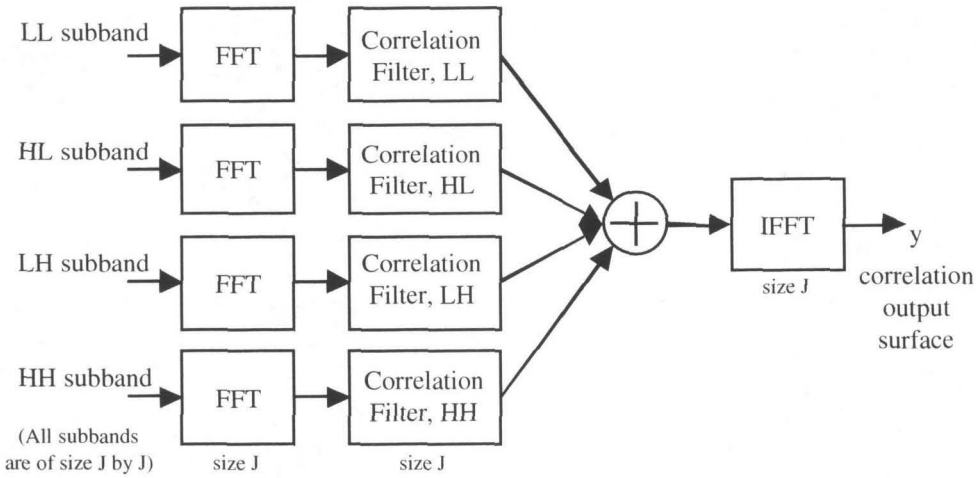



Figure 5.15 **SDCF Computational Blocks**
The parallel architecture requires four FFTs and correlations; one for each subband. Only one IFFT is necessary after all four channels are summed. All computations are for the same size data array.

We wish to quantify the *total computation* of the system for a $k$ level decomposition. Given an original image of size $N$ by $N$, the dimension (on one side) of a subband on

level $k$ is given by $\frac{N}{2^k}$. Now, using the computational building blocks of Section 3.3.1, we establish the following terms for the system of Figure 5.15.

$$\text{FFT Computation} = 4\left(\frac{N}{2^k}\right)^2 \log_2\left(\frac{N}{2^k}\right) \qquad (5.23)$$

$$\text{Correlation Computation} = 4\left(\frac{N}{2^k}\right)^2 \qquad (5.24)$$

$$\text{IFFT Computation} = \left(\frac{N}{2^k}\right)^2 \log_2\left(\frac{N}{2^k}\right) \qquad (5.25)$$

After summing Equations 5.23 - 5.25, we arrive at the total computation of the SDCF system.

$$\text{SDCF Total Computation: } \mathrm{P} = \left(\frac{N}{2^k}\right)^2 \left[5\log_2 N - 5k + 4\right] \qquad (5.26)$$

where $N$ is the size (on one side) of the original image, and $k$ is the level of the decomposition.

Because the system yields a parallel implementation, the time to reach a result is dictated by only one leg of the architecture. We define this as the *effective computation* (akin to the definition in Section 3.3).

$$\text{SDCF Total Effective Computation: } \mathrm{P}' = \left(\frac{N}{2^k}\right)^2 \left[2\log_2 N - 2k + 1\right] \qquad (5.27)$$

Both computation terms (Equations 5.26 and 5.27) are plotted as a function of $N$ and $k$ in Figures 5.16 and 5.17.

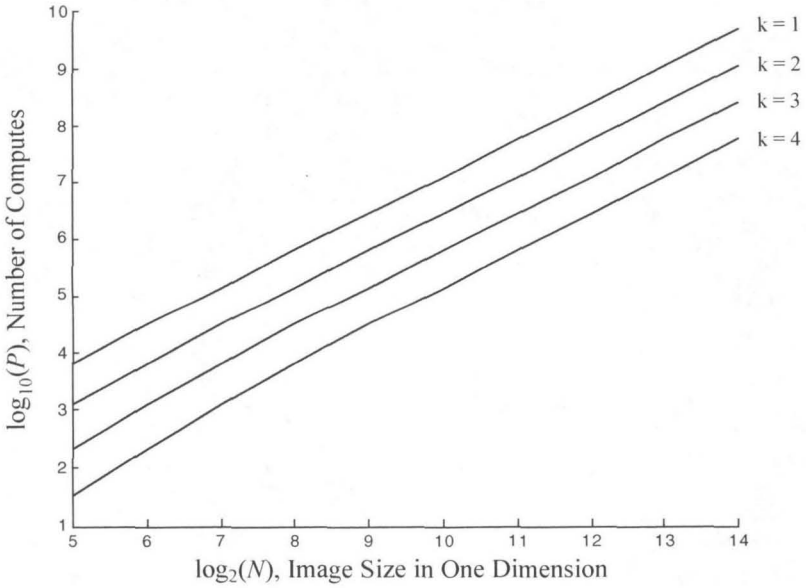



Figure 5.16 **SDCF Total Computation**
The plot above displays the total computations of a level $k$ SDCF system for an image of size $N$. As $k$ increases, the computational savings grows by a constant factor. The total computation also grows exponentially with increasing $N$.

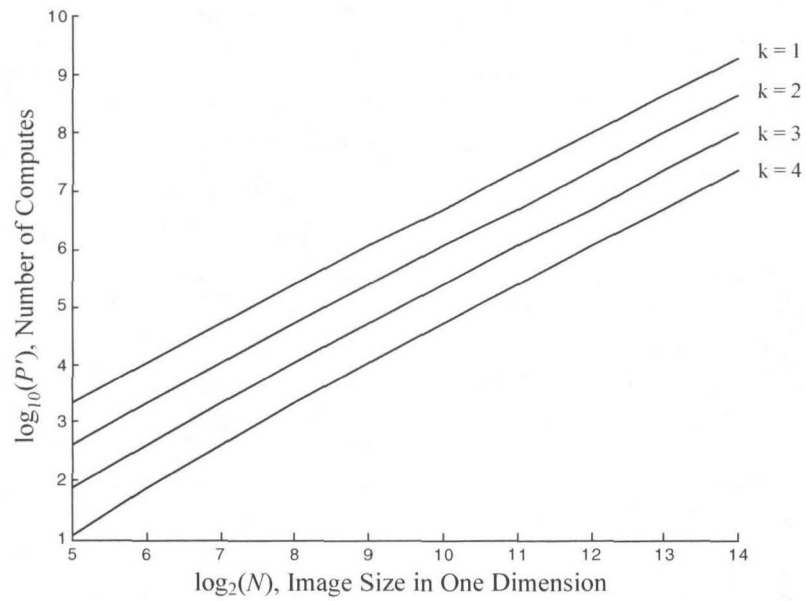



Figure 5.17 **SDCF Effective Computation**
The number of computations required to arrive at an answer in the SDCF system is shown in the plot above as a function of the decomposition level $k$ and image size, $N$. As we increase $k$, the computational savings grows by a constant factor; the total effective computation value grows exponentially with $N$.

Now we turn our interest to the computational savings generated by halving the resolution of the input data (i.e., going down one level in the subband decomposition). By compiling the data from Equations 5.23 - 5.25, the computations follow for subbands of neighboring levels:

Total Computation:

$$P = J^2\left[5\log_2 J + 4\right], \text{ for subband size } J \qquad (5.28)$$

$$P = \left(\tfrac{J}{2}\right)^2\left[5\log_2\left(\tfrac{J}{2}\right) + 4\right] \text{ for subband size } \tfrac{J}{2} \qquad (5.29)$$

Effective Computation:

$$P' = J^2\left[2\log_2 J + 1\right], \text{ for subband size } J \qquad (5.30)$$

$$P' = \left(\tfrac{J}{2}\right)^2\left[2\log_2\left(\tfrac{J}{2}\right) + 1\right] \text{ for subband size } \tfrac{J}{2} \qquad (5.31)$$

Now, the savings in number of computations by halving the resolution is found by taking the difference of Equations 5.28 and 5.29 (likewise 5.30 and 5.31).

$$\text{Total Computation Savings: } D = \tfrac{1}{4}J^2\left[15\log_2 J + 17\right] \qquad (5.32)$$

$$\text{Effective Computation Savings: } D' = \tfrac{1}{4}J^2\left[6\log_2 J + 5\right] \qquad (5.33)$$

In addition, we can use a ratio of Equations 5.28 and 5.29 (also 5.30 and 5.31) to estimate the savings *factors* achieved by halving the resolution.

$$\text{Total Computation Savings Factor: } F = \frac{20\log_2 J + 16}{5\log_2 J - 1} \qquad (5.34)$$

$$F_{J\to\infty} = 4$$

$$\text{Effective Computation Savings Factor: } F' = \frac{8\log_2 J + 4}{2\log_2 J - 1} \qquad (5.35)$$

$$F'_{J\to\infty} = 4$$

where $J$ is the size of the original subband (in one-dimension) in Equations 5.30 - 5.37. For the smallest values of $J$, both $F$ and $F'$ are greater than six, but in the limit, both factors go to the value of four, as shown in the plot below.

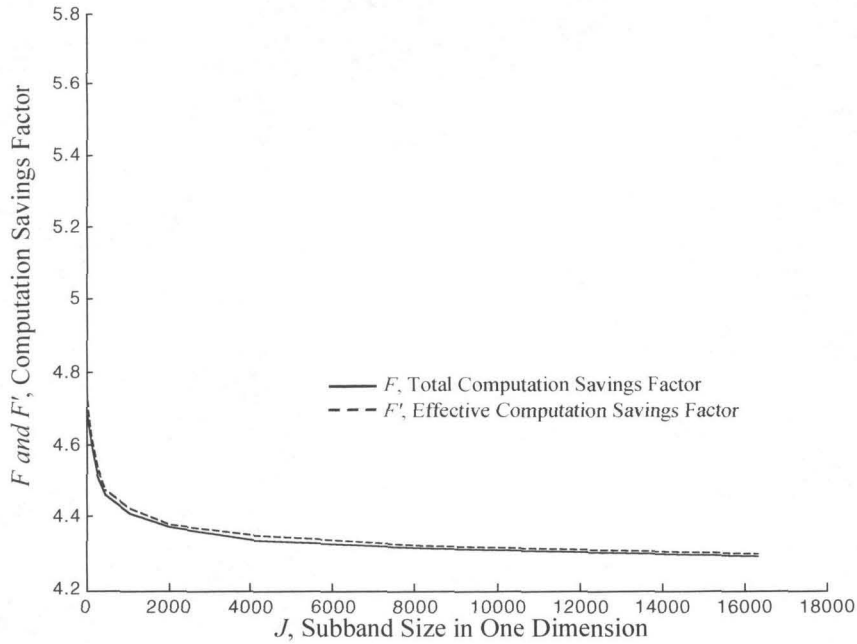



Figure 5.18 **Computation Savings Factor for Halving Input Resolution** While larger than for subbands of reasonable image sizes, the computational savings for halving the input resolution asymptotes to four in the limit. This is true for both the total and effective computation. $J$ is the size of the original subband (i.e., before halving the resolution).

## 5.7 Future Extensions

In this section, we provide a brief synopsis of possible extensions to the SDCF system discussed so far.

First, our marriage of PCFs with a subband decomposition currently weights all the input channels equally. It is simple to scale each input channel (i.e. subband output) differently, which may be fruitful for some applications. Given that most of the downsampling effects arise in the upper subbands, then attenuating their output response

may actually improve the general performance. In addition, a system may want to apply a weight vector to the spectral terms of Equation 5.1 to amplify (or attenuate) specific frequencies.

One extreme case of the above scenario would be to completely drop the HH subband. This subband has the least tolerance to downsampling. Not using its correlation response may improve performance.

Another suggestion we have is to try the minimal training method outlined in Section 5.3.

In addition, we recommend integrating all the subbands from a multilevel decomposition. (That is, incorporate subbands from more than just one level.) This modification, however, will require postprocessing because the correlation responses from different levels are of varying sizes.

Finally, we suggest the inclusion of an additional term, similar to Equation 4.22, in Equation 5.19. One could jointly optimize the SDCFs with respect to recognition, just as in the PRSC system.

## 5.8 Summary

The SDCF system overcomes the downsampling effects of a subband coder to offer multiresolution object recognition. Of the two proposed methods for downsampling tolerance, training over multiple shifts of the data is the most effective. Optimizing the QMF for downsampling tolerance offers an additional slight improvement in the downsampling tolerance, but the resulting decreased fidelity of the QMF transform impairs recognition accuracy.

In summary, we achieve the correct classification performance denoted in Table 5.10.

Table 5.10 **Multiresolution Performance Summary of SDCF System**
The table reports that the SDCF achieves 96.2% correct classification at compression ratios, $R$, equal to or less than 2:1 on two foot resolution data; likewise the system results in 89.9% accuracy at compression ratios equal to or less than 20:1 on the same data. The remaining data are read in an analogous manner.

| | **level 1 (2' resolution)** | **level 2 (4' resolution)** | **level 3 (8' resolution)** |
|---|---|---|---|
| **Probability of Correct Classification** | 96.2%, $R \leq 2{:}1$<br>89.9%, $R \leq 20{:}1$ | 84.8%, $R \leq 8{:}1$<br>82.3%, $R \leq 26{:}1$ | 74.7%, $R \leq 8{:}1$<br>64.6%, $R \leq 32{:}1$ |

With a more sophisticated encoder, the performance reported here would improve at low bit rates. Another benefit of the SDCF system is that its output does not exhibit aliasing, even when low bit rates require the encoder to drop whole subbands of information.

Thus, we have answered three of the questions issued at the beginning of this thesis investigation.

- The SDCF system can perform *object recognition at reduced resolutions* of the input data.
- The performance of the SDCF is successful for *compression ratios* of at least *20:1* for all three resolutions tested. Depending on the level of resolution, accurate recognition occurs at even higher compression ratios.
- For every *reduction in input resolution of one-half*, the SDCF *recognition accuracy decreases by approximately 10%*, but the system's *computational requirements* (both total and effective) also *diminish by a factor of four* or more.
- SDCF performance effects a *tuning curve* similar to a typical rate distortion curve, allowing an application to select an *operating point* balancing the requirements of recognition and compression.

# Chapter 6

# Concluding Remarks

This thesis presents two methods, the PRSC and SDCF, for performing object recognition within compressed imagery. The systems meet our initial objectives:

- To obtain the result directly in the compressed domain. That is, to avoid reconstruction of the image.
- To jointly optimize the performance trinity of recognition, compression, and computation.
- To obtain results at both full and reduced resolutions of the data.
- To make accurate object recognition our primary objective, high compression rates secondary, and faithful reconstruction of the image ternary.

Now we provide answers to the five questions submitted at outset of this thesis investigation.

- The PRSC combines correlation and subband filters in a manner that is *more computationally efficient* than their separate implementation.
- Through parallelism, the PRSC arrives at a result *1.6 times faster*, in the limit, than a correlation system designed for *uncompressed* data.

- With the PRSC, we can provide object recognition on compressed imagery with *superior accuracy* to that of correlation on the *original* imagery, for *compression ratios as high as 6:1*.
- The SDCF performs *successful recognition* for data up to *one-eighth the resolution* of the original data, for *compression ratios of at least 20:1*. It can still be effective at higher compression ratios, depending on the level of resolution.
- Both of our systems provide a *tuning curve* that allows an application to select an *operating point* commensurate with its requirements of recognition and compression.

Previously, image compression and object recognition have generally been treated as separate systems. With the PRSC, we provide a holistic approach to the dual problem by integrating correlation and subband filters into one homogeneous signal processing architecture. Additionally, the SDCF system dissolves the boundary between image compression and object recognition by allowing correlation filters to operate directly on the subband coefficients. Each of these models benefits different aspects of the object recognition in compressed imagery problem. In the table below, we provide brief descriptions outlining the dualities of the two systems.

Table 6.1 **Comparison of PRSC and SDCF**
The two systems investigated in this thesis embody the following attributes.

| **Pattern Recognition Subband Coder** | **Subband Domain Correlation Filters** |
|---|---|
| Full Resolution processing | Reduced Resolution processing |
| Image Domain correlation filters | Subband Domain correlation filters |
| Correlation and subband filters are integrated | Correlation and subband filters are separate entities |
| Downsampling Invariant | Downsampling Tolerant |
| Aliasing at low bit rates | No Aliasing at low bit rates |

It is worth remarking that we used a very simple compression scheme in this thesis. With a more sophisticated coder, like the Embedded Zerotree Wavelet (EZW) algorithm [Sha93], the results at low bit rates would certainly improve.

With that in mind, we summarize this thesis investigation with the following message: *"Through our solutions with compressed imagery, we outperform correlation results on the equivalent original imagery in terms of both speed and accuracy, as well as provide success at reduced resolutions of the data."*

## 6.1 Summary of Results

Due to our tridimensional approach, we are interested in the performance of all three components: compression, recognition, and computation. The opposing demands of compression and recognition have to balance each other, so we present these results together. The computational performance follows separately.

We provide a summary of the performance for four different systems: 1) standard correlation filters operating on original imagery, 2) a baseline compression recognition system which fully reconstructs the image data prior to correlation, 3) the PRSC, and 4) the SDCF. The first three systems provide only full resolution responses, while the last system operates over three levels of reduced resolution.

**Recognition and Compression Performance**

Figure 6.1 characterizes object recognition performance over decreasing bit rate. Several remarks are in order:

- The PRSC provides *superior performance* over that of correlation on the *original* imagery, for *compression ratios as high as 6:1.*
- At *all bit rates*, the PRSC effects *better performance* than the *baseline* method.

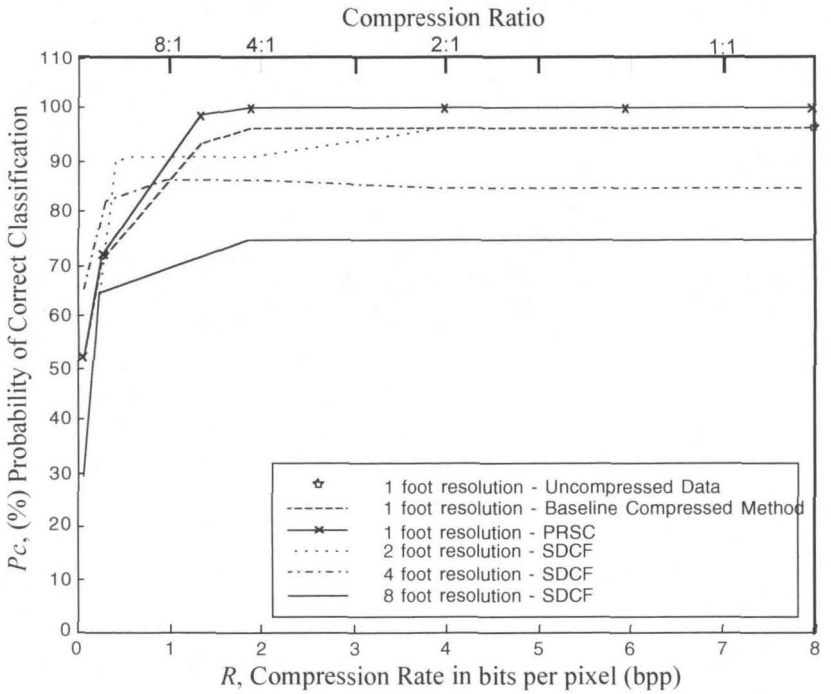



Figure 6.1 **Object Recognition Performance Comparisons**

- The SDCF system operating on *two foot resolution* data *equals* the performance of the *baseline and uncompressed* system (both operating on one foot resolution data), for *compression ratios as high as 6:1.*
- At low bit rates, the SDCF system operating on *two and four foot resolution* data produces *better* results than the PRSC operating on *one foot resolution* because of *aliasing* in the PRSC at low bit rates (due to dropped subbands). This could be alleviated with a more sophisticated coder like the EZW.
- The SDCF results on *two foot resolution* data do not degrade significantly until a *compression ratio of 20:1*; likewise the significant drop-off occurs at a *compression ratio of 26:1 and 32:1* for the *four and eight foot resolution* data, respectively.
- For every *reduction in input resolution of one-half*, the SDCF *recognition accuracy decreases by approximately 10%*, but the system's *computational requirements also*

*diminish by a factor of four* or more, and its *speed increases* by the same factor, as seen in next section.

**Computational Complexity**

In this section, we examine both total and effective computation of the four systems studied in this thesis: 1) standard correlation filters operating on original imagery, 2) a baseline compression recognition system which fully reconstructs the image data prior to correlation, 3) the PRSC, and 4) the SDCF. Again, the first three systems provide only full resolution responses, while the last system operates over three levels of reduced resolution.

We display the total and effective computation as a function of image size in Figures 6.2 and 6.3, respectively. Effective computation is defined as the total number of computations required to reach a result, and is thus a measure of the speed of a system. It can be different than the total computation because of inherent parallelism in certain architectures.

Several observations are in order:

- The PRSC requires *fewer computations* than the *baseline* system. The disparity grows with increasing image size.
- The PRSC operates *faster* than correlation filtering on *uncompressed* imagery by a *factor of 1.6*, in the limit, as image size goes to infinity.
- The *computational requirements* (both total and effective) of the SDCF *decrease* by a *factor of four* by *halving the resolution* of the input data.

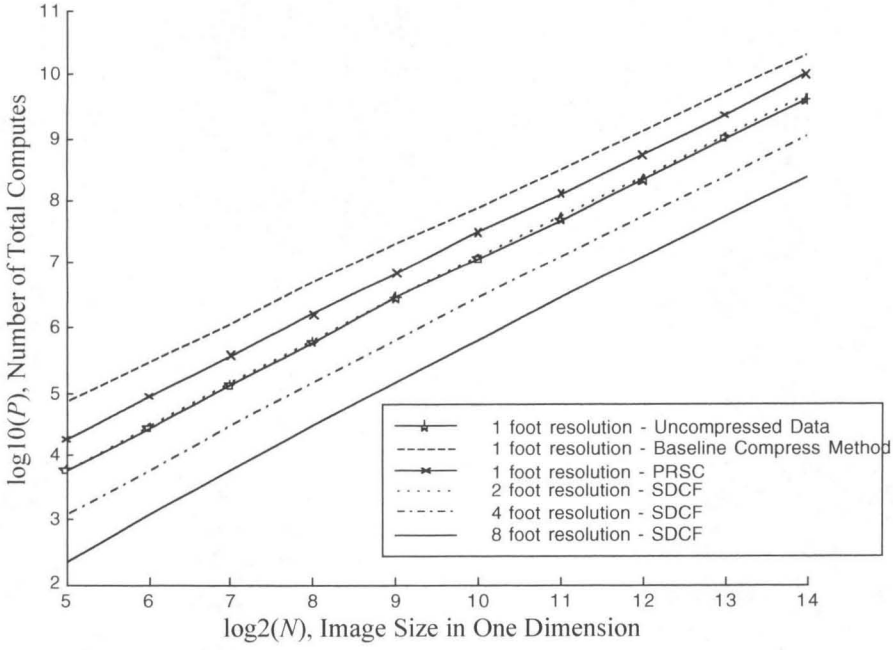



Figure 6.2 **Total Computation Performance Comparisons**

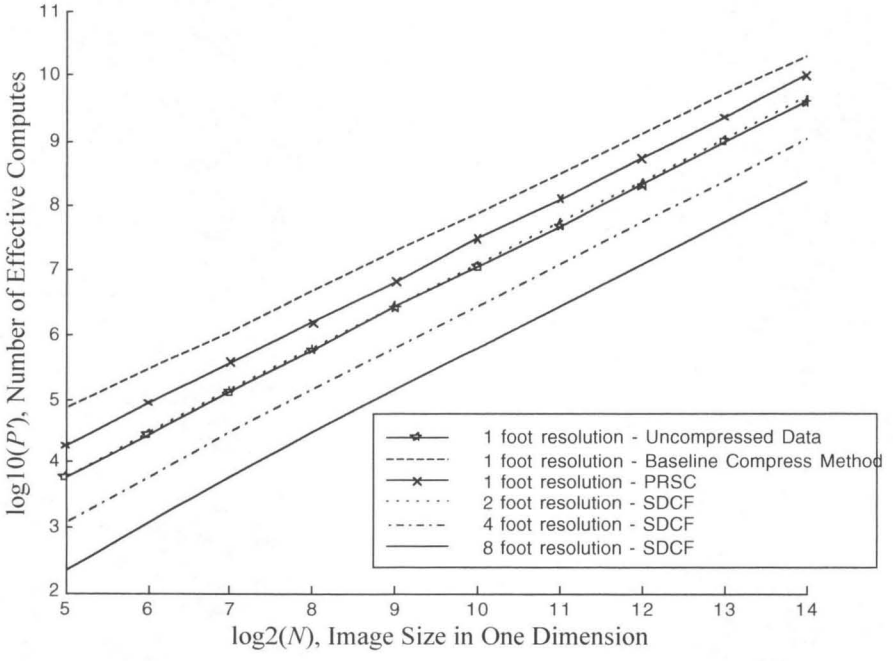



Figure 6.3 **Effective Computation Performance Comparisons**

## 6.2 Summary of Novel Contributions

This thesis presents two distinct methods of performing object recognition in compressed imagery. We reiterate the major contributions made by the author here.

- A new signal processing architecture which *integrates* correlation and subband filters into a *homogeneous* operation, and facilitates effective object recognition directly in the compressed domain. It possesses the following characteristics.
  - It is a *universal architecture* for operating any linear filter directly on the multiresolution data of a subband coder.
  - It produces a response *identical* to that produced by the *reconstructed* image.
  - It results in a response *identical* to that effected by the *original* image, within the fidelity of the subband transform.
  - It is *more computationally efficient* than the separate implementation of the two filters.
  - It provides a *faster response* than that resulting from the *original* data, by a *factor of* $F' = 1.6$, in the limit, as image size goes to infinity.
- A matrix vector formulation for operating linear systems directly in a transform domain. It has the following attributes.
  - It is *universal* for a linear operator and unitary transform.
  - It results in a response *identical* to that produced by the *untransformed* input.
- A method of *jointly optimizing* the *compression capability* of a QMF subband filter and the *recognition performance* of a correlation filter. This method provides:
  - *superior* performance than operating the two filters *independently*
  - *superior* performance to operating on the *uncompressed* data

- A suggestion for *jointly optimizing* the compression and recognition performance to *compensate for quantization.*
- A new pattern recognition algorithm which utilizes correlation filters synthesized in the subband coefficient domain, and allows for object recognition at the *multiple resolutions* of a subband coder.
- A new *training* methodology to *accommodate the downsampling* of the subband coefficients.
- A novel method of *jointly optimizing* the compression capability of a QMF subband filter and the recognition performance of a correlation filter to *compensate for downsampling* in the subband coefficients.

## 6.3 Epilogue

To date, subband coders have afforded the image processing world with effective image compression. The PRSC architecture introduced in this thesis provides more than correlation within a subband coder; it offers a technique for filtering in the subband transform domain. Thus, any image processing operation (e.g., boundary detection, segmentation, feature extraction, or feature description, just to name a few) that can be written as a linear filter, can now be performed in the transform domain with no loss of information, within the fidelity of the subband transform. This step then allows the introduction of quantization, and hence, compression. It remains to be seen what effects compression has on various image processing operations, but effective coding techniques could bring the future of image processing closer to operating directly on compressed imagery.

# Appendix

The following tables provide information detailing the results in Table 5.2. They confirm that the Shift QMF/ Shift Training SDCF System has the most stability. The systems referred to in the tables below are defined in Section 5.5.1.

Table A.1 Confusion Matrix for Baseline QMF/ Traditional Training SDCF

| | | Computed Object Class | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Reference Test Data** | | | | **Shifted Test Data** | | | |
| | | **BMP** | **BTR** | **T72** | **Rej** | **BMP** | **BTR** | **T72** | **Rej** |
| True Object Class | **BMP** | 30 | 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| | **BTR** | 1 | 28 | 0 | 0 | 1 | 27 | 1 | 0 |
| | **T72** | 1 | 0 | 27 | 0 | 2 | 0 | 26 | 0 |

Table A.2 Confusion Matrix for Baseline QMF/ Shift Training SDCF

| | | Computed Object Class | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Reference Test Data** | | | | **Shifted Test Data** | | | |
| | | **BMP** | **BTR** | **T72** | **Rej** | **BMP** | **BTR** | **T72** | **Rej** |
| True Object Class | **BMP** | 30 | 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| | **BTR** | 0 | 29 | 0 | 0 | 1 | 28 | 0 | 0 |
| | **T72** | 0 | 0 | 27 | 1 | 0 | 0 | 27 | 1 |

Table A.3 Confusion Matrix for Sshift QMF/ Traditional Training SDCF

| | | Computed Object Class | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Reference Test Data** | | | | **Shifted Test Data** | | | |
| | | **BMP** | **BTR** | **T72** | **Rej** | **BMP** | **BTR** | **T72** | **Rej** |
| True | **BMP** | 30 | 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| Object | **BTR** | 1 | 27 | 1 | 0 | 2 | 27 | 1 | 0 |
| Class | **T72** | 1 | 0 | 27 | 0 | 1 | 0 | 25 | 2 |

Table A.4 Confusion Matrix for Shift QMF/ Shift Training SDCF

| | | Computed Object Class | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Reference Test Data** | | | | **Shifted Test Data** | | | |
| | | **BMP** | **BTR** | **T72** | **Rej** | **BMP** | **BTR** | **T72** | **Rej** |
| True | **BMP** | 30 | 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| Object | **BTR** | 0 | 29 | 0 | 0 | 0 | 29 | 0 | 0 |
| Class | **T72** | 0 | 0 | 27 | 1 | 0 | 0 | 27 | 1 |

# References

[AP68] H.C. Andrews and W.K. Pratt, "Fourier Transform Coding of Images," *Proc. of Hawaii Int. Conf. on Sys. Science*, pp. 677-679, Jan. 1968.

[AP68] H.C. Andrews and W.K. Pratt, "Transform Image Coding," *Proc. of Comp. Proc. Comm.*, pp. 63-84, Polytechnic Press, 1969.

[And70] H.C. Andrews, *Computer Techniques in Image Processing*, pp. 55-71, Academic Press, 1970.

[BBC98] M. Baldoni, C. Baroglio, and D. Cavagnino, "XFF: A Simple Method to Extract Fractal Features for 2D Object Recognition," *Proc. of Joint Int. Assoc. of Patt. Rec. Workshop*, **SSPR98**, pp. 382-389, 1998.

[BS98] R. Baxter and M. Seibert, "Synthetic Aperture Radar Image Coding," *Linc. Lab. J.*, 11(2), pp. 121-157, 1998.

[BBC76] M. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital Filtering by Polyphase Network: Application to Sample Rate Alteration and Filter Banks," *IEEE Trans. on Acoust., Speech and Sig. Proc.*, **24**, pp. 109-114, Apr. 1976.

[BK88] W.M. Boerner and A.B. Kostinski, "On the Concept of the Polarimetric Matched Filter in High Resolution Radar Imaging," *Proc. of IEEE 1988*

*International Symposium Digest: Antennas and Propagation*, **2**, pp.533-536, 1988.

[Bon95] R. Bonneau, "Using the Centroid Operator for Faster Multiresolution Image Compression and Pattern Recognition," *Proc. of SPIE Conf. on Wave. Appl. in Sig. and Image Proc. III*, **2569**, pp. 813-824, 1995.

[BP98] V. Brennan and J. Principe, "Face Classification Using a Multiresolution Principal Component Analysis," *Proc. of the 1998 IEEE Sig. Proc. Society Workshop*, pp. 506-515, 1998.

[BP93] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates," *IEEE Trans. Patt. Anal. Mach. Intell.*, **15**(10), pp. 1042-1052, October 1993.

[BFP+94] M.C. Burl, U.M. Fayyad, P. Perona, P. Smyth, and M.P. Burl, "Automating the Hunt for Volcanoes on Venus," *Proc. of IEEE Comp. Soc. Conf. on Comp. Vis. and Patt. Rec.*, Seattle, June 1994.

[Bur97] M.C. Burl, *Recognition of Visual Object Classes*, Ph.D. Thesis, California Institute of Technology, 1997.

[Car96] D.W. Carlson, *Optimal Tradeoff Composite Correlation Filters*, Ph. D. Thesis, Carnegie Mellon University, 1996.

[CP84] D. Casasent and D. Psaltis, "Position, Rotation, and Scale Invariant Optical Correlation," *App. Opt.*, **23**, 1984.

[CPG96] N. Chaddha, K.O. Perlmutter, and R.M. Gray, "Joint Image Classification and Compression Using Hierarchical Table Lookup Vector Quantization," *Proc. of IEEE Data Comp. Conf.*, pp.23-32, 1996.

[CR81] R.E. Crochiere and L.R. Rabiner, "Interpolation and Decimation of Digital Signals: A Tutorial Review," *Proc. of IEEE*, **69**, pp. 300-331, Mar. 1981.

[CR83] R.E. Crochiere and L.R. Rabiner, *Multirate Digital Signal Processing*, Prentice Hall, 1983.

[CEG76] A. Croisier, D. Esteban, and C. Galand, "Perfect Channel Splitting by Use of Interpolation/ Decimation Techniques," *Proc. Int. Conf. on Info. Sci. and Sys.*, 1976.

[Cul52] C.C. Culter, "Differential Quantization of Communication Systems," *U.S. Patent 2 605 361*, July 29, 1952.

[CL+60] L.J. Cutrona, E.N. Leith, C.J. Palermo, and L.J. Porcello, "Optical Data Processing and Filtering Systems," *IRE Trans. Info. Theory*, **6**(3), pp. 386-400, June 1960.

[DW98] J. Diemunsch and J. Wissinger, "Moving and Stationary Target Acquisition and Recognition (MSTAR) Model Based Automatic Target Recognition: Search Technology for a Robust ATR," *Proc. of SPIE Conf. on Alg. for SAR Imag. V*, **3370**, pp.481-492, Apr. 1998.

[DH73] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, pp. 276-284, Wiley-Interscience, 1973.

[Eli55] P. Elias, "Predictive Coding - Part I and Part II," *IRE Trans. Info. Theory*, **1**, pp. 16-33, Mar. 1955.

[Ett00] G. Ettinger, "Matchers and Features", *Proc. of SPIE Aerosense Conference*, **4053**, in print, Apr. 2000.

[GW93] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Chs. 6 and 9, Addison-Wesley, 1993.

[Hem95] S.S. Hemami, *Reconstruction of Compressed Images and Video for Lossy Packet Networks*, Ph.D. Thesis, Stanford University, 1995.

[HG95] S.S. Hemami and R.M. Gray, "Subband Filters Optimized for Lost Coefficient Reconstruction," *Proc. of Int. Conf. on Acoust., Speech and Sig. Proc. 1995*, May 1995.

[HC80] C.F. Hester and D. Casasent, "Multivariant Technique for Multiclass Pattern Recognition," *App. Opt.*, **19**, pp. 1758-1761, 1980.

[Hig61] W.H. Highleyman, "An Analog Method for Character Recognition," *IRE Trans. Elec. Comp.*, **10**(3), pp. 502-510, Sep. 1961.

[Hor86] B.K.P. Horn, *Robot Vision*, MIT Press, 1986.

[Hos99] L. Hostetler, "Template Based ATR," *Presentation given at 52nd Automatic Target Recognizer Working Group*, SSC San Diego, CA, June 8, 1999.

[Hot33] H. Hotelling, "Analysis of a Complex of Statistical Variables into Principal Components," *J. Educ. Psych.*, **24**, pp. 417-441, 498-520, 1933.

[INW97] W.W. Irving, L.M. Novak, and A.S. Willsky, "A Multiresolution Approach to Discrimination in SAR Imagery," ," *IEEE Trans. Aerosp. Elec. Sys.*, **33**(4), pp. 1157-1169, Oct. 1997.

[Jai81] A.K. Jain, "Image Data Compression: A Review," *Proc. of the IEEE*, **69**(3), pp. 349-389, Mar. 1981.

[Jai89] A.K. Jain, *Fundamentals of Digital Image Processing*, Chs. 9 and 11, Prentice-Hall, 1989.

[JC84] V.K. Jain and R.E. Crochiere, "Quadrature Mirror Filter Design in the Time Domain," *IEEE Trans. on Acoust., Speech and Sig. Proc.*, **32**(2), pp. 353-360, Apr. 1984.

[JW95] B. Javidi and J. Wang, "Optimum Distortion Invariant Filters for Detecting a Noisy Distorted Target in Background Noise," *J. Opt. Soc. Am. A*, **12**, pp. 604-614, Dec. 1995.

[KG99] T. Keaton and R. Goodman, "A Compression Framework for Content Analysis," *Proc. of IEEE Workshop on Content Based Access of Image and Video Libraries*, pp. 69-73, June 1999.

[L+93] M. Lades, et. al., "Distortion Invariant Object Recognition in the Dynamic Link Architecture," *IEEE Trans. on Comp.*, **42**, pp. 300-311, 1993.

[LM85] K.G. Leib and J.C. Mendelsohn, "Application of the Matched Filter Image Correlator to IR, SAR, and Visual Target Data," *Proc. of SPIE*, **519**, pp. 96-105, 1985.

[LW00] C. Liu and H. Wechsler, "Robust Coding Schemes for Indexing and Retrieval from Large Face Databases," *IEEE Trans. on Image Proc.*, **9**(1), pp. 132-137, Jan. 2000.

[LM96] T-C. Liu and S. Mitra, "Fingerprint Recognition of Wavelet Based Compressed Images by Neuro-Fuzzy Clustering," ," *Proc. of SPIE Conference on Applications of Fuzzy Logic Technology III*, **2761**, pp. 76-86, April 1996.

[Mah87] A. Mahalanobis, *New Correlation Filters for Symbolic and Rule Based Pattern Recognition*, Ph. D. Thesis, Carnegie Mellon University, 1987.

[MF+94] A. Mahalanobis, A. Forman, N. Day, M. Bower, and R. Cherry, "Multiclass SAR ATR Using Shift Invariant Correlation Filters," *Patt. Rec.*, **27**(4), pp. 619-626, 1994.

[MO+99] A. Mahalanobis, L. Ortiz, and B.V.K. Vijaya Kumar, "Performance of the MACH/DCCF Algorithms on the Ten Class Public Release MSTAR Dataset," *Proc. of SPIE*, **3721**, pp. 285-291, Apr. 1999.

[MO+99] A. Mahalanobis, L. Ortiz, B.V.K. Vijaya Kumar, and A. Ezekial, "ATR Performance Using XPATCH (Synthetic) Training Data," *Proc. of SPIE*, **4053**, in print, Apr. 2000.

[MS+93] A. Mahalanobis, S. Song, M. Petragalia and S.K. Mitra, "Adaptive FIR Filters Based on Structural Subband Decomposition for System Identification Problems," *IEEE Trans. on Circ. and Sys.*, **40**, pp. 354-362, June 1993.

[MVK96] A. Mahalanobis and B.V.K. Vijaya Kumar, "Prediction and Comparison of Performance of Correlation Filters," *Proc of SPIE Adv. in Opt. Info. Proc. VII*, **2754**, pp. 136-141, 1996.

[MVK97a] A. Mahalanobis and B.V.K. Vijaya Kumar, "Polynomial Filters for Higher Order Correlation and Multi-Input Information Fusion," *Proc. of SPIE Euro American Workshop on Optoelec. Info. Proc.*, pp. 221-231, June 1997.

[MVK97b] A. Mahalanobis and B.V.K. Vijaya Kumar, "Optimality of the Maximum Average Correlation Height Filter for Detection of Targets in Noise," *Opt. Eng.*, **36**(10), pp. 2642-2648, Oct. 1997.

[MVK+87] A. Mahalanobis, B.V.K. Vijaya Kumar, and D. Casasent, "Minimum Average Correlation Energy Filters," *App. Opt.*, **26**, pp. 3633-3640, Sep. 1987.

[MVKS96] A. Mahalanobis, B.V.K. Vijaya Kumar, S.R.F. Sims, "Distance Classifier Correlation Filters for Multiclass Target Recognition," *App. Opt.*, **35**(17), pp. 3127-3133, June 1996.

[MVK+94] A. Mahalanobis, B.V.K. Vijaya Kumar, S. Song, S.R.F. Sims, and J.F. Epperson, "Unconstrained Correlation Filters," *App. Opt.*, **33**(17), pp. 3751-3759, June 1994.

[Mal89] S.G. Mallat, "A Theory for Multiresolution Signal Decomposition – the Wavelet Representation," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, **11**(7), pp. 674-693, 1989.

[Mar82] D. Marr, *Vision*, W.H. Freeman and Company, 1982.

[MED56] W. Meyer-Eppler and G. Darius, "Two-Dimensional Photographic Autocorrelation of Pictures and Alphabet Letters," *Proc. of 3rd London Symp. on Info. Theory*, pp. 34-36, 1956.

[Mil95] P.C. Miller, "Reduced Resolution Synthetic Discriminant Function Design by Multiresolution Wavelet Analysis," *Appl. Opt.*, **34**(5), pp. 865-878, Feb. 1995.

[MN95] H. Murase and S. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance," *Int. J. of Comp. Vis.*, **14**, pp. 5-24, 1995.

[NS95] J.W. Nahm and M.J. Smith, "Optimized Encoder Design Algorithm for Joint Compression and Recognition," *Proc. of SPIE*, **2484**, pp. 236-245, 1995.

[Nor43] North, Title unavailable. Technical Report, RCA, 1943.

[NO+97] L.M. Novak, G.J. Owirka, W.S. Brower, and A.L. Weaver, "The Automatic Target Recognition System in SAIP," *Linc. Lab. J.*, **10**(2), pp. 187-202, 1997.

[NO+98] L.M. Novak, G.J. Owirka, and W.S. Brower, "Performance of a Twenty Target MSE Classifier," *Proc. of SPIE Conference on Radar Sensor Tech. III.*, **3395**, pp. 138-151, Apr. 1998.

[NO+99] L.M. Novak, G.J. Owirka, and A.L. Weaver, "Automatic Target Recognition Using Enhanced Resolution SAR Data," *IEEE Trans. Aerosp. Elec. Sys.*, **35**(1), pp. 157-175, Jan. 1999.

[Oli52] B.N. Oliver, "Efficient Coding," *Bell Sys. Tech. J.*, **31**, pp. 724-750, July 1952.

[PPS96] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook – Content Based Manipulation of Image Databases," *Int. J. of Comp. Vis.*, **18**(3), pp. 233-254, June 1996.

[P+96] K.O. Perlmutter, S.M. Perlmutter, R.M. Gray, R.A. Olshen, and K.L. Oehler, "Bayes Risk Weighted Vector Quantization with Posterior

Estimation for Image Compression and Classification," *IEEE Trans. of Image Proc.*, **5**(2), pp.347-60, Feb. 1996.

[PL95] A.R. Pope and D.G. Lowe, "Learning Feature Uncertainty Models for Object Recognition," *IEEE Int. Symp. on Comp. Vis.*, 1995.

[Pra78] W.K. Pratt, *Digital Image Processing*, Parts 5 and 6, John Wiley and Sons, 1978.

[RJ91] M. Rabbani and P. Jones, *Digital Image Compression Techniques*, SPIE Press, 1991.

[Say96] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, 1996.

[SS88] G.F. Schils and D.W. Sweeney, "Optical Processor for Recognition of Three-Dimensional Targets Viewed from any Direction," *J. Opt. Soc.Am. A*, **5**, pp. 1309-1321, 1988.

[Sha95] S. Shams, "Multiple Elastic Modules for Visual Pattern Recognition," *Neural Networks*, **8**(9), pp. 1439-1456, 1995.

[Sha48] C.E. Shannon, " A Mathematical Theory of Communication," *Bell Sys. Tech. J.*, **27**, pp. 379-423, 623-656, 1948.

[Sha93] J.M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Trans. on Sig. Proc.*, **41**(12), pp. 3445-3462, 1993.

[SK97] F.B. Shin and D.H. Kil, "Integrated Approach to Bandwidth Reduction and Mine Detection in Shallow Water with Reduced Dimension Image Compression and Automatic Target Recognition Algorithms," *Proc of SPIE*, **3079**, pp. 203-212, April 1997.

[Sim99] K. Simonson, "Statistical Models in ATR", *Proc. of SPIE Aerosense Conference*, **4053**, in print, Apr. 2000.

[SK87] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," *J. of Opt. Soc. of Am.*, **4**(3), pp. 519-524, 1987.

[SN96] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

[TJ97] I. Ternovskiy and T. Jannson, "Novel Video Processing Schemes Integrating Image Compression and Pattern Recognition," *Proc of SPIE*, **3164**, pp. 179-185, 1997.

[Tur60] G.L. Turin, "An Introduction to Matched Filters," *IRE Trans Info. Theory*, **6**(3), pp. 311-329, June 1960.

[TK91] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. of Cog. Neurosc.*, **3**(1), 1991.

[Vai87a] P.P. Vaidyanathan, "Theory and Design of M-Channel Maximally Decimated Quadrature Mirror Filters With Arbitrary M, Having the Perfect Reconstruction Property," *IEEE Trans. on Acoust., Speech and Sig. Proc.*, **35**(4), pp. 476-492, Apr. 1987.

[Vai87b] P.P. Vaidyanathan, "Quadrature Mirror Filter Banks, M-Band Extensions and Perfect Reconstruction Techniques," *IEEE ASSP Magazine*, **4**, pp. 4-20, June 1987.

[Vai88] P.P. Vaidyanathan, "A Tutorial on Multirate Digital Filter Banks" *Proc. of IEEE Symp. on Circ. and Sys.*, pp. 2241-2248, June 1988.

[Vai90] P.P. Vaidyanathan, "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial" *Proc. of the IEEE*, **78**(1), pp. 56-93, Jan. 1990.

[Vai93] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1993.

[VL97] N. Vasconcelos and A. Lippman, "Library Based Coding: A Representation for Efficient Video Compression and Retrieval," *Proc. of IEEE Data Comp. Conf.*, pp. 121-130, Mar. 1997.

[VK96] A. Vellaikal and C-C.J. Kuo, "Joint Spatial Spectral Indexing for Image Retrieval," *Proc. of IEEE Int. Conf. on Image Proc.*, **3**, pp. 867-870, Sep. 1996.

[Vet87] M.A. Vetterli, "A Theory of Multirate Filter Banks," *IEEE Trans. on Acoust., Speech and Sig. Proc.*, **35**(3), pp. 356-372, Mar. 1987

[VK95] M.A. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, 1995.

[VKu86] B.V.K. Vijaya Kumar, "Minimum Variancs Synthetic Discriminant Functions," *J. of Opt. Soc. of Am. A*, **3**, pp. 1579-1584, 1986.

[VKu92] B.V.K. Vijaya Kumar, "Tutorial Survey of Composite Filter Designs for Optical Correlators," *App. Opt.*, **31**, pp. 4773-4801, Aug. 1992.

[VKMJ00] B.V.K. Vijaya Kumar, A. Mahalanobis, and R. Juday, *Correlation Pattern Recognition*, Cambridge Press, in press.

[VKM+92] B.V.K. Vijaya Kumar, A. Mahalanobis, S. Song, S.R.F. Sims, and J.F. Epperson, "Minimum Squared Error Synthetic Discriminant Functions," *Opt. Eng.*, **31**(5), pp. 915-922, May 1992.

[Wal91] G.K. Wallace, "The JPEG Still Picture Compression Standard," *Communications of the ACM*, **34**, pp. 31-44, Apr. 1991.

[WM99] B. Walls and A. Mahalanobis, "Performance of the MACH Filter and DCCF Algorithms in the Presence of Data Compression," *Proc. of SPIE Conf. on Alg. for SAR Imag. VI*, **3718**, pp.376-387, Apr. 1999.

[WG90] T. Walsh and M. Giles, "Statistical Filtering of Time Sequenced Peak Correlation Responses for Distortion Invariant Recognition of Multiple Input Objects," *Opt. Eng.*, **29**, 1052-1064, 1990.

[Wel84] T.A. Welch, "A Technique for High Performance Data Compression," *IEEE Computer*, pp. 8-19, June 1984.

[WvdM93] L. Wiskott and C. von der Malsburg, "A Neural System for the Recognition of Partially Occluded Objects in Cluttered Scenes," *Int. J. of Patt. Rec. and Art. Intell.*, **7**(4), pp. 935-948, 1993.

[WR+99] J. Wissinger, R. Ristroph, J.R. Diemunsch, W.E. Severson, E. Fruedenthal, "MSTAR's Extensible Search Engine and Model Based Inferencing Toolkit," *Proc. of SPIE Conf. on Alg. for SAR Imag. VI*, 3721, pp.554-570, Apr. 1999.

[Yui91] A.L. Yuille, "Deformable Templates for Face Recognition," *J. of Cog. Neurosc.*, **3**(1), pp. 59-70, 1991.

[ZL77] J. Ziv and A. Lempel, "A Universal Algorithm for Data Compression," *IEEE Trans. on Info. Theory*, **23**(3), pp. 337-343, May 1977.

[ZL78] J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable Rate Coding," *IEEE Trans. on Info. Theory*, **24**(5), pp. 530-536, Sept. 1978.