Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

A COMPARATIVE STUDY OF IMAGE COMPRESSION SCHEMES

by
Kui Wang

Image compression is an important and active area of signal processing. All popular image compression techniques consist of three stages: Image transformation, quantization (lossy compression only), and lossless coding (of quantized transform coefficients).

This thesis deals with a comparative study of several lossy image compression techniques. First, it reviews the well-known techniques of each stage. Starting with the first stage, the techniques of orthogonal block transformation and subband transform are described in detail. Then the quantization stage is described, followed by a brief review of the techniques for the third stage, lossless coding.

Then these different image compression techniques are simulated and their rate-distortion performance are compared with each other. The results show that two-band multiplierless PR-QMF bank based subband image codec outperforms other filter banks considered in this thesis. It is also shown that uniform quantizers with a "dead-zone" perform best. Also, the multiplierless PR-QMF bank outperforms the DCT based on uniform quantization, but underperforms the DCT based on uniform quantization with a "dead-zone".

A COMPARATIVE STUDY OF IMAGE COMPRESSION SCHEMES

by
Kui Wang

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering
Department of Electrical and Computer Engineering

January 1996

APPROVAL PAGE

A COMPARATIVE STUDY OF IMAGE COMPRESSION SCHEMES

Kui Wang

Dr. Ali N. Akansu, Thesis Advisor — Date
Associate Professor of Electrical and
Computer Engineering, NJIT

Dr. Nirwan Ansari, Committee Member — Date
Associate Professor of Electrical and
Computer Engineering, NJIT

Dr. Zoran Siveski, Committee Member — Date
Assistant Professor of Electrical and
Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Kui Wang

Degree: Master of Science in Electrical Engineering

Date: January 1996

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology,
  Newark, New Jersey, 1996

- Master of Science in Industrial Engineering,
  Eastern Illinois University,
  Charleston, Illinois, 1994

- Bachelor of Science in Electrical Engineering,
  Nanjing Aeronautical Institute of Technology,
  Nanjing, P.R.CHINA, 1990

Major: Electrical Engineering

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Dr. Ali Akansu for his valuable advice and contribution. I am also appreciative for his support and encouragement during this research period.

I am grateful to Dr. Nirwan Ansari and Dr. Zoran Siveski for their effort and time for reviewing this thesis.

Special thanks to my parents and family members for their moral support and love.

Finally, I would like to thank the members of the Center for Communication and Signal Processing Research at New Jersey Institute of Technology and all my friends for their help which made this research period a memorable one.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

Figure Page

CHAPTER 1

INTRODUCTION

Since the late 1980's, we have witnessed two trends: A rapid technology advancement in telecommunication and computer hardware, and an explosion of multimedia applications in every aspect of our daily life. The second trend is translated into the huge demand for better and faster image transmission and storage techniques. The basic concepts involved in an image compression scheme are presented below.

1.1 Image Compression

In image compression, original images are compressed to less number of representation bits in order to save transmission bandwidth or storage costs. The objective of image compression is to represent an image with less number of bits while maintaining an acceptable quality of the image depending on the application under consideration.

Many competitive techniques have been developed in image compression area. Technically speaking, those techniques can be broadly grouped into two types (Wong, Zaremba, Gooden, and Huang 195). The first group is lossless image compression, also referred to as reversible compression. A lossless scheme provides roughly a 2-to-1 compression factor for most of still frame images, but still allows an exact recovery of the original image from its compressed version. A lossy or irreversible image compression technique does not perfectly represent original images but strives only to

maintain a particular level of subjective image quality. It can provide compression factors from 10-to-1 to 100-to-1 and above (Baxes 28-29).

A lossless compression technique is used when an image information must be exactly preserved. Such images often include ones from medical and other scientific sources. A lossy compression technique is used when the quality of the reconstructed image is maintained at some acceptable level, but need not be identical to that of the original image. Images appropriate for lossy compression include those in video game programming, videoconferencing, and some printing applications.

Any image compression technique consists of three stages: Image transformation, quantization (lossy compression only), and lossless coding of quantized transform coefficients. This is displayed in figure 1.1. The relative importance of each stage varies from one compression technique to another, and all of these stages are not necessarily included in a particular scheme. The lossless compression techniques naturally do not include quantization. Next, each stage will be discussed in detail.

1.1.1 Image Transformation

Image transformation, consisting of conversion and inverse conversion, is the first stage of an image compression technique. It maps the image signal from its original form to the transform domain in which data is better represented for the quantization stage. An orthogonal image transformation reserves the signal and does not introduce any information loss. Its purpose is to decompose the original image into another domain. The transform domain representation of an ordinary image tends to be uncorrelated. Therefore, the transform coefficients can be treated independently in the quantization

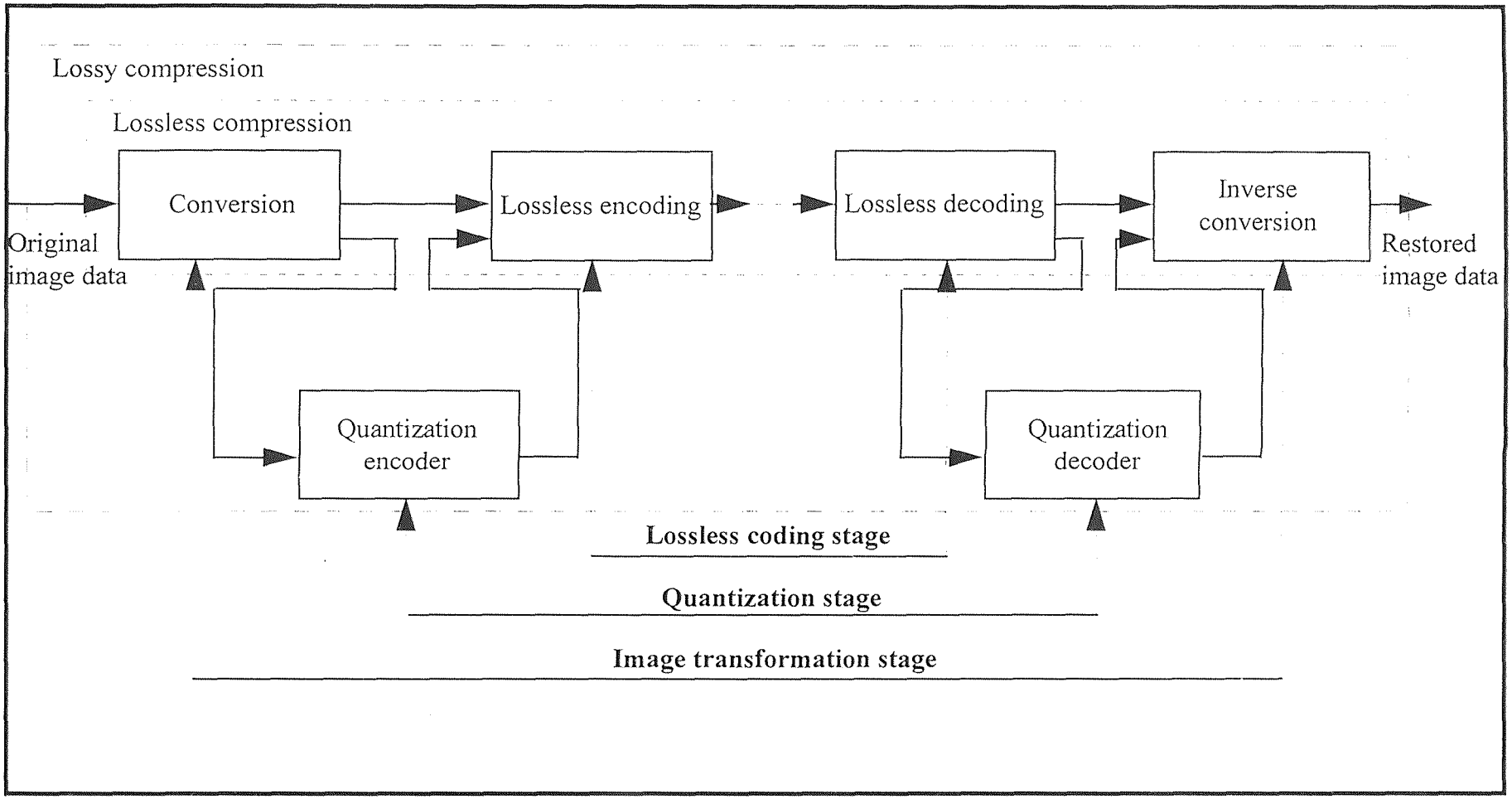


Figure 1.1 Three Stages of an Image Compression Technique.

stage. Several image transformation methods have been developed for mapping, such as linear prediction, orthogonal block transformation, and subband transform.

1.1.1.1 Linear Prediction The philosophy underlying the linear prediction is to reduce correlation between successive pixels. In general, linear prediction techniques use previous pixels to generate a predicted value of the current pixel. Then take the difference between the current pixel and its predicted value (Gersho and Gray 185-202). Since the pixel difference has less correlation than the pixel itself, quantization or lossless coding of the difference is more efficient than those of the pixel itself. One of the most widely used prediction methods is *differential pulse code modulation* (DPCM).

1.1.1.2 Orthogonal Block Transformation Orthogonal block transformation is currently the most widely used approach for image transformation. In this technique, an orthogonal transform is applied to a block of image pixels so that the transform domain coefficients tend to become less correlated with their energy concentrated in as few coefficients as possible. This energy compaction property leads to the prioritization of transform coefficients based on their relative significance. There are many orthogonal block transformation techniques based on different orthogonal transforms (Akansu and Haddad 9-90). Among the proposed transforms, the *discrete cosine transform* (DCT) is the most popular because of its superior decorrelation properties. The DCT is used in the baseline JPEG (Joint Photographic Experts Group) standard of still frame image compression (Pennebaker and Mitchell 65-79).

1.1.1.3 Subband Transform Subband transform, recently shown as a generalization of orthogonal block transformation, has also been proposed to provide an improved signal decorrelation and has been frequently used for image compression (Akansu and Haddad 4). In subband transform, a filter bank is utilized in order for an image to be decomposed into its nearly uncorrelated subbands. This allows independent treatment of subband signals in the quantization stage. There are a number of well-known subband transform techniques by applying different filter banks in the literature (Akansu and Haddad 241-351). Among them the one-dimensional (1-D) two-band *paraunitary perfect reconstruction quadrature mirror filer banks* (PR-QMF banks) are the most fundamental and popular. The subband filtering based on PR-QMF bank families have been widely used in the literature as an alternative to the DCT for image transformation purposes (Woods and O'Neil 1278-1288).

1.1.2 Quantization

Quantization is the process of many-to-one mapping of transform coefficients. Therefore, quantization is by definition lossy and pertaining to lossy compression only. In quantization, while maintaining a desired image quality, representation of transform coefficients is minimized. Thus, image compression is achieved.

The quantization is the most critical stage of a lossy image compression technique for low bit rates. It can be further categorized into *scalar quantization* (SQ) and *vector quantization* (VQ). The SQ is an operation on a single sample value. The quantizer replaces the single sample by the closest one of a finite set of reproduction values. In the VQ, a set of k single samples is mapped into one of a finite set of reproduction vectors.

In the JPEG baseline standard scalar quantization is used. In such an operation, a DCT coefficient is mapped to an integer by rounding off the ratio of the coefficient over a quantization interval (Pennebaker and Mitchell 65-79).

1.1.3 Lossless Coding

The third stage of an image compression scheme is lossless coding of the quantized transform coefficients in order to achieve some compression. Lossless coding may be as simple as using *fixed-length* digital codewords to represent the outputs from the previous stage, or it may use *variable-length* codes such as *Huffman coding*, *arithmetic coding*, and *run length coding*, in order to achieve higher compression ratio (Gersho and Gray 225-260). For example, Huffman coding and arithmetic coding are used separately in JPEG (Pennebaker and Mitchell 65-79).

1.2 Summary of the Thesis Work

In this thesis, we focus on the first two stages of lossy image compression techniques, which are image transformation and quantization. Comparative studies of the compression techniques are presented based on the methods of image transformation and quantization.

Simulation programs were developed, consisting of image transformation and quantization stages for different codec scenarios. Lossless coding stage is not included in this thesis. But it seems that it is included in the simulation programs because one of the performance measures which are used for comparison purpose, the bit rate of the compressed image, is approximated by the entropies of the quantizer outputs.

During the comparative performance studies, subband transform methods are the main focus for image transformation. The rate-distortion performance of these techniques are compared. In addition, the coding performance of scalar and vector quantization methods are compared under the same test conditions. For the completeness of the study the industry standard, 2-D 8 × 8 DCT, based image codec is also included in this thesis.

This thesis is organized as follows. Chapter 2 describes the first stage, image transformation techniques. Chapter 3 presents the second stage techniques considered. Scalar and vector quantization schemes are described. Chapter 4 briefly describes the lossless coding techniques. Chapter 5 deals with the rate-distortion performance comparison of the different lossy image compression techniques studied in this thesis. The simulation programs and the results are displayed and discussed in detail.

CHAPTER 2

IMAGE TRANSFORMATION

Image transformation can be viewed as the "preprocessor" of an image compression scheme. In image transformation, image samples are mapped from their original form to the transform domain in which they are better represented for the quantization stage. Then, the resulting quantized data are mapped back to the original domain by the corresponding inverse transformation. The block diagram of an image transformation is displayed in figure 2.1. The purpose of image transformation is to analyze the original image in the transform domain. An image signal tends to be less correlated in transform domain. Therefore, each transform coefficient can be treated independently in the quantization step. It improves the compression efficiency. There are many image transformation techniques such as linear prediction, orthogonal block transformation, and subband transform. In this chapter, we will briefly present orthogonal block transformation, and then focus on subband transform.

2.1 Orthogonal Block Transformation

2.1.1 Definitions

In 1-D orthogonal block transformation, a square unitary matrix A which is also called a 1-D orthogonal transform is applied to a 1-D data sequence $\{x(n);\ n = 0, 1,..., N\text{-}1\}$ in order to obtain a set of almost uncorrelated transform coefficients $\{y(n);\ n = 0, 1,..., N\text{-}1\}$.

These coefficients in turn can be transformed back to the original signal domain as $x(n)$ by applying the corresponding inverse transform. This is summarized as

Forward transformation (FT) $\quad \mathbf{y} = \mathbf{A}\mathbf{x}$

Inverse transformation (IT) $\quad \mathbf{x} = \mathbf{A}^{*T}\mathbf{y}$

where $\quad \mathbf{y} = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}, \quad \mathbf{A}^{-1} = \mathbf{A}^{*T}$

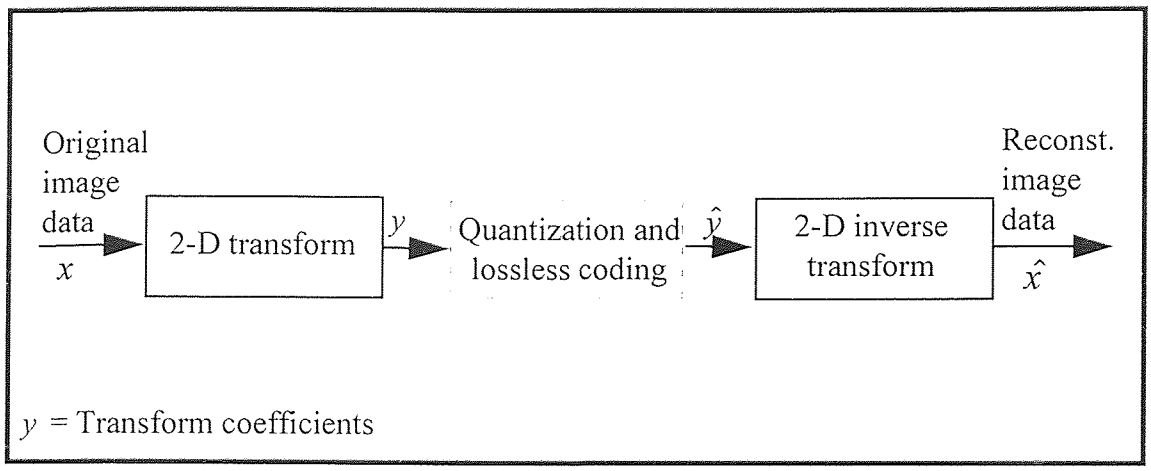


Figure 2.1 A Block Diagram of Image Transformation Stage.

The above 1-D orthogonal block transformation can be extended to 2-D image arrays. For the 2-D separable case, a 1-D orthogonal transform is applied first to each column of the image array, and then to each row of the resulting intermediate image array. For an $N \times N$ input image array $\mathbf{X}$, the forward and inverse transform operations are summarized as (Jain 134-140)

$$\begin{aligned} &\text{FT:} && \mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{A}^{T} \\ &\text{IT:} && \mathbf{X} = \mathbf{A}^{*T}\mathbf{Y}\mathbf{A}^{*} \end{aligned} \tag{2.1}$$

where

$$\mathbf{X} = \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,N-1} \\ x_{1,0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ x_{N-1,0} & x_{N-1,1} & \cdots & x_{N-1,N-1} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_{0,0} & y_{0,1} & \cdots & y_{0,N-1} \\ y_{1,0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ y_{N-1,0} & y_{N-1,1} & \cdots & y_{N-1,N-1} \end{bmatrix}, \quad \mathbf{A}^{-1} = \mathbf{A}^{*T}$$

In practice, a given image array is usually first divided into its nonoverlapping square blocks of pixels. Then each of these blocks is transformed independently to the transform domain.

Among many proposed orthogonal transforms, the DCT has superior decorrelation properties and it has been used in the current image compression standards such as JPEG. In next section, we will explain DCT in detail.

2.1.2 The Discrete Cosine Transform

Based on equation (2.1), the 2-D DCT of a $K \times K$ image block is given by (Jain 150-151)

$$\text{FT:} \quad y(m,n) = c(m)c(n)\sum_{p=0}^{K-1}\sum_{q=0}^{K-1} x(p,q)\cos((\pi(2p+1)m)/2K)\cos((\pi(2q+1)n)/2K) \quad (2.2)$$

$$\text{IT:} \quad x(p,q) = \sum_{m=0}^{K-1}\sum_{n=0}^{K-1} c(m)c(n)y(m,n)\cos((\pi(2p+1)m)/2K)\cos((\pi(2q+1)n)/2K) \quad (2.3)$$

where

$$c(m) = \sqrt{\frac{1}{K}} \qquad \text{for m = 0}$$

$$c(m) = \sqrt{\frac{2}{K}} \qquad \text{for m = 1, 2,..., } K\text{-1}$$

$$c(n) = \sqrt{\frac{1}{K}} \qquad \text{for n = 0}$$

$$c(n) = \sqrt{\frac{2}{K}} \qquad \text{for n = 1, 2,..., } K-1$$

$x(p,q)$ = 2 - D input image pixel
$y(m,n)$ = 2 - D output DCT coefficient

In this transformation, the given $N \times N$ image is first divided into $\left(\frac{N}{K}\right) \times \left(\frac{N}{K}\right)$ blocks, each of size $K \times K$. Then each image block is transformed independently by using equation (2.2). Finally the reconstructed image blocks can be obtained by inverse transformation, equation (2.3).

2.2 Subband Transform

2.2.1 Introduction

Subband transform, also providing signal decorrelation, is another widely used technique for image transformation. In subband coding, a given $N \times N$ image is decomposed into its several subbands by the analysis filters in the filter bank. Then each band is decimated following the Nyquist Criterion. Finally in the inverse transformation process, the resulting quantized subbands are interpolated and filtered by the synthesis filters of the filter bank, and then summed together to obtain the reconstructed image.

This approach, in general, demands the design of a sophisticated 2-D m-band filter bank, which consists of the analysis and synthesis filters, in order to achieve perfect reconstruction in the absence of quantization, lossless coding, and transmission errors.

The 2-D separable case is considered here due to its simplicity. In this case, a 1-D filter bank is extended to 2-D input image arrays where first the columns of an image array are filtered, and decimated by the 1-D analysis filters, and then the same operations are applied to the rows of the resulting image array. This is an alternative method to the design of a 2-D filter bank.

Two-band filter banks are considered in this study. In subband transform with two-band filter banks, first a low-pass filter and a high-pass filter are applied to the columns of an image array. Two overlapping subbands are obtained. They are then decimated to fit their Nyquist rates. Then the similar splitting and decimating are performed on the rows of each of the two resulting bands. The above process is repeated until the desired split is achieved. Figure 2.2 shows the block diagram of this splitting process, also known as subband tree structure.

Now we have simplified the design of sophisticated 2-D m-band filter banks to the design of simple 1-D two-band filter banks. For more discussions on filter bank design in general, please see Reference (Akansu and Haddad).

2.2.2 1-D Two-Band Filter Bank Design

The fundamental goal of two-band filter bank design is to define analysis and synthesis filters which can provide perfect reconstruction (PR) of the original input signal in the absence of quantization, lossless coding, and transmission errors. This perfect reconstruction requirement is illustrated as the following.

Perfect Reconstruction Requirements:

Figure 2.3 displays a two-band filter bank. Here both the *z-domain* and the *time-domain* expressions are used. The input signal $x(n)$ is divided into its two subbands by the analysis filters $H_L(z)$ and $H_H(z)$. Then, according to the Nyquist theorem, $x_{L1}(n)$ and $x_{H1}(n)$ are each down sampled by 2 to provide $x_{L2}(n)$ and $x_{H2}(n)$. In the absence of quantization, lossless coding, and transmission errors, we assume that $x_{L2}(n)$ and $x_{H2}(n)$ are inputs to the synthesis stage. Both of them are up sampled by 2 and processed by the

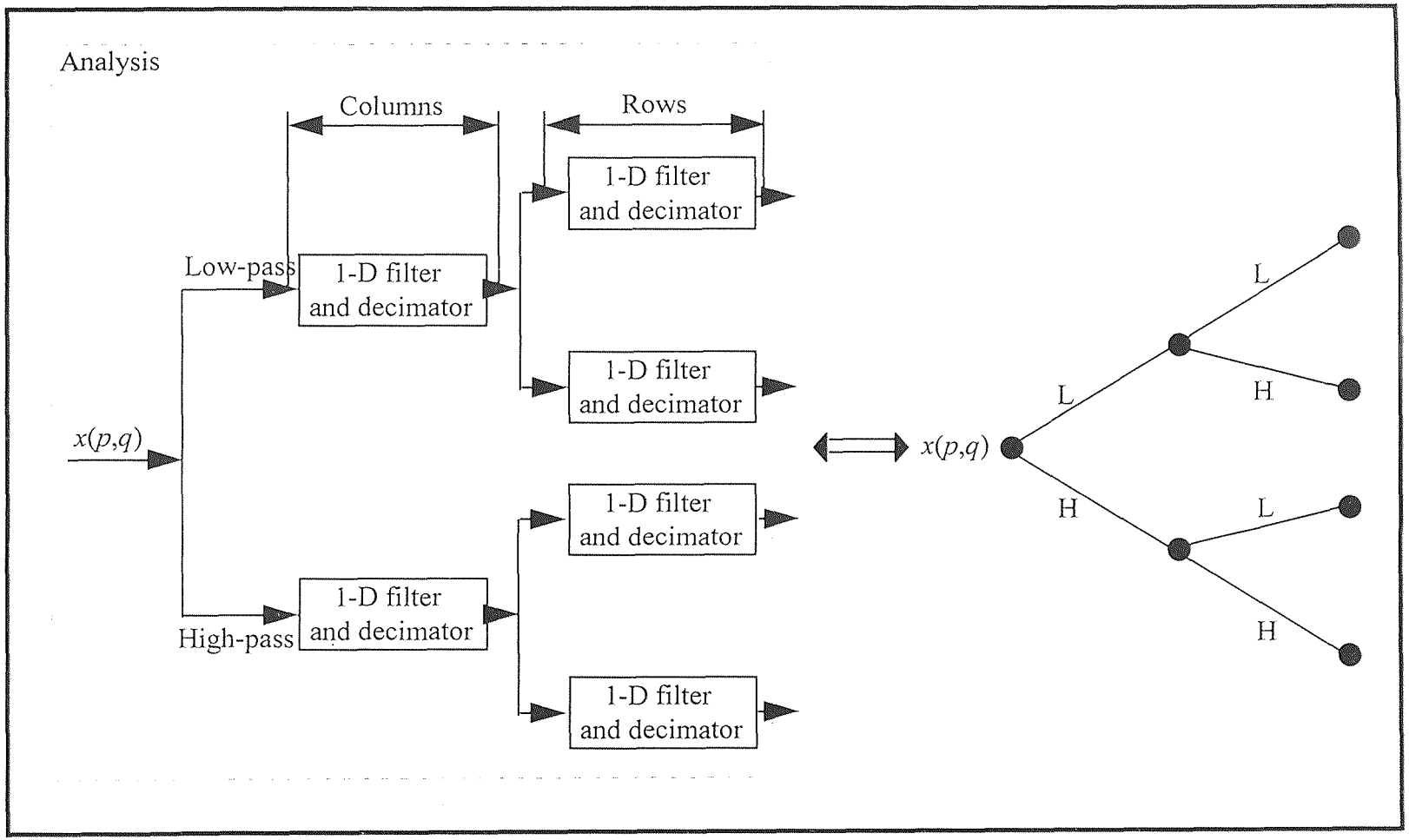


Figure 2.2 A 4-Band Subband Tree.

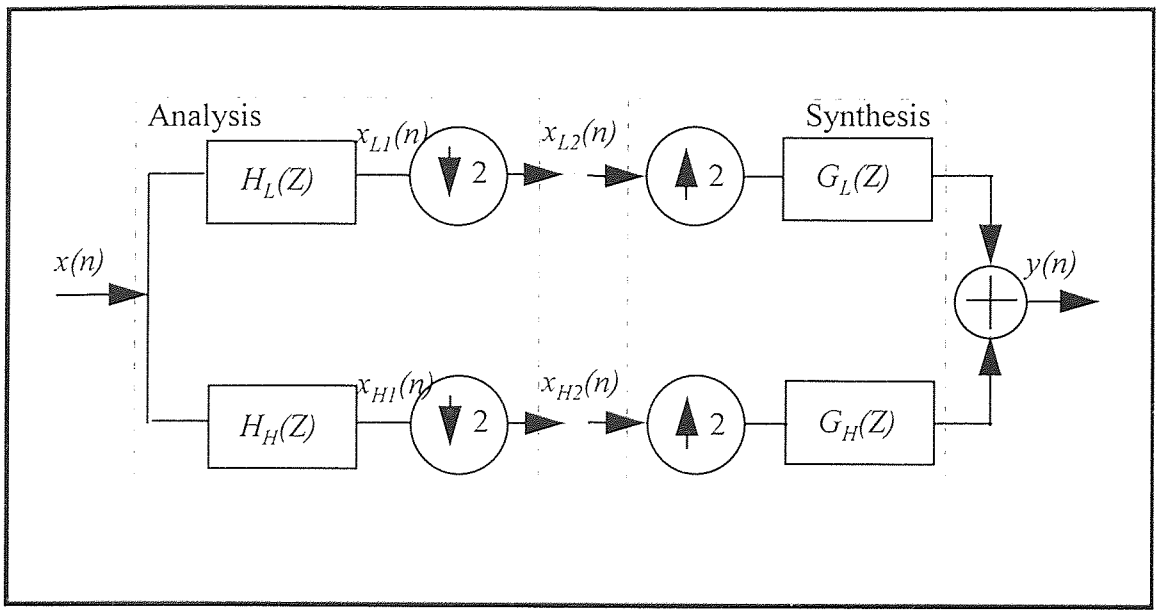


Figure 2.3 A Two-Band Filter Bank.

synthesis filters $G_L(z)$ and $G_H(z)$, and finally summed up to yield the reconstructed signal $y(n)$, which can be expressed in z-domain as:

$$Y(z) = \frac{1}{2}[H_L(z)G_L(z) + H_H(z)G_H(z)]X(z) + \frac{1}{2}[H_L(-z)G_L(z) + H_H(-z)G_H(z)]X(-z) \quad (2.4)$$

In order to achieve perfect reconstruction, we select:

$$G_L(z) = -H_H(-z)$$

$$G_H(z) = H_L(-z)$$

$$H_H(z) = z^{-(N-1)}H_L(-z^{-1})$$

where $H_L(z)$, $H_H(z)$, $G_L(z)$, and $G_H(z)$ are N-tap *finite impulse response* (FIR) with N even. Now equation (2.4) is simplified to (Akansu and Haddad 101-239)

$$Y(z) = \frac{1}{2}z^{-(N-1)}[H_L(z)H_L(z^{-1}) + H_L(-z)H_L(-z^{-1})]X(z)$$

The perfect reconstruction requirement reduces to finding $H_L(z)$ such that

$$H_L(z)H_L(z^{-1}) + H_L(-z)H_L(-z^{-1}) = \text{constant}$$

The above perfect reconstruction requirements are summarized in the time-domain as (Akansu and Haddad 101-239):

$$\begin{aligned}
&g_L(n) = h_L(N-1-n)\\
&g_H(n) = (-1)^n h_L(n)\\
&h_H(n) = (-1)^{n+1} h_L(N-1-n)\\
&\text{and with } \rho(n) = h_L(n) \otimes h_L(-n) \ (\otimes \text{ denotes convolution})\\
&\qquad\qquad \rho(2n) = \delta(n)\\
&h_L(n), h_H(n), g_L(n), \text{ and } g_H(n) \text{ are all N - tap FIR with N even}
\end{aligned} \quad (2.5)$$

The two-band filter bank who satisfies equation (2.5) is also called 1-D two-band paraunitary perfect reconstruction quadrature mirror filter bank (PR-QMF bank).

In addition to the PR requirements, some other design criteria are proposed for the

design of two-band filter banks, such as uncorrelated subband signals, zero mean high-pass filter, energy compaction, computational efficiency, etc. (Akansu and Haddad 260-272). Based on these different design criteria, and the PR conditions of equation (2.5), a set of PR-QMF banks can be generated.

2.2.3 PR-QMF Bank Families

2.2.3.1 Uncorrelated PR-QMF Bank The uncorrelated PR-QMF bank is obtained from both the PR conditions of equation (2.5) and the following requirements (Akansu and Haddad 264):

$$R_{LH}(0) = E\{x_{L2}(n)x_{H2}(n)\} = \sum_m \sum_l h_L(l)h_H(m)R_{xx}(m-l) = 0 \qquad \text{for all } n \tag{2.6}$$

$$\sum_{n=0}^{N-1} h_H(n) = 0 \tag{2.7}$$

$$\text{Maximize} \qquad \sigma_{L1}^2 = \mathbf{h}_L^T \mathbf{R}_{xx} \mathbf{h}_L \tag{2.8}$$

Equation (2.6) requires that the cross-correlation of the two subband signals $x_{L2}(n)$ and $x_{H2}(n)$, which are at the output of the analysis stage as shown in figure 2.3, is forced to be zero. This can approximately achieve uncorrelated subbands, which is a desired property for any subband transform technique. $R_{xx}(m-l) = E\{x(m)x(l)\}$ is the autocorrelation function of the input signal. However, this expression is hard to evaluate, in practice $R_{xx}(n)$ can be approximated by

$$R_{xx}(n) = \rho^{|n|} \qquad n = 0, \pm 1, \pm 2, \ldots \tag{2.9}$$

provided that input signal is first-order autoregressive (AR(1)), and with mean = 0, ρ = 0.95. This is a crude approximation to the real-world images.

Equation (2.7) requires that the high-pass filter $h_H(n)$ has a zero mean. In equation (2.8), $\mathbf{R}_{xx}$ is the correlation or covariance matrix of the assumed AR(1) input, and σ_{L1}^2 is the variance or power of subband $x_{L1}(n)$. $\mathbf{h}_L$ is the vector form representation of $h_L(n)$. The purpose of these two equations is to try to achieve maximum energy compaction among the subbands. This will provide desired conditions for the quantization stage operations in order to achieve a reasonable degree of compression.

2.2.3.2 Multiplierless PR-QMF Bank Multiplierless PR-QMF bank is of great practical interest because of its computational efficiency. In this filter bank, $h_L(n)$ has only the allowed coefficient values as

$$h_L(n) = \pm 2^{\pm k_n} \quad \text{or} \quad h_L(n) = \pm 2^{\pm k_n} + 1 \qquad n = 0,\ 1,\ldots,\ N-1 \qquad (2.10)$$

where k_n is an integer. This equation states that any low-pass filter coefficient is expressed as a binary shift or shift and add operations (Akansu 723-725). The multiplierless PR-QMF bank is obtained from equations (2.5), (2.8), and (2.10).

2.2.3.3 Binomial PR-QMF Bank The binomial PR-QMF bank is obtained from both the PR conditions of equation (2.5) and the following "binomial" requirement (Akansu and Haddad 241-246):

$$h_L(n) = \sum_{r=0}^{\frac{N-1}{2}} \theta_r x_r(n) \qquad (2.11)$$

where $\theta_0, \theta_1, \ldots, \theta_{\frac{N-1}{2}}$ are $\frac{N+1}{2}$ unknowns,

$$x_0(n) = \begin{cases} \binom{N}{n} = \dfrac{N!}{(N-n)!n!} & 0 \le n \le N \\ 0 & \text{otherwise} \end{cases},$$

$$x_r(n) = \nabla^r \binom{N-r}{n}$$

$$= \binom{N}{n} \sum_{v=0}^{r} (-2)^v \binom{r}{v} \frac{n^{(v)}}{N^{(v)}} \qquad 0 \le r, n \le N,$$

$$n^{(v)} = \begin{cases} n(n-1)\cdots(n-v+1) & v \ge 1 \\ 1 & v = 1 \end{cases}.$$

Notice that for simplicity of the expression, the tap of the filter in equation (2.11) is designated as $N + 1$, with N odd.

2.2.3.4 Smith-Barnwell Filter Bank The Smith-Barnwell filter bank is essentially the same as the two-band PR-QMF bank, except for a possible difference at the phase responses of the filters. It is summarized in the time-domain as (Smith and Barnwell 436)

$g_L(n) = h_L(N-1-n)$
$g_H(n) = (-1)^{n+1} h_L(n)$
$h_H(n) = (-1)^{n+2} h_L(N-1-n)$
and with $\rho(n) = h_L(n) \otimes h_L(-n)$ ($\otimes$ denotes convolution)
$\rho(2n) = \delta(n)$
$h_L(n), h_H(n), g_L(n)$, and $g_H(n)$ are all N - tap FIR with N even

2.2.4 Subband Tree Structure

We have designed the desired 1-D two-band PR-QMF banks in the previous sections. The separable m-band decomposition of the input image can be generated by extending the two-band structure in a hierarchical subband tree, as shown in figure 2.4.

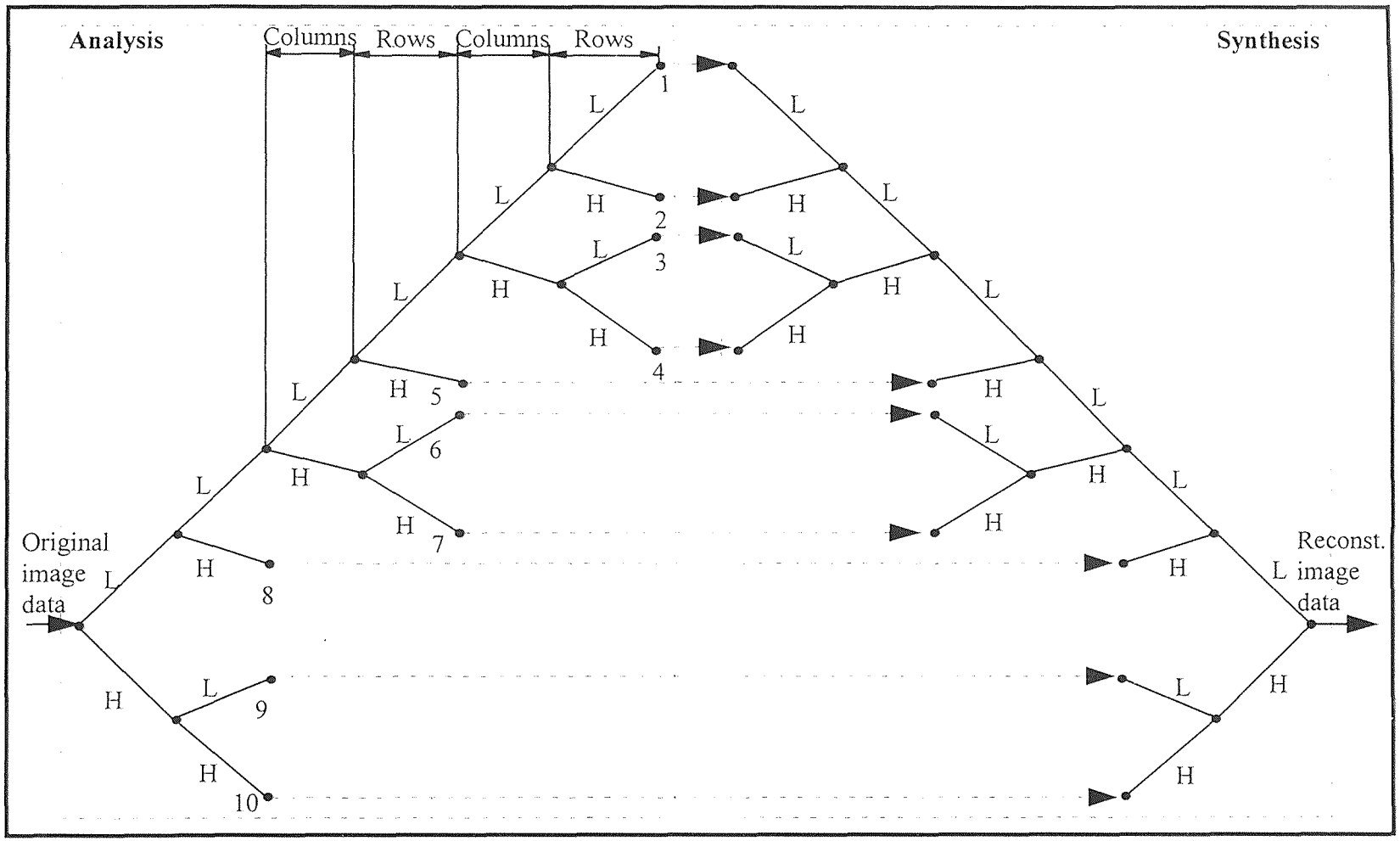


Figure 2.4 One Technique of Image Transformation Stage, 2-D Subband Filtering Employing a 10-Band Subband Tree Structure.

CHAPTER 3

QUANTIZATION

The subsequent step to image transformation in lossy image compression techniques is quantization. It achieves image data compression by represent transform coefficients with no greater precision than is necessary to achieve desired image quality.

Quantization, represented by Q, is a mapping of k-dimensional Euclidean space R^k into a finite subset $\mathbf{Y}$ of R^k. Thus,

$$Q : R^k \rightarrow \mathbf{Y}$$

where $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ and $\mathbf{y}_i$ is in R^k for each i. When $k = 1$, it is referred to as scalar quantization, and $k > 1$, as vector quantization.

In practice quantization, whether it is scalar quantization or vector quantization, does not exist as a single entity. It is actually the composite of two separate functions: the encoder and the decoder. The encoder maps the input k-dimensional Euclidean space R^k into the *partition regions* and generates the index i (i = 1, 2,..., N) for each region. The decoder replaces i with the corresponding $\mathbf{y}_i$ from a finite set $\mathbf{Y}$. Figure 3.1 shows a block diagram of quantization stage.

3.1 Scalar Quantization

3.1.1 Definition

Scalar quantization is defined as the process of transforming a continuous variable x into a discrete variable y, which takes values from a finite set of possible numbers. Here we

assume that the quantization process is memoryless and instantaneous, which means that the quantizer operates on one input symbol at a time, and the output value depends only on that input. Note that the quantizer mapping is irreversible; that is, for a given quantizer output, the input value cannot be determined uniquely. Hence, a quantizer introduces distortion, which any reasonable design method must attempt to minimize.

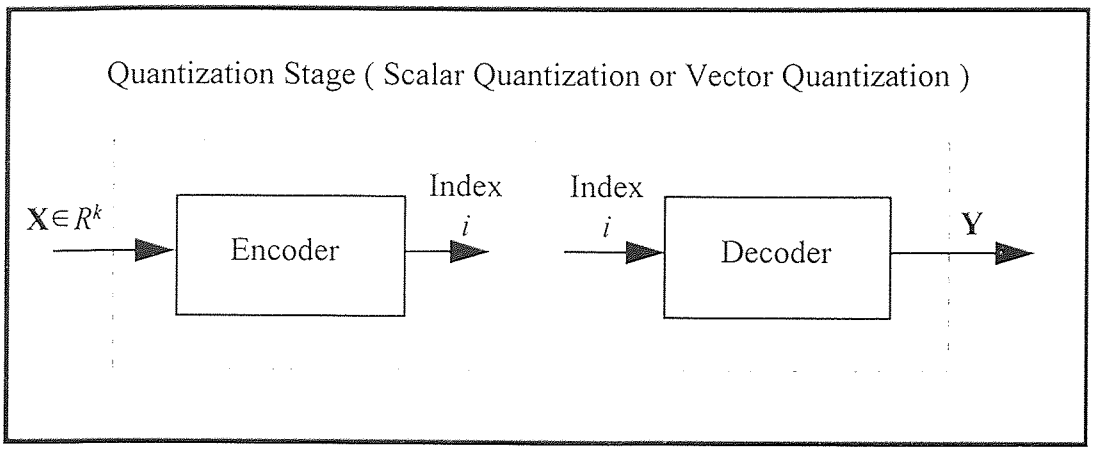


Figure 3.1 A Block Diagram of Quantization Stage.

3.1.2 Scalar Quantizer Design

The principal goal of scalar quantizer design is to select the *reproduction levels* and the *transition levels* or the partition regions so as to provide the minimum possible *average distortion* for a fixed number of levels N. Here the average distortion is focused mainly on *average squared error distortion* between a given input random variable x and the reproduced discrete variable y. In general, it can be written as

$$D = E\{(x-y)^2\} \tag{3.1}$$

It is also commonly called *mean square error*.

There are several optimal quantizer design methods available that minimize the average distortion (Gersho and Gray 159-184). Among them the Lloyd II algorithm is particular important.

3.1.2.1 The Lloyd II Algorithm Let x be a real scalar input random variable with a continuous *probability density function* (pdf) $p_x(x)$. Define $\{t_k; k = 1,..., N+1\}$ as a set of increasing transition levels with t_1 and t_{N+1} as the minimum and maximum values, respectively of x. If x lies in interval $[t_k, t_{k+1})$, then it is mapped to r_k, the kth reproduction level, from a finite set $\{r_1, r_2, ..., r_N\}$ of numbers.

In the Lloyd II algorithm it is desired to find the transition levels t_k and the reproduction levels r_k for an N-level quantizer such that the mean square error

$$\varepsilon = E[(x-y)^2] = \int_{t_1}^{t_{N+1}} (x-y)^2 p_x(x)\mathrm{d}x \tag{3.2}$$

is minimized. Rewriting this as

$$\varepsilon = \sum_{i=1}^{N} \int_{t_i}^{t_{i+1}} (x-r_i)^2 p_x(x)\mathrm{d}x \tag{3.3}$$

The necessary conditions for minimization of ε are obtained by differentiating it with respect to t_k and r_k and equating the results to zero. This gives

$$\begin{aligned}
\frac{\partial \varepsilon}{\partial t_k} &= \frac{\partial}{\partial t_k}\left(\int_{t_1}^{t_2}(x-r_1)^2 p_x(x)\mathrm{d}x + \int_{t_2}^{t_3}(x-r_2)^2 p_x(x)\mathrm{d}x + \cdots + \int_{t_{k-1}}^{t_k}(x-r_{k-1})^2 p_x(x)\mathrm{d}x + \int_{t_k}^{t_{k+1}}(x-r_k)^2 p_x(x)\mathrm{d}x\right. \\
&\quad \left. + \cdots + \int_{t_N}^{t_{N+1}}(x-r_N)^2 p_x(x)\mathrm{d}x\right) \\
&= (t_k - r_{k-1})^2 p_x(t_k) - (t_k - r_k)^2 p_x(t_k) = 0 \qquad 2 \le k \le N \\
\frac{\partial \varepsilon}{\partial r_k} &= -2\int_{t_k}^{t_{k+1}} (x-r_k) p_x(x)\mathrm{d}x = 0 \qquad 1 \le k \le N
\end{aligned}$$

Using the fact that $t_{k-1} \leq t_k$, simplification of the preceding equations gives

$$t_k = \frac{r_k + r_{k-1}}{2} \qquad 2 \leq k \leq N \tag{3.4}$$

$$r_k = \frac{\int_{t_k}^{t_{k+1}} x p_x(x)\mathrm{d}x}{\int_{t_k}^{t_{k+1}} p_x(x)\mathrm{d}x} \qquad 1 \leq k \leq N \tag{3.5}$$

The optimum values t_k and r_k can be calculated iteratively from equation (3.4) and (3.5) given the boundary values t_1 and t_{N+1}:

1. Choose a value for r_1. Set $k = 2$.
2. Solve equation (3.5) for t_k, and then solve equation (3.4) for r_k.
3. If $k = N$, continue. Otherwise set $k + 1 \rightarrow k$ and go to step 2.
4. Check if r_N is close enough to the right hand term of equation (3.5). If not, perturb r_1 in an appropriate direction, set $k = 2$, and go to step 2.
5. Stop. $\{t_k; k = 1, 2,..., N+1\}$ and $\{r_k; k = 1, 2,..., N\}$ are the final optimum solutions.

Figure 3.2 shows a block diagram of a scalar quantizer designed based on the Lloyd II algorithm. This algorithm has been widely used for designing scalar quantizers with average squared error distortion criteria and known pdfs that are sufficiently well behaved to ensure the existence of the derivatives in question. This technique was also independently developed by Max (Max 7-12). So the resulting quantizer is commonly known as the Lloyd-Max quantizer.

3.1.2.2 The Lloyd-Max Quantizer for Laplacian Density For Laplacian pdf, which is defined as follows:

$$p_x(x) = \frac{1}{\sigma^2} \exp(-\frac{2}{\sigma^2}|x - \mu|) \tag{3.6}$$

where μ and σ^2 denote the mean and variance, respectively, of x. Following equation (3.4), (3.5), and the iterative algorithm introduced above, optimal transition levels and reproduction levels can be obtained. Table 3.1 lists the design values for several Lloyd-Max quantizers for the Laplacian density.

Table 3.1 Lloyd-Max Quantizers for the Laplacian Density with Zero Mean and Unity Variance.

Levels	3		5		7	
k	t_k	r_k	t_k	r_k	t_k	r_k
1	- ∞	-1.1414	- ∞	-2.2538	- ∞	-2.8533
2	-0.7071	0.0	-1.5467	-0.8395	-2.1462	-1.4391
3	0.7071	1.1414	-0.4198	0.0	-1.0194	-0.5996
4	∞		0.4198	0.8395	-0.2998	0.0
5			1.5467	2.2538	0.2998	0.5996
6			∞		1.0194	1.4391
7					2.1462	2.8533
8					∞	

Source: Jain, Anil K. *Fundamentals of Digital Image Processing*. (Englewood Cliffs: Prentice Hall, 1989) 108.

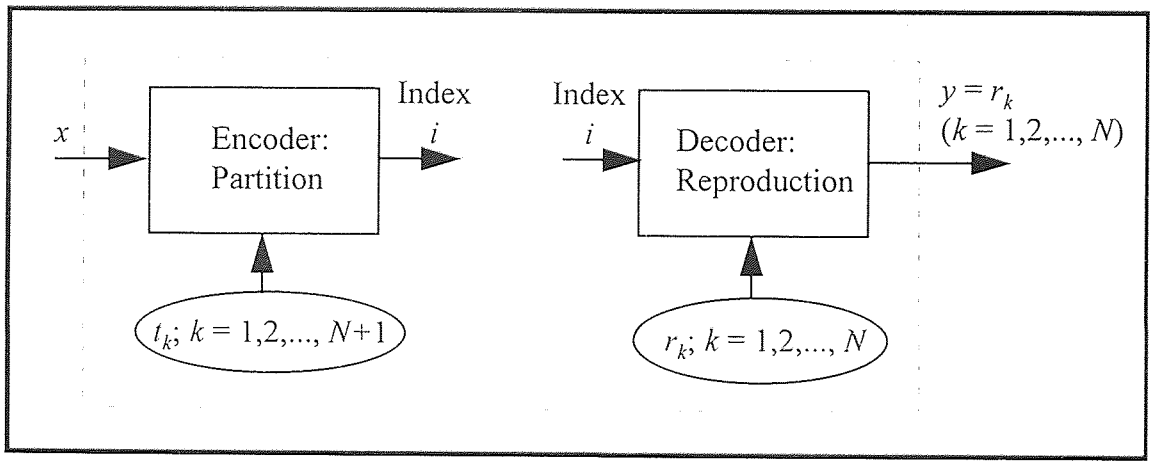


Figure 3.2 A Block Diagram of a Scalar Quantizer Designed Based on the Lloyd II Algorithm.

3.1.2.3 The Lloyd-Max Quantizer for Uniform Distribution In the case of a uniform input, the Lloyd-Max quantizer equations become linear, giving equally spaced transition levels and reproduction levels. Also the reproduction levels are the midpoints of the transition levels (Jayant and Noll 124-129):

$$r_k - r_{k-1} = \text{constant} \Rightarrow q, \quad t_k - t_{k-1} = q, \quad r_k = t_k + \frac{q}{2} \tag{3.7}$$

The Lloyd-Max quantizer for uniform distribution is also called the *linear* or *uniform quantizer*.

3.2 Vector Quantization

3.2.1 Definition

Vector quantization is an extension of scalar quantization to a higher dimensional space ($k > 1$). Formally, vector quantization of dimension k and size N is a mapping, Q, that assigns to each input vector, $\mathbf{x} = (x_1, x_2, ..., x_k)$, a reproduction vector, $\mathbf{y} = Q(\mathbf{x})$, drawn from a finite set $\mathbf{C}$ containing N reproduction vectors, $\mathbf{C} = \{\mathbf{y}_i;\ i = 1, 2, ..., N\}$. These reproduction vectors are also called *codewords*. The set $\mathbf{C}$ is called *codebook*. Vector quantizers are completely described by the codebook $\mathbf{C}$ together with the partition of the input vector space into the sets $\mathbf{S}_i = \{\mathbf{x}: Q(\mathbf{x}) = \mathbf{y}_i\}$ of input vectors mapping into the ith codeword.

3.2.2 Vector Quantizer Design

Similar to the design of scalar quantizers, the principal goal in design of vector quantizers is to find a codebook and a partition or encoding rule that will maximize an overall

measure of performance considering the entire sequence of vectors to be encoded over the lifetime of the quantizer.

Statistical average of a suitable distortion or *worst-case* value of a distortion is commonly used as an overall performance measure of a vector quantizer (Gersho and Gray 299). We focus here on statistical criteria. The performance of a vector quantizer is good if the statistical average distortion is small.

The statistical average distortion of a vector quantizer can be expressed as:

$$D = E\{d(\mathbf{x}, \mathbf{y})\}$$

$d(\mathbf{x}, \mathbf{y})$ is a nonnegative distortion measure caused by reproducing an input vector $\mathbf{x}$ by a reproduction vector $\mathbf{y}$.

Many distortion measures, $d(\mathbf{x}, \mathbf{y})$, have been proposed in Gersho and Gray's book (277-279). We focus here on the *squared error distortion* because it is the most mathematically convenient and widely used,

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2 = \sum_{i=1}^{k} (x_i - y_i)^2 \tag{3.8}$$

If the input sequence of vectors is stationary and ergodic, the time-averaged distortion,

$$\frac{1}{n} \sum_{i=1}^{n} d(\mathbf{x}_i, \mathbf{y}_i),$$

converges with probability one to the statistical average distortion D as $n \to \infty$. So D can be expressed as:

$$D = \frac{1}{n} \sum_{i=1}^{n} d(\mathbf{x}_i, \mathbf{y}_i) \tag{3.9}$$

This is also called the average squared error distortion. Until now we have defined the overall measure of performance for a vector quantizer. We proceed to design an encoding rule and a codebook that will minimize the average squared error.

3.2.2.1 Encoding Rule Design: Nearest Neighbor Method *Nearest neighbor* (NN) encoding rule means that each input vector $\mathbf{x}$ is mapped into $\mathbf{y}_i$, which minimizing the distortion $d(\mathbf{x}, \mathbf{y}_i)$. NN encoding rule is commonly used by almost every vector quantizers (Gersho 157-166). This method can be summarized by the following simple algorithm:

1. $\mathbf{x}$ is a k-dimensional input vector, and $\mathbf{y}_1 \in \mathbf{C} = \{\mathbf{y}_i; i = 1, 2, ..., N\}$. Calculate $d(\mathbf{x}, \mathbf{y}_1)$ from equation (3.8). Set $d_0 = d(\mathbf{x}, \mathbf{y}_1)$ and $i = 2$.
2. Compute $D_i = d(\mathbf{x}, \mathbf{y}_i)$.
3. If $D_i < d_0$, set $D_i \rightarrow d_0$, and $i \rightarrow k_i$.
4. If $i < N$, set $i + 1 \rightarrow i$, and go to step 2.
5. Stop.

The resulting value of k_i gives the output of the encoder, which is the index or address of the reproduction vector $\mathbf{y}_i$. The final value d_0 is the distortion between $\mathbf{x}$ and $\mathbf{y}_i$, which satisfies

$$d_0 = d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j) \qquad j \neq i, \quad 1 \leq j \leq N$$

From above we can see that in an NN encoder the partition of the input vector space is completely determined by the codebook and the distortion measure. Also, for a given codebook such an encoder is in fact optimal in the sense of minimizing average squared error.

3.2.2.2 Codebook Design Codebook design is the most important and most crucial step for the effective design of a vector quantizer. There are no known closed form solutions to the design of an optimal codebook. Generally, two groups of codebook design methods have been invented. The first group based its design on the known joint pdf of the input vector. However, in reality, it is not common to obtain the pdfs for input vectors. Thus the second group based its design on empirical observations of the input vector to generate a codebook. These observations are also called the *training sequence*. This second approach has become a standard technique for vector quantizer design in recent years (Linde, Buzo, and Gray 86).

In this method, the training sequence, $\{\mathbf{t}_j; j = 1, 2, ..., n\}$, is used to design a codebook that minimizes the statistical average distortion for the training sequence iteratively. The iteration begins with a vector quantizer consisting of its initial codebook. Then the steps of finding the corresponding NN partition for the codebook and then optimizing the codebook for the partition are proceeded until the optimality or smallest average distortion is achieved. Thus the final optimal codebook is obtained. The most important algorithm of this approach, the Linde-Buzo-Gray (LBG) algorithm, also called generalized Lloyd I algorithm, will be given in the following for iterative codebook improvement.

If we find a vector quantizer that is optimal for the training sequence and the training sequence is sufficiently long, based on the ergodic theorem, this quantizer should also be optimal for future data produced by the same source. We begin with the codebook design problem of how to obtain the initial codebook $\mathbf{C}_0$.

Codebook Initialization:

There are a variety of techniques to initialize a codebook (Gersho and Gray 308-311). We survey only two of the most widely used.

The first method is called *k-means* method. It is simply to select the first N training vectors as codewords, which form the initial codebook.

The second one is called *pruning* method. A training sequence is used to populate an initial codebook recursively as follows: Put the first training vector in the codebook. Then compute the distortion between the next training vector and the first codeword by equation (3.8). If it is less than some predefined threshold value, then discard the second training vector, or the vector is "pruned". If it is greater than the threshold, add the new vector to the codebook as a codeword. With each new training vector, find the nearest neighbor in the codebook. Then by using the same pruning process just described the codewords will be obtained until the initial codebook has enough words.

We now come to the iterative codebook improvement algorithm.

The LBG Algorithm:

The algorithm for an unknown distribution training sequence is given as below:

1. Initialization: Fix k = block length, n = length of training-vector sequence, N = size of codebook $\mathbf{C}$, distortion threshold $\varepsilon \geq 0$. Given an initial codebook $\mathbf{C}_0$, and a training sequence $\{\mathbf{t}_j; j = 1, 2,..., n\}$. Set the number of iterations $m = 0$, and the statistical average distortion $D_{-1} = \infty$.
2. Given $\mathbf{C}_m = \{\mathbf{y}_i; i = 1, 2,..., N\}$, find the nearest neighbor partition $P(\mathbf{C}_m) = \{\mathbf{S}_i; i = 1, 2,..., N\}$ of the training sequence: $\mathbf{t}_j \in \mathbf{S}_i$ if $d(\mathbf{t}_j, \mathbf{y}_i) \leq d(\mathbf{t}_j, \mathbf{y}_l)$, for $1 \leq l \leq N, l \neq i$. Use

a suitable tie-breaking rule when it is necessary. Compute the average distortion by equation (3.9): $D_m = D[(\mathbf{C}_m, \mathrm{P}(\mathbf{C}_m))] = \frac{1}{n}\sum_{j=1}^{n} \min_{\mathbf{y}\in\mathbf{C}_m} d(\mathbf{t}_j, \mathbf{y})$.

3. If $\frac{D_{m-1} - D_m}{D_m} \le \varepsilon$, stop iteration with $\mathbf{C}_m$ as the final codebook. Otherwise continue.

4. Find the optimal codewords $\mathbf{y}(\mathrm{P}(\mathbf{C}_m)) = (E\{\mathbf{t} \mid \mathbf{t} \in \mathbf{S}_i\}; i = 1, 2, \ldots, N)$ for $\mathrm{P}(\mathbf{C}_m)$.

5. Set $\mathbf{C}_{m+1} = \mathbf{y}(\mathrm{P}(\mathbf{C}_m))$ and $m + 1 \rightarrow m$, and go to step 2.

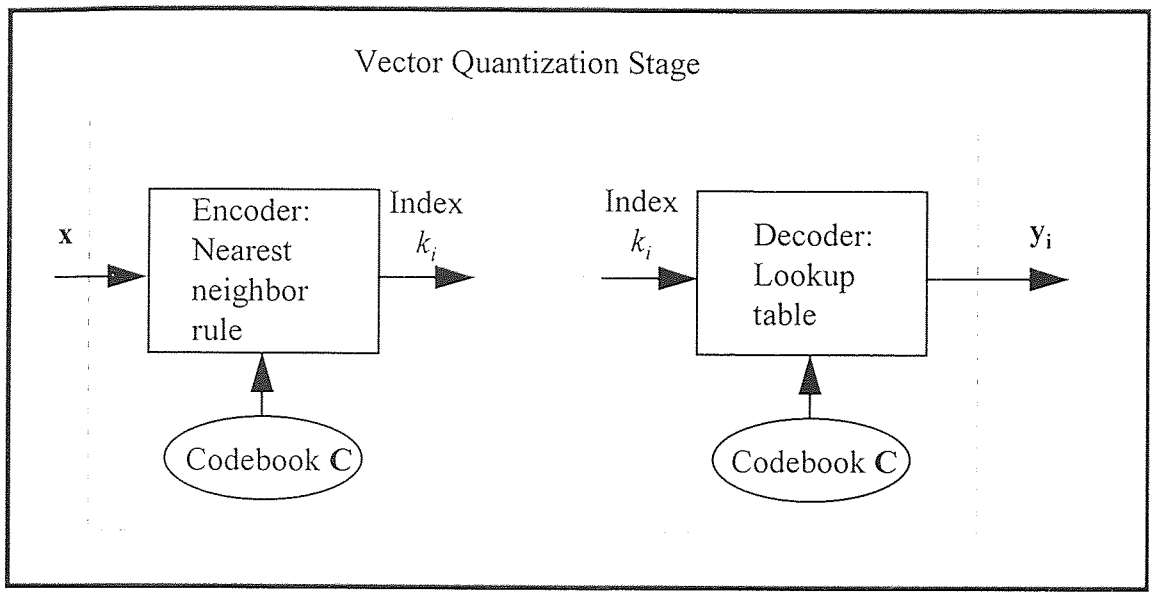


Figure 3.3 A Flow Diagram of the Obtained Optimal Vector Quantizer.

Note that while designing the codebook for vector quantizers, only partitions of the training sequence are considered. Once the final codebook **C** is obtained, this is the end of vector quantizer design. The final codebook together with the NN encoding rule are used on new input data outside the training sequence for the purpose of vector quantization. Figure 3.3 shows the flow diagram of the designed optimal vector quantizer. The encoder views the k-dimensional input vector **x** and searches through the

codebook $\mathbf{C}$ to find the index k_i of a reproduction vector $\mathbf{y}_i$, which is the nearest neighbor of the input vector. The decoder looks up through the codebook and replaces k_i with $\mathbf{y}_i$ as the approximation of $\mathbf{x}$.

By applying the described scalar quantization and vector quantization to the transform coefficients, produced from the first stage, we can achieve some considerable degree of compression and obtain a smaller set of discrete-valued symbols, which are most closely matching those coefficients.

CHAPTER 4

LOSSLESS CODING

Lossless coding is the third, and final, stage in image compression procedure. It translates the discrete-valued symbols, which are from the encoder outputs of the quantization stage, to a more appropriate form of symbols best suited to transmission or storage. We focus mainly on *binary* form of symbols here. The purpose of lossless coding is to provide a binary codeword for each of the encoder outputs. The codeword is not only efficient in terms of the average number of bits per symbol but also exact in the sense that the encoder outputs can be reconstructed with no loss of information.

The lossless coding may be as simple as using fixed-length binary codewords to describe the outputs from the previous stage, or it may use variable-length codewords to achieve higher compression ratio. The latter coding technique is also referred to as *entropy coding*. Figure 4.1 shows the flow diagram of lossless coding stage.

This chapter concentrates on entropy coding techniques. It begins with a quick introduction to some fundamental concepts that is very important to entropy coding.

4.1 Information and Entropy

Information gives an idea of the degree of surprise, the degree to which things are unpredictable and unexpected. If a symbol occurs that is very improbable, we are surprised, and we would therefore expect that the information transferred in coding this symbol would be large. Conversely, if the symbol is very probable, we are not at all

surprised and have learned very little: We already expected that symbol. This qualitative concept, the degree of surprise, is formally expressed by the following relationship. The amount of information I gained after observing the symbol which occurs with probability p is given by:

$$I = \log_2\left(\frac{1}{p}\right) \tag{4.1}$$

The base of the logarithm in equation (4.1) is quite arbitrary. Nevertheless, it is the standard practice today to use a logarithm base 2. The resulting unit of information is called the *bits*.

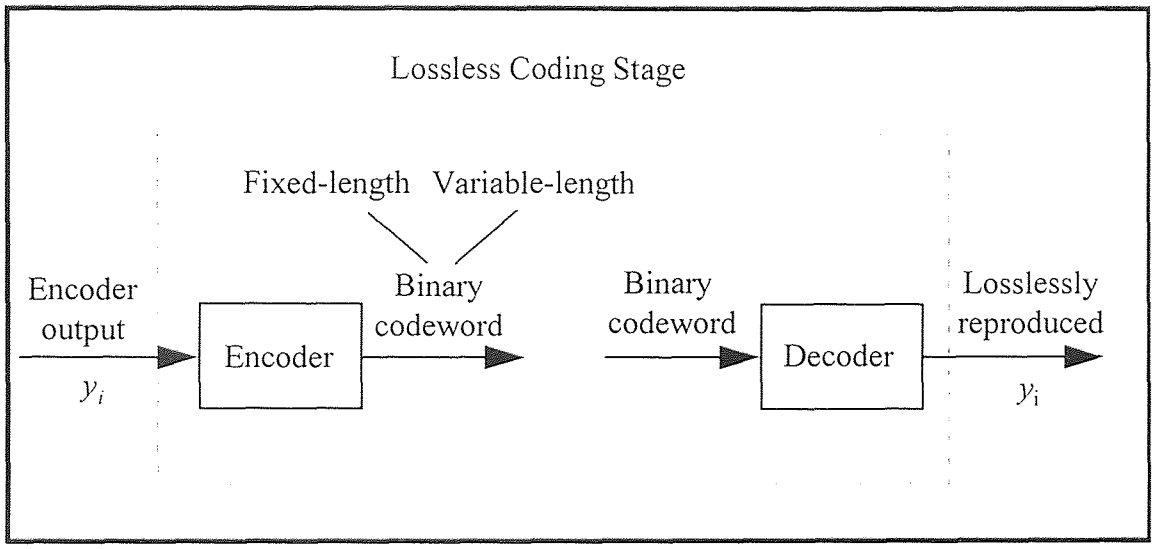


Figure 4.1 A Flow Diagram of Lossless Coding Stage.

Entropy H is a measure of information. It is defined as the average information content per symbol for a set of statistically independent symbols :

$$H = \sum_k p_k \log_2\left(\frac{1}{p_k}\right) \tag{4.2}$$

The concepts of both information and entropy are extremely important in entropy coding. The number of bits of information for a symbol is equal to the ideal codeword length, the number of bits in the optimum codeword for that symbol. The importance of H stems from the *source coding theorem* (Haykin 622), which states that the entropy of a set of discrete-valued symbols gives the lower bound on the average codeword length $\overline{R}$ for any lossless coding,

$$\overline{R} \geq H \tag{4.3}$$

$\overline{R}$ can be made arbitrarily close to H by means of sophisticated coding procedures.

There are at least three techniques that can realize this ideal, *Huffman coding*, *arithmetic coding,* and *run length coding*.

4.2 Entropy Coding Schemes

4.2.1 Huffman Coding

The basic idea behind Huffman coding is to assign each symbol of an alphabet a sequence of bits roughly equal in length to the amount of information conveyed by the symbol in question. That is the highly probable symbol is represented by small length codeword, and vice versa. The end result is a set of codewords whose average code length approaches the lower bound in equation (4.3). In fact, if the input probabilities are integer powers of 2, the bound is achieved. Huffman codeword design algorithm can be summarized in a concise form as follows:

1. Arrange the set of discrete-valued symbol probabilities $\{p_i;\ i = 1, 2,..., N\}$ in a decreasing order.
2. While there is more than one symbol, merge the two symbols of the smallest

probabilities to form a new symbol whose probability is the sum of the two merged probabilities. Arbitrarily assign 1 and 0 to this pair of merged symbols.

3. The procedure is repeated until we are left with a final list of symbols of only two for which a 0 and a 1 are assigned.
4. The binary codeword for each symbol is found by reading backward and tracing the sequence of 0s and 1s to where that symbol is located.

The preceding algorithm gives the Huffman codebook for any given set of probabilities. This also means that the probabilities of all the symbols in an alphabet must be known before a Huffman codebook can be constructed. This is done, in image compression case, by counting symbol occurrences, usually for a large group of images that are considered to be typical for the application. Encoder and decoder operations as shown in figure 4.1 are then done simply by looking up values in the codebook.

As we mentioned before, unless the input probabilities are integer powers of 2, the average code length will not equal to the lower bound. So if we want to improve the coding efficiency relative to that bound, we should code successive pairs or larger blocks of input symbols, that is, consider the input alphabet to be vectors of symbols instead of only single symbols. Then Huffman coding can be applied to this alphabet. For detailed discussion in this topic see Reference (Gersho and Gray 225-260).

4.2.2 Arithmetic Coding

Arithmetic coding is another method of coding that approaches the entropy limit. In arithmetic coding the symbols are ordered on the number line in the probability interval from 0 to 1 in a sequence that is known to both the encoder and the decoder. Each

symbol is assigned a subinterval equal to its probability. Note that since the symbol probabilities sum to one, the subintervals precisely fill the interval from 0 to 1. Figure 4.2 illustrates a possible ordering for the symbols y_1, y_2, y_3, y_4 with probabilities p_1, p_2, p_3, p_4 respectively.

The objective in arithmetic coding is to create a binary codeword pointing to the interval of the symbol being coded (Gersho and Gray 242-248). Thus, if the symbol is y_2, the codeword is greater than or equal to q^2, but less that q. If the symbol is y_4, the code is greater than or equal to $q + q(1 - q)$, but less than 1. If the codewords follow these rules, a decoder can see which subinterval is pointed to by the codeword and decode the appropriate symbol.

Coding additional symbols is a matter of subdividing the probability interval into smaller and smaller subintervals, always in proportion to the probability of the particular symbol. As long as we follow the rules and never allow the codewords to point outside the subinterval assigned to the symbol, the decoder will decode that symbol. Note that the boundary between two intervals is always assigned to one of the intervals.

In cases where extra compression is desired, arithmetic coding is normally used. But the negative about this scheme is added complexity.

4.2.3 Run Length Coding

Run length coding is a simple entropy coding technique originally designed for data compression and later modified for facsimile. It essentially encodes runs of different symbol lengths. Consider a simplest example: After an appropriate two-level quantization, the encoder outputs from the quantization stage, y_1s and y_2s, can be coded

as the number of y_1s between two successive y_2s, that is, the length of the runs of y_1 is coded. Run length coding is very useful when large runs of symbols are expected. It works well in the black-and-white facsimile world.

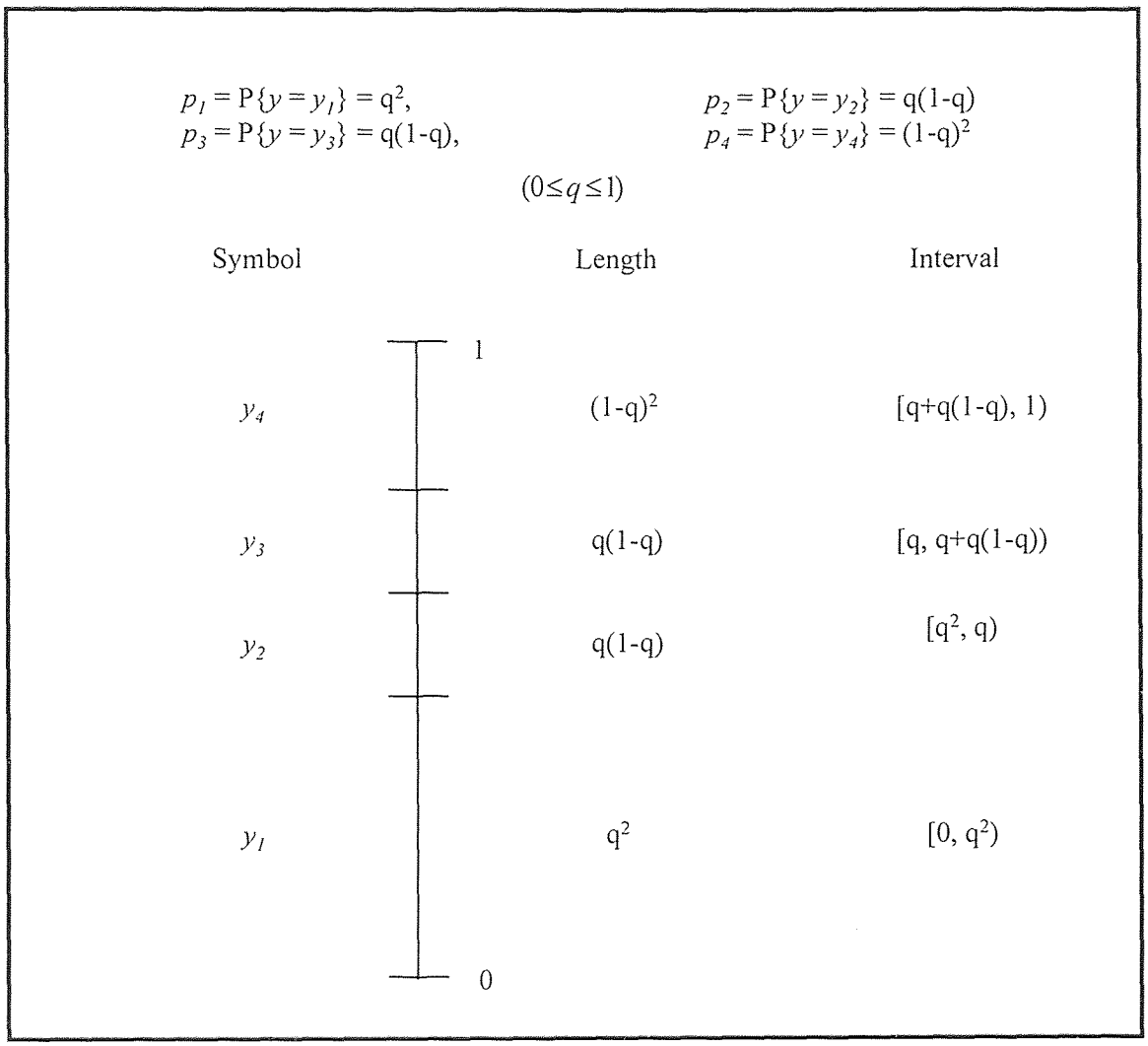


Figure 4.2 A Possible Ordering for the Symbols y_1, y_2, y_3, y_4 with Probabilities p_1, p_2, p_3, p_4 Respectively.

CHAPTER 5

COMPARATIVE STUDIES OF RATE-DISTORTION PERFORMANCE

In the previous chapters, the lossy image compression techniques were presented. Now we are going to simulate some of these image compression techniques. Then we will compare their performances on three test images.

5.1 Simulation of Lossy Image Compression Techniques

Simulation programs which utilize a few popular compression techniques, shown in figure 5.1, were developed. The programs consist of only image transformation and quantization stages. Lossless coding stage is excluded, due to the scope of this thesis. In image transformation, we only focused on DCT and subband transform techniques. In the quantization stage, both scalar quantization and vector quantization were considered.

A flow chart of the simulation programs can be found in figure 5.1. First we input the test image, and the two-band filter bank coefficients, which will be used repetitively for subband transform. The test image is the input to the analysis stage of subband filter bank. Then after the optimal bit allocation, which distributes the number of bits among the subbands from the analysis stage based on the signal energy in each band, each subband is quantized separately according to the quantization bits allocated. It is assumed that an ideal lossless coding with error-free transmission used. Therefore, the input of the dequantization stage equal to the output of the quantization stage. The synthesis stage of subband transform comes right after the dequantization. And finally

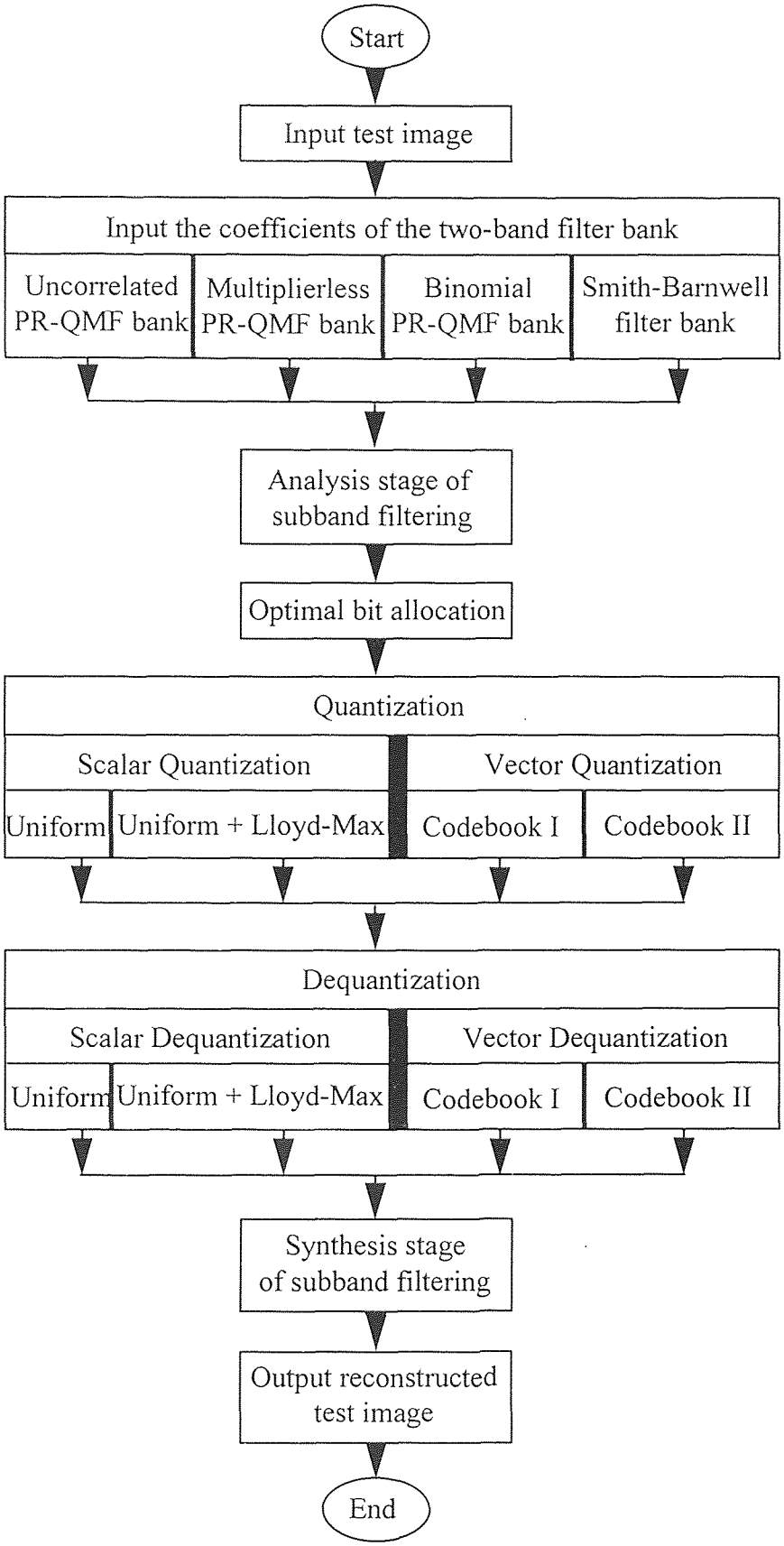


Figure 5.1 Flow Chart of the Simulation Programs.

we obtain the reconstructed test image. The codec simulation programs are further elaborated as the following.

Analysis and Synthesis:

The analysis and the synthesis stages as in figure 5.1 form the subband transform technique for image transformation. The 10-band 2-D subband tree structure, as shown in figure 2.4, is used for the implementation of subband transform. The two-band 8-tap uncorrelated PR-QMF bank (Akansu and Haddad 263-264), multiplierless PR-QMF bank (Akansu 723-734), binomial PR-QMF bank (Akansu and Haddad 241-246), and Smith-Barnwell filter bank (Smith and Barnwell 434-440) are employed separately in this tree structure for the purpose of performance comparison.

Optimal Bit Allocation:

Optimal bit allocation means that the total available bits are efficiently distributed among the subbands, which are from the analysis stage of subband transform, based on their relative subband energies. The optimal bit allocation equation is found as (Jayant and Noll 493)

$$r_i = r + \frac{1}{2}\log_2\left[\frac{\sigma_i^2}{\left[\prod_{i=1}^{K}\sigma_i^2\right]^{\frac{1}{K}}}\right] \qquad (5.1)$$

where r_i = optimal number of bits per ith subband sample

r = the given average number of bits per sample

σ_i^2 = the variance of the ith subband

Equation (5.1) holds only for regular tree structures of subband transform as shown in figure 2.2. For an irregular tree structure, as is the case in these simulation

programs of using figure 2.4, we first properly modified this 10-band structure to a regular one based on the Parseval theorem, and then used equation (5.1) to obtain the optimal bit allocation. Finally the resulting r_i is adjusted in order to fit to one of the existing quantizers: In these simulation programs 2^{r_i} needs to be an integer which equals to one of the following number of quantization levels: 3, 5, 7,..., 33, 35, 64, and 128. Moreover, subbands with negative r_i are truncated to zero and dropped from the future calculation.

Quantization and Dequantization:

Quantization and dequantization operations in figure 5.1 form the quantization stage of a lossy image compression procedure. In quantization, whether it is SQ or VQ, each subband is quantized using the corresponding number of levels found by the optimal bit allocation. Both SQ and VQ are examined in this study.

In SQ, both the uniform quantizers and the normalized Lloyd-Max quantizers for Laplacian pdf, which are also called the *Laplacian quantizers*, are used. Since the histograms of the 10 bands after the analysis operation match either the uniform or the Laplacian pdf quite well. The histograms of subband signals are displayed in figure 5.2. In uniform quantizers, equation (3.7) is used for the quantization. In Laplacian quantizers, the bands are quantized according to the design values in table 3.1 and the values in the more extensive tables, which can be found in References (Jain 108-111).

In VQ, first two sets of codebooks which are called codebook I and II are obtained using the LBG algorithm based on the two different initialization methods of codebook. One is the pruning method, the other is the k-means method. The images "mitp" and "photog", which will be introduced in the next section, are used as training images with 4

× 4 vector size. Then the nearest neighbor method is used to quantize each band based on codebook I and II separately.

5.2 Test Images

The test images used in this study are three randomly picked black and white 256 × 256 images, all with 8 *bits per pixel* (bpp). They are called "lena", "mitp", and "photog", and are shown in figure 5.3.

Now we have the simulation programs which utilize a few lossy image compression techniques, and the test images, so the comparison of these compression techniques can be proceeded.

5.3 Rate-Distortion Performance Comparison of Image Compression Techniques

5.3.1 Performance Measures

Two criteria are used in this study as the performance measures for different compression techniques: The *peak signal-to-noise ratio* (SNR) at the output of the synthesis stage and the bit rate of the compressed image, R.

The SNR is defined as

$$\mathrm{SNR(dB)} = 10\log_{10}\left[\frac{255^2}{mse}\right] \tag{5.2}$$

where *mse* is the mean square reconstruction error (Akansu 727).

The entropies of the 10 quantizer outputs, calculated by equation (4.2), are used to approximate the bit rates of the 10 quantized bands for SQ. For VQ, the bit rate of each quantized band is found as

$$R_i = \frac{\log_2 N_i}{K} \tag{5.3}$$

where R_i = bit rate of the ith band

N_i = codebook size for the ith band

K = vector size (fixed to $4 \times 4 = 16$)

Then by appropriate combination of these bit rates, the bit rate R of the compressed image can be obtained.

5.3.2 Comparative Methods and the Results

Comparison among the various lossy image compression techniques can be made as follows: First the subband transform techniques are compared with each other based on the same quantization procedure. Then the quantization schemes are compared with each other based on the same subband transform. Finally a representative from the subband transform techniques considered in this thesis is selected and compared with the industry standard DCT.

Comparison of the Subband Transform Techniques:

The quantization operation is fixed for subband transform comparison. We arbitrarily chose uniform quantization. This would be the case of comparing different subband transform techniques. We used four different two-band filter bank designs: Uncorrelated PR-QMF bank, multiplierless PR-QMF bank, binomial PR-QMF bank, and Smith-Barnwell filter bank in this thesis. The SNR performance as a function of the bit rate R is compared for these four different filter banks. The rate-distortion curves are displayed in

figure 5.4 for three test images: "lena", "mitp", and "photog". From the graphs, we can draw the following conclusions:

1. The multiplierless PR-QMF bank gives marginally better performance.
2. At bit rates ≤ 0.5, multiplierless PR-QMF bank, binomial PR-QMF bank, and uncorrelated PR-QMF bank perform almost equally well.

Comparison of the Quantization Schemes:

Here the subband transform is fixed. The 10-band subband tree structure with two-band binomial PR-QMF bank is used for subband decomposition.

1. The two scalar quantization schemes are first compared with each other. Figure 5.5 displays the SNR curves as a function of the bit rate R for uniform quantization and the mixture quantization, in which band 1 is uniform quantized and the other 9 bands are Laplacian quantized in order to match the band histograms at the input of the quantization stage. It is observed that uniform quantization performs better at lower bit rates, ≤ 0.75 bpp; while the mixture quantization performs better at higher bit rates, 1-2 bpp.
2. Figure 5.6 shows the performance curves of uniform quantization containing a "dead-zone" for quantizing bands 2-10. Band 1 is uniform quantized. A "dead-zone" represents an interval within which all quantized values are forced to be zero. The length of the "dead-zone", z, ranges from 0-4 in figure 5.6. The results demonstrate that uniform quantization with $z = 2$ gives the best performance.
3. This quantization scheme is also compared with the mixture quantization method. The result is shown in figure 5.7.

So among the considered scalar quantization schemes, the best performance can be obtained if band 1 is uniform quantized and bands 2-10 are quantized by uniform quantizers containing a "dead-zone" of length 2.

For vector quantization, the performance curves by using two different sets of codebooks I and II are displayed in figure 5.8. Vector quantization with codebook II which is obtained by k-means method gives better performance.

Finally the best performer of VQ is compared with the best performer of SQ. The results are shown in figure 5.9. The compressed images are shown in figure 5.10. It is observed that uniform quantization with a "dead-zone" of length 2 performs better.

Comparison with the DCT:

For the completeness of comparisons, the industry standard, 2-D DCT of size 8×8 is also included in this study. First the best performer from the subband transform techniques considered here, multiplierless PR-QMF bank, is compared with the 2-D 8×8 DCT based on uniform quantization. Then these two image transformation methods are compared again based on the uniform quantization with a "dead-zone" of length 2. All these comparisons are displayed in figure 5.11. Some of the compressed images are shown in figure 5.12. The conclusions can be drawn as the following:

1. The 10-band subband tree structure employing the two-band multiplierless PR-QMF bank provides a superior performance to the 2-D 8×8 DCT at bit rates ≤ 1.3 bpp based on uniform quantization.
2. The 2-D 8×8 DCT provides marginally better performance than the 10-band subband tree structure employing the two-band multiplierless PR-QMF bank when using uniform quantization with a "dead-zone" of length 2.

5.4 Conclusions

From the above simulation results, the following conclusions are drawn:

1. In a 10-band subband tree structure, two-band multiplierless PR-QMF bank gives the best performance over the other three filter banks: Uncorrelated PR-QMF bank, binomial PR-QMF bank, and Smith-Barnwell filter bank. Also, at bit rates ≤ 0.5, multiplierless PR-QMF bank, binomial PR-QMF bank, and uncorrelated PR-QMF bank perform almost equally well.
2. Among the quantization schemes considered in this thesis, uniform quantization with a "dead-zone" of length 2 performs best.
3. With uniform quantization, the 10-band subband tree structure employing two-band multiplierless PR-QMF bank outperforms the industry standard 8×8 2-D DCT at bit rates ≤ 1.3 bpp. However, with uniform quantization of a "dead-zone" of length 2, DCT performs better than the multiplierless PR-QMF bank.

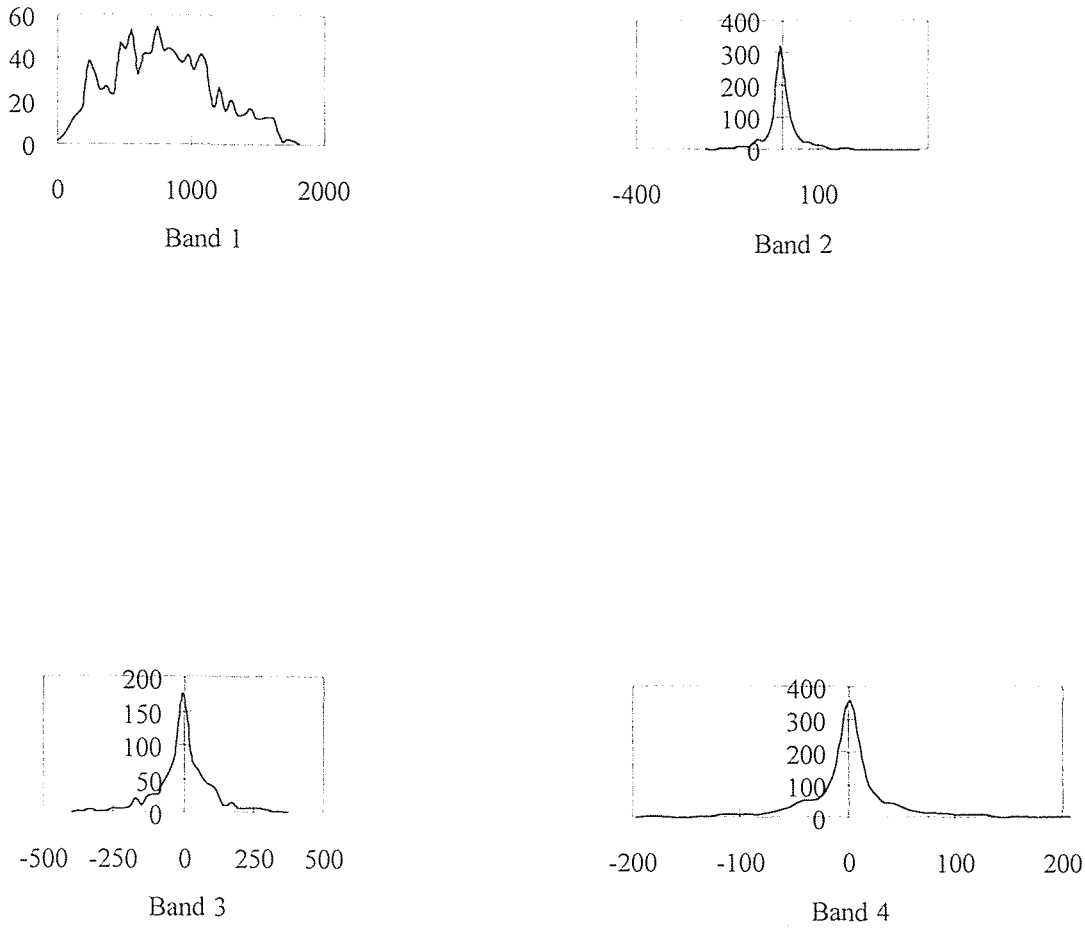


Figure 5.2 (a) The Histograms of Band 1, 2, 3, 4 in the 10-Band Subband Tree Structure Employing Binomial PR-QMF Bank Based on Image A.

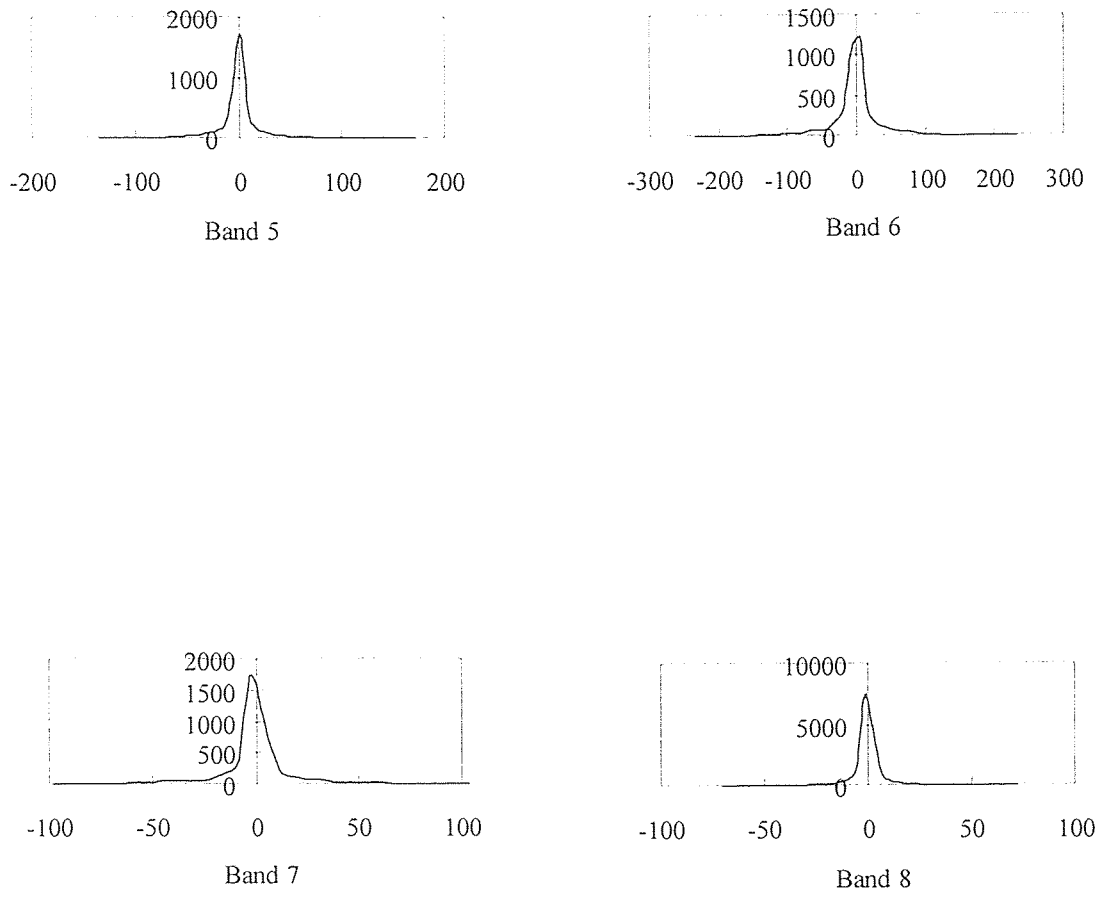
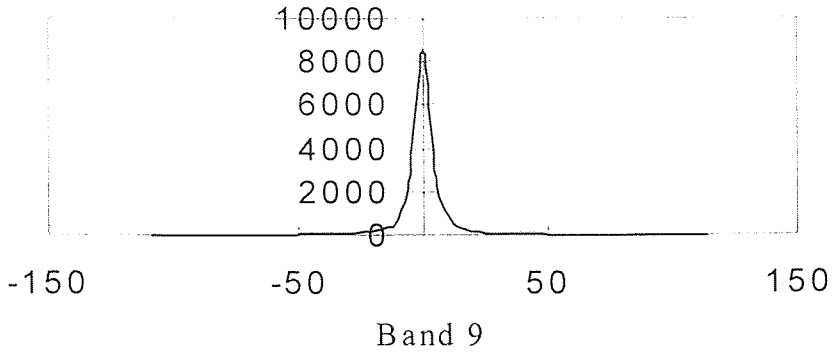


Figure 5.2 (b) The Histograms of Band 5, 6, 7, 8 in the l0-Band Subband Tree Structure Employing Binomial PR-QMF Bank Based on Image A.



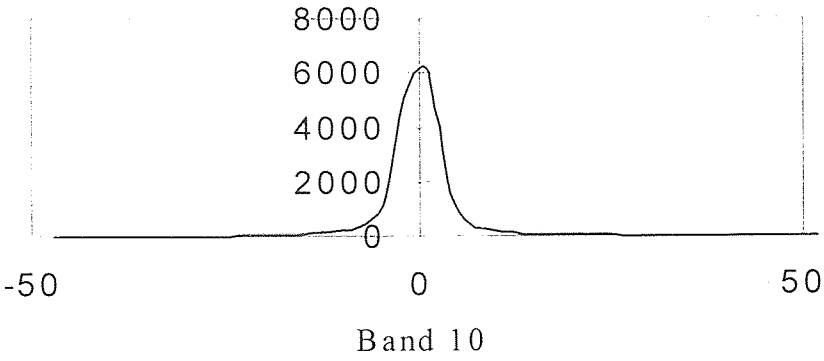


Figure 5.2 (c) The Histograms of Band 9, 10 in the 10-Band Subband Tree Structure Employing Binomial PR-QMF Bank Based on Image A.

Figure 5.3 (a) Image "lena" (256 × 256, 8 bpp).

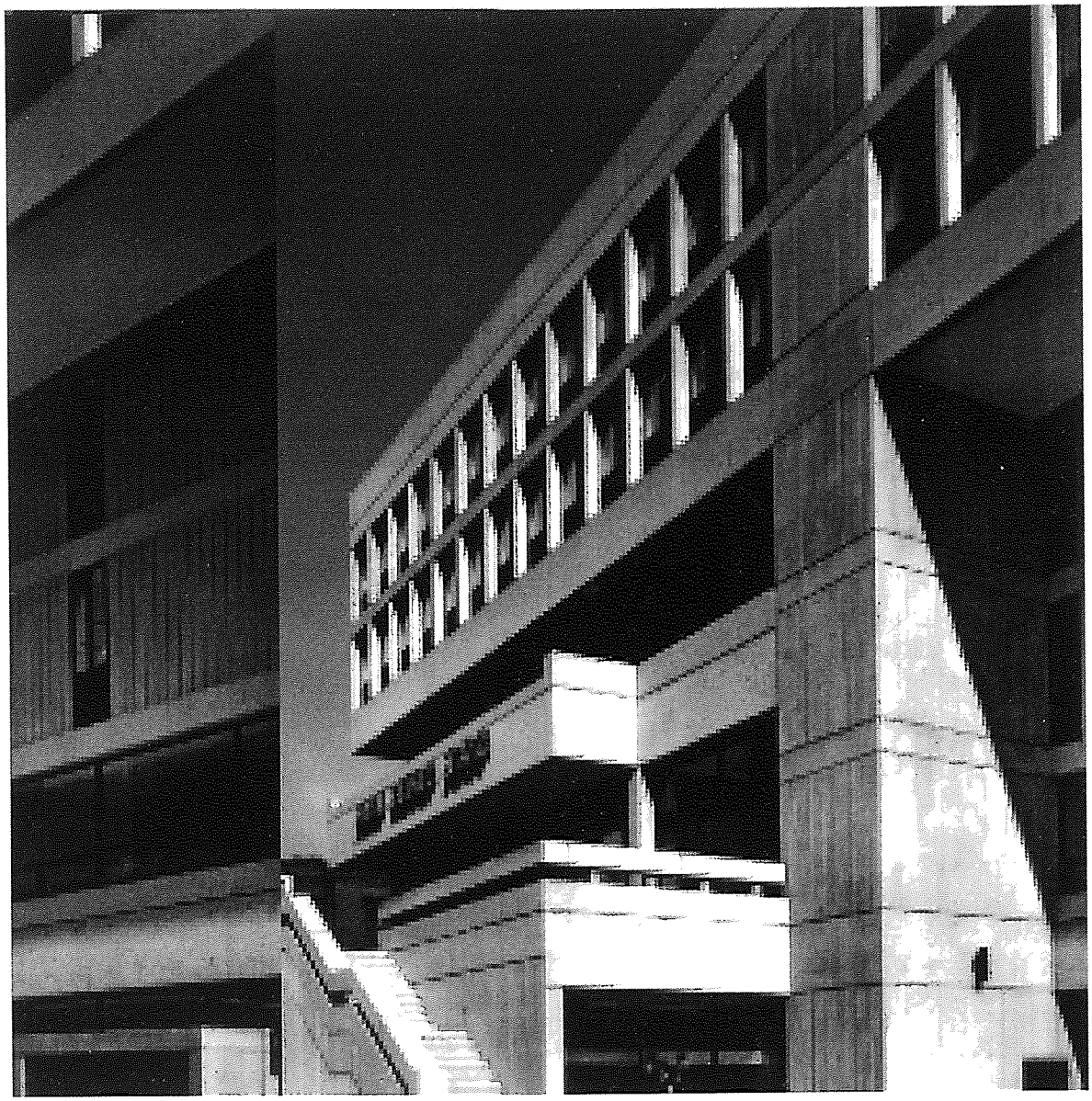


Figure 5.3 (b) Image "mitp" (256 × 256, 8 bpp).

Figure 5.3 (c) Image "photog" (256 × 256, 8 bpp).

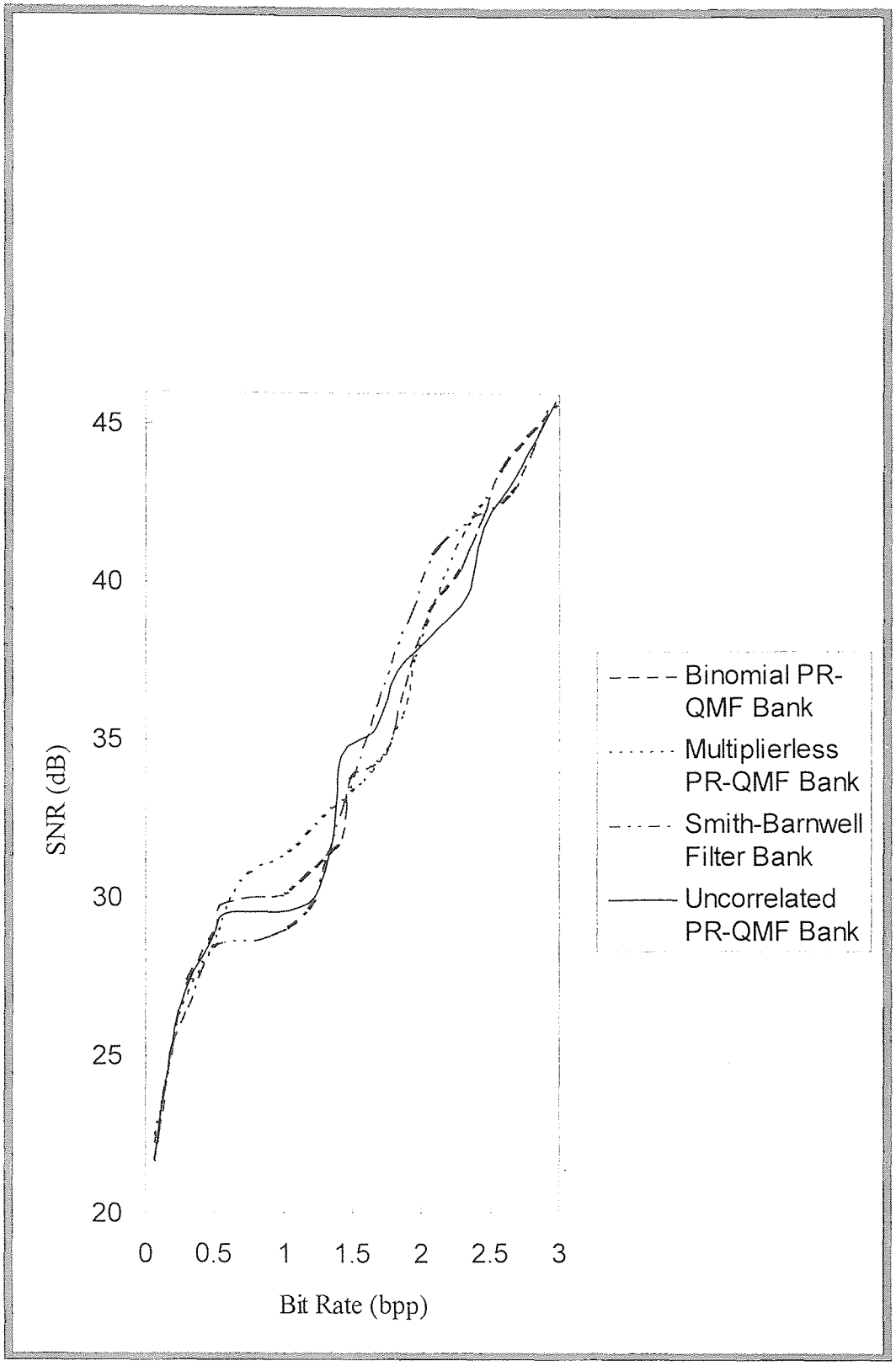


Figure 5.4 (a) SNR and the Bit Rate of Four Different Filter Banks for "lena" Image (Uniform Quantization).

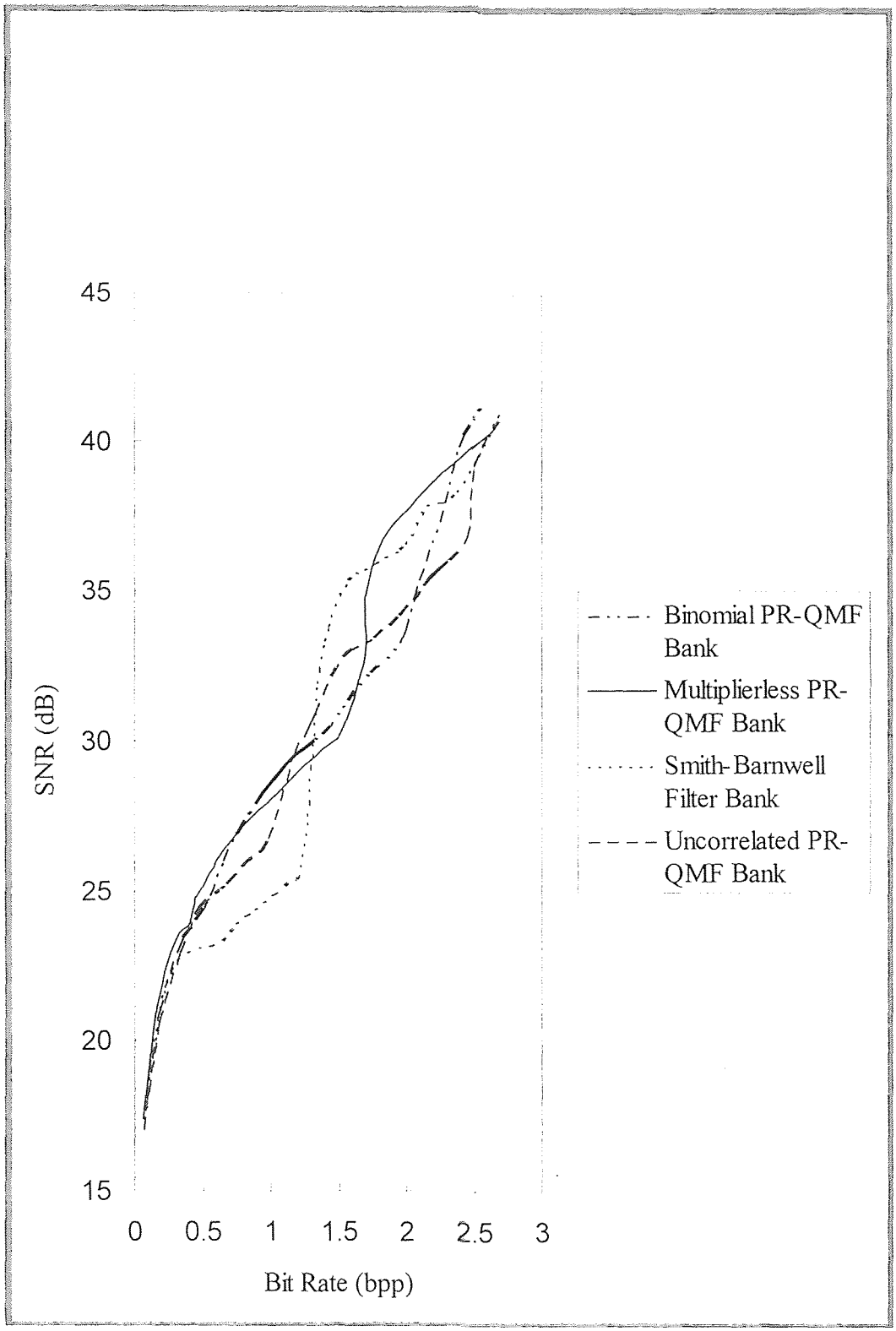


Figure 5.4 (b) SNR and the Bit Rate of Four Different Filter Banks for "mitp" Image (Uniform Quantization).

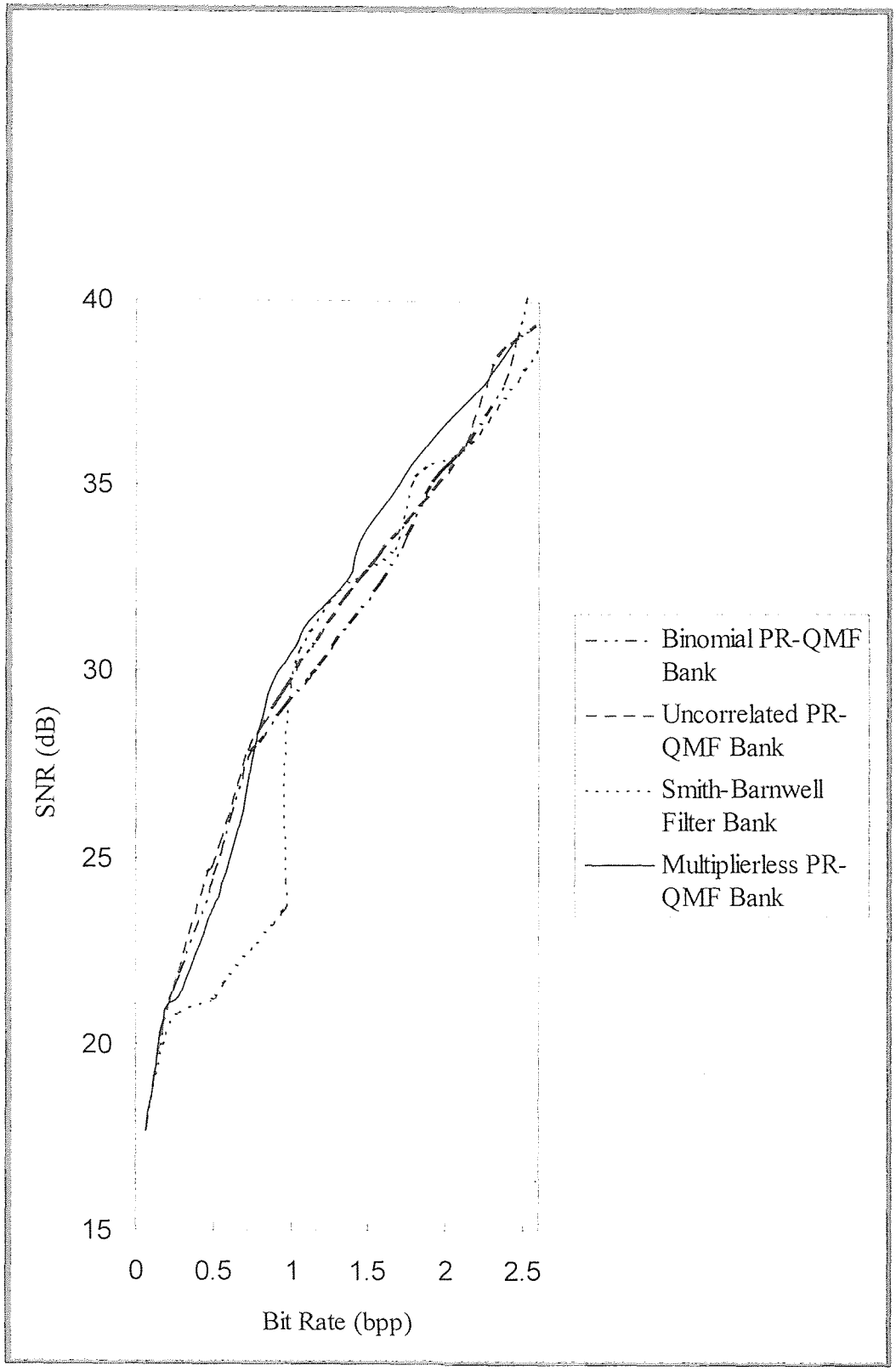


Figure 5.4 (c) SNR and the Bit Rate of Four Different Filter Banks for "photog" Image (Uniform Quantization).

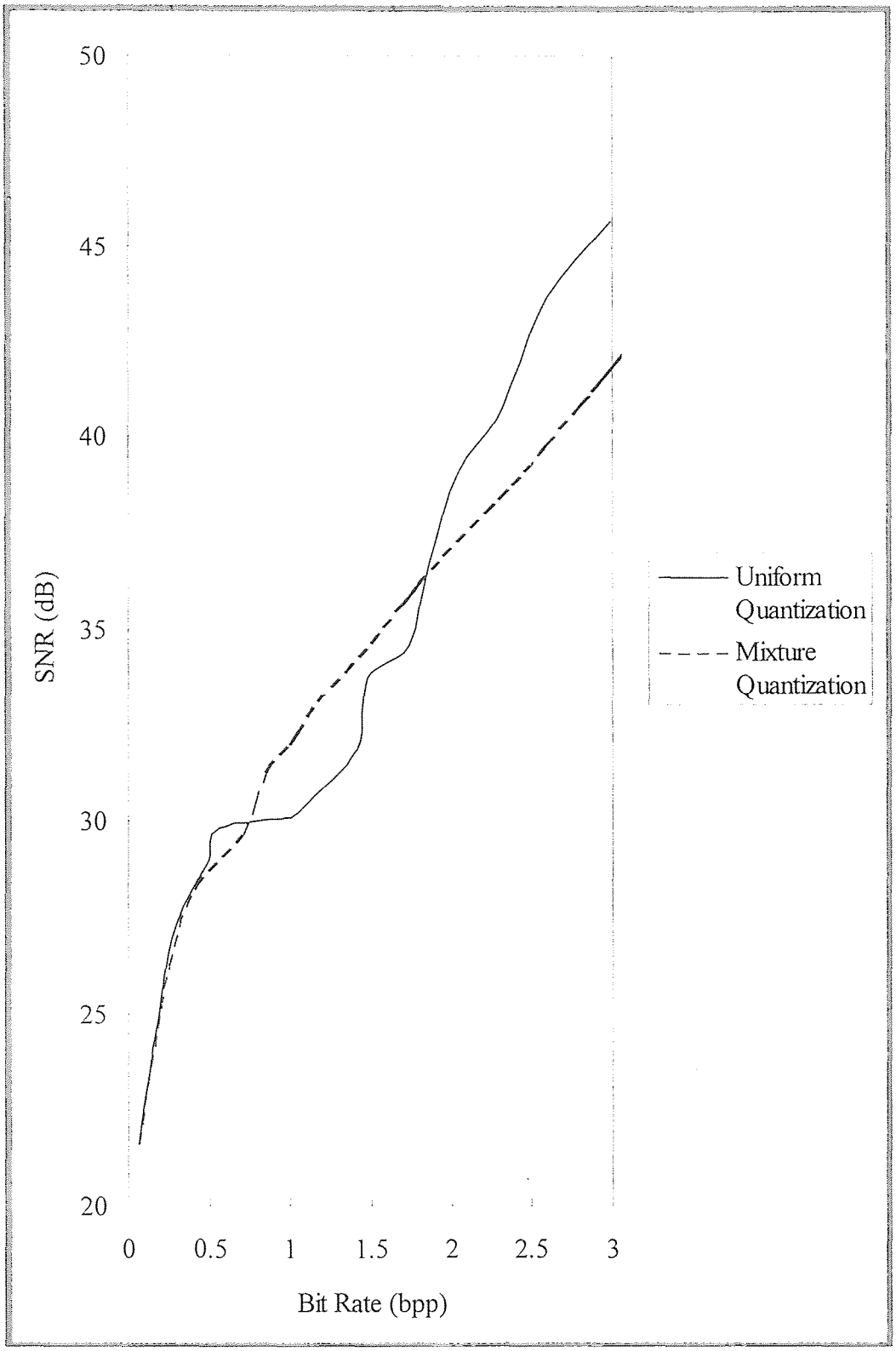


Figure 5.5 SNR and the Bit Rate for Uniform Quantization and the Mixture Quantization Based on "lena" Image (Binomial PR-QMF Bank).

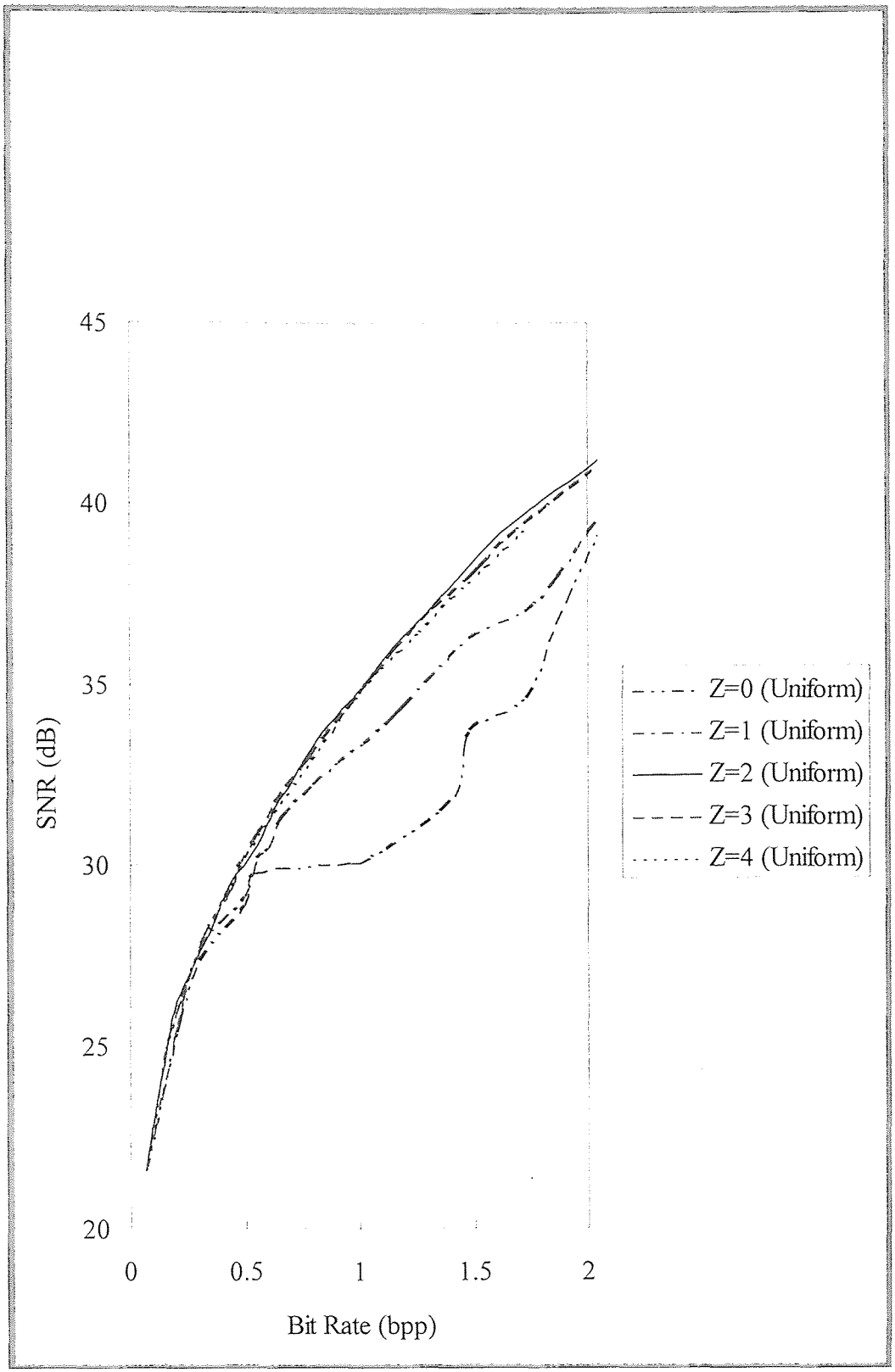


Figure 5.6 SNR and the Bit Rate for Uniform Quantization Containing a "dead-zone" of Different Length for "lena" Image (Binomial PR-QMF Bank).

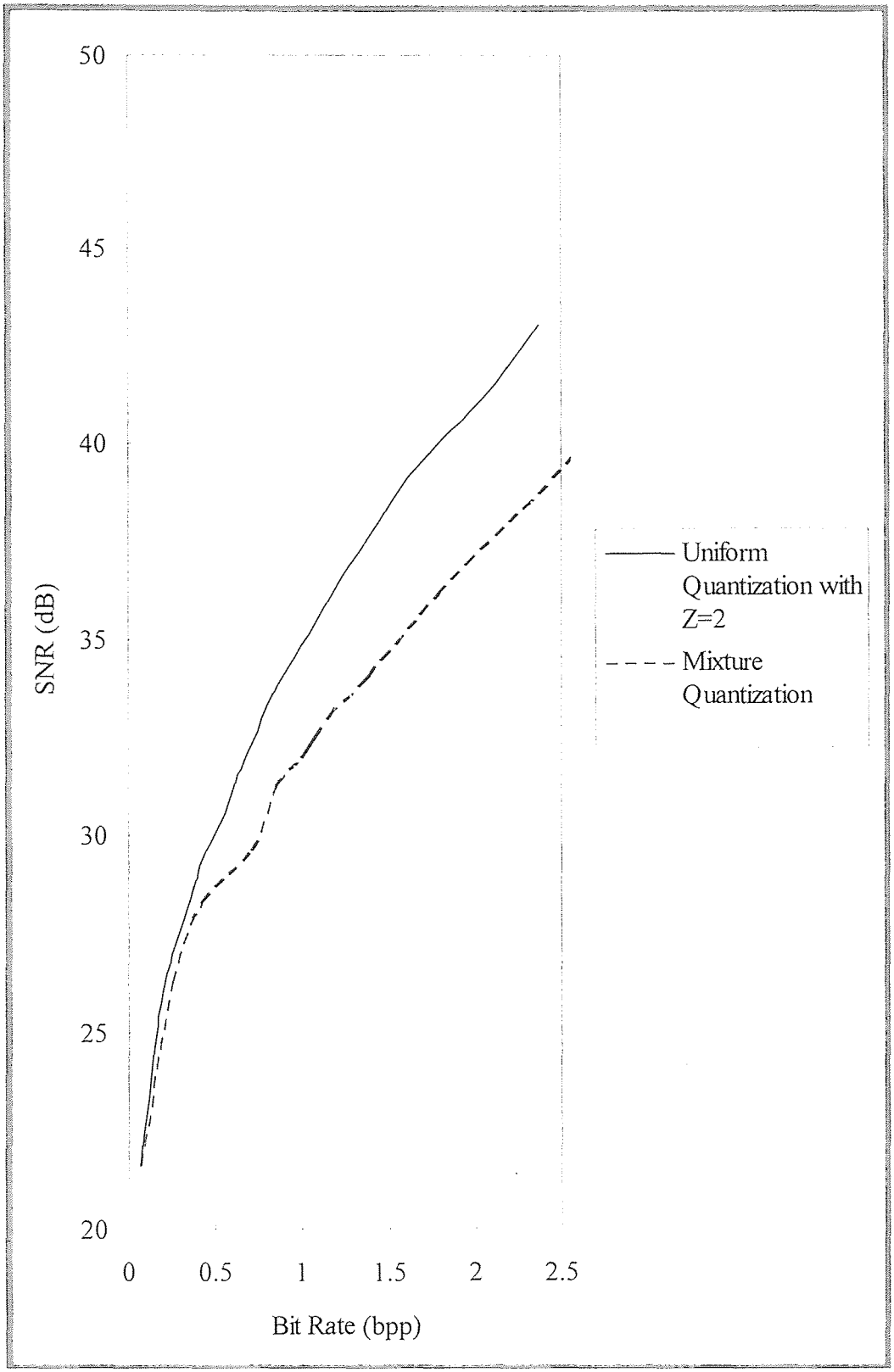


Figure 5.7 SNR and the Bit Rate for Uniform Quantization with $z = 2$ and the Mixture Quantization for "lena" Image (Binomial PR-QMF Bank).

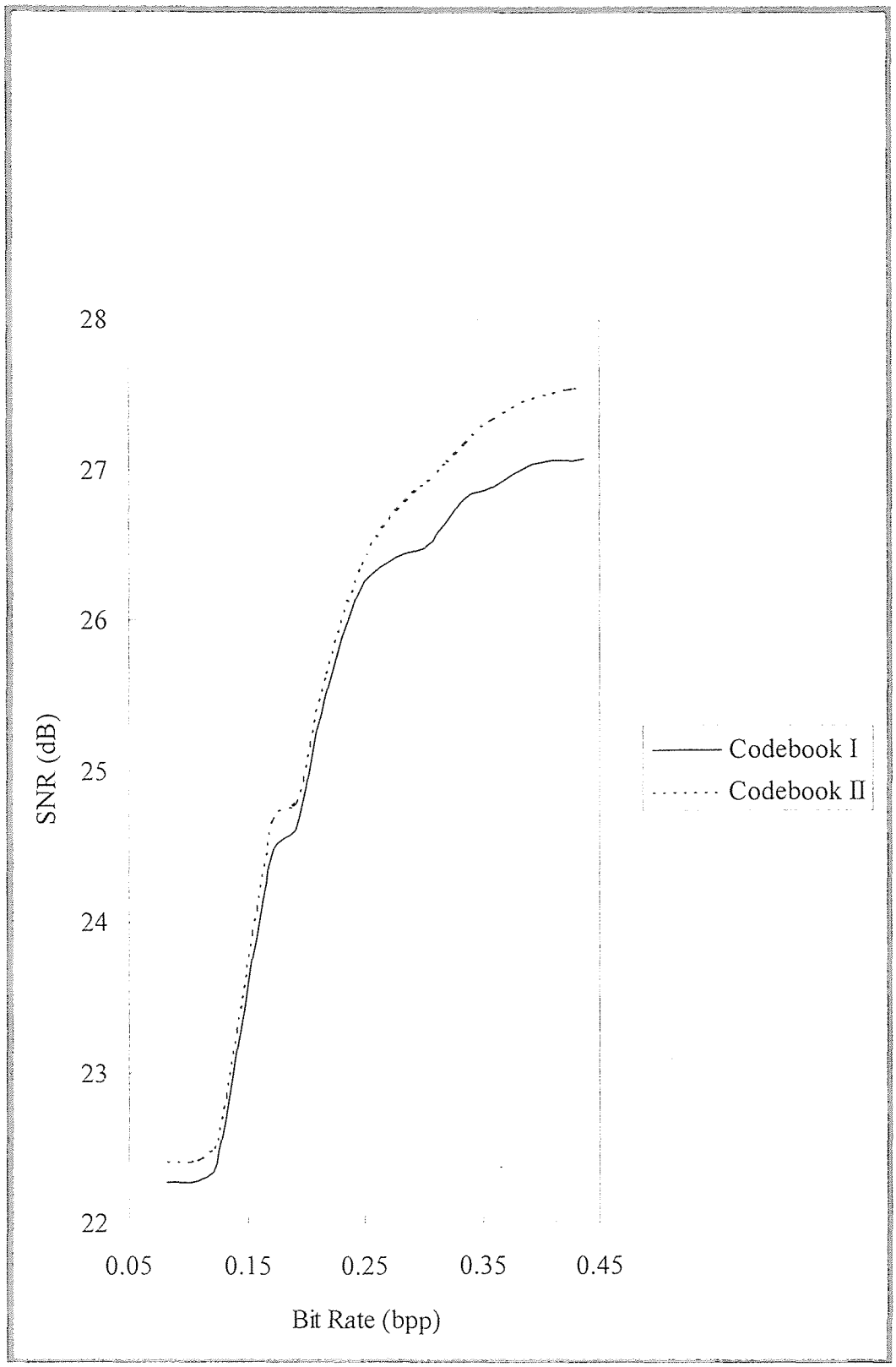


Figure 5.8 SNR and the Bit Rate for Vector Quantization with Codebook I and II Separately for "lena" Image (Binomial PR-QMF Bank).

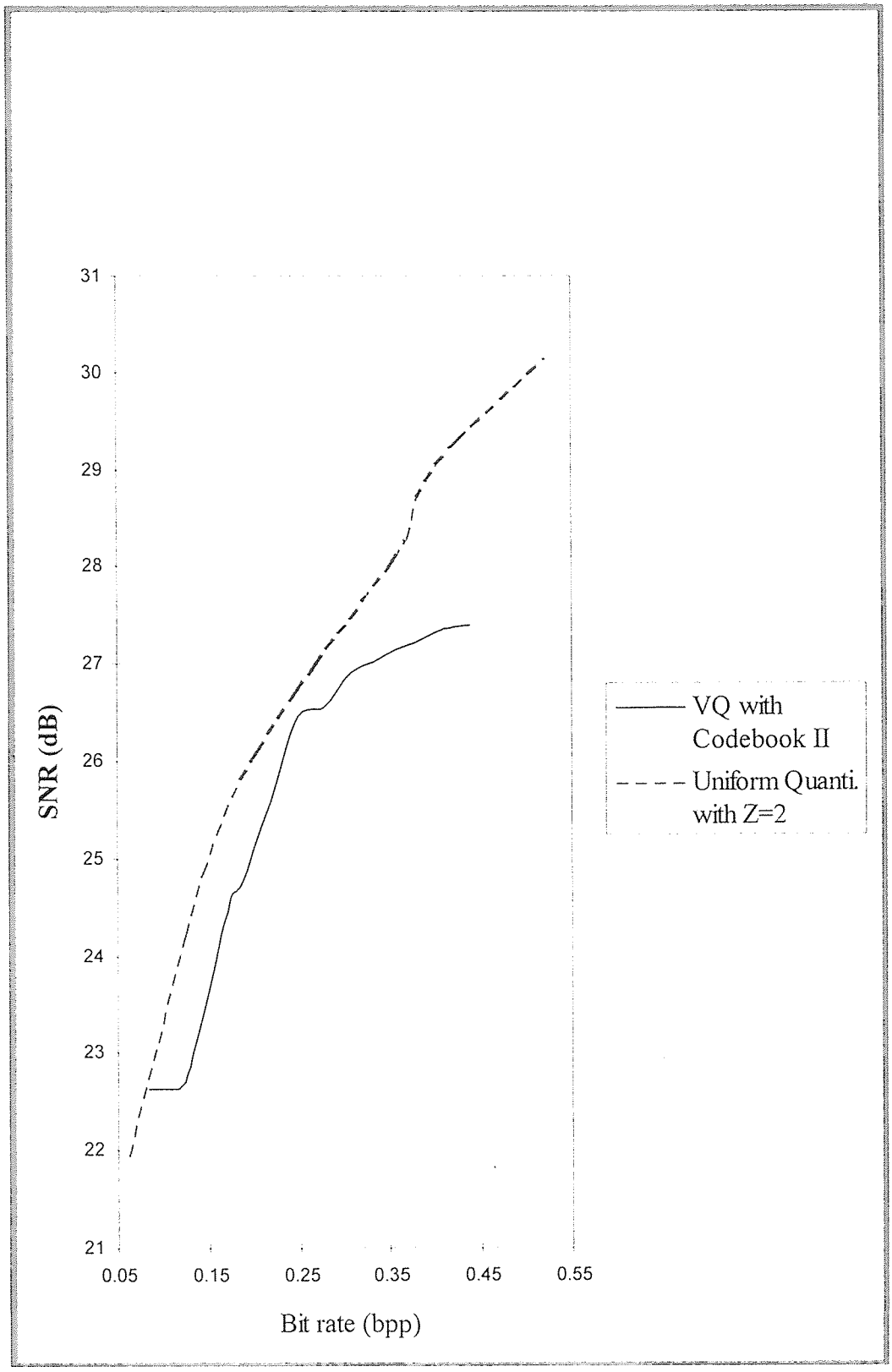


Figure 5.9 SNR and the Bit Rate for Uniform Quantization with $z = 2$ and Vector Quantization with Codebook II Based on "lena" Image (Binomial PR-QMF Bank).

Figure 5.10 (a) Compressed "lena" Image by Uniform Quantization with $z = 2$ at $R = 0.40$ bpp, SNR = 29.04 dB (Binomial PR-QMF Bank).

Figure 5.10 (b) Compressed "lena" Image by Vector Quantization with Codebook II at R = 0.40 bpp, SNR = 27.48 dB (Binomial PR-QMF Bank).

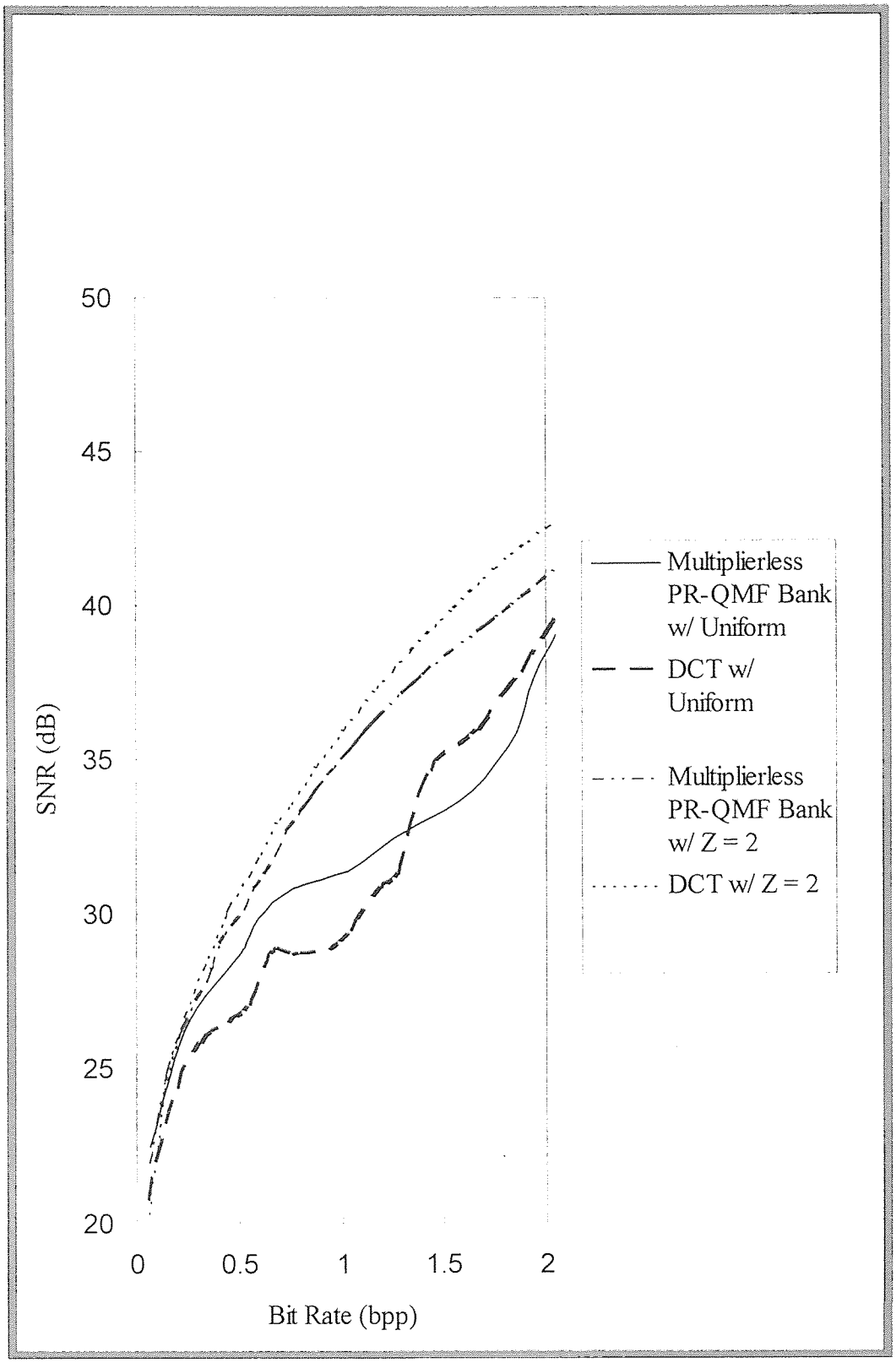


Figure 5.11 (a) SNR and the Bit Rate for 2-D 8 × 8 DCT and the Multiplierless PR-QMF Bank Based on the Different Quantization Schemes for "lena" Image.

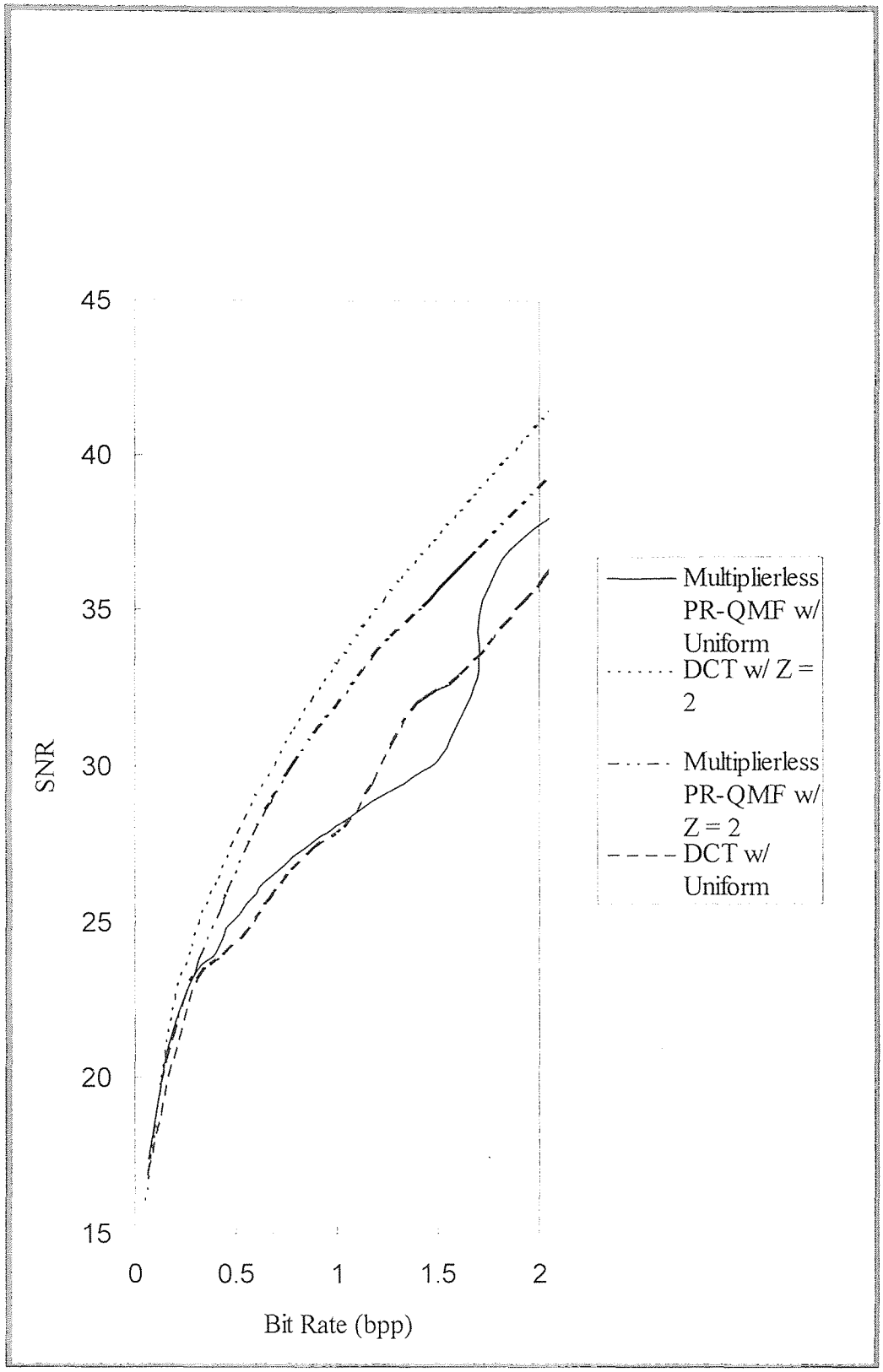


Figure 5.11 (b) SNR and the Bit Rate for 2-D 8 × 8 DCT and the Multiplierless PR-QMF Bank Based on the Different Quantization Schemes for "mitp" Image.

Figure 5.12 (a) Compressed "lena" Image by Multiplierless PR-QMF Bank with Uniform Quantization of $z = 2$ at $R = 0.87$ bpp, SNR = 34 dB.

Figure 5.12 (b) Compressed "lena" Image by 2-D 8 × 8 DCT with Uniform Q uantization of $z = 2$ at $R = 0.87$ bpp, SNR = 34.78 dB.

REFERENCES

Akansu, Ali N. "Multiplierless Suboptimal PR-QMF Design." *SPIE Visual Communications and Image Processing* 1818 (1992): 723-734.

Akansu, Ali N., and Richard A. Haddad. *Multiresolution Signal Decomposition.* San Diego: Academic Press, 1992.

Baxes, Gregory A. *Digital Image Processing.* New York: John Wiley & Sons, 1994.

Gersho, Allen. "On the Structure of Vector Quantizers." *IEEE Transactions on Information Theory* 28 (1982):157-166.

Gersho, Allen, and Robert M. Gray. *Vector Quantization and Signal Compression.* Massachusetts: Kluwer Academic Press, 1990.

Haykin, Simon. *Communication Systems.* New York: John Wiley & Sons, 1994.

Jain, Anil K. *Fundamentals of Digital Image Processing.* Englewood Cliffs: Prentice Hall, 1989.

Jayant, N. S., and Peter Noll. *Digital Coding of Waveforms.* Englewood Cliffs: Prentice Hall, 1984.

Linde, Yoshph, Andres Buzo, and Robert M. Gray. "An Algorithm for Vector Quantizer Design." *IEEE Transactions on Communications* 28 (1980): 84-95.

Max, J. "Quantizing for Minimum Distortion." *IRE Transactions on Information Theory* 6.3 (1960): 7-12.

Pennebaker, William B., and Joan L. Mitchell. *JPEG.* New York: Van Nostrand Reinhold, 1993.

Smith, Mark J. T., and Thomas P. Barnwell. "Exact Reconstruction Techniques for Tree-Structured Subband Coders." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34 (1986): 434-440.

Wong, Stephen, Loren Zaremba, David Gooden, and H. K. Huang. "Radiologic Image Compression-A Review." *Proceedings of the IEEE* 83 (1995): 194-219.

Woods, John W., and Sean D. O'Neil. "Subband Coding of Images." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34 (1986): 1278-1288.