A DUAL HYBRID CODER FOR HIGH QUALITY IMAGE COMPRESSION

By

HANG-CHAN LEE

Bachelor of Science
Inha University
Incheon, Korea
1986

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1992

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 1997

A DUAL HYBRID CODER FOR HIGH QUALITY

IMAGE COMPRESSION

Thesis Approved:

Thesis Adviser

Bennett Basore

John Wolfe

Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor, Dr. Keith A. Teague, for his broad knowledge, intelligent guidance and inspiration, invaluable reviews, and enthusiastic encouragement throughout this research. I would also like to express my gratitude to my committee members, Dr. B. L. Basore, Dr. Scott Acton, and Dr. John Wolfe. Without their valuable suggestions and advise, I would have never finished this project.

I wish to thank to my parents, Dr. Young-Joo Lee and Jong-Rim Cho for their immeasurable love. The rest of my family, Uung-Chan Lee, Kyung-Jin Cha, and Hae-Kyong Lee for their love and unconditional enthusiasm.

My deep appreciation must be accorded to my former professor, Dr. O-Kyu Kwon, of Inha university in Korea for his endless encouragement and concern of me. Finally, my recognition also extends to Dr. Eugene Miles, Mrs. Laverne K. J., and Yong-Jun Na for their assistance to complete this project.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

Table Page

ACRONYMS

AD	Average Difference
AVQ	Adaptive Vector Quantizer
BPP	Bits Per Pixel
CCITT	International Telegraph and Telephone Consultative Committee
COC	Correlation Coefficient
CVQ	Classified Vector Quantizer
DCT	Discrete Cosine Transform
DHC	Dual Hybrid Coder
DPCM	Differential Pulse Code Modulation
EOB	End of Block
HSP	Homogeneity based on Spatial frequency
HVQ	Hierarchical Vector Quantizer
IF	Image Fidelity
ICS	Important Coefficient Selection
ICS-QT	Important Coefficient Selection - Quantization Table
ISO	International Standard Organization
JPEG	Joint Photographic Experts Group
LBG	Linde - Buzo - Gray algorithm
LMSE	Laplacian Mean Square Error

LUT	Look Up Table
MD	Maximum Difference
MOS	Mean Opinion Score
MR	Mean Removal
MR-ICS-JPEG	Mean Removed - Important Coefficient Selection - JPEG
MSE	Mean Square Error
MSVQ	Multi - Stage Vector Quantizer
MT	Multi - Thresholding
NK	Normalized Cross Correlation
NMSE	Normalized Mean Square Error
PI	Proportional and Integral
PMSE	Perceptual Mean Square Error
PSNR	Peak Signal to Noise Ratio
QT	Quantization Table
QTS	Quadtree Segmentation
SPF	Spatial Frequency
SPFO	Spatial Frequency of Original block
SPFR	Spatial Frequency of high frequency Removed block
ST	Single - Thresholding
STD	Standard Deviation
TC	Transform Coding
TVQ	Transform Vector Quantization

CHAPTER I

INTRODUCTION

1.1 Scope

The fundamental goal of *image compression* is to maintain an acceptable fidelity of image quality while bit rate is reduced for data transmission or storage. Since Claude Shannon introduced a measure of information, based on probability definition in 1948 [Shannon, 1948], many compression technologies and their applications have been developed.

Most applications involving digital images are limited unless data compression is used. An example can be seen in multi-media systems such as CD-ROM applications. If we consider full motion video at 30 fps and 720x480 pixel resolution, a 20.736 Mbytes/s data rate is required. As a result, only 31 seconds of video can be stored on a 650 Mbyte CD-ROM. Using current compression technologies, a 650 Mbyte CD-ROM can store more than an hour of video [Bhaskaram, 1995]. As can be seen from the example shown above, compression technology makes many applications more feasible.

The main compression of most transform based compression schemes is achieved by the removal of high frequency components and redundancies using energy compaction property. However, the removal of high frequency components sometimes produces blockiness or lowpass effects at low bit rates. In order to reduce this problem, a *dual hybrid coder (DHC)* which combines the advantage of *vector quantization (VQ)*, high compression, and the advantage of *DCT* based *transform coding (TC)*, high quality, is proposed in this thesis.

A dual hybrid coder can be considerd as a synthetic compression technique which combines somewhat complementary compression schemes, *VQ* and *DCT* based *TC*. Usually, the human eye is less sensitive to the low detail regions such as backgrounds while more sensitive to the high detail region such as edges or curves. Therefore, each image block needs to be treated separately according to its activity. In other words, low detail regions are coded roughly while high detail regions are coded finely to represent edges or curves more accurately. This is the main strategy of the proposed dual hybrid coder (DHC) for improving the image quality.

Before compression, an input image is equally divided into 8x8 blocks. The VQ part of DHC is implemented by codebook adaptive *VQ (AVQ)* where each codebook is generated from the different homogeneous training vectors. Even though the codebooks are not universal, they represent image blocks without annoying distortions because only low detail regions are coded by the AVQ.

Transform coding is fundamentally different from VQ because it is a scalar based coding scheme while VQ attempts to code groups of parameters together. Some additional bits which are made available by AVQ are allocated to the DCT based TC to represent the high detail regions more accurately. The TC of DHC is implemented by a modified JPEG scheme, *Mean Removed-Important Coefficient Selection- JPEG (MR-ICS-JPEG)*. The DHC produces about 3 dB improvement in terms of PSNR over the standard JPEG at the bit rates higher than about 0.30 bpp. However, the performance in terms of PSNR of this scheme is shown to be worse than that of *hierarchical VQ (HVQ)* which is implemented by the proposed optimal *quadtree decomposition* method at bit rates less than about 0.30 bpp. Therefore, the DHC is switched to HVQ at the bit rates lower than about 0.30 bpp. Because of this, the proposed image compression scheme is called the dual hybrid coder. However, this bit rate switching point will be slightly different from image to image. In order to get the general description for bit rate switching point, a linear regression equation as a function of the average block homogeneity for a given image is developed.

Since Habibi introduced a transform-DPCM hybrid coder [Habibi, 1974], several hybrid schemes have been reported [William, 1984; Ngan, 1991; Wen, 1993; Chen, 1993]. In general, *hybridizations* are welcome if they produce better performance at comparable complexity than when each compression scheme is used alone [Clarke, 1985]. Usual hybrid schemes employ *classified VQ (CVQ)* based on edge classifications [Ngan,1991; Wen,1993]. However, these schemes produce high overhead, often more than 5 bits per block, because many edge patterns exist. Moreover, it is very hard to design an efficient classifier. For these reasons, the performance of these schemes is not outstanding.

Instead of classifying image blocks according to their edge patterns, image blocks are classified into eight classes by their homogeneities in the proposed DHC. Next, seven classes of image blocks are coded by the corresponding homogeneous codebooks, where each codebook is generated from different homogeneous training vectors. The remaining class of image blocks, which produces higher activity than the blocks of the other seven classes, is coded by the MR-ICS-JPEG. In other words, the overhead bits are restricted to no more than 3 bits for each 8x8 image block in the DHC. The adaptivity which allows higher bit allocation to high detail regions, and low overhead, are the main sources of the *DHC* used to produce high quality compressed images. Sections 1.2 and 1.3 below introduce DCT-based transform coding and HVQ.

1.2 Transform Coding Based on DCT

The reference quantization table of the *Joint Photographic Experts Group (JPEG)* standard [Pennebaker, 1993] is based on psychovisual thresholding and is derived empirically. The visible image quality has been determined for an image with a pixel resolution of 720x576 at a viewing distance equal to six times the screen width [Pennebaker, 1993; Lohscheller, 1984]. Each DCT coefficient is quantized with the corresponding psychovisual thresholding value. These psychovisual thresholding values are decreased or increased until differences between the results, which are obtained with the quantized and unquantized DCT coefficients, are not detectable visually. The final 64 psychovisual thresholding values, which have been adjusted perceptually, form the quantization table which is used for the quantization of luminance signals in the JPEG scheme. However, this technique may not fully exploit the role of each coefficient in the transformed image blocks.

A new quantization table, referred to here as Important Coefficient Selection - Quantization table (ICS-QT), is constructed so as to exploit the generalized degree of importance of each DCT coefficient in the transform domain using the orthonormal property of DCT. The relative importance of each coefficient is determined by the squared magnitude of each coefficient. This importance ranking forms an effective technique for constructing a quantization table which is capable of producing improved PSNR. As an experiment, we implemented a modified version of JPEG, referred to here as Mean Removed-ICS-JPEG (MR-ICS-JPEG) which makes use of this new quantization table technique. In addition to the changes noted above, we removed the mean of each block before processing so that the DC coefficients in the DCT domain are identically zero. Each mean is transmitted separately in spatial domain to the decoder and later restored in the reconstructed image. This modified scheme, with mean removal and ICS-QT, produces consistently more than 2dB improved PSNR at the output when compared to JPEG alone. Further, we observed experimentally that the performance of the modified procedure does not fall below that of unmodified JPEG; instead, the performances of the two procedures approach each other asymptotically at very low bits per pixel.

Finally, an important issue when comparing competing coding schemes is to ensure that systems being compared are truly operating at the same bit rates. This aspect may also be important in practical use of image compression systems. The user is often interested in knowing prior to compression how much storage space a particular compressed image will occupy or how much bandwidth may be required for transmission. Setting the compression rate in JPEG and similar compression schemes is problematic since compression is data dependent. We present a technique for adjusting the final bit rate based on fitting linear regression equations which are adjusted with ICS-QT and generated using the *Proportional and Integral (PI)* control algorithm in conjunction with statistical approximation [Steel, 1980; Nakagawa, 1992; Sasaki, 1992]. The resulting set of equations guarantees the convergence of bit rate in a few iterations, allowing relatively straightforward specification and attainment of compressed bit rate.

1.3 Vector Quantization and its Hierarchical Implementation Using Quadtree

The purpose of vector quantization is to achieve a low bit rate for storage or transmission with tractable computational complexity and tolerable distortion. In the encoding process, each input vector is replaced by an index of the closest codeword from a set of codebooks. This process requires intensive calculation which increases linearly with the codebook size. The bit rate is expressed as follow:

$$BR = \frac{\log_2 N_c}{k} \quad (bpp) \tag{1.1}$$

where N_c : size of codebook and

k : vector dimension

Equation 1.1 states that if we can increase the vector dimension, *k*, a low bit rate can be achieved. However, increasing *k* results in an exponentially increasing number of calculations. Because of this limitation, input vectors which have a dimension of more than 64 are not recommended even though it is known that the higher dimensionality produces better performance [Gersho, 1991].

In order to reduce encoding complexity, many fast codeword search algorithms have been proposed[1.1] . These methods require some tradeoff between encoding speed and performance. Another approach, *hierarchical vector quantization (HVQ)*, is based on the statistical inter-dependency of contiguous samples and decomposes images with variable resolution. Large blocks and small blocks retain low and high levels of detail, respectively. Therefore, large and small blocks are encoded with codebooks which have low and high quantization levels, respectively. With this strategy, we can reduce the encoding complexity and increase the compression performance.

One of the most commonly used algorithms for implementing HVQ is by the *quadtree segmentation* method.[1.2] An intensive and broad survey of the quadtree segmentation method is given in Samet's work [Samet, 1984]. The general idea of

[1.1] [Cheng, 1984; Cheng, 1986; Lowry, 1987; Soleymani,1987; Paliwal, 1989; Chen, 1991; Torres, 1994; Lee, 1994; Sitaram, 1994].

[1.2] [Jackins, 1983; Grosky, 1983; Samet, 1984; Shaffer, 1987; Strobach, 1989; Shusterman, 1994].

quadtree segmentation is a successive decomposition of a super vector, usually having a block size of more than 8x8. The super vector is divided into new small vectors by a threshold in order to make each region of blocks more homogeneous. This is a very efficient method to reduce the inter-block dependency. However, a problem of the quadtree segmentation (QTS) method is that it is very difficult to determine the optimal threshold to efficiently decompose an image block. Two approaches to overcome this limitation, *single-thresholding and multi-thresholding*, are proposed.

In addition, a *homogeneity* test method based on *spatial frequency (SPF)* is proposed, where *Student's t-testing* method [Steel, 1980] is applied to this method in order to determine whether the homogeneity based on this testing method is classified into success or not. This method not only analyzes the amount of high frequency components included in blocks or regions but also provides homogeneous training vectors for codebook adaptive VQ.

1.4 Quality Measures

Several objective quality measures are intensively exploited and applied to the compressed images generated in the previous chapters in order to ascertain which quality measures are well correlated with the human visual response and which compression scheme produces subjectively more meaningful outputs. Five rating scales (subjective quality measures) are used in conjunction with the results from the objective quality measures. Finally, the *Pearson product-moment correlation coefficient* [Bajpai,1979] method is applied to explore the subjectivity of the objective quality measures. The subjective qualities of several compression schemes, DHC, MR-ICS-JPEG, Standard-JPEG, and HVQ, are compared and ranked. The highest subjective score is achieved by the proposed DHC, and the next ranking is followed by the MR-ICS-JPEG.

1.5 Contributions of the Thesis

A main contribution of this thesis is the design of an efficient image compression scheme, dual hybrid coder, which produdes about 3 dB improved performance in terms of PSNR over the international image compression standard, JPEG. The detailed contributions are as follows:

A new quantization table is developed using the important coefficient selection (ICS) method. It produces more than a 2 dB improvement in terms of PSNR when this table is substituted to the reference table which is included in JPEG. Mean removal (i.e., DC coefficients are zero) before processing also produces a 0.5 dB improvement in terms of PSNR. Proportional integral (PI) control algorithm is applied to calibrate the bit rate. In addition, five linear regression equations are generated by averaging 1,260 scaling factors which provide convergence to PI control in a single iteration at given bit rates and data statistics. These linear regression equations guarantee to provide initial scaling factors which allow PI control to converge within four iterations no matter what the data statistic is.

Two optimal thresholding methods are proposed to decompose an image using quadtree segmentation method; Single-Thresholding (ST) based on statistical approximations and Multi-Thresholding (MT) based on a mathematical modeling. These two thresholding methods are applied to the quadtree segmentation method for hiererarchical implementation of images.

A homogeneity test method is proposed. This method explores the amount of high frequency components which are included in an image block. The Student's t-test method is applied to test the homogeneity of a region. The transition point of homogeneity from '*Fail(F)*' to '*Success (S)*' is located around the bit rate which corresponds to the optimal threshold of the ST method.

1.6 Applications

Image compression has played a crucial role to the growth of multimedia computing technology. The range of applications of image compression is expanded from tele-communication or construction of data bases to the gamut of digital technology. This section explores several possible applications of these.

The areas of application of image compression includes remote sensing to transmit weather or other earth-resource information from satellites, televideo conferencing, document and medical imaging, hazardous waste control applications, facsimile transmission, and the control of remotely piloted vehicles in the military. In addition to these, many other applications are possible.

Image compression can be applied to any multimedia software product to support image display, storage, transmission, and printing. It is also used for personal or public purposes. Driver's photograph, signature, and finger prints can be stored in a digital computer in compressed format to be shared with other jurisdictions including police inspection systems. Publishing areas such as magazines and newspapers need image compression for publishing or archiving their documents in a limited storage space. Image compression also provides economic and creative solutions to complex security and information management. In other words, electronic pass and security systems can be designed using an image compression system by installing ID badges in the data base. In addition, image compression systems can be used for business purposes such as auto or real-estate sales. Images of vehicles or real-estate are stored in a digital computer in compressed forms and can be transmitted into a buyer's terminal to decrease travel expenses.

1.7 Brief Description of Chapters

Chapter 2 explores the fundamentals of VQ, various VQ schemes , and the most commonly used codebook generation algorithm, LBG [Linde, 1980]. The basic structure of the JPEG baseline system is represented in the first half of this chapter.

Chapters 3 through 5 are devoted to the proposed work. Chapter 3 proposes a modified JPEG baseline system. A new quantization table (QT) is generated by the Important Coefficient Selection (ICS) method. Substitution of the JPEG reference QT by ICS-QT produces more than 2 dB improved performance. Next, instead of subtracting 128 uniformly from pixel intensities of an image before processing, which is defined by the JPEG standard, a mean-removed (MR) scheme is applied. This scheme is shown to produce slightly better performance than that of the JPEG standard. In order to calibrate the bit rate, proportional and integral (PI) control algorithm [Auslander, 1990] is applied to the quantizers, and five linear regression equations based on the average block standard deviations of images are proposed. These equations provide five initial scaling factors for the quantization table. These equations are shown to allow the quantizer to converge to the desired bit rate within no more than four iterations. The five bit rates, 0.25, 0.5, 0.75, 1.0, and 2.25, are set by the JPEG standard for comparison purposes [Pennebaker, 1993].

Chapter 4 describes the hierarchical vector quantizer (HVQ) based on quadtree segmentation (QTS) [1.3] and comparison of its performance to that of normal VQ. Two thresholding methods for QTS, single-thresholding (ST) and multi-thresholding

[1.3] [Grosky, 1983; Hanan, 1984; Jackins, 1983; Samet, 1984; Shaffer, 1987; Chiu, 1989; Strobach, 1989; Strobach, 1991; Shusterman, 1994; Sallivan, 1994]

(MT) methods, are proposed and simulated. In addition, A *homogeneity* test method is proposed.

Chapter 5 presents the dual hybrid coder which combines HVQ, AVQ, and MR-ICS-JPEG. The DHC can be considered as a *synthetic compression scheme* which reflects the proposed compression scheme of previous chapters.

Chapter 6 is devoted to various objective and subjective quality measures. The objective quality measures are applied to images compressed by the methods of Chapters 3 through 5. Subjective quality is measured by thirty human observers, and the *Mean Opinion Score (MOS)* [Bhaskaram, 1995] is calculated for each image. The correlation between the results of the objective quality measures and MOS is calculated by the *Pearson Product-Moment Correlation Coefficient* [Bajpai, 1979]. This method provides subjectivity of objective quality measures.

CHAPTER II

OVERVIEW OF IMAGE COMPRESSION AND QUANTIZERS

2.1 Introduction

This chapter is devoted to express the general description and fundamentals of digital lossy image compression techniques which have been successfully used for digital picture communication. The basic concepts of VQ and transform coding, which comprise the proposed dual hybrid coder, are provided.

As one of the commonly used signal compression techniques, transform coding has a relatively long history when we compare it to that of VQ. The basic idea of transform coding is to remove the redundancy in signals using orthogonal transforms such as *Fourier Transform (FT), Discrete Cosine Transform (DCT),* and *Karhunen-Loeve Transform (KLT)* in order to reduce bit rate for storage or transmission. These orthogonal transforms decorrelate pixels from each other. For an image with high inter-pixel correlation, the high frequency coefficients can be coded with few bits or removed completely within tolerable distortion at the rates between 0.5 and 1.0 bits/pixel [Clarke, 1985].

Because a transform which has the capability of mapping many signals to one representative quantity in a perfectly recoverable manner does not exist so far, quantization techniques have been used for image compression at the cost of some loss of image information. This technique provides relatively good subjective visual quality rather than exact matching between original and compressed signals.

Vector quantization techniques have been developed since the early 1980's [Gray, 1984; Gersho, 1983] in order to reduce the bit rate for data transmission or storage. The decoding process of vector quantization is a simple look up table (LUT), and its performance can be comparable to that of transform coding [Clarke, 1985]. These features make vector quantization an attractive signal compression technique. The fundamentals of image compression, basic concepts and variations of transforms coding, VQ, and hybrid coder are provided in the following sections.

2.2 Fundamentals of Image Compression

2.2.1 Redundancy

In general, *data compression* is defined as a process of reducing or removing the amount of *redundancy*, which contains irrelevant information, or simple restates what is already known. The terms, *redundancy* and *compression ratio*, are defined in Eqs. 2.1 and 2.2.

$$R_D = 1 - \frac{1}{C_R} \tag{2.1}$$

where R_D: amount of redundancy and

$$C_R = \frac{UC_I}{C_I}\text{: compression ratio.} \tag{2.2}$$

In Eq. 2.2, UC_I and C_I represent uncompressed and compressed information, respectively [Gonzalez, 1992]. For example, the compression ratio 5 means that five data units (bits) are represented by one data unit (bit). As a result, the redundancy is 0.8. In other words, 80 % of the set of data is considered as an amount of redundancy. Three kinds of redundancies, *coding*, *inter-pixel*, and *psychovisual redundancies*, are identified in digital image compression.

The strategy for removing the *coding redundancy* is to use variable length coding. In other words, the shortest codeword is assigned to a sample which has the most probable occurrence. Average number of bits required to represent each picture element (pixel) is defined in Eq. 2.3 [Gonzalez, 1992].

$$B_{avg} = \sum_{k=0}^{L-1} l(p_k) P_r(p_k) \tag{2.3}$$

where p_k : random variables (pixels),

$P_r(p_k)$: occurring probability of p_k, and

$l(p_k)$: length of codeword assigned to p_k.

Using this approach, we can get compression by achieving average bits shorter than those of assigning a fixed length codeword to each pixel.

A pixel value can be predicted by its neighbors because each pixel is correlated with each other. As a result, the visual contribution from a pixel is relatively small or redundant. This redundant information is called the *inter-pixel redundancy* and can be efficiently removed by a process such as differential pulse code modulation (DPCM).

The human eye does not respond with equal sensitivity for all ranges of pixel intensity. Therefore, a certain range brings relatively less importance to the human eye. This is called the *phycovisual reduandancy* and it might be removed by quantization which is irreversible. As a result of quantization, some information which brings relatively less importance is lost, so this process results in lossy compression. Lossy compression based on quantization is explored in this thesis.

2.2.2 Taxonomy and Trade Offs of Image Compression

The performance of a vector quantization is directly related to the size of the codebook, the training data, and the overall quality of design of the VQ system. VQ systems are usually quite complex to design and implement. A *VQ* scheme allows one to achieve a high compression ratio with tolerable visual distortion.

The transfrom coding technique uses the energy compaction property of transforms such as DCT, FT, and KLT. In this coding technique, each block is individually transformed into a certain coordinate and considered separately. This is usually much simpler than using VQ. Transform coding based on DCT was selected as an international compression standard by the Joint Photographic Expert Groups (JPEG) [Pennebaker, 1993; Wallace, 1991]. Figure 2.1 represents the taxonomy of various lossy compression techniques [Bhakaram, 1995].

Two decent compression algorithms, VQ and Transform coding based on DCT, are explored and modified in this thesis in order to achieve better performance in the senses of low bit rate and high image quality. These two techniques are also applied for implementing the dual hybrid coder in this thesis. Because these techniques are basically lossy compression which is irreversible, they will produce some distortions. Therefore, these lossy compression techniques involve tradeoffs along three dimensions shown in Figure 2.2. Because none of these three factors can be ignored, we may want to achieve moderate compression ratio and medium image quality with tractable computation complexity. The weight of importance of these three factors is application dependent. These three factors have been main goals of image compression.

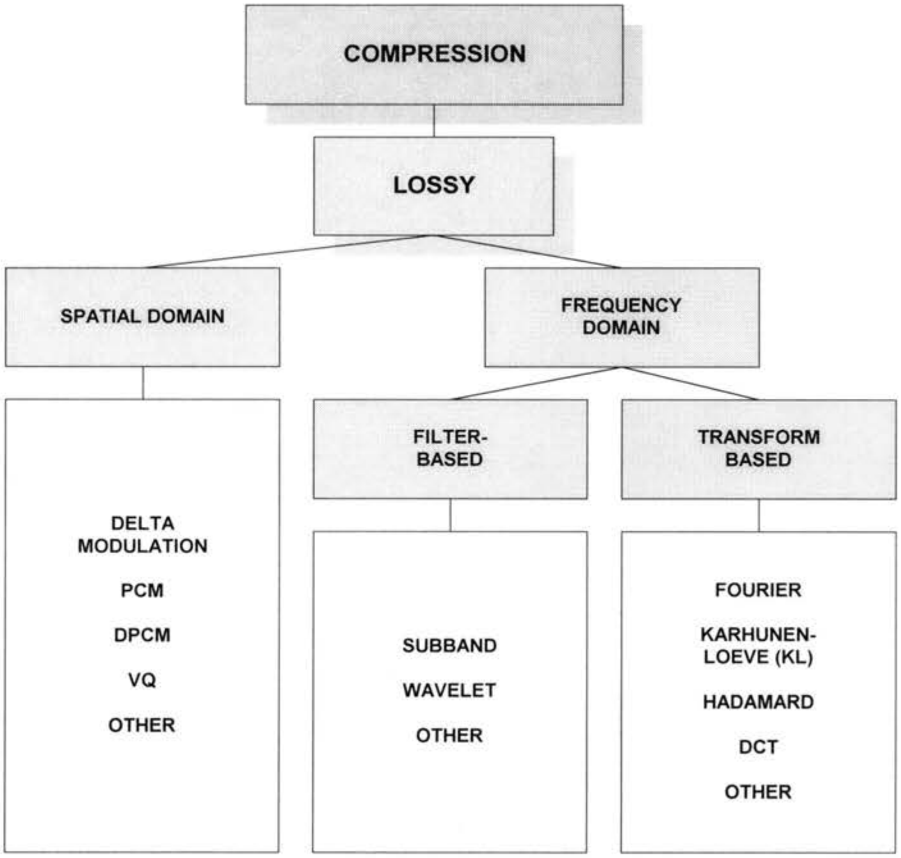


Figure 2.1 A Taxonomy of Compression [Clarke, 1995]

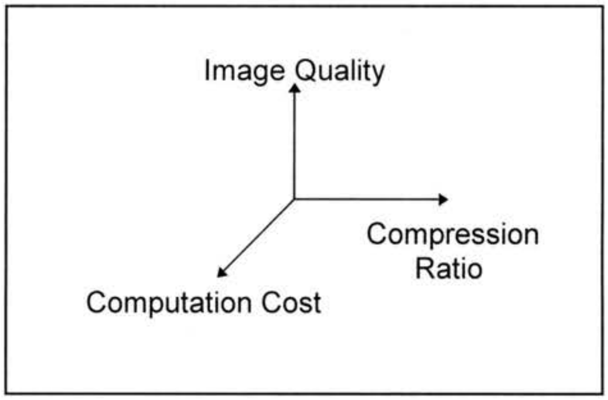


Figure 2.2 Trade Offs in Lossy Compression [Dasarathy, 1995]

2.3 Transform Coding

In transform coding, image blocks are individually transformed into certain coordinates which can be considered independently. Therefore, the purpose of applying a transform before processing is to convert statistically dependent pixels into uncorrelated coefficients. If pixels of an image are highly correlated, the high frequency components will be small so they can be coded with few bits or disregarded completely by a proper quantization scheme. These quantized coefficients are transmitted to the receiver. At the receiver, each quantized coefficient is dequantized and the inverse transform is applied to get a recognizable spatial domain picture. The parameters to decide the performance of transform coding are as follows: selection of coefficients to be transmitted, shapes and sizes of blocks, and types of transforms used [Netravali, 1980]. The basic structure of transform coding is shown in the block diagram of Figure 2.3.

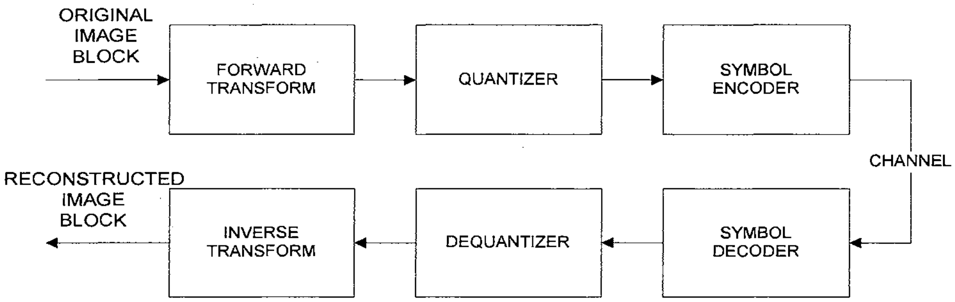


Figure 2.3 Basic Block Diagram of Transform Coding

2.3.1 Comparison of Transforms

The *decorrelation efficiency* and the *energy compaction* property have been used as measures for the performance analysis of transforms. The energy compaction property and the decorrelation efficiency are defined in Eq. 2.4 and Eq. 2.5 [Clarke, 1985], respectively. The energy compaction property (η_{EC}) is defined as a ratio between the sum of M out of N diagonal covariance elements and the sum of N diagonal covariance elements of a $N\times N$ block in the transform domain. The unity value of η_{EC} represents the maximum energy compaction. The decorrelation efficiency (η_{DC}) describes the ratio of the sum of non diagonal covariance elements between the spatial and transform domains. A value of unity indicates complete decorrelation.

$$\eta_{EC} = \frac{\sum_{i=1,j=i}^{M} T_{ij}}{\sum_{i=1,j=i}^{N} T_{ij}} \tag{2.4}$$

$$\eta_{DC} = 1 - \frac{\sum_{i,j=1,i\neq j}^{N} |T_{ij}|}{\sum_{i,j=1,i\neq j}^{N} |S_{ij}|} \tag{2.5}$$

where T : Transform domain covariance matrix and

S : Spatial domain covariance matrix.

Many transforms such as Karhunen-Loeve transform (KLT), discrete cosine transform (DCT), discrete Fourier transform(DFT), Walsh-Hadamard transform (WHT), Haar transform (HT), and discrete sine transform (DST) [Clarke, 1985] have been applied to transform coding . The KLT has been considered as an optimal transform because of its maximum energy compaction and high decorrelation properties of data. However, its implementation complexity and dependency on input data make it very difficult to apply to data compression schemes. Figures 2.4 and 2.5 show the comparison of transforms in terms of their energy compaction and decorrelation properties, respectively.

From Figure 2.4, the performance in the sense of energy compaction of DCT is comparable with that of KLT even though the DCT is not an optimal transform. Figure 2.5 describes the decorrelation efficiency of several transforms when the block size varies from 4 to 16. The decorrrelation efficiency decreases when the block size increases [Clarke, 1985]. The decorrelation efficiency of DCT is relatively higher than others. Therefore, DCT has been the most commonly used transform in the data compression area after Jain documented it [Jain, 1976] because of its high efficiency and easy implementation for fast transform.

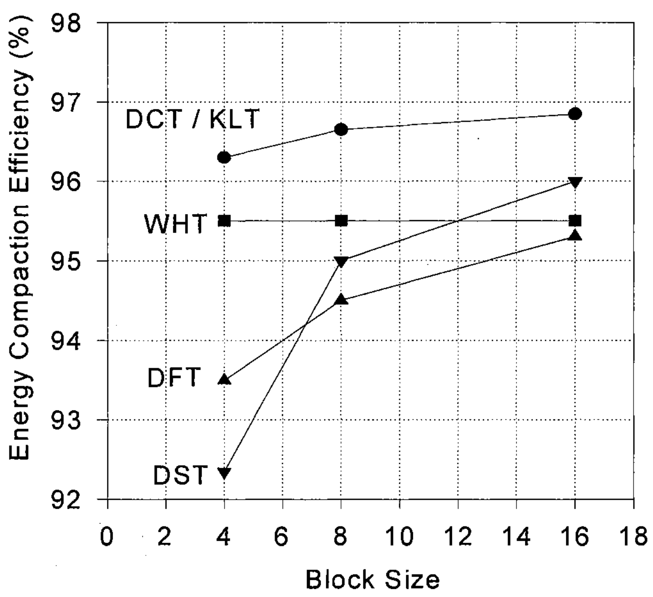


Figure 2.4 Energy Packing Efficiency (η_{EC}) as a Function of Block Size [Clarke, 1985].

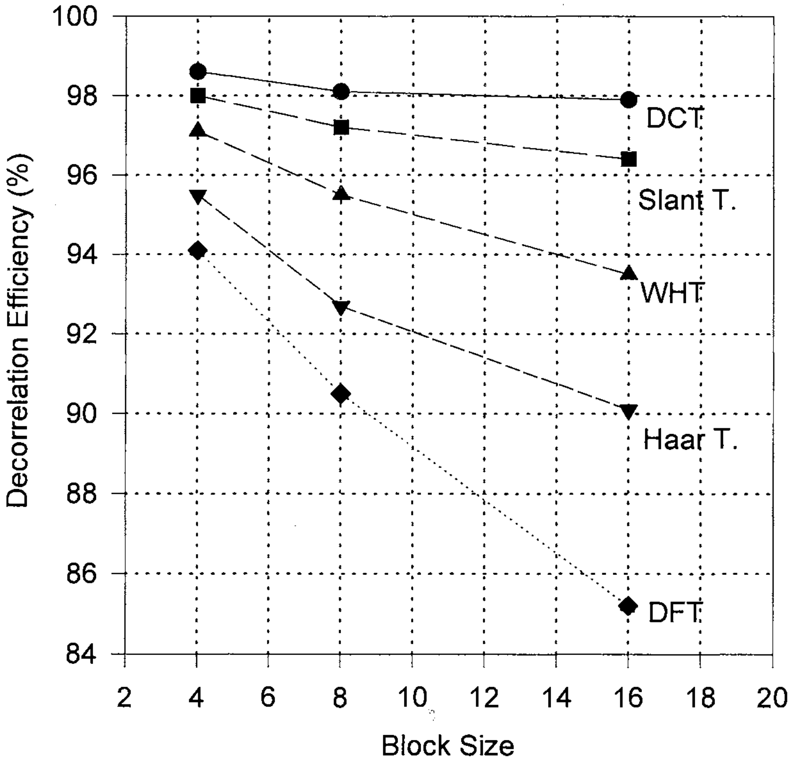


Figure 2.5 Decorrelation Efficiency η_{DC} as a Function of Block Size [Clarke, 1985]

2.3.1.1 Discrete Cosine Transform (DCT)

In general, the DCT is used in block-based image compression algorithms to alleviate the discrete nature of the Discrete Fourier Transform (DFT). The DFT is performed with N point periodic samples where the beginning and end part of the repeated segment are next to one another.

In order to eliminate the discontinuity of DFT, input data is folded about the vertical and horizontal axies in the 2-dimensional spatial domain to make the data be symmetric. After a $2N$ point transform is performed, N point coefficients are selcted because of real and even symmetry of the transform resulting in the DCT. The mathematical definition of the DCT is as follows [Lim, 1990]:

Two-D sequence, *d(x,y),* of $N_1 \times N_2$ points is defined in Eq. 2.6.

$$\begin{cases} d(x,y) & for \quad 0 \le x \le N_1 - 1, \quad 0 \le y \le N_1 - 1 \\ 0 & otherwise \end{cases} \tag{2.6}$$

The 2-D DCT is calculated by the procedure of Eq. 2.7.

$$d(x,y) \quad \overset{folding}{\longleftrightarrow} \quad f(x,y) \quad \overset{DFT}{\longleftrightarrow} \quad F(u,v) \quad \longleftrightarrow \quad C(u,v) \tag{2.7}$$

where *d(x,y)* : $N_1 \times N_2$ point 2-D sequence,

f(x,y) : $2N_1 \times 2N_2$ point 2-D sequence,

F(u,v) : $2N_1 \times 2N_2$ point DFT, and

C(u,v) : $N_1 \times N_2$ point DCT of *d(x,y).*

Because of the symmetric property, the computational burden of a $2N_1 \times 2N_2$ point DFT can be reduced to just a $N_1 \times N_2$ point DFT. This symmetric property reduces the discontinuity of the DFT. As a result, the decorrelation efficiency is improved as shown in Figure 2.5. With these advantages, the DCT has become a basic transform of the international image compression standard (JPEG) which will be discussed in the following section. Finally, the 2-D DCT pair is defined in Eqs. 2.8 and 2.9 [Gonzalez, 1992].

$$C(u,v)=\begin{cases}\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}d(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \\ \quad \mathit{for} \quad 0\le u\le N_1-1, \quad 0\le v\le N_2-1 \\ 0 \quad \mathit{otherwise}\end{cases} \tag{2.8}$$

$$d(x,y)=\begin{cases}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \\ \quad \mathit{for} \quad 0\le x\le N_1-1, \quad 0\le y\le N_2-1 \\ 0 \quad \mathit{otherwise}\end{cases} \tag{2.9}$$

$$\text{where } \alpha(u) \quad \mathit{and} \quad \alpha(v)=\begin{cases}\sqrt{\frac{1}{N}} & \mathit{for} \quad u,v=0 \\ \sqrt{\frac{2}{N}} & \mathit{for} \quad u,v=1,2,3,\ldots,N-1\end{cases}$$

2.3.2 Joint Photographic Experts Group (JPEG)

There are many different kinds of compression methods to transport and store image signals. Although many kinds of compression techniques exist today, they cannot communicate with one another if they employ different coding algorithms. Consequently, a standardization of compression is necessary.

A joint *ISO/CCITT* committee known as JPEG [Wallace, 1991; Wallace, 1992; Wu, 1992] established the international compression standard of continuous tone still images. JPEG is classified into two basic compression methods: a lossy compression and a predictive lossless compression. A simple lossy compression method is called the *baseline method.* This section places emphasis on describing the baseline algorithm in order to study the basic idea of the JPEG coding method which will be used in Chapter 3. The following sub-sections describe each part of the baseline algorithm: quantization, zig-zag run encoder, and entropy coding. Figure 2.6 shows the block diagram of the baseline algorithm included by the JPEG.

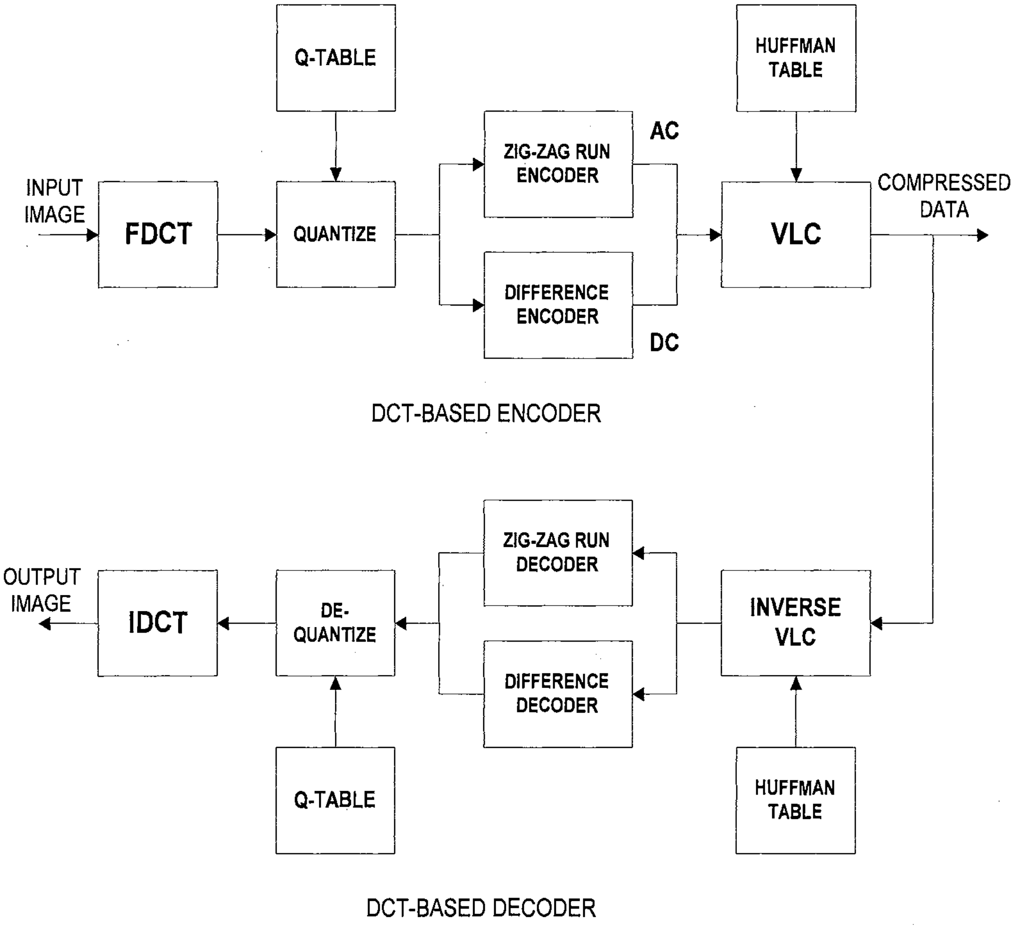


Figure 2.6 Baseline Algorithm Included by *JPEG* (VLC: Variable Length Coding) [Pennebaker, 1993]

2.3.2.1Quantization and Dequantization

Quantization is the principle source of lossiness in DCT-based encoders. Eq. 2.10 and 2.11 define the quantization and dequantization step, respectively.

$$F^{Q}(u,v) = Integer \;\; Round(\frac{F(u,v)}{Q(u,v)}) \qquad (2.10)$$

$$F^{DQ}(u,v) = F^{Q}(u,v)Q(u,v) \qquad (2.11)$$

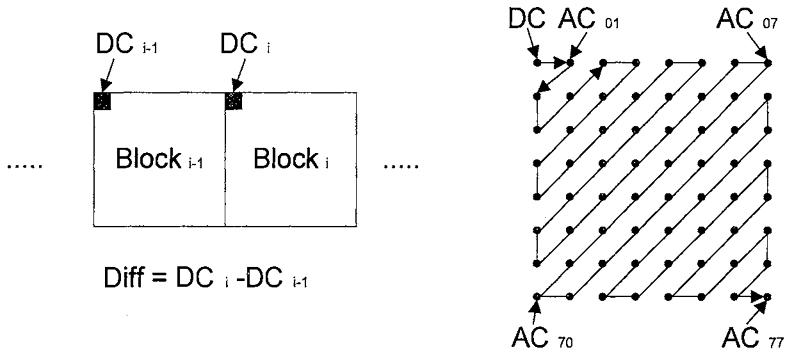
where *F(u,v)* : output of DCT and

Q(u,v) : qunatization table.

The standard block size selected by JPEG is 8×8 because smaller and larger blocks require more bits to indicate the end of block (*EOB*) and the number of zeros run, respectively. An example of the quantization table included by JPEG is shown in A.II.2.1.

2.3.2.2 Data Ordering

After quantization, DC coefficients are treated separately from AC coefficients because they are not only a measure of the average pixel intensities of each block, but they also contain a significant fraction of total image energy. DC coefficients are encoded as differences between current and previous blocks because the DC coefficients of adjacent 8×8 blocks are highly correlated with each other. Finally, AC coefficeients are ordered in zig-zag sequences for next entropy coding. This procedure is shown in Figure 2.7.



Differential DC Encoding

Zig-zag Sequence

Figure 2.7 Preparation of Quantized Coefficients for Entropy Coding [Pennebaker, 1993]

2.3.2.3 Entropy coding

This is the final step of the encoding process of baseline JPEG. Additional compression is achieved in this step by encoding quantized DCT coefficients more compactly based on their statistical characteristics. Two entropy coding methods, *Huffman coding* and *arithmetic coding*, are proposed by JPEG members [Pennebaker, 1993]. However, the Huffman table is predefined as a default because it is much simpler than arithmetic coding. The Huffman table indicates the number of bits required to represent DCT coefficients and it also represents the number of zero-runs of AC coefficients. Table A.II.3.1 summarizes the number of bits required to represent the actual sign, magnitude, and catagories for Huffman coding of DC coefficients. Usually, DC coefficients are coded by DPCM difference. Actual Huffman codes corresponding to the categories are given in Table A.II.3.2 [Pennebaker, 1993].

Table A.II.3.3 shows the number of bits required to encode sign and magnitude of AC coefficients for baseline algorithm. Table A.II.3.4 represents the indicators of Huffman table correspondng to number of zero-runs and addititional bits of Table

A.II..3.3. Tables A.II.3.3 and A.II.3.4 are applied to encode AC coefficients. Actual Huffman codes corresponding to indicators of Table A.II.3.4 are listed in Table A.II.3.5 [Pennebaker, 1993].

2.4 Vector Quantization

Usually, scalar quantization follows two steps: first, partitioning the range of possible input values into finite subsets or subranges; and second, choosing a representative value for each subset when an input value is in that subrange. With vector quantization (VQ), the same two operations take place not in one-dimensional scalar space but in a N-dimensional vector space [Gersho, 1982]. In other words, VQ can be considered as a form of pattern recognition where a predetermined set of standard patterns is replaced for approximating an input pattern.

2.4.1 Principles of Vector Quantization

A vector quantizer is defined as a mapping *Q* of *k*-dimensional Euclidean space E^k into a finite subset *Z* of E^k. Thus,

$$Q\!: E^k \rightarrow Z \tag{2.12}$$

where $Z = (\hat{X}_i\!:i = 1,2,.......N)$ is a set of reproduction vectors and

N is the number of vectors in *Z*.

E^k consists of *n* partitions of *N* point quantization, i.e.,

$$E_i = E_1, E_2, ...E_n = Q^{-1}(\hat{X}_i) = \{X \in E^k : Q(X) = \hat{X}_i\}\,. \tag{2.13}$$

With this definition, we can write as follows:

$$\bigcup_{i=1}^{N} E_i = E^k \text{ and } E_i \cap E_j = 0, \text{ for } i \neq j\,. \tag{2.14}$$

The quantizer Q is a combination of two functions: an encoder, which generates the address of the reproduction vector specified by $Q(X)$, and a decoder, which uses this address to generate the reproduction vector $\hat{X}$. The encoder EC and the decoder DC are defined as a mapping of E^k into the index set I and into the output Z, respectively. Thus,

$$EC{:}E^k \rightarrow I \text{ and } DC{:}I \rightarrow Z \qquad (2.15)$$

where $I = \{1,2,3, \ldots\ldots N\}$.

The basic structure of the vector quantizer is illustrated in Figure 2.8.

In the encoding process, each input vector is matched to the closest template or codeword from a set of codebooks. The number of input vectors of Figure 2.8, M, must always be larger than the number of codevectors, N, in order to get compression. These encoded indexes of the input vectors are transmitted to the decoder. The decoder reproduces the quantized output vectors, which correspond to their indexes of codewords using a simple look up table (LUT) procedure.

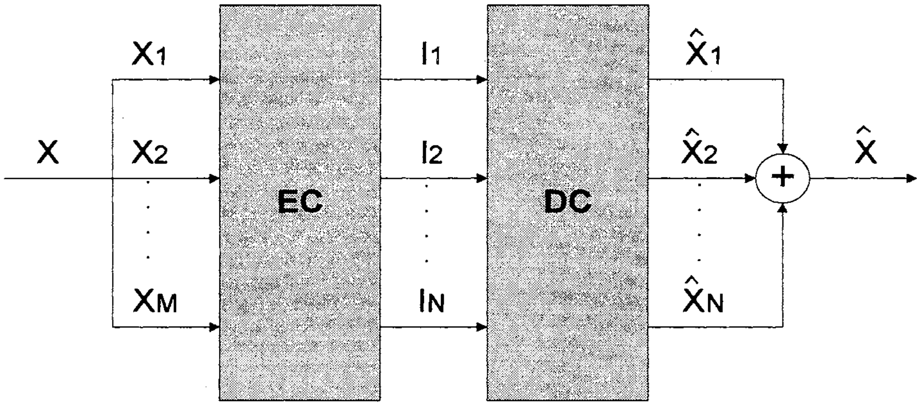


Figure 2.8 Basic Structure of Vector Quantizer

2.4.2 Codebook Generation

The most important part of vector quantization is the design of the codebook. After the *Linde-Buzo-Gray (LBG)* algorithm was published in 1980 [Linde, 1980], several modified schemes and different approaches for generating codebooks have been reported. Although the *LBG* algorithm has drawbacks such as empty cell problem and the requirement of intensive calculation, it is the most commonly used codebook generation algorithm because of its implementation simplicity and applicability for most kinds of images with resulting relatively good performance. This section describes several codebook generation algorithms with emphasis on LBG, which is used in every VQ simulation in this thesis.

2.4.2.1 Linde-Buzo-Gray (LBG) Algorithm

An intuitive and iterative codebook generation algorithm, LBG, is a generalization of *Lloyd's Method I* [Lloyd, 1957] for designing scalar quantizers. Because of this, this algorithm is sometimes called the *Generalized Lloyd Algorithm (GLA)* or *k-means algorithm* after it was studied as a statistical clustering procedure by MacQueen [MacQueen, 1967].

The LBG algorithm starts from an initial guess for the codebook, and the training vectors are partitioned by the nearest neighbor rule. Next, the centroids of each partitioned cell are calculated and the codebook is updated. This procedure is shown in Table 2.1.

In order to calculate the centroids of the clusters, a centroid condition is introduced as in Eq. 2.16.

$$c^* = cent(P) \;\; if \;\; E[d(X,c^*)|X \in P_i] \leq E[d(X,c)|X \in P_i]; \;\; all \;\; c \neq c^* \qquad (2.16)$$

Step1) Given
An initial codebook, $C_m = \{Z_i\}$,
A set of training vectors, T, and
Cluster set P_i, which are partitioned by the nearest neighbor rule (locally optimal partition):

$P_i = \{X \in T{:}d(X,Z_i) \leq d(X,Z_j);$ all $j \neq i$
where, X : input vectors and d : distortion.

Step 2) Find centroids from the just found cluster set in order to update the codebook, $C_{m+1} = \{cent(P_i)\}$.

Table 2.1 The Nearest Neighbor Rule for Updating the Codebook [Gersho, 1992]

However, the centroid is not always uniquely defined. Therefore, the empirical observation and iterative approach of a sample distribution is used to calculate the centroid. At each iteration, the centroid is calculated using the arithmetic mean. The more iterations we perform, the more accurate centroids can be found.

One of drawbacks of the LBG algorithm is the empty cell problem; the empty cell is defined as a codevector that does not have any training vectors or has few training vectors, usually less than three, which satisfy the nearest neighbor rule. Two solutions have been suggested [Gersho, 1992]. The first method is that a cluster which has the highest number of training vectors is split by its centroid, and the empty cell is substituted by one of the centroids of two clusters. The other method also involves splitting a cluster which has the highest partial distortion, and the empty cell is replaced by one of the two centroids of the binary split clusters. The detailed procedure of the LBG codebook generation algorithm is summarized as follows [Linde, 1980]:

Step 1) Initial codebook by binary splitting

1) Initialization: Set *M=1* and define $C_0(1) = X(T)$, where C_0 is the initial codebook, and *X(T)* is a centroid of a set of training sequence.

2) Given the codebook $C_0(M)$, containing *M* vectors $\{Z_i : i = 1,2,.....M\}$. Split each vector Z_i into two close vectors $Z_i + \varepsilon$ and $Z_i - \varepsilon$, where ε is a small fixed perturbation vector. The collection, $\widetilde{C}$, of

$\{Z_i + \varepsilon,\ Z_i - \varepsilon,\ i = 1,..M\}$ has 2*M* vectors. Replace *M* by 2*M.*

3) If $M{=}N \rightarrow C_0 = \widetilde{C}(M)$ and stop.

C_0= initial codebook for *N*-level quantization algorithm.

If $M \neq N \rightarrow$ run the algorithm for an M-level quantizer on $\widetilde{C}(M)$ to produce a good codebook $C_0(M)$ and then return to 2).

Step 2) Codebook Generation

1) Initialization: Given *N*: number of quantization levels,

TH: distortion threshold, $TH \geq 0$,

C_0: an initial codebook, and

$\{T_j : j = 0,...n-1\}$: training sequence.

Set m=0 and $D^{-1} = \infty$.

2) Given $C_m = \{Z_i : i = 1,2,...N\}$.

Find the minimum distortion partition of the training sequence:

$P(C_m) = \{S_i : i = 1,2,....N\}$.

$T_j \in S_i$ if $d(T_j, Z_i) \leq d(T_j, Z_l)$ for all $l \neq i$.

Compute the average distortion

$$D_m = E[D[\{C_m, P(C_m)\}]] = \frac{1}{n}\sum_{j=0}^{n-1} \min_{Z \in C_m} \ d(T_j, Z) \tag{2.17}$$

3) If $\frac{D_{m-1} - D_m}{D_m} \le TH$ stop and then C_m = final codebook.

Otherwise, continue.

4) Find the optimal codebook $\hat{X}(P(C_m)) = \{\hat{X}(S_i) : i = 1,2,\ldots N\}$ for $P(C_m)$. Set $C_{m+1} \cong \hat{X}(P(C_m))$. Replace *m* by *m+1* and go to 2).

An example of this algorithm and a flow chart are given in Appendix II.1 and Figure 2.9, respectively.

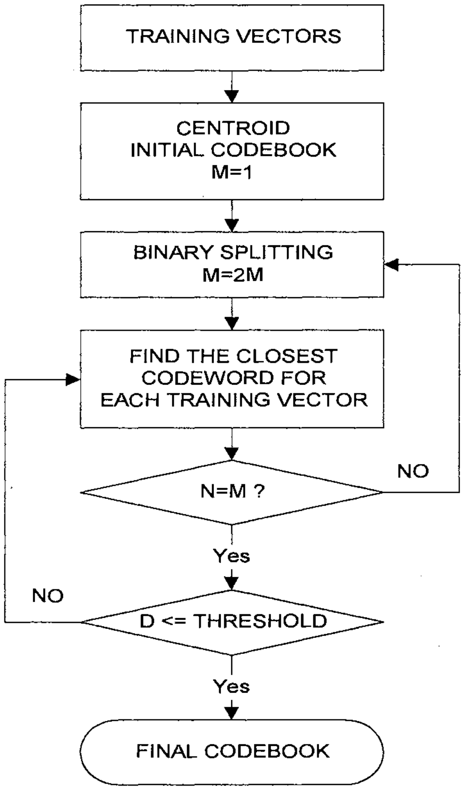


Figure 2.9 Flow Chart of LBG Algorithm

The LBG algorithm is the most commonly used algorithm for generating VQ tables (codebooks). The iteration procedure of the LBG can be combined with other codebook generation algorithms to improve their performances. Because of this, other design algorithms are usually considered as a way to generate an initial codebook. The LBG algorithm is a decent algorithm because each iteration always reduces the average distortion even though its cost cannot be ignored.

2.4.2.2 Other Codebook Generation Algorithms

The LBG algorithm, which was described in the previous section produces a locally optimal codebook [Linde, 1980; Llyod, 1957]. Because only a finite number of iterations can be used when we generate a codebook using the LBG algorithm, a locally optimal codebook is produced. Recently, several alternative algorithms have been developed such as the *pairwise nearest neighbor (PNN)* algorithm [Equitz, 1984; Equitz, 1989] and the *simulated annealing algorithm* [Flanagan, 1989; Kirkpatrick, 1983; Vaisey, 1989]. However the PNN algorithm like the LBG still generates a locally optimal codebook. In order to remove or reduce this problem, the simulated annealing algorithm was developed. Originally this idea came from the *Metropolis Algorithm* [Metropolis, 1953] for atomic simulation. In physical chemistry, the equilibrium state is achieved at the lowest energy state. Using this idea, the simulated annealing algorithm generates a candidate perturbation of partitions at each iteration, and the distortion is calculated at each iteration. Each candidate perturbation of partitions is either accepted or rejected to find the optimal partition. After finding the optimal perturbation of partition, an almost optimal equilibrium state is achieved. However, the complexity of this algorithm is too high and it takes too long to generate a codebook.

In early 1994, Chok-Ki Chan proposed a *maximum decent (MD)* method [Chan, 1994] as an alternative to the LBG algorithm. This algorithm starts by considering whole training vectors as a global cluster, and the global cluster is partitioned into two clusters by an optimal hyperplane. These two clusters are then further subdivided by approaching the desired number of codebook levels. The hyperplane, defined as $H_i = P^t X + k$, partitions a cluster (A_i) into two clusters (A_{ia}, A_{ib}) as follows:

$$\begin{aligned} A_{ia} &= \{X \in A_i : P^t X + k < 0\} \\ A_{ib} &= \{X \in A_i : P^t X + k \geq 0\} \end{aligned} \tag{2.18}$$

where A_i : an ancestor cluster and

A_{ia} and A_{ib} : two partitioned descendent clusters.

The maximum distortion reduction function (R_i) is defined as follow:

$$R_i(H_i) = D(A_i) - [D(A_{ia}) + D(A_{ib})] \tag{2.19}$$

where R_i : always positive if A_{ia} and A_{ib} are not empty,

$D(A_i)$: distortion of the ancestor cluster,

$D(A_{ia})$ and $D(A_{ib})$: distortions of two descendent clusters, and

H_i : hyperplane which produces maximum value of R_i.

Various search algorithms to find the optimal hyperplane have been proposed [Chan, 1994]. These algorithms use a *trial and error* approach; i.e., the distortion reduction values which correspond to a huge number of possible hyperplanes are calculated, and one hyperplane which produces the maximum distortion reduction is selected. In order to generate a *N-level* (*N-cluster*) codebook, *N-1* optimal partitioning is necessary because each partition produces one cluster.

In addition to the algorithms above, many other algorithms using *tree structure* [Ting, 1994; Kiang, 1992; Riskin, 1991] or *triangular inequality elimination pproach* [Chen, 1991] have been proposed. Although some investigators have reported that the results of some of these algorithms are better than that of the LBG, the LBG still remains as the most commonly used algorithm for generating VQ codebooks because it is very simple to implement, reliable, and produces relatively high quality results.

2.4.3 Variations of VQ

Because the normal VQ technique causes complexity and unsatisfactory performance, its direct use for practical applications is somewhat limited. Several modified designs of codebook structure or encoding schemes provide very efficient and favorable trade-offs between complexity and performance. Since the smarter design of the constraint VQ produces the more efficient outputs; i.e., produces low complexity and high performance, the designing of *constraint VQ* can be considered as a realm of art rather than direct use of mathematical theory, which gives solutions for various engineering or science problems. This section describes the most commonly used *constraint VQ* techniques briefly.

Ramamurthi and Gersho designed a *classified vector quantizer (CVQ)* [Ramamurth, 1986] to address the edge degradation and high computational complexity. In this method, each block of a given size is classified by a distinct perceptual feature such as edge, and is coded by vector quantizer codebooks, which are designed especially for that class. This scheme produces a good visual quality for small size 4x4 blocks at the range of 0.6 ~ 1.0 bpp. However, good visual quality cannot be produced for blocks larger than 4x4. Moreover, designing the classifier to classify an input vector with its feature is very difficult.

In *transform VQ (TVQ)*, each pixel of a block is mapped into a transform domain by a linear orthogonal transform such as DCT,DFT, or KLT. An advantage of designing VQ in conjunction with a linear transform is that we can use the energy compaction property; i.e., image information is compacted in a subset of that image. In other words, some of the high frequency components can be ignored or coarsely coded with tolerable distortion in order to reduce bit rate because most information of that image is compacted in low frequency components. Several TVQ algorithms in conjunction with CVQ have been published [Lee, 1994; Kim, 1991; Kim,1992].

If a vector dimension of an input image is quite large, a set of large size codebooks is necessary to code input vectors with acceptable performance. However, the large codebook causes encoding complexity and necessitates a large storage requirement. In order to reduce these problems, *residual VQ* [Frost, 1991; Barnes, 1990], or so called *multi-stage VQ* [Juang, 1982], was developed. In the first stage, input vectors are coded with a considerably reduced size of codebook, and the error vectors between input and coded vectors are transferred to the second stage. These error vectors are coded again with a set of error codebooks in the second stage. This procedure is continued until the desired number of stages is reached. Finally, the coded error vectors for each stage are added to the coded vectors from the first stage. Using this scheme, the total codebook search complexity is considerably reduced. In general, quantization error vectors have more random distribution or less statistical dependence than that of input vectors. As a result, more than two stages do not considerably improve the performance. At this time, more than four stages have not been reported [Gersho, 1992].

Another approach, *hierarchical vector quantization (HVQ)*, was proposed by Nasrabadi [Nasrabadi, 1989]. In this scheme, an image is subdivided into blocks of different sizes by the *quadtree segmentation method* [Hanan, 1984; Shusterman, 1994; Sullivan, 1994]. As a result, each block is almost homogeneous and has little variation. Next, small blocks such as 2x2 and 4x4 are finely quantized, and super blocks are coarsely quantized with low level codebooks or block means. The HVQ examines the correlation between blocks and reduces the inter block correlation automatically. This advantage is a favorable point in the field of HVQ. A detailed exploration of the HVQ is given in Chapter 4.

Table 2.2 summarizes the advantages and disadvantages of several VQ algorithms described above. As can be seen from Table 2.2, the main issues of designing VQ schemes are reducing computational complexity and lowering bit rate.

Methods	**Advantages**	**Disadvantages**
CVQ	- Can reduce edge degradation and computational complexity	- Cannot give good subjective quality when block size is more than 4x4 - Design complexity for classifier.
TVQ	- Can use energy compaction property. - Can use large block size. - Can increase size of codebook without additional complexity for computation.	
MSVQ	- Reduce encoding complexity and storage requirement for codebook.	- Stage limitations: more than two stages are not effective.
HVQ	- Eliminate inter-block correlation automatically. - Reduce the computational complexity.	

Table 2.2 Comparison of Several VQ Schemes

Usually, HVQ is implemented by quadtree segmentation. There are several quadtree segmentation methods. These methods have tried to make segmented blocks or regions become homogeneous. Nasrabadi used variance as a threshold to decompose images, however, no optimality for thresholding is included in this scheme [Nasrabadi, 1989]. One method capable of finding optimal structures was recently dscribed by Chou et al. using a generalized version of the Brieman, Friedman, Olshen, and Stone (BFOS) algorithm [Chiu, 1989]. Sullivan applied the Lagrange multiplier concept to find the optimal multiplication factor for each quad tree level [Sullivan, 1994]. Another approach using plane parameters (PP's) including mean and gradient parameters to get a threshold is studied by Strobach [Strobach, 1989, 1991]. However, none have incorporated a method to find a truly optimal quadtree structure and have not quantified the homogeneities of segmented blocks or regions.

2.5 Hybrid Coding

The purpose of *hybridization* of data compression schemes is to enhance performance by combining two or more simpler compression methods. In 1974, Habibi introduced a *transfom-DPCM hybrid* coder which combined the advantages of the energy compaction property of transform coding and the hardware simplicity of DPCM [Habibi, 1974]. This scheme operates efficiently when input data is highly correlated. Hybridizations are generally welcome if they produce better performance at comparable complexity than when each compression scheme is used alone [Clarke, 1985]. William proposed a *DFT-DPCM hybrid* coder [William, 1984], and a *DCT-DST hybrid* coder was developed by Chen [Chen, 1993].

After VQ techniques were introduced in the early 1980's, these techniques have been used for hybridization. Ngan introduced a hybrid image coder which combined the DCT and classified vector quantization (CVQ) in 1991 [Ngan, 1991]. In this scheme, vectors of low (shade) and high (edge) activity classes are coded by CVQ and DCT, respectively. In order to design the classifier, the block mean parameter is used. In other words, each image block is classified into 64 mean classes where each class employs 64 representative code vectors. As a result, a total of 4,096 code vectors are used for CVQ.

Another hybrid scheme which combines DCT domain VQ and spatial domain VQ was proposed by Wen [Wen, 1993]. In this scheme, input 8x8 image blocks are classified into edge and smooth areas. Then, smooth blocks are coded by a reduced size DCT domain CVQ where a total of nineteen bits are necessary to code a block of smooth area. Next, edge areas are furthur subdivided into 4x4 blocks. These 4x4 blocks are classified again into active and inactive regions. Inactive blocks are coded by block means while active ones are classified into 32 edge patterns and coded by CVQ in the spatial domain. This scheme produces too many overhead bits for consecutive classifications and results in 29.52 (dB) in terms of PSNR for Lenna image at the bit rate of 0.3979 bpp. In addition to the schemes listed above, other notable hybrids which include *wavelet-DCT* and *normal VQ - lattice VQ hybrid* coders were designed by Ohta [1993] and Bage [1986], respectively.

CHAPTER III

TRANSFORM CODING BASED ON DCT

Two conflicting goals are encountered in the design of any image compression system: attainment of the highest possible compression while maintaining acceptable image fidelity. In any realistic compression system, we cannot arbitrarily maximize one without, at some point, beginning to sacrifice the other. This chapter describes a method for generating a new quantization table which, when implemented within the framework of the *ISO/CCITT* compression standard for continuous tone still images (JPEG - Joint Photographic Experts Group) [Wallace, 1992; Wu, 1992; Leger, 1991], is capable of providing several dB higher peak signal to noise ratio (PSNR) at a comparable bit rate than the quantization table in the JPEG standard. Although the application example and results presented in this chapter are directed toward JPEG, this method should be applicable to other scalar compression schemes as well.

3.1 Quantization Table and Quantization

One of the most important factors affecting the performance of an image compression scheme is how well the quantization table matches the data being compressed [Wallace, 1992; Pennebaker, 1993]. This section describes a new method, *Important Coefficient Selection (ICS),* for constructing a quantization table which is shown by example to give better performance based on maximizing the

PSNR. PSNR is defined as follows:

$$PSNR = 10\log_{10}(\frac{255^2}{MSE}) \tag{3.1}$$

The construction technique makes use of the *orthonormal* property of DCT for maximizing the PSNR. In the following paragraphs we present this technique, along with several variations, followed by results obtained experimenatlly with JPEG.

3.1.1 Important Coefficient Selection(ICS) Method

Analysis by Synthesis Technique: The Important Coefficient Selection (ICS) method exploits the fundamental role of each coefficient in an 8x8 standard size block for maximizing the PSNR. In the DCT domain, the values of the AC coefficients represent the interpixel activities, and the values of DC coefficients represent the block size times the corresponding block means. The basic idea of the ICS method using analysis by synthesis technique is very simple. After a block DCT is computed, each DCT coefficient is removed, one by one, and the given block is reconstructed by applying the IDCT. The mean square errors (MSEs), computed between the original and each of the reconstructed blocks, are calculated for all the AC coefficients in an 8x8 block (63 MSEs per block).

The MSE value corresponding to a particular 8×8 block is used to form one element of a corresponding MSE matrix of size 8×8. The entire 8×8 MSE matrix is constructed using the set of MSEs just computed above. The highest MSE in the matrix corresponds to the DCT coefficient that is the most important in a given block, while the lowest MSE corresponds to the DCT coefficient which is the least important in terms of preserving PSNR. In other words, the largest value in the MSE matrix corresponds to the block DCT coefficient that, if removed, would impact the PSNR most significantly.

We can develop, based on the MSE matrix, a new matrix which represents the ranking of importance of each coefficient in a block. The ranking procedure is performed by sorting the elements of a MSE matrix in descending order. The highest MSE in a MSE matrix is assigned to the ranking '1' while the lowest MSE corresponds to ranking '63'. The DCT coefficients corresponding to high ranking (high MSE) should be retained accurately after qunatization while those having low ranking become candidates to be quantized coarsely or removed. The ranking matrix results in a quantization table constructed so as to maximize block PSNR. Figure 3.1 below shows a block diagram which illustrates the *ICS-QT* method.

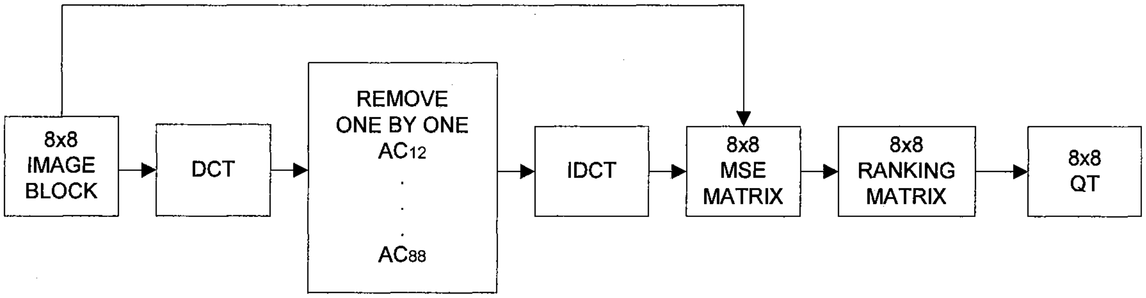


Figure 3.1 Block Diagram for Generating ICS-QT

Orthogonal Property of DCT : Orthogonality is a fundamental property for representing a data vector in terms of a transform basis function [Clarke, 1985]. Suppose we have a set of eight different cosine waveforms of uniform amplitude as shown in Figure 3.2, each sampled at eight points.

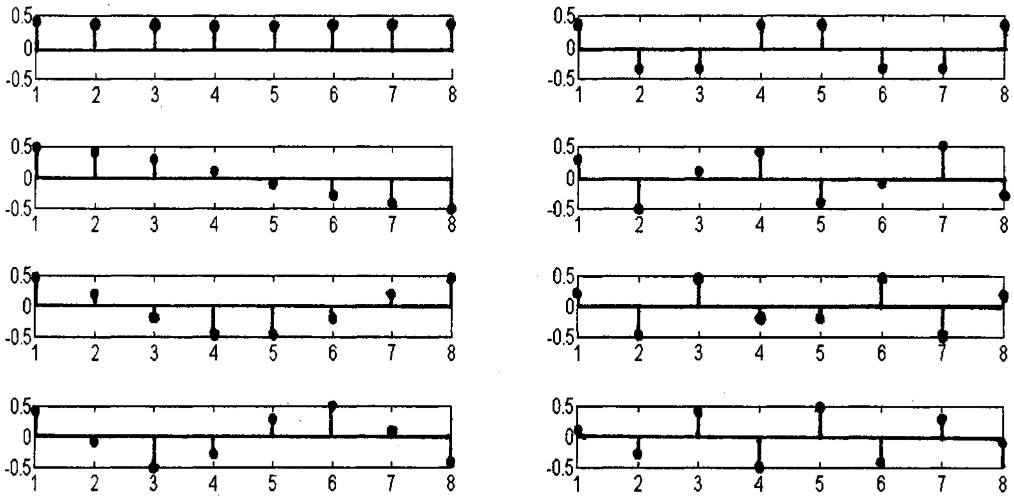


Figure 3.2 DCT Basis Vectors ; N=8 [Clarke, 1985]

If we select any two waveforms in Figure 3.2 and take the product of corresponding sampling points, the sum of these products over all sampling points results in zero. On the other hand, if any waveform is multiplied by itself, the resulting sum is constant. These are defined as orthogonal properties of a waveform, here cosine waveform. In addition, the orthonormal property results in the same total energy in both data and coefficient domains. This property is expressed as in Eq. 3.2 with mathematical notations.

$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|x_{ij}|^2 = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}|X_{uv}|^2 \qquad (3.2)$$

where x_{ij} : picture elements in data domain

X_{uv} : coefficients in DCT domain

Using the property of Eq. 3.2, the analysis by synthesis technique for generating ICS-

QT can be greately simplified. When we remove one of the X_{uv}, the resulting error is exactly $|X_{uv}|^2$. Therefore, we do not need to take IDCT to calculate the resulting MSE when a coefficient is removed. As confirmation, an example is shown in Figure 3.3. As can be seen from Figure 3.3, both the analysis by synthesis technique and the method using the orthonormal property produce the same MSE ranking matrix.

Example Block

56	62	48	40	45	63	57	45
45	51	41	55	56	57	41	39
45	41	45	69	53	39	23	22
45	45	65	70	40	23	16	12
53	64	74	57	33	24	23	12
50	72	69	46	31	26	28	15
41	69	60	41	36	26	26	21
35	61	50	43	46	40	30	23

Ranking Matrix Constructed by Analysis by Synthesis Technique

1	2	4	16	13	35	22	23
8	5	4	10	15	20	27	21
14	3	11	9	18	12	29	39
17	19	6	34	7	49	63	25
38	36	47	42	30	28	33	41
24	31	46	26	50	37	57	45
52	59	60	32	58	40	63	51
53	44	43	48	55	56	61	62

Ranking Matrix Constructed by Orthonormal Property of DCT

1	2	4	16	13	35	22	23
8	5	4	10	15	20	27	21
14	3	11	9	18	12	29	39
17	19	6	34	7	49	63	25
38	36	47	42	30	28	33	41
24	31	46	26	50	37	57	45
52	59	60	32	58	40	63	51
53	44	43	48	55	56	61	62

Figure 3.3 An Example of Ranking Matrix Constructed by Two Different Approaches

In order to generate the quantization table for our tests, we used a set of seven 576x720 gray scale images (a total of 45,360 8x8 blocks) for training. Four different central tendency measures, arithmetic mean, geometric mean, median, and mode

[Steel, 1980] were tested to evaluate which measure gave the best result. In this simulation two 256x256 gray scale images, Lenna and four different texture images (fourtex), were used. Table 3.1 presents a set of simple simulation results which illustrate that for this test data the arithmetic mean produces the best quantization result. As a result of these tests, the algorithm which is discussed in this chapter employs the arithmetic mean to merge the 45,360 8x8 ranking matrixes for constructing the quantization table.

Figures 3.4 and 3.5 show generated ICS-QT and graphically the flow of the ICS algorithm, respectively. The first element of ICS-QT is 'N/A'. This element does not affect to total bit rate count because all of the DC coefficients will be zero as a result of the mean removal strategy described in the following section.

(Set point:0.75 bpp)

Centroids	PSNR (dB)	
	Lenna	Fourtex
Arithmetic Mean	36.916	31.527
Geometric Mean	36.412	31.127
Median	36.195	30.977
Mode	35.068	30.151

Table 3.1 Comparison of Averaging Methods for QT

N/A	10	16	20	23	24	25	30
11	17	21	24	26	27	28	32
18	21	24	27	28	29	29	33
23	27	30	31	32	32	31	35
29	32	34	36	35	35	34	37
34	37	39	40	39	37	35	38
38	41	42	41	39	37	39	39
41	44	44	44	42	40	38	39

Figure 3.4 ICS-QT
(N/A : Not Applicable)

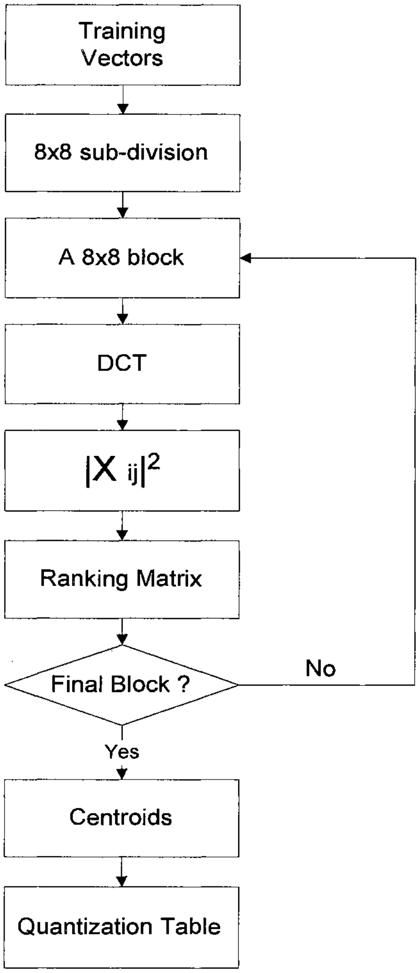


Figure 3.5 Flow Chart of ICS Method Using Orthonormal Property of DCT ($|X_{ij}|^2$: Square of AC coefficients)

3.1.2 Mean Removed Quantization

In the DCT transformed domain, a given DC coefficient has the value of block size times the corresponding block mean. Therefore, the possible magnitude range of the DC coefficient ranges from 0 to 2,048 for an 8-bit image when the block size is 8x8. As a result, even though *DPCM* [Jain, 1989] may be used to code the DC coefficients, sometimes 8 bits or even 9 bits may not be enough to represent accurately the DC values.

If the mean of each block is removed, the statistical expected value of each block is zero and the maximum block mean in the spatial domain is less than or equal to 255 for 8-bit images, which is much smaller than the magnitude of the average DC coefficient in the transform domain. Therefore we can achieve more compression by removing block means; the values of the DC coefficients will all be zero while the activities among adjacent AC coefficients in the DCT domain will remain the same.

The scalar random variable, mean, and the mean removed residual (R), are defined as follows [Gersho, 1991]:

$$m_{lm} = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} X_{ij} \,, \tag{3.3}$$

where m_{lm} corresponds to the mean of the (l,m)th block.

$$\widetilde{R}_{lm} = \widetilde{X}_{lm} - \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} X_{ij} = \widetilde{X}_{lm} - m_{lm} \tag{3.4}$$

$$\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} R_{ij} = 0 \tag{3.5}$$

where M,N : block size, $\widetilde{X}_{lm}$: (l,m)th image block, $\widetilde{R}_{lm}$: (l,m)th residual block, and X_{ij} : each pixel of (l,m)th block.

Therefore, we can decompose the original vector into two separate features, the scalar mean and residual vector, as follow:

$$\widetilde{X}_{lm} = m_{lm} + \widetilde{R}_{lm} \tag{3.6}$$

The DCT transformed residual vectors are quantized using the quantization table which was generated using the ICS method described in the previous section. The block means are transmitted to the decoder using DPCM in the *spatial domain*. Finally, the appropriate mean values are added to the corresponding reconstructed residual vectors. Figure 3.6 below illustrates by means of a block diagram the Mean Removed - Important Coefficient Selection - JPEG (MR-ICS-JPEG) procedure.

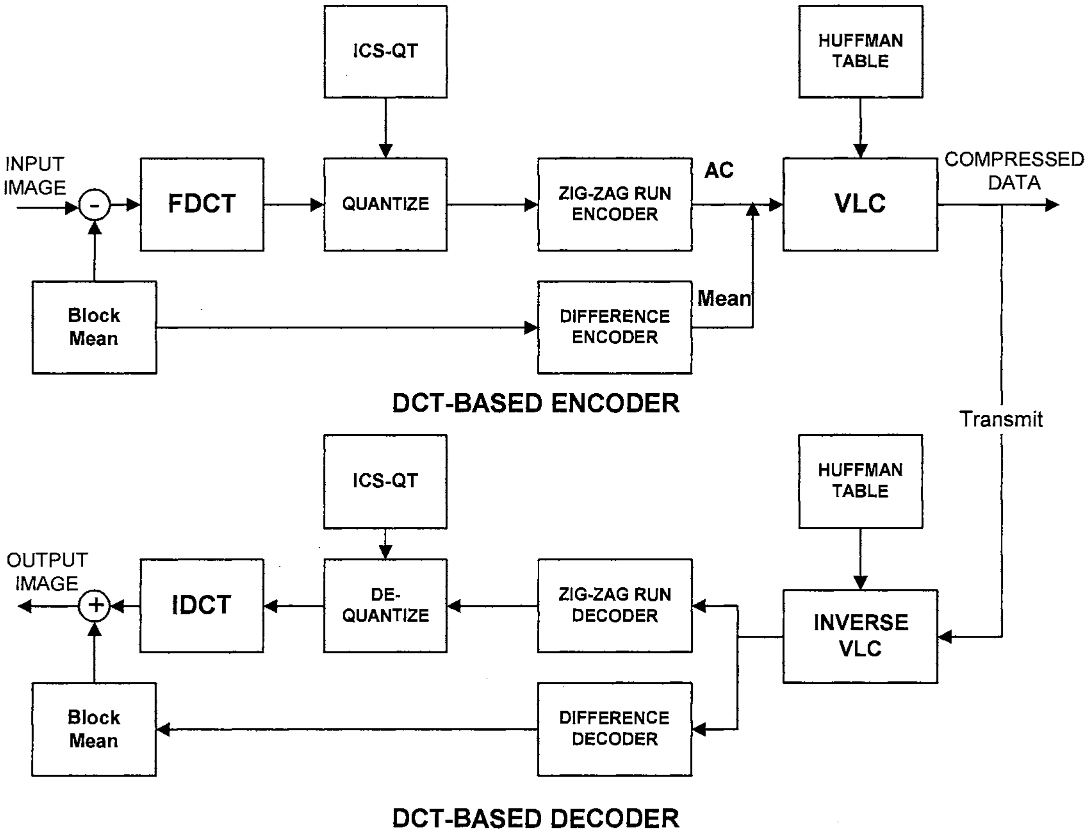


Figure 3.6 Block Diagram of MR-ICS-JPEG

Figure 3.7 illustrates by example the quantization method described above. Figure 3.7.2 shows the mean removed block. Figure 3.7.5 represents the quantized block, where the scaling factor of 1 is chosen arbitrarily for use in this example. The choice of scaling factor is discussed in more detail in a later section. The algorithm can be summarized as follows:

Original Block(mean = 43.4219)

56	62	48	40	45	63	57	45
45	51	41	55	56	57	41	39
45	41	45	69	53	39	23	22
45	45	65	70	40	23	16	12
53	64	74	57	33	24	23	12
50	72	69	46	31	26	28	15
41	69	60	41	36	26	26	21
35	61	50	43	46	40	30	23

Mean -Removed Block

12.58	18.58	4.58	-3.42	1.58	19.58	13.58	1.58
1.58	7.58	-2.42	11.58	12.58	13.58	-2.42	-4.42
1.58	-2.42	1.58	25.58	9.58	-4.42	-20.42	-21.42
1.58	1.58	21.58	26.58	-3.42	-20.42	-27.42	-31.42
9.58	20.58	30.58	13.58	-10.42	-19.42	-20.42	-31.42
6.58	28.58	25.58	2.58	-12.42	-17.42	-15.42	-28.42
-2.42	25.58	16.58	-2.42	-7.42	-17.42	-17.42	-22.42
-8.42	17.58	6.58	-0.42	2.58	-3.42	-13.42	-20.42

Quantization Table

N/A	10	16	20	23	24	25	30
11	17	21	24	26	27	28	32
18	21	24	27	28	29	29	33
23	27	30	31	32	32	31	35
29	32	34	36	35	35	34	37
34	37	39	40	39	37	35	38
38	41	42	41	39	37	39	39
41	44	44	44	42	40	38	39

MR-DCT Block

0	81.79	-35.77	-15.68	-17.38	-3.33	-6.97	-6.88
23.99	-32.11	-0.60	21.42	16.06	8.81	5.87	-8.39
16.47	-39.19	20.46	23.63	-12.47	-18.70	-5.14	-2.57
13.05	11.86	26.93	-3.68	-24.11	-1.20	-0.25	5.88
2.63	3.28	1.46	-2.33	-5.13	5.14	3.86	2.37
-6.15	5.07	1.48	-5.88	1.16	2.98	0.50	1.79
0.89	0.37	0.36	4.23	-0.38	2.39	-0.22	-0.92
0.81	1.94	2.18	1.26	0.53	-0.52	-0.34	0.31

Quantized Block(scale:1)

0	8	-2	-1	-1	0	0	0
2	-2	0	1	1	0	0	0
1	-2	1	1	0	-1	0	0
1	0	1	0	-1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Dequantized Block

0	80	-32	-20	-23	0	0	0
22	-34	0	24	26	0	0	0
18	-42	24	27	0	-29	0	0
23	0	30	0	-32	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Reonstructed Block

56	61	42	34	56	66	57	57
49	48	44	50	58	50	39	40
44	38	52	66	53	30	20	19
44	45	66	69	40	22	18	8
48	63	78	58	25	25	29	9
49	73	78	46	22	33	37	16
43	67	66	41	32	36	31	22
37	57	51	53	41	44	35	24

Difference Block

0	1	6	6	11	3	0	12
4	3	3	5	2	7	2	1
1	3	7	3	0	9	3	3
1	0	1	1	0	1	2	4
5	1	4	1	8	1	6	3
1	1	9	0	9	7	9	1
2	2	6	0	4	10	5	1
2	4	3	2	2	5	8	1

Bit Stream (AC Coefficients)

1	0	1	1	1	0	0	0	0	1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	0
0	1	1	0	1	1	0	1	0	0	1	1	1	1	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	1	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0		

(From left to right and from top to bottom : Figure 3.7.1 ~ Figure 3.7.9)

Figure 3.7 An Example of Quantization Using the Proposed Technique

step 1) Mean removal and DCT computation; remove the mean from every block and compute the 8x8 DCT.

step 2) Quantization, and transmission of block means and quantized AC coefficients; quantized blocks = round (mean removed DCT blocks / (quantization table*1))

step 3) Dequantization; dequantized blocks = quantized blocks*(quantization table*1)

step 4) Reconstruction; reconstructed blocks = IDCT(dequantized blocks) + corresponding quantized block means

The single-block example shown above produces a PSNR of 34.68 (dB). Figure 3.7.9 shows the *Huffman* coded bit stream of the quantized AC coefficients. Seventy three bits are needed to encode the AC coefficients. If DPCM difference of the DC coefficient is three, the DC coefficient is coded by 01111. As a result, the total bits required is seventy eight bits, compression ratio is 6.56, and bit rate is 1.21 (bpp).

3.2 Calibration of Bit Rate

One of the problems associated with testing and applying compression schemes such as JPEG is that it is difficult to produce a constant desired output bit for comparison with other image compression schemes [Gray, 1984] because the variance of AC coefficient in each block and the number of bits required to represent the block are data dependent. However, the calibration of bit rate is highly desirable because it allows one to compare one compression scheme to another and to predict with higher accuracy how much storage space or transmission bandwidth is going to be required to store or transmit a compressed image.

Several papers have been published which have covered this topic [Wu, 1994; Supangkat, 1995]. These algorithms have used an iterative procedure in order to let them converge to a fixed bit rate. However these algorithms depend highly on the initial scaling factor for the quantization table. If one selects a "bad" initial scaling factor, many iterations may be required for the algorithm to converge to the desired bit rate. In the following sections we demonstrate a procedure using regression equations based on the *Proportional and Integral (PI)* control algorithm to properly choose scaling factors so as to achieve a desired bit rate within a reletively small number of iterations [Auslander, 1990; Steel, 1980; Nakagawa, 1992; Sasaki, 1992].

3.2.1 Proportional and Integral (PI) Control

This classical control algorithm produces a compensating value for the scaling factors using bit error [Auslander, 1990]. The following equations define the PI control algorithm.

$$\Delta S = K_p E + K_i \int E dt \tag{3.7}$$

where ΔS : adjusting value of the scaling factor,

K_p: Proportional gain, and

K_i: Integral gain.

For computer simulation, the above integral term is converted into discrete form as follow:

$$\Delta S = K_p E + K_i \sum E \tag{3.8}$$

where $\sum E$: accumulation of errors.

A sufficiently small integral gain must be chosen so as to prevent unstable behavior such as oscillation (failure to converge to minimum error). If it is too small, however, the convergence may be slow. In this experiment, 0.0001 is determined empirically as an integral gain. Figure 3.8 is the block diagram of the PI control algorithm

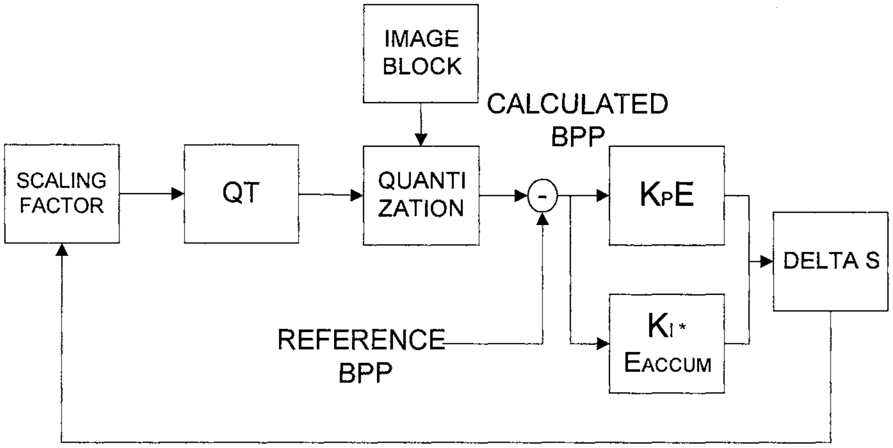


Figure 3.8 Block Diagram of the PI control

3.2.2 Initial Scaling factor

The optimal selection of initial scaling factor reduces the number of iterations required to establish a desired bit rate. In this experiment, the relationship between standard deviations and scaling factors of 1,260 48×48 gray scale images is exploited. In order to perform this experiment, each image is quantized using the quantization table which was generated in Section 3.1.1, and the bit rate is controlled by the PI control algorithm.

Although the overall trend of the scaling factor is increasing when the standard deviation is increasing, the actual variations of the scaling factors are roughly random. In order to reduce their randomness and to find the general representative scaling factors for given bit rates, the similar scaling factors and the similar corresponding standard deviations of 1,260 48x48 images are merged together using the arithmetic mean when they converge to fixed bit rates.

In this experiment, five different set points (data rate, given as bits per pixel - bpp) are considered (0.25, 0.5, 0.75, 1 and 2.25 bpp) to calculate the linear regression equations. In order to calculate these equations , the least square regression method, which minimizes the average distance between fitted lines and actual data, is applied [Steel, 1980]. After developing the five linear regression equations to approximate the initial scaling factors, the optimal proportional gains, *Kp*, are determined by further simulations with the five linear regression equations. That is, scaling factors for σ to determine *Kp* are varied from 0.001 to 10 and the optimal scaling factors for σ, which gives the smallest number of iterations for convergence are determined for 1,260 48x48 gray scale images. Finally these scaling factors for each set point (bpp) are merged together using the arithmetic mean. Table 3.2 represents the optimal regression equations, based on a 95% confidence interval [Steel,1980], and their corresponding proportional gains. Using this technique, no more than about four iterations are required to converge to a specified constant bit rate. Figure 3.9 shows an example of linear regression line in the case of bit rate 0.5 bpp.

Set Point(BPP)	Linear Regression Equations	Kp	Ki
0.25	$S=0.261844*\sigma - 0.380124$	$0.64308*\sigma$	0.0001
0.5	$S=0.067992*\sigma + 0.054204$	$0.13846*\sigma$	0.0001
0.75	$S=0.036687*\sigma + 0.015799$	$0.05733*\sigma$	0.0001
1.00	$S=0.020517*\sigma + 0.053998$	$0.02963*\sigma$	0.0001
2.25	$S=0.004259*\sigma + 0.227768$	$0.00552*\sigma$	0.0001

(σ : Average Block Standard Deviation of Image in Spatial Domain, S: Initial Scaling Factors, and Kp, Ki: Proportional and Integral Gain for PI control)

Table 3.2 Linear Regression Equations and Their Corresponding Optimal Gains

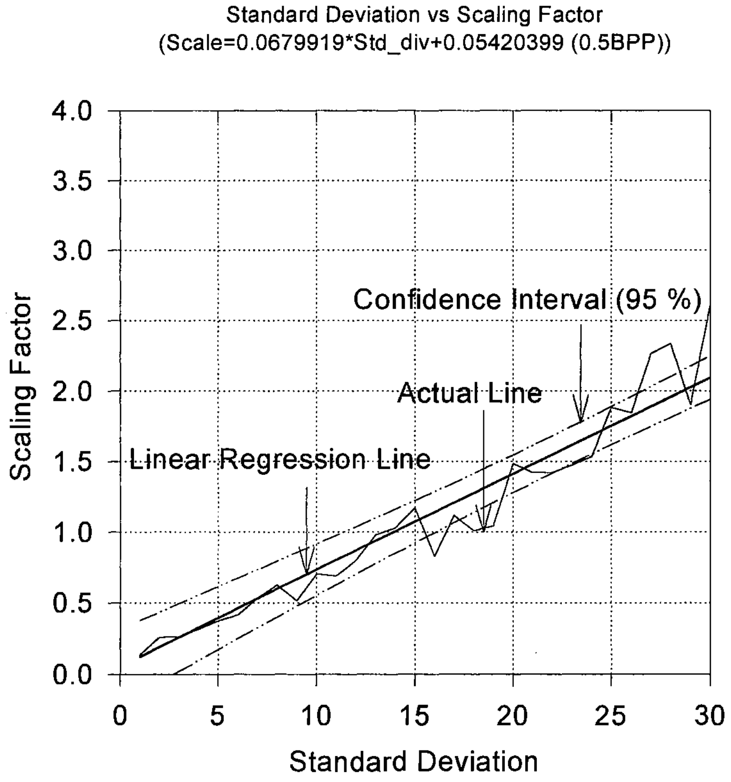


Figure 3.9 Linear Regression Equation (0.5 bpp)

3.3 Simulation Results and Discussion

To test the performance of the new quantization table presented earlier, several simulations have been performed and compared to the performance of a reference coder (chosen to be JPEG for this study). The mean removed scheme is combined with both ICS-QT and the reference quantization table technique of JPEG. In these simulations two standard gray scale images, Lenna and Boat, are used for the comparison of performance. Performance is measured in terms of PSNR.

Figures 3.10 illustrates the performance of the three different methods, Standard-JPEG, MR-JPEG, and MR-ICS JPEG at a variety of bit rates. The MR-ICS-JPEG and the MR-JPEG give more than 2dB and 0.5 dB improved performance when referenced to standard JPEG, respectively. When the MR scheme is combined with the reference quantization table of standard JPEG, the performance improvement is just 0.5 dB. When it is combined with ICS-QT, the performance improvement is in excess of 2dB. So we can conclude that an improved quantization table (ICS-QT) plays a more important role than the MR-scheme for elevating performance.

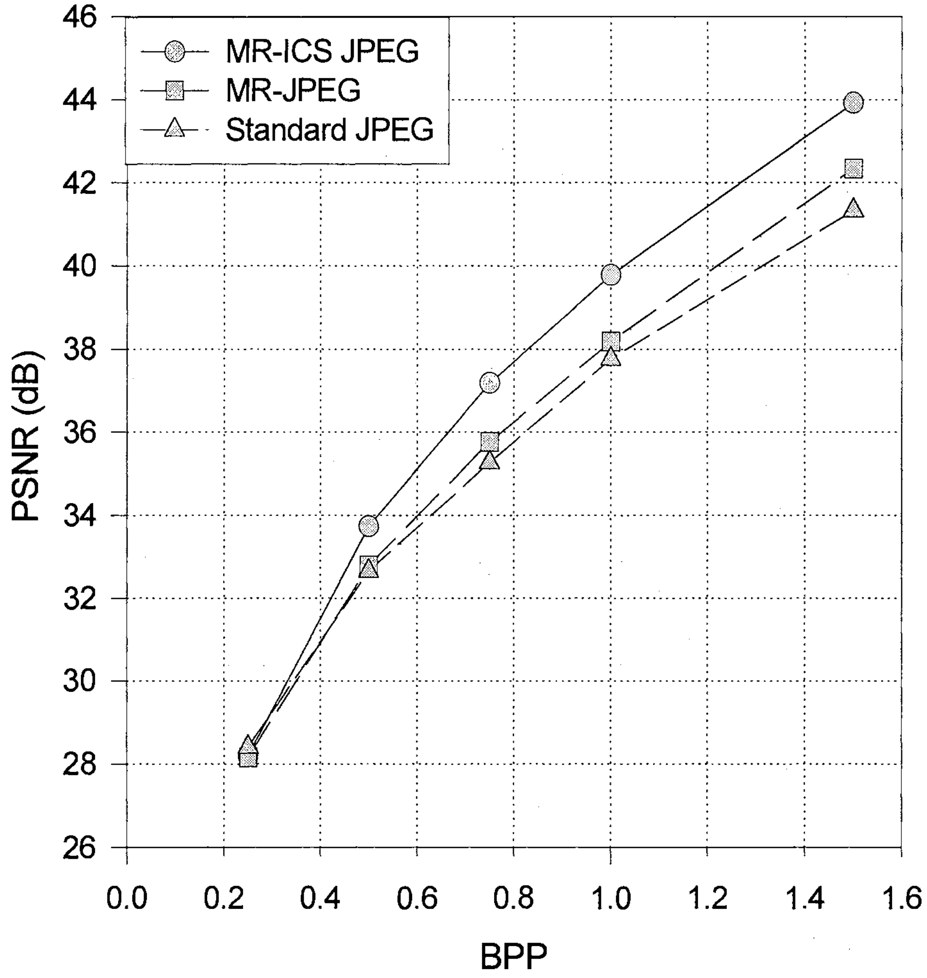
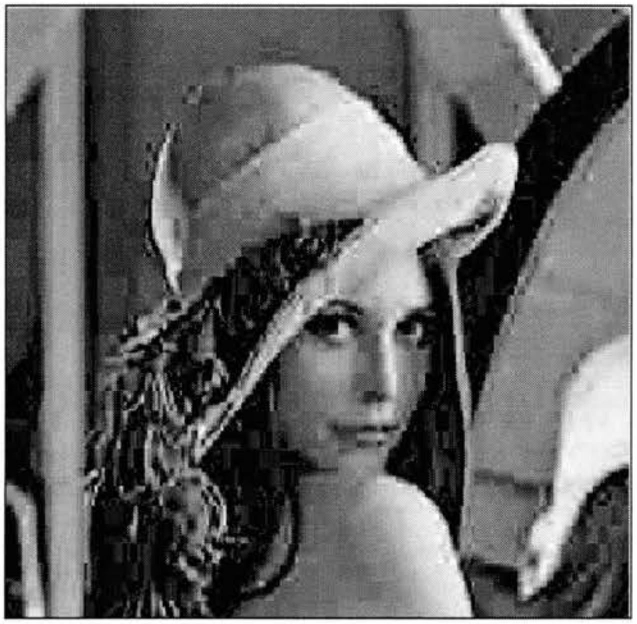


Figure 3.10 Performances of the Three Different Schemes (Lenna)

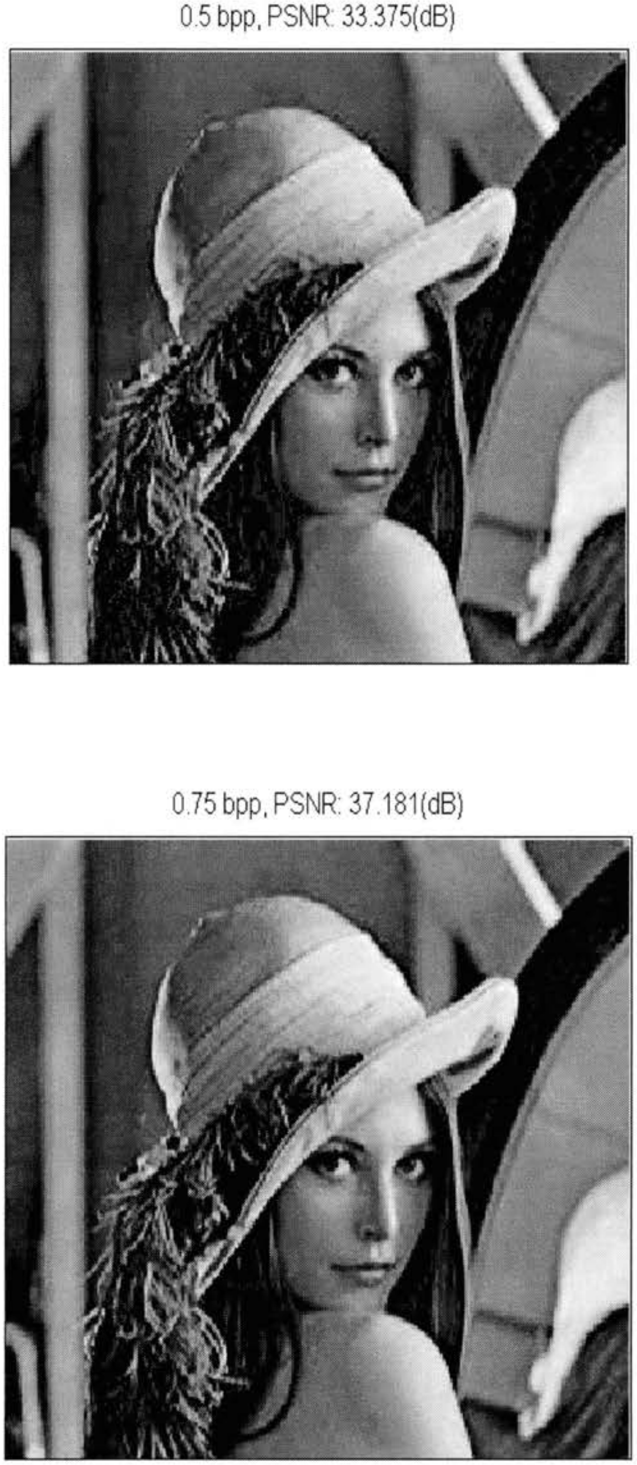
In the range of lower compression, the performance improvement is more than 3dB because the quantized DC coefficients in standard JPEG scheme are much larger than the maximum mean value of the spatial domain. On the other hand, in the range of high compression, standard JPEG produces almost the same performance as that of the MR schemes because the multiplication of large scaling factors to the quantization table in standard JPEG scheme makes the magnitudes of DC coefficients close to or smaller than the corresponding block means of the spatial domain. In addition, an additional 4 bits is neceaasry to indicate end of blocks (EOB), so just remaining few bits are allocated to code actual quantized coefficients. Therefore, the performance will not be improved no matter what kinds of quantization tables we are using at low bit rates. Figures 3.11 and 3.12 show examples of the actual compressed images (Lenna and Boat) for the proposed MR-ICS-JPEG at fixed bit rates.

In order to validate the performance of the five linear regression equations, additional simulations are perfromed for fifty 256×256 gray scale images for five bit rates. In this simulation, this procedure has never been observed to require more than four iterations for convergence to a specified bit rate. The average number of iterations for each bit rate is 2.82, 1.92, 1.86, 1.82, and 2 for the bit rates of 0.25, 0.5, 0.75, 1.0, and 2.25, respectively. The total number of iterations required for the fifty images, each has five set points, is 521. Therefore, the global average number of iterations to converge to a fixed bit rate is 2.084. Figures 3.13 through 3.17 shows the number of iterations for convergence to fixed bit rates for the fifty test images.

Lenna(original)



0.25 bpp, PSNR: 28.182(dB)

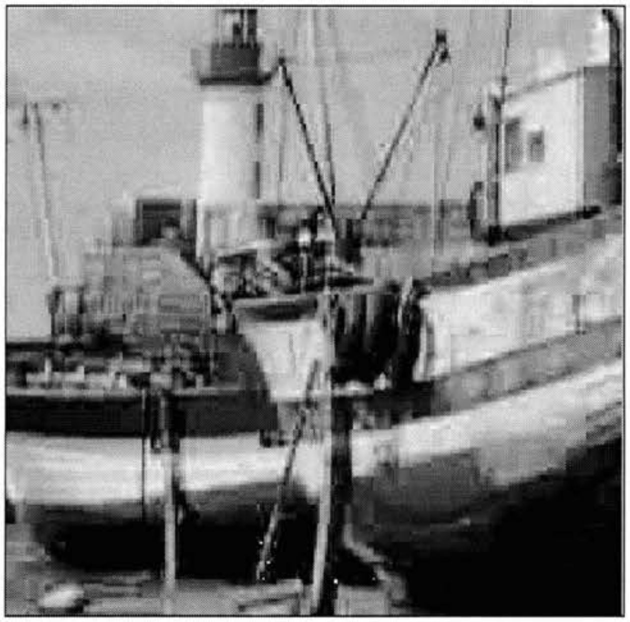


0.5 bpp, PSNR: 33.375(dB)

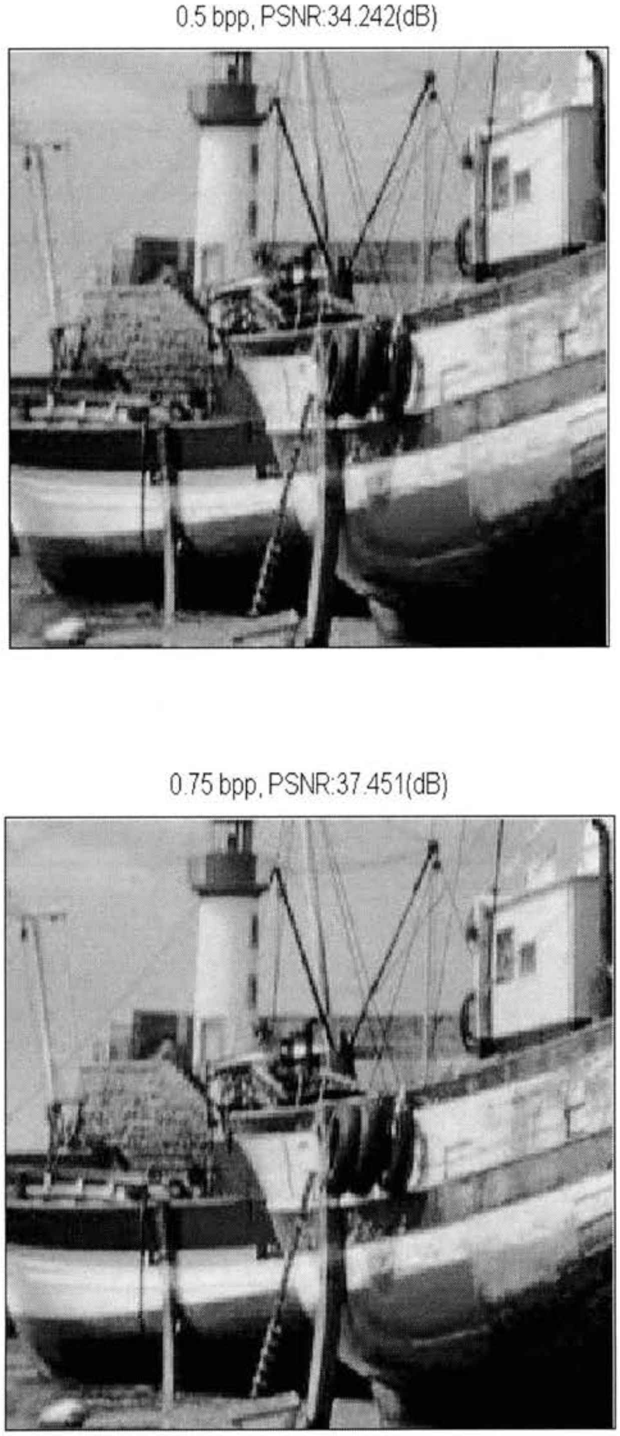
0.75 bpp, PSNR: 37.181(dB)

Figure 3.11 Original and Compressed Images (Lenna)

Boat (original)



0.25 bpp, PSNR:29.448(dB)



0.5 bpp, PSNR:34.242(dB)

0.75 bpp, PSNR:37.451(dB)

Figure 3.12 Original and Compressed Images (Boat)

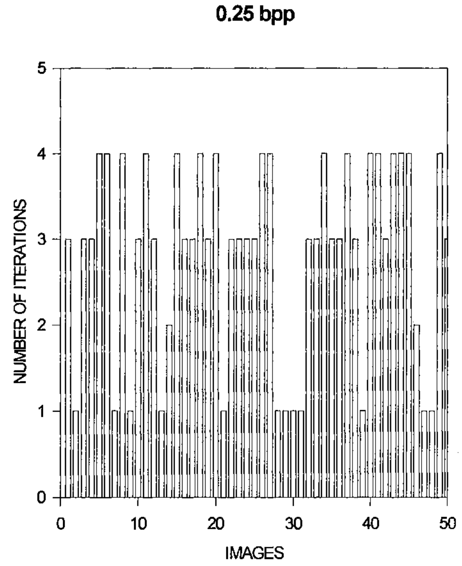


Figure 3.13 Number of Iterations
Max: 4, Min: 1, Average :2.82

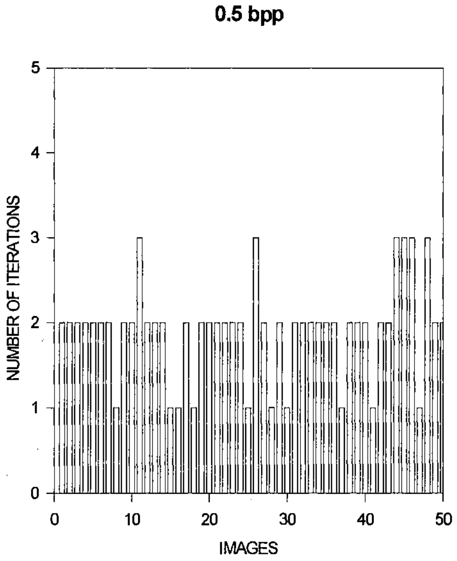


Figure 3.14 Number of Iterations
Max: 3, Min: 1, Average :1.92

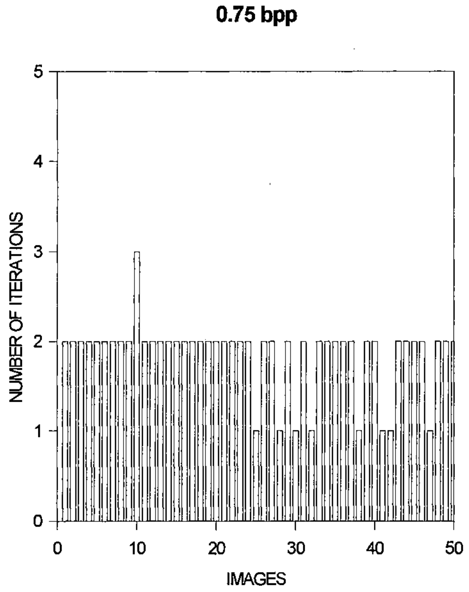


Figure 3.15 Number of Iterations
Max: 3, Min: 1, Average: 1.86

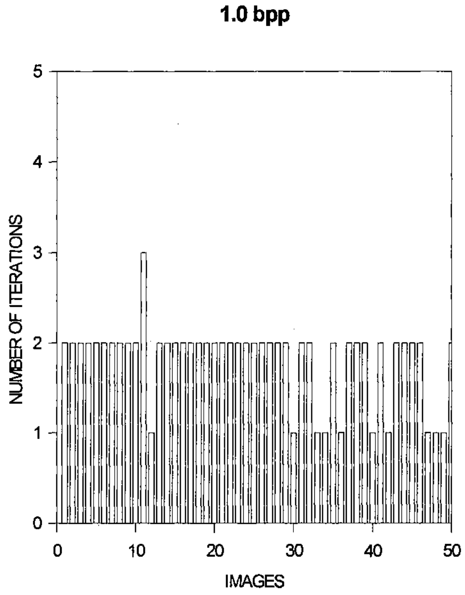


Figure 3.16 Number of Iterations
Max: 3, Min : 1, Average : 1.82

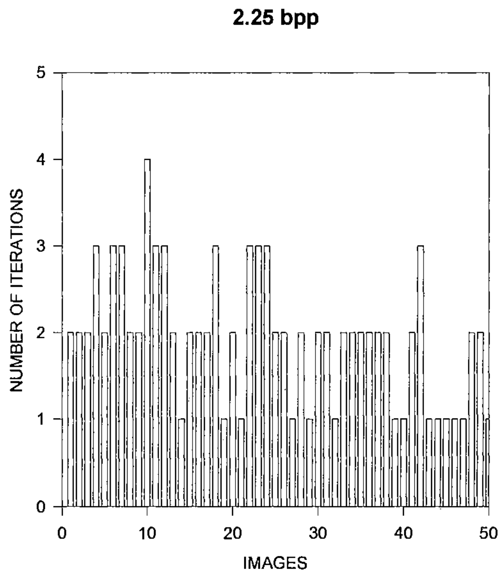


Figure 3.17 Number of Iterations
Max : 4, Min : 1, Average : 2

3.4 Conclusion

The new quantization table using ICS method is introduced and the MR scheme is applied to both the ICS-QT and standard JPEG. The MR-ICS-JPEG and the MR-JPEG produce more than 2dB and 0.5 dB improved performances when compared to that of standard JPEG, respectively. The five different linear regression equations which are adjusted with ICS-QT are presented. These equations will not fit well when we employ other quantization tables; however, there is value in combining ICS-QT and its corresponding linear regression equations because ICS-QT gives enhanced performance, in terms of PSNR, when compared to standard JPEG. Using the PI control with the linear regression equations, the desired constant bit rates were produced no more than about four iterations.

CHAPTER IV

VECTOR QUANTIZATION AND ITS HIERARCHICAL IMPLEMENTATION USING QUADTREE

Hierarchical data structures are important representations in data compression problems because of their ability to focus on the interesting subsets of data. They are based on the principle of recursive decomposition. The decomposition practically results in an image segmentation, which is a very useful technique for a variety of image processing problems such as pattern recognition [Chien, 1984].

One of the most commonly used methods to implement a hierarchical data structure is *quadtree segmentation (QTS)* [4.1]. This chapter is devoted to develop optimal thresholding methods for efficient quadtree segmentation. The choice of threshold to decompose an image affects the total compression performance. In other words, low compression ratio and low image quality are resulted if the selected threshold is too low and too high, respectively, in the case of top down construction of quadtree. The second half of this chapter describes *homogeneity* testing methods of quadtree segmented blocks.

4.1 Quadtree Segmentation for Hierarchical Implementation

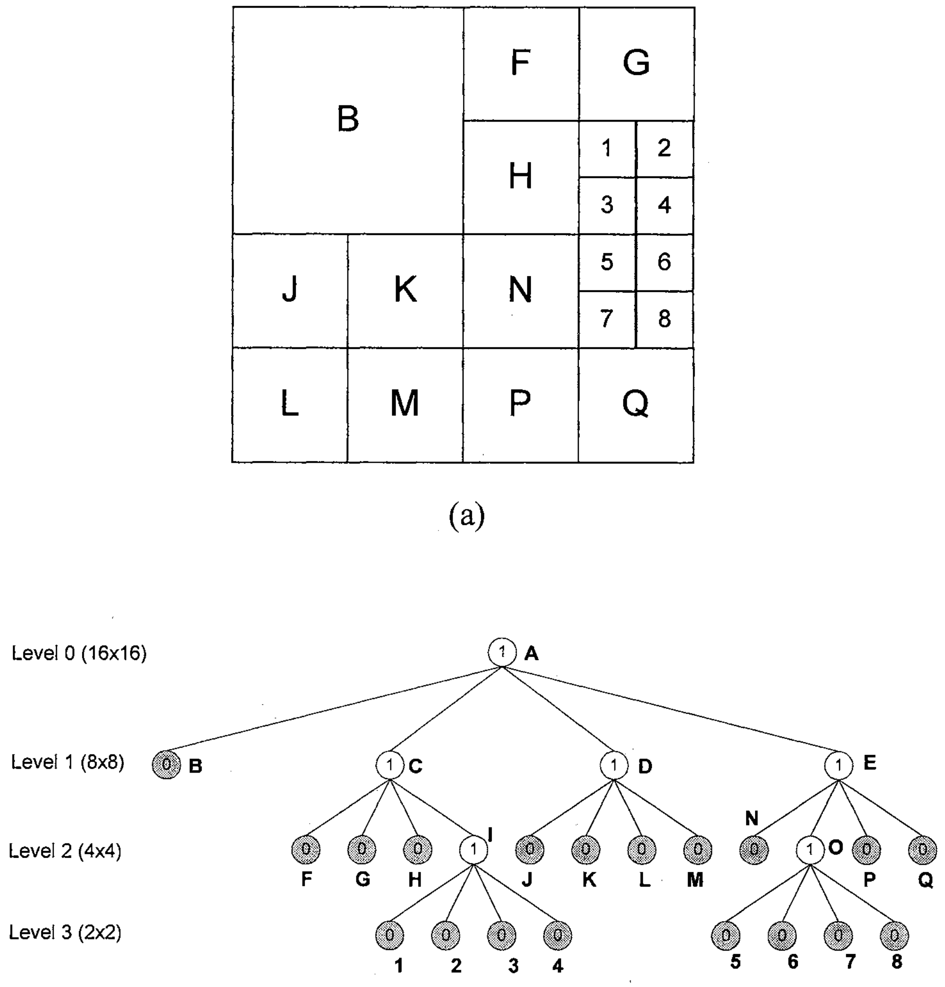
The basic idea of QTS is consecutive subdivision of an image into its

[4.1] [Grosky,1983; Jackin, 1983; Samet, 1984; Shaffer, 1987; Strobach, 1989; Shusterman, 1994] homogeneous quadrants whose sides are all a power of two long if the image is square. This strategy is continued recursively until the activity of every block is less than or equal to predefined thresholds. When we describe a block, if its intensity is relatively constant, it is coded with a set of large dimensional codebooks which have low quantization levels. If a block is relatively active, its standard deviation is large, it is segmented into four smaller blocks, and these small blocks are coded with a set of small dimensional codebooks which has high quantization levels.

In the quadtree representation of an image, a *nonleaf node* has four children nodes which correspond to subblocks of the image. A *leaf node* is defined as a block which has no necessary further subdivision. The root node represents an entire image, and the first level involves four disjointed subblocks. This segmentation procedure is called the *top-down* construction of a quadtree.

Figure 4.1 describes an example of the top-down QTS, its quadtree structure, and corresponding binary map. In the quadtree representation of Figure 4.1 (b), white and shaded nodes represent nonleaf and leaf nodes, respectively. The white circles are coded by a binary value of '*1*' while the shaded circles are represented by a binary value of '*0*' in the order of top to bottom and left to right. With this strategy, the binary bitmap of Figure 4.1 (c) is produced. An attractive point of the quadtree representation is its low overhead in number of bits, where overhead means the number of bits required to describe the structure of the quadtree. Each node requires just one binary digit to specify whether a given node needs to be split or will be a terminal node which does not require further subdivision. For example, a 16×16 block of Figure 4.1 (a) needs only (17/256 = 0.0664 bpp).

In order to represent overhead bits in a general form of the top-down quadtree, an *empirical probability* [Shusterman, 1994] of finding leaf nodes at level k is defined as follows:



(a)

(b)

Code <u>1</u> <u>0 1 1 1</u> <u>0 0 0 1</u> <u>0 0 0 0</u> <u>0 1 0 0</u> (17 bits)

Level 0 1 2 2 2

(c)

Figure 4.1 Typical Quadtree Representation of a block (16x16)

(a) a segmented image block, (b) quadtree representation, and (c)bitmap

$$P_k = \frac{N_k}{4^k} \tag{4.1}$$

where N_k : number of nodes that do not have children at level k and

k : number of level.

Cumulative probability of finding leaf nodes at level k is defined as

$$q_k = \begin{cases} q_{k+1} - p_{k+1} & k = 0,\ldots,n-1 \\ 1 & k = n \end{cases} \tag{4.2}$$

The number of leaves included in a quadtree and the number of bits to represent a quadtree are defined as Eq. 4.3 and Eq. 4.4, respectively.

$$N_{L(qt)} = \sum_{k=0}^{n} 4^k p_k \tag{4.3}$$

$$N_{B(qt)} = \frac{4^n - 1}{3} - \sum_{k=0}^{n-2} 4^{k+1} q_k \tag{4.4}$$

where $N_{L(qt)}$:number of leaves,

$N_{B(qt)}$:number of bits, and

n : number of levels in a quadtree.

Finally, the bit rate of overhead is calculated by the following equation.

$$BR_{qt} = \frac{N_{B(qt)}}{4^{n+1}} \tag{4.5}$$

As an example, the overhead in number of bits of Figure 4.1 is calculated as in Table 4.1. The calculation of the example below shows overhead of a particular block. Figure 4.3 represents the overhead of a 256×256 image when the level of quadtree varies from 1 to 7, and the minimum allowed block size is 2×2. As can be seen from this graph, the more levels we have, the more overhead necessary. In addition, if we choose more than 16×16 as the largest block, it requires a large amount of storage space for the codebook, and large block areas will suffer from blockiness even though their intensities are relatively constant. Because of this, the 8x8 block (level 2) is selected as the largest block in the hierarchical implementation of this thesis. Figure 4.2 shows the block diagram of HVQ.

$$n = 3$$

$$p_0 = \frac{0}{1} = 0 \qquad q_3 = 1$$

$$p_1 = \frac{1}{4} \qquad q_2 = q_3 - p_3 = 1 - \frac{1}{8} = \frac{7}{8}$$

$$p_2 = \frac{10}{16} = \frac{5}{8} \qquad q_1 = q_2 - p_2 = \frac{7}{8} - \frac{5}{8} = \frac{1}{4}$$

$$p_3 = \frac{8}{64} = \frac{1}{8} \qquad q_0 = q_1 - p_1 = \frac{1}{4} - \frac{1}{4} = 0$$

$$N_{L(qt)} = 4^0 \times 0 + 4^1 \times \frac{1}{4} + 4^2 \times \frac{5}{8} + 4^3 \times \frac{1}{8} = 1 + 10 + 8 = 19$$

$$N_{B(qt)} = \frac{63}{3} - (4 \times 0 + 4^2 \times \frac{1}{4}) = 21 - 4 = 17$$

$$BR_{qt} = \frac{17}{4^4} = 0.0664 \cong 0.07 bpp$$

Table 4.1 An Example of Overhead Calculation in Bits

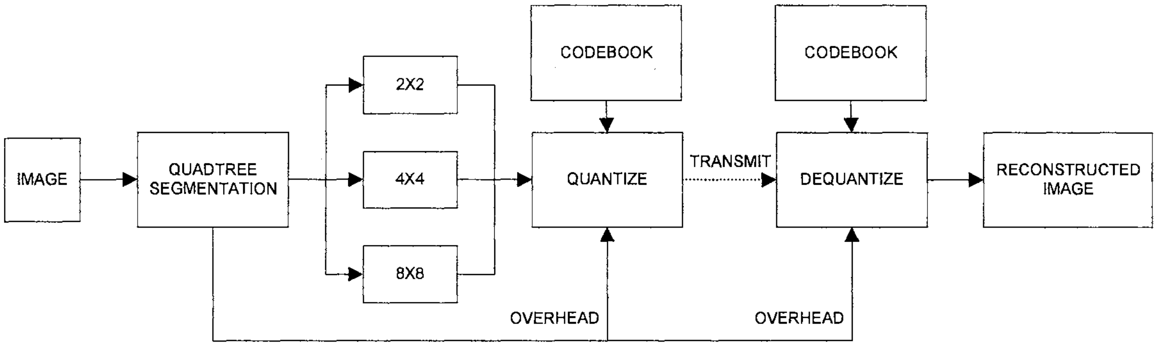


Figure 4.2 Block Diagram of HVQ

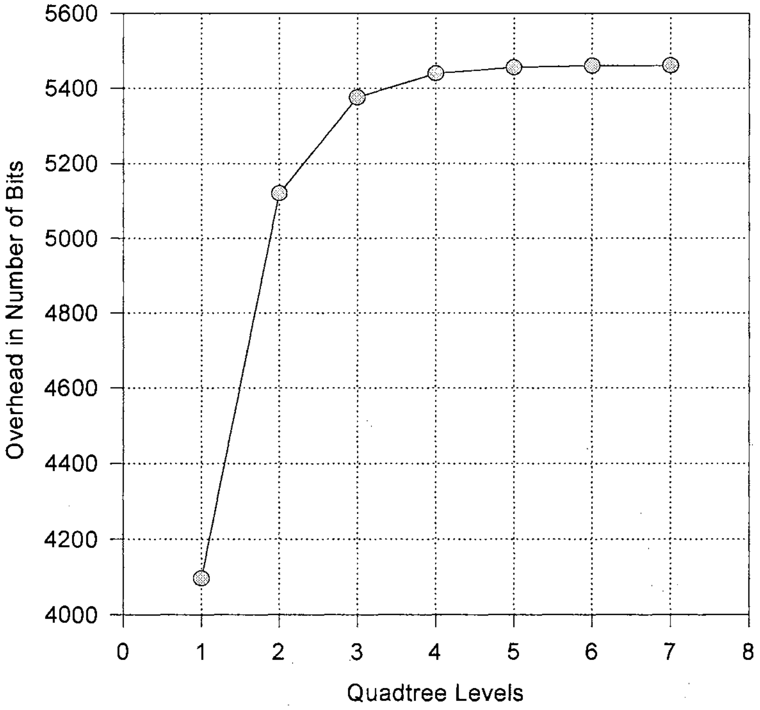


Figure 4.3 The Overhead in Bits of a 256x256 Image.

4.2 Optimal Thresholding

In general, the efficiency of the quadtree decomposition is measured by MSE, bpp, and number of leaves achieved. These parameters depend on the statistics of given images and the decomposition *threshold* of each quadtree level. O*ptimal thresholding* has been considered by many researchers[4.2] ; however, a true optimal solution has not been found. This section proposes methods for finding the optimal threshold and testing the resulting *homogeneity* of segmented image blocks.

[4.2] [Vaisey, 1987; Chiu, 1989; Strobach, 1990; Strobach, 1991; Vaisey, 1992].

4.2.1 Single - Thresholding

Because most images do not have a *Gaussian* distribution, their statistical behavior is unpredictable. Usually the bit rate and the MSE are inversely increasing. Thus, these two quantities need to be traded off. This section discusses a method for finding an optimal decomposition point for trading off these two quantities.

Standard deviation has often been used as a good measure of image property or statistics. In this research, the standard deviation of image blocks is used as a threshold. A single threshold is used for each level of the quadtree to achieve implementation simplicity. Figure 4.4 shows an example which represents the relationship between MSE and bit rate of a common image when the thresholds are varied from *0* to *50*. The inverses of MSE and bpp are normalized by their maximum values to constrain their values from *0* to *1* for comparison purposes. These two quantities and their product are shown on the *y-axis*. In the case of an image having a Gaussian distribution, the cross point of (1/MSE) and (1/bpp) gives a maximum product value, and its corresponding point to the *x-axis* becomes the optimal threshold which produces minimum *bpp* and MSE. Equation 4.6 shows this relationship.

$$Optimal\,point = Max(\frac{1}{MSE_i} \times \frac{1}{bpp_i}) \qquad i=0, \ldots\ldots, 50 \qquad (4.6)$$

However, most images do not have a Gaussian distribution. As a result, the cross point does not give an optimal solution for the threshold. For example, the optimal threshold of Figure 4.4 is six, which does not correspond to the cross point which is four.

The optimal threshold depends highly on the standard deviation of a given image. In order to derive a general relationship between the standard deviations of input images and the optimal thresholds, 1,260 48x48 image blocks are used, and the corresponding 1,260 optimal thresholds are estimated from the graphs. These 1,260 optimal thresholds are merged using the arithmetic mean and plotted with their corresponding standard deviations. A *polynomial* fit to the given data is calculated and displayed in Figure 4.5. When the standard deviation of a given image is less than 40, the threshold decreases according to the increasing of the standard deviation and results in increasing bit rates. However, the bit rate increase does not greatly influence the reduction of MSE when the standard deviation is larger than 40. Therefore, the threshold increases at the standard deviations higher than 40 because this polynomial is constructed by trading off between MSE and bit rate. We can now calculate approximations of the optimal thresholds very easily using this polynomial for the standard deviations of given images.

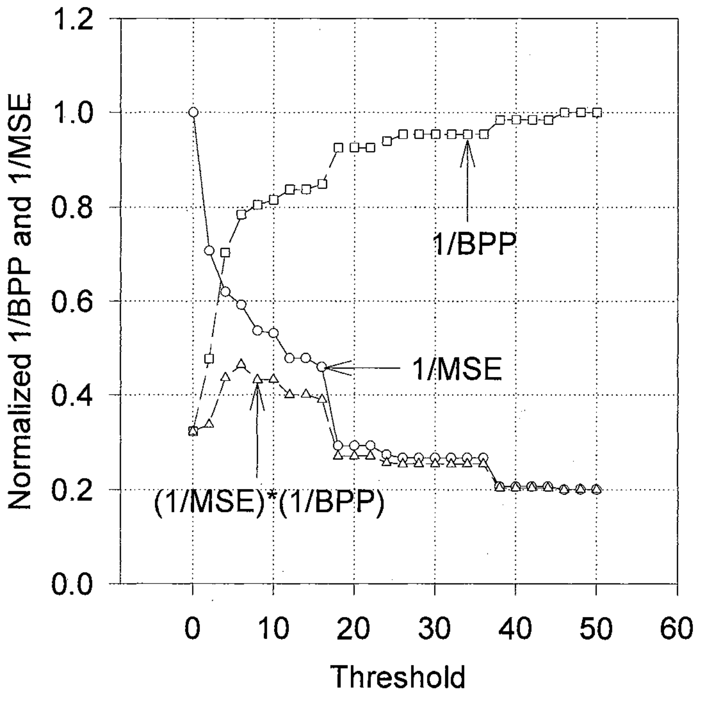


Figure 4.4 An Example of the Relationship Among MSE, BPP, and Threshold

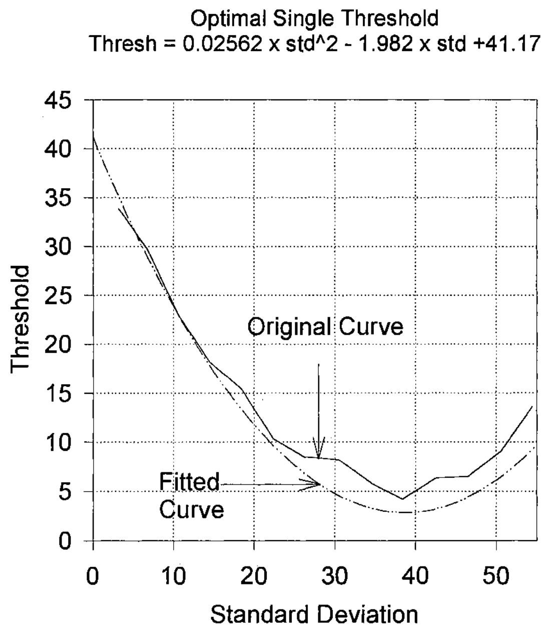


Figure 4.5 An Optimal Polynomial for Finding Optimal Single Thresholds.

4.2.2 Multi - Thresholding

After a super block (8×8) is segmented by a given threshold, the standard deviations of subblocks can be larger, equal to, or smaller than that of their parent block. Therefore, using different thresholds at each quadtree level is sometimes necessary. This section proposes a mathematical model for approximating optimal thresholds of each quadtree level.

Equations (4.1) through (4.5) represent the overhead calculation in the case of the top-down quadtree segmentation method. In order to avoid the complexity for mathematical modeling, we consider *bottom-up quadtree* construction, where the minimum allowed block size used is 2×2. The bottom-up quadtree construction is started by merging 4 leaves if their standard deviations are less than or equal to a predefined threshold, and this process is continued recursively until it meets the root node. As we mentioned before, a standard deviation after merging four leaves can be larger, equal to, or smaller than that of its parent blocks. However, we can construct a *hypothesis* that the standard deviation of a current level is less than or equal to some relationships of the thresholds of previous levels, because our intention is to reduce the total number of blocks in order to reduce the bit rate without sacrificing the image quality. With this hypothesis, the standard deviation of the first level can be bounded as shown in Eq. 4.7

$$STD_1 \le \frac{1}{4} T_0[\gamma_0(1) + \gamma_0(2) + \gamma_0(3) + \gamma_0(4)] \tag{4.7}$$

where $\gamma_0(k) = \dfrac{STD_0(k)}{T_0} \le 1$, *k= 1, 2, 3, and 4.*

The worst case merge results in $\gamma_0(k)$ and STD_1 equal to 1 and T_0, respectively. Similarly, the relationship of the second level is derived by the hypothesis as in Eq. 4.8.

$$\begin{aligned} STD_2 \le & \frac{1}{4} T_1[\gamma_1(1) + \gamma_1(2) + \gamma_1(3) + \gamma_1(4)] \\ & + \frac{1}{16} T_0[\gamma_0(1,1) + \gamma_0(1,2) + \gamma_0(1,3) + \gamma_0(1,4) \\ & \quad + \gamma_0(2,1) + \gamma_0(2,2) + \gamma_0(2,3) + \gamma_0(2,4) \\ & \quad + \gamma_0(3,1) + \gamma_0(3,2) + \gamma_0(3,3) + \gamma_0(3,4) \\ & \quad + \gamma_0(4,1) + \gamma_0(4,2) + \gamma_0(4,3) + \gamma_0(4,4)] \end{aligned} \tag{4.8}$$

If we consider the worst case again, STD_2 equals to $T_1 + T_0$, and the standard deviation of the second level is upper bounded as in Eq. 4.9.

$$STD_2 \le T_1 + T_0 \tag{4.9}$$

$$\vdots$$

$$STD_k \le T_{k-1} + T_{k-2} + T_{k-3} \cdots \quad T_0 = \sum_{m=0}^{k-1} T_m \tag{4.10}$$

where STD_k : standard deviation of the root node.

In the bottom-up construction, Equations (4.1), (4.2), and (4.3) are modified to Equations (4.11), (4.12), and (4.13), respectively.

$$P_k = \frac{N_k}{4^{n-k}} \tag{4.11}$$

$$q_k = \begin{cases} 1 & k = 0 \\ q_{k-1} - p_{k-1} & k = 1,\dots,n \end{cases} \tag{4.12}$$

$$N_{L(qt)} = \sum_{k=0}^{n} 4^{n-k} p_k \tag{4.13}$$

recall p_k: Empirical probability of finding leaf nodes at level *k*,

q_k: Cumulative probability of finding leaf nodes at level *k*, and

$N_{L(qt)}$: Number of leaves.

A general expression of Eq. 4.10 is as follow:

$$\sum_{k=1}^{n} p_k \times [STD_k] \le \sum_{k=1}^{n} p_k \times [\sum_{m=0}^{k-1} T_m]\,. \tag{4.14}$$

From Eq. 4.12, Eq. 4.14 becomes Eq.4.15.

$$\sum_{k=1}^{n} (q_k - q_{k+1}) \times [STD_k] \le \sum_{k=1}^{n} (q_k - q_{k+1}) \times [\sum_{m=0}^{k-1} T_m]\,. \tag{4.15}$$

The right side of Eq. 4.15 can be rewritten as follows:

$$\sum_{k=1}^{n} (q_k - q_{k+1}) \times [\sum_{m=0}^{k-1} T_m]$$

$$= (q_1 - q_2)T_0 + (q_2 - q_3)(T_0 + T_1) + (q_3 - q_4)(T_0 + T_1 + T_2) +$$

$$\dots\dots(q_n - q_{n+1})(T_0 + T_1 + \cdots\ T_{n-1})\,.$$

$$= q_1T_0 - \cancel{q_2T_0} + \cancel{q_2T_0} + q_2T_1 - \cancel{q_3T_0} - \cancel{q_3T_1} + \cancel{q_3T_0} + \cancel{q_3T_1} + q_3T_2$$

$$-\cancel{q_4T_0} - \cancel{q_4T_1} - \cancel{q_4T_2} + \dots\dots - q_{n+1}T_0 \ \cdots\cdots\ - q_{n+1}T_{n-1}\,.$$

$$= q_1T_0 + q_2T_1 + q_3T_2 + \cdots\cdots + q_nT_{n-1} - q_{n+1}T_0 - \cdots - q_{n+1}T_{n-1}\,.$$

$$= \sum_{k=1}^{n} q_k T_{k-1} - q_{n+1} \sum_{m=1}^{n} T_{m-1} \,. \tag{4.16}$$

Because the level *n+1* does not exist, the second term of Eq. 4.16 becomes zero. Combining Eq. 4.15 and Eq. 4.16, we can get Eq. 4.17.

$$STD_k \le \sum_{k=1}^{n} q_k T_{k-1} \tag{4.17}$$

For the bottom-up case, Eq. 4.13 can be rewritten as follows:

$$N_{L(qt)} = \sum_{k=0}^{n} 4^{n-k} p_k = 4^n p_0 + \sum_{k=1}^{n} 4^{n-k} p_k$$

$$= 4^n (1 - q_1) + \sum_{k=1}^{n} 4^{n-k} (q_k - q_{k+1})$$

$$= 4^n (1 - q_1 + \frac{q_1}{4} - \frac{q_2}{4} + \frac{q_2}{16} - \frac{q_3}{16} + \frac{q_3}{64} - \frac{q_4}{64} + \cdots + \frac{q_n}{4^n} - \frac{q_{n+1}}{4^n})$$

Recall, level *n+1* does not exist.

$$\therefore N_{L(qt)} = 4^n (1 - \frac{3q_1}{4} - \frac{3q_2}{16} - \frac{3q_3}{64} - \cdots\cdots - \frac{3q_n}{4^n})$$

$$= 4^n [1 - \frac{3}{4}(q_1 + \frac{q_2}{4} + \cdots\cdots + \frac{q_n}{4^{n-1}})]$$

$$= 4^n (1 - 3\sum_{k=1}^{n} \frac{q_k}{4^k}) = 4^n (1 - \frac{3}{4} q_1 - 3\sum_{k=2}^{n} \frac{q_k}{4^k}) \tag{4.18}$$

From Eq. 4.18, we can get q_1.

$$q_1 = \frac{4}{3}(1 - 4^{-n} N_{L(qt)}) - \sum_{k=2}^{n} 4^{1-k} q_k \tag{4.19}$$

Eq. 4.17 can be rewritten as Eq. 4.20.

$$STD_k \le q_1 T_0 + \sum_{k=2}^{n} q_k T_{k-1} \tag{4.20}$$

Substituting Eq. 4.19 for q_1 in Eq. 4.20, we can get Eq. 4.21.

$$STD_k \le \frac{4}{3}(1 - 4^{-n} N_{L(qt)}) T_0 - \sum_{k=2}^{n} 4^{1-k} q_k T_0 + \sum_{k=2}^{n} q_k T_{k-1}$$

$$= \frac{4}{3}(1 - 4^{-n} N_{L(qt)})T_0 + \sum_{k=2}^{n} q_k (T_{k-1} - 4^{1-k} T_0) . \tag{4.21}$$

In Eq. 4.21, the first term of the right side, $1 - 4^{-n} N_{L(qt)}$, is always positive. In order to minimize the standard deviation of the root node, the second term of Eq. 4.21 should be less than or equal to zero. Thus, we can get the following relationship among thresholds as shown in Eq. 4.22.

$$T_{k-1} - 4^{1-k} T_0 \le 0$$
$$\therefore T_{k-1} \le 4^{1-k} T_0 \tag{4.22}$$

Eq. 4.22 represents the relationship among thresholds of each level in the case of the bottom-up quadtree construction. If we consider the case of the top-down quadtree construction, then Eq. 4.22 should be modified to Eq. 4.23.

$$T_{k-1} \le 4^{k-1} T_0 \tag{4.23}$$

As can be seen from the Eq. 4.23, a threshold of each level is upper bounded as a function of the first threshold. The second threshold is upper bounded by the first threshold with a factor of four. Because the infinite number of solutions for Eq. 4.23 may exist, a computer simulation is performed to estimate an optimal multiplication factor. Even though the bound of the optimal threshold of Eq. 4.23 starts from the hypothesis of Eq. 4.7, it works well for any image no matter what the image statistic is. Simulation results are given in section 4.4.

4.3 Homogeneity Test

This section is devoted to testing homogeneities of image blocks, which are decomposed by quadtree, and constructing relationships between bit rate and number of leaves in terms of homogeneity. Usually, HVQ uses different codebooks for each block size and each codebook is generated from different training vectors. If a set of training vectors is homogeneous, the resulting codebook produces good performance when it is used to code the homogeneous blocks. Therefore, the homogeneity test is necessary.

In general, homogeneity means that each block or region includes only a small amount of high frequency components. If we examine the variation of *spatial frequency (SPF)* after a fixed amount of high frequency components is removed, the homogeneity of a given block can be measured. In other words, if a low *SPF* is achieved after removing a fixed amount of high frequency components, we can conclude that the given area is homogeneous.

The Spatial frequency (SPF) of a block is defined as in Eq. 4.26.

$$row\,frequency = \frac{1}{N^2}\sum_{i=0}^{N-2}\sum_{j=0}^{N-1}|x_{i,j} - x_{i+1,j}|. \qquad (4.24)$$

$$column\,frequency = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-2}|x_{i,j} - x_{i,j+1}|. \qquad (4.25)$$

$$spatial\,frequency = \sqrt{(row\quad freq)^2 + (col\quad freq)^2}\ . \qquad (4.26)$$

where $x_{i,j}$: each pixel in spatial domain.

In order to remove high frequency components from a block, the energy compaction property of the DCT is used. After taking the DCT of a block, 25 % of the number of coefficients in a high frequency area are removed as shown in Figure 4.6. The shaded areas of Figure 4.6 represent the removal of high frequency components. The SPF of the high frequency removed block is calculated by Eq. 4.26. To calculate the SPF of a high frequency removed block, IDCT is necessary. Finally, the homogeneities based on SPF is defined in Eq. 4.27.

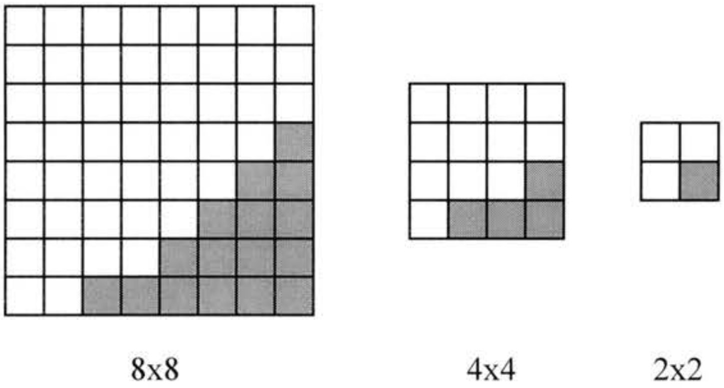


Figure 4.6 Removal of High Frequency Components

$$HSP = \frac{SPFO - SPFR}{SPFO} \times 100 \ (\%) \ (\text{when } SPFO \neq SPFR) \qquad (4.27)$$

where HSP : homogeneity based on spatial frequency,

$SPFO$: spatial frequency of original block, and

$SPFR$: spatial frequency of high frequency removed block.

HSP explores the activity of each block. The value of SPFR is always less than or equal to SPFO. In Eq. 4.27, the value of SPFR must be low if a block has few high frequency components. As a result, the homogeneity, HSP, should be high. If the 25 % of the number of coefficients in the high frequency area is originally zero before removal, SPF does not change after the removal. In this case, the homogeneity becomes 100 % with the constraint of 25 % removal. If the percentage of removing high frequency components is too high or too low, the differences of homogeneities at different bit rates are too small to compare with each other. Because of this, 25 % of the removal is empirically chosen in this experiment.

Another homogeneity testing method, which is based on statistical hypothesis, is applied to determine whether a distribution of SPF is homogeneous or not. This test explores global homogeneity of regions of different block sizes. In other words, the similarity of original and high frequency removed block is considered in terms of SPF using *Student's t-test* criteria [Steel, 1980]. Table 4.2 describes an example of pair of SPF.

Because the magnitude of SPFR is upper bounded by that of SPFO, the magnitude of SPFR is always less than or equal to that of SPFO. If two magnitudes, SPFO and SPFR, are close enough to each other, we can conclude that each block has many high frequency components, and therefore homogeneity fails. Similarly, when the differences of two magnitudes are large (i.e., the magnitude of SPFR is small), we may conclude that each block has few high frequency components, and the homogeneity is successful. With these strategies, the *hypothesis* test is performed.

SPFO (original)	$SPFO_1\ SPFO_2\ SPFO_3\ \cdots\cdots\ SPFO_N$
SPFR (high freq removed)	$SPFR_1\ SPFR_2\ SPFR_3\ \cdots\cdots\ SPFR_N$

Table 4.2 An Example of Pairs for Homogeneity Test (*N*: number of blocks)

With a hypothesis that two sample means are not different from each other (i.e., a pair is not homogeneous), Eq. 4.28 and Eq. 4.29 [Steel, 1980] are used as criterion to determine whether a pair of samples is homogeneous or not.

$$t = \frac{(\overline{O} - \overline{R}) - \mu_0}{\sqrt{\frac{S_1^{\ 2}}{n_1} + \frac{S_2^{\ 2}}{n_2}}} \qquad (4.28)$$

where $\overline{O}$: sample mean of original quantities (SPFO),

$\overline{R}$: sample mean of high frequency removed quantities (SPFR),

μ_0 : mean difference of *hypothesis* (mean difference is zero),

n_1, n_2 : number of samples of two quantities of a pair, and

S_1, S_2 : standard deviations of two quantities of a pair.

$$\begin{cases} |t| \geq t_{\frac{\alpha}{2},df} & \text{:Reject hypothesis (homogeneous)} \\ otherwise & \text{: Accept hypothesis (not homogeneous)} \end{cases} \tag{4.29}$$

$$\text{where } df = \frac{(\frac{S_1^{\ 2}}{n_1} + \frac{S_2^{\ 2}}{n_2})^2}{\frac{(S_1^{\ 2}/n_1)^2}{n_1 - 1} + \frac{(S_2^{\ 2}/n_2)^2}{n_2 - 1}}, \tag{4.30}$$

α : % of error (0.05 in this experiment).

The value of the right side of Eq. 4.29 is obtained from the Table for t-test [Steel, 1980] (Appendix IV.1). After estimating the success of each block region, *Success (S)* and *Fail (F)* are mapped into '1' and '0', respectively. The overall decision is made by the following simple relationship.

$$\textit{Total homogeneity (TH)} = B_2 \times Per_2 + B_4 \times Per_4 + B_8 \times Per_8 \tag{4.31}$$

where B_2, B_4, and B_8 : Binary value of '1' or '0', and

Per_2, Per_4, and Per_8 : percentages of each size of blocks occupy.

$$if \begin{cases} TH \geq 95 & :Success\ (S) \\ Otherwise: Fail\ (F) \end{cases} \tag{4.32}$$

This test method is effective for a sample which has a normal distribution.

Even though the SPF distribution of common images is not exactly normal, it is close to normal as can be seen from the example in Figure 4.7. This test method, therefore, is applicable. Section 4.4 represents the simulation results and discussion.

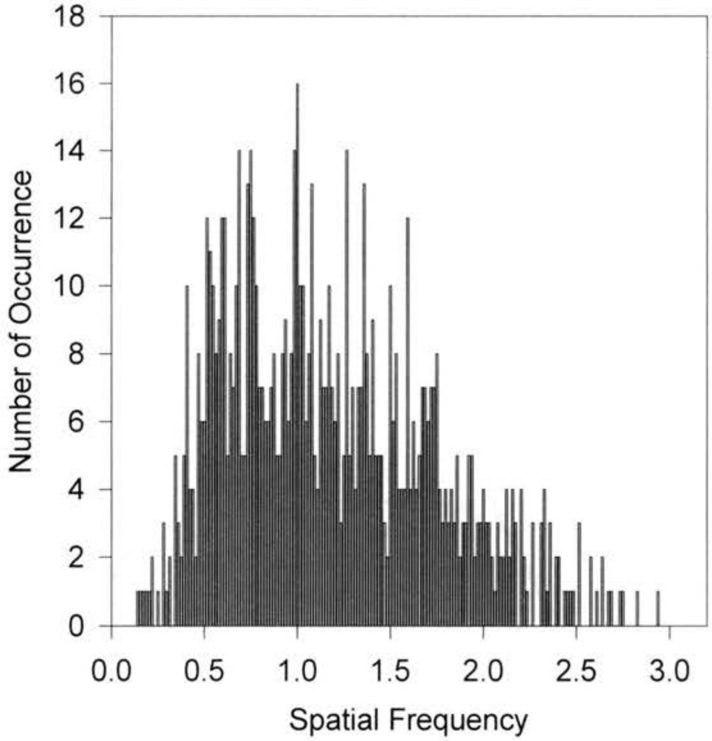


Figure 4.7 Distribution of Spatial Frequency

4.4 Simulation Results and Discussion

In order to compare the performance of the single-thresholding (ST) method to that of the multi-thresholding (MT) method, a HVQ scheme is implemented and simulated for six test images shown in Figure 4.8. Two different *homogeneity* test methods are also applied to six compressed images to explore how they are homogeneous and if their homogeneities are satisfactory.

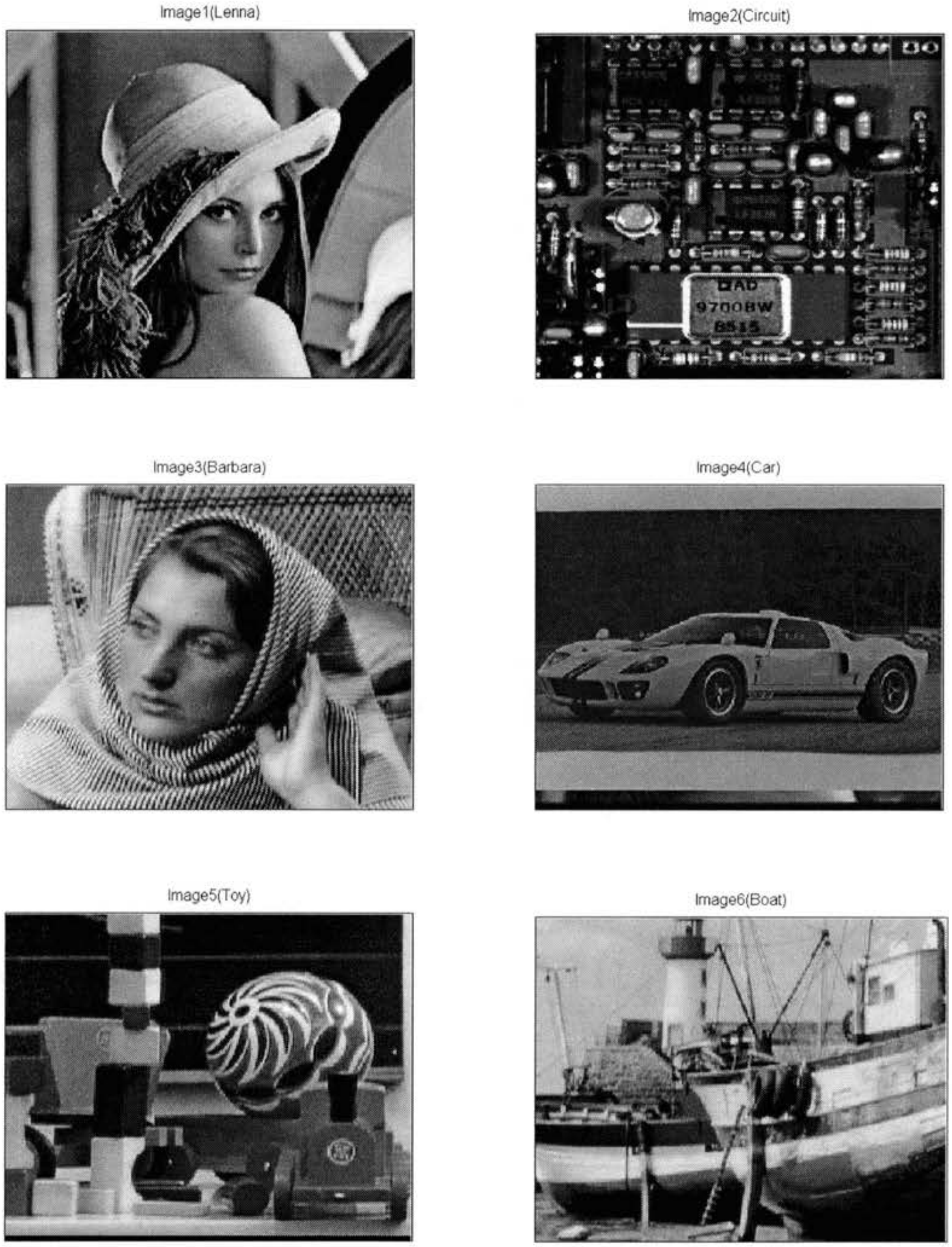


Image1(Lenna)

Image2(Circuit)

Image3(Barbara)

Image4(Car)

Image5(Toy)

Image6(Boat)

Figure 4.8 Test Images

4.4.1 Single-Thresholding and Multi-Thresholding

As mentioned in section 4.2.2, an infinite number of solutions, which satisfy Eq. 4.23, may exist. Figure 4.9 shows the relationship between MSE and nineteen point multiplication factors which are less than or equal to four. As seen from the graph, the minimum MSE (optimal MSE) is achieved at the point 2.6. The optimal point may be different from image to image because every image has various distributions of pixel values. However, the most probable occurring point for optimality is near 2.6 for common images. This point is estimated by empirical simulations.

Figure 4.10 represents the performances in terms of MSE of two thresholding methods (ST and MT) which are applied to six test images. For fair comparison, their own optimal multiplication factors instead of using 2.6 of six test images, decided by simulation, are used in the case of MT. As can be seen from the graph, the optimal ST method produces better performance than that of the MT in terms of MSE at their optimal bit rates. However, the ST method produces more leaves as shown in Figure 4.11. If we consider the number of leaves produced as a measure of efficiency, the MT method is more efficient at the cost of some more MSE.

Because the homogeneitiy is derived from SPFs of high frequency removed blocks, this value depends highly on the amount of high frequency components which is affected by thresholds. Actual homogeneities of the ST method for each image are slightly higher than those of the MT method as shown in Figure 4.12. Because the second threshold of the MT method is multiples of the first threshold, the dynamic range of SPFR of 2x2 blocks are relatively higher than that of the ST method. Actual values used for thresholding in both methods are summarized in Table. 4.4.

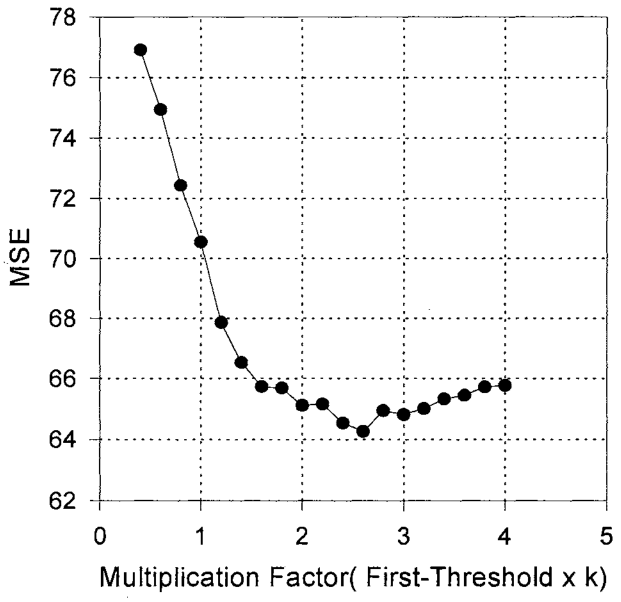


Figure. 4.9 Optimal Multiplication Factor of MT ($T_1 \le 4T_0$)

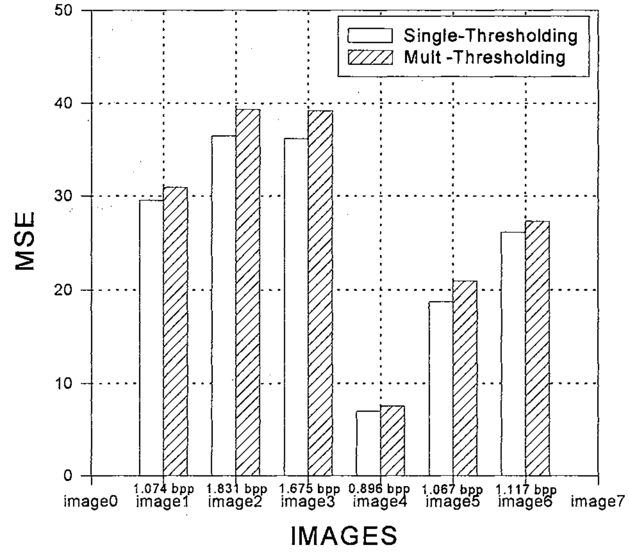


Figure. 4.10 Performances of the MT and ST

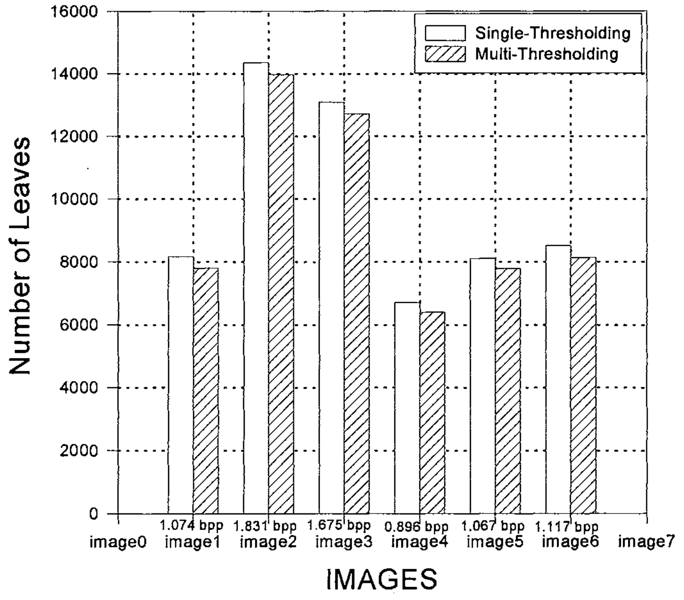


Figure 4.11 Number of Leaves of the ST and MT

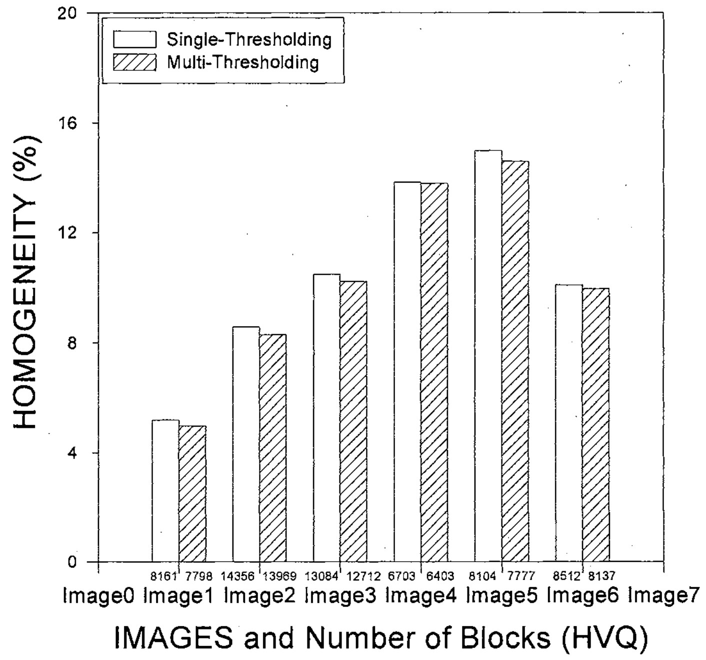


Figure 4.12 *Homogeneity* of the *ST* and *MT* at the Optimal Bit Rates Which Correspond to the Optimal Single Thresholds.

4.4.2 Relationship Between Homogeneity and Bit Rate

The relationship between homogeneity and bit rate in the HVQ scheme is explored in this section. For this simulation, *Image 1 (Lenna)* is used as a test image.

Figure 4.13 shows the homogeneities of the ST and the MT based on SPF when the bit rate varies from 0.25 *bpp* to 1.5 *bpp*. As can be seen from Figure 4.13, the homogeneity increases monotonically when the bit rate increases. An interesting point of this observation is that the MT method produces slightly higher homogeneities outside of the range around 1.0 *bpp*. The optimal threshold based on the ST method for the image I (Lenna) is 7.563026, which produces 1.074341 *bpp*. This optimal threshold is estimated from the polynomial which is displayed in Figure 4.5. Because of this, the homogeneities near the bit rate 1.0 of the ST method are higher than those of the MT method. However, the advantage of optimality cannot be obtained outside of this range when we employ the ST method. On the other hand, the MT method provides optimality for the overall range of bit rates even though its performance is slightly worse than that of the ST method at some ranges. In addition, attention cannot be fixed on a finite optimal range to get flexible compressed bit rates. Therefore, the conclusion can be drawn that the MT method is a better choice to obtain a global optimality.

In order to support the above conclusion, attention must be turned to the objective qualities of compressed images. Figure 4.14 and Figure 4.15 represent performances in terms of PSNR for the image I (Lenna) and the image VI (Boat), respectively. In both cases, the MT method produces better performance in terms of PSNR outside of the optimal range of the ST method; near 1.07 *bpp* for Lenna and 1.12 *bpp* for the Boat.

Another approach uses the statistical Student's t-test method to test homogeneity of each block region or entire image. This method provides two opposite sides, Success (S) or Fail (F), rather than producing the percentage of homogeneity. The Student's t-test is applied to SPF of six test images at their optimal bit rates. Especially, the test images I and VI undergo the testing at different *bit rates*.

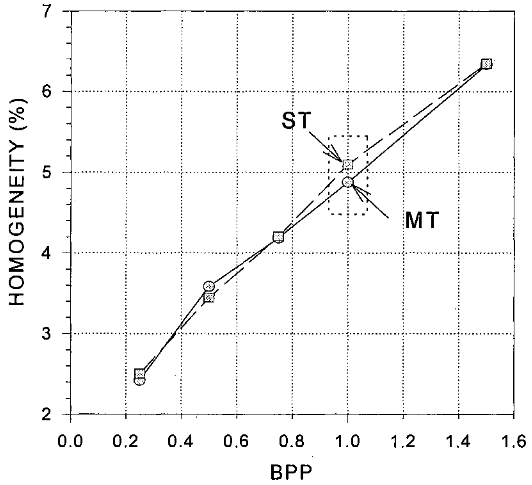


Figure 4.13 Total Homogeneity of the ST and MT Based on SPF (Lenna)

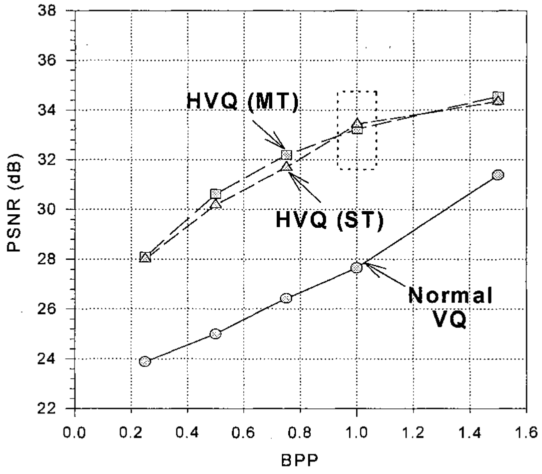


Figure 4.14 Performance of HVQ Implemented by the MT and ST (Lenna)

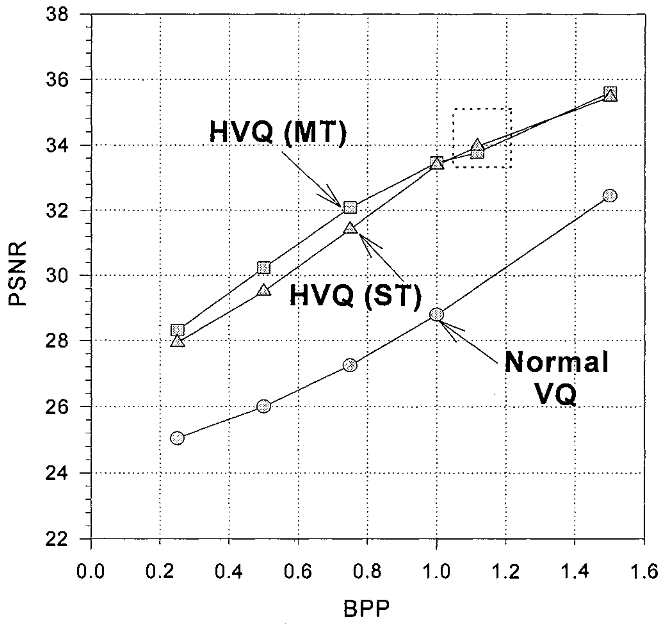


Figure 4.15 Performance of HVQ Implemented By the ST and MT (Boat)

Table 4.3 summarizes the testing results of the test images I and VI at different bit rates. As expected from the fact that the homogeneity increases when the bit rate increases, the homogeneity is successful at high bit rates. An interesting observation of this table is that the border of transition from '*F*' to '*S*' is located around their optimal bit rates which correspond to their optimal *ST* points (Lenna : 1.074 bpp and Boat : 1.117 bpp). Table 4.4 describes the information of the test images and the testing results of homogeneity at optimal bit rates. Because the optimal thresholding points are determined from the 'trading off' between bit rate and MSE in the case of the ST method, the homogeneities of test images should be increased if bit rates are increased to higher than the optimal points.

Images	BPP	MT	ST
		SPF	SPF
Lenna	0.25	F	F
optimal	0.5	F	F
bit rate :	0.75	F	F
(1.074 bpp)	1.0	S	F
	1.5	S	S
Boat	0.25	F	F
optimal	0.5	F	F
bitr ate:	0.75	F	F
(1.117 bpp)	1.0	F	F
	1.5	S	S

Table 4.3 Homogeneities of Lenna and Boat

		Lenna (1)		Circuit (2)		Barbara (3)		Car(4)		Toy (5)		Boat (6)	
		ST	MT	ST	MT	ST	MT	ST	MT	ST	MT	ST	MT
Std_Div		52.26	52.26	42.74	42.74	48	48	36.74	36.74	49.68	49.68	54.17	54.17
bpp (opt)		1.074	1.074	1.831	1.831	1.675	1.675	0.896	0.896	1.067	1.067	1.117	1.117
MSE		29.54	30.92	36.46	39.31	36.15	39.18	7.005	7.567	18.75	20.95	26.13	27.31
PSNR		33.43	33.23	32.51	32.19	32.55	32.2	39.68	39.34	35.4	34.92	33.96	33.77
TH	1st	7.563	4.049	3.26	1.07	5.06	3.748	2.934	1.947	5.936	2.389	8.983	4.327
	2nd	7.563	10.53	3.26	4.28	5.06	5.997	2.934	4.674	5.936	9.076	8.983	11.25
2x2	%	42.02	34.77	84.64	80.74	75.78	72.02	33.86	25.51	41.73	34.91	44.58	37.84
	num	6884	5696	13868	13228	12416	11800	5548	4180	6836	5720	7304	6200
4x4	%	22.24	46.68	10.77	17.7	13.67	20.36	15.55	47.53	21.85	45.26	20.85	42.33
	num	911	1912	441	725	560	834	637	1947	895	1854	854	1734
8x8	%	35.74	18.55	4.59	1.563	10.55	7.617	50.59	26.95	36.43	19.82	34.57	19.82
	num	366	190	47	16	108	78	518	276	373	203	354	203
HO	SPF	S	S	S	S	S	S	S	S	S	S	S	S

Table 4.4 Image Information (TH:Threshold, HO: Homogeneity)

Table 4.5 shows the comparison of performance of HVQ in this experiment to several compression schemes from recently published paper. The HVQ based on the MT method produces 1 dB enhanced performance when compared to that of quadtree which uses MSE as a basis for quadtree segmentation.

	PSNR (dB)			
Rate (bpp)	**HVQ**	**QT**	**AWPVQ**	**PVQ**
0.5	30.61	28.91	28.41	27.62
1.0	33.31	32.55	32.20	31.67

Table 4.5 Comparison of Several Compression Schemes (Lenna) [Shusterman, 1994]
(HVQ : Hierarchical VQ in this thesis, QT: Quadtree coder based on MSE,
PVQ : Pyramid Vector Quantizer, and AWPVQ : Adaptive weighted PVQ)

4.5 Conclusion

In order to solve the optimal thresholding problem of the quadtree segmentation method for implementing HVQ, two thresholding methods, ST and MT, are proposed and simulated. The ST method produces just one optimal thresholding point for a given input image by the polynomial. On the other hand, the MT method keeps its optimality overall ranges of bit rates. When we compare the ST and the MT methods with their homogeneities and MSE achieved, the MT method produces better results than those of the ST method outside of the range near the optimal bit rate. Although the ST method produces better performances in the sense of MSE and homogeneity at an optimal bit rate, attention cannot be fixed on a particular bit rate. Therefore, it can be concluded that the MT method is a better choice to keep optimality for the overall range of bit rates.

Because the optimal polynomial is estimated by trading off between bpp and

MSE, the optimality will decrease in the sense of trading off if the bit rates are higher than optimal bit rates. On the other hand, the homogeneity increases monotonically when bit rate increases. Thus, the optimal thresholding point of the ST method can be a transition point of homogeneity from '*Fail*' to '*Success*'. In other words, if the bit rate employed is higher than the bit rate which corresponds to the optimal thresholding point of the ST method, homogeneity turns to 'Success' as can be seen from the simulation results shown in Table 4.3. So, the optimality in the sense of 'trade off' between bpp and MSE of the optimal single threshold is verified in terms of homogeneity.

HVQ is a very efficient digital compression technique because codebooks, which have more quantization levels, can be used at the same bit rate for the representation of high detail regions. This strategy results in more than 5 dB improvement in terms of PSNR when compared to that of normal VQ as shown in Figures 4.14 and 4.15.

CHAPTER V

HYBRID CODER

Vector quantization attempts to code groups of parameters together. As a result, a large dimensionality produces a high compression ratio while resulting in an exponentially increasing number of calculations. Because of this encoding complexity, there are some limitations for employing high dimensional codebooks having a large number of codewords. If a codebook does not include enough codewords to adequately represent edges or curves, a VQ system cannot avoid producing edge degradation. This limitation restricts VQ for widespread use when it is compared to a DCT based transform coder such as JPEG.

A DCT based transform coder uses a fundamentally different coding scheme than VQ in the sense that it is based not on a vector space but a scalar space. Because each DCT coefficient is coded and transmitted separately, it produces a high quality compressed image while being somewhat limited in achieving high compression. Moreover, attempting to get a high compression ratio results in edge blurring or a lowpass effect because most of the high frequency components are removed by the quantization procedure.

In order to alleviate the inherent problems of VQ and DCT based transform coders which are listed above, two coding schemes are combined into a hybrid coder. Low detail regions are coded roughly by VQ to get the advantage of high compression while some bits which are saved by VQ are allocated to the DCT based transform coder to represent the high detail regions more accurately.

The hybridization of DCT and VQ are found in the literature [Ngan, 1991; Wen, 1993]. Wen designed a hybrid coder which combines the DCT domain VQ and spatial domain VQ. Smooth regions are coded by DCT domain classified VQ (CVQ) using the energy compaction property of the DCT, and edge regions are classified again into high and low detail regions. Low detail edge blocks are coded by the block mean while high detail edge blocks are coded by spatial domain CVQ. This scheme produces 29.52 (dB) in terms of PSNR for the Lenna image at a bit rate of 0.3979 bpp. The performance produced by this scheme is not outstanding because it produces too many overhead bits for consecutive classifications. In this scheme, the resulting total bits required to code a 8x8 smooth block is nineteen.

A modified-JPEG (MR-ICS-JPEG) scheme and a hierarchical VQ (HVQ), which uses an optimal thresholding method for quadtree segmentation, were studied in Chapters III and IV, respectively. This chapter describes a new hybrid coder, the *dual hybrid coder*, which combines HVQ, MR-ICS-JPEG and adaptive VQ (AVQ) schemes. Codebooks which are used to code low detail 8x8 blocks are generated from homogeneous training vectors. In order to reduce the overhead bits, the maximum number of classes of codebooks are restricted to seven. One class is reserved to the MR-ICS-JPEG. Therefore, eight total classes (3 bits) are necessary for overhead. In this scheme, low detail 8x8 blocks are coded by homogeneous codebooks (10 bits for each codebook index) to get high compression. Next some bits, made available by the VQ scheme, are allocated to MR-ICS-JPEG to code high detail 8x8 blocks more accurately.

5.1 Adaptive Vector Quantization

Instead of classifying image blocks into their edge classes, each block is classified by its homogeneity and coded by a specific codebook which is based on its homogeneity. A set of homogeneous codebooks was generated by one hundred 256 × 256 images (102,400 training blocks) of different characteristics using the classical LBG algorithm [Linde, 1980]. In other words, training vectors were classified into seven classes according to their homogeneities. Table 5.1 shows the classification of training vectors and their corresponding homogeneities. Average homogeneity increases when block standard deviation decreases. Figure 5.1 shows the histogram of the homogeneity distribution of the training blocks.

Training Set	Homogeneity Interval (HI)	Average Homogeneity (%)
Set 1	$10 \leq HI \leq 15$	12.9107
Set 2	$15 < HI \leq 20$	17.9660
Set 3	$20 < HI \leq 25$	22.8752
Set 4	$25 < HI \leq 30$	27.8299
Set 5	$30 < HI \leq 35$	32.7365
Set 6	$35 < HI \leq 40$	37.7343
Set 7	$HI > 40$	43.2458

Table 5.1 Classification of Training Vectors

For each set of training vectors, the LBG algorithm is applied to generate a set of homogeneous codebooks. Figure 5.2 represents the PSNR when the training vectors are coded by their corresponding codebooks. As can be seen from the graph, the performance is good for highly homogeneous training vectors. We can expect from this graph that the percentages which are occupied by VQ should be large and small at high and low compression, respectively, to get better performance in terms of PSNR. Figure 5.3 presents the block diagram of the adaptive VQ scheme. Adaptive VQ is applied to only low detail (highly homogeneous) blocks in this coder.

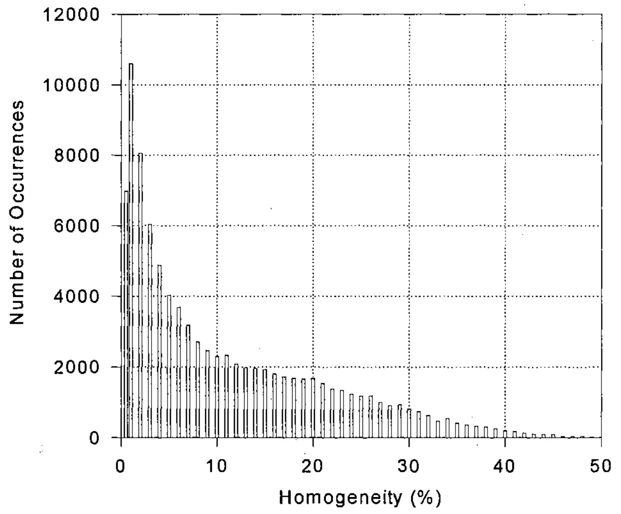


Figure 5.1 Homogeneity Distribution of the Training Blocks

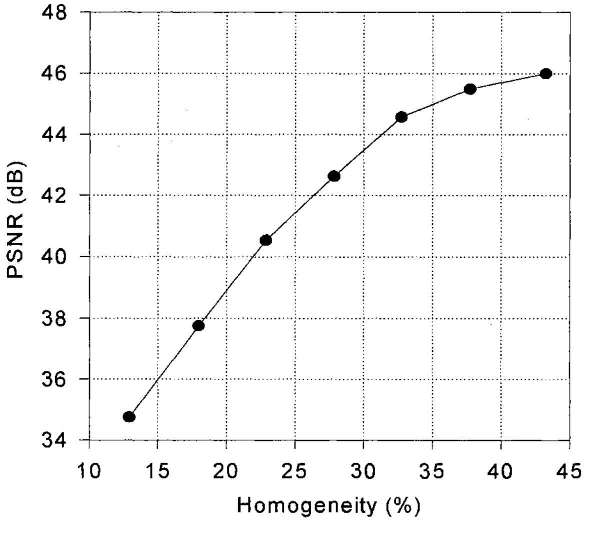


Figure 5.2 Performance of Homogeneous Codebooks (8x8 Block (Training Vectors))

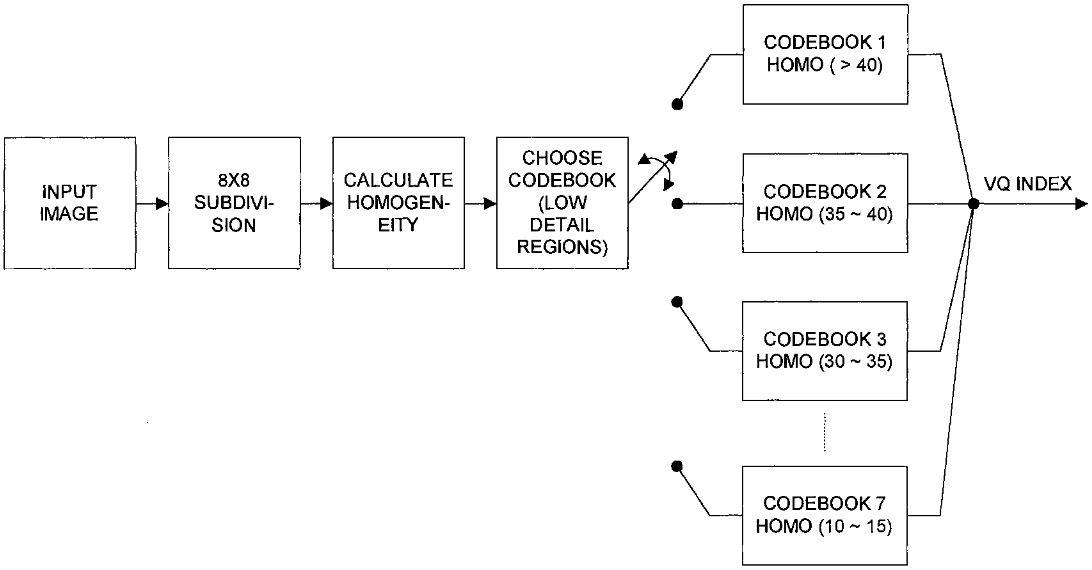

INPUT IMAGE
8X8 SUBDIVI-SION
CALCULATE HOMOGEN-EITY
CHOOSE CODEBOOK (LOW DETAIL REGIONS)
CODEBOOK 1 HOMO (> 40)
CODEBOOK 2 HOMO (35 ~ 40)
CODEBOOK 3 HOMO (30 ~ 35)
CODEBOOK 7 HOMO (10 ~ 15)
VQ INDEX

Figure 5.3 Block Diagram of Adaptive VQ (AVQ)

5.2 Structure of a new hybrid coder

Table 5.2 represents the possible candidates for hybridization. Because the best performance of DCT based transform coding is achieved at block size 8x8 [Pennebaker, 1991], methods II and IV are not good choices. Moreover, method IV requires too many overhead bits which are produced by quadtree segmentation and adaptive VQ. Therefore, the combination of quadtree and adaptive schemes is not a good choice. Method III produces reasonable overhead; however, AVQ requires too many bits to code blocks as small as 2x2. In method V, the error vectors are usually more random than the original ones. As a result, it is hard to achieve energy compaction from the DCT coding step. Therefore, the objective qualities from the methods II to V in terms of PSNR are lower than that of MR-ICS-JPEG alone.

On the other hand, method I produces no more than 3 bits of overhead, and more bits can be allocated to MR-ICS-JPEG because low detail 8x8 blocks are coded by AVQ. As a result, method I produces better performance than when MR-ICS-JPEG is used alone. Figure 5.4 shows the simulation result of method I. As can be seen from Figure 5.4, method I produces about a 3.5 dB improved performance compared to that of standard JPEG when the bit rate is larger than 0.30. However, in the high compression areas, the performance is below than that of HVQ. Performance in the high compressed bit ranges is shown in Figure 5.5. This reduction in performance is caused by an inherent property of the DCT based coder (MR-ICS-JPEG). In other words, an additional 4 bits is necessary to represent the end of block (EOB) in the DCT based coder. Because of this overhead, it is very hard to achieve a high compression ratio with the DCT based scheme.

Methods Block size		I	II	III	IV	V
8×8	Low detail	AVQ	AVQ	MR-ICS-JPEG	AVQ	Multi-stage coding: 1st stage: HVQ 2nd stage: MR-ICS-JPEG
	High Detail	MR-ICS-JPEG				
4×4		×	MR-ICS-JPEG	×	MR-ICS-JPEG	
2×2		×	×	AVQ	AVQ	

Table 5.2 Possible Candidates for Hybridization

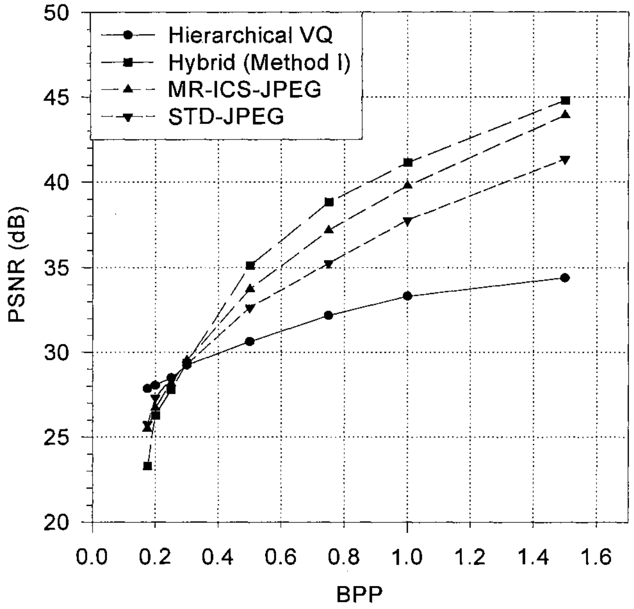


Figure 5.4 Performance of Method I and Other Compression Schemes (Lenna)

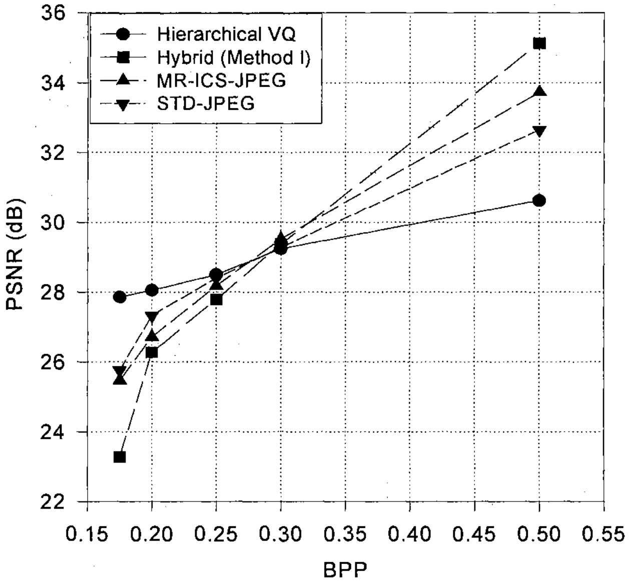


Figure 5.5 Performances of Method I and Other Compression Schemes
(Low Bit Ranges, Lenna 256x256)

Because of reasons listed above, a *new hybrid coder* is the combined version of HVQ and method I. In other words, the coding scheme is switched to HVQ when the bit rate set point we want to get is equal to or lower than about 0.30 bpp. Otherwise, it is switched to method I. Therefore, the new hybrid coder is referred to as a *dual hybrid coder*. However, this bit rate switching point will be slightly different from image to image.

In order to get the general description of this switching point, the switching points of one hundred 256 × 256 gray scale images were examined in terms of corresponding average block homogeneities. Even though the distribution of the switching points, distributed from about 0.25 to 0.3 bpp, is not tightly clustered, it is clearly evident that the switching point decreases when the average block homogeneity increases. When the value of average block homogeneity for a given image is small, most blocks are coded by the DCT based coder, MR-ICS-JPEG. As a result, there is some difficulty in obtaining high compression. Because of this, the switching point from Method I to HVQ increases. Moreover, at low bit rates, about from 0.3 to 0.25 bpp, the overhead of HVQ is reduced to 1 bit for each 8 × 8 image block because the block size of 2 × 2 is removed at high compression. As a result, HVQ produces better performance than the hybrid coder (Method I) at low bit rates. Figure 5.6 shows the linear regression equation as a function of average block homogeneity for a given image. Using this linear regression equation, we can estimate the bit rate switching point as a function of average block homogeneity (ABH) between Method I and HVQ. As an example, Table 5.3 shows the bit rate switching points for six test images which were used in the previous chapter. Finally, the flow chart of the proposed new hybrid coder is shown in Figure 5.7.

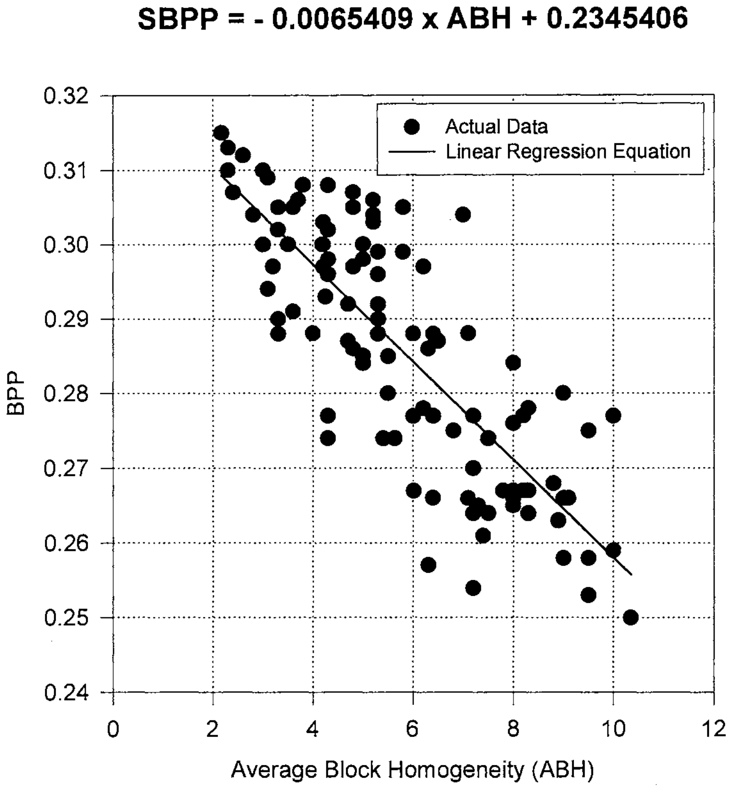


Figure 5.6 The Switching Point of bpp as a Function of Average Block Homogeneity (SPBP : Switching Point of Bits Per Pixel)

Images	ABH (%)	Switching Point (SPBP)
Lenna	2.169	0.309260
Circuit	1.141	0.315986
Barbara	4.186	0.296069
Car	10.356	0.255712
Toy	10.255	0.256373
Boat	7.302	0.275688

Table 5.3 Bit Rate Switiching Points Based on the Linear Regression Equation for Six Test Images

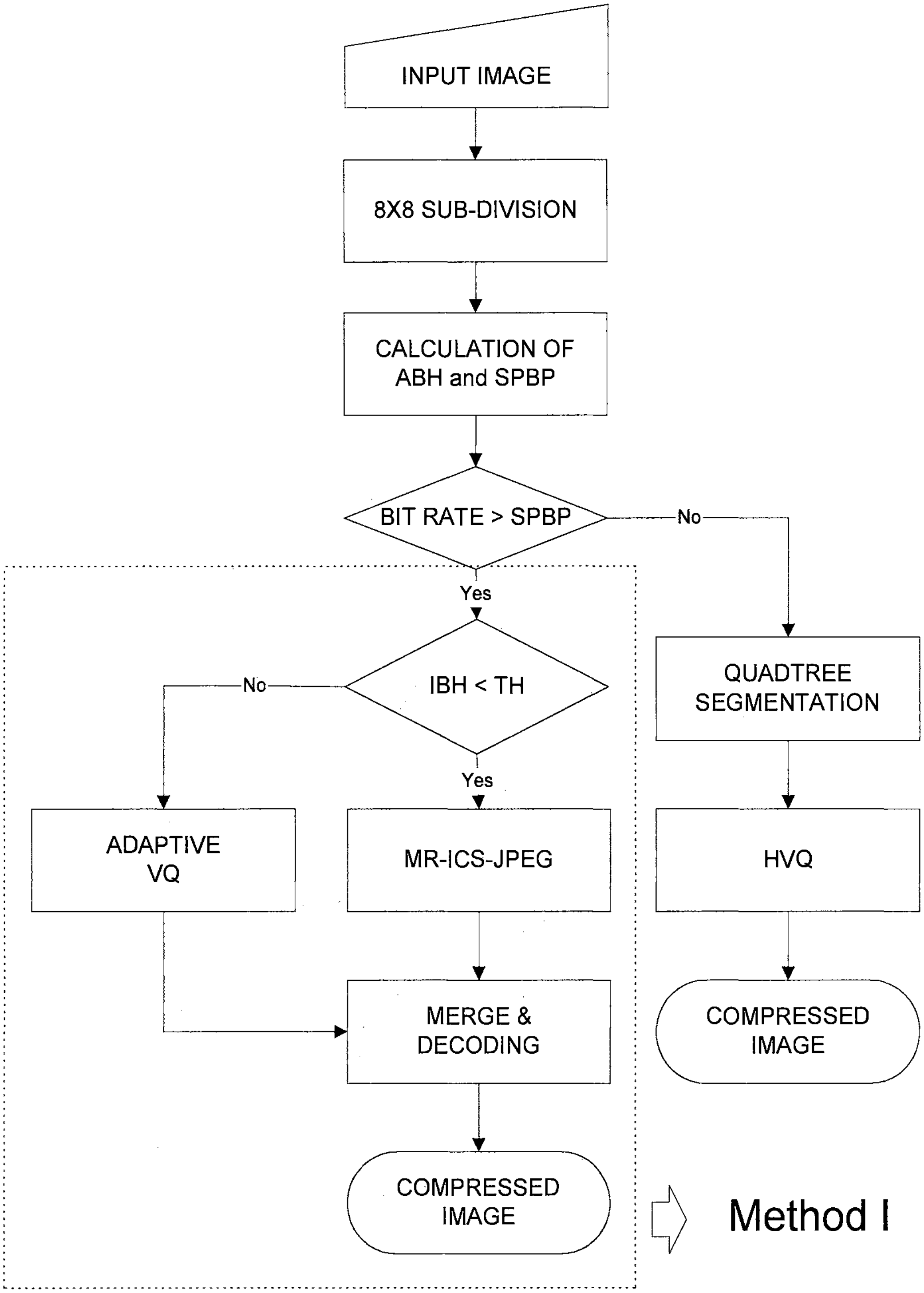


Figure 5.7 Flow Chart of the Dual Hybrid Coder
(ABH : Average Block Homogeneity; IBH: Individual Block Homogeneity)

5.3 Discussion of Simulation Results

Simulation is performed for two test images, Lenna and Barbara. Lenna is included in the training vectors while Barbara is outside the training vectors. When the AVQ scheme is compared to the MR-ICS-JPEG, the AVQ and the MR-ICS-JPEG produce better performances in the range of high and low compression, respectively. Therefore, at a bit rate of 1.5 bpp, the highest performance is achieved at 2 bits of overhead which corresponds to 28.16 (%) VQ coded area for the Lenna image (homogeneity > 35). On the other hand, when the bit rate is lowered, the highest performance is obtained when the percentage of the VQ coded area is increased. In other words, the peak values of PSNR for Lenna are achieved at 55.56 (%), 61.33 (%), and 65.92 (%) of the VQ coded area for the bit rates of 1.0, 0.75, and 0.5 bpp, respectively. Figure 5.8 summarizes this phenomenon.

When the percentage of the VQ coded area increases, some additional bits can be allocated to high detail regions. However, VQ coded areas produce some blockiness at the shoulder area of the Lenna image when the less homogeneous areas, which are less than 30 percents in terms of homogeneity, are coded by VQ. Therefore, the best subjective qualities at bit rates larger than 0.75 bpp are achieved when VQ coded areas are restricted to 28.61 (%) for the Lenna image.

One of the advantages of the DHC is its adaptivity. In other words, low detail regions are coded roughly by AVQ while high detail regions are coded finely to represent the edges or curves more accurately. By this strategy, the DHC produces not only objectively, but also subjectively enhanced quality of compressed images. The details of the subjective judgment of image quality are given in Chapter VI. Figure 5.9 represents how many additional bits can be allocated to high detail regions using the method I of DHC scheme compared to MR-ICS-JPEG used alone. The x-axis represents the sequence of blocks which are coded by MR-ICS-JPEG and sorted in terms of block standard deviation. As can be seen in Figure 5.9, bit rates of approximately 1.5, 1.0, 0.5, and 0.25 bpp can additionally be allocated to MR-ICS-JPEG at the bit rates of 1.5, 1.0, 0.75, and 0.5 bpp, respectively. This is the main source for enhancement of the objective and subjective qualities.

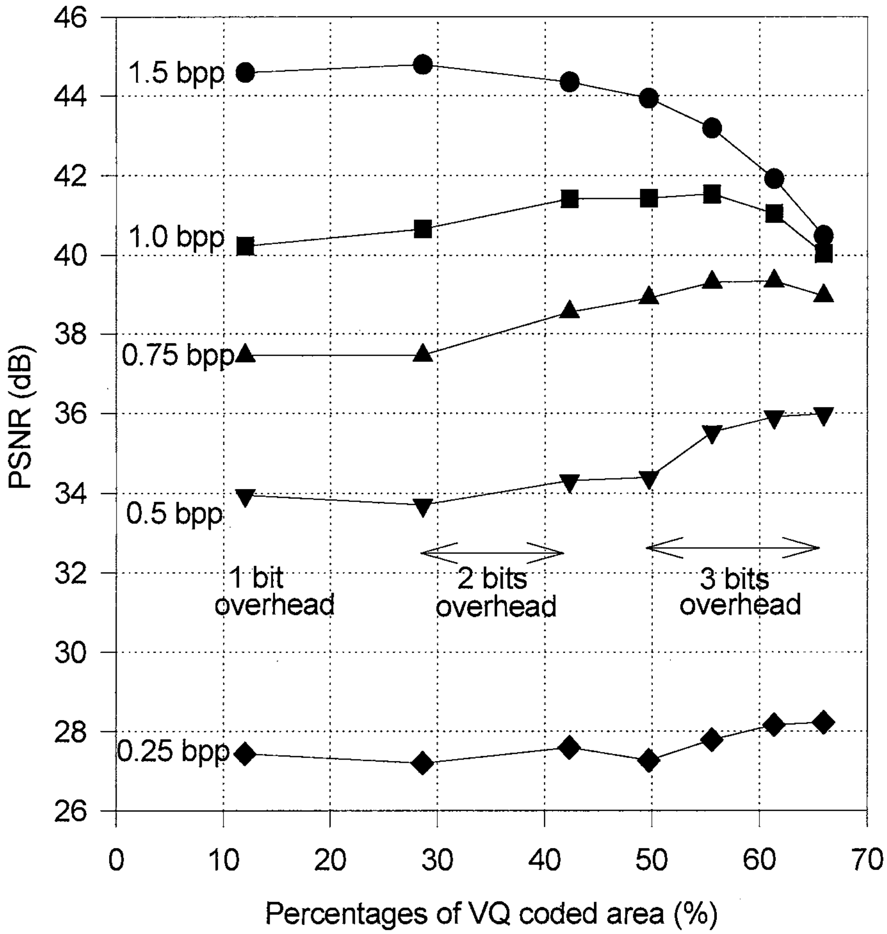


Figure 5.8 Performances when the percentages of VQ coded area are varied (Lenna)

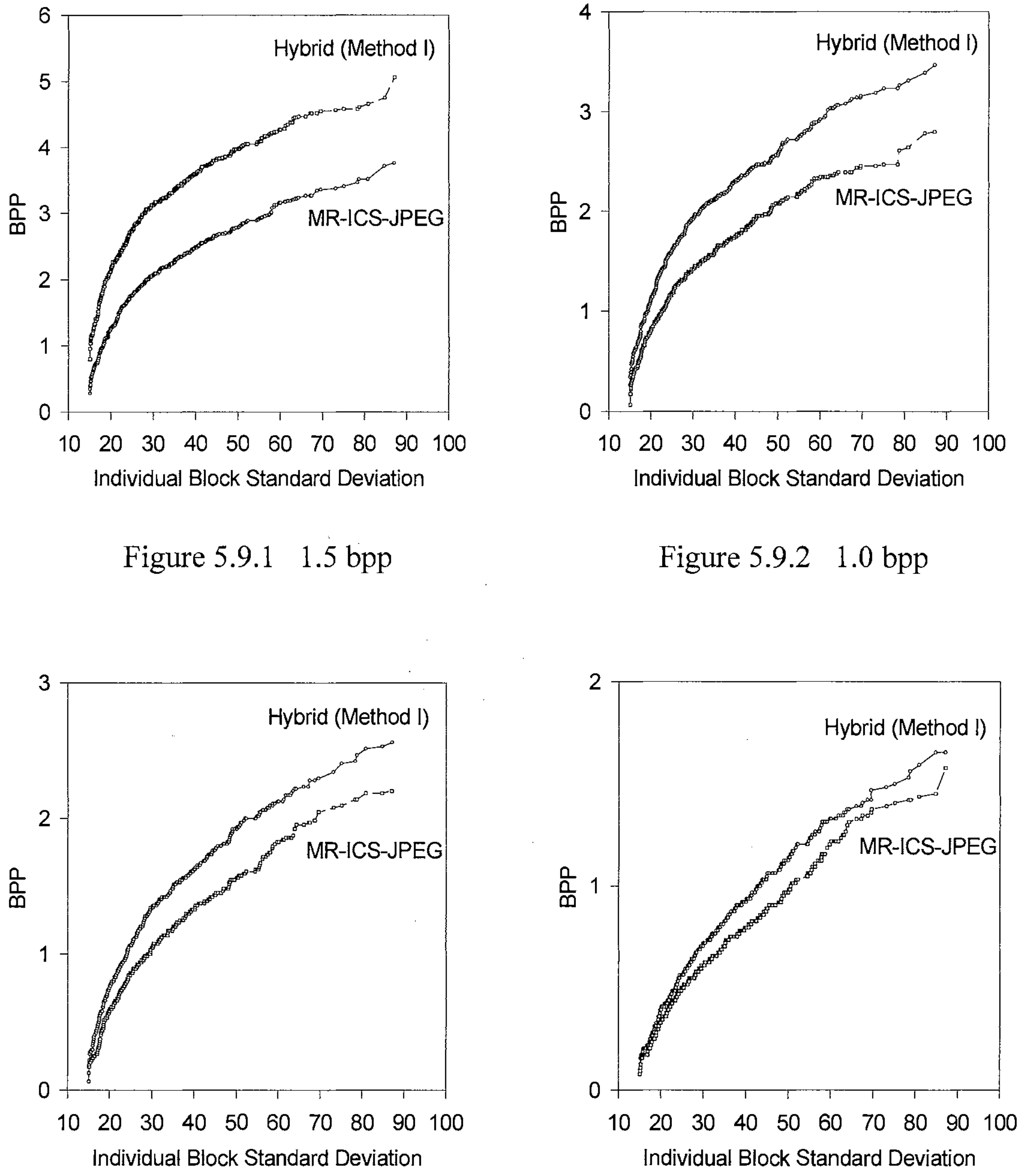


Figure 5.9.1 1.5 bpp

Figure 5.9.2 1.0 bpp

Figure 5.9.3 0.75 bpp

Figure 5.9.4 0.5 bpp

Figure 5.9 Additional bit allocation to MR-ICS-JPEG.

When an input image is outside the training vectors, the performance in terms of PSNR of the DHC is lower than when an input image is inside the training vectors. Figure 5.10 shows the performances in terms of PSNR of Barbara image which is outside the training vectors. The performance of the DHC is about 2.5 dB higher than that of JPEG. Even though the objective quality improvement of the DHC is not significant, it produces better subjective quality because some additional bits which are reduced by AVQ are allocated to high detail active areas.

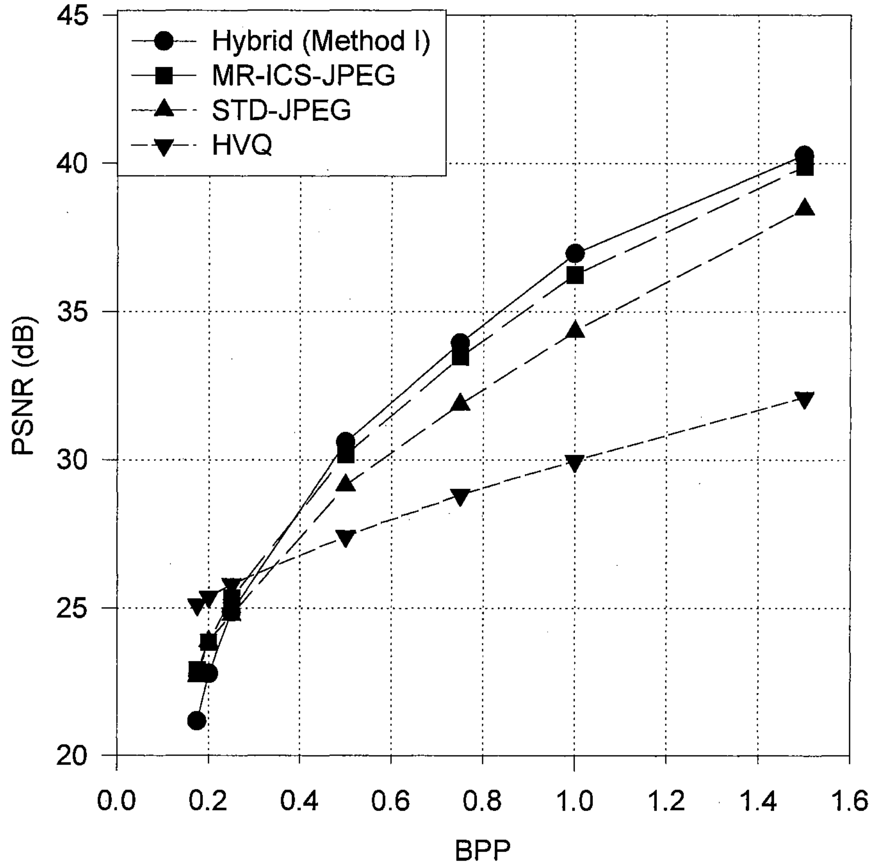


Figure 5.10 Performances of Different Compression Schemes (Barbara)

Figure 5.11 represents the performances when the bit rates are less than 0.5 bpp. As can be seen from the graph, HVQ produces better performances than those of others at the bit rates lower than about 0.30 bpp, 0.296069 from the linear regression equation, similar to Figure 5.5. Therefore, the DHC of Figure 5.7 is effective in both cases of images which are inside and outside of training vectors.

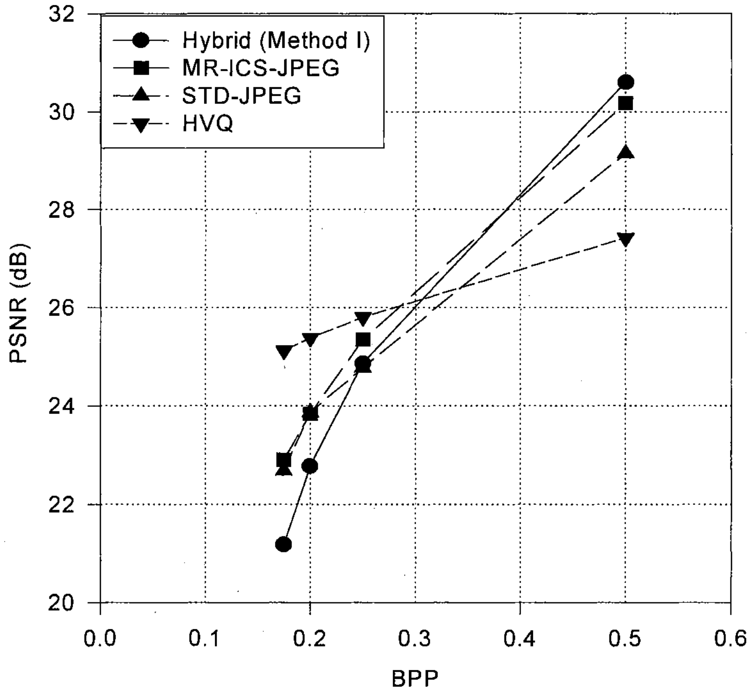
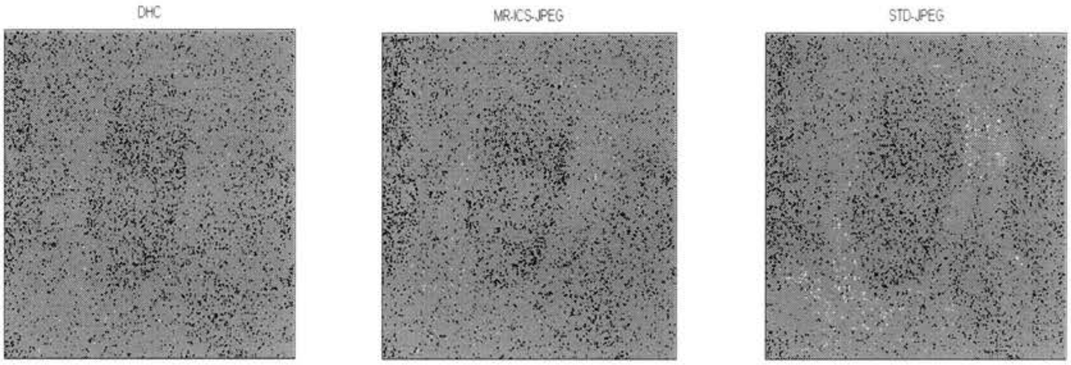


Figure 5.11 Performances of Different Compression Schemes (Low Bit Range, Barbara)

Figure 5.12 represents the error images of Barbara generated by different schemes. In Figure 5.12, pixels which produce zero error, less than or equal to '30', and larger than '30' are displayed by pure black, gray, and white, respectively. In this comparison, the DHC produces the fewest white spots which represent pixel error larger than '30'.



Black : 5,050	Black: 4,723	Black: 4,676
Gray: 60,360	Gray : 60,645	Gray: 60,261
White : 126	White : 168	White : 599

Figure 5.12 Error Images of Barbara (0.5 bpp)
(zero error ('black'), less than or equals to 30 ('gray'), and larger than 30 ('white'))

Another representation of image quality, the histogram of error images for Barbara which are generated by different compression schemes at diferent bit rates, is presented in Figure 5.13. The stronger spike at zero error means the better image quality. The strongest spikes at zero error can be observed in the DHC. Because some additional bits which are reduced by AVQ are allocated to the MR-ICS-JPEG for representing high detail regions more accurately in this scheme, the number of occurrences of zero error are increased. When the bit rates increase, the magnitude of the spike, which represents the number of occurrences of zero error, increases. On the other hand, the width of the main lobe and the number of pixels which produces large errors decrease when the bit rates increase because the increasing bit rate reduces the magnitudes of pixel errors.

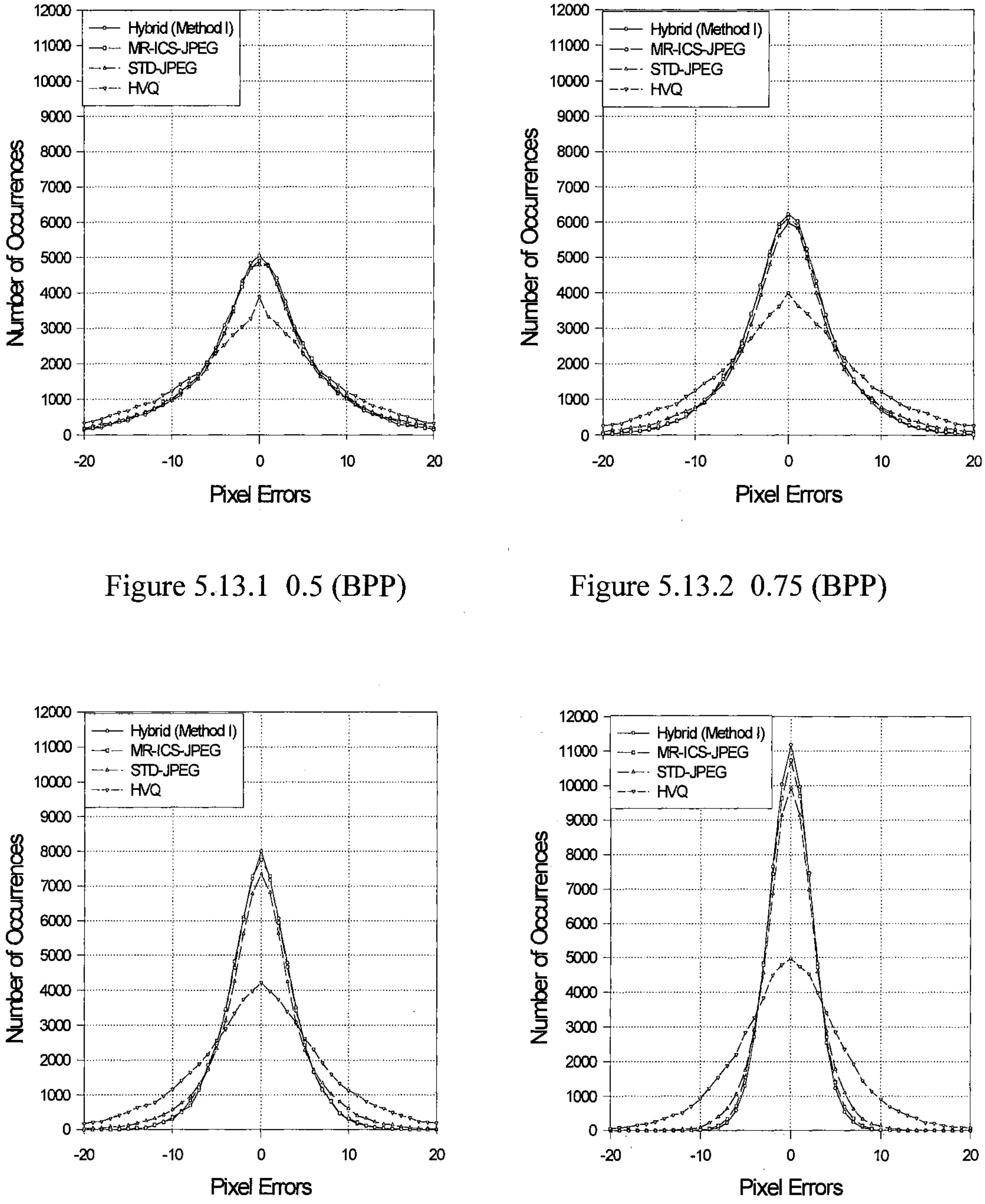


Figure 5.13.1 0.5 (BPP)

Figure 5.13.2 0.75 (BPP)

Figure 5.13.3 1.0 (BPP)

Figure 5.13.4 1.5 (BPP)

Figure 5.13 Histograms of Error Images (Barbara (outside the training vectors))

The other advantage of the proposed DHC is the low overhead in bits. Maximum overhead in an 8x8 block is restricted to 3 bits in this scheme, and this overhead can be reduced to 2 bits in the range of low compression (refer to Figure 5.8) without sacrificing image quality.

In order to verify the superiority of the proposed schemes, DHC and MR-ICS-JPEG, direct comparisons in terms of PSNR with other published results are given in Figure 5.14.

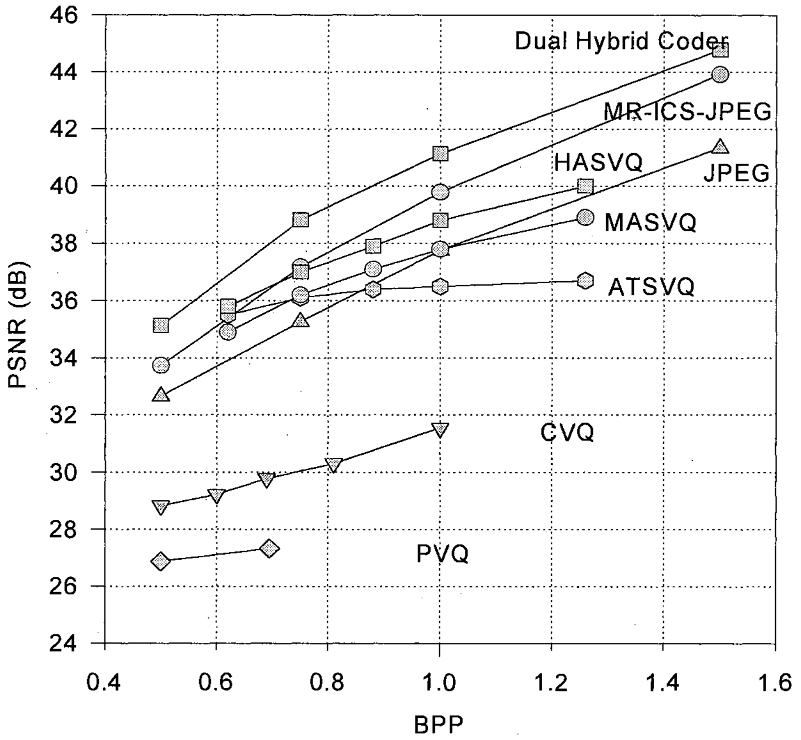


Figure 5.14 Comparison of Several Decent Compression Schemes (Lenna)[5.1]

In high compression areas, the performance of transform based coders such as

[5.1] HASVQ (Hierarchical Adaptive Search VQ), MASVQ (Multi-Stage Adaptive Search VQ), and ATSVQ (Adaptive Tree Search VQ) [Ghafourian, 1995]. CVQ (Classified VQ) [Ramamurthi, 1986]. PVQ (Pyramid VQ) [Kim, 1992].

MR-ICS-JPEG and DHC is not outstanding because an additional 4 bits is necessary to represent the end of block (EOB) for each block. However, this additional 4 bits is just a small portion of the total allowed bit rates at the ranges of low compression. As a result, the proposed dual hybrid coder produces better performance in terms of PSNR than HASVQ, MASVQ, and ATSVQ.

5.4 Conclusions

A dual hybrid coder (DHC) which combines HVQ, MR-ICS-JPEG, and AVQ was described and simulated. This coder was shown experimentally to produce about 3.5 dB and 2.5 dB improved performance when an input image is inside and outside of the training vectors, respectively, compared to JPEG. Because the HVQ produces better performance than Method I at low bit rates, the DHC is switched to HVQ when bit rates are less than or equal to about 0.30 (bpp). However, at bit rates which are larger than about 0.30 (bpp), the DHC is switched to Method I. However, this bit rate switching point will be slightly different from image to image. The linear regression equation, given in Figure 5.6, estimates the bit rate switching point for a given image as a function of the average block homogeneity.

The advantages of the proposed DHC are its adaptivity and low overhead in bits. In other words, low detail regions are coded roughly by AVQ, and some additional bits, which are made available by AVQ, are allocated to MR-ICS-JPEG to represent high detail regions more accurately. This strategy and low overhead produce the subjectively and objectively enhanced image quality.

CHAPTER VI

QUALITY MEASURES AND THEIR PERFORMANCES

In general, an image quality measure should have three main factors: First, it should be subjectively meaningful in the sense that large and small distortions correspond to subjectively bad and good quality, respectively. Second, it should be tractable for mathematical modeling. Third, the computational cost should not be too expensive.

The subjective quality measures require a deeper analysis of *the human visual system (HVS)*. However, the HVS is so complex that we can not fully understand it with present psychophysical methods [Eskicioglu, 1995]. In addition, the HVS has inherent complexities that make it very hard to develop a mathematical model, and intensive computation is necessary. Because of inherent complexities and drawbacks of the subjective quality measures, objective quality measures have been used either in graphical or numerical forms.

A graphical objective quality measure, histogram, has been reported [Eskiciglu, 1992,1995]. The histogram constructed by plotting the number of occurrences of the same intensities of error images represents the amount of degradation: A strong and weak spike at the point of zero error correspond to good and bad image quality, respectively.

Even though numerical objective quality measures are used for measuring image qualities, these numerical qualities, sometimes, are not exactly matched with the human visual response. In this chapter, several objective quality measures are described and applied to the compressed images from the previous chapters in order to ascertain which quality measures are well correlated with the human visual response.

Several subjective quality rating scales are reported in the literature [Eskicioglu, 1993]. This chapter attempts to find the correlations between objective and subjective qualities. The subjective judgments are performed by thirty human observers. Finally, the *Pearson product-moment correlation coefficient* [Bajpai,1979] is applied to determine the subjectivity of the objective quality measures. Using this method, the best subjectively meaningful objective quality measure as well as the best compression method in the sense of subjectivity are decided.

6.1 Objective Quality Measures

6.1.1 Quality Measures Based on Mean Square Error

The objective quality measures discussed below are based on mean square error (MSE) criteria. MSE is the most commonly used difference measure because it is not only correlated resonably with subjective visual quality test but is also mathematically tractable [Pratt, 1991].

1) Mean Square Error (MSE)

For any value of x and y in the range $0,1,...,N\text{-}1$, the error between an input pixel and the corresponding output pixel is

$$e(x,y) = f(x,y) - \hat{f}(x,y). \tag{6.1}$$

The squared error averaged over the image array is

$$MSE = \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[e(x,y)]^2 = \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[f(x,y) - \hat{f}(x,y)]^2. \qquad (6.2)$$

Eq. 6.2 defines the normal definition for *MSE* of images. All the following error criteria are variations of Eq. 6.2.

2) Normalized Mean Square Error (NMSE)

$$NMSE = \frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[f(x,y) - \hat{f}(x,y)]^2}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f^2(x,y)} \qquad (6.3)$$

This measure represents the division of the sum of the squared error with the unweighted summation of squared pixel values. The possible values of NMSE range from '0' to '1'. The values of '0' and '1' correspond to lossless and the worst case reconstruction, respectively. This measure gives an intitutive quantification of image quality. In other words, the value of NMSE details how close a compressed image relates to the original one. The variations of this measure with different weighting functions are given in Eq. 6.4 through Eq. 6.8.

3) Perceptual MSE (PMSE)

$$PMSE = \frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}e^2(x,y)[f(x,y) - \hat{f}(x,y)]^2}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}e^2(x,y)f^2(x,y)} \qquad (6.4)$$

This equation is a weighted version of NMSE by the squared error image.

Usually, normal MSE is not exactly related to the subjective quality of image. PMSE attempts to make MSE more subjectively meaningful by weighting more heavily in the large error pixels. The possible values of PMSE, like those of NMSE, range from '0' to '1'. No other empirical data has been reported to verify its performance [Tannas, 1985].

4) Laplacian Mean Square Error (LMSE)

$$LMSE = \frac{\sum_{x=1}^{N-2}\sum_{y=1}^{N-2}[O\{f(x,y)\} - O\{\hat{f}(x,y)\}]^2}{\sum_{x=1}^{N-2}\sum_{y=1}^{N-2}[O\{f(x,y)\}]^2} \tag{6.5}$$

where *O{f(x,y)}* is an estimation of the *Laplacian Operator* defined as

$$O\{f(x,y)\} = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y).$$

Usually, local feature detection operators are used to extract the local features such as edges and curves that are defined by local patterns of gray levels [Ekstrom, 1984]. Combining MSE with *Laplacian,* as one of local feature detection operators, LMSE explores the amount of distortion, especially of edges and curves between original and compressed images. This measure emphasizes perceptual features because errors at edges or curves are more visible than most other types.

5) Image Fidelity (IF)

$$IF = 1 - \frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[f(x,y) - \hat{f}(x,y)]^2}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[f(x,y)]^2} \tag{6.6}$$

In 1960, Linfoot defined the *image fidelity (IF)* as unity minus the fidelity deficit which is defined earlier as NMSE in Eq. 6.3. The IF represents the overall quality of the compressed image rather than the variations of individual pixel intensities. In other words, IF is a measure which shows the degree of closeness between the original and compressed images. Fidelity value '1' corresponds to lossless reconstruction.

6) Peak Signal to Noise Ratio (PSNR)

$$PSNR = 10\log_{10}(\frac{255^2}{MSE}) \quad \text{for 8-bit image} \qquad (6.7)$$

PSNR is defined as 10 times the *log* to the base 10 of the ratio of peak input signal strength of 255 squared to the MSE. This measure is the most commonly used for ordinary still images. This measure attempts to quantify image quality; however, high PSNR values do not always correspond to signals with perceptually high quality.

6.1.2 Others

1) Average Difference (AD)

$$AD = \frac{1}{N^2}\sum_{x=0}^{N\text{-}1}\sum_{y=0}^{N-1}[|f(x,y) - \hat{f}(x,y)|] \qquad (6.8)$$

This error measure is based on the absolute value of error. It only explores the average difference of each pixel value while the squared error measure emphasizes large differences.

2) Maximum Difference (MD)

$$MD = Max\{|f(x,y) - \hat{f}(x,y)|\} \tag{6.9}$$

Maximum difference, often called the worst case pixel difference, represents the maximum distortion taken over all possible input pixels. Because this measure is unbounded, the possible value can be extended to the maximum pixel intensity in certain cases.

3) Normalized Cross-Correlation (NK)

$$NK = \frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\hat{f}(x,y)}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f^2(x,y)} \tag{6.10}$$

The value of cross correlation (numerator) is always less than the energy (denominator) of the original image except for lossless reconstruction. The possible values of NK range from '0' to '1'. Closeness to '1' and '0' correspond to good and bad image quality, respectively. This measure also can be applied to the area of template matching as an indicator of similarity.

4) Lp - Norm

$$Lp = \{\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}|f(x,y)-\hat{f}(x,y)|^p\}^{\frac{1}{p}}, \; p=1,2,3 \tag{6.11}$$

when $p=1$: *average difference (AD)* and

$p=2$: *root mean square error (RMSE).*

The factor p determines the relative importance of small and large errors. The larger the p value, the greater relative emphasis is given to large errors in an image. The L_2 - norm is the most commonly used. The statistics of the deviations determines which is the correct norm to be taken. The L_2 - norm corresponds to the normal distribution [Menke, 1984].

The quality measures above are based upon the quantification of the visual system or some theoretical approach for modelling of image quality. However, at the present time there is no quality measure that can be singled out as the best quality measure [Tannas, 1985].

6.2 Subjective Quality Measures

In the previous section, several objective quality measures are listed. Because the final users of compressed images are human observers, the exploration of the correlation between objective and subjective quality measures is important. One of the subjective quality measures, *Rating Scale Method (RSM)*, is a commonly used method to judge picture qualities by human observers. In this research, five different rating scales are used as shown in Table 6.1. The degraded images compressed by different compression schemes (DHC ,MR-ICS-JPEG, Std-JPEG, and HVQ) are judged by thirty human observers using RSM as defined in Table 6.1.

A	B	C
5: Excellent	5: Imperceptible	10, 9: Very good
4: Good	4: Perceptible but annoying	8, 7: Good
3: Fair	3: Slightly annoying	6,5,4: Fair
2: Poor	2: Annoying	3, 2: Bad
1: Unsatisfactory (bad)	1: Very annoying	1, 0: Very bad

D	E
7: Not noticeable (perceptible)	7: Best
6: Just noticeable (perceptible)	6: Well above average
5: Definitely noticeable (perceptible) but only slight impairment	5: Slightly above average
4: Impairment not objectionable	4: Average
3: Somewhat objectionable	3: Slightly below average
2: Definitely objectionable	2: Well below average
1: Extremely objectionable	1: Worst

Table 6.1 Quality and Impairment Rating commonly used [Eskicioglu, 1993]
(each number represents the score for the corresponding judgement)

6.2.1 Rating Scale Methods

The results of the rating scale experiment are usually represented by computing a *Mean Opinion Score (MOS)* [Bajpai, 1979].

$$MOS = \frac{\sum_{k=1}^{N} S_k N_k}{\sum_{k=1}^{N} N_k} \qquad (6.12)$$

where S_k = the score corresponding to the kth rating,

N_k = the number of observers with this rating, and

N = the number of grades in the scale.

The rating-scale methods are commonly used in the broadcast television area because they are very useful for setting up and maintaining an appropriate grade of service [Netravaii, 1995].

6.3 Pearson Product-Moment Correlation Coefficient

This section introduces a correlation measure, *Pearson Product-Moment Correlation coefficient*, which is used to exploit the correlation between objective and subjective qualities of compressed images. Suppose that the results from the objective quality measures are O_k, and the results from the rating scale methods (subjective quality measures) are S_k. For a sample pair of observations of O_k and S_k, the value *COC* (Correlation Coefficient) of this correlation is given by the following formula [Bajpai, 1979].

$$COC = \frac{\sum_{k=1}^{N}(O_k - \overline{O})(S_k - \overline{S})}{\sqrt{\sum_{k=1}^{N}(O_k - \overline{O})^2 \sum_{k=1}^{N}(S_k - \overline{S})^2}} \tag{6.13}$$

where N: the number of pairs of sampling ,

$\overline{O}$: Average value of O_k, and

$\overline{S}$: Average value of S_k.

The possible values of COC range between +1 and -1. If the value of COC is close to +1 or -1, two quality measures (subjective and objective) are strongly correlated with each other. On the other hand, if it is close to zero, we can decide that they are weakly or not correlated with each other. In other words, if the value of COC is close to +1 or -1, the corresponding objective quality measure is subjectively meaningful.

6.4 Experimental Results and Discussions

Figure 6.1 represents the several objective performances for the test image Barbara which is outside of the training vectors. Because all graphs except for Figures 6.1.5 and 6.1.6 are based on MSE, they show similar shapes. At the bit rate 0.25 bpp, MR-ICS-JPEG produces smaller MSE than that of the DHC scheme because it does not include any overhead bits. However, the DHC produces smaller MSE than that of MR-ICS-JPEG at bit rates larger than 0.30 bpp. Figure 6.1.2 shows the NMSE. Even though its theoretically possible values range from '0' to '1', its actual values are close to '0'. Therefore, we cannot get the intuitive NMSE difference among different bit rates. In Figure 6.1.3, attempting to get more perceptually meaningful results by emphasizing large errors makes STD-JPEG produce higher PMSE. Therefore, the PMSE difference between MR-ICS-JPEG and STD-JPEG is larger than the MSE difference of Figure 6.1.1. The DHC scheme produces smaller MD and L_1(AD) than others as can be seen from Figures 6.1.5 and 6.1.6. LMSE of Figure 6.1.8 shows the errors of edges or curves. LMSE measures the intuitive image qualities because the values of LMSE are well distributed from '0' to '1' unlike NMSE.

Table 6.2 summarizes the correlation coefficients between objective and subjective qualities. Subjective qualities of the compressed image of Barbara were judged by 30 human observers using the 5 subjective rating scales of Table 6.1, and MOS is calculated. The correlations between each of 5 MOS based on 5 rating scales and each of 10 objective qualities are calculated by the *pearson product moment correlation coefficient* method. The average correlations of 5 rating scales are calculated to each of 10 objective quality measures and written at the bottom lines of Table 6.2. Horizontal averages, average correlations of 10 objective qualities to each of 5 rating scales, are calculated and written at the very right sides of Table 6.2.

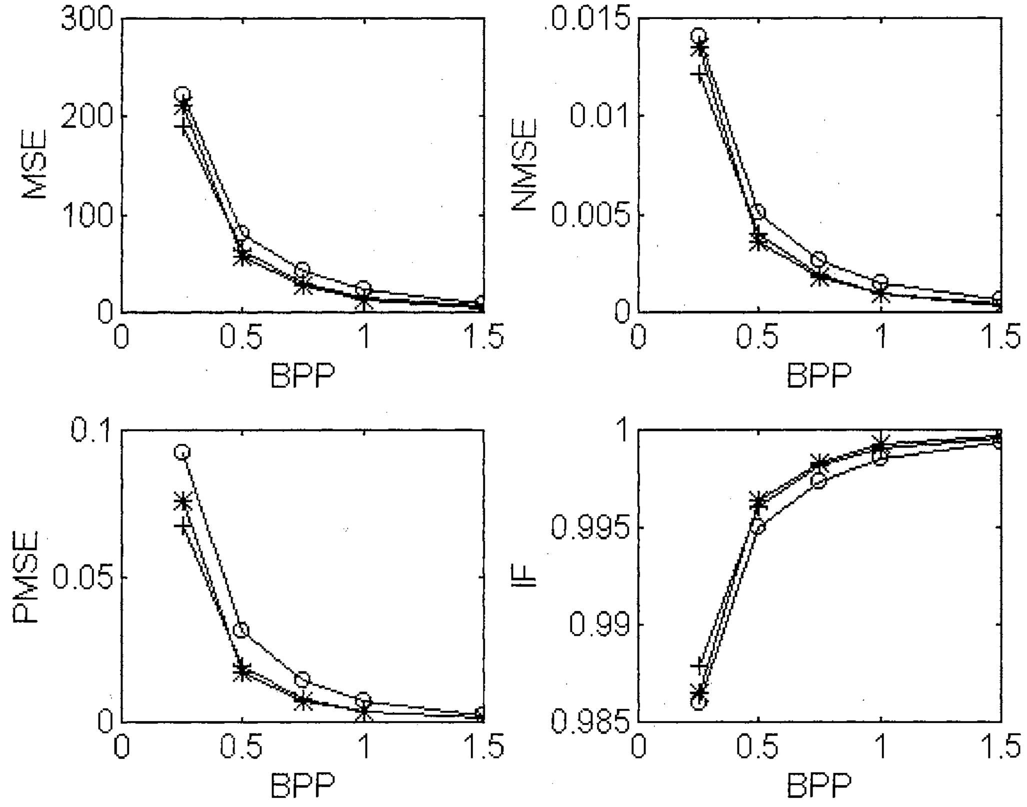


Figure 6.1 Several Objective Performances for the Test Image Barbara
(Top to bottom and left to right: Figures 6.1.1, 6.1.2, 6.1.3, and 6.1.4;
* :DHC, +: MR-ICS-JPEG, and o :STD-JPEG)

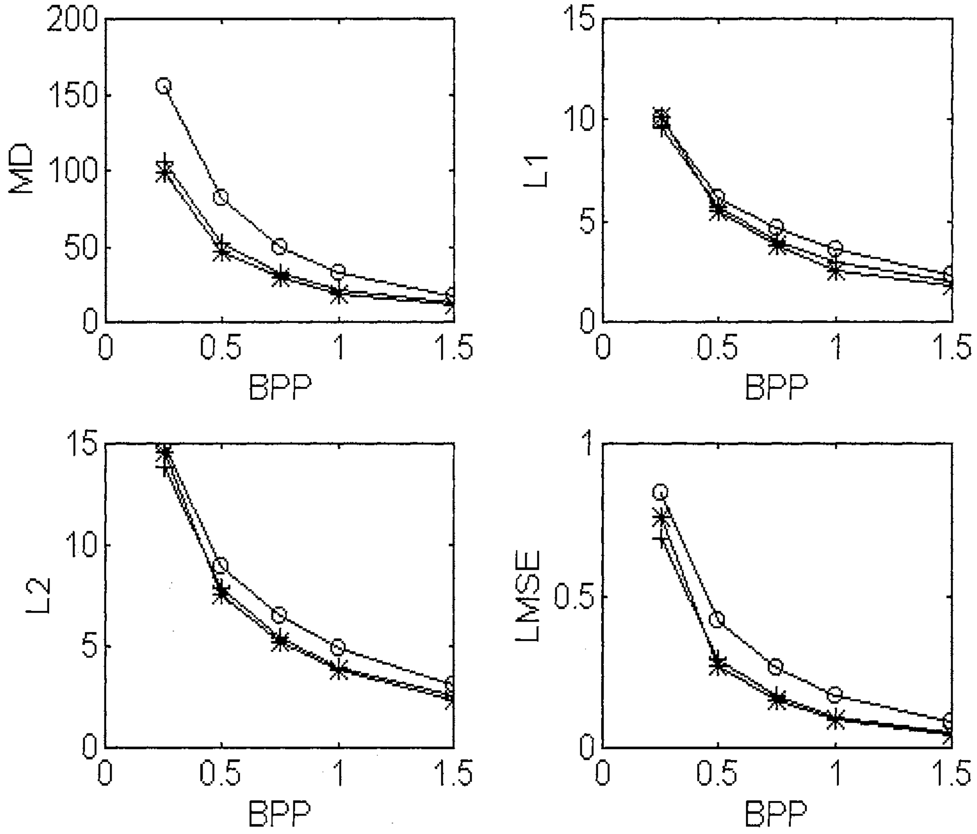


Figure 6.1 Several Objective Performances for the Test Image Barbara
(Top to bottom and left to right: Figures 6.1.5, 6.1.6, 6.1.7, and 6.1.8;
* :DHC, +: MR-ICS-JPEG, and o :STD-JPEG)

As can be seen from Table 6.2, the highest subjective opinion score is achieved by the DHC. Next rankings are followed by MR-ICS-JPEG, and STD-JPEG, and HVQ in the order of their listing. Usually, the human eye is less sensitive to the defects in backgrounds while it is more sensitive to the distortions in the highly active areas such as edges or curves. The strategy of DHC is the additional bit allocation to high detail regions while low detail regions are coded roughly by AVQ. This is the main source to improve the subjective qualities of images which are compressed by the DHC.

Dual Hybrid Coder (Total subjective score: 3,725, Rank: 1st)

OB SB	*MSE*	*NMSE*	*PMSE*	*IF*	*PSNR*	*MD*
A	-0.9261	-0.9261	-0.9119	0.9261	0.9932	-0.9501
B	-0.9286	-0.9287	-0.9041	0.9287	0.9933	-0.9521
C	-0.8967	-0.8967	-0.9081	0.8967	0.9944	-0.9314
D	-0.9150	-0.9150	-0.9015	0.9150	0.9892	-0.9393
E	-0.9270	-0.9270	-0.9129	0.9270	0.9979	-0.9566
MEAN of OB	*-0.9187*	*-0.9187*	*-0.9077*	*0.9187*	*0.9936*	*-0.9459*

OB SB	*NK*	*L1*	*L2*	*LMSE*	*Absolute MEAN of SB*
A	0.9295	-0.9744	-0.9770	-0.9901	*0.9505*
B	0.9319	-0.9759	-0.9783	-0.9905	*0.9512*
C	0.8994	-0.9573	-0.9611	-0.9810	*0.9323*
D	0.9195	-0.9665	-0.9696	-0.9857	*0.9416*
E	0.9285	-0.9764	-0.9792	-0.9921	*0.9525*
MEAN of OB	*0.9218*	*-0.9701*	*-0.9731*	*-0.9879*	

MR-ICS-JPEG (Total subjective score : 3,535, Rank : 2nd)

OB SB	*MSE*	*NMSE*	*PMSE*	*IF*	*PSNR*	*MD*
A	-0.9146	-0.9146	-0.9057	0.9146	0.9989	-0.9485
B	-0.9339	-0.9339	-0.9222	0.9339	0.9891	-0.9402
C	-0.9119	-0.9119	-0.8997	0.9119	0.9991	-0.9285
D	-0.8955	-0.8955	-0.8833	0.8955	0.9986	-0.9235
E	-0.9167	-0.9167	-0.9058	0.9167	0.9994	-0.9394
MEAN of OB	*-0.9145*	*-0.9145*	*-0.9033*	*0.9145*	*0.9970*	*-0.9360*

OB SB	*NK*	*L1*	*L2*	*LMSE*	*Absolute MEAN of SB*
A	0.8881	-0.9592	-0.9645	-0.9470	*0.9356*
B	0.9112	-0.9716	-0.9739	-0.9624	*0.9472*
C	0.8835	-0.9594	-0.9647	-0.9487	*0.9319*
D	0.8653	-0.9475	-0.9535	-0.9349	*0.9193*
E	0.8898	-0.9621	-0.9669	-0.9508	*0.9364*
MEAN of OB	*0.8876*	*-0.9591*	*-0.9647*	*-0.9488*	

STANDARD-JPEG (Total subjective score: 3,509, Rank : 3rd)

OB *SB*	*MSE*	*NMSE*	*PMSE*	*IF*	*PSNR*	*MD*
A	-0.8691	-0.8691	-0.8589	0.8691	0.9940	-0.9203
B	-0.8738	-0.8738	-0.8631	0.8738	0.9930	-0.9235
C	-0.8338	-0.8338	-0.8176	0.8338	0.9840	-0.8707
D	-0.8440	-0.8440	-0.8305	0.8440	0.9883	-0.8897
E	-0.8411	-0.8411	-0.8274	0.8411	0.9876	-0.8892
MEAN *of OB*	*-0.8524*	*-0.8524*	*-0.8395*	*0.8524*	*0.9894*	*-0.8987*

OB SB	*NK*	*L1*	*L2*	*LMSE*	*Absolute MEAN* *of SB*
A	0.8369	-0.9262	-0.9339	-0.9180	*0.8988*
B	0.8429	-0.9294	-0.9359	-0.9138	*0.9023*
C	0.7963	-0.9015	-0.9096	-0.8858	*0.8667*
D	0.8080	-0.9085	-0.9169	-0.8926	*0.8766*
E	0.8051	-0.9063	-0.9145	-0.8899	*0.8743*
MEAN *of OB*	*0.8178*	*-0.9144*	*-0.9222*	*-0.8986*	

HVQ (Total subjective score : 2,229, Rank : 4th)

OB *SB*	*MSE*	*NMSE*	*PMSE*	*IF*	*PSNR*	*MD*
A	-0.8848	-0.8848	-0.7781	0.8848	0.9703	-0.7613
B	-0.8830	-0.8830	-0.7767	0.8830	0.9676	0.7582
C	-0.8596	-0.8596	-0.7450	0.8596	0.9560	-0.7269
D	-0.8365	-0.8365	-0.7156	0.8365	0.9424	-0.6977
E	-0.8511	-0.8511	-0.7338	0.8511	0.9516	-0.7167
MEAN *of OB*	*-0.8630*	*-0.8630*	*-0.7498*	*0.8680*	*0.9576*	*-0.7321*

OB *SB*	*NK*	*L1*	*L2*	*LMSE*	*Absolute MEAN* *of SB*
A	0.8500	-0.9624	-0.9268	-0.9346	*0.8838*
B	0.8460	-0.9612	-0.9256	-0.9385	*0.8823*
C	0.8197	-0.9477	-0.9068	-0.9219	*0.8603*
D	0.7986	-0.9310	-0.8861	-0.8930	*0.8374*
E	0.8104	-0.9425	-0.8998	-0.9153	*0.8523*
MEAN *of OB*	*0.8249*	*-0.9490*	*-0.9090*	*-0.9207*	

Table 6.2 Correlations Between Objective and Subjective Quality Measures
(SB : Subjective Quality Measures and OB: Objective Quality Measures)

From the correlations of Table 6.2, we know that PSNR, AD (L_1), L_2, and LMSE are relatively subjectively meaningful objective quality measures because they produce relatively higher correlation coefficients. Especially, the most commonly used objective quality measure, PSNR, produces the highest correlation coefficients. Therefore, PSNR is still the best objective quality measure even though it is not always the best at representing image qualities in every application. Attempting to produce more perceptually meaningful results with PMSE by emphasizing large error is not successful for all compression schemes of this simulation. It produces lower correlation coefficients than MSE. The correlation coefficients of NMSE and IF are same as MSE because they are just another expressions of MSE.

Ranking Quality	1	2	3	4
Objective Quality	*Dual Hybrid Coder*	*MR-ICS-JPEG*	*STD-JPEG*	*HVQ*
Subjective Quality	*Dual Hybrid Coder* (3,725)	*MR-ICS-JPEG* (3,535)	*STD-JPEG* (3,509)	*HVQ* (2,229)

Table 6.3 Objective and Subjective Ranking of Different Compression Schemes
(numbers in parentheses are total subjective opinion scores obtained from 5 rating scales)

The correlation of LMSE is relatively high in most compression schemes because the detection of edge errors by the Laplacian operator makes this objective quality measure more subjectively meaningful. Table 6.3 describes the performance rankings, objective rankings in terms of PSNR and subjective rankings in terms of the total subjective scores obtained, of different image compression schemes. As we expected, the performances of objective quality measures are not exactly related to the visual subjective quality. For example, the average difference of performance in terms of PSNR between DHC and MR-ICS-JPEG is about 0.5 dB in the overall range of bit rates. The difference of total subjective score achieved by two compression schemes is 190. However the difference of total subjective score achieved by MR-ICS-JPEG and STD-JPEG is only 26 even though the average difference of objective qualities in terms of PSNR between two schemes is more than 2 dB.

CHAPTER VII

SUMMARY AND CONCLUSION

Several state of the art compression techniques which result in high quality images are proposed. Two independent experiments are performed in Chapters 3 and 4. The new quantization table using the ICS method is introduced and the MR scheme is applied to both ICS-QT and standard JPEG. The MR-ICS-JPEG and the MR-JPEG methods produce more than 2dB and 0.5 dB improved performances when compared to that of standard JPEG, respectively. The five different regression equations which are adjusted with the ICS-QT are presented. These equations will not fit well when we employ other quantization tables; however, there is value in combining ICS-QT and its corresponding linear regression equations because ICS-QT gives enhanced performance, in terms of PSNR, when compared to standard JPEG. Using *PI* control with the linear regression equations, the desired constant bit rates were produced within no more than about four iterations.

In Chapter 4, two optimal thresholding methods, *single-thresholding (ST)* and *multi-thresholding (MT),* are proposed to efficiently decompose an image using the quadtree segmentation method. These optimal thresholding methods for the quadtree decomposition are a basis for implementing hierarchical vector quantization. The ST method, based on the optimal *polynomial*, produces a slightly better performance than that of the MT method in terms of PSNR over a narrow optimal range; however, it loses its optimal property outside of that range. On the other hand, the MT method, based on a mathematical model, keeps its optimal property at most ranges of bit rates. As a result, it produces better performance (more than 0.5 dB) in terms of PSNR at most ranges of bit rates outside of the narrow optimal range of the ST method. Therefore, we can conclude that the MT method is a better choice for optimal thresholding.

A *homogeneity* test based on *spatial frequency (SPF)* is performed for the blocks decomposed by the proposed optimal quadtree segmentation method. The performance of this test method is measured by the *Student's t-test* in order to classify it into '*Success (S)*'or '*Fail (F)*'. The homogeneity increases monotonically when the bit rate increases. The transition point from '*F*' to '*S*' of the homogeneity based on the Student's t-test is located near the bit rate which corresponds to the optimal threshold of the ST method. Therefore, the optimality, in the sense of 'trading off' between bit rate and MSE, of the optimal single threshold is verified in terms of the homogeneity.

HVQ based on the proposed optimal quadtree decomposition method is a very efficient digital compression technique because codebooks which have more quantization levels can be used at the same bit rate for representing high detail regions. This strategy based on the optimal thresholding methods produced more than 5 dB improved performances in terms of PSNR when compared to that of fixed block size normal VQ.

Chapter 5 is the primary focus of this thesis. Vector quantization and transform coding are fundamentally different coding schemes. Vector quantization attempts to code groups of parameters together. Therefore, we can get high compression with this method. On the other hand, DCT based transform coding is a scalar based coding method and results in high image quality. However, it is hard to get high compression with this method. The advantage of VQ, high compression, and the advantage of DCT based transform coder, high quality, are combined and result in a new *hybrid coder,* which is referred to as the *dual hybrid coder (DHC).*

Each part of transform coding and VQ of the DHC is implemented by MR-ICS-JPEG and AVQ where codebooks are generated by homogeneous training vectors, respectively. Low detail 8x8 blocks are classified by their homogeneities and coded by AVQ. In this procedure, we can get high compression. Some additional bits which are reduced by AVQ are allocated to the MR-ICS-JPEG to represent high detail regions more accurately. This strategy allows the DHC to produce high quality images subjectively and objectively. The DHC produces about 3.5 dB and 2.5 dB improved objective performances in terms of PSNR over the standard JPEG for the images inside and outside of the training vectors, respectively, when the bit rates are larger than about 0.30 bpp. On the other hand, the performance of the proposed DHC is worse than that of the proposed HVQ when the bit rate is less than about 0.30 bpp. This problem is a result of the additional 4 bits which are necessary to represent the EOB as is common with transform based coder such as JPEG. Because of this, the proposed DHC is switched to HVQ at bit rates less than about 0.30 bpp. Therefore, the proposed hybrid scheme is called the *dual hybrid coder*. However, this bit rate switching point will be slightly different from image to image. The linear regression equation, given in Figure 5.6, estimates the bit rate switching point for a given image as a function of the average block homogeneity.

The DHC obtained the highest subjective score which was judged by 30 human observers using 5 subjective rating scales. The main source to enhance the subjective quality of the DHC is some additional bit allocation to the high detail regions, because the human eye is less sensitive to the low detail regions such as backgrounds while more sensitive to high detail regions such as edges or curves.

The first advantage of the proposed DHC is its adaptivity. In other words, low detail regions are coded by codebook adaptive VQ according to their homogeneities, and high detail regions are coded by MR-ICS-JPEG with some additional bit allocations. The other advantage of this scheme is its low overhead in bits. Usually, hybrid coders [Ngan, 1991; Wen,1993] employ classified VQ based on edge classification. Because so many edge pattens exist, more than 5 bits are necessary as overhead. On the other hand, the proposed hybrid scheme restricts its overhead to no more than 3 bits. Even though a maximum 3 bits of overhead for classifying image blocks is employed, the adaptive homogeneous codebooks adequately represent input image blocks because only low detail (highly homogeneous) regions are coded by VQ.

The correlation coefficients between objective and subjective qualities represent the subjectivity of objective quality measures. The highest correlation is achieved by PSNR. Therefore, PSNR is still the best objective quality measure even though it is not a panacea for every image processing application. Another possible candidate as a reasonable objective quality measure is LMSE because its value is well distributed from '0' to '1' and gives us intuitive image quality. In addition, it produces a relatively high correlation coefficient as well as representing the errors of edges or curves to which the human eye is more sensitive.

Future Work

There are three areas of suggested future work: estimate more accurate initial scaling factors for calibrating bit rate, develop an efficient codebook design algorithm, and find the optimal threshold to decide the optimal percentage of VQ coded area of the DHC.

The linear regression equations generated in chapter III approximate the initial scaling factor for the quantization table to produce disired bit rates within about four iterations. In order to calculate these equations, the least square regression method, which minimizes the average distance between fitted lines and actual data, is used. In this procedure, some of the outliers are weighted more heavily than desired. If we can generate a higher order polynomial which is a better fit to the actual data instead of using the first order method, the number of iterations which is required to converge to a fixed bit rate will be reduced. Therefore, more accurate initializers are necessary.

The LBG algorithm has been a conerstone to generate vector codebooks. However, this algorithm uses the training approach for constructing a codebook which requires a long codebook generation time. Although current computer technology allows one to calculate huge amounts of data, the long codebook generation time has been an obstacle of VQ. If we develop an analytical method which is not using the training approach to generate a codebook, the VQ technique will be more feasible for practical applications.

An optimal threshold to decide the optimal percentage of VQ coded area in DHC affects the overall compression performance. The optimal threshold is different from image to image because data distributions are unpredictable. Therefore, this threshold needs to be adjusted according to the data distribution of a given image. This area still has room for further deveopment.

BIBLIOGRAPHY

Auslander D.M and Tham C.H., “Real time Software for Control”, Pages 41-42, prentice- Hall, 1990.

Bage M. J., “ Interframe Predictive Codig of Images Using Hybrid Vector Quantization”, IEEE Transactions on Communications, Vol. COM-34, No. 4, Pages 411-415, April 1986.

Bajpai A. C., Calus I. M., and Fairley J. A., “Statistical Methods for Engineers and Scientists”, Page 370, New York: Wiley, 1979.

Barnes C. F. and Frost R. L., “ Necessary Conditions for the Optimality of Residual Vector Quantizers”, In Abstracts of the 1990 IEEE International Symposium on Information Theory, Page 34, San Diego, CA., Jan. 1990.

Bhaskaram V. and Konstantinides K., “Image and Video Compression Standard Algorithms and Architectures”, Kluwer Academic Publishers, Mass., 1995.

Chan C. K. and Ma C. K., “ A Fast Method of Designing Better Codebooks for Image Vector Quantization”, IEEE Transactions on Communications, Vol. 42, No. 2/3/4, Pages 237 - 242, Feb. /Mar. /Apr. 1994.

Chen C. F. and Pang K. K., “ The Optimal Transform of Motion-Compensated Frame Difference Images in a Hybrid Coder”, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 40, No. 6, Pages 393-397, June 1993.

Chen S. H. and Hsieh W. M., “ Fast Algorithm for VQ Codebook Design”, IEE Proceedings-I, Vol. 138, No. 5, Pages 357-362, OCT. 1991.

Cheng D.Y. and Gersho A., “A Fast Codebook Search Algorithm for Nearest-Neighbor Pattern Matching”, Proceedings of ICASSP, Tokyo, Pages 6.14.1-6.14.4, 1986.

Cheng D.Y., Gersho A., Ramamurthi B., and Shoham Y., “Fast Search Algorithms for Vector Quantization and Pattern Matching”, Proceedings of ICASSP, Pages 9.11.1-9.11.4, 1984.

Chien C. H. and Aggarwal J. K., “A Normalized Quadtree Representation”, Computer Vision, Graphics, Image Processing, Vol. 26, 1984.

Chiu C.-Y. and Baker R. L., “Quad-Tree Product Vector Quantization of Images”, in Proceedings of SPIE Conference, Advances Image Compression Automat. Target Recognition, Vol. 1099, Pages 142-153, Mar. 1989.

Clarke R. J., “Digital Compression of Still Images and Video”, Academic Press, London, Page 90, 1995.

Clarke R. J., “ Orthogonal Transforms for Image Coding” in Transform Coding of Images, Pages 72-133, Academic Press, London, 1985.

Clarke R. J., “ Transform Coding of Images”, Page 4, Academic Press, London, 1985.

Dasarathy B. V., “ Image Data Compression”, Page 2, IEEE Computer Society Press, Los Alamitos, 1995.

Ekstrom M. P., “Digital Image Processing Techniques”, Pages 264-265, Academic Press, Inc., 1984.

Equitz W. H., “ A New Vector Quantization Clustering Algorithm”, IEEE Transactions on ASSP., Vol.37, No. 10, Pages 1568-1575, Oct. 1989.

Equitz W. H., “ Fast Algorithms for Vector Quantization Picture Coding”, Master’s thesis, M. I. T., Cambridge, June 1984.

Eskicioglu A.M. and Fisher P.S., “ A Survey of Quality measures for gray Scale Image Compression”, in Proc. 1993 Space and Earth Science Data Compression Workshop (NASA Conference Publication 3191), Pages 49-61, Snowbird, Utah, Apr. 2, 1993.

Eskicioglu A. M. and Fisher P.S., “Image Quality Measures and Their performance”, IEEE Transactions on Communications, Vol. 43, No.12, Pages 2959-2965, Dec. 1995.

Eskicioglu A. M. and Fisher P.S., “ Image Quality Measures and Their performance”, in Proc. 1992 Space and Earth Science Data Compression Workshop, Pages 55-67, Snowbird, Utah, 1992.

Flanagan J. K., Morrell D.R., Frost R.L., Read C. J., and Nelson B. E., “ Vector Quantization Codebook Generation Using Simulated Annealing”, Proc. IEEE ICASSP, Vol.3, Pages 1759-1762, 1989.

Frost R. L., Barnes C. F., and Xu F., “Design and Performance of Residual Quantizers”, In J. A. Storer and J. H. Reif., Editors, Proceedings Data Compression Conference, IEEE Computer Society Press, Pages 129-138, Snowbird, Utah, Apr. 1991.

Gersho A., “ On the Structure of Vector Quantizers”, IEEE Transactions on Information Theory, Vol. IT.-28, No.2, Pages 157-166, Mar. 1982.

Gersho A. and Cuperman V., “ Vector Quantization: A Pattern Matching Technique for Speech Processing”, IEEE Communication Magazine, Pages 15-21, Dec. 1983.

Gersho A. and Gray R.M., “ Vector Quantization and Signal Compression”, Kluwer Academic Publisher, Mass., Page 366, 1992.

Gersho A. and Gray R.M., “Vector Quantization and Signal Compression”, Kluwer Academic Publishers, Boston, 1991.

Ghafourian M. A. and Huang C. H., “Comparison between Several Adaptive Search Vector Quantization Schemes and Standard for Image Compression”, IEEE Transactions on Communications, Vol. 43, No. 2/3/4, Pages 1308-1312, Feb./Mar./Apr. 1995.

Gonzalez R. C. and Wintz P., “Image Encoding” in Digital Image Processing, Page 229, Addison - Wesley Publishing Co. Inc., 1977.

Gray R.M., “Vector Quantization”, IEEE ASSP Magazine, Pages 4-28, Apr. 1984.

Grosky W. I. and Jain R., “Optimal Quadtrees for Image Segmentations”, IEEE Transactions on Pattern Analysis an Machine Intelligence, Vol. PAMI-5, No. 1, Pages 77-83, Jan. 1983.

Habibi A., “ Hybrid coding of pictorial data”, IEEE Transactions on Communications, COM-22, Pages 614-626, May 1974.

Hanan H. S., “ The Quadtree and Related Hierarchical Data Structures”, Computing Surveys, Vol. 16, No. 2, Pages 187-260, June 1984.

Jackins C. L. and Tanimoto S. L., “Quad-Tree, and k-Trees: A generalized Approach to Recursive Decomposition of Euclidean Space”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAM-5, No. 5, Pages 533-539, Sept. 1983.

Jain A.K., “Fundamentals of Digital Image Compression”, Pages 483-495, Prentice-Hall, 1989.

Jain A. K., “ Some New Techniques in Image Processing” in Image Science Mathematics, Pages 201-223, Western Period, North Hollywood, CA., 1976.

Juang B. H. and Gray A. H., “Multiple Stage Vector Quantization for Speech Coding”, In International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, Pages 597-600, Paris, Apr. 1982.

Kiang S. Z.. Baker R. L., Sullivan G. J., and Chiu C. Y., “ Recursive Optimal Pruning with Applications to Tree Structured Vector Quantizers”, IEEE Transactions on Image Processing, Vol.1, No. 2, Pages 162-169, Apr. 1992.

Kim D. S. and Lee S. U., “Image Vector Quantizer Based on Classification In the DCT domain”, IEEE Transactions on Communications, Vol. 39, No. 4, Pages 549-556, Apr. 1991.

Kim J. W. and Lee S. U., “ A Transform Domain Classified Vector Quantizer for Image Coding”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 2, No. 1, Pages 3-14, Mar. 1992.

Kim S. H. and Rao K. R., “ Pyramid VQ for Image Coding using DCT classifier based on HVS”, Proceedings of ICIP-92, Pages 151-155, Singapore, Sept. 1992.

Kirkpatrick S., Gellatt C. D., Jr., and Vecchi M. P., “ Optimization by Simulated Annealing”, Science, Vol. 220, Pages 671-680, May. 1983.

Lee C.H. and Chen L. H., “Fast Codeword Search Algorithm for Vector Quantization”, IEE Proceedings Vision Image Signal Processing, Vol. 141, No.3, June 1994.

Lee M. H. and Crebbin G., “ Classified Vector Quantization with Variable Block-Size DCT Models”, IEE Proc. Vision and Image Processing, Vol. 141, No. 1, Pages 39-48, Feb. 1994.

Leger A., Omachi T. and Wallace G.K., “JPEG Still Picture Compression Algorithm”, Optical Engineering, Vol.30, No.7, Pages 947-954, July 1991.

Lim J. S., “ Two dimensional Signal and Image Processing”, Pages 140 - 157, Prentice- Hall, New Jersey, 1990.

Linde Y., Buzo A., and Gray R. M., “ An Algorithm for Vector Quantizer Design”, IEEE Transactions on Communications, Vol. COM.-28, No.1, Pages 84-95, Jan.

1980.

Lloyd, S. P., " Least Squares Quantization in PCM's", Bell Telephone Laboratories Paper, Murray Hill, NJ., 1957.

Lohscheller H., "Subjectively Adapted Image Communication System", IEEE Transactions on Communications, COM-32(12), Pages 1316 - 1322, Dec. 1984.

Lowry A., Hassain S., and Miller W., "Binary Search Trees for Vector Quantization", Proceedings of ICASSP, Pages 51.8.1-51.8.4, 1987.

MacQueen J., " Some methods for classification and analysis of multivariate observations". In Proc. of the Fifth Berkeley Symposium on Math. Stat. and Prob., Vol.1, Pages 281-296, 1967.

Menke W., "Geographical data analysis: discrete inverse theory", Academic Press, Orlando, 1984.

Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., and Teller E., " Equations of State Calculations by Fast Computing Machines", J. Chem. Phys., Vol. 21, Pages 1087- 1091, 1953.

Nakagawa M. and Sasaki M., "DCT-Based Still Image Compression ICS with Bit-Rate Control", IEEE Transactions on Consumer Electronic , Pages 711-716, Vol.38, No.3, Aug. 1992.

Nasrabadi N.M. and King R.A., "Image Coding Using Vector Quantization: A Review", IEEE Transactions on Communications, Vol.36, No.8, Pages 957-971, Aug. 1988.

Nasrabadi N. M., Lin Se. e., and Feng Y., "Interframe Hierarchical Vector Quantization", Optical Engineering, Vol. 28, No. 7, Pages 717-725, July 1989.

Netravali and Haskell, "Digital Pictures", Page 257, AT&T Bell Laboratories, Plenum Press, New York, 1995.

Netravali A. N. and Limb J. O., " Picture Coding : A Review", Proceedings of the IEEE, Vol. 68, No. 3, Pages 336 - 407, Mar. 1980.

Ngan K. N., Koh H. L., and Wong W. C., " Hybrid Image Coding Scheme incorporating Human Visual System Characteristics", Optical Engineering, Vol. 30, No.7, Pages 940-946, July 1991.

Ohta M. and Nogaki S., " Hybrid Picture Coding with Wavelet Transform and overlapped Motion-Compensated Interframe Prediction Coding", IEEE

Transactions on Signal Processing, Vol. 41, No. 12, Pages 3416-3424, Dec. 1993.

Paliwal K. K. and Ramasubramanian V., "Effect of Ordering the Codebook on the Efficiency of the Partial Distance Search Algorithm for Vector Quantization", IEEE Transactions on Communications, Vol. 37, No. 5, Pages 528-540, May 1989.

Pennebaker W. B. and Mitchell J. L., "JPEG Still Image Compression Standard", Van Nostrand, New York, 1993.

Pratt W. K., "Digital Image Processing", John Wiley and Sons, Inc., USA, 1978.

Pratt W. K., 'Digital Image Processing,' Page 685, Wiley Interscience, 1991.

Ramamurth B. and Gersho A., " Classified Vector Quantization of Images", IEEE Transactions on Communications, Vol. Com-34, No. 11, Pages 1105-1115, Nov. 1986.

Riskin E. A. and Gray R. M., " A Greedy Tree Growing Algorithm for the Design of Variable Rate Vector Quantizers', IEEE Transactions on Signal Processing, Vol. 39, No. 11, Pages 255-2507, Nov. 1991.

Samet H., " The Quadtree and Related Hierarchical Data Structures", Computing Survey, Vol. 16, No. 2, Pages 187-260, June 1984.

Sasaki M. and Kimisima T., "Bit-Rate Control for JPEG Coding", Proc. of the 1992 ITE Convention 16 - 15, Pages 323 - 324, 1992.

Shaffer C. A. and Samet H., "Optimal Quadtree Construction Algorithms", Computer Vision, Graphics, and Image Processing , Vol. 37, Pages 402-419, 1987.

Shannon C. E., "The mathematical Theory of Communication", Urbana: University of Illinois Press, 1978.

Shusterman E. and Feder M., "Image Compression via Improved Quadtree Decomposition Algorithms", IEEE Transactions on Image Processing, Vol. 3, No. 2, Pages 207-215, Mar. 1994.

Sitaram V. S., Huang C. M., and Israelsen P. D., "Efficient Codebooks for Vector Quantization Image Compression with an Adaptive Tree Search Algorithm", IEEE Transactions on Communications, Vol. 42, No. 11, Nov. 1994.

Soleymani M. R. and Morgera S. D., "A High Speed Search Algorithm for Vector

Quantization", Proceedings of ICASSP, Pages 45.6.1-45.6.3, 1987.

Soleymani M. R. and Morgera S. D., "An Efficient Nearest Neighbor Search Method", IEEE Transactions on Communications, Vol. COM-35, No. 6, Pages 677-679, June 1987.

Steel R. G. D. and Torrie J. H., "Comparisons Involving Two Sample Means", in Principles and Procedures of Statistics, Pages 86-121, Mc Graw-Hill, Inc., 1980.

Steel R.G.D. and Torrie J.H., "Principles and Procedures of Statistics", Pages 17-19, Pages 239-256, and Pages 244-252, Mc Graw Hill, 1980.

Strobach P., "Image Coding Based on Quadtree-Structured Recursive Least-Squares Approximation", Proceedings of IEEE ICASSP, Pages 1961 - 1964, 1989.

Strobach P., "Quadtree-Structured Linear Prediction Models for Image Sequence Processing", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, Pages 742-748, July 1989.

Strobach P., "Quadtree-Structured Recursive Plane Decomposition Coding of Image", IEEE Transactions on Signal Processing, Vol. 39, Pages 1380-1397, June 1991.

Strobach P., "Tree-Structured Scene Adaptive Coder", IEEE Transactions on Communications, Vol. 38, Pages 477-486, Apr. 1990.

Sallivan G. J. and Baker R. L., "Efficient Quadtree Coding of Image and Video", IEEE Transactions on Image Processing, Vol. 3, No. 3, Pages 327-331, May 1994.

Supangkat S.H. and Murakami K., "Quantity control for JPEG image Data Compression using Fuzzy Logic Algorithm", IEEE Transactions on Consumer Electronics, Vol.41, No.1, Pages 42-48, Feb. 1995.

Tannas I. and Lawrence E., "Image Quality : Measures and Visual Performance" in Flat-Panel Displays and CRTs, Van Nostrand Reinhold, 1985.

Ting M. Y. and Riskin E. A., "Error-Diffused Image Compression Using a Binary- to-Gray-Scale Decoder and Predictive Pruned Tree-Structured Vector Quantization", IEEE Transactions on Image Processing, Vol.3, No. 6, Pages 854-858, Nov. 1994.

Torres L. and Huguet J., "An Improvement on Codebook Search for Vector

Quantization", IEEE Transactions on Communications, Vol. 42, No. 2/3/4, Pages 208-210, Feb./ Mar./ Apr., 1994.

Vaisey J. and Gersho A., "Image Compression with variable block size segmentation", IEEE Transactions on Signal Processing, Vol. 40, Pages 2040-2060, Aug. 1992.

Vaisey J. and Gersho A., "Simulated Annealing and Codebook Design", Proc. IEE ICASSP., Vol.3, Pages 1176-1179, Apr. 1989.

Vaisey J. and Gersho A., "Variable block size Image Coding", in Proceedings of IEEE International Conference Acoust. Speech Signal Processing (ICASSP), Pages 25.1.1-25.1.4, Apr. 1987.

Wallace G. K., "The JPEG Still Picture Compression Standard", Communication of ACM., Vol. 34, No. 4, Pages 31-44, April 1991.

Wallace G. K., "The JPEG Still Picture Compression Standard", IEEE Transactions on Consumer Electronics. Vol. 38, No. 1, Pages xviii-xxxiv, Feb. 1992.

Wen K. A. and Lu C. Y., " Hybrid Vector Quantization", Optical Engineering, Vol. 32, No. 7, Pages 1496-1502, July 1993.

William A. P. and Priyadarshan J., " The Effectiveness and Efficiency of Hybrid Transform/DPCM Interframe Image Coding", IEEE Transactions on Communications, Vol. COM-32, No. 7, Pages 832-838, July 1984.

Wu C.J. and Sung A.H., "The Application of Fuzzy Logic to JPEG", IEEE Transactions on Consumer Electronics, Vol.40, No.4, pp 976-984, Nov. 1994.

Wu S. W., "Improved Decoder for Transform Coding with Application to the JPEG Baseline System", IEEE Transactions on communications, Vol. 40, No. 2, Pages 251-254, Feb. 1992.

APPENDIX II

APPENDIX II.1

An Example of LBG Algorithm (1-D)

(0) Initialization : N=4, k=2, Th=0.001, and n=12.

Training Sequence (Table A.II.1.1)

X_1 = (-0.37449, 0.98719)	X_7 = (-0.59161, 0.17968)
X_2 = (0.63917, -0.11875)	X_8 = (0.14093, 1.76413)
X_3 = (-0.83293, 0.60654)	X_9 = (0.70898, -0.35017)
X_4 = (-0.70534, -1.12186)	X_{10}= (0.30038, 0.79836)
X_5 = (-0.28952, -0.94821)	X_{11}= (0.30165, 1.06552)
X_6 = (1.09924, 0.51600)	X_{12}= (0.37801, -0.32708)

$C_0 = [(2,2), (2, -2), (-2, 2), (-2,-2)] = [y_1, y_2, y_3, y_4]$

$(1/D)=9.99E+62$ (∞ on a micro computer), Set $m=0$.

$m=0$ (1) Find $P(\hat{C}_0) = [S_1, S_2, S_3, S_4]$

$X_j \in S_i$ if $d(X_j, y_i) \leq d(X_j,y_m)$, for all m

$S_1 = [X_6, X_8, X_{10}, X_{11}]$

$S_2 = [X_2, X_9]$

$S_3 = [X_1, X_3, X_7]$

$S_4 = [X_4, X_5, X_{12}]$

Compute D_0

$$D_0 = \frac{1}{12}\sum_{j=1}^{12} \min d(X_j, y) = 2.0172$$

(2) $((1/D)-D_0)/D_0 > 0.001$, continue

(3) Find the optimal reproduction alphabet:

$$\hat{C}_1 \cong X(P(\hat{C}_0)) = [\hat{X}(S_i), i = 1,\ldots,4]$$

$$\hat{X}(S_1) = (X_6 + X_8 + X_{10} + X_{11}) / 4 = (0.46055, 1.036)$$

$$\hat{X}(S_2) = (X_2 + X_9) / 2 = (0.674085, -0.2346)$$

$$\hat{X}(S_3) = (X_1 + X_3 + X_7) / 3 = (-0.5997, 0.59111)$$

$$\hat{X}(S_4) = (X_4 + X_5 + X_{12}) / 3 = (-0.45762, -0.83128)$$

Set $m=1$, Go to (1)

(1) Find $P(\hat{C}_1)$

Evaluating distortion:

$$D_1 = \frac{1}{12}\sum_{j=1}^{12} \min d(x_j, y) = 0.0997308$$

(2) $(D_0 - D_1)/D_1 \cong 19 > 0.001$

(3) $\hat{C}_2 \cong X(P(\hat{C}_1)) = \hat{C}_1$, and

$$\hat{X}(P(\hat{C}_1)) = X(P(\hat{C}_0)) = \hat{C}_1.$$

Thus, $\hat{C}_1$ is a fixed point, set $m=2$. Go to (1)

$m = 2$ (1) $P(\hat{C}_1) = P(\hat{C}_0)$,$D_2 = D_1$, and $(D_1 - D_2)/D_2 = 0 < 0.001$.

Halt with final quantizer described by $[\hat{C}_1, P(\hat{C}_1)]$.

APPENDIX II.2.1

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table A.II.2.1 *Q - Table* for Luminance Included by *JPEG* [Pennebaker, 1993]

APPENDIX II.3

CAT	DPCM Difference	ANB	Additional Bits
0	0	0	-
1	-1,1	1	0,1
2	-3,-2,2,3	2	00,01,10,11
3	-7,...,-4,4,...,7	3	000,...,011,100,...,111
4	-15,.....,-8,8,.....,15	4	0000,...,0111,1000,...,1111
5	-31,......,-16,16,......,31	5	00000,...,01111,10000,...,11111
6	-63,......,-32,32,......,63	6	000000,..., ,...,111111
7	-127,......,-64,64,......,127	7	0000000,..., ,...,1111111
8	-255,......,-128,128,......,255	8	00000000,..., ,...,11111111
9	-511,......,-256,256,......,511	9	000000000,..., ,...,111111111
10	-1023,......,-512,512,......,1023	10	0000000000,..., ,...,1111111111
11	-2047,......,-1024,1024,......,2047	11	00000000000,..., ,...,11111111111

Table A.II.3.1 Additional Bits for Sign and Magnitude for DC Differences
(CAT: Categories, ANB: Additional Number of Bits) [Pennebaker, 1993].

Categories	Code Length	Code Word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Table A.II.3.2 *Huffman* Table for Luminance DC Differences [Pennebaker, 1993]

ANB	AC COEFFICIENTS	ADDITIONAL BITS
1	-1,1	0,1
2	-3,-2,2,3	00,01,10,11
3	-7,...,-4,4,...,7	000,...,011,100,...,111
4	-15,...,-8,8,...15	0000,...,0111,1000,...,1111
5	-31,...,-16,16,...,31	00000,...,01111,10000,...,11111
6	-63,...,-32,32,...,63	000000,..., ,...,111111
7	-127,...,-64,64,...127	0000000,..., ,...,1111111
8	-255,...,-128,128,...,255	00000000,..., ,...,11111111
9	-511,..,-256,256,...,511	000000000,..., ,...,111111111
10	-1023,..,-512,512,...,1023	0000000000,..., ,...,1111111111

Table A.II.3.3 Additional Bits for Sign and Magnitude of AC Coefficient
(ANB : Additional Number of Bits Required) [Pennebaker, 1993].

		ANB										
		0	1	2	3	4	5	6	7	8	9	10
NZR	0	EOB	01	02	03	04	05	06	07	08	09	0A
	1	N/A	11	12	13	14	15	16	17	18	19	1A
	2	N/A	21	22	23	24	25	26	27	28	29	2A
	3	N/A	31	32	33	34	35	36	37	38	39	3A
	4	N/A	41	42	43	44	45	46	47	48	49	4A
	5	N/A	51	52	53	54	55	56	57	58	59	5A
	6	N/A	61	62	63	64	65	66	67	68	69	6A
	7	N/A	71	72	73	74	75	76	77	78	79	7A
	8	N/A	81	82	83	84	85	86	87	88	89	8A
	9	N/A	91	92	93	94	95	96	97	98	99	9A
	10	N/A	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA
	11	N/A	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA
	12	N/A	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA
	13	N/A	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA
	14	N/A	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA
	15	ZRL	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA

Table A.II.3.4 *Huffman* AC Statistical Model for Run-Length / Sign / Magnitude Combination [Penebaker, 1993] (NZR:Number of Zero Run, N/A: Not Applicable for Baseline Algorithm, and ZRL: Zero Run Length).

APPENDIX II.3

Table A.II.3.5 *Huffman* Table for Luminance AC Coefficients [Pennebaker, 1993]

Run Size	Code Length	Code Word	Run Size	Code Length	Code Word
0/0(EOB)	4	1010			
0/1	2	00	4/1	6	111011
0/2	2	01	4/2	10	1111111000
0/3	3	100	4/3	16	1111111110010110
0/4	4	1011	4/4	16	1111111110010111
0/5	5	11010	4/5	16	1111111110011000
0/6	7	1111000	4/6	16	1111111110011001
0/7	8	11111000	4/7	16	1111111110011010
0/8	10	1111110110	4/8	16	1111111110011011
0/9	16	1111111110000010	4/9	16	1111111110011100
0/A	16	1111111110000011	4/A	16	1111111110011101
1/1	4	1100	5/1	7	1111010
1/2	5	11011	5/2	11	11111110111
1/3	7	1111001	5/3	16	1111111110011110
1/4	9	111110110	5/4	16	1111111110011111
1/5	11	11111110110	5/5	16	1111111110100000
1/6	16	1111111110000100	5/6	16	1111111110100001
1/7	16	1111111110000101	5/7	16	1111111110100010
1/8	16	1111111110000110	5/8	16	1111111110100011
1/9	16	1111111110000111	5/9	16	1111111110100100
1/A	16	1111111110001000	5/A	16	1111111110100101
2/1	5	11100	6/1	7	1111011
2/2	8	11111001	6/2	12	111111110110
2/3	10	1111110111	6/3	16	1111111110100110
2/4	12	111111110100	6/4	16	1111111110100111
2/5	16	1111111110001001	6/5	16	1111111110101000
2/6	16	1111111110001010	6/6	16	1111111110101001
2/7	16	1111111110001011	6/7	16	1111111110101010
2/8	16	1111111110001100	6/8	16	1111111110101011
2/9	16	1111111110001101	6/9	16	1111111110101100
2/A	16	1111111110001110	6/A	16	1111111110101101
3/1	6	111010	7/1	8	11111010
3/2	9	111110111	7/2	12	111111110111
3/3	12	111111110101	7/3	16	1111111110101110
3/4	16	1111111110001111	7/4	16	1111111110101111
3/5	16	1111111110010000	7/5	16	1111111110110000
3/6	16	1111111110010001	7/6	16	1111111110110001
3/7	16	1111111110010010	7/7	16	1111111110110010
3/8	16	1111111110010011	7/8	16	1111111110110011
3/9	16	1111111110010100	7/9	16	1111111110110100
3/A	16	1111111110010101	7/A	16	1111111110110101

Run Size	Code Length	Code Word	Run Size	Code Length	Code Word
8/1	9	111111000	C/1	10	1111111010
8/2	15	111111111000000	C/2	16	1111111111011001
8/3	16	1111111110110110	C/3	16	1111111111011010
8/4	16	1111111110110111	C/4	16	1111111111011011
8/5	16	1111111110111000	C/5	16	1111111111011100
8/6	16	1111111110111001	C/6	16	1111111111011101
8/7	16	1111111110111010	C/7	16	1111111111011110
8/8	16	1111111110111011	C/8	16	1111111111011111
8/9	16	1111111110111100	C/9	16	1111111111100000
8/A	16	1111111110111101	C/A	16	1111111111100001
9/1	9	111111001	D/1	11	11111111000
9/2	16	1111111110111110	D/2	16	1111111111100010
9/3	16	1111111110111111	D/3	16	1111111111100011
9/4	16	1111111111000000	D/4	16	1111111111100100
9/5	16	1111111111000001	D/5	16	1111111111100101
9/6	16	1111111111000010	D/6	16	1111111111100110
9/7	16	1111111111000011	D/7	16	1111111111100111
9/8	16	1111111111000100	D/8	16	1111111111101000
9/9	16	1111111111000101	D/9	16	1111111111101001
9/A	16	1111111111000110	D/A	16	1111111111101010
A/1	9	111111010	E/1	16	1111111111101011
A/2	16	1111111111000111	E/2	16	1111111111101100
A/3	16	1111111111001000	E/3	16	1111111111101101
A/4	16	1111111111001001	E/4	16	1111111111101110
A/5	16	1111111111001010	E/5	16	1111111111101111
A/6	16	1111111111001011	E/6	16	1111111111110000
A/7	16	1111111111001100	E/7	16	1111111111110001
A/8	16	1111111111001101	E/8	16	1111111111110010
A/9	16	1111111111001110	E/9	16	1111111111110011
A/A	16	1111111111001111	E/A	16	1111111111110100
B/1	10	1111111001	F/0(ZRL)	11	11111111001
B/2	16	1111111111010000	F/1	16	1111111111110101
B/3	16	1111111111010001	F/2	16	1111111111110110
B/4	16	1111111111010010	F/3	16	1111111111110111
B/5	16	1111111111010011	F/4	16	1111111111111000
B/6	16	1111111111010100	F/5	16	1111111111111001
B/7	16	1111111111010101	F/6	16	1111111111111010
B/8	16	1111111111010110	F/7	16	1111111111111011
B/9	16	1111111111010111	F/8	16	1111111111111100
B/A	16	1111111111011000	F/9	16	1111111111111101
			F/A	16	1111111111111110

APPENDIX IV

APPENDIX IV.1

Values of t (Student's t-test) [Steel, 1980]

df	Probability of a numerically larger value of t								
	0.5	0.4	0.3	0.2	0.1	0.05	0.02	0.01	0.001
1	1	1.376	1.963	3.078	6.314	12.706	31.821	63.657	636.619
2	0.816	1.061	1.386	1.886	2.92	4.303	6.965	9.925	31.598
3	0.765	0.978	1.25	1.638	2.353	3.182	4.541	5.841	12.941
4	0.741	0.941	1.19	1.533	2.132	2.776	3.747	4.604	8.61
5	0.727	0.92	1.156	1.476	2.015	2.571	3.365	4.032	6.859
6	0.718	0.906	1.134	1.44	1.943	2.447	3.143	3.707	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	5.405
8	0.706	0.889	1.108	1.397	1.86	2.306	2.896	3.355	5.041
9	0.703	0.883	1.1	1.383	1.833	2.262	2.821	3.25	4.781
10	0.7	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.587
11	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.437
12	0.695	0.873	1.088	1.356	1.782	2.179	2.681	3.055	4.813
13	0.694	0.87	1.079	1.35	1.771	2.16	2.65	3.012	4.221
14	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	4.14
15	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	4.073
16	0.69	0.865	1.071	1.337	1.746	2.12	2.583	2.921	4.015
17	0.689	0.863	1.069	1.333	1.74	2.11	2.567	2.898	3.965
18	0.688	0.862	1.067	1.33	1.734	2.101	2.552	2.878	3.922
19	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.883
20	0.687	0.86	1.064	1.325	1.725	2.086	2.528	2.845	3.85
21	0.686	0.859	1.063	1.323	1.721	2.08	2.518	2.831	3.819
22	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.792
23	0.685	0.858	1.06	1.319	1.714	2.069	2.5	2.807	3.767
24	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.745
25	0.684	0.856	1.058	1.316	1.708	2.06	2.485	2.787	3.725
26	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.707
27	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.69
28	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.674
29	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.659
30	0.683	0.854	1.055	1.31	1.697	2.042	2.457	2.75	3.646
40	0.681	0.851	1.05	1.303	1.684	2.021	2.423	2.704	3.551
60	0.679	0.848	1.046	1.296	1.671	2	2.39	2.66	3.46
120	0.677	0.845	1.041	1.289	1.658	1.98	2.358	2.617	3.373
inf	0.674	0.842	1.036	1.282	1.645	1.96	2.326	2.576	3.291
df	0.25	0.2	0.15	0.1	0.05	0.025	0.01	0.005	0.0005
	Probability of a larger positive value of t								

VITA

Hang-Chan Lee

Candidate for the Degree of

Doctor of Philosophy

Dissertation: A DUAL HYBRID CODER FOR HIGH QUALITY IMAGE COMPRESSION

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Seoul, Korea, March 11, 1963, the son of Young-Joo Lee and Jong-Rim Cho.

Education: Graduated from Kwan-Ak High School, Seoul, Korea, in March, 1981; received Bachelor of Science Degree in Electrical Engineering from Inha University in March, 1986; received Master of Science Degree in Electrical and Computer Engineering from Oklahoma State University in December, 1992; completed the requirements for the Doctor of Philosophy in Electrical and Computer Engineering at Oklahoma State University in July, 1997.

Experience: R. O. T. C., Korean Army, 1986 to 1988; Researcher, Energy Conversion Lab, Inha University in Korea, 1989 to 1990; Teaching Assistant, Department of Electrical and Computer Engineering, Oklahoma State University, 1993 to 1996.

Professional Membership: IEEE, Sigma, Association of Korean-American Scientists and Engineers (K. A. S. E).