IMAGE CODING FOR DIGITIZED LIBRARIES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Ömer Nezih Gerek
September 1998

IMAGE CODING FOR DIGITIZED LIBRARIES

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Ömer Nezih Gerek

Ömer Nezih GEREK

September 1998

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

A. Enis Çetin, Ph. D.(Supervisor)

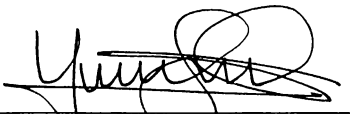
I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Levent Onural, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

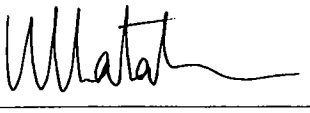
Orhan Arıkan, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.



Uğur Güdükbay, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.



Volkan Atalay, Ph. D.

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

IMAGE CODING FOR DIGITIZED LIBRARIES

Ömer Nezih Gerek
Ph.D. in Electrical and Electronics Engineering
Supervisor: A. Enis Çetin, Ph. D.
September 1998

In this thesis, image coding methods for two basic image types are developed under a digitized library framework. The two image types are gray tone or color images, and binary textual images, which are the digitized image versions of text documents. The gray tone images are encoded using an adaptive subband decomposition followed by zerotree quantizers. The adaptive subband decomposition filter bank adaptively updates the filter bank coefficients in which the values of one of the subbands is predicted from the other subband. It is observed that the adaptive subband decomposition performs better than a regular subband decomposition with a fixed filter bank in terms of compression. For the binary textual images, a compression algorithm using binary subband decomposition followed by a textual image compression (TIC) method that exploits the redundancy in repeating characters is developed. The binary subband decomposition yields binary sub-images, and the TIC method is applied to the low band sub-image. Obtaining binary sub-images improves compression results as well as pattern matching time of the TIC method. Simulation results for both adaptive subband decomposition and multiresolution TIC methods indicate improvements over the methods described in the literature.

Keywords: Digitized Libraries, Image Compression, Adaptive Subband Decomposition, Textual Image Compression,Binary Subband Decomposition, Binary Image Coding, Document Retrieval.

ÖZET

SAYISALLAŞTIRILMIŞ KÜTÜPHANELER İÇİN GÖRÜNTÜ KODLAMA

Ömer Nezih Gerek
Elektrik ve Elektronik Mühendisliği Doktora
Tez Yöneticisi: Dr. A. Enis Çetin
Eylül 1998

Bu tezde, sayısallaştırılmış kütüphane yapısı altında iki temel görüntü türü için görüntü kodlama yöntemleri geliştirilmiştir. Bu iki görüntü tipi gri tonlu / renkli görüntü ve ikili yazılı döküman görüntüleridir. Gri tonlu görüntüler, uyarlamalı altbant ayrıştırma ardından uygulanan bir "sıfır-ağacı" (zerotree) kodlayıcısı ile sıkıştırılmaktadır. Uyarlamalı altbant ayrıştırma süzgeç bankası, bir altbant işaretinin diğer altbant işareti kullanılarak kestirildiği süzgeç katsayılarını güncelleştirmektedir. Uyarlamalı altbant ayrıştırmaya dayalı yöntemin, sıradan altbant ayrıştırma yöntemine nazaran sıkıştırma açısından daha iyi sonuç verdiği gözlenmiştir. İkili yazı görüntüleri için ise ikili altbant ayrıştırma, ve ardından özel bir Yazılı Görüntü Sıkıştırma (YGS) metodu kullanan bir yöntem geliştirilmiştir. YGS yöntemi, yazılı görüntü içinde kendini tekrar eden harf resimlerinin oluşturduğu gereksiz bilgiyi, sadece harflerin tekrar ettiği yerleri kodlayarak açığa çıkarmaktadır. İkili altbant ayrıştırma, iki seviyeli altbant görüntüleri oluşturmaktadır. YGS yöntemi, önerilen sistemde düşük salınım içeren (low-low) altbant görüntüsü üzerinde çalıştırılmıştır. İkili altbant görüntüleri elde ederek hem sıkıştırma oranları, hem de karakter tarama hızı, orijinal YGS yöntemine nazaran artırılmıştır. Benzeşim çalışmaları, hem gri tonlu resim sıkıştırmada hem de ikili yazılı görüntü sıkıştırmada literatürdeki yöntemlere nazaran gelişme elde edildiğini ortaya koymuştur.

Anahtar Kelimeler: Sayısallaştırılmış Kütüphaneler, Görüntü Sıkıştırma, Uyarlanabilir Altbant Ayrıştırma, Yazılı Görüntü Sıkıştırma, İkili Görüntü Kodlama, Döküman Tarama.

ACKNOWLEDGEMENT

I gratefully thank my supervisor Prof. Dr. Enis Çetin for his supervision, guidance, and suggestions throughout the development of this thesis. He was much more than a supervisor.

It is a pleasure to express my special thanks to my mother, and father for their love, support and encouragement.

Many thanks to all of my close friends for their help and friendship throughout all these years.

To whom it may concern.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

In this thesis, various coding algorithms are developed to handle different types of images in digitized libraries which are typical applications of a Visual Information Management Systems (VIMS) [1]- [6]. Coding, communication, and visualization of results are the basic elements for such systems. Many researchers are currently interested in the development of new techniques for (VIMS) [1]- [6]. Efficiency of a VIMS directly depends on developing new techniques in all aspects of databases, computer vision, coding, and knowledge representation and management.

Efficient coding (compression) of visual information is an important issue in many applications, including digitized libraries. There are many such well known image database applications that are under extensive research [14]- [16]. These include image databases for educational purposes [17] (educational network, lectures on videos, interactive encyclopedia, etc), CAD and other engineering media to increase productivity, medicine (Picture archiving and communication systems - PACS [3], [18], [19], inter-hospital communication and automatic radiological image annotation), satellite communication (maps, weather, transportation, video, transportation, etc.) [20], and archiving of large amount of printed documents [61].

In this thesis, the coding of images of different content is considered. The types of images for which the compression algorithms are developed are gray tone images and binary textual images, which are the two basic types of images found in a digitized library.

Image archive is a typical database type for a VIMS. The efficiency in image coding plays an important role in determining the success of the application [7]. In digitized libraries, the two basic types of images should be handled with different compression tools. In the next two chapters, the gray tone / color images and binary textual images will be considered differently and coding algorithms for the two different image types will be developed.

Almost all image database applications require data compression. In the literature, there are commonly used standards for still image compression [8], motion picture (video) compression [9], [10], and mixed audiovisual data [11]. It is difficult to cope with the high demand and supply of image data unless the images are efficiently compressed [12], as well as they are organized and processed for quick retrieval and decoding on demand [13]. In VIMS and other multimedia applications, the image databases should also contain information about their contents. The user should be able to search image databases with image-based and keyword queries. These two types of queries for images require organization of the coded bit stream, indexing, and query processing in a manner different than most of the alphanumeric databases. In this thesis, the binary textual image coding method is modified to improve the efficiency of keyword search. The modification is done by organizing the repeating character images that form the textual image.

In this thesis, the encapsulating frame work of "digitized library" stands for the image database composed of *literally* digitizing the contents and images of a conventional library. Since most of such digitized images contain the two different types of images that were indicated, these images should be separated from each other by segmentation algorithms, and then they should be fed to the encoders corresponding to their image type. High compression of the images is the main goal for both the gray tone / color images and the binary textual images. However, a special fast keyword search issue is also investigated for textual images. In other words, the coding strategies for the textual images should also enable quick and efficient database search.

The need for a gray tone / color image compression algorithm and textual image compression algorithm in a digitized library can be illustrated with the following example. Consider a visualization system for library images which is a very important problem for a good user interface library browsing system. Although a person can usually search for the required book using author, title or keyword search, most of the time it is more efficient to go to the library and search for the books inside the shelves, manually. Usually, the books with similar subjects are put together in a shelf. By previewing the whole shelf, the user can access not only to the book sought, but to other relevant books, as well. When the image of a library shelf is considered, gray tone or color image coding techniques should be utilized. Typically, continuous tone coding can be used for the encoding of graphics and pictures on book covers and image parts inside the books, and binary text compression techniques can be used for printed text parts of the books. The user can hierarchically proceed from the shelf image to the book covers, and from there to the pages inside the book, and each of these images can be compressed using the appropriate method.

In order to improve the coding efficiency, the color, gray-tone and binary parts of the document image will be considered separately throughout this thesis. In Chapter 2, the compression methods devised for continuous tone images will be investigated. An efficient adaptive subband decomposition method for compression will be developed in a polyphase structure, and simulation examples will be presented. The signal adapted filter bank of the subband structure removes the predictable parts of one of the polyphase components from the other polyphase component using an adaptive prediction filter. The adaptive method provides high compression ratios especially for images that contain sharp edges, such as subtitles, graphics, or text. The ringing effect which is a commonly encountered artifact in wavelet or subband based coders, is eliminated with the adaptive scheme.

In Chapter 3, the compression of textual images will be considered. High compression is, again, the main goal for textual images, However, most of the querying data for a digitized library is contained in the printed pages, so the compression methods are designed to be suitable for database search and multiresolution viewing, as well. An efficient image coding for the documents should satisfy the constraints on

- high compression ratio,
- fast decoding capability,
- quick keyword search, and
- quick preview capability, simultaneously.

A binary textual image compression method based on a multiresolution decomposition followed by a textual image coding (TIC) method which was previously proposed by Witten *et al.* [60] is developed. The multiresolution property plus the organization of the compressed bit stream provides efficient keyword search. The chapter starts with presenting some examples of different binary image compression techniques, and then elaborates on the compression methods which are mainly developed for textual images. The developed multiresolution TIC method is obtained by applying a binary subband decomposition, and constructing a symbol library consisting of character images which represents the repeated character images in a text image. The binary subband decomposition structures are developed using modulo-2 subband filter banks, and they are observed to have nice subband properties similar to those of real subband decomposition filter banks. In the extraction of the character images to form the symbol library, classical template matching methods, as well as a neural network approach are presented. Fast keyword searching property is obtained by optimum organization of character symbols in the symbol library. The organizing strategy is presented in Chapter 3 and Appendix A.

Two examples of specialized libraries are presented in Chapter 4. In this chapter, the methods presented in Chapters 2 and 3 are modified to illustrate the use of these methods in specialized libraries. The two examples of libraries consist of document images composed of historical Ottoman archives and the fingerprint image documents. The historical documents contain connected Arabic script which requires special treatment. The differences between the discrete character textual images and the Ottoman script images are indicated and the textual image compression method of Chapter 3 is modified. The fingerprint image database is another important application. The criminal databases contain huge amounts of fingerprint images. We present a method based on the adaptive subband decomposition structure of Chapter 2 for the compression of gray tone and binary fingerprint images.

In Chapter 5, the compression methods proposed for different types of images and their compression performances together with the keyword search properties are discussed in a unified framework. Possible extensions and/or changes in different applications are investigated.

Chapter 2

COMPRESSION OF IMAGES USING ADAPTIVE SUBBAND DECOMPOSITION

2.1 INTRODUCTION

The proposed interface for the digital document library is similar to the real library consisting of books, magazines and videos. The user should reach the desired information through a hierarchical interface starting from the images of the shelves and ending at the images of the pages inside a book. In all these images, different types of graphics, textual fonts, or pictures can be encountered. In this chapter, the compression methods developed for continuous tone images, i.e. gray-tone or color images, are presented. Since most of the pictures inside the books or magazines are interfered with graphical figures or text, special interest have been paid on such images which contain sharp edges. The developed methods are found out to perform better than the other compression methods presented so far.

The image compression algorithm presented in this chapter is based on an adaptive subband decomposition scheme. Subband decomposition is widely used in signal processing applications including speech, image and video compression [21]. In most practical cases, the goal is to obtain subband signals corresponding to different spectral regions of the original signal. The frequency content of some audio and visual data are suitable for this kind of frequency selective coding. However, this approach leads to ringing artifacts in image and video signals containing text, subtitles or other sharp edges. Furthermore, it is usually not possible to find a filter bank which is optimum in the sense of compressing image portions whose characteristics change inside the image. The ringing artifact is mainly due to constant analysis filter banks which cannot cope with the sudden changes in the input signal.

In this chapter, Perfect Reconstruction (PR) polyphase filter bank [22], [23] structures in which the analysis and synthesis filters adapt to the changing input conditions [24], [25] are presented. As a result of the adaptation of filter banks, the subband signals have smaller variances which leads to higher compression results for gray tone images, and the relative compression performance improves for images that contain sharp edges, text, and subtitles. Since most of the disturbing ringing artifacts occur on the boundaries of sharp edges, an adaptive filter bank can update its coefficients accordingly and can eliminate the disturbing overshoots at the edges. Furthermore, most images and video signals consist of regions which are separated from each other by their probability distribution function (pdf) or texture characteristics, therefore an adaptive filter bank can achieve higher efficiency by adapting the analysis and synthesis filters for different regions. As a result of these observations, polyphase filter bank structures with PR property which allow the use of Least Squares (LS) type FIR and nonlinear order statistics based adaptive filters are presented.

The concepts of adaptive filtering and subband decomposition have been previously used together by a number of researchers [26]- [29]. Most of the proposed adaptation algorithms for subband decomposition filter banks [30] consider the problems of system identification and noise removal [26]- [29]. The system identification and noise removal problems are the main issues of adaptive filtering, and some researchers attack these problems inside the subband domain, i.e. they first subband decompose the desired signal and the

input signal with subband filters, and then perform adaptation in each subband component of the desired and input signals. In other words, the adaptive filtering problem is considered inside the subband filter bank. There are other studies with more similar motivations to our method in the literature, as well. In one of those studies, the unknown system outputs are used for adapting the analysis filter bank coefficients so that the filter bank approximates the unknown system [31], [32]. This is a filter bank adaptation scheme, which is similar to our work, in some sense. However, in that work, the adaptation system does not consider the adaptation of the synthesis filter bank which provides the reconstruction. Therefore, that adaptation scheme is not suitable for coding purposes.

The concept of signal adapted filter banks for coding is also considered by researchers [33]- [39]. However, the main goal of these works is usually to find the best wavelet basis corresponding to a specific choice of data. For example, the autocorrelation matrix of the image data is used for determining a good basis for decomposition in [33]- [36]. In these studies, the basis selection scheme is applied at each decomposition layer, and this concept was considered as an adaptation of subband decomposition filter banks. In [32], [38], optimal coding after decomposition is considered, and optimum quantizers and optimum entropy coders are studied. In [39], a polyphase lifting structure is used, however the structure was still for determining a fixed best prediction filter bank at each scale.

In [33]- [37], fixed filters chosen according to an optimality criterion are used throughout the entire duration or extent of the signal, whereas in our work, the filters vary as the nature of the input changes. The problem addressed here is the coding of the input data. The adaptation scheme in our method neither tries to estimate an unknown system nor uses a fixed filter bank throughout the entire duration of the signal. The filter coefficients are updated to remove the unnecessary information among the neighboring subsignal coefficients. Due to the non-stationary characteristics of most image data, this improves the coding efficiency. In this aspect, the work in [45] is related with our work. In [45], the previously determined linear and nonlinear filters were used in a switchable manner in different regions of the image. In our work, there is no need to select from pre-determined filters in different regions of the image because the

proposed adaptive algorithm inherently modifies the filter bank to the optimum filter bank while preserving the perfect reconstruction property.

In Section 2.2, the PR polyphase structure concept [22], [23] is reviewed. Specifically, the lifting stages are investigated [40]- [43] and a procedure is presented to make them adaptive. In this section, multichannel extensions of adaptive filter banks are also presented. As pointed above, either linear or nonlinear filters can be used in the decomposition structure without disturbing the PR property.

In Section 2.3, another adaptive polyphase structure which contains a fixed anti-aliasing filter for the upper branch and an adaptive prediction filter for the lower branch is described. This structure is especially useful when a multiresolution viewing feature is needed.

Simulation examples and image compression results are given in Section 2.5 and conclusions about the gray tone image compression are presented in Section 2.7.

2.2 ADAPTIVE PREDICTION FILTERS IN POLYPHASE FORM

The subband decomposition of a signal corresponds to a transformation of the input signal to a domain where one can utilize his signal processing applications more efficiently. A well known decomposition is the subband decomposition with a pair of low- and high-pass filters [21]- [23]. The two-band subband decomposition is shown in Fig. 2.1. For most practical purposes, this structure splits the frequency content of the input signal into two, and obtains two half size signals corresponding to low and high frequency regions.

Perfect reconstruction of the decomposed subband signals is a desired property for almost all applications. For the decomposition scheme shown in Fig. 2.1, the reconstruction stage is formed as in the right part of the same figure. The perfect reconstruction analysis-synthesis pair should satisfy the

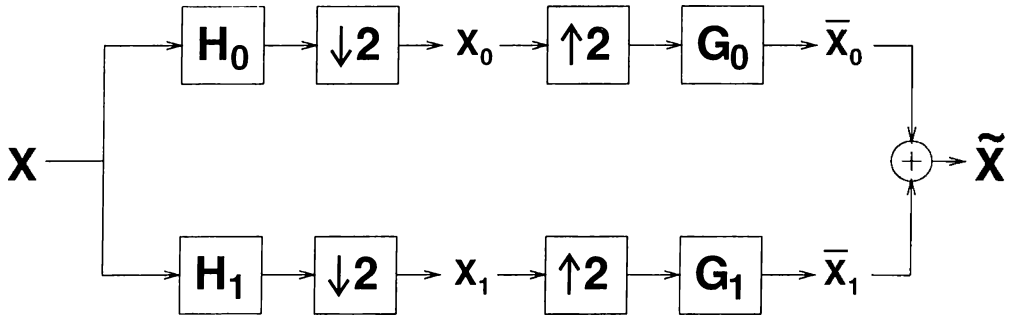


Figure 2.1: *QMF Subband analysis/synthesis.*

constraint of $\hat{x}(n)$ being equal to $x(n)$, probably within a shift. This equations for satisfying this constraint can be obtained as follows

$$X_0(z) = \frac{1}{2}\left[X(z^{1/2})H_0(z^{1/2}) + X(-z^{1/2})H_0(-z^{1/2})\right] \quad (2.1)$$

$$X_1(z) = \frac{1}{2}\left[X(z^{1/2})H_1(z^{1/2}) + X(-z^{1/2})H_1(-z^{1/2})\right] \quad (2.2)$$

$$\overline{X_0} = \frac{1}{2}\left[X_0(z^2)G_0(z)\right] \quad (2.3)$$

$$= \frac{1}{2}\left[X(z)H_0(z)G_0(z) + X(-z)H_0(-z)G_0(z)\right] \quad (2.4)$$

$$\overline{X_1} = \frac{1}{2}\left[X_1(z^2)G_1(z)\right] \quad (2.5)$$

$$= \frac{1}{2}\left[X(z)H_1(z)G_1(z) + X(-z)H_1(-z)G_1(z)\right] \quad (2.6)$$

$$\begin{aligned} \Rightarrow \hat{X}(z) = &\tfrac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right]X(z) \\ &+\tfrac{1}{2}\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right]X(-z) \end{aligned} \quad (2.7)$$

In Eq. 2.7, the first term filters the original signal $X(z)$ and the next term filters the alias term $X(-z)$. Ideally, we want $\hat{X}(z) = z^{n_0}X(z)$. This requires the two equations

$$H_0(z)G_0(z) + H_1(z)G_1(z) = T(z) \quad (2.8)$$

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0 \quad (2.9)$$

The cancelation of the aliasing term can be satisfied with the Quadrature Mirror Filter (QMF) solution:

$$G_0(z) = H_0(z) \quad (2.10)$$

$$G_1(z) = -H_0(-z) \quad (2.11)$$

In the frequency domain, if we choose $H_1(\omega) = H_0(\omega - \pi)$, the transfer function can be obtained as:

$$T(\omega) = H_0^2(\omega) - H_0^2(\omega - \pi) \tag{2.12}$$

With careful design, $T(\omega)$ can be made close to a constant delay, that is, $|H_0^2(\omega) - H_0^2(\omega - \pi)| = 1$.

Another approach for the design of filter banks was proposed by Smith and Barnwell [44]. They had designed a set of exact reconstruction filters specified by the H_0 filter as:

$$G_0(z) = H_0(z^{-1}) \tag{2.13}$$

$$G_1(z) = H_0(-z) \tag{2.14}$$

$$H_1(z) = H_0(-z^{-1}) \tag{2.15}$$

This structure has limited design flexibility. However, since most of the real life signals are suitable for frequency separation type analysis, this structure has been widely used in many applications [21].

Another decomposition structure is called the polyphase subband decomposition [22], [23]. The polyphase structure is more suitable for designing arbitrary decomposition filter banks. The block diagram of the basic 2-band polyphase subband structure is shown in Fig. 2.2. In this structure, the input

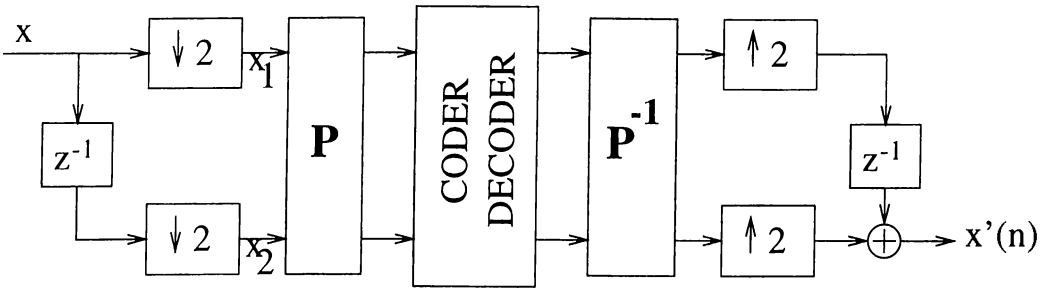


Figure 2.2: *Polyphase decomposition*

polyphase components x_1 and x_2 are multiplied by a 2×2 matrix, $\mathbf{P}$.

For perfect reconstruction, the only constraint on this matrix is invertability. One can try to optimize the $\mathbf{P}$ matrix according to the application without considering the frequency band decomposition. In many cases, the matrix

elements can even perform nonlinear operations on the input data without disturbing the perfect reconstruction property. In the next subsection, a class of polyphase structures in which the **P** matrix is not fixed is introduced. A description about how the filters that form **P** can be chosen is given.

2.2.1 THE BASIC FILTER BANK STRUCTURE WITH A LIFT STAGE

Consider the polyphase filter bank structure shown in Fig. 2.3. This structure has a simple transform matrix:

$$\mathbf{P} = \begin{bmatrix} 1 & -P_1(.) \\ 0 & 1 \end{bmatrix} \tag{2.16}$$

where P_1 is an operator that generates a single output with the input elements

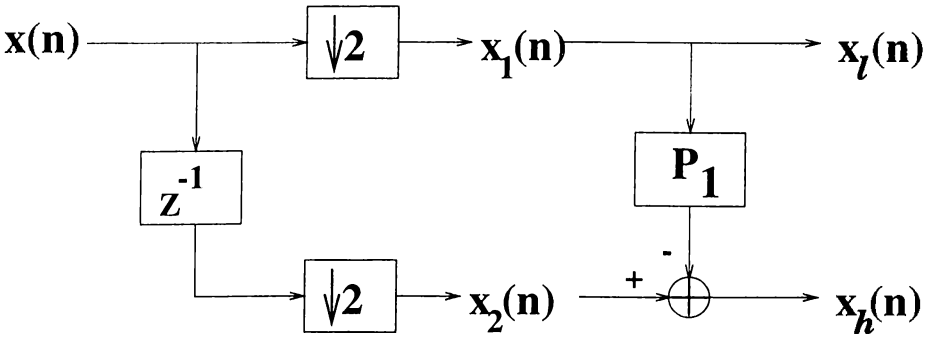


Figure 2.3: *Simple structure analysis stage*

from the signal $x_l(n)$. This structure can be considered as a special case of a two band polyphase decomposition scheme. In Fig. 2.3, the filter P_1 need neither be a fixed nor a linear operator for perfect reconstruction as **P** is invertible regardless of the nature of P_1. Furthermore, the PR property is preserved as **P** is invertible at all time instants. Therefore, both non-linear filters and time varying filters can be used in this structure. This stage can also be considered as the prediction portion of a lift stage [40]- [43].

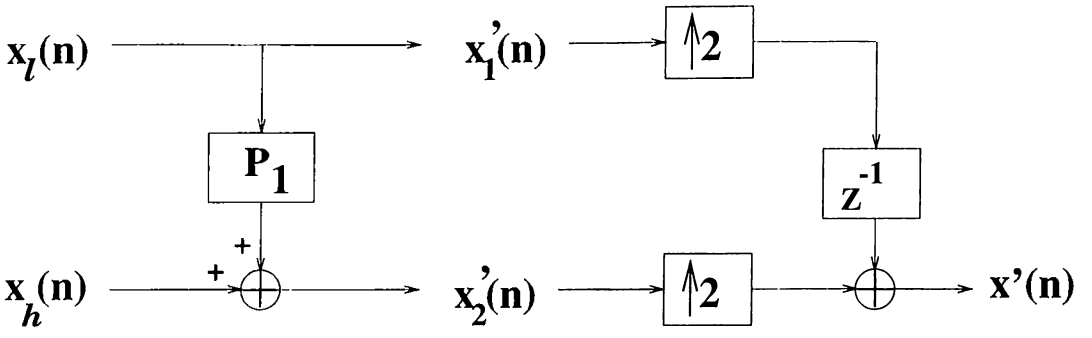


Figure 2.4: *Simple structure synthesis stage*

The inverse of the $\mathbf{P}$ matrix in Eq. 2.16 is given as:

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 & P_1(.) \\ 0 & 1 \end{bmatrix} \tag{2.17}$$

The resulting synthesis structure corresponding to the synthesis matrix is shown in Fig. 2.4 [46].

In this case, the low-band signal x_l is obtained by down-sampling the original signal, x and it is directly passed to the encoder. Therefore, a good way of obtaining the subsignal, $x_h(n)$, is to predict the samples of the second polyphase component x_2 from the first polyphase component x_1 which is equal to x_l. For many signals, the polyphase components x_1 and x_2 are strongly correlated at near time indices. Therefore, the elements in x_2 can be efficiently predicted by the values of x_1. This approach is suitable for coding applications, in which the goal is to remove the predictable portion of the original signal as much as possible. In this way, the correlation between the channels is eliminated.

Usually, the prediction filters are of low pass nature, because the samples of the input signal has a low pass nature within a neighborhood of time indices. For some specific signal types, a good prediction filter can be put in this filter bank, and the decomposition can be performed.

However, a great portion of the images contain parts with different statistical characteristics. A predictor should be adaptive for such image and video signals which are nonstationary in nature. This reasoning leads to the polyphase structure shown in Fig. 2.5 in which the prediction filter adapts itself

to minimize the high-band signal $x_h(n)$. This is especially useful when there are sharp transition regions in an image such as subtitles, text and graphics.

2.2.2 THE ADAPTIVE FILTER BANK STRUCTURE

The adaptive estimator for $x_h(n)$ is shown in Fig. 2.5. This structure retains the important property of perfect reconstruction without transmitting any side information to the synthesis stage.

The reconstruction filter bank is shown in Fig. 2.6. Perfect reconstruction

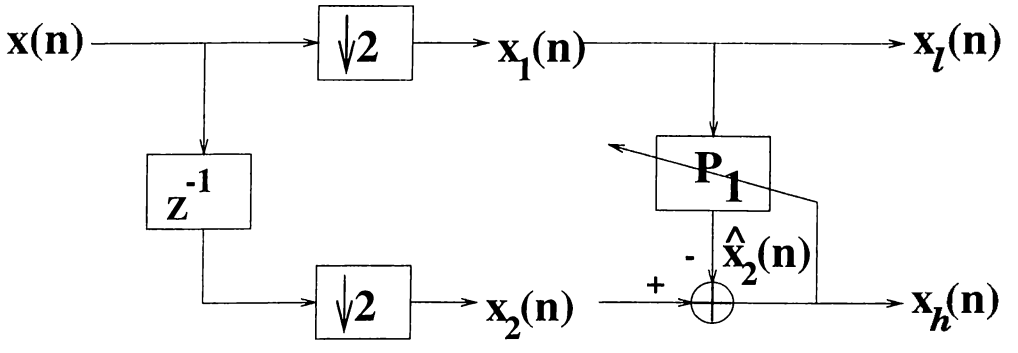


Figure 2.5: *Adaptive structure analysis stage*

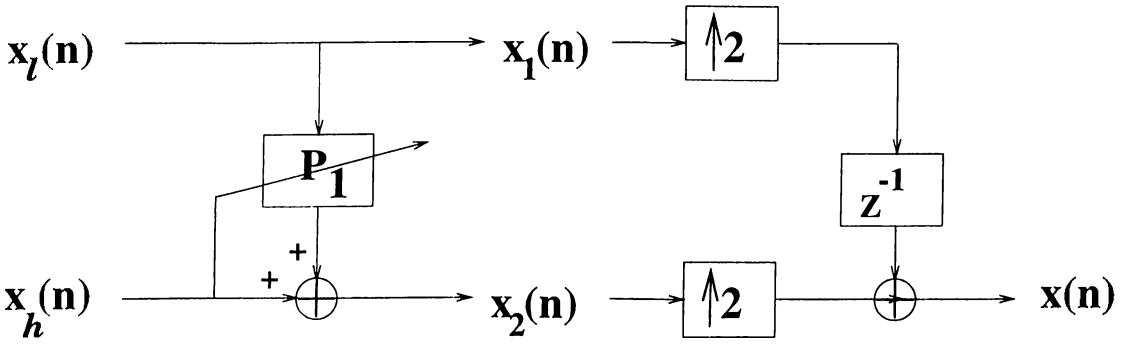


Figure 2.6: *Adaptive structure synthesis stage*

without sending any side information can be better understood by observing that the adaptation parameters for updating the prediction filter in the analysis stage are readily available in the synthesis stage, as well. As a result, the same adaptation algorithm is used at both analysis and synthesis stages, and the

prediction filters at both stages attain the same filter tap values at each time instance.

Various adaptation schemes are considered in this work. The linear FIR estimator is found to perform good for the sample images that have been considered. The FIR estimator is obtained by predicting $x_2(n)$ from $x_1(n)$ in a Linear Minimum Mean Squared Error (LMMS) sense as follows:

$$\hat{x}_2(n) = \sum_{k=-N}^{N} w_{n,k} x_1(n-k) = \sum_{k=-N}^{M} w_{n,k} x(2n-2k) \tag{2.18}$$

where the filter coefficients $w_{n,k}$'s are updated using an LMS-type or RLS type algorithms [47],and the subsignal x_h is given by

$$x_h(n) = x_2(n) - \hat{x}_2(n). \tag{2.19}$$

In our simulations, we first use the LMS type gradient estimators for linear FIR filters as well as for order statistics filters in adaptation. The FIR LMS adaptation is performed in a conventional manner. An estimate of the gradient vector

$$\nabla(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n) \tag{2.20}$$

is calculated at each step as

$$\hat{\nabla}(n) = -2\tilde{\mathbf{x}}_n x_2(n) + 2\tilde{\mathbf{x}}_n \tilde{\mathbf{x}}_n^T \hat{\mathbf{w}}(n) \tag{2.21}$$

where

$$\tilde{\mathbf{x}}_n = [x_1(n-N), x_1(n-N+1), \cdots, x_1(n+N-1), x_1(n+N)]^T, \tag{2.22}$$

and the filter updates are performed as

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu \frac{\tilde{\mathbf{x}}_n e(n)}{\|\tilde{\mathbf{x}}_n\|^2} \tag{2.23}$$

where $\hat{\mathbf{w}}(n) = [w_{n,-N}, \cdots, w_{n,N}]$ is the weight vector at time instant n. The subsignal x_h is given by

$$x_h(n) = x_2(n) - \hat{x}_2(n). \tag{2.24}$$

and

$$e(n) = x_h(n) = x_2(n) - \tilde{\mathbf{x}}_n^T(n)\hat{\mathbf{w}}(n). \tag{2.25}$$

Both $\mathcal{L}^1$ and $\mathcal{L}^2$ norms can be used in normalizing the update equations depending on the characteristics of the signal [47]. In our simulations, these norms are successfully used [47].

The scalar μ determines the step size of the adaptive algorithm. It is well known that the convergence speed of adaptation is low when μ is small, but the steady state error is smaller. For large values of μ, the opposite happens and the convergence speed increases with a higher steady state error. There are various methods to change the value of μ during adaptation in the LMS algorithm [49], [50]. Usually, the value of μ can be set to a large number between 1 and 2 in the beginning and, it can be gradually decreased to a smaller value between 0 and 1. In our case, the value of μ is altered according to the range of the input. Since the input data $\tilde{\mathbf{x}}_1$ is available at the decoder side, the decoder can alter the parameter of its μ value for reconstruction, accordingly. Finally, the actual update equation is given by,

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu(\tilde{\mathbf{x}}_n)\frac{\tilde{\mathbf{x}}_n e(n)}{\|\tilde{\mathbf{x}}_n\|^2} \tag{2.26}$$

where the $\mu(\tilde{\mathbf{x}}_1)$ function is experimentally set to

$$\mu(\tilde{\mathbf{x}}_1) = \begin{cases} 0.4, & \Delta_{\tilde{\mathbf{x}}} < 10 \\ 0.6, & 10 \leq \Delta_{\tilde{\mathbf{x}}} < 30 \\ 0.8, & 30 \leq \Delta_{\tilde{\mathbf{x}}} < 80 \\ 1.0, & 80 \leq \Delta_{\tilde{\mathbf{x}}} < 200 \\ 1.2, & 200 \leq \Delta_{\tilde{\mathbf{x}}} < 256 \end{cases} \tag{2.27}$$

and

$$\Delta_{\tilde{\mathbf{x}}} = max(\tilde{\mathbf{x}}_n) - min(\tilde{\mathbf{x}}_n) \tag{2.28}$$

Many simulations should be made for finding the best μ values over a large number test images with various subjective evaluation strategies. However, these values of the μ parameter are experimentally observed to perform good for the test images in our simulations, so more elaboration on fine tuning will be beyond the scope of this thesis.

In order to avoid extreme overshoots in filter tap updates, thresholds are also used [51] both in the encoder and in the decoder. The reason to put such

thresholds is to avoid divergence at very low bit rates which require coarse quantization of the transform data. In our simulation studies, a limiting threshold of -256 and 256 for each filter tap for image coding applications is used.

The PR property is preserved in this structure as long as the same adaptation algorithm is used at the encoding and the decoding stage. Since the subsignal $x_h(n)$ as well as $\tilde{\mathbf{x}}_n$ are available both at the encoder and at the decoder, the synthesis stage can adapt the filter P_1 with the same filter tap coefficients $\hat{\mathbf{w}}(n)$. Therefore, no side information needs to be transmitted.

We also used Recursive Least Squares (RLS) type adaptation in update equations, as well. In RLS type adaptation, the weighted sum of magnitude-squared errors between the desired and estimated signals is used as the minimization criterion.

$$\mathcal{E}_M = \sum_{l=0}^{n} w^{n-l} \|e_M(l,n)\|^2 \tag{2.29}$$

where the error is the difference between the desired signal and its estimate

$$e_M(l,n) = d(l) - \mathbf{h}_M^t(n)\mathbf{x}_M(l) \tag{2.30}$$

and w is a weighing factor between 0 and 1. The minimization of $\mathcal{E}_M$ with respect to the filter coefficient vector $\mathbf{h}_M^t(n)$ gives the set of linear equations

$$\mathbf{R}_M(n)\mathbf{h}_M^t(n) = \mathbf{D}_M(n) \tag{2.31}$$

where $bfR_M(n)$ is the estimated signal correlation matrix

$$\mathbf{R}_M(n) = \sum_{l=0}^{n} w^{n-l}\mathbf{X}_M^*(l)\mathbf{X}_M^t(l) \tag{2.32}$$

and $\mathbf{D}_M(n)$ is the estimated cross correlation vector

$$\mathbf{D}_M(n) = \sum_{l=0}^{n} w^{n-l}\mathbf{X}_M^*(l)d(l) \tag{2.33}$$

The solution of Eq. 2.31 is

$$\mathbf{h}_M^t(n) = \mathbf{R}_M^{-1}(n)\mathbf{D}_M(n) \tag{2.34}$$

The RLS algorithm solves this matrix inversion in a recursive manner [24]. However, the equations in RLS formulation requires causal filters. In order

to obtain non-causal filter supports, a delay can be applied to the second polyphase component, which is the desired signal, in the analysis stage. For perfect reconstruction, the same amount of delay should be put to the first polyphase component just before upsampling at the synthesis stage. In our simulations, a delay of two is used in the filter banks. This delay corresponds to the input vector regions of support described in Sec. 2.4.

It was observed in [45], [46] that, in coding applications, the Order Statistics (OS) filters and especially the median filter perform better than the linear FIR filters for the images containing sharp variations like text [45]. This observation motivates the use of adaptive OS filters in the structures shown in Figures 2.5 and 2.6. The rank ordering of the input elements produces better coding results especially for the images that contain sharp edges.

The implementation of the Order Statistics (OS) type adaptation [52] is similar to the linear FIR filter coefficient update. Actually, the OS adaptive filter bank still uses the linear LS type adaptation, but this time on a rank ordered and modified version of the input sequence. Specifically, for the OS case, the input vector $\tilde{\mathbf{x}}_n$ is first rank ordered from the largest to the smallest elements. The largest and the smallest values of the vector are removed from the list. As a result, another ordered vector with a shorter size is obtained. This vector is then used as an input to the update Equations (2.25) and (2.26) for adapting the filter coefficients. In our simulation studies, a region of support with 9 elements is used. After rank ordering and eliminating the largest and smallest elements, only 7 elements are left. These 7 elements are used as an input vector to the LS algorithms.

2.2.3 THE CODING ALGORITHM

Using the two channel adaptive decomposition filter banks, the overall coder for gray tone images can be summarized as follows:

- The M×N image is read to the memory to form a matrix with M columns and N rows.

- The number of decompositions for horizontal direction is $n_h = \lfloor log_2 M - 3 \rfloor$ and for vertical direction is $n_v = \lfloor log_2 N - 3 \rfloor$ where $\lfloor \rfloor$ operation indicates a downward truncation. Select the minimum of n_h and n_v as the number of decomposition.

- In the horizontal decomposition, each row of the matrix is processed by the one dimensional adaptive subband decompostion structure shown in Fig. 2.5. Assume the row is a 1×M vector, called **X**. For each row, the processing is performed as follows:(*)

  - **X1** = downsample(**X**);
  - **X2** = downsample(delay(**X**,1));
  - Feed the **X1** vector as the input signal (see Eq. 2.22, and the **X2** vector as the desired signal to the LMS algorithm.
    * Start with initial filter [0 0 0 1/2 1/2 0 0 0]
    * Use the filter update equation (Eq. 2.23)
    * Calculate the error sequence using Eq. 2.25
  - Assign X1 as the low band signal,**XL**, the error sequence as the high band signal, **XH**.

- Obtain an $M/2 \times N$ low band image, and an $M/2 \times N$ high band image.

- Apply the above processing steps (*) to the columns ofthe low band and high band images.

- Obtain an $M/2 \times N/2$ low-low image, $M/2 \times N/2$ low-high image, $M/2 \times N/2$ high-low image, $M/2 \times N/2$ high-high image.

- Apply all the above decomposition to the low-low image, and proceed until the number of decompositions is reached.

- Obtain a pyramid of subband images (Fig 2.16).

- Apply the Zerotree Coder [56] to the pyramid:

  - Find the maximum subband value in the pyramid, assign to T_0.
  - $T_1 = T_0/2$ is the threshold.

- The trees and their descendents are shown in Fig 2.16. For each element in the subbands:
  - Is its absolute value more than T_1?
  - If yes
    - Code its sign: Positove Symbol or Negative Symbol.
  - If no
    - Is it a descendent of a zerotree root?
      - If yes, code nothing
      - If no, does it have descendents with absolute value larger than T_1?
        - If yes, code as an Isolated Zero symbol.
        - If no, code as a Zerotree Root.
- For subband locations with positive or negative symbols, take the new interval between T_1 and $2 \times T_1$, reduce the threshold to half: $T_1/2$, and apply the above zerotree coding algorithm.
- For subband locations with positive or isolated zero or zerotree root symbols, take the new interval between 0 and T_1, reduce the threshold to half: $T_1/2$, and apply the above zerotree coding algorithm.
- Continue until the amount required for the coded bit-stream is full.

- The coded bit stream consists of four distinct symbols: Positive, Negative, Isolated Zero, Zerotree Root [56].

The zerotree coder produces quantized values for the subband data. In the decoding stage, these values are fed to the synthesis structure (Fig 2.6 and the reconstructed image is obtained.

The adaptation stage of the decomposition structure can have various methods for adaptively predicting the values of x_2 from x_1. For the case of RLS type adaptation, the updates require solving Eq. 2.34 in a recursive manner and obtain the error sequence as given in Eq. 2.30. For the case of order statistics adaptation, the input vector in Eq 2.22 is sorted and its maximum and minimum values are eliminated to produce a shorter input vector for the LMS algorithm. This vector is then fed to the linear LMS adaptation function. In

practice, any adaptive algorithm which uses $\tilde{\mathbf{x}}_n$ as the input and produces an error sequence can be used in the adaptive subband decomposition analysis stage, as long as the same adaptive algorithm is used in the synthesis stage for reconstruction.

2.2.4 CASCADED ADAPTIVE PR BLOCKS

The structure described in Section 2.2.2 can be generalized by cascading matrices similar to the matrix in Eq. 2.16.

The analysis and synthesis stages of the cascaded two band decomposition structure can be generated using Equations (2.16) and (2.17). The overall cascaded transformation matrix is obtained by multiplying triangular matrices which correspond to basic building blocks as follows:

$$\mathbf{P} = \begin{bmatrix} 1 & -P_1(.) \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ G_1(.) & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -P_2(.) \\ 0 & 1 \end{bmatrix} \times \cdots \tag{2.35}$$

where the filters $P_1, G_1, P_2, \cdots$ can be linear, nonlinear or adaptive. In this way, the upper and lower branch subsignals can be filtered a number of times.

The inverse matrix is given as

$$\mathbf{P^{-1}} = \cdots \times \begin{bmatrix} 1 & P_2(.) \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ -G_1(.) & 1 \end{bmatrix} \times \begin{bmatrix} 1 & P_1(.) \\ 0 & 1 \end{bmatrix} \tag{2.36}$$

The overall scheme is illustrated in Fig. 2.7. The synthesis filter bank corresponding to the synthesis matrix $\mathbf{P^{-1}}$ can be easily constructed as shown in Fig. 2.7.

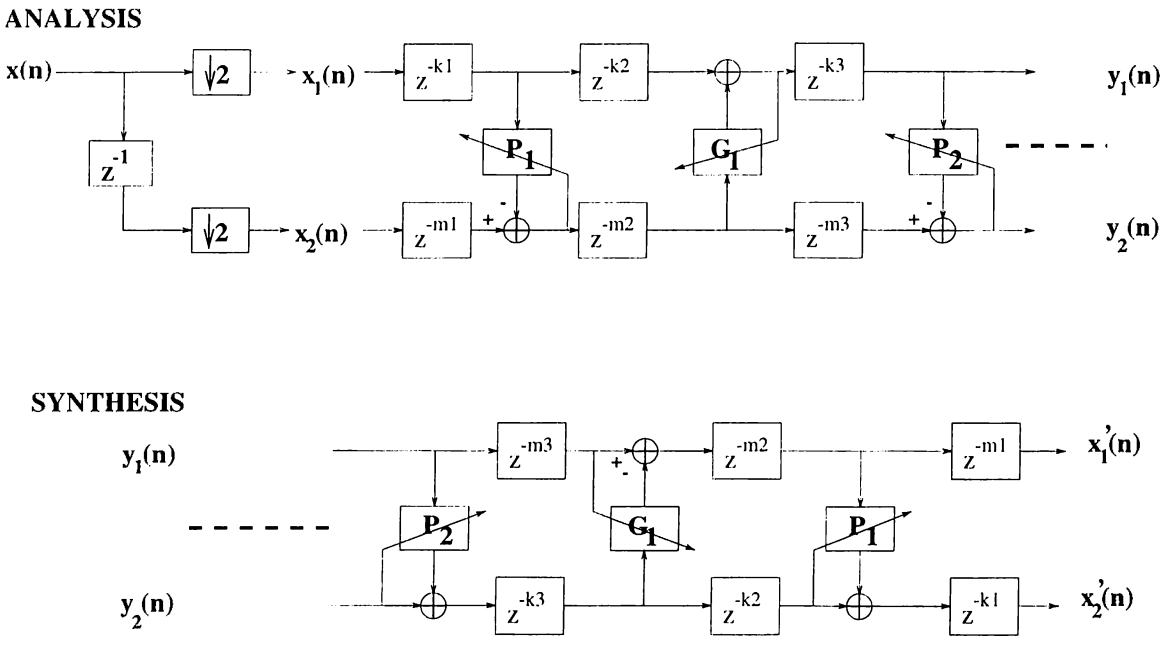


Figure 2.7: *Cascaded polyphase filters*

2.2.5 MULTICHANNEL EXTENSION OF THE PR STRUCTURE

For most practical purposes, a tree structured subband decomposition obtained by recursively decomposing the sub-signals of a two band decomposition output is a good way of obtaining scale-space representations. In this way, logarithmic or balanced trees of n subsignals where n can be powers of two. However, it may also be required to have arbitrary number of subband signals for a general decomposition.

In a generalized frame work, the filter bank structures described in Section 2.2.2 can also be extended to handle decompositions to subbands other than the powers of two. This extension can be performed in various ways. Consider the multiband decomposition structure shown in Fig. 2.8.

In this figure, an M band decomposition with two cascaded PR building blocks is illustrated. The PR property of this structure can be proved easily.

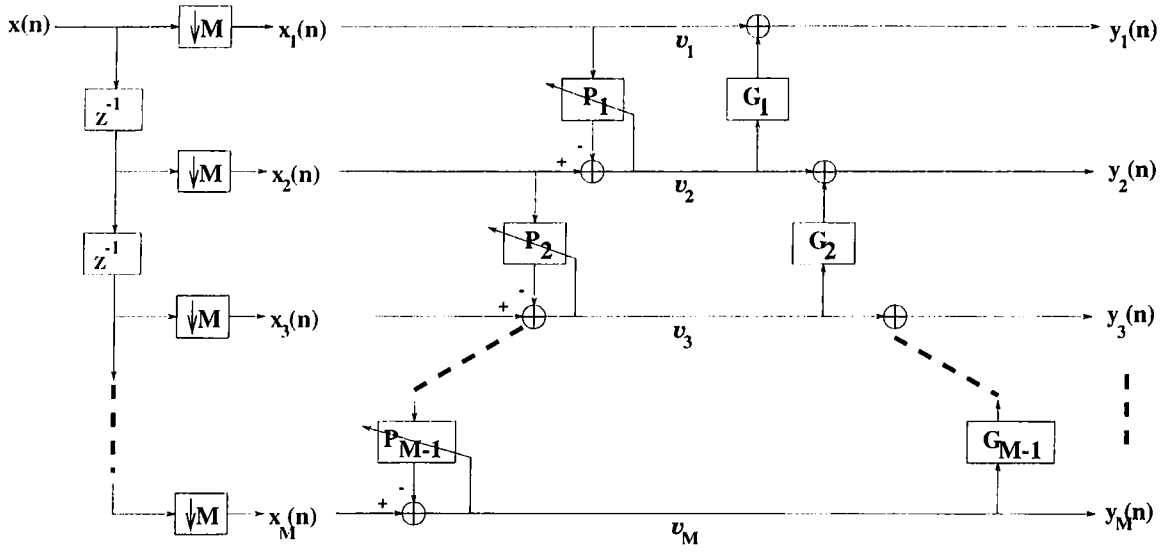


Figure 2.8: *Multi-band analysis structure - 1*

In the analysis stage,

$$\begin{aligned} v_1 &= x_1 \\ v_i &= x_i - P_{i-1}(v_{i-1}), \quad i = 2, 3, ..., M \\ y_i &= v_i + G_i(v_{i+1}), \quad i = 1, 2, ..., M-1 \\ y_M &= v_M \end{aligned} \tag{2.37}$$

The corresponding $\mathbf{P}$ matrix for this case is given by :

$$\mathbf{P} = \begin{bmatrix} 1 & -P_1 & 0 & 0 & \cdots \\ 0 & 1 & -P_2 & 0 & \cdots \\ 0 & 0 & 1 & -P_3 & \cdots \\ \vdots & & & & \ddots \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ G_1 & 1 & 0 & 0 & \cdots \\ 0 & G_2 & 1 & 0 & \cdots \\ & & & & \ddots \end{bmatrix} \tag{2.38}$$

Since the matrix $\mathbf{P}$ is formed by multiplying upper and a lower triangular matrices, it can be inverted regardless of the filters P_i's and G_i's. Therefore, PR can be achieved with any choice of the nonlinear operators. This leads to

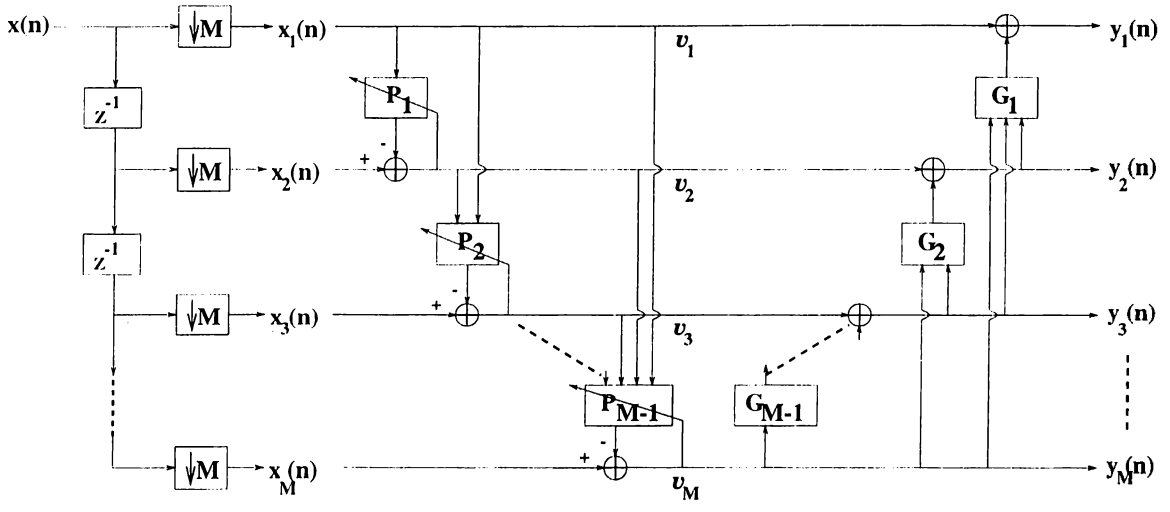


Figure 2.9: *Multi-band analysis structure - 2*

the following synthesis equations:

$$\begin{aligned} v'_M &= v_M \\ v'_i &= y_i - G_i(v'_{i+1}) = v_i, \quad i = M-1, ..., 1 \\ v'_1 &= v_1 = x_1 \\ x'_i &= v'_i + P_{i-1}(v'_{i-1}) = v_i + P_{i-1}(v_{i-1}) = x_i, \quad i = 2, ..., M \end{aligned} \tag{2.39}$$

The outputs, x'_i, of the synthesis filters are the same as the polyphase components, x_i, of the analysis filter bank. This implementation of the multichannel adaptive filter bank exploits the redundancy between two consecutive polyphase components.

Another multichannel extension structure is shown in Fig. 2.9. In the previous structure only the samples of x_k are considered to estimate x_{k+1}, $k = 2, 3, .., M$. On the other hand, the structure in Fig. 2.9 uses all of the previous polyphase components for prediction as the index of the subsignals increase. In this way, the redundancy between each polyphase component and all the components with smaller subband index is eliminated. The analysis equations for this structure are given as follows:

$$
\begin{aligned}
v_1 &= x_1 \\
v_i &= x_i - P_{i-1}(v_1, v_2, ..., v_{i-1}) \\
y_M &= v_M \\
y_i &= v_i + G_i(y_M, ..., y_{i+1}), \quad i = 1, 2, ..., M-1
\end{aligned}
\tag{2.40}
$$

The synthesis equations are given by:

$$
\begin{aligned}
v_i' &= y_i - G_i(y_M, ..., y_{i+1}) &&= v_i, \quad i = 1, 2, ..., M-1 \\
v_M' &= y_M &&= v_M \\
x_1' &= v_1 &&= x_1 \\
x_i' &= v_i + P_{i-1}(v_1, v_2, ..., v_{i-1}) &&= x_i, \quad i = 2, 3, ..., M
\end{aligned}
\tag{2.41}
$$

This later structure also yields analysis matrices which can be decomposed to upper and lower triangular matrices with elements containing P_i's and G_i's only. In this structure, for predicting v_i's, the number of data used increases with increasing index i. Conversely, more v_i samples are used for predicting y_i's when the index i is small. The computational complexity of this structure is high as compared to the structure in Fig. 2.8.

2.3 ADAPTIVE PR STRUCTURE WITH AN ANTI-ALIASING FILTER

In many applications, multiresolution display of an image is a desirable property. Since $x(n)$ is simply down-sampled in the upper branch of Fig. 2.5, the visual quality of the subsignal $x_1(n)$ is poor due to aliasing. This may be unwanted for multiresolution viewing applications.

In QMF type subband decomposition structures, the H_0 filter acts as a low pass anti aliasing pre-filter. A similar anti aliasing pre-filter can also be put in the adaptive structure. For this purpose, a two stage cascaded $\mathbf{P}$ matrix is

used. The matrix **P** should be designed in such a way that the first stage should reduce the aliasing and the second stage should produce a good "high-band" signal. In some sense, one of the polyphase components should correspond to a pre-filtered plus down-sampled version of the input signal.

If the low pass filter of a QMF filter bank is a half-band filter [22], [53], i.e, $H(z) = \frac{1}{2}[1 + z^{-1}A(z^2)]$, then the "noble identity" [22] can be used and the filtering operations can be carried out after down-sampling as shown in Fig. 2.10.

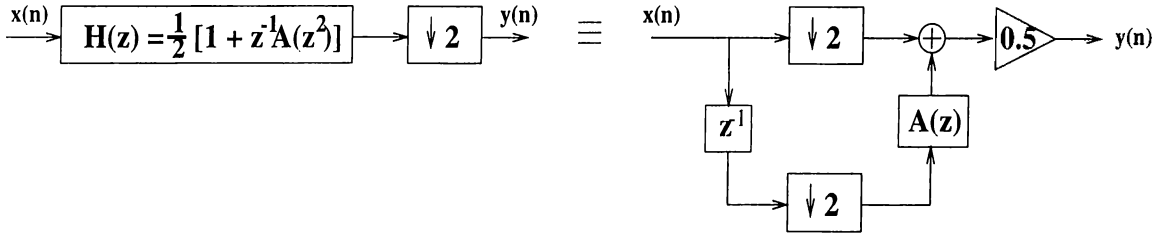


Figure 2.10: *Equivalent structures.*

The first stage of the analysis system is, therefore, a low pass filtering stage for $x_l(n)$. The second stage of the system consists of adaptive prediction of subsignal $x_h(n)$, as described before. In this case, the samples of the low pass filtered subsignal $x_l(n)$ are used to predict $x_h(n)$. The overall analysis structure is shown in Fig. 2.11. Due to the half band characteristics of the low pass filtering stage, perfect reconstruction can be achieved using the synthesis block shown in Fig. 2.12.

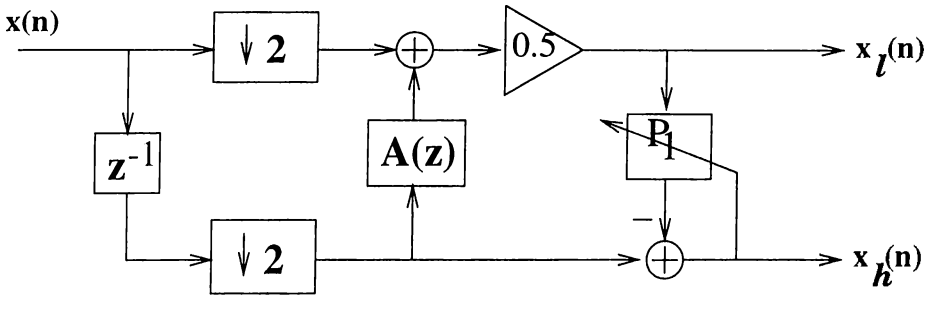


Figure 2.11: *Adaptive filter bank structure with an anti-aliasing filter.*

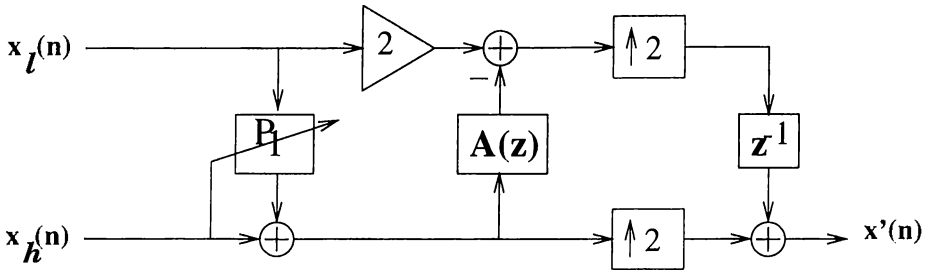


Figure 2.12: *Synthesis stage corresponding to Figure 2.11*

Perfect reconstruction can also be shown in matrix form. The analysis polyphase structure has the following matrix:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ \mathbf{A}(z) & 1 \end{bmatrix} \times \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -P_1(.) \\ 0 & 1 \end{bmatrix} \tag{2.42}$$

and the synthesis matrix is simply:

$$\mathbf{P^{-1}} = \begin{bmatrix} 1 & P_1(.) \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ -\mathbf{A}(z) & 1 \end{bmatrix} \tag{2.43}$$

In our simulation studies, we use the half-band Lagrange family for low pass filtering [53]. The first two Lagrange filters have the following impulse response:
$\mathbf{h_3} = \{1/4, 1/2, 1/4\}$, and
$\mathbf{h_7} = \{-1/32, 0, 9/32, 1/2, 9/32, 0, -1/32\}$.
In the first case, the $A(z)$ filter in the polyphase form becomes

$$A(z) = \frac{1}{2} + \frac{1}{2}z^1 \tag{2.44}$$

and in the second case,

$$A(z) = -\frac{1}{16}z^{-1} + \frac{9}{16} + \frac{9}{16}z^1 - \frac{1}{16}z^2. \tag{2.45}$$

By using these two filters, satisfactory low-low images are obtained.

2.4 TWO DIMENSIONAL FILTER BANK STRUCTURES

The extension of the adaptive structure to the two dimensional case is needed for image coding purposes. A straightforward two dimensional generalization can be achieved by applying one dimensional filters to the image data in a separable manner. In this way, first the columns of the image are filtered, then this data is row-wise processed. This is a conventional method to implement multi-dimensional filters with one dimensional modules.

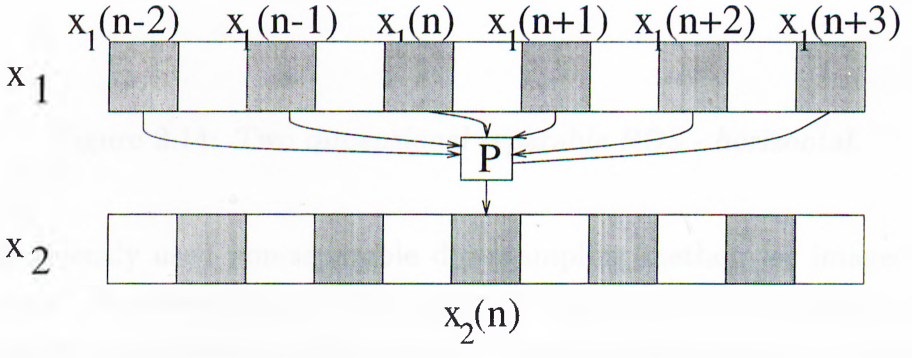


Figure 2.13: *One dimensional prediction.*

The prediction procedure in one dimensional filter bank structure is illustrated in Fig. 2.13 in which a symmetric filter support is assumed. In this figure, the input signal x is split into the polyphase components x_1 and x_2 which are represented by gray pixels in the upper and lower arrays, respectively. The pixel $x_2(n)$ which corresponds to an element in the polyphase component x_2 is predicted from the elements of the polyphase component x_1.

On the other hand, better prediction performance than consecutive one dimensional row-wise and column-wise processing can be achieved. Consider the region of support shown in Fig. 2.14 for horizontal processing. The gray pixel can be predicted from the black pixels using an adaptive algorithm. Since more samples are used in the support region, better prediction performance is achieved. Once the row-wise processing is finished, the column-wise adaptive filtering is carried out. In our simulation studies, the region of support in

Fig. 2.14 is used. It is also experimentally observed that this produces better coding results.

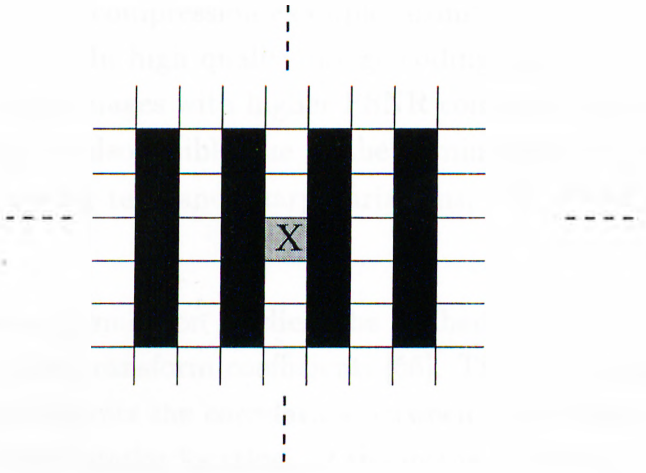


Figure 2.14: *Two dimensional separable ROS - horizontal.*

A frequently used non-separable downsampling method for images is the "quincunx" downsampling. The region of support of the prediction filter can readily be extended to the quincunx downsampling method as shown in Fig. 2.15. This decomposition method might be useful for some specific class of images. However, the coders used in our simulation studies were not optimized for this type of downsampling, therefore, they are not used in our experiments.

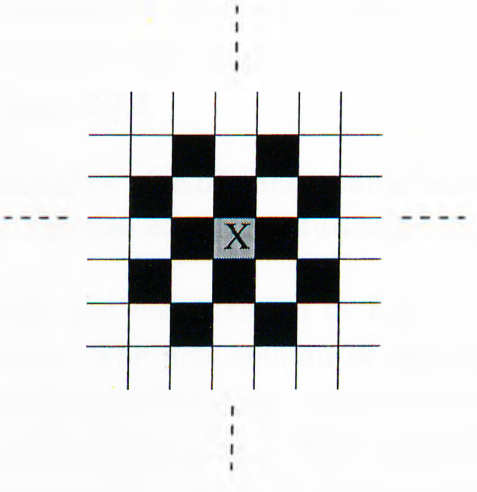


Figure 2.15: *Two dimensional quincunx prediction.*

2.5 SIMULATION STUDIES

In this section, image compression examples using the adaptive subband filter banks are presented. In high quality image coding applications, the adaptive filter bank produces images with higher PSNR compared to fixed filter banks. This improvement is also visible due to the elimination of the ringing effects. For images containing text and sharp variations, the PSNR improvement is higher.

In the following simulation studies, the Embedded ZeroTree (EZT) coder is used to encode the transform coefficients [56]. The EZT coder is an efficient lossy coder which exploits the correlation between the scales of decomposition corresponding to the similar locations of the image. A tree in a decomposition is shown in Fig. 2.16. The root of the tree at lower scales of decomposition corresponds to the location represented by a pixel in the higher scale image whith the descendents as shown in Fig. 2.16. The idea of the EZT coder is the assumption of the fact that if the value of a pixel in the higher scale is less than a threshold, the values in the nodes of all its descendents will probably be less than that threshold, too. As a result, with the quantization value corresponding to the threshold, an efficient representation for the whole tree is obtained. If the compression ratio set for the compressed bit stream allows more bits, the quantization is refined by halving the quantization level and forming another tree representation on top of the previous trees with coarse quantization. For example, the first quantization level obtains a binary quantization of the decomposed image, and the next level improves the quantization level to 2 bits, etc, as described in Section 2.2.3.

Due to the characteristics of EZT, the coding results in our simulation studies are obtained by the tree-structured two-band decompositions.

The coding results for the image shown in Fig. 2.17 at 1 bits/pixel bitrate is given in Table 2.1. The first column of the table shows the results without using the anti-aliasing filter stage, and the second column shows the results with the anti-aliasing filter stage. The Embedded Zerotree Wavelet (EZW) coder [56] with fixed filter banks of biorthogonal Barlaud filter [57], and orthogonal Coiflet filter [37] produces PSNRs of 36.10dB and 36.12dB,

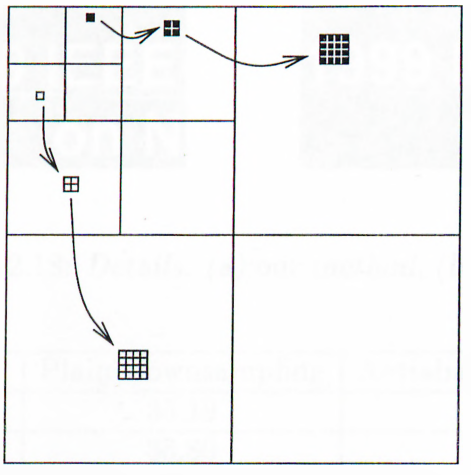


Figure 2.16: *Zerotrees in a decomposed image.*

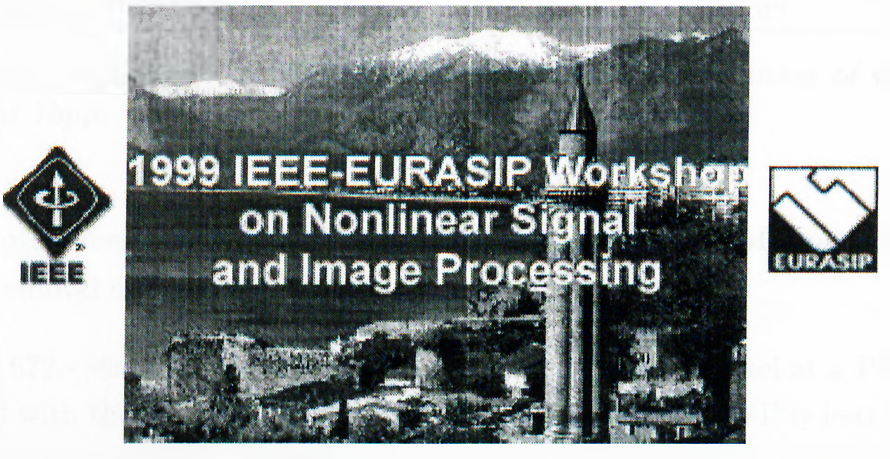


Figure 2.17: *Test image*

respectively. These PSNRs are 0.86dB less than the PSNR obtained using the adaptive decomposition method. In addition to the improved PSNR, the adaptive filter bank eliminates the ringing effects which are apparent in the EZW coder as shown in Fig. 2.18. Fig. 2.18(a) shows the enlarged detail of our encoder output, and Fig. 2.18(b) shows the EZW output of the same place.

A similar test is done over the compund graphics-text image in Fig 2.19, which is contained in the JPEG-2000 test images. This image contains various sharp transitions at the edges of graphics regions, therefore, the elimination of ringing artifacts at these portions is important. The adaptive OS compression of this image at 1bpp gives a PSNR of 38.51dB, whereas the EZW method at 1bpp gives a PSNR of 35.39dB. Furthermore, as it can be seen from Fig. 2.20,

(a) (b)

Figure 2.18: *Details. (a):our method, (b):EZW*

P_1 filter	Plain Downsampling	Antialiased Downsampling
Median	36.19	36.00
Adaptive FIR LMS	36.80	36.76
Adaptive FIR RLS	37.02	36.96
Adaptive OS LMS	36.96	36.90
Adaptive OS RLS	37.16	37.09

Table 2.1: *Experiment results (PSNR) for 5-level decomposition of the test image at 1bpp.*

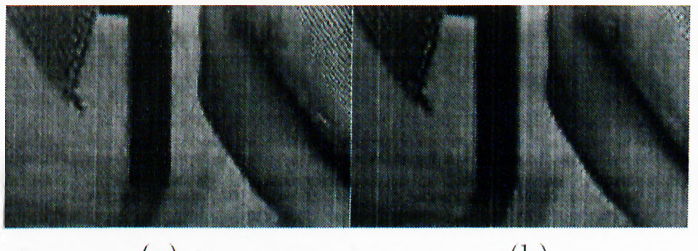
the adaptive method gives better visual performances at the edge portions due to the removal of ringing artifacts.

The 672×560 "barbara" image is compressed to 1 bits/pixel at a PSNR of 35.91dB with the adaptive OS type prediction filter. This PSNR is better than the conventional EZW compression scheme which produces 35.90dB PSNR. Consider the detail images shown in Fig. 2.21. In this case, both of the images are compressed at 0.4bpp (CR=20) to emphasize the ringing effects of the fixed filter bank. Although the PSNR of the image at left corresponding to EZW (18.42dB) is almost the same as that of the image at right (18.43dB), the details show that EZW with a fixed filter bank produces visually more disturbing ringing effects at the edges.

A set of 28 images is compressed using the adaptive subband coding scheme and the EZW with a fixed filter bank. The last 8 of these images are JPEG-2000 test images. In all cases, the adaptive method achieves higher PSNRs at 1 bpp. The coding results for these images are presented in Tables 2.2 and 2.3. The thumbnailed test images are shown in Figures 2.22, 2.23, and 2.24.

Figure 2.19: *JPEG-2000 test image : compound text/graphics*

Figure 2.20: *Details of coded compound image (a):EZW, (b):Adaptive method*



(a) (b)

Figure 2.21: *Details from compressed barbara image at 0.4bpp (a)EZW, (b)Adaptive method*

		LMS			
		1-D filter		2-D separable filter	
Image Name	EZW	Adapt. FIR	Adapt. OS	Adapt. FIR	Adapt. OS
Call_papers	36.50	36.80	36.96	36.87	36.99
Sci_Tech1	36.19	36.30	36.41	36.33	36.48
Sci_Tech2	31.14	31.56	31.61	31.60	31.65
House	38.80	38.97	39.10	39.08	39.22
Baboon	30.50	30.46	30.56	30.50	30.61
Tourism1	30.20	30.14	30.20	30.15	30.22
Tourism2	27.50	27.88	28.02	27.92	28.07
Tourism3	32.25	32.20	32.27	32.22	32.31
TR_map	31.50	31.66	31.86	31.70	31.92
News0	34.21	34.15	34.19	34.20	34.23
News1	32.25	32.19	32.22	32.22	32.25
News2	23.90	24.12	24.20	24.18	24.27
Map_Africa	33.02	33.29	33.43	33.36	33.50
s_text1	34.10	36.10	36.28	36.30	36.33
Pepper	38.15	38.18	38.17	38.50	38.44
Zelda	39.87	39.79	39.71	39.91	39.85
Barbara	35.90	35.78	35.85	35.81	35.91
Bookshelf	35.90	35.79	35.87	35.84	35.95
Bookcover1	33.11	33.08	33.10	33.15	33.17
Bookcover2	34.17	34.43	34.51	34.50	34.59
Bike	35.10	34.96	35.08	34.99	35.14
Cafe	30.21	30.13	30.18	30.16	30.22
Cats	40.88	40.77	40.80	40.81	40.86
Cmpnd1	41.58	41.59	41.63	41.60	41.66
Hotel	37.77	37.75	37.78	37.75	37.80
Tools	31.02	31.03	31.09	31.07	31.13
Water	42.00	41.92	41.95	41.91	41.97
Woman	35.52	35.49	35.52	35.48	35.52

Table 2.2: *Experiment results (PSNR) of test images at 1bpp with LMS adaptation.*

		RLS			
		1-D filter		2-D separable filter	
Image Name	EZW	Adapt. FIR	Adapt. OS	Adapt. FIR	Adapt. OS
Call_papers	36.50	37.02	37.16	37.07	37.18
Sci_Tech1	36.19	36.34	36.45	36.36	36.52
Sci_Tech2	31.14	31.61	31.65	31.65	31.69
House	38.80	39.00	39.13	39.11	39.26
Baboon	30.50	30.52	30.60	30.55	30.66
Tourism1	30.20	30.19	30.25	30.21	30.27
Tourism2	27.50	27.90	28.05	27.94	28.10
Tourism3	32.25	32.22	32.29	32.24	32.34
TR_map	31.50	31.72	31.92	31.76	31.98
News0	34.21	34.20	34.25	34.24	34.29
News1	32.25	32.23	32.26	32.26	32.30
News2	23.90	24.15	24.23	24.21	24.30
Map_Africa	33.02	33.35	33.49	33.43	33.56
s_text1	34.10	36.13	36.31	36.33	36.35
Pepper	38.15	38.20	38.19	38.53	38.46
Zelda	39.87	39.81	39.73	39.93	39.89
Barbara	35.90	35.82	35.90	35.85	35.95
Bookshelf	35.90	35.84	35.92	35.88	35.99
Bookcover1	33.11	33.13	33.15	33.20	33.22
Bookcover2	34.17	34.50	34.58	34.58	34.68
Bike	35.10	34.99	35.12	34.03	35.19
Cafe	30.21	30.17	30.22	30.20	30.27
Cats	40.88	40.78	40.82	40.82	40.90
Cmpnd1	41.58	41.64	41.70	41.65	41.72
Hotel	37.77	37.77	37.81	37.77	37.83
Tools	31.02	31.05	31.12	31.10	31.18
Water	42.00	41.93	41.97	41.94	42.02
Woman	35.52	35.52	35.55	35.50	35.55

Table 2.3: *Experiment results (PSNR) of test images at 1bpp with RLS adaptation.*

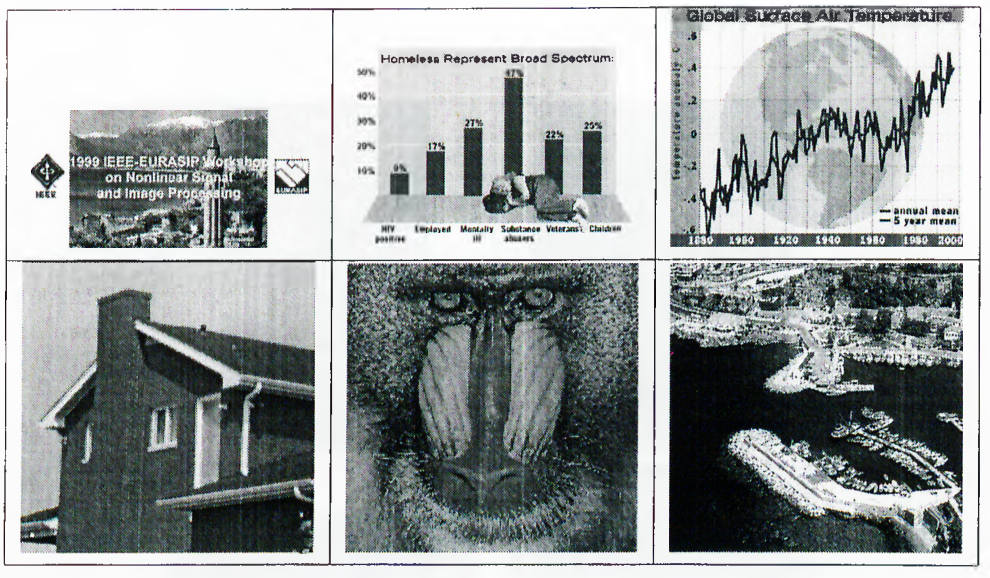


Figure 2.22: *Test images: Call_papers, Sci_tech1, Sci_tech2, House, Baboon, Tourism1.*

The adaptive decomposition can also be used in conjunction with the regular fixed wavelet transform. For example, one level of decomposition can be made with the adaptive filter bank, and the next decomposition can be made with the fixed filter bank, for coding. For some of the test images, two level of adaptive subband decomposition followed by fixed wavelet decompositions up to the required number of levels for the EZT coder, gives better compression results. These results are presented in Table 2.4. The improvements may be due to the loss of predictability of the polyphase components for very small images. In some other cases, the different decomposition structures yields slightly worse results. The worse results indicate that these images produce incompatible sub-images for EZT coding.

In a filter bank structure, the perfect reconstruction depends on the lossless transmission of the subsignals to the synthesis side. In adaptive filter banks, the high-band signal is also used for adapting the synthesis filter. Therefore, at very low bit rates the performance of the adaptive filter bank coder deteriorates because of severely perturbing the values of $x_l(n)$ and $x_h(n)$. Consider the PSNR vs. CR plots shown in Fig. 2.25 for the image of Figure 2.25 . Above CR=20 level, fixed filter banks start producing better results than the adaptive

Figure 2.23: *Test images: Tourism2, Tourism3, TR_map, News0, News1, News2, Map_Africa, s_text1, Pepper, Zelda.*

Figure 2.24: *Test images: Barbara, Bookshelf, Bookcover1, Bookcover2, JPEG-2000 images: Bike, Cafe, Cats, Cmpnd1, Hotel, Tools, Water, Woman.*

Image Name	2 level adaptive FIR	2 level adaptive OS
Call_papers	36.84	36.95
Sci_Tech1	36.30	36.45
Sci_Tech2	31.62	31.67
House	39.03	39.17
Baboon	30.52	30.64
Tourism1	30.14	30.20
Tourism2	27.92	28.06
Tourism3	32.23	32.32
TR_map	31.71	31.93
News0	34.22	34.24
News1	32.22	32.25
News2	24.19	24.26
Map_Africa	33.37	33.52
s_text1	36.32	36.34
Pepper	38.48	38.43
Zelda	39.89	39.84
Barbara	35.82	35.91
Bookshelf	35.85	35.97
Bookcover1	33.16	33.16
Bookcover2	34.50	34.57
Bike	35.10	35.15
Cafe	30.22	30.24
Cats	40.78	40.83
Cmpnd1	41.62	41.64
Hotel	37.79	37.80
Tools	31.11	31.15
Water	41.94	41.96
Woman	35.53	35.54

Table 2.4: *Experiment results (PSNR) for 2 level LMS adaptive decomposition followed by fixed wavelet decomposition.*

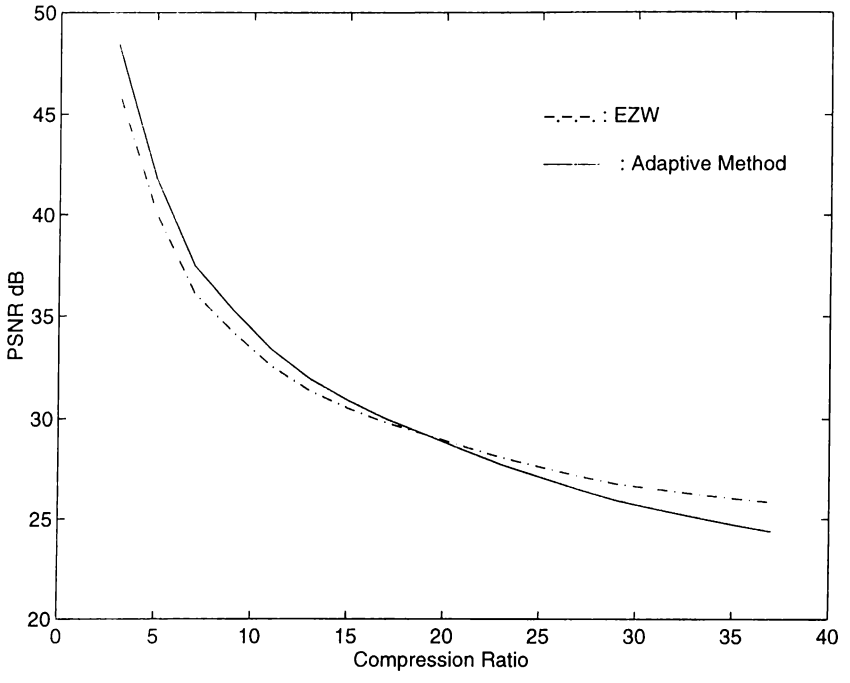


Figure 2.25: *EZW versus adaptive method at different CR's*

filter bank. However, at these CR levels, the coding is not visually transparent, and disturbing coding artifacts become visible at the encoded images.

The aim of this work is to achieve high quality image coding, therefore the use of adaptive predictors is meaningless at very low bit rates. If the subsignal x_h is corrupted during transmission, then the adaptive scheme should be cancelled and the adaptive predictor should be substituted with a fixed half-band low pass filter. The corruption or the divergence of the adaptive algorithm can be checked by comparing the low-low sub-image and the reconstructed image. If the reconstructed image is too much different than the upsampled low-low image, then the adaptive algorithm can be cancelled.

The Set Partitioning (SP) algorithm [58] is known to increase the performance of the EZW. The SP algorithm also gives a similar amount of improvement in PSNR for adaptive subband decomposition algorithm. For example, the SP algorithm improves the PSNR of the wavelet transform coder for the image in Fig. 2.17 to PSNR=36.81 at 1bpp, and the adaptive filter bank with the SP algorithm gives a PSNR of 37.50 at the same bit rate. Typically, The SP algorithm improves the PSNR by about 0.5dB to 1dB at around CR=8:1

for both the fixed wavelet decomposition and the adaptive subband decomposition.

The Adaptive Morphological Subband Decomposition described in [45] uses an alternating coding strategy for different regions in the image. Specifically, it chooses linear filters for natural looking regions and morphological filters for regions of graphical images. In the "baboon" image this algorithm achieves a PSNR of 25.858dB at 0.49bpp, whereas our adaptive algorithm gives a PSNR of 26.91dB at the same bit rate. Furthermore, the filter selection method used in [45] can also be used in our algorithm to switch between adaptive FIR and adaptive OS filter banks, as well. The adaptive nonlinear filter banks are successful in coding images with regions separated by sharp edges. On the other hand, the adaptive FIR filter bank is useful for regions with uniform textures like images of grass or forest. The texture detection algorithm described in [45] can be used for separating such regions and using alternating adaptive filter banks. The PR property is preserved since the texture detection is based on the low-low component of the decomposed signal which are available both at the encoder and the decoder side.

2.6 CODING OF COLOR IMAGES

The extension of gray tone image coding to color images is straightforward. Most of the coding algorithms that are developed for gray tone images are directly applied to the color components of the color image [8], [98], [12].

Three color components are used for reproduction purposes. This is due to the sensitivity of the human eye to three basic color wavelengths, namely Red, Green, and Blue (RGB). The human eye does not act like a spectrum analyzer to perceive the exact wavelength of a color. Instead, it perceives the stimuli generated by three perceptron nodes (sensors) which have color sensitivity curves centered around red, green, and blue colors [12]. If the sensitivity of the three color sensors are represented by $S_i(\lambda)$ which is a function of the wavelength, λ, then the spectral response of the sensor i to the spectral energy

distribution of a colored light $C(\lambda)$ is

$$\alpha_i(C) = \int_{\lambda_{min}}^{\lambda_{max}} S_i(\lambda)C(\lambda)d\lambda, \quad i = 1,2,3 \tag{2.46}$$

With this equation, any color can be reproduced by mixing appropriate amount of primary colors. This yields to another useful fact that the when the three primary colors are transformed to other three-element coordinate systems, they can be transformed back to RGB system to obtain the desired reproduction, if the transformation matrix is invertible. There are several color coordinate systems [12] which are practically used in various standards. For example the X,Y,Z system of C.I.E. uses the following coordinate transformation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.47}$$

where Y stands for the luminance (the gray level) component.

Similar to the C.I.E. standard, the NTSC transmission system uses the following transformation:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.48}$$

where Y, again, stands for the luminance component.

The back transformation to the RGB domain is performed by multiplying the transform domain components with the inverses of the transformation matrices.

It is usually more practical to apply the gray tone coding algorithms over the transform domain components, which are approximately decorrelated using the transformation matrix. Typically, the luminance component Y stands for the gray tone counterpart of the colored image, and the other components carry the extra color information. Our coding algorithm can also be used over the transformed color components, just like in the case of other image coding standards, such as JPEG, EZW, etc.

2.7 SUMMARY

An adaptive subband decomposition scheme for high quality gray tone image coding is introduced in this chapter. Gray tone image compression is essential for coding some portions of the digitized library, as well as for many other image archival applications. The proposed adaptive decomposition scheme performs better than the conventional fixed wavelet coders. Furthermore, the coding improvement is more for images with sharp variations. Since most of the gray tone images in a digitized library contains sharp edges and subtitles inside the gray tone image, the proposed algorithm efficiently encodes them.

In the implementation of the coder, the adaptive prediction filters are embedded into a polyphase structure. In the two channel structure, the high-band subsignal is estimated from the low-band subsignal using a least squares type adaptation algorithm. The perfect reconstruction property of the decomposition structure is retained as long as the same adaptation algorithm is used at the analysis and synthesis stages.

The adaptive estimation of the high-band subsignal ($\mathbf{x_2}$) from the low-band input signal ($\mathbf{x_1}$) eliminates the parts of $\mathbf{x_2}$ which are predicted from $\mathbf{x_1}$ by the least squares adaptive methods. Due to the nature of the least squares type adaptive algorithms, the prediction is a pixel-wise local operation.

We also introduced an adaptive filter bank with anti-aliasing filtering for the low-band signal. In this structure, the low-band signal is obtained from the input using half-band low pass filters followed by downsampling. The high-band subsignal is then estimated from the low-band subsignal using adaptive prediction. In this structure, a high quality multiresolution viewing capability is possible due to the anti-aliasing pre-filter.

The structure is also extended to two dimensional image decomposition in various ways. It is shown that separable and quincunx downsampling extensions are possible.

The overall structure is observed to be efficient for compressing a wide variety of images. Specifically, for the images of the digitized library that contain sharp variations such as text, subtitles and graphics, our algorithm

performs better than other methods in terms of PSNR. Our algorithm also improves the perceptual quality of the compressed image by eliminating the ringing artifacts which are disturbing at sharp transitions or subtitles.

Chapter 3

TEXTUAL IMAGE COMPRESSION AND ARCHIVING

A document image mainly consists of printed text and some marks and drawings. There can also be gray tone and color images inside a document. In order to increase the coding efficiency, the bi-level part - specifically the text part - should be removed from the rest of the document image and coded separately. There are efficient algorithms to separate the graphics parts of an image from the textual parts [72]. The textual part of the image usually contains important information for keyword search and retrieval. The digitally scanned pages need to be compressed to overcome the storage and retrieval problems in a digital library. Another application of textual image compression is the facsimile. High compression of textual images is crucial for digital facsimile applications because facsimile documents generally contain images of printed text. The recognition of characters is usually unreliable and may give unwanted results for documents in which every character and punctuation is important. In that case, the compression of the text as an image is important.

Historical documents are other applications where the documents should be kept as images [60].

Many methods have been designed for compressing images, text, speech, audio and many other waveforms [54]. In the literature, the problem of binary image compression is also widely investigated [59] - [70]. Since most of the commonly used coding techniques for arbitrary image compression do not take into account the special characteristics and the fast database search properties of the text images in a database, one has to deal with digital compression and archiving of binary text images differently. The difference of a binary text image from an arbitrary image is that the textual image contains repetitions of small patterns which are the images of characters or letters all over the image. Coding methods specific to this type of images will be described in this chapter.

Quick database search and retrieval is also an essential property for digital document libraries. The compression method should enable the user to perform keyword searches and similar multimedia operations. A large fraction of image compression techniques (such as subband coding [21] and transform domain techniques) do not enable direct pattern matching in the coded data after compression.

A text image, similar to a text itself, contains repetitions of small character images, i.e. letters. Exploiting the redundancy of these repetitions is the key step in some of the textual image coding algorithms [59] - [67]. A good approach to take advantage of this redundancy is to encode the repeated character images and their locations. This method efficiently compresses the textual image and it is appropriate for fast database search with some modifications. With this method, the lossy compression ratio is around 60:1 - 100:1, and the lossless compression ratio is around 15:1 - 30:1. The suitability for keyword search can be understood by observing that since the character images and labels are preserved in a symbol library, the keyword search can be carried out inside the character library and the table which shows the locations of characters.

In this chapter, we consider the problem of textual image compression in subband domain. The subband characteristics of the binary text images is suitable for higher compression. Our approach is based on finding the repetitions of small character images in the subband decomposed images. A number

of advantages about using subband decomposition has been observed in our experiments. The final lossless and lossy compression ratios in our method are higher than the lossless and lossy compression ratios of the previous methods, and the time required for encoding decreases by a factor of 2^M, where M is the level of wavelet decomposition, when compared to the direct use of the method described in [60]. The encoding speed is obtained by constructing the symbol library and performing the searches inside a smaller subband image.

In Section 3.1, the method of extracting the character images is described. After a brief explanation of the subband decomposition methods that are used, the compression scheme in subband domain is presented in Section 3.5. In Section 3.3, the compression of the character image libraries in subband domain is described in detail. Finally, the experimental results are presented in Section 3.5.

3.1 TEXTUAL IMAGE COMPRESSION TECHNIQUES IN THE LITERATURE

The information in a textual image mainly consists of text data, therefore the recent compression algorithms treat the problem of compression different than the compression of arbitrary binary images that do not contain text images [59] - [71]. Most of the binary image compression methods are based on pixel-wise prediction of binary values inside the image. Some techniques have been developed in order to achieve a better performance for documents that contain both textual and arbitrary images. These techniques separate the textual part of the image from the non-textual part [72], [73], and enhance the textual image [74]. After the enhancement, the textual part is compressed differently.

Optical character recognition (OCR) methods [75] - [78] for compressing textual images result in high compression ratios, however it is usually not reliable. Most of the time, the operation should be pre and post-processed by a person to eliminate errors. Furthermore, it may not yield a faithful reproduction of the document in terms of fonts, printing sizes and spacings, which may

be required in same applications. If the document is a historical document from a library, or a page that contains information about the printing environment (this can be critical for criminology), simple optical character recognition is not suitable. Due to this reason, the image compression of the document image may be required instead of transcribing the document. Depending on the application, the OCR output can accompany the image for keyword search and similar applications. Or reversely, this coding scheme can serve as a preprocessing for OCR. Since the only characters to be considered are inside the character library, the OCR can be done inside this library, and the text reproduction can be done together with the locations of the characters. With this motivation, our attention is focused to a well known textual image compression method which is based on finding isolated characters and determining their positions inside a document [60].

A number of authors [63] - [67] consider the lossy compression of text images. The first method by Ascher and Nagy [63] used the method of finding the isolated characters and encoding their locations, however the constructed symbol library was not coded. Other lossy compression methods that have been successfully used in the past include "combined symbol matching (CSM)" [64], "pattern matching and substitution (PMS)" [67] and "weighted and-not (WAN)" [66].

The CSM was proposed by Pratt et al. [64]. The difference from the previous scheme was that, the rare symbols were left uncoded, and the marks that do not correspond to text symbols were coded by using two dimensional runlength coding. This method was later improved by pre-loading a static library constructed by the most common typewriter fonts. By pre-loading the most common 150 words in the English language and representing them with single marks, the compression ratio was further improved to around 145:1. There are certain problems with this method. It cannot be applied to any text document scanned at arbitrary resolutions. The type and the size of text fonts is important in this method, and the compression decreases when the text is not in English. In [66], Holt et al. introduced an improvement for CSM using a WAN pattern matching criterion.

Johnsen et al. proposed PSM which was better than CSM in terms of compression and robustness. This method decomposes large symbols into smaller

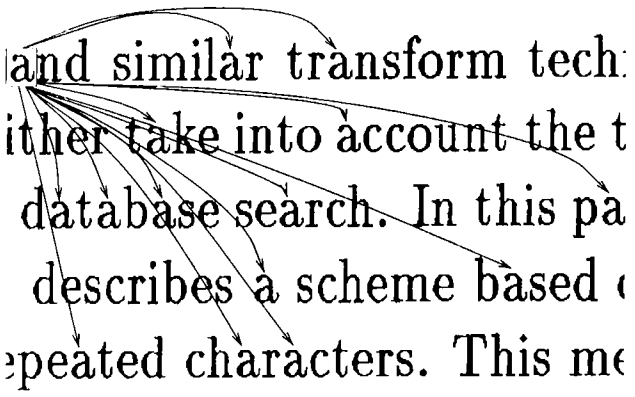


Figure 3.1: *Part of the original document image where the repetitions of letter "a" are illustrated.*

parts so the matching techniques can be used to compress arbitrary graphical shapes.

Among other lossless binary image compression techniques which do not take into account the content of the image [68] - [70], Witten et al. introduces a second phase to the lossy methods to accomplish a lossless compression [60]. In this case, the residue image, which is the difference image obtained by subtracting the original image from the lossy compressed image, is encoded in a lossless manner.

In all these methods, the procedure depends on the same character image extraction idea. The general outline of these procedures can be described in a sequence as follows (see Fig. 3.1):

i) Find and extract a character in the image.

ii) Compare it with the symbol library consisting of the separate character images.

iii) If the character exists in the symbol library, then take the location only, otherwise add it to the library.

iv) Compress the constructed library and the symbol locations.

A further step is proposed by Witten *et al.* [60] who encode the residue image so that lossless compression can be achieved.

In Section 3.3, I describe how this procedure can be applied to subband decomposed textual images. Some experimental results about document retrieval issues are given in Section 3.4. The advantages of using a smaller sized images are explained and the compression results are presented in Sections 3.5 and 3.6.

3.2 IMAGE CODING USING WAVELET TRANSFORM

Image coding using wavelet transform has been widely investigated for gray-tone and color images [79]- [81]. By using the relation between subband coding and wavelet transformation [82]- [84], the wavelet transform is implemented with a perfect reconstruction filter bank, in practice. The idea is to divide the image data into subbands corresponding to different frequency contents. Let us assume that $H_0(\omega)$ and $H_1(\omega)$ are the low-pass and high-pass filters of a perfect reconstruction filter bank, respectively. In 1-D case with one level decomposition, the input signal $x[n]$ is filtered by $h_0[n]$ and $h_1[n]$ and the resultant signals are down-sampled by a factor of two. In this way two sub-signals $x_0[n]$ and $x_1[n]$ are obtained, i.e.,

$$x_i[n] = \sum_k h_i[k]x[2n-k], \quad i = 0, 1 \tag{3.1}$$

The sub-signals $x_0[n]$ and $x_1[n]$ contain the low-pass and the high-pass information of the original signal $x[n]$ respectively. The overall operation is illustrated in Fig. 2.1.

The synthesis part of this decomposition is performed by the complementary synthesis filters after upsampling stage [82]- [84].

In 2-D case, implementation of one level decompositions in horizontal and vertical directions results in four subband images, ll, lh, hl, and hh. The subband image ll is obtained by first lowpass filtering the image in horizontal direction and then lowpass filtering in the vertical direction, lh is obtained by first lowpass filtering the image in horizontal direction and then highpass filtering in the vertical direction, and so on.

For most of the gray tone images, the corresponding subband images can be quantized without introducing much perceptual degradation. After the quantization step, the data set in each subband can be encoded separately by using the most suitable encoding strategy for the subband image. Typically, the ll image is coded using DCT [85] and other block based transform domain techniques [86], and other bands are coded by entropy coding and run-length encoding. Recently, the cross correlation between the subband images has been used for achieving higher compression [56]. This idea is also used in our work.

The text images that we deal with are binary images. If perfect reconstruction is desired, the number of levels in the subband images certainly increase as a result of linear filtering. (There are also some group theoretical approaches to subband filtering [87] - [89] that keeps the number of levels the same after the transformation). In our first experiments, we used the simplest FIR perfect reconstruction filter bank

$$h_0[0] = h_0[1] = 1/2, \quad h_0[n] = 0 \quad for \quad n \neq 1, 2 \tag{3.2}$$

$$h_1[n] = (-1)^n h_0[n] \tag{3.3}$$

This filter bank corresponds to Haar Wavelet Transform [21] and it came out to have a number of advantages in our application for the reasons that will be described in Sections 3.3 and 3.5. With this pair of filters, the subband images of a binary image become 5-level images after one stage decomposition, however the encoding stage for the subband images will have a performance to come over the disadvantage of increasing the number of levels.

Two more subband decomposition strategies will be investigated in this study as well. The first one is the Binary Subband Decomposition which is derived from the Binary Wavelet Transform work by Swanson and Tewfik [89], and the next one is the Binary Nonlinear Subband Decomposition which is introduced here.

3.2.1 BINARY SUBBAND DECOMPOSITION - BINARY WAVELET TRANSFORM

In Binary Wavelet Transform (BWT) [89], the binary images are decomposed into binary transform images. The operations that are used in this transform scheme require simple modulo-2 operations. An advantage of this method is that it shares many of the important characteristics of the real wavelet transform. Typically, the binary wavelet transform (BWT) yields an output similar to the thresholded output of a real wavelet transform operating on the image. The theory of binary wavelets is developed in terms of perfect reconstruction filter banks in GF(2) (Galois Field 2) [87]- [92].

The analysis stage of the BWT consists of binary filters followed by decimation by a factor of two. Since all the additions are made in modulo - 2, the operations can be replaced by logical "xor" operations. The counterparts of the frequency characteristics and vanishing moments are available in Galois Field - 2 GF(2).

Let $I = [x[n_1, n_2]]_{N \times N}$ be the image matrix whose entries are pixel values. The decomposition is performed by pre- and post-multiplying the image matrix with the analysis matrix T and its transpose T^t.

$$I_d = T \times I \times T^t \tag{3.4}$$

where I_d is the decomposed image.

The transform matrix T is constructed from a pair of lowpass and high-pass filter vectors $\mathbf{h}_0 = [h_0[0]\ h_0[1] \cdots h_0[P-1]]$ and $\mathbf{h}_1 = [h_1[0]\ h_1[1] \cdots h_1[Q-1]]$ as follows :

$$T = \begin{bmatrix} 2-circulant(\mathbf{h}_0) \\ 2-circulant(\mathbf{h}_1) \end{bmatrix} = \begin{bmatrix} \mathbf{h}_0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & \mathbf{h}_0 & 0 & 0 & \cdots \\ & & & & & \\ - & - & - & - & - & - \\ \mathbf{h}_1 & 0 & 0 & 0 & 0\cdots & \\ 0 & 0 & \mathbf{h}_1 & 0 & 0\cdots & \\ & & & & & \end{bmatrix} \tag{3.5}$$

where

$$2-circulant(d_0, d_1, \cdots, d_{N-1}) = \begin{bmatrix} d_0 & d_1 & d_2 & & d_{N-1} \\ d_{N-2} & d_{N-1} & d_1 & & d_{N-3} \\ d_{N-4} & d_{N-3} & d_2 & \cdots & d_{N-5} \\ & & & & \\ d_2 & d_3 & d_4 & & d_1 \end{bmatrix} \tag{3.6}$$

If the filter size is less than N, then the filter vector is padded with zeros. In this way, the matrix T has two 2-circulant parts.

The matrix I_d in Eq. 3.4 has the following form

$$I_d = \begin{bmatrix} I_{ll} & I_{lh} \\ I_{hl} & I_{hh} \end{bmatrix} \tag{3.7}$$

where I_{ll}, I_{lh}, I_{hl} and I_{hh} correspond to low-low, low-high, high-low and high-high images, respectively.

Due to the structure of the transformation matrix T, the BWT can also be implemented using a binary subband decomposition filter bank. This implementation will be named as the Binary Subband Decomposition (BSD). Consider the following choice of analysis filter vectors

$$h_0 = [1 \;\; 0], \quad h_1 = [1 \;\; 1] \tag{3.8}$$

In 1-dimensional case, the subband signals are obtained as

$$x_l[n] = \sum_{i=0}^{1} x[2n-i]h_0[i] \tag{3.9}$$

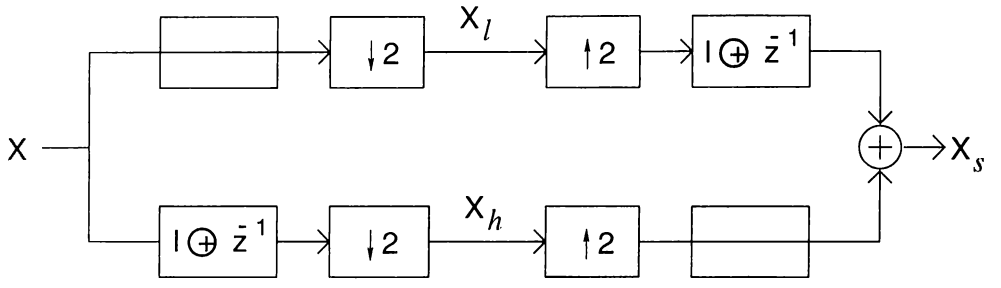


Figure 3.2: *One stage subband decomposition with "xor" filter.*

$$x_h[n] = \sum_{i=0}^{1} x[2n-i]h_1[i] \tag{3.10}$$

where the arithmetic operations are carried out in modulo-2 arithmetic. Note that the modulo-2 summation corresponds to the logical "xor" operation. The lowband signal $x_l[n]$ is nothing but a decimated version of the original signal $x[n]$. The signal $x[n] * h_1[n]$ is zero everywhere except at the 0 to 1 or 1 to 0 transitions, and $x_h[n]$ is a decimated version of $x[n] * h_1[n]$. This operation corresponds to the analysis part of Fig. 3.2.

The synthesis operation of the BWT representation is performed by the inverse matrices. In the case of the BSD described in Eq.'s 3.10 and 3.10, the reconstruction can be done by simple FIR synthesis filters

$$g_0 = [1 \quad 1], \quad g_1 = [1 \quad 0] \tag{3.11}$$

The reconstruction stage of this BSD is shown in the right part of Fig. 3.2.

For an arbitrary BWT, it may not be always possible to carry out the synthesis operation by a filter bank as in Fig. 3.2. In those cases, the reconstruction can be performed via binary matrix multiplications instead of synthesis filters [89].

An example of decomposing a letter image into binary sub images is given in Fig. 3.3. The left image is the original image of letter "a", and the right four images are four subband images generated using the BWT.

The next subband decomposition method is based on the work by Egger et al [45]. In this work, the decomposition filter bank consists of non-linear filters instead of the standard perfect reconstruction linear filters. The general

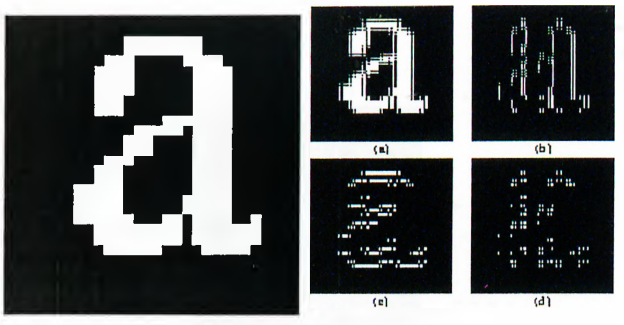


Figure 3.3: *Binary wavelet decomposition of letter "a"*

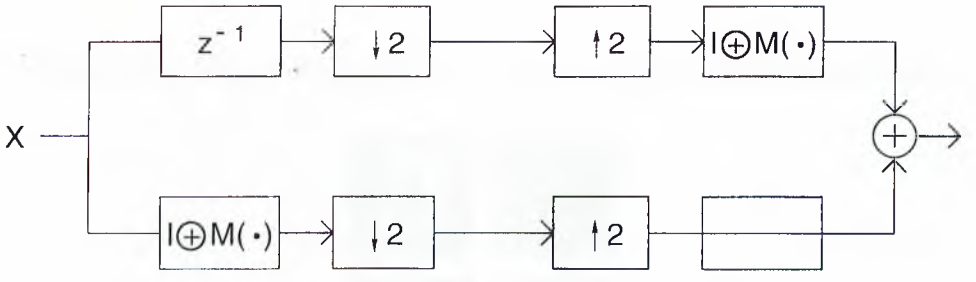


Figure 3.4: *One stage nonlinear subband decomposition*

one stage, one dimensional analysis and synthesis operation is illustrated in Fig. 3.4. In this figure, M stands for a nonlinear operation It can be observed that the perfect reconstruction of the input is possible in this structure if the nonlinear function M satisfies the generalized half-band condition described in [45]. Specifically, the nonlinear operation that we use is the median filtering. In the horizontal direction case, the operation is as follows :
The downsampling is directly performed on the data in horizontal direction, so the dark pixels in Fig. 3.5 are transmitted through the upper branch. In order to predict the value of the white pixel in the figure, the pixels numbered from 1 to 6 are used. The prediction is made by simply taking the median of these six pixel values. The prediction error which is the difference of the prediction value from the true value is transmitted through the lower branch. In the reconstruction step, the same prediction is made by using the data transmitted from the upper branch, and that prediction value is added to the prediction error which is transmitted from the lower branch.

This scheme has various advantages over the linear subband decomposition when the image has sharp edges. It should be pointed out that when

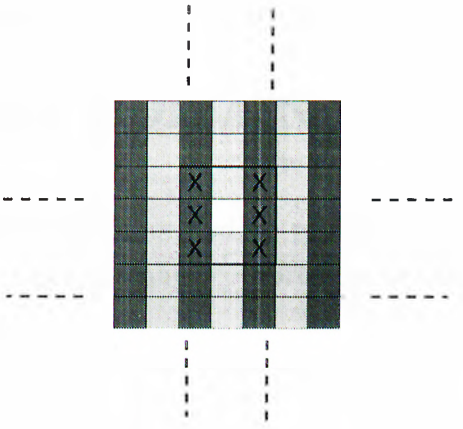


Figure 3.5: *Horizontal direction nonlinear subband decomposition*

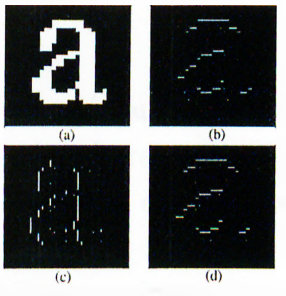


Figure 3.6: *Nonlinear binary subband decomposition of letter "a"*

the subtraction is performed for transmitting the difference between the pixel value and the median prediction value from the lower branch, the modulo-2 subtraction can be used. In this case, the resultant difference is again binary. By using this modification, the subband images can be obtained to be binary. In the synthesis part, the corresponding addition should be again modulo-2.

Since the document images are binary images, clearly it is advantageous to use binary subband decompositions. The pattern search speed improves for the binary images. Moreover, the matching criterion is simpler than the multi level matching criterion that is used in the *ll* subband image obtained by Haar Wavelet transform. Since the resultant character libraries in the subband images are also binary images for these transformations, they can be more efficiently compressed.

3.3 TEXTUAL IMAGE COMPRESSION IN WAVELET DOMAIN

Our approach to the textual image coding problem is to combine the wavelet transformation and the textual image compression techniques described in Section 3.1.

3.3.1 CHARACTER MATCHING BASED COMPRESSION

In a digital library, most of the computation time to acknowledge a query is spent in finding the matches of a character image inside the text image. We take the character image and slide it over the whole text image to find the sum of the absolute difference at every location. In this way, a matching error matrix $E(i,j)$ is constructed. Our matching criterion is based on a combination of absolute error between the pixel values at each location and a constraint on the sizes of the letters. The E matrix thus has high and low values of the matching errors. The points corresponding to the local minima of matrix are then tested for being detected as the searched character. The latter test is for removing the errors that occur when both the pixel-wise error is small and the letter sizes match.

After one level of decomposition, the time required for computing the E matrix decreases approximately by a factor of 16 because the sizes of the character images and the sizes of the text images are one fourth of the original image (i.e. the width and the height of the images are half of the originals). For most of the test images which are 11 or 12 point printed text scanned at 300dpi, one level of subband decomposition preserves the intelligibility of individual character images. Furthermore, the character locations in the low-low sub-image for this resolution is sufficient for determining the locations of the characters in the reconstructed image. Due to this observation, the steps in Section 3.1 are carried out in the ll subband of the original image. Each mark is determined in

the *ll* image and checked for the repetitions inside the same image. The places of occurrence and the boundaries of the marks obtained from the *ll* image is used for all other subbands (Method 1). In other words, we do not carry out other search algorithms in any subband image other than the *ll* image. The time required to make the search in the subband improves the speed of both the encoding process and the query time. Certainly, the pattern matching is now done for five-level images obtained by one stage decomposition in the case of Haar wavelet.

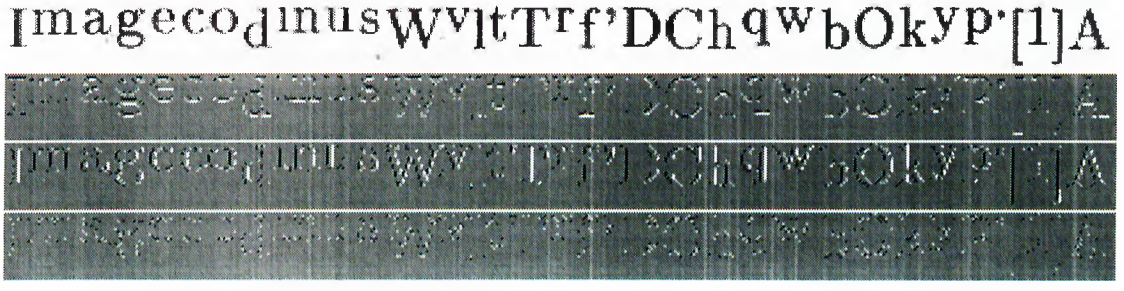


Figure 3.7: *Symbol libraries of subband images. ll, lh, hl, and hh*

Three other library of symbols is generated corresponding to the library obtained for the *ll* subband image (Fig. 3.7). These subband library (SL) images constitute the subband decomposition of the original library (OL) of symbols which is obtained by using the original binary image and the same pattern matching method (Method 2). These SL images have one fourth the size of the OL image and they are five-level images.

Since the same method is used for encoding the locations of the repeated characters (step 5.b) in Method 1 and Method 2, the amount of bits required for encoding the locations for Method 2 is larger. This is due to the size of the page to be encoded. The size of the original page is four times greater than the size of the subband images, so the location coordinates for Method one is twice larger. The amount of bytes for encoding the locations of the repeated characters (step 5.b) in Method 1 is about 1.4 times smaller than the amount of bytes for encoding the locations for Method 2.

By the elimination of the repeated character images from each subband image, we have four subband residue images which may be coded as a last step for achieving lossless compression as in [60].

After these operations, the compression efficiency is completely determined by how successful the SL images are compressed. Since the OL image is a binary text image, it can be efficiently compressed by some lossless entropy coding methods such as predictive coding or Lempel-Ziv coding [61], [59]. The challenge here is to find a coding method for the SL images which results in a better overall performance. The number of bits required for encoding the locations and symbol labels is small when compared to the number of bits to represent the symbol library. This situation is valid as long as the number of characters in a document is approximately less than 4000. If the document size is large and the number of characters inside the document is more than 4000, then the number of bits to represent the locations becomes comparable to the number of bits to represent the symbol library. In this work, most of the attention is concentrated on the compression of symbol library images.

3.3.2 EFFICIENCY OF SUBBAND DECOMPOSITION BEFORE T.I.C.

Let us suppose that the document image consists of L symbols and the symbol library contains M characters and punctuation marks. If there is no subband decomposition, the total number of bits required to store this document is given by

$$T_o = L((p+q) + H_l) + B_0 \tag{3.12}$$

where p and q are the number of bits for representing the x and y coordinates of a character, H_l is the entropy of the source in terms of symbols, and B_0 is the number of bits required for the compressed symbol library. The entropy H_l is given by $H_l = -\sum_{m=1}^{M} p_m log p_m$ where p_m is the probability of the m-th character. Clearly, the entropy of the character library is language and context dependent.

After one level of subband decomposition, the total number of bits changes to

$$T_1 = L((p'+q') + H_l) + B_1 \tag{3.13}$$

where p' and q' are the number of bits for representing the x and y coordinates of a character in the subband domain, and B_1 is the number of bits required for the new compressed symbol library. Since both the old and the new libraries

should contain the same number of symbols, the entropy term H_l remains the same in both 3.12 and 3.13. If the locations of the characters are determined in the subband domain then $p' = p/2$ and $q' = q/2$ because the low-band image is a quarter size image. This may result in a one pixel shift of the character boundaries in the coded image. In the simulations, it is observed that human eye does not distinguish one pixel jitter in character locations for a text printed in 11 or 12 point font and scanned at 300dpi resolution. The jitter can be eliminated by recording the locations of the characters in the original image in which case $p' = p$ and $q' = q$.

The number of bits required for representing the multiresolution library is smaller than the original character library, i.e., $B_1 < B_0$ for all the examples studied in this work (see Tables 1 and 2). The high-band textual images consist of dots and lines at the boundaries of the character images. These dots and lines are highly correlated with the low-band character images. Vectors of the multiresolution codebook are compressed by first appending the bit-planes of the subband character images followed by arithmetic coding. This method results in higher compression rates than individual compression of the subband codebooks.

Since the low-low subimage, $x_{ll}[\underline{n}]$, is a coarse version of the original textual image, the TIC algorithm described in Section 3.1 can be applied to $x_{ll}[\underline{n}]$. The procedure starts with determining every character image in $x_{ll}[\underline{n}]$ and comparing it with the images in the symbol library. If a similar image exists in the low-low library, then the character boundary and the location of this character is used for other subband images $x_{lh}[\underline{n}]$, $x_{hl}[\underline{n}]$, $x_{hh}[\underline{n}]$, as well. If the image does not exist in the library, it is added to the library and the corresponding highband character images are added to their symbol libraries. In this way, four symbol libraries are constructed from the subband images as shown in Fig. 3.7.

The total number of bits to represent the four Symbol Libraries (SL) is smaller than the number of bits needed for the SL generated by the direct use of the TIC method. The efficiency of the subband domain textual compression is achieved by making use of the correlation between subband images corresponding to a textual image. As a consequence of the binary time localized transforms, the edges of the character images are almost at the same locations

in all subbands. The bit-planes of four subimages are appended as shown in Fig. 3.10 and the resulting multilevel image is compressed using an arithmetic coder which encodes the predicted four bit pixel value.

In the case of the Haar WT, the SL images are not binary. The bit-plane appended image has 4×5 gray levels. It is observed that this image can be quantized to 12 levels with almost no visual degradation. These library images are also used in our simulation studies. Since the Haar transform images are multilevel images, both keyword search and compression times are higher compared to the binary transforms because of the increase in the pattern matching time for multilevel images.

Subband decomposition levels can be increased, if the character sizes are large or the scanner resolution is high. For example 2^K subband images are generated for a K level decomposition. In such a case, 2^K character libraries have to be encoded etc.

Our coding results are in parallel with the results in [89] who observed that the entropy of the subband images decreases. The experimental results are also in parallel with the work in [57] in the sense that subband decomposition followed by VQ is more efficient than straightforward vector quantization of images.

3.3.3 PATTERN MATCHING CRITERIA

The pattern matching criteria for extracting and comparing the character images inside the document is an important step that determines the success of the coding algorithm. In conjunction with the multiresolution TIC method, any pattern matching technique can be used in the construction of the symbol library.

In our simulation studies, we used two methods for symbol matching. The first one is the Local Template Matching (LTM) criterion used in [60] and [61]. This matching criterion is an efficient algorithm which is well tuned for printed text. It detects the differences between an image-wise similar looking "c"

and "e" symbols, etc. In this work, the LTM criterion is used in the low-low subimage. A detailed description of the LTM method can be found in [61]. The second symbol matching method is based on the Self Organizing Map (SOM) neural network model. This neural network model is used in order to illustrate the effects of using various symbol matching criteria in the multiresolution TIC.

A SOM is an unsupervised neural network which is mainly used for vector quantization and clustering [93]. It consists of a map of output nodes each receiving signals from a vector of input units. There is a weighted connection between each input unit and output node, forming a weight vector associated with each output node. The training algorithm modifies the network weights to extract the statistical properties of the training set and groups similar vectors into similar classes. A spatially localized region of nodes is formed at a location on the output map where the similarity of an input vector and the weight vector is maximum. SOM forms complex shapes following the data distributions in the feature space, so that regions of the most activated nodes on the output map can be interpreted as clusters in the feature space. A single layer SOM divides the input (feature) space into convex regions analogous to one layer feedforward neural network.

The SOM defines a mapping from the input data space onto a regular two dimensional array of nodes [94]. A reference vector $m_i \in R^n$ is associated with every node i. An input vector $x \in R^n$ is compared with m_i and the best match is taken as the winning node or the response. That is, the input is mapped to onto the winning node. The best match node c can be taken as the one whose reference vector has the smallest Euclidean distance with the input vector x.

$$\begin{aligned} \|x - m_c\| &= \min_i\{\|x - m_i\|\} \text{ or} \\ c &= arg \ \min_i\{\|x - m_i\|\} \end{aligned} \tag{3.14}$$

In the learning (training) phase, the reference vector m_c of the winning node and the nodes which are its topographical neighborhood are updated. This update can be described by

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \tag{3.15}$$

where t is the discrete time coordinate (iterations) and h_{ci} is the neighborhood kernel. In general, $h_{ci}(t) = h(\|r_c - r_i\|, t)$ where $r_c \in R^2$ and $r_i \in R^2$ are the radius vectors of nodes c and i, respectively.

In this study, horizontal projections of the symbol images extracted from the textual image are used as the input vectors. The horizontal projection of a symbol produces a one dimensional vector with elements consisting of horizontal summations of pixel values along the rows in the symbol image. Similar components are mapped to topologically neighboring nodes. As a result, the symbol image with the least Euclidean distance to a winning node is selected as a representative symbol, and it is included in the symbol library. In the subband decomposed images, this task is carried out in the low-low subimage.

The SOM matching differs from the LTM matching in the sense that the representative symbols in the symbol library generated by the SOM are not necessarily the first of the similar symbols found in the document image. After the construction of the symbol library using the SOM criterion, the repetition of the symbols are found by running the SOM neural network again, and obtaining the nearest representative node for each symbol in the textual image. In this phase of the coding, the node values are not updated.

3.4 DOCUMENT RETRIEVAL

The issue of document retrieval is important for document archiving and should be considered simultaneously with the compression of the documents. In most applications, a query in a document library corresponds to a keyword search. There can also be queries regarding the search for a group of character images which corresponds to the image of a word or a phrase. The later query can be performed by marking a portion of the document image. The search algorithm can use only the subband library images and the data corresponding to the locations of the characters. If the character images of the word or phrase occur consecutively in the location database then a match occurs in the database.

It is evident that the pattern search can be carried out over the character library of the x_{ll} subband image as long as the character images in the low-low sub-image are still intelligible. In this way, the keyword search time is reduced by a factor of 2^2 for the binary transformations. Both the template images and the numbers corresponding to the repetition locations are small for the subband images. The improved speed for this kind of querying system in a document archiving application makes the subband decomposition scheme efficient.

Other advantages of the subband scheme include multiresolution image viewing and computational efficiency. As a direct consequence of the subband decomposition, a low resolution image x_{ll} is obtained and it can be used for fast preview purposes to decrease the band-width usage. The compression of the textual image is a computationally costly operation. When the steps for TIC described in Section 3.1 are applied to smaller sized images, the number of computations are reduced.

Assume that there are N possible character images in the textual image database. A document image may be composed of a subset of these N character images. Let us denote the query probability of the image I_i by q_{I_i} and the probability of occurrence of the character I_i in the document by p_{I_i}, which will be called the symbol probability.

We start by defining a cost function C to be minimized. This function is written in terms of symbol probabilities of the character images and their query probabilities. Assume that the considered textual image is composed of N character images $I_1, I_2, \ldots, I_N$ which constitute a subset of the database.

In order to eliminate the search redundancy, assume that the symbols for the character images can be ordered as $\alpha_1, \alpha_2, \ldots, \alpha_N$ in such a way that if α_1 and α_2 both exist in the header, α_1 comes before α_2. This ordering is denoted by an associating function Φ between α_j and I_i in the form of $\alpha_j = \Phi(I_i)$.

$$\text{minimize } C = E[\text{Search Length}] + \lambda E[\text{Library File Length}] \tag{3.16}$$

In the case of textual images, the library file consists of previously compressed images therefore the library file length is fixed and the cost function

reduces to:

$$C = E[\text{Search Length}] = \sum_{i=1}^{N} L_{\alpha_i} \hat{p}_{\alpha_i} \tag{3.17}$$

where L_{α_i} is the compressed size of the α_i character image. The parameter $\hat{p}_{\alpha_i}$ is given by :

$$\hat{p}_{\alpha_i} = (p_{\alpha_i} - \sum_{j=1}^{i-1} q_{\alpha j} p_{\alpha_i \alpha_j}) \tag{3.18}$$

and $p_{\alpha_i \alpha_j}$ is the probability that the character images α_i and α_j both exist in the document.

The minimization of the cost function is performed by switching the places of the two adjacent elements in the order list. Suppose these two ordered sets are ξ and ξ' defined as :

$$\xi = \ldots, I_i, I_{i+1}, \ldots, I_N \tag{3.19}$$

$$\xi' = \ldots, I_{i+1}, I_i, \ldots, I_N \tag{3.20}$$

If the expected search length C' corresponding to ξ' is less than expected search length C corresponding to ξ, that is

$$\frac{L_{I_{i+1}}}{q_{I_{i+1}}} < \frac{L_{I_i}}{q_{I_i}} \tag{3.21}$$

then a better ordering can be obtained by switching the places of i^{th} and $i+1^{st}$ elements. This result indicates that the search length minimization is independent of the probabilities p_{α_i}. This inequality is proven in Appendix A.

Using this inequality, the expected search length can be minimized in a bubble-sort manner. Define an arbitrary ordering ξ. In the first pass, compare α_{N-1} with α_N using inequality 3.21. If the search length decreases for the swapped order, swap the two terms. Then compare α_{N-2} with α_{N-1} and repeat the process. After N comparisons, the character I with the lowest L_I/q_I will be assigned to α_1. In the next pass, α_2 is determined in $N-1$ compares, and so on.

If the query probabilities for all symbols are not known a priory then the symbols (character library images) can be ordered according to their size from the smallest to the largest one. In this way, the decoder can stop the search

when a symbol with larger size than the queried symbol is encountered. This case corresponds to the inequality 3.21 if all the query probabilities q_I are equal. The size comparisons should have some safety margin, because the queried symbol image may have slightly different compressed size than the similar symbol image in the document in practice.

In a multiresolution framework the query search can be done over the low-low subband image which has smaller character images, thus smaller sizes for compressed symbols. The query time decreases by 2^{2K} for a K level subband decomposition. Such a speed-up for a query is especially important in large document archives. On a Sun Sparc 10 computer, the query response time for the encoded images with multiresolution TIC is less than half of that required for images encoded by regular TIC. As an example, on a Sun Sparc 10 computer, the query response times are on the order of 0.2 second per page for the compressed with our method, and it is about 0.5 second per page for the data coded with TIC, directly.

We use the textual images in the *NIST Special Database 8* [95] for testing the keyword search and document retrieval performance of the multiresolution TIC method. Ten pages containing Times New Roman fonts are scaled to the same font size and then compressed together using our method with the LTM criterion, to form a single compressed file and the compressed data is tested with arbitrarily selected six key strings consisting of two letter symbols. The queries are made by example images within the documents. Tables 3.1 and 3.2 give the number of correct identifications of the words over all the occurrences of these words. The amount of misses and false identifications are presented in individual columns. When these pages are compressed separately, the overall lossless compression ratio falls from 58.1:1 to 49.2:1, and the corresponding retrieval results are given in Tables 3.3,3.4. These experimental results indicate that the number of missing symbols out of all 61 key strings is 1 for the document of separately coded 10 pages. The total number of false alarms for these key strings is 3. When the TIC is used without subband decomposition, the number of misses is also found to be 1, and the number of false alarms is determined as 1. When the SOM pattern matching criterion is used, the retrieval accuracy is reduced both for the TIC and the multiresolution TIC at the same amount. With the SOM method, the number of misses for the key strings is 5

for both the TIC and the multiresolution TIC. The SOM based pattern matching method is used to illustrate the fact that various pattern matching methods can be incorporated to the multiresolution TIC. As described in Section 3.3.3, the coding performance and the retrieval accuracy of the SOM based method can be improved by increasing the input vector size.

Image	Correct hit					
	"rt"	"zE"	"3V"	"Po"	"va"	"&9"
No4	3/3	3/3	5/6	2/2	0/0	1/2
No5	2/3	2/2	2/2	2/2	2/2	3/3
No6	0/0	0/0	0/0	3/3	0/0	0/0
No8	1/1	3/3	2/2	2/2	2/2	1/1
No10	2/2	0/0	2/2	2/2	0/0	1/1
No11	0/0	0/0	0/0	0/0	2/2	1/1
No12	1/1	1/1	0/0	0/0	0/0	0/0
No15	0/0	0/0	0/0	1/1	0/0	0/0
No17	1/1	0/0	1/1	0/0	1/1	0/0
No20	0/0	1/1	0/0	0/0	0/0	0/0

Table 3.1: *Query results for 10 compressed NIST images for strings "rt", "zE", "3V", "Po", "va", and "&9" at CR = 58:1*

The performance of the TIC method in noisy documents is determined by the behaviour of the character matching criteria. The retrieval of a character is possible only if the noisy character image can still be identified as the representative symbol in the symbol library. Using the software obtained from the University of Maryland WWW server [96], [97], it is observed that with the speckle noise at level 1/10, half of the characters which should be found in the symbol library start to mismatch the symbol library. Furthermore, the compression ratio falls from 39.2:1 to 24:1. The same effect is observed for the TIC method applied without subband decomposition. At the same noise level, 35% of the characters are mismatched, and the compression ratio falls to 23:1. In such cases, a noise eliminating pre-processing is necessary. These results are obtained by the LTM symbol matching method. Due to the inaccuracy of the SOM method described in Section 3.3.3, the compression ratios are slightly lower, and the retrieval errors are 15% higher. The size of the character images

Image	False alarm					
	"rt"	"zE"	"3V"	"Po"	"va"	"&9"
No4	1	0	0	0	0	1
No5	0	0	0	0	1	0
No6	0	0	0	0	0	1
No8	0	0	0	0	0	0
No10	1	0	0	0	0	0
No11	0	0	0	0	0	0
No12	0	0	0	0	0	0
No15	0	0	0	0	0	1
No17	0	0	0	0	0	0
No20	0	0	0	0	0	0

Table 3.2: *False alarm results for 10 compressed NIST images with same key strings at CR = 58:1*

Image	Correct hit					
	"rt"	"zE"	"3V"	"Po"	"va"	"&9"
No4	3/3	3/3	5/6	2/2	0/0	2/2
No5	3/3	2/2	2/2	2/2	2/2	3/3
No6	0/0	0/0	0/0	3/3	0/0	0/0
No8	1/1	3/3	2/2	2/2	2/2	1/1
No10	2/2	0/0	2/2	2/2	0/0	1/1
No11	0/0	0/0	0/0	0/0	2/2	1/1
No12	1/1	1/1	0/0	0/0	0/0	0/0
No15	0/0	0/0	0/0	1/1	0/0	0/0
No17	1/1	0/0	1/1	0/0	1/1	0/0
No20	0/0	1/1	0/0	0/0	0/0	0/0

Table 3.3: *Query results for 10 compressed NIST images for strings "rt", "zE", "3V", "Po", "va", and "&9" at CR = 49:1*

Image	False alarm					
	"rt"	"zE"	"3V"	"Po"	"va"	"&9"
No4	0	0	0	0	0	1
No5	0	0	0	0	1	0
No6	0	0	0	0	0	0
No8	0	0	0	0	0	0
No10	0	0	0	0	0	0
No11	0	0	0	0	0	0
No12	0	0	0	0	0	0
No15	0	0	0	0	0	1
No17	0	0	0	0	0	0
No20	0	0	0	0	0	0

Table 3.4: *False alarm results for 10 compressed NIST images with same key strings at CR = 49:1*

is also critical in noise sensitivity. It is more difficult to make symbol matching over small character images, so the number of decomposition levels should be adjusted according to the size of the character images. It is observed that good coding and retrieval results can be obtained by a single level subband decomposition for 12 point document images scanned at 300dpi.

3.5 TEXTUAL IMAGE COMPRESSION SIMULATION STUDIES

In this section, we present coding examples for various pages of printed text. The first example is a 12pt Times New Roman printed text scanned at 300 dots per inch (dpi) with size 2500x720 pixels. The compressed image consists of bit-streams corresponding to the character library images, the locations of the characters, and their symbol sequences. Most of the bits are allocated to represent the character library image. This image is constructed as a binary image in Method 2 (the method in the literature), and it is constructed as four five-level or two-level, half-size images in Method 1 (proposed method) with

one level decomposition (Fig. 3.7). The compression of these four SL images is investigated in detail in this section.

Experimentally, the compression ratio for separate coding of each of the four SL images is inferior to the compression ratio for the lossless coding of the OL image. As an example, the OL image that we generated by using Method 2 can be compressed to 23824 bits by simple LZ compression. The overall compression ratio (CR) is 63.47 if 23896 bits are allocated for the library image. This compression is the result of the lossy part of the method described in [60]. On the other hand, each of the four SL images are compressed to

ll : 12080 bits
lh : 6981 bits
hl : 8792 bits
hh : 5536 bits
total : 33392 bits.

which corresponds to an overall CR of 49.12. In calculating the overall CR, the bits required for encoding the repetition locations and symbol sequences are included. It can be seen that the proposed approach cannot achieve a higher compression than the previous schemes when used in this form.

A reasonable approach to overcome this problem is to perform quantization on the subband images. As a result of using Haar wavelet on the binary image, the subband images has only five levels and the dilation and ripple effects due to filtering is minimal. These effects are important in our application. We do not want to have a large number of gray levels in the subband images because this affects the compressibility of the image. Furthermore, the dilation and ripple effects make it difficult to perform the character image searches. Another drawback for the ripple dilation and ripple effects is that it is difficult to determine the necessary corner points for the character images being extracted from the subband images when the dilation and ripple area is large.

A couple of quantization schemes have been tested and it is found that quantizing the sub images to three levels do not cause visual degradation. The quantization is performed as follows :
For the ll subband image,

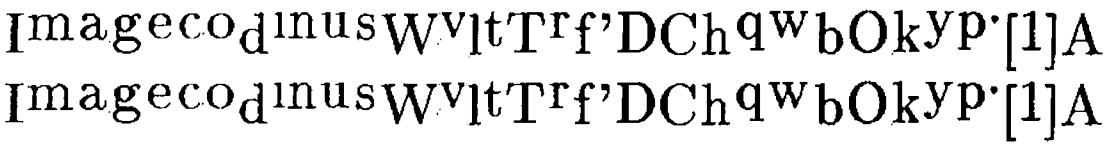


Figure 3.8: *Reconstructed library images before and after quantization*

$Q(0) = 0; Q(1) = 0; Q(2) = 2; Q(3) = 4; Q(4) = 4$
for lh and hl subband images,
$Q(-2) = -2; Q(-1) = -2; Q(0) = 0; Q(1) = 2; Q(2) = 2$
and the hh image is totally ignored.

The original and the reconstructed library image after quantization is shown in Fig. 3.8. In this way, the compression of these sub images yield

ll : 9360 bits
lh : 6192 bits
hl : 6728 bits
hh : 0 bits
total : 22280 bits, CR = 70.51.

This compression is certainly better than the direct compression of the SL images (33392 bits). Although this quantization causes a lossy compression, the number of pixels that differ between the original library and the quantized and synthesized library is only 102 out of the library image with size 846x54 = 46656 pixels. As seen from Fig. 3.8, this effect is not noticeable. After all, the use of the character images in the library for reconstructing the text image is, by itself, a lossy scheme, so this additional loss is not significant for the overall scheme. If the residue images are to be coded for lossless compression, the quantization step should not be used. When the residue images are used to build the exact subband images by inserting the character images, the quantized subband character images will not yield a prefect reconstruction.

Another approach to get the advantage of subband decomposition is to exploit the correlation between the subband images. If the subband images are to be realized at top of each other, they form 3-D data, as indicated in Fig. 3.10. We used the bit-planes of these subband images for appending them to each other. The bit-planes of each subband image is appended at top of

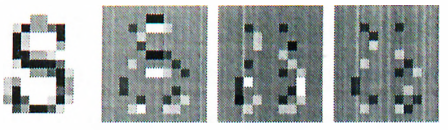


Figure 3.9: *Detail images from four subband library images*

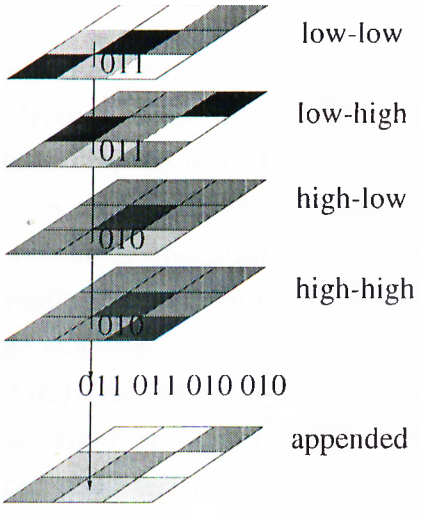


Figure 3.10: *Visualization of appending the bit-planes of subband images*

each other. The total number of bit-planes become 12 with this way. Although this many bit-planes correspond to a high number of pixel values, not all bit combinations are available in the subband images. Furthermore, when the images contain large smooth areas with the same pixel value, the cross band combinations also contain few different combinations. As an example a text image contains large white background areas and groups of black pixel regions. This cross band combination is illustrated in Fig. 3.9, 3.10.

Experimentally, we found out that for the test documents that we used, a maximum number of 17 different bit combinations could occur inside the 12 bit compound image. This is an important result because with this number of levels, the compound image could be compressed to 14296 bits (CR = 102.60) for the test image that we consider throughout this paper.

The same combination method is used for the quantized subband images. Since these quantized subband images contain only three levels, they could be represented with fewer bits, and the total number of bits for the compound image is only 6. The number of bit combinations in this case is 14 which

is not much smaller than 17. This compound image could be compressed to 13904 bits (CR = 104.94). This result may be considered as a trade off between image quality and compression ratio, but since the compression ratio is slightly improved, the previous scheme with non-quantized subband images seems preferable.

By using binary transformations, we obtain two level subband images. The morphological subband decomposition is used for generating binary subband images. The method is based on the work by Egger et al [45], furthermore, the operations are performed in modulo-2, so the resultant subband images came out to be binary. The synthesis operation gives perfect reconstruction. We tested the same bit appending procedure for these four subband images. The bit-plane appended image is a 4-bit (16 level) image and it can be compressed to 14400 bits (CR = 102.00). Although this result is near to the previous result obtained by using the Haar basis, it is interesting not to obtain a better compression ratio. One can consider the saving in computation time (1.3 times faster) as a good point for using the morphological binary subband decomposition.

The overall compression results are given in Table 3.5.

image size = 2500 × 720						
		Haar Basis, Method 1				Nonlinear filter Method 1
	Method 2	Direct	with quant.	with bit-plane append.	with quant. and bit-plane append.	with bit-plane append.
Total num. of bits for symbol lib.	23896	33392	22280	14296	13904	14400
Total num. of bits for locations	4280	3064	3064	3064	3064	3064
Total num. of bits for symbol sequence	184	184	184	184	184	184
Total num. of bits	28360	36640	25528	17544	17152	17648
Compression ratio	63.47	49.12	70.51	102.60	104.94	102.00

Table 3.5: *Textual image compression results: Times New Roman*

The next example is a 12 point Sans-Serif font textual image scanned at a size 2533 x 3370 shown in Fig. 3.11. The compression results are higher in this case because the image size is bigger. In this example, the importance of size about bit allocations for symbol sequence and symbol locations is more noticeable. The effects of subband decomposition over symbol library compression is more or less the same as the Times Roman example. Compression results for the Sans-Serif image with LTM matching criterion are given in Table 3.6. Similar to the previous case, the SOM matching criterion gives comparable, but smaller compression ratios.

Sans-serif font image size = 2700 × 3600					
		Haar Basis, Method 1	Nonlinear filter Method	XOR filter Method	JBIG
	T.I.C.	with bit plane appending	with bit plane appending	with bit plane appending	
Tot. no. of bits for symbol lib.	24421	16121	11022	11358	
Tot. no. of bits for locations	21289	15592	15592	15592	
Tot. no. of bits for symbol sequence	7395	4174	4174	4174	
Tot. no. of lossy bits	53105	35887	30724	30788	
Lossy CR	181.68	270.85	316.37	315.71	
Tot. no. of bits for residue	129568	129568	129568	129568	
Lossless CR	53.21	58.75	60.64	60.62	46.15

Table 3.6: *Textual image compression results: Sans-Serif*

is an important task for summarizing the content of a video. It is usually desired to perform this operation by using the readily produced compressed video data. Most of the video streams are already stored in MPEG-I or MPEG-II form and one should avoid full decompression of the data in order to reduce the computational complexity.

This paper describes a method for selecting key frames by using a number of parameters extracted from the MPEG video stream. The parameters are directly extracted from the compressed video stream without decompression. A combination of these parameters are then used in a rule based decision system. There is very little calculation in both extracting the parameters and the in the decision for key frame rule computation. As a result of these, the overall operation is very quickly performed. The experimental results show that this method can select the distinctive frames of a video stream successfully.

The histogram measure depends on the DC values of the DCT coefficients of each block, and is somewhat more tricky. For intra blocks, the DC value is used directly. For inter blocks, the DC value corresponding to that block is interpolated from the DC values of the reference frame blocks according to the given motion vectors. Typically, the motion vectors point to an 8×8 region which has regions inside four 8×8 regular blocks. The motion compensated DC value is calculated by interpolating the DC values of the four regular blocks at the point indicated by the motion vector.

The thresholds have to be chosen according to the general characteristics of the sequence. In our experiments, we used CIF size (352×288) frames for the MPEG sequence. The frame rate is 8 frames/sec. The motion measure and the thresholds should be modified accordingly if the frame rate is altered because the motion characteristics of the sequence changes according to the frame rate. The Group Of Picture (GOP) structure has the sequence

Figure 3.11: *The test document image - Sans-serif.*

We observe that the binary subband decomposition produce higher compression ratios when the size of the document or the number of textual image pages increase.

The separation of text and graphics parts of the image is very important in achieving high compression ratios. As an example, consider the test images in Fig 3.12. These images were previously compressed using the adaptive decomposition structure, and the PSNRs of 36.35dB and 41.72dB were achieved at CR=8:1. If the text parts are separated from the graphics parts and are encoded to a total (including the compressed textual and graphics images) bitrate of 1bpp, the graphic image part of the first image results in a PSNR of 39.22dB and its textual part is lossless compressed at a ratio of CR=33.10:1, and the graphics image part of the second image results in a PSNR of 44.02dB and its textual part is lossless compressed at a ratio of CR=39.09:1. The overall compression performance of the separate images is higher than the compression performance of the original image without text-graphics separation.

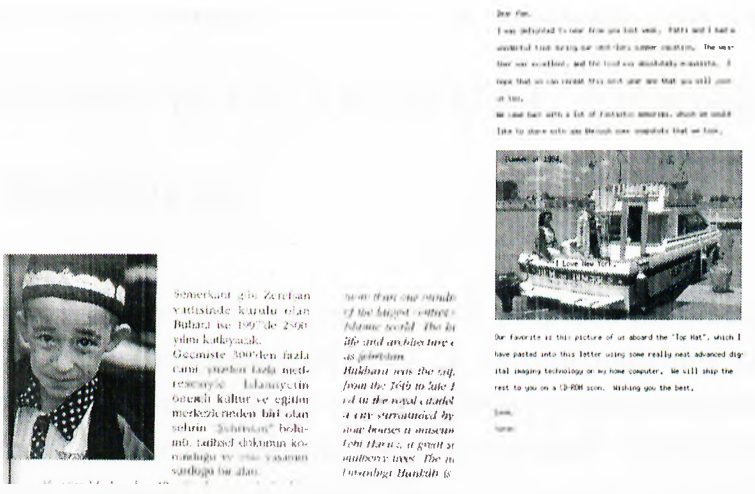


Figure 3.12: *Mixed text-graphics images*

The *NIST Special Database 8* [95] Times New Roman test documents are also compressed with the multiresolution TIC method. This test data contains 10 documents of random ASCII characters. The average lossless compression ratio is obtained as 39.2:1 with the LTM criterion. With the SOM criterion, the CR is 36.4:1. When the regular TIC method is used, the CR is 35.2:1 with the LTM criterion.

In this study, we use the SOM_PAK Program Package developed by Kohonen *et. al* [94]. Due to the restrictions of this software, only the horizontal projections of the character images are used as input vectors. As a result of this reduction in the representation, some inaccuracies occured, yielding less compression ratios and inferior retrieval performance. By using more data in the input vectors, such as vertical projections, etc., the performance can be improved.

The compression speed is also improved in our method. With the introduction of binary subband decomposition prior to TIC, it is experimentally observed that the overall coding time has been reduced by a factor of 1.5 on Sun Sparc 10 and Sun Sparc 20 computers.

3.6 SUMMARY AND POSSIBLE DIRECTIONS FOR TEXTUAL IMAGE COMPRESSION

A wavelet based image compression scheme is proposed for binary textual images. The textual compression is done by pattern matching and collecting method as described in Section 3.1. In addition to the direct implementation of this method, the use of subband images is proposed. The pattern matching and search for constructing the symbol library is carried out over the low resolution images. These symbols obtained from the ll images supply the necessary information to extract the lh, hl, and hh image counterparts. Basically, the exact places and dimensions of these extracted patterns are used for constructing the high-band pattern libraries. Since we use Haar basis, Binary Wavelet Transform or Nonlinear Wavelet Transform, the ripples corresponding to edge locations shift by only one pixel. When the new libraries for the detail images are constructed, the pointers found from the ll image are used. Certainly, the ll image obtained after subband decomposition must be sufficiently clear for easy pattern extraction and matching. The use of a low resolution image improves the search time by a factor of 16. In addition to this, the low resolution part of the text image can be supplied as a first preview page for multimedia applications.

The resultant compression ratio also came out to be superior to the results described in [60]. By using the methods mentioned in Section 3.3, the symbol library images are more efficiently compressed. The results for the test image is shown in the form of a table in the previous section.

For the database search, only the symbol library and the location pointers need to be decompressed. When a keyword is entered, the search can be done if the keyword is itself a text image with the same sized fonts. In this case, the occurrences of the individual characters can be checked first, and then their locations can be tested to see whether their occurrences are in sequential and correct order within a limited space. One other method for keyword check can be possible if the ASCII values of the character images are stored together with

those images. In our test image, we found 35 different symbols to encode. This makes 35 bytes (280 bits) for the ASCII values. The symbol library images could be compressed to about 15000 bits and the total number of bits for the compressed bit stream is about 32500, so this extra data does not decrease the compression significantly. In this case, the keyword can be entered directly from the keyboard, and the search can be done through the ASCII values and the locations of the characters.

In the decoding part, the character images in the library are inserted to the appropriate locations inside each subband image. These appropriate locations are also compressed in the encoding scheme and the locations corresponding to all subband images are the same for the same character image in the library. After the insertion process, the subband residue images can be added to the lossy reconstructed images to obtain a perfect reconstruction after applying the synthesis filters.

The transforms that yield binary subband images are under investigation to achieve higher compression. Appending the bit-planes of the subband images came out to achieve a high compression. When the subband images are binary, the number of bit combinations will is less than the number of bit combinations using multi-level subband images. As a result, both the compression ratio and keyword search speed is higher. An interesting problem is about finding the best number of decompositions for a given font size and scanning resolution. We had experimentally observed that one level decomposition is sufficient to improve compression results for a 11 or 12 point printed text scanned at 300dpi. More experiments can be made to find a rule to determine the best number of decomposition levels for the given font size and scan resolution. The retrieval accuracy should also be considered as a parameter in this determination since each level of subband decomposition reduces the character image size and hence the intelligibility of the character image.

The extraction of parts of the letter images will be considered as a future work. How to extract the basic marks that are overlapping remains to be an open problem. The elimination of repeating marks can be done simultaneously to solve the overlapping mark problem. In this way, the breaking of some basic marks during removing another mark can be avoided.

Figure 3.13: *Letter "b" and its components "l" and "o"*

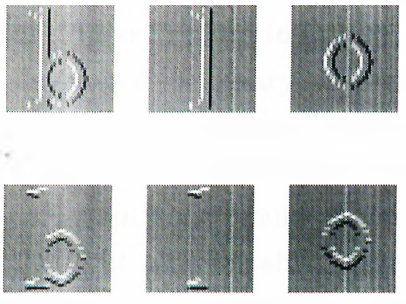


Figure 3.14: *lh and hl subband images of "b", "l" and "o"*

Another problem about extracting these marks is encountered when considering the 5-level subband images. It is even more difficult to find and extract these marks from multi-level images. Furthermore, it may be difficult to consider the problem of overlapping shapes in subband images other than the *ll* image. In that case, the reconstructed image after the synthesis filters must be correct combinations of these basic marks and curves. As an example, consider the letter "b". Suppose we extract two marks from the letter, similar to letters "l" and "o" (Fig. 3.13). The left side of "o" is overlapping with bottom part of "l". If we consider the *lh* and *hl* subband images of "b" and their sub-curves "l" and "o" in Fig. 3.14 , it is evident that there is a problem of obtaining two overlapping *lh* and *hl* marks from the subband images of "b" that corresponds to *lh* subband images of "l" and "o". Some of the edge points for subband images of "l" and "o" do not occur in the subband images of the compound letter "b". This problem is still under research.

As another future work, the compressibility of the mentioned residue images in the subbands can be considered to achieve lossless compression within the subbands.

In the previous chapter, adaptive subband decomposition structures have been developed for decomposing gray tone images. The adaptation algorithm produces real values. However, in this chapter, the subband decomposition

structures produce binary outputs. The idea of signal adapted prediction filters in Fig 2.5 and 2.6 requires arbitrary adaptation as long as the same adaptive structure is used in both analysis and synthesis stages. The idea is to eliminate the predictable parts of the second polyphase component using the first polyphase component and the adaptive prediction filter. Therefore, the adaptation can also produce binary inputs to binary outputs. Although there are not many techniques developed for binary adaptation which is suitable for subband coding, the subject of signal adapted binary subband decomposition is an interesting problem, and it is a suitable subject for a future work. Even more than that, the idea of adaptive subband decomposition in Galois Field - (N) (GF-N) can be worked out and suitable decomposition structures can be developed. The idea of producing statistically independent binary subband signals may be used instead of trying to decorrelate the two signals using least squares kind of algorithms.

Figure 3.15: *The residue image for ll subband*

Chapter 4

SPECIALIZED LIBRARY APPLICATIONS

In this chapter, two types of special library applications are considered. The coding methods described in Chapters 2 and 3 are mostly sufficient for coding arbitrary digitized libraries. However, it is sometimes more appropriate to consider the characteristics of the images in specialized libraries, and modify the coding algorithms, accordingly. In this chapter, we demonstrate our coding algorithms in two types of special libraries, namely the library of Ottoman Document Archives, and the Fingerprint Image Library, with the modifications specific to these types of images. A textual image coding type of method is developed for coding the Ottoman Document Images. The coding of these images differs from the bi-level textual image compression method in the sense that it requires coding of gray-tone images which contain connected script of Arabic letters. The coding of fingerprint images is considered both for bi-level and for gray tone fingerprint images. The adaptive subband decomposition is used for fingerprint images. As an extension of the adaptive subband decomposition developed for gray-tone images, the algorithm is modified to handle binary images. Simulation examples are presented, and compression performances are compared to the methods in the literature.

4.1 CODING OF OTTOMAN DOCUMENT IMAGES

A special class of documents can be summarized as the historical documents. In most cases, the compression of such documents need special elaboration. Ottoman documents are such historical documents which are printed in Arabic letters. In this section, the digital coding of Ottoman printings is considered. The Ottoman archives are within the interests of over thirty countries which were once inside the Ottoman Empire until its breakdown in World War I. Due to the large number of archival documents, it takes significantly long to read and transcribe them. Meanwhile, the documents are deteriorating physically because of storage conditions and time. Digital storing seems like the only way to save these invaluable documents.

4.1.1 COMPRESSION OF GRAY TONE OTTOMAN SCRIPT IMAGES

The Ottoman documents were written with Arabic letters. These letters form a connected compound structure which may correspond to a word or a phoneme. Due to this fact, the method by Witten *et al.* [60] cannot be directly applied to Ottoman or Arabic documents as the concept of character is different in Ottoman or Arabic. Typically, one has to deal with compound structures consisting of a group of letters. Therefore, the matching criterion will be modified according to those compound structures. The modification will be described in detail. Another important difference is that, the text images are gray tone or color images for Ottoman scripts because the marks, shady areas and ink smears in the page are important for a historian.

In our method the compound structure matching is carried out in wavelet domain which reduces the search space and increases the compression ratio. In addition to the wavelet transformation which corresponds to the linear subband decomposition, the nonlinear subband decomposition, similar to the ones

Figure 4.1: *Part of the original document image*

used in normal textual image compression, is used again. The filters in the nonlinear subband decomposition have the property of preserving edges in the low resolution subband image.

The encoding is somehow similar to the textual image coding method. The repetitions of compound structures or letters are found inside the image. This operation is illustrated in Fig. 4.1. The repetitions of letter "waw" are found. Note that the letter can be connected to other compound structures.

Similar to the previous section, the steps of the TIC are carried out in the *ll* subband of the original image (Method 1). Again, we will call the direct use of the textual image compression method on the original image "Method 2".

The difference in extraction of the symbol library for Ottoman documents can be explained as follows. The new symbol extraction scheme should enable us to handle connected text symbols. In compression of documents with isolated characters, a symbol is marked and extracted from the document image. This symbol is then compared with the previously selected symbols in the symbol library. If it matches to any of the characters in the library, only its location

is encoded, otherwise this character image is included in the library. In other words, the extracted single character is only compared to the symbols in the library. In the modified version of the TIC, the document is processed in multiple passes. In Ottoman documents, there are isolated symbols corresponding to single letters as well as long connected symbols which are combination of letters. Usually, the longer symbols include some letters which can also be found separately inside the document. When the smaller isolated symbols are encoded and removed from the document, the longer symbols which entirely contains the considered symbol split into smaller symbols. In other words, we are left with another image in which the occurrences of the considered compound structure image are eliminated. Each of these smaller symbols may also be contained in other connected symbols. Therefore, they will later cause other connected symbols to split further. The removal of symbols from the document image is performed by sliding the symbol image all over the document image and keeping track of the correlation. If the correlation between the symbol and the document image is high at a location, the marked symbol is subtracted from that location of the image, and the location is encoded. When the end of the document is reached during correlation, all the places of the document which contains shapes similar to the marked symbol are removed from the image. In this way, even if the marked symbol is connected to some other symbols inside the document, the similar portions will be extracted, and the longer connected scripts will be broken into smaller pieces. After several such passes, all of the document is encoded.

The extraction of marks from the document image is shown in Fig. 4.3. The resulting symbol library mainly contains basic compound structures, curves and lines that eventually form the Ottoman script. This method could also be applied to the regular printed textual images. However, since the letters are isolated, it is redundant to slide a symbol all over the document image to find similarities.

The compression efficiency is mainly determined by how successful the subband library (SL) images are compressed. Experiments show that the images in the library created from the subband images can be more efficiently compressed than the direct application of Method 2.

When we apply Method 1 to Ottoman document images, we have the same advantages about computational time, retrieval time and compression ratio.

4.1.2 SIMULATION STUDIES FOR OTTOMAN DOCUMENT COMPRESSION

The Ottoman document compression experiments are performed in two ways. In the first case, we performed the textual compression in the original image without any transform or decomposition. In the second case, a subband decomposition applied to the original image, and the textual compression is performed over the sub-images, like in the normal textual image compression described in the previous section.

Similar to the binary textual image compression results, the compression ratio for separate coding of each of the four subband symbol library (SL) images was inferior to the compression ratio for the coding of the original library (OL) image obtained from the original image without subband decomposition. In this work, we used a 1300×1900 textual Ottoman script with 8 bits per pixel which was scanned at 300 dpi. When we do not consider the coding of the residue images, the lossless compression of the OL image results in an overall compression ratio (CR) of 20.02 : 1. On the other hand, the lossless compression of four SL images generated from Haar basis results in an overall CR of 18.13 : 1. However, by simple quantization, this ratio can be increased to 23.77 : 1 without introducing a perceptual difference.

When we consider the nonlinear subband decomposition, the lossy compression of four SL images with quantization results in an overall CR of 24.00 : 1.

Instead of compressing the SL and OL library images in a lossless manner, a lossy scheme like JPEG [8] can be used. The results for JPEG encoding (@ quality setting = 75%) the library images are as follows :

- For direct OL images, total CR = 42.44 : 1

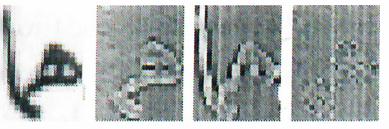


Figure 4.2: *Detail images to show the pixel-wise correlation between subbands*

- For Haar subband decomposition, total CR = 40.78 : 1
- For Haar subband decomposition + quantization, total CR = 48.41 : 1
- For nonlinear subband decomposition, total CR = 50.19 : 1

In order to get the advantage of subband decomposition, we again used the correlation between the subband images. Since the document images mostly have regions of black pixels and a gray tone background, the subband images including the *ll* image are highly correlated with each other.

If the bit-planes of subband images are to be realized at top of each other, they form a 3-D data. In this way, the total number of bit-planes increases. In order to keep the total number of bit-planes reasonable, the SL images are quantized to less number of bits. By taking into account the variances of the subband images, 5 bits are assigned to *ll* subband image, 3 bits are assigned to *lh* and *hl* images and 2 bits are assigned to *hh* image. As a result of this quantization, the appended image becomes a 13-bit image. Although this corresponds to 2^{13} gray levels, the total number of unique gray tones in this bit plane appended image for our test document is only 230. The reason for this is the flatness of the regions inside the textual image. As a matter of fact, there are not many gray tones inside the textual image. The correlation between subband library images is illustrated in Fig. 4.2.

When the appended library image is Lempel-Ziv compressed, the overall CR for the document is 59.74:1 for Haar subband decomposition and 60.10:1 for nonlinear subband decomposition. The residue images are, again, not coded.

The residue images are rather difficult to compress separately. Since these images look like a noise image, the lossless encoding of the residue images decreases the compression ratio drastically. Two approaches for lossy encoding the residue images are JPEG compression and Lempel-Ziv encoding the quantized version of the residue image. The result for JPEG was visually better than the result for Lempel-Ziv encoding the quantized version of the residue

image. With the addition of the compression bits from the residue image, the compression ratio is decreased. The decrease in the compression ratio is valid for all the methods described above, including the original textual compression and the compression after subband decomposition.

The final compression ratio results are given in Tables 3.2 and 3.3. The portion of the original and decoded image after using Haar subband decomposition and bit-plane appending the SL images are given in Fig.s 4.4 and 4.5. The residue images are JPEG encoded in this reconstructed image.

4.1.3 SUMMARY AND EXPERIMENTAL RESULTS FOR OTTOMAN DOCUMENT COMPRESSION

We presented a method to compress Ottoman document images which are composed of Arabic letters. The steps for compression is the same as the steps for binary textual image compression. In addition to the steps described in Section 3.1, the residue images at each subband is JPEG encoded and the subband library images are quantized to different number of levels.

The matching criteria for finding the repetitions of the compound structures is based on a modified version of the pattern matching criteria of the original TIC as described in Sec. 4.1.2. In this way, smaller compund structures which may be included in larger compound structures may be extracted from the textual image 1.3.

Figure 4.3: *Two compound structures.*

While extracting the library pattern from the text image, the background level of the text image is preserved. As a result mostly the background gray tone remains in the residue image and the dynamic range decreases. Compression of this residue image is very efficient.

The decoding procedure starts from the subband images. The library subimages are inserted to the locations in the pointer list and the decoded residue images corresponding to each subband image are added to obtain a good approximation to the subband decomposition of the original image. These subband images are still approximations, because the residue images as well as the library images are encoded in a lossy manner. The last step is synthesizing the sub-band images by the synthesis filter bank of the wavelet transformation or the nonlinear subband decomposition.

This method is suitable for fast database search if the properties of each extracted compound structure are supplied. The possibility of a compound structure to be concatenated with another compound structure is a grammatic property, and this possibility for each extracted compound structure must be stored inside the coded bit-stream in order to enable keyword searches.

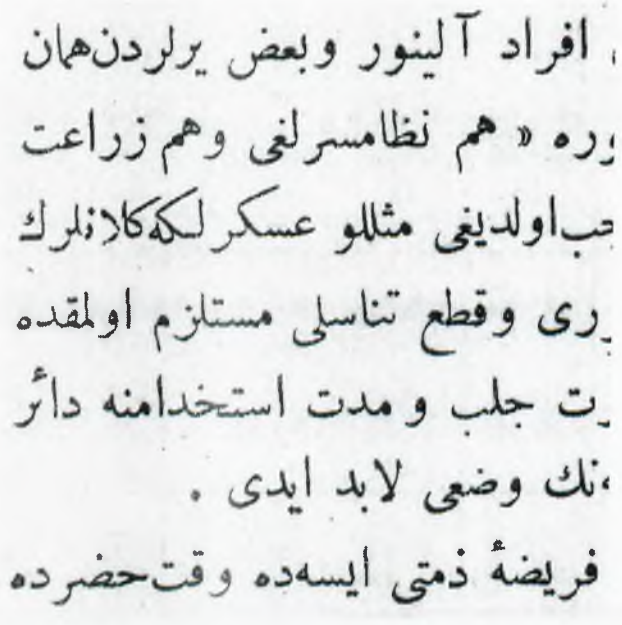


Figure 4.4: *Part of the original document image*

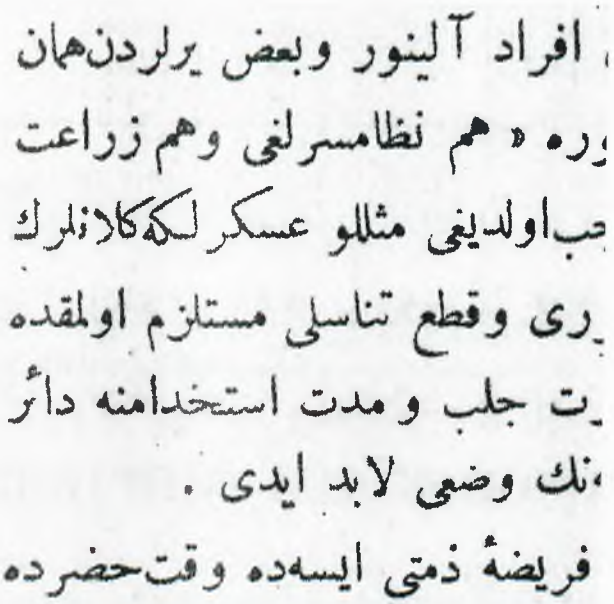


Figure 4.5: *Reconstructed document image*

	original image		Haar subband decomposition				
	lossless OL	JPEG OL	lossless SL	quant. SL	JPEG SL	JPEG of quant. SL	Bit-plane append SL
Number of bits for library image	970216	451000	1077904	819304	472552	396184	318768
CR w/o residue	20.02	42.44	18.13	23.77	40.78	48.41	59.74
CR with JPEG residue	16.65	29.59	15.08	18.80	28.05	31.46	35.88

Table 4.1: *Ottoman document compression results - part 1.*

	nonlinear subband decomposition				
	lossless SL	quantized SL	JPEG SL	JPEG of quantized SL	Bit-plane appended SL
Number of bits for library image	1033194	811336	445200	381704	316784
CR w/o residue	18.96	24.00	43.22	50.19	60.10
CR with JPEG residue	15.62	18.94	29.18	32.20	36.01

Table 4.2: *Ottoman document compression results - part 2.*

4.2 FINGERPRINT IMAGE COMPRESSION

In this section, we use the adaptive perfect reconstruction (PR) polyphase filter bank structures for compressing fingerprint images. The use of adaptive filter banks leads to higher compression results for images containing sharp edges such as fingerprint images. The fingerprint image compression is an important problem due to the high amount of fingerprint images in databases [99]. For example, the FBI database contains 30 million sets of fingerprints. The digitization of the database speeds up the querying and classification operations, but the storage problem is not solved until good compression algorithms are developed for fingerprint images. Recently, FBI selected a wavelet/scalar quantization algorithm for coding 8-bit scanned fingerprint images and for typical images, a compression ratio of 10:1 to 25:1 is achieved by this technique [22]. We experimentally observed that the EZW [56] based coder outperforms the wavelet/scalar quantization method, therefore, we compared our method with the EZW.

The previous work was concentrated on the compression of 8-bit gray level images [22], [99]- [101], however, in many cases 8-bit resolution for fingerprint images is unnecessary because the fingerprint information is essentially binary (although, due to ink smearing, scanning noise etc. higher resolution digitization may be necessary in some cases). If a binary level fingerprint image show the so-called "minutiae" details such as core and delta points which are necessary for identification then there is no need to represent this image in

Figure 4.6: *Lasso - tented arch fingerprint.*

8-bits. There are some important classes and subclasses of fingerprint images, such as Wirbel and Lasso classes and whorl, twin loop, arch, tented arch, right loop and left loop subclasses. The compression ratio can be increased as long as the classification and identification of the fingerprint is possible. A sample Lasso class tented arch subclass fingerprint is shown in Figure 4.6.

In this section, algorithms for coding binary and 8-bit fingerprint images are presented. It is experimentally observed that adaptive subband decomposition performs well for binary and gray-valued fingerprint images.

4.2.1 GRAY TONE FINGERPRINT IMAGE COMPRESSION

Gray tone fingerprint images are very suitable for the adaptive filter bank structures presented in Chapter 2. They consist of sharp edges of the fingerprint curves.

The compression of gray tone fingerprint images is performed by applying the adaptive subband decomposition followed by an Embedded Zero Tree (EZT) type coder [56], [58]. The level of decomposition is determined by the size of the image. The OS adaptive filters in Chapter 2 produces slightly better results than the FIR adaptive filters for the fingerprint images. Two level adaptive subband decomposition followed by three more levels of regular wavelet decomposition is used for 256×256 images. Experimental results show that

our decomposition scheme with the EZT or SPIHT coder outperforms the traditional Embedded Zerotree Wavelet (EZW) coder [56] and the wavelet SPIHT coder [58] in which a fixed wavelet decomposition is performed first, and the coefficients are compressed by the EZT or SPIHT method. Experimental results of gray tone fingerprint image compression are presented in Section 4.2.3.

4.2.2 BINARY FINGERPRINT IMAGE COMPRESSION

The compression of binary fingerprint images need more elaboration. The straight application of the adaptive subband decomposition followed by the EZT or SPIHT coder increases the dynamic range of the reconstructed image. To avoid this increase in the range, a modified decomposition and quantization scheme to keep the decoded signal in the binary range is developed.

The LMS algorithm usually produces non-binary coefficients, so the filtered signal is not binary in general. Therefore, a quantization to binary levels is necessary for the reconstructed image. Another problem is that, for the black regions of a fingerprint image where most of the pixel values are zero, the LMS filter may become unstable. In order to avoid this situation, the LMS algorithm is disabled in the regions where the all the pixels are either black or white, and the prediction value is set to zero (one) for black (white) regions. In the flat regions (black or white) of a fingerprint image, the high band subsignal becomes zero. The prediction errors occur in the transition regions which correspond to the edge portions of the fingerprint image. As a result, in Fig. 2.5, high pass signal $x_h(n)$ contains non-zero elements only in transition regions. This shows that the low and high band signals are completely decorrelated by the adaptive subband decomposition scheme. At both encoder and decoder sides, the filter bank switches between the adaptive and fixed prediction filter banks depending on the homogeneity of black or white pixels in the region of support in $\tilde{\mathbf{x}}_n$ (Eq. 2.22), synchronously. Therefore, PR property is retained. Finally, the reconstructed image is quantized to binary with a threshold. It is observed that the quantization of these subband signals does not produce ringing effects

in the decoded image. The only visible effect in the reconstructed image is an occasional shift in the edge region which separates the black and white regions.

4.2.3 FINGERPRINT COMPRESSION RESULTS

In our simulation studies, we used 20 fingerprint images taken from the National Institute of Standards and Technology (NIST) database examples for compression. All of the fingerprint images used are 8 bit 256×256 images. For the binary image simulation studies, we quantized these images to binary with an appropriate threshold level. Examples of these images are shown in Fig. 4.7.

In Fig. 4.8, two coded versions of a gray-tone fingerprint image are shown. For this fingerprint image, our nonlinear adaptive algorithm followed by the SPIHT coder accomplished a compression PSNR of 28.46dB at 1bpp, whereas the well known EZW and SPIHT coders [56], [58] have PSNRs of 27.86dB and 28.35dB13.0:1 at the same bit rate, respectively. Furthermore, the ringing effects present in the EZW and SPIHT coded image are eliminated with the use of our method.

We also compare the average PSNRs obtained by the adaptive filter bank algorithm to the results obtained by a SPIHT coder for a set of 20 gray tone fingerprint images. According to the test results, at 1bpp level, the average PSNRs over these images obtained by SPIHT and adaptive filter bank method are 28.26 and 28.36 dB, respectively. We observe that the adaptive filter bank produces the best compression results.

In Fig. 4.9, the compressed version of Figure 4.7 with CR=13.92:1 is shown. The coded image preserves the discriminating features of fingerprint images such as core and delta points. The JBIG standard compresses the same image only to CR=5.88:1.

We compare the average compression ratios obtained by the adaptive filter bank algorithm to the results obtained by a SPIHT wavelet coder for a set of 20 binary fingerprint images. According to the test results, at the same

high visual quality level (35dB PSNR), the average CRs over these images obtained by SPIHT and adaptive filter bank method are 13.66:1 and 13.80:1, respectively. The JBIG lossless compression standard compresses only at a ratio of 6.0:1.

The coding results for the set of binary fingerprint images are in parallel with the gray tone image coding results. Improvements in compression performance are obtained due to the use of the adaptive filter bank.

Figure 4.7: *Two fingerprint images. Left: binary, right: gray tone*

Figure 4.8: *Reconstructed images at 1bpp(left) and 0.5bpp(right).*

Figure 4.9: *Reconstructed binary fingerprint image at CR=13.92:1.*

Chapter 5

CONCLUSIONS

In this work, image coding algorithms for digitized library archives are developed. The framework of a digitized library consists of a set of tools for coding images of different content. For example, the adaptive subband coding of images is suitable for gray tone or color images that contain sharp edges or graphics. Usually, such images include some pages in the books and the images on the book covers. On the other hand, the pages of books mainly consist of bi-level text images. Therefore, these images are coded using textual image compression methods.

For the compression of gray tone images that may have sharp transition regions such as subtitles, text, or graphics, an adaptive subband decomposition based image compression method is developed. This method improves compression performance over conventional fixed wavelet based compression methods. Furthermore, the ringing effects which may arise in sharp transition regions are essentially eliminated with the adaptive scheme. Since the proposed method is based on subband decomposition, multiresolution viewing capability is available. Although the decomposition algorithm is proposed for images with sharp edges like subtitles, graphics, or text, the experimental results show that this algorithm also performs very good for regular gray tone images. The extension of these algorithms to color images are done in a straightforward

manner. Typically, Y-U-V components of color images are encoded using the algorithm developed for gray tone images.

The basic idea of the adaptive subband decomposition is to let the polyphase filter bank of the encoder and the decoder adapt to the characteristics of the input signal in a synchronized manner. The adaptation of the filter bank at the encoder is performed by an LMS type algorithm to update the coefficients of the prediction filter which predicts one of the polyphase components from the other. The LMS update requires the input signal, which is one of the polyphase components, and the prediction error. Since these two signals are sent to the decoder side, the decoder can also update the synthesis filter bank with the same LMS algorithm, and with the same signals. In this way, perfect reconstruction is possible. This class of adaptive perfect reconstruction filter banks is also extended to two or higher dimensions in a straightforward manner. The filter banks are also extended to perform multi-channel decompositions for general decomposition purposes. The significance of this signal adapted decomposition is its ability to change the prediction filter bank coefficients when the input signal characteristics change. In this respect, the algorithm differs from the previous work in which the main idea is to find the optimum decomposition filter bank for a specific input signal.

The issue of bi-level textual image compression is considered for the coding of typical pages which contain printed text. A method that exploits the redundancy of repeating characters in a printed text is integrated with a binary subband decomposition based coder. The binary subband decomposition (BSD) methods are based on linear and nonlinear operations that gives binary outputs to binary input images. The BSD concept is compared to the Binary Wavelet Transform (BWT) concept of Swanson *et al.*, and it is found out that implementation of specific BWT's which require matrix multiplications can be performed by the equivalent BSD's which require simple subband filtering operations. The BSD framework is also extended to a more general Nonlinear BSD (NBSD) structure, in which nonlinear binary operators can be used without disturbing the perfect reconstruction property of the filter bank.

The previously used methods to compress textual images are based on an algorithm which is specifically called the Textual Image Compression (TIC)

algorithm [60]. This algorithm encodes the repeated characters and their repeating locations. In the literature, there are some methods to perform pattern matching for extracting the repeating characters. In our simulation studies, these extraction and pattern matching methods are compared to a new method based on a SOM neural network model. The former methods choose the first characters in the raster scan order of the textual image as representative characters and look for the repetitions of characters which are similar to them, whereas the SOM based method finds the optimum representative symbol of a group of character images to put in the symbol library.

These two algorithms are integrated in the textual image coder to form the multiresolution TIC. The textual image is processed by a binary subband decomposition filter bank, and the TIC algorithm with various pattern matching criteria is applied to the subband images. The compression ratios obtained in this way are higher than the compression ratios obtained by directly using TIC over the original image, without subband decomposition. Experimentally, we also obtained compression ratios for textual images higher than the JBIG compression standard. As a bi-product, the proposed compression scheme enables multiresolution viewing and fast keyword search by organizing the locations of the character images in the compressed domain. The experimental results obtained for the NIST database images show that our method is suitable for keyword search.

Two specialized library applications are considered as potential uses of the coding algorithms described in Chapters 2 and 3. One of the specialized libraries consists of document images of Ottoman archives. Due to the historical importance of these document images, they should be coded in gray tone. Furthermore, the textual image compression character extraction method is not suitable for the Ottoman script images which are composed of connected Arabic letters. As a result, the pattern matching and character extraction algorithms in the multiresolution TIC is modified to handle the Ottoman document images which mainly contain connected scripts. With this modification, the experimental results of multiresolution TIC are presented. Similar to the results obtained for regular textual images, the multiresolution compression methods perform better than the compression methods applied to the original image, without subband decomposition. The next specialized library consists of gray

tone and binary fingerprint images. The coding of binary fingerprint images is a very important problem due to the large amount of images in criminal databases. The adaptive subband decomposition method described in Chapter 2 is used for gray tone fingerprint image compression, and simulation results are presented. For the purpose of binary fingerprint image compression, the adaptive decomposition structure is modified to handle binary inputs. The modification consists of a switching adaptive/fixed prediction filter bank in the subband decomposition stage. In order the LMS adaptive algorithm not to diverge in regions containing all zeros, the filter bank is switched to fixed prediction filters in such regions.

These two specialized library applications, as well as the regular digitized library image coding applications show that the two methods proposed in this thesis, namely the multiresolution TIC and the adaptive subband decomposition have immediate and potential uses in a variety of image coding applications.

Appendix A

Derivation of inequality 3.21

If the expected search length C' corresponding to the listing ξ' is less than expected search length C corresponding to the listing ξ, then

$$\frac{L_{\alpha_{i+1}}}{q_{\alpha_{i+1}}} < \frac{L_{\alpha_i}}{q_{\alpha_i}} \tag{A.1}$$

where the two ordered sets, ξ and ξ' are defined as :

$$\xi = \ldots, \alpha_i, \alpha_{i+1}, \ldots, \alpha_N \tag{A.2}$$

$$\xi' = \ldots, \alpha_{i+1}, \alpha_i, \ldots, \alpha_N \tag{A.3}$$

and L_{α_i} is the compressed size of the α_i character image. The query probability of the image α_i is denoted by q_{α_i} and the probability of occurrence of the character α_i in the document is denoted by p_{α_i}, which will be called the symbol probability. The parameter $\hat{p}_{\alpha_i}$ is given by :

$$\hat{p}_{\alpha_i} = (p_{\alpha_i} - \sum_{j=1}^{i-1} q_{\alpha_j} p_{\alpha_i \alpha_j}) \tag{A.4}$$

and $p_{\alpha_i \alpha_j}$ is the probability that the character images α_i and α_j both exist in the document.

We have

$$C = E[\text{Search Length}] = \sum_{i=1}^{N} L_{\alpha_i} \hat{p}_{\alpha_i} = \sum_{i=1}^{N} L_{\alpha_i} (p_{\alpha_i} - \sum_{j=1}^{i-1} q_{\alpha_j} p_{\alpha_i \alpha_j}) \tag{A.5}$$

Suppose k^{th} and $k+1^{st}$ symbols are two adjacent elements. In the listing ξ, k^{th} element comes before $k+1^{st}$ element and in the listing ξ', $k+1^{st}$ element comes just before the k^{th} element at the same location. The expected search length for ξ is :

$$C = \sum_{i=1}^{N} L_{\alpha_i}(p_{\alpha_i} - \sum_{j=1}^{i-1} q_{\alpha_j} p_{\alpha_i \alpha_j}) \tag{A.6}$$

$$= \ldots + L_{\alpha_k}(p_{\alpha_k} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_k \alpha_j})$$

$$+ L_{\alpha_{(k+1)}}(p_{\alpha_{(k+1)}} - \sum_{j=1}^{k} q_{\alpha_j} p_{\alpha_{(k+1)} \alpha_j}) + \ldots \tag{A.7}$$

$$= \ldots + L_{\alpha_k}(p_{\alpha_k} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_k \alpha_j})$$

$$+ L_{\alpha_{(k+1)}}(p_{\alpha_{(k+1)}} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_{(k+1)} \alpha_j} - q_{\alpha_k} p_{\alpha_{(k+1)} \alpha_k}) + \ldots \tag{A.8}$$

and the expected search length for ξ' is :

$$C' = \sum_{i=1}^{N} L_{\alpha_i}(p_{\alpha_i} - \sum_{j=1}^{i-1} q_{\alpha_j} p_{\alpha_i \alpha_j}) \tag{A.9}$$

$$= \ldots + L_{\alpha_{(k+1)}}(p_{\alpha_{(k+1)}} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_{(k+1)} \alpha_j})$$

$$+ L_{\alpha_k}(p_{\alpha_k} - \sum_{j=1}^{k} q_{\alpha_j} p_{\alpha_k \alpha_j}) + \ldots \tag{A.10}$$

$$= \ldots + L_{\alpha_{(k+1)}}(p_{\alpha_{(k+1)}} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_{(k+1)} \alpha_j})$$

$$+ L_{\alpha_k}(p_{\alpha_k} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_k \alpha_j} - q_{\alpha_{(k+1)}} p_{\alpha_k \alpha_{(k+1)}}) + \ldots \tag{A.11}$$

If $C' < C$ then

$$(\ldots + L_{\alpha_{(k+1)}}(p_{\alpha_{(k+1)}} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_{(k+1)} \alpha_j})$$

$$+ L_{\alpha_k}(p_{\alpha_k} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_k \alpha_j} - q_{\alpha_{(k+1)}} p_{\alpha_k \alpha_{(k+1)}}) + \ldots) <$$

$$(\ldots \quad + L_{\alpha_k}(p_{\alpha_k} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_k \alpha_j})$$

$$+ L_{\alpha_{(k+1)}}(p_{\alpha_{(k+1)}} - \sum_{j=1}^{k-1} q_{\alpha_j} p_{\alpha_{(k+1)}\alpha_j} - q_{\alpha_k} p_{\alpha_{(k+1)}\alpha_k}) + \ldots) \quad \text{(A.12)}$$

Cancelling the equal terms, we have

$$(-L_{\alpha_k}(q_{\alpha_{(k+1)}} p_{\alpha_k \alpha_{(k+1)}})) < (-L_{\alpha_{(k+1)}}(q_{\alpha_k} p_{\alpha_{(k+1)}\alpha_k})) \quad \text{(A.13)}$$

Notice that $p_{\alpha_{(k+1)}\alpha_k}$ and $p_{\alpha_k \alpha_{(k+1)}}$ are the same, so

$$\frac{L_{\alpha_{k+1}}}{q_{\alpha_{k+1}}} < \frac{L_{\alpha_k}}{q_{\alpha_k}} \quad \text{(A.14)}$$

Bibliography

[1] NSF Workshop on Visual Information Management Systems, Ed. Ramesh Jain.

[2] R. K. Srihari, "Combining Text and Image Information in Content-based Retrieval," *Proc. IEEE ICIP'95,* Washington D.C., vol. I, pp. 326-329, October 1995.

[3] H. K. Huang, O. Ratib, A. R. Bakker, and G. Witte (Ed.), *Picture archiving and communication systems (PACS) in medicine*, Springer-Verlag, Berlin Heidelberg, 1991.

[4] J. J. D'Lugin, R. L. Hill, R. G. Jost, A. P. Reuter, and J. B. Zimmerman, "Design considerations for a picture archive and communication system (PACS) display station," *IEEE Engineering in Medicine and Biology Society 10th Annual International Conference*, pp. 489-90, 1988.

[5] S. Bhagat, "Multimedia basics for today and tomorrow," *The NCR Journal,* vol. 4, no. 2, pp. 38-45, December 1990.

[6] T. Arndt, "A survey of recent research in image database management," *1990 IEEE Workshop on Visual Languages, pp. 92-97,* University of Pittsburgh, Knowledge Systems Institute, IEEE Computer Society Press, October 1990.

[7] M. D. Swanson, "Issues in image databases: coding for content-based browsing and retrieval, data hiding, and copyright protection," Ph.D Thesis Proposal, Dept. of Electrical Eng., Univ. of Minnesota, Minneapolis, MN, June 3, 1996.

[8] W. B. Pennebaker and J. L. Mitchell, JPEG: Still Image Data Compression Standard, New York, NY: Van Nostrand Reinbold, 1993

[9] ISO/IEC IS 11172:1992, "Information Technology - Generic Coding of Moving Pictures Associated Audio for Digital Storage Media up to about 1.5 Mbit/s (MPEG)", 1992.

[10] ISO/IEC CD 13818-2: "Generic Coding of Moving Pictures and Associated Audio," Nov. 1993.

[11] L. Chiariglione, "The development of an integrated audiovisual coding standard: MPEG," *Proc. IEEE,* vol. 83, no. 2, pp. 151-157, Feb. 1995.

[12] A. K. Jain , *Fundamentals of Digital Image Processing*, Prentice - Hall, 1989

[13] M. D. Swanson, S. Hosur and A. H. Tewfik, "Image Coding for Content-Based Retrieval," in *Proc. 1996 SPIE Conf. on Visual Comm. and Image Proc.,* vol. I, pp.1-12, Orlando, FL, 1996.

[14] G. Kutlu, B. Draper, J. E. B. Moss, and E. Riseman, "Support Tools for Visual Information Management," *Fifth Symposium on Document Analysis and Information Retrieval,* pp.101-112, (SDAIR 96), Las Vegas, NV, April 1996.

[15] B. Cahoon, and K. McKinley, "An Architecture for Distributed Information Retrieval," *Proceedings of the 19th International Conference on Research and Development in Information Retrieval* (SIGIR 96), pp. 110-118, Zurich, Switzerland, 1996.

[16] R. Manmatha, "Indexing and Retrieval Research at the Center for Intelligent Information Retrieval," *Proceedings of Symposium on Document Image Understanding* SDIUT97, Annapolis, MD, April 30, 1997.

[17] W. Lehnert, "Automating the Construction of a Hypertext System for Scientific Literature," *Proceedings of the AAAI Workshop on Communicating Scientific and Technical Knowledge,* pp. 17-23, 1992.

[18] D. Aronow and F. Feng, "Ad-Hoc Classification of Electronic Clinical Documents," in *D-Lib Magazine,* an online publication, January 15, 1997.

[19] M. Hirsch and D. Aronow, "Suggesting Terms for Query Expansion in a Medical Information Retrieval System," *Proceedings of the 19th Annual*

Symposium on Computer Applications in Medical (SCAMC). JAMIA, p. 965, 1995.

[20] M. Maybury, A. Merlino, and J. Rayson, "Segmentation, Content, Extraction, and Visualization of Broadcast News Video using Multistream Analysis," *AAAI Spring Symposium*, Stanford, CA, 1997.

[21] J. W. Woods, Ed., *Subband Image Coding*, Kluwer, 1991.

[22] G. Strang, *Wavelets and filter banks*, Wellesley - Cambridge Press, Wellesley, MA, 1996.

[23] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ, Prentice Hall, 1983.

[24] S. Haykin, *Adaptive Filter Theory*, Englewood Cliffs, NJ, Prentice Hall, 1986.

[25] B. Widrow, S. D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ, Prentice Hall, 1985.

[26] S. Hosur, A. H. Tewfik, "Wavelet domain adaptive FIR filtering," to appear in *IEEE Transactions on Signal Processing*, 1996.

[27] N. Erdöl, F. Başbuğ, "Performance of wavelet transform based adaptive filters," *Proc. IEEE International Conference on ASSP* vol. III, pp. 500-503, Minneapolis, April 1993.

[28] S. Attallah, M. Najim, "On the convergence enhancement of the wavelet transform based LMS," *IEEE International Conference on ASSP*, pp. 973-976, Detroit, Michigan, May 1995

[29] N. Himayat and S. A. Kassam, "A Structure for Adaptive Order Statistic Filtering,", *IEEE Transactions on Image Processing*, Vol. 3, pp.265-280, May 1994.

[30] Ö. N. Gerek, M. N. Gürcan, and A. E. Çetin, "Frequency Band Characteristics of Tree-Structured Filter Banks," *Electronics Letters*, Vol 32, No 8, pp. 724-726, 11 Apr. 1996.

[31] Y. Higa, H. Ochi, S. Kinjo, "A subband adaptive filter with a variable analysis filter bank," *IEEE International Conference on ASSP*, pp. 2309-2312, Munich, Germany, April 1997.

[32] P. Moulin, K. Ramchandran, and V. Pavlovic, "Transform Image Coding Based on Joint Adaptation of Filter Banks and Tree Structures," *IEEE International Conference on Image Processing* (Lausanne, Switzerland), Sept. 1996.

[33] A. H. Tewfik, D. Sinha, and P. Jorgensen, "On the Optimal Choice of a Wavelet for Signal Representation," *IEEE Transactions on Information Theory,* Vol. 38, No. 2, pp.747-765, March 1992.

[34] P. Desarte, B. Acq, and D. T. M. Slock, "Signal-Adapted Multiresolution Transform for Image Coding," *IEEE Transactions on Information Theory,* Vol. 38, No. 2, pp.897-903, March 1992.

[35] P. Moulin and M. K. Mihcak, "Theory and Design of Signal-Adapted FIR Paraunitary Filter Banks," *IEEE Trans. Signal Processing*, Vol. 46, No. 4, pp. 920-929, April 1998.

[36] A. Kirac and P. P. Vaidyanathan, "Theory and Design of Optimum FIR Compaction Filters," *IEEE Trans. Signal Processing*, Vol. 46, No. 4, pp. 903-919, April 1998.

[37] R. R. Coifman and M. V. Wickerhauser, "Entropy-Based Algorithms for Best Basis Selection," *IEEE Transactions on Information Theory*, Vol. 38, No. 2, pp.713-718, March 1992.

[38] V. K. Goyal, J. Zhuang, M. Vetterli, and C. Chan, "Transform Coding Using Adaptive Bases and Quantization," *IEEE International Conference on Image Processing* (Lausanne, Switzerland), Sept. 1996.

[39] R. L. Claypoole, Jr., R. G. Baraniuk, Robert D. Nowak, "Adaptive wavelet transforms via lifting," *Proc. ICASSP'98,* Seattle, WA, May 12-15, 1998.

[40] W. Sweldens, "The Lifting Scheme: A new philosophy in biorthogonal wavelet constructions," In A. F. Laine and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pp. 68-79, Proc. SPIE 2569, 1995.

[41] R. Claypoole, G. Davis, W. Sweldens, and R. Baraniuk, " Nonlinear wavelet transform for image coding," proceedings of the 1997 Asilomar conference on Signals, Systems, and Computers, 1997.

[42] I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps," *Journal of Fourier Analysis and Appl.*, Vol. 4, Nr. 3, pp. 247-269, 1998.

[43] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *J. of Appl. Comp. Harm. Anal.*, vol. 3, no. 2, pp. 186-200, 1996.

[44] M. Smith and T. Barnwell, "The design of digital filters for exact reconstruction in subband coding," *Trans. on Acoustics, Speech, and Signal Proc., ASSP-34(3),* pp. 434-441, June 1986.

[45] O. Egger, W. Li, and M. Kunt, "High Compression Image Coding Using an Adaptive Morphological Subband Decomposition," *Proceedings of IEEE,* vol. 83, no. 2, pp.272-287, February 1995.

[46] F. J. Hampson and J. C. Pesquet, "A nonlinear subband decomposition with perfect reconstruction," *IEEE International Conference on Image Processing* (Lausanne, Switzerland), 1996.

[47] O. Arıkan, A. E. Çetin, Engin Erzin, 'Adaptive Filtering for non-Gaussian stable processes,' *IEEE Signal Processing Letters*, vol. 1, No. 11, pp. 163-165, November 1994.

[48] G. Aydın, O. Tanrıkulu, A. E. Çetin, "Robust least mean mixed norm adaptive filtering algorithms for α-stable random processes," *IEEE- ISCAS'97,* Hong Kong, June 1997.

[49] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Transactions on Signal Processing,* vol. 40, pp. 1633-1642, July 1992.

[50] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-Type algorithm: Analysis and Simulations," *IEEE Transactions on Signal Processing,* vol. 45, No. 3, pp. 631-639, March 1997.

[51] N. J. Bershad, "On Weight Update Saturation Nonlinearities in LMS Adaptation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 4, pp. 623-630, April 1990.

[52] P. Salembier, "Adaptive rank order based filters," *EURASIP Signal Processing*, 27(1):1-25, 1992.

[53] S-M. Phoong, C. W. Kim, P.P Vaidyanathan, R. Ansari, "A new class of two channel biorthogonal filter banks and wavelet bases," *IEEE Transactions on Signal Processing*, Vol.43, No.3, pp. 649-665, March 1995.

[54] N. S. Jayant, P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ, Prentice Hall, 1984.

[55] A. Karaş, B. S. Yarman, "A New Approach for Constructing Base Functions via Gram Schmidt Orthogonalization Method," *Workshop on Signal Processing*, Istanbul Üniversitesi Mühendislik Fakültesi, 1994.

[56] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445 - 3462, Dec. 1993.

[57] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transforms," *IEEE Transactions on Image Processing*, vol.1, pp. 205-220, April 1992.

[58] A. Said and W. A. Pearlman, "An Image Multiresolution Representation for Lossless and Lossy Image Compression," *IEEE Transactions on Image Processing*, vol. 5, pp. 1303-1310, Sept. 1996.

[59] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*, Englewood Cliffs, NJ: Prentice Hall, 1990.

[60] I. H. Witten, T. C. Bell, Hugh Emberson, Stuart Inglis, and Alistair Moffat, "Textual Image Compression: Two-Stage Lossy/Lossless Encoding of Textual Images," *Proceedings of the IEEE*, Vol. 82, No.6, June 1994.

[61] I. H. Witten, A. Moffat, T. C. Bell, "Managing Gigabytes," *ITP Publishing Co.*, 1994.

[62] A. Moffat, "Two level context based compression of binary images," *Proc. IEEE Data Compression Conf.*, J. A. Storer and J. H. Reif Eds. Los Alamitos, CA; IEEE Computer Society Press, pp.382-391, 1991.

[63] R. N. Ascher and G. Nagy, "A means for achieving a high degree of compaction on scan-digitized printed text," *IEEE Trans. Comput.*, Vol. C-23, No. 11, pp. 1174-1179, Nov. 1974.

[64] W. K. Pratt, P. J. Capitant, W. H. Chen, E. R. Hamilton, and R. H. Wallis, "Combined symbol matching facsimile data compression system," *Proc. IEEE* vol. 68, no. 7, pp. 786-796, July 1980.

[65] M. J. J. Holt and C. S. Xydeas, "Recent developments in image data compression for digital facsimile," *ICL Tech. J.*, pp. 123-146, May 1986.

[66] M. J. J. Holt, " A fast binary template matching algorithm for document image data compression," in *Pattern Recognition*, J. Kittler Ed., Berlin, Germany, Springer Verlag, 1988.

[67] O. Johnsen, J. Segen, and G. L. Cash, "Coding of two-level pictures by pattern matching and substitution," *Bell Syst. Tech. J.*, vol. 62, no. 8, pp. 2513-2545, May 1983.

[68] CCITT Draft Recommendation T.82 ISO/IEC Draft International Standard 11544, Coded Representation of Picture and Audio Information Progressive Bi-level Imamge Compression, WG9-S1R5.1, April 3, 1992.

[69] R. Hunter and A. H. Robinson, "International digital facsimile coding standards," *Proc. IEEE*, vol. 68, no. 7, pp. 854-867, July 1980.

[70] Y. Yasuda, Y. Yamazaki, T. Kamae, and K. Kobayashi, "Advances in FAX," *Proc. IEEE*, vol. 73, no.4, pp. 706-730, Apr. 1985.

[71] O. V. Sarca, "On constrained nonlinear image filters and their optimization," *Tampere University of Technology Publications*, No:235 - Thesis for the degree of Doctor of Technology, June 1998.

[72] L. A. Fletcher and R. Kasturi, "A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images," *IEEE Trans. on PAMI*, vol. 10, no. 6, pp. 910 November 1988.

[73] M. Kamel and A. Zhao, *Extraction of Binary Character/Graphics Images from Grayscale Document Images*, CVGIP: Graphical Models and Image Processing, vol. 55, no. 3, pp. 203-217, May 1993.

[74] Ö. N. Gerek, M. Akar, A. E. Çetin, 'Yazı bigisinin gri tonlu doküman görüntülerinden ayırımı için histograma dayalı yöntemlerin incelenmesi,' 2. SIU kurultayı, pp. 90, Gökova, Muğla, Nisan 1994. (In Turkish)

[75] V. K. Govindan and A. P. Shivaprasad, "Character Recognition - A Review," *Pattern Recognition,* vol. 23, no. 7, pp. 671-683, 1990.

[76] C. Y. Suen, R. De Mori, Ed., *Computer Analysis and Perception*, Vol. I Visual Signals, CRC Press, 1982.

[77] M. Berthod, "On-Line Analysis of Cursive Writing," Chapter 4 in [76].

[78] C. C. Tappert, C. Y. Suen and T. Wakahara, "The State of the Art in On-Line Handwriting Recognition," *IEEE Trans. on PAMI*, vol. 12, no. 8, August 1990.

[79] E. H. Adelson, E. Simoncelli, and Hingorani, "Orthogonal pyramid transforms for image coding," *Proc. SPIE Conf. VCIP*, pp. 50-58, Cambridge, MA, 1987.

[80] J. C. Feauveau, P. Mathieu, M. Barlaud, and M. Antinoni, "Recursive biorthogonal wavelet transform for image coding," *Proc. ICASSP'91*, pp.2649-2652, Toronto, Canada, 1991.

[81] M. Vetterli, J. Kovačević and Le Gall, " Perfect reconstruction filter banks for HDTV representation and coding," *Image Communication 2*, pp.349-364, 1990.

[82] Y. Meyer, *Ondelettes et Opeŕateurs*, Hermann, 1988.

[83] I. Daubechies, *Orthogonal bases of compactly supported wavelets, Commun. Pure and Applied Math*, Vol. XLI, pp. 909-996, 1988.

[84] S. G. Mallat, *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol II, no 7, pp. 674-693, July 1989.

[85] N. Ahmed, T. Natarajan, K. R. Rao, "Discrete Cosine Transform", *IEEE Trans. on Computers*, vol. 23, pp. 90-93, 1974.

[86] A. E. Çetin, Ö. N. Gerek, Ş. Ulukuş, "Block Wavelet Transforms for Image Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 3, No. 6, pp. 433-435, December 1993.

[87] H. Feichtinger and K. Gröchenig, "Gabor wavelets and the Heisenberg group: Gabor expansions and short time Fourier transform from the group theoretical point of view," in *Wavelets: A Tutorial In Theory and Applications*, C. K. Chui (ed.), Academic Press, Cambridge, MA, pp.359-397, 1992.

[88] G. Caire, L. Grossman an H. V. Poor, "Wavelet Transforms Associated with Finite Cyclic Groups," *IEEE Trans. Information Theory*, vol. 39, no. 4, pp. 1157-1166, July 1993.

[89] M. D. Swanson and A. H. Tewfik, "A Binary Wavelet Decomposition of Binary Images," Submitted to *IEEE Trans. Image Processing* (IP-941).

[90] Ö. N. Gerek, M. N. Gürcan, A. E. Çetin, "Binary Morphological Subband Decomposition For Image Coding," *IEEE Int. Symp. on Time-Frequency and Time Scale Analysis*, 1996.

[91] M. N. Gürcan, Ö. N. Gerek, A. E. Çetin, "A Morphological Subband Decomposition Structure using GF(N) Arithmetic," *Proc. of IEEE Int. Conf. on Image Proc.*, Switzerland, Sep. 1996.

[92] M. N. Gürcan, Ö. N. Gerek, A. E. Çetin, "Nonlinear Subband Decomposition Structures in GF-(N) Arithmetic," *Signal Processing*, vol. 64, no. 2, pp 209-213, Feb. 1998.

[93] T. Kohonen, "Self Organizing Maps," *Berlin: Springer-Verlag*, 1997.

[94] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, "SOM_PAK: The Self-Organizing Map Program Package," *Technical Report A31*, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, 1996.

[95] R. A. Wilkinson, "NIST Special Database 8: Machine Print Database," National Institute of Standards and Technology, Advanced Systems Division, Image Recognition Group, Oct. 1, 1992.

[96] D. S. Doermann and S. Yao, "Generating Synthetic Data for Text Analysis Systems," Documents Group Technical Report, UMD:95-0006, Jan. 1, 1995.

[97] A. K. Zafar and D. S. Doermann, "Document Understanding and Character Recognition WWW Server," http://documents.cfar.umd.edu/

[98] D. Salomon, *Data Compression*, Springer-Ferlag New York, 1998.

[99] Federal Bureau of Investigation, "WSQ gray-scale fingerprint image compression specification," document IAFIS-IC-0110v2, Feb. 1993.

[100] M. Gokmen, I. Ersoy, A. K. Jain, "Compression of fingerprint images using hybrid image model," Proc. IEEE ICIP'96, Vol III, pp. 398-298, Lausanne, 1996.

[101] B. G. Sherlock and D. M. Monro, "Psychovisually tuned wavelet fingerprint compression," Proc. IEEE ICIP'96, Vol II, pp. 585-588, Lausanne, 1996.