HIGH EFFICIENCY BLOCK CODING TECHNIQUES FOR IMAGE DATA

by

Lo Kwok-tung
(盧 國 棟)

A Dissertation
Submitted in partial fulfilment of the requirements
for the degree of
Doctor of Philosophy

in

Department of Electronic Engineering
The Chinese University of Hong Kong

May 1992

Dedicated

to the memory of

my sincere brother

LO kwok-wai

and

my truely good friend

HO Nai-man

ABSTRACT

In recent years, with the continuing growth of the modern communication and computer technologies, the need for transmission and storage of digital image data is rapidly increasing. Due to the limitation of channel and storage capacity, the need for compression of image data is therefore obvious. Various methods have been proposed to compress image to as low bit-rate as possible, and block coding technique is found to be one of the most efficient method for handling the highly non-stationary image data. There are three existing block coding techniques : transform coding, vector quantization (VQ) and block truncation coding (BTC). Basically, they can be divided into three basic operations, which are mapping, quantization and coding. In this thesis, we are going to build up a high efficiency block-based image coding system, which has low computation complexity and high compression ability, by enhancing the mapping and the quantization process.

Two new orthogonal transforms, called weighted cosine transform (WCT) and simplified cosine transform (SCT), are proposed for image transform coding. When compared with the well known discrete cosine transform (DCT), WCT has better energy packing ability but requires more operations. On the other hand, implementation of SCT is simpler than that of the DCT while it maintains similar performance as the DCT. Pruned fast computational algorithms for the DCT, WCT and SCT are also derived to further improve the computation effectiveness of the transformation process. In a transform coding system, AC and DC coefficients generated from the transform process are normally encoded and transmitted separately. Since DC coefficients contain the luminance information of an image, large error in coding DC coefficients will lead to severely degradation of reconstruction images. A simple predictor called minimum edge difference (MED) predictor is then developed for efficient encoding of DC coefficients in a transform coding system. The MED predictor predicts DC coefficients by minimizing the edge difference between the current and adjacent blocks of data. In comparison with other existing predictors, the MED predictor results the smallest values of the prediction error variance and performs close to its optimum case in terms of mean square error when the adjacent element correlation coefficient is larger than 0.65.

Considering the quantization process, two fast encoding algorithms, called sub-codebook searching (SCS) algorithm and predictive sub-codebook searching (PSCS) algorithm, are derived in order to reduce the computation loading at the encoder of VQ image coding systems. The two algorithms are very efficient since they require the least number of operations when compared with other existing VQ encoding algorithms. Since then, a new vector quantization scheme, called predictive classified address vector quantization (PCAVQ), is proposed to further improve the compression ability of VQ coding systems. In PCAVQ, a simple classification method based on the optimum three-level block truncation coding technique is developed to divide an image into different sub-sources, such as edge, non-edge and high activity. Each sub-source is then encoded using a particular codebook. A simplified address vector quantization technique is also derived in the proposed system to reduce the bit-rate requirement. Experimental results shows that the proposed PCAVQ scheme not only improves both subjective and objective fidelity of decoded images but also increases the compression ratio of standard VQ scheme.

ACKNOWLEDGEMENTS

It is my pleasure to express my sincere gratitude to my supervisor, Dr. Wai-kuen Cham, for his most stimulating guidance, valuable advice and encouragement throughout the course of this research work, and for his patient reading and useful comments in the preparation of this thesis and other manuscripts.

I would also like to express my gratitude to the donor of the Cheng Yick Chi Graduate Fellowship whose financial assistant and support are proved to be valuable.

Special thanks also to Mrs. Yum for preparing the photographic results. This work was done in the Image Processing Laboratory at the Chinese University of Hong Kong and it would not have been as much fun without Dr. Tong Lee, Fu-lai Chung, Fuk-sing Wu, Chun-tat See, Mi Bi, Yiu-man Tham, Kwok-wai Cheung, Wai-kuen Lai and Jian Ye.

Finally, but not the least, I wish to express my deepest gratitude to my parents for their moral and financial support throughout my whole education.

LIST OF PRINCIPLE SYMBOLS AND ABBREVIATIONS

σ	Standard deviation
ρ	Adjacent element correlation coefficient
C_N^A	Order-N type A cosine transform
$\otimes$	Kronecker product operator
$\ominus$	Operator for direct multiplication of matrices
Ⓗ	H-product operator
α	Parameter of the even part of the WCT
β	Parameter of the odd part of the WCT
$[COV_x(\rho)]$	Covariance matrix of the first order Markov process
$[x]^t$	Transpose of the matrix x
σ^2	Variance
AVQ	Address vector quantization
BTC	Block truncation coding
CCD	Charge coupled device
CCITT	International consultative committee for telephone and telegraph
CMT	C-matrix transform
$COV_x(i,j)$	The $(i,j)^{th}$ element of the covarince matrix $[COV_x(\rho)]$
D	Distortion
d_c	Decision distortion
DCT	Discrete cosine transform
DCT-X	Type X discrete cosine transform
DPCM	Differential pulse code modulation
DSCT-X	Type X discrete sine transform
DST	Discrete sine transform
E{ }	Mathematical expectation
FCT	Fast cosine transform

FSVQ	Finite state vector quantization
G_{TC}	Transform coding gain
HCT	High correlation transform
HDCT	Hadamard structured discrete cosine transform
HDTV	High definition television
HT	Haar transform
ICT	Integer cosine transform
ISDN	Integrated services digital network
ISO	International organization for standardization
JPEG	Joint photographic experts group
KLT	Karhunen-Loeve transform
LCT	Low correlation transform
LOT	Lapped orthogonal transform
MDCT	Modified cosine transform
MED	Minimum edge difference
MRB	Maximum reducible bit
MRVQ	Mean removal vector quantization
MSE	Mean square error
n	Picture size
N	Block size
NEV	Normalized error variance
NMSE	Normalized mean square error
NTSC	National television system committee
PCAVQ	Predictive classified address vector quantization
PDS	Partial distance search
PMRVQ	Predictive mean removal vector quanitzation
POS	Percentage of operation saving
PSCS	Predictive sub-codebook searching
PSCT	Phase shift cosine transform

R(D)	Rate distortion function
RAM	Random access memory
RC	Residue correlation
SCS	Sub-codebook searching
SCT	Simplified cosine transform
SFCT	Simplified structures fast cosine transform
S_i	The i^{th} Voronoi region
SNR	Signal to noise ratio
ST	Slant transform
t_a	Threshold value of the simplified address VQ technique
Thd_1	Lower threshold value of classifier II of the PCAVQ scheme
Thd_2	Upper threshold value of classifier II of the PCAVQ scheme
TIE	Triangular inequality elimination
TSVQ	Tree-searched vector quantization
VLSI	Very large scale integration
VQ	Vector quantization
WCT	Weighted cosine transform
WHT	Walsh-Hadamard transform
Y	Codebook
y_c	Control codeword
Y_c	Sub-codebook for the codeword y_c

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND - THE NEED FOR IMAGE COMPRESSION

In the last few years, with the rapid development of the semiconductor technologies and the continuing growth of the communication industries, the integrated services digital network (ISDN) became applicable in real life. With this communications network, exchange of data, voice as well as images in digital format between different computers becomes feasible. Such technological advances affect the trend for development of modern computer systems. In early and mid 80's, computer systems can only manipulate data, text and graphics due to limitation of computing power and memory storage. In the next generation of computer systems, all kinds of information including data, text, stereo audio, high-resolution color images as well as high definition motion video can be managed and processed [FOX91]. To take advantages of such development, digital image processing techniques have already been applied in conventional photographic and television services in recent few years. For photographic service, a digital image can be obtained through a digital still image camera. In such a camera, a charge coupled device (CCD) first samples the image in analogue format, and then a high speed analogue-to-digital converter changes the signal into digital form and stores it on solid state RAM, flash memory or floppy diskette. By using this digital camera, we can process and manage the images in our own computer and perform sophisticated operations such as editing, enhancement and zooming. With further improvement of the resolution of the CCD camera and the continuing decreasing of the cost of memory storage, the digital camera will probably replace the conventional camera in the near future. For television service, high definition television (HDTV) [HABE91] [NINO91] has been a very hot topic in consumer electronic industries during last few years. HDTV system is the next generation television system with high picture resolution. Normally, a picture with resolution around 1125 x 1125 [NINO91] is desired. A HDTV system is now reaching the stage of practical use. In Japan, HDTV programs have been broadcasting daily. At present, each HDTV set costs about HKD

$100,000 and such high price cannot be afford by most families. With the continuous decrease of hardware cost, HDTV system will become popular in our daily life to replace the conventional TV broadcasting systems in the coming years when the price of HDTV set drops down to an acceptable level, says about HKD$20,000.

The systems mentioned before and other existing visual services, such as video-conferencing, remote sensing, image archiving and medical imaging, require efficient methods for storage and transmission of the image or video data in digital format. Representing a 1024 x 1024 monochrome image with 256 gray levels requires 8 Mbits storage capacity, and that of a color images even triples this size. The data rate of NTSC video could even exceed 400 Mbits per second. For such large memory storage and channel capacity requirement, the need for image data compression is manifested.

1.2 IMAGE COMPRESSION - AN OVERVIEW

The goal of image compression is to reduce the number of bits for representing an image as much as possible and at the same time provide a faithful reconstruction of the original image. Research works on image compression have been going on for over 30 years. Extensive studies of image compression can be found in various review papers [NETRl80] [JAIN81] [MUSMpg85] [KUNTik85] [KUNTbl87] [NASRk88] [FORCk89] and text books [PRAT79] [ROSEk82] [JAYAn84] [CLAR85] [GONZw87] [NETRh88] [CHANG89] [JAIN89] [LIM90] [FARR90] [WOOD91]. Many different kinds of image compression methods have been proposed and they can be classified into two main categories : lossless and lossy methods. In a lossless compression system, the encoding process and the decoding process are reversible. In other words, the encoded image can be decoded or reconstructed to obtain the original image exactly. However, the obtainable compression ratio, which is defined as the ratio of number of bits required to represent the original image to that of encoded image, is rather limited at somewhere between 2 to 3. Typical examples of lossless compression method are Huffman coding [HUFF52] and run length coding [GALL75]

[GONZw87] [NETRh88]. Nevertheless, Our research works have been focusing on developing high efficiency block coding techniques, which is a kind of lossy compression methods, therefore, lossless compression method will not be discussed in this thesis.

In comparison with lossless compression method, lossy compression technique can achieve higher compression ratio but introduce difference or distortion between the original image and reconstructed image. Distortion can be measured by subjective and objective methods which will be discussed in more details in the next chapter. The goal of designing a lossy compression system is to achieve a certain compression ratio while producing good image quality, or vice versa, to maintain a desired quality while minimizing the data rate. Many kinds of lossy compression methods have been developed and basically they can be divided into two categories: waveform coding [JAYAn84] and perceptual based image coding [KUNTik85] [KUNTbl87] which is a human visual oriented coding scheme and is also called the 'second generation' image coding technique. In our research works, we concentrate only on methods of waveform coding, therefore, the perceptual based coding techniques will not be discussed in this thesis. Typical examples of waveform coding are predictive coding, sub-band coding, transform coding, vector quantization (VQ) and block truncation coding (BTC).

1.2.1 Predictive Coding - DPCM

Predictive coding or differential pulse code modulation (DPCM) [NETRl80] [JAYAn84] [NETRh88], which is one of the simplest image compression methods, exploits the redundancy of image data directly in spatial domain. Figure 1.1 shows the block diagram of a DPCM system. Since image data is highly correlated, on average the pixels lying in the neighbour will tend to have similar gray levels. Therefore, rather than transmitting the pixel value x(i) directly, the values of one or more early pixels (previously encoded) in both horizontal and vertical directions can be used to predict the value of current pixel and the difference between the predicted value $\hat{x}(i)$ and the original one, that is prediction error d(i), will then be quantized for transmission. On the receiver side, data are first decoded and then added to the estimated data to construct the image. The predictor on the receiver is a duplicate of that at the transmitter. It is shown that [JAYAn84] when the adjacent element correlation coefficient of the image is close to 1, the variance of the quantizer input d(i) is much smaller than that of the coder input x(i). Since quantization error is proportional to quantizer input variance for a given number of bits per pixel, the above reduction of quantizer input variance leads directly to reduction of quantization error variance. This also means, for the same quantization error, d(i) requires fewer number of bits than x(i). Compression ratio of DPCM coder is therefore dependent on the selection of predictors.

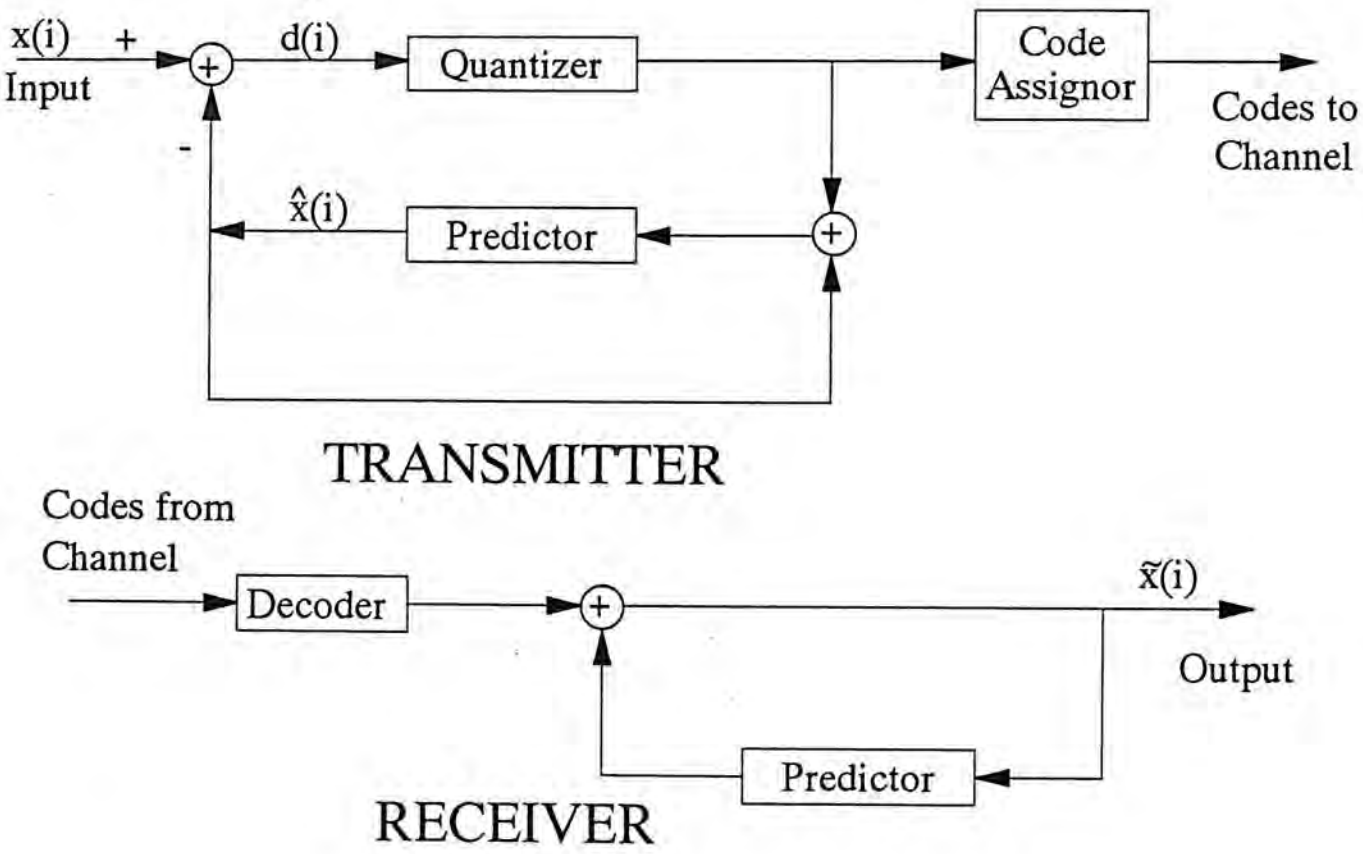


Figure 1.1 Block diagram of DPCM system

One kind of research works on predictive coding is to improve the performance of the quantizer by making it optimal for signals having a particular probability density function [MAX60] [NETR77] [LLOY82] [KRISf86] such as Gaussian and Laplacian. Another type of works is to adapt the predictor to local statistics of an image by employing some sophisticated algorithms. The adaptive predictors can be divided into two types. The first [HABI77] [CHENw91] is to separate an image into many areas. Each area has its own predictor. An overhead must be sent to the receiver to tell which predictor is available for each pixel. This however will reduce the compression ratio. The second [ALEXr82] [PIRS82] [ALEXr85] is to determine which predictor is available for the current pixel by the characteristics of the previous pixels and so eliminate the need of overhead. This type of algorithms dynamically adjusts the predictive coefficients of the predictor according to the values and the prediction errors of previous pixels. Since DPCM is suitable only for high to medium bit-rate applications (one bit or more per pixel) [JAYAn84], the technique of DPCM is normally used together with other sophisticated coding methods likes transform coding [PLOYr82] [CLAR84] and subband coding [WOODn86] in low bit-rate image compression applications. In our research works, we utilize DPCM technique to encode DC coefficients in a transform coding system.

1.2.2 Sub-band Coding

Sub-band coding [JAYAn84] has been used in speech coding for many years but drawn attentions in image coding only in last few years. Sub-band coding is a general waveform coding technique and can be shown to include the transform coding as a special case. As in transform coding, the sub-band coder aims to divide the correlated image data into uncorrelated parts. The principle of sub-band coding is based on the decomposition of the input signal into narrow bands where each band is then decimated, and coded separately. Since the spatial energy of the image is concentrated on low frequency bands in most cases, compression will be achieved by allocating more bits to low frequency bands and fewer bits to high frequency bands. Coding of different bands can be performed in several ways. Such a system requires sophisticated bandpass filters to eliminate or minimize aliasing effects. By using sub-band coding, image transmission rate as low as 0.5 bpp can be achieved.

In early stage, sub-band coding is a technique for one-dimensional signal only. In 1984, Vetterli extended this method for multidimensional signals [VETT84]. However, this paper only provided signal processing aspects of multidimensional filterbank but not a coding scheme. Brandt [BRAN85] was the first to apply sub-band coding in image compression, and this work was extended by several researchers [WOODn86] [GHARt86] in next year. Later on, various sub-band coding systems have been proposed for coding monochrome and color images [GHARt88] [WOOD91]. In recent years, several researchers have applied sub-band coding for packet video [KARLv88] [WANGr91] and high definition television (HDTV) [LEGAt88] applications. Some hybrid systems have also been investigated [KIMbsm88] [KIMsm89]. A review on sub-band coding of images was reported by Forchheimer and Kronander [FORCk89] in 1989. Nevertheless, this thesis addresses only the block coding techniques for image data and the sub-band coding system will not be discussed in details.

1.2.3 Transform Coding

In transform coding systems [JAIN81] [ROSEk82] [CHAM83] [JAYAr84] [CLAR85], image data are divided into subsequent blocks of data, typical block sizes are 8x8 and 16x16. Data compression in transform coding is achieved by the transformation of a block of pixels into another block of coefficients where maximum energy is packed into a minimum number of coefficients. Therefore, the overall quantization error can be minimized by allocating more bits for transmission of coefficients having larger variances, and fewer bits for coefficients having smaller variances. The optimum bit allocation can be derived from rate distortion theory [BERG71] which states that the output of a source can be transmitted with average distortion D if the transmission rate is larger than R(D). If D is the mean square error and the source has Gaussian probability distribution, then R(D) is found to be

$$R(D) = \begin{cases} \frac{1}{2}\log_2\left(\frac{\sigma^2}{D}\right) & \text{for } \sigma > \sqrt{D} \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

where σ is the standard deviation. Eqn. (1.1) means that R(D)-bits quantization of a coefficient having standard deviation σ would result average distortion D. Therefore, given the average distortion D, eqn.(1.1) can be used to determine the number of bits required for each transform coefficient [DAVI72] [JAIN81] [JAYAn84] [FARR90]. Usually, bits are allocated to the coefficients such that all the coefficients receive the same amount of distortion. Since spatial energy is concentrated in a few coefficients, only a small fraction of the transform coefficients need to be coded.

In early transform coding systems, two simple methods [JAIN] [JAIN89] [LIM90] used to determine which transform coefficients to code are zonal coding and threshold coding. In zonal coding, only the coefficients within a specified region are coded. Many factors, such as the transform used and the available number of bits, affect the zone shape and size. Whilst in threshold coding, transform coefficients are compared with a threshold, and those above the threshold are coded. Threshold coding is an adaptive method. The choice of which transform coefficients are coded depends on the local image characteristics. For a fixed threshold, the number of coefficients coded varies from different images. Since some regions of an image consist of highly correlated pixels and some regions contain a high degree of activity, a coding scheme adapts to local statistics of the image is necessary. At the expense of increased complexity and computation time, adaptive coding schemes always outperform non-adaptive ones. Numerous adaptive schemes have been proposed. In 1977, Chen and Smith [CHENs77] introduced an efficient adaptive scheme for transform image coding. In their scheme, image blocks are first classified into four activity classes according to the energy of the AC transform coefficients of the blocks. Each class of block will then be encoded by Max quantizers [MAX60] according to its bit allocation table which is obtained by calculating the transform coefficient variances of the class. Follow on, several adaptive schemes [WONGs81] [NGAN82] have been proposed to enhance the performance of Chen & Smith's scheme. However, the method of Chen & Smith requires 'two passes' over the image, and thus incurs a one frame delay - whilst the 'present' frame is having its statistics measured for classification, the previous one is actually encoded for transmission. This will make the implementation of the coding system complicated. Therefore, research work was devoted to find other adaptive coding schemes requiring only 'one pass'. In 1984, Chen & Pratt [CHENp84] proposed an 'one pass' transform coder called scene adaptive coder. In this coder, entropy coded uniform quantization is used to code the transform coefficients and a feedback mechanism is applied to adapt the step-size of the uniform quantizer to local statistics of an image. This work was extended by Ngan et al. [NGANls89] by incorporating human visual properties into the coding scheme. In last few years, several sophisticated transform coding algorithms have been proposed to further improve the system performance at the expense of complexity. Farrelle and Jain [FARRj86] [FARR89] developed a new approach to transform coding called recursive block coding. In their scheme, an image block is decomposed into boundary and residue processes and each process is coded with different transforms. Rose et al. extended Farrelle's idea and proposed a DCT/DST alternate-transform image coding scheme [ROSEhd90]. On the other hand, Malvar et al. introduced the lapped orthogonal transform (LOT) [MALVs88] [MALVs89] [CASSsj89] [MALV90] which is a new tool for block transform coding with basis functions that overlap adjacent blocks in order to reduce the blocking effect of decoded images. By using those transform coding techniques, image can be encoded at bit-rates of 0.4-0.5 bpp. Recently, an international standard for still image compression called JPEG scheme [JPEG90] [WALL92] has been proposed. It is basically a transform coding system that taken into account of human visual system and will be discussed in more details in chapter 2.

1.2.4 Vector Quantization

Vector quantization (VQ) [GRAY84] [LIM90] is another block coding technique for image compression. It originates from the fact that better performance can always be obtained by quantizing the source as a series of vectors rather than scalars, even if the source is memoryless, that is each element of the sequence is independent of each others [GRAY84] [MAKHrg85]. In vector quantization of images, each block of data is linearly scanned to form a vector. A set of vectors from different images is then chosen as the training sequence from which a codebook of representative vectors is generated. In the quantization process, the codebook is searched for the closest match representative vector for each input vector. Compression is achieved by representing the codeword using the index whose length is much shorter than that of the input vector. At the decoder, the vector can be reconstructed by simple table look-up methods using the index as the address to a table containing the codewords. For example, if we consider a vector of length 16 which is due to the block size of 4x4, then if the codebook contains 256 vectors which need 8 bits to represent and the average bit-rate is 8/16 = 0.5 bits per pixel (bpp).

The idea of VQ is not new but it attracts a great deal of attention in image coding as the rapidly development of computer technologies in the last ten years. The extensive study of applying VQ in image coding is due to the simplicity of the decoder design when compared with transform coding and sub-band coding. A review on image coding using vector quantization was given by Nasrabadi and King [NASRk88] in 1988. The main problems of the standard VQ scheme described before are the complexity of the encoding process and the edge degradation of decoded images. To overcome such shortcomings, various vector quantization coding schemes have been proposed. In 1985, Foster et al. proposed a new VQ coding scheme called finite state vector quantization (FSVQ) [FOSTgd85]. In this scheme, an input vector is first determined its present state according to a specific state function. After the state is decided, the corresponding sub-codebook which is a sub-set of the whole codebook will be searched to find the representing codeword of the input vector. Since searching is performed on a smaller space, a lot of computations can be saved. Follow on, several enhanced finite state vector quantizers [ARAVg86] [KIM88] [NASRf90a] have been developed. On the other hand, Ramamuthi and Gersho proposed a classified vector quantization scheme [RAMAg86] to tackle the problem of poor edge reproduction ability of the standard VQ scheme in 1986. In their scheme, an image is first divided into different sub-sources, such as edge and shade, and each sub-source is then vector quantized using different codebooks. In the following few years, different classified VQ schemes [NGANk89] [DAVI90] [POc91] [KIMI91] [KIMI92] have been proposed. In the last two years, Nasrabadi and Feng [FENGn88] [FENGn89] [NASRf90b] [NASRf90c] proposed an efficient coding algorithm called address vector quantization for reducing the bit-rate requirement of the standard VQ scheme by exploiting the interblock correlation. By using such sophisticated VQ techniques, transmission rates as low as 0.3-0.4 bpp can be achieved. In our research works, we will propose methods to tackle several problems occurred in standard VQ scheme.

1.2.5 Block Truncation Coding

Block truncation coding (BTC) is another block-based image coding method proposed by Delp and Mitchell [DELPm79] in 1979. The principle of standard BTC algorithm is to use a simple one-bit quantizer that adapts to local statistics of an image. The two reconstructed levels of the quantizer are selected such that the first and second moments of the original block of data are preserved. Since one-bit quantizer is used, each quantized pixel can be represented by either '0' or '1'. After processing each block of data, two kinds of information are required to send : a bitmap, which has the values of '0' and '1', contains the reconstructed levels of each pixel and the local statistics (mean and standard deviation) of the block. If the mean and standard deviation require eight bits to represent, then the overall bit-rate required for the standard BTC scheme is equal to 2 bpp. In 1984, Halverson et al. [HALVgw84] extended the idea of Delp and Mitchell to develop a generalized block truncation coding algorithm by preserving the first three moments of an image block. Several modified BTC algorithms [LEMAm84] [UDPIr85] have also been proposed at the same time. Hui developed an optimum mean square error (MSE) [HUI89] one-bit quantizer for BTC system in 1989. In recent years, research works on block truncation coding is mainly concentrated on increasing the compression ratio and image quality while maintaining its simplicity. Udpikar and Raina [UDPIr87] used vector quantization for encoding the bit-map of the image and reduced the overall bit-rate to 1.5 bpp. This work was then extended by Efrati et al. in their classified block truncation coding system [EFRAlm91]. On the other hand, different hierarchical schemes [NASRc90] [KAMEsg91] and adaptive schemes [GRIShw87] [HUI90] have also been proposed. By using such BTC methods, image can be encoded at the bit-rates of 1-1.5 bpp. In our works, we will not utilize the BTC technique to code the pixels of the image directly, instead we will use the BTC in vector quantization system.

1.3 BLOCK-BASED IMAGE CODING TECHNIQUES

There are three major block coding techniques for image compression : they are transform coding, vector quantization and block truncation coding. Since image data are statistically non-stationary, we can't treat the whole image as any kind of signal and there is still no good mathematical model to represent. In order to adapt to local statistics of an image, a straightforward method is to divide the image into non-overlapping blocks of data such that stationary holds within a block. Compression method using such image segmentation method is called block-based image compression method. This thesis investigates high efficiency block coding techniques for image data. The common structure of the block-based image coding technique can be described as follows. Consider an image with size n x n and the block size is N x N, it can be decomposed into m x m subsequent non-overlapping blocks as shown in Figure 1.2, where m=n/N.

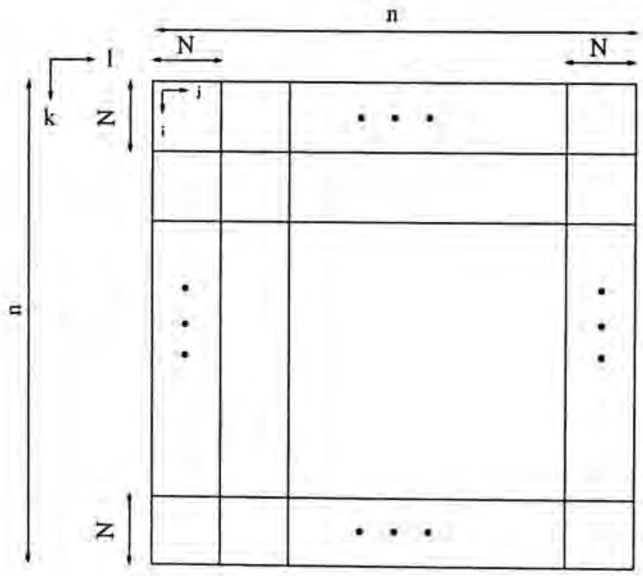


Figure 1.2 Block classification of an image

The index (k,l) as shown in Figure 1.2 indicates the position of the block in the image while the index (i,j) represents the position of the pixel in the block. For example, $x_{k,l}(i,j)$ represents the $(i,j)^{th}$ pixel in the $(k,l)^{th}$ block of the image. The block diagram of a typical block-based image coding system is shown in Figure 1.3.

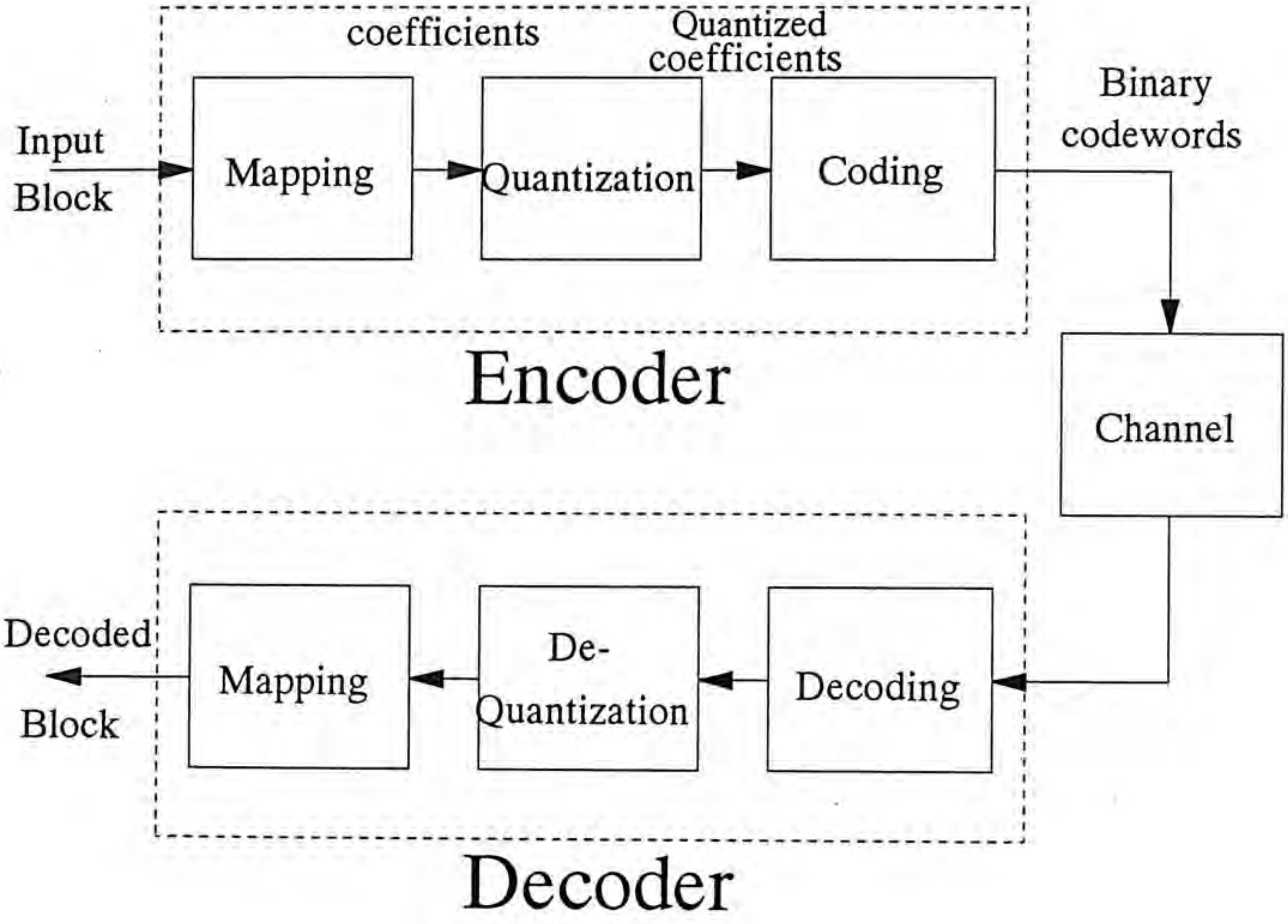


Figure 1.3 Block diagram of a block-based image coding system

Compression of data by a block-based coding system is achieved by three basic operations : mapping, quantization and coding. The first operation is to map the highly correlated blocks of image into coefficients having a better characteristics or format for quantization. In transform coding, the mapping operation is an orthogonal transformation which packs maximum energy into minimum number of coefficients. The mapping operation in vector quantization and block truncation coding is simply to re-arrange two-dimensional block data into an one-dimensional array. Among the three block coding techniques, the influence of mapping operations to the overall system efficiency is the largest in transform coding. After mapping the spatial data into a set of coefficients, the next operation is to quantize the coefficients.

Quantization is the most important part affecting the efficiency of the coding systems. Quantization can either be in the form of scalar or vector. In transform coding and block truncation coding, scalar quantization is normally used. After quantization, it is necessary to encode the quantized coefficients for transmission through the channel, that is to assign binary codeword for each quantized coefficient. Binary codeword may either be assigned in fixed-length or variable length. When the quantization output levels of the quantizers are not evenly distributed, further compression can be obtained by using a variable-length Huffman coding technique to the quantization output levels of the quantizer. The binary codewords are then transmitted through the channel to the decoder. In our works, the channel is assumed to be error-free. At the decoder, the reverse operations are performed to obtain the decoded blocks of the image. Detail discussion of the block-based image coding systems will be found in next chapter.

1.4 GOAL OF THE WORK

In general, the term 'high efficiency' means to use minimum operations to achieve maximum output. When applying in a block-based image coding system, the aim at high efficiency is to use as less number of computations as possible to achieve as much compression ratio as possible while maintaining a high quality reconstructed image at the output. In our work, we will concentrate on finding techniques to improve the efficiency of the mapping process and the quantization process in block based image coding systems. We use two different approaches to achieve the goal. One is to improve the compression ratio of the existing system while keeping the computational requirement similar to the original. Another approach is to reduce the computation complexity of the existing system while maintaining similar quality of decoded images and compression ratio. In our works, we first develop two new orthogonal transforms and their pruned fast computational algorithms for improving the efficiency of the transformation process in transform coding system. Since DC coefficients contain most of the spatial energy of an image, we propose a simple predictor for efficient encoding of DC coefficients in transform coding systems. For the quantization process, concentration will be put on vector quantization for its superior rate-distortion performance than the scalar quantization. We propose several methods to improve the efficiency of the standard VQ scheme by either reducing the computational requirement of its encoding process or improving the visual quality of decoded images.

1.5 ORGANIZATION OF THE THESIS

Having stated our goal of work which is to achieve a high efficiency block based image coding system, we now describe the organization of this thesis. Following this chapter, a background of the notation and terminology will first be given at the beginning of chapter 2. The basic theories related to the existing block-based image coding systems - transform coding, vector quantization and block truncation coding will then be reviewed in the same chapter.

In chapter 3, two approaches of generating new orthogonal transforms will be discussed and two new transforms, namely weighted cosine transform (WCT) and simplified cosine transform (SCT) will be formulated accordingly for image compression. Different methods will be utilized to evaluate the performance of the two transforms in image coding systems. Pruned fast computational algorithms for DCT, WCT and SCT will then be developed in chapter 4. Detail analysis of the proposed pruned algorithms will be given in such chapter.

Since DC coefficients contain most of the spatial energy of an image, an efficient method to encode DC coefficients in a transform coding system will be discussed in chapter 5. Both analytical methods and computer simulations will be used to evaluate the efficiency of the proposed algorithm.

Chapter 6 describes two fast encoding algorithm for vector quantization of images. Computer simulations using real images will be used to examine the performance of the two proposed algorithms. In chapter 7, a new vector quantization coding scheme, called predictive classified address vector quantization (PCAVQ), will be developed to improve the computation effectiveness, image fidelity and compression ratio of standard VQ scheme. Finally, an overall summary of the thesis will be given in chapter 8. The topics for further investigation will also be highlighted in that chapter.

CHAPTER 2 BLOCK-BASED IMAGE CODING TECHNIQUES

In this chapter, we will establish a common background of notation and terminology in image coding systems which will be used throughout the thesis. A review of the basic theories of block-based image coding systems will also be provided. In section 2.1, the statistical model of an image will be reviewed and the image fidelity criteria will be discussed in section 2.2. The basic theories related to transform coding, vector quantization and block truncation coding will then be given in sections 2.3, 2.4 and 2.5 respectively.

2.1 STATISTICAL MODEL OF IMAGES

In the design and analysis of an image coding system, it is convenient and often necessary to mathematically characterize the image data to be processed. There are two basic mathematical models : deterministic model and statistical model [PRAT79] [FORCk89]. For deterministic model, a mathematical function is defined and point properties of image data are considered. In statistical representation, image data are specified by average properties like mean and covariance function. In our works, we concentrate only on the statistical model for image representation. In early image coding systems, since image signal is sampled in a row-by-row basis, it is usually considered as one-dimensional. Different compression methods were developed based on one-dimensional statistical models of image. However, image signal is in fact two-dimensional signal and so two-dimensional statistical model is more appropriate for developing efficient coding algorithms. Most recent image coding techniques were derived based on two-dimensional statistical models of image, which often have better performance than those early methods using one-dimensional models.

2.1.1 One-Dimensional Model

Consider an input vector x sampled from an one-dimensional, zero mean, unit-variance first-order Markov process with adjacent element correlation coefficient ρ, the covariance matrix of x is given as

$$[\mathrm{COV}_x(\rho)] = E\{xx^t\}$$

$$= \begin{bmatrix} 1 & \rho & \rho^2 & . & . & . & \rho^{N-1} \\ \rho & 1 & \rho & . & . & . & \rho^{N-1} \\ \rho^2 & \rho & 1 & . & . & . & \rho^{N-3} \\ . & . & . & . & . & . & \\ . & . & . & . & . & . & . \\ \rho^{N-2} & \rho^{N-3} & \rho^{N-4} & . & . & . & \rho \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & . & . & \rho & 1 \end{bmatrix} \qquad (2.1)$$

where the superscript t means transpose. In other words, the (i,j)th element of the covariance matrix $[\mathrm{COV}_x(\rho)]$ is given by

$$\mathrm{COV}_x(i,j) = \rho^{|i-j|}$$

Such a Markov process is always used to represent one-dimensional image data.

2.1.2 Two-Dimensional Model

Consider N by N arrays x(i,j) and x(k,l) which are sampled from a zero-mean 2-D stationary random field with unit variance and covariance function $\mathrm{COV}_x(i,j,k,l)$. Suppose the covariance function is separable and is given by

$$\begin{aligned} \mathrm{COV}_x(i,j,k,l) &= E\{x(i,j)\ x(k,l)\} \\ &= \mathrm{COV}_x(i,k)\ \mathrm{COV}_x(j,l) \end{aligned} \qquad (2.2.a)$$

Therefore, we can extend our assumption on one-dimensional image data to two-dimensional as follows :

$$\mathrm{COV}_x(i,j,k,l) = \rho_h^{|i-k|} \cdot \rho_v^{|j-i|} \qquad (2.2.b)$$
$$\text{for } i,j,k,l \in [0..N-1]$$

where ρ_h and ρ_v are the adjacent element correlation coefficients in the horizontal and vertical directions respectively. The covariance matrix $[\mathrm{COV}_x(\rho_h, \rho_v)]$ is of size N^2 x N^2, and can be obtained by applying the Kronecker product between two matrices as follows

$$[COV_x(\rho_h, \rho_v)] = [COV_x(\rho_h)] \otimes [COV_x(\rho_v)] \tag{2.3}$$

where ⊗ is the Kronecker product operator, $[COV_x(\rho_h)]$ and $[COV_x(\rho_v)]$ are the covariance matrices of the first-order Markov process with ρ_h and ρ_v respectively as defined in (2.1).

For the non-separable isotropic model, the covariance function is given by

$$COV_x(i,j,k,l) = e^{-\sqrt{[(i-k)\ln\rho_h]^2 + [(j-l)\ln\rho_v]^2}} \tag{2.4}$$

$$\text{for } i,j,k,l \in [0..N-1]$$

Similar to the separable case, the non-separable covariance function can also be represented as a N^2 x N^2 matrix.

2.2 IMAGE FIDELITY CRITERIA

2.2.1 Objective Fidelity Criteria

Since most image compression methods are lossy, a criterion should be used to measure the difference or distortion between the original image and the reconstructed image. Examples of objective fidelity criteria are mean square error (MSE), peak signal to noise ratio (PSNR) and normalized mean square error (NMSE).

2.2.1.1 Mean Square Error (MSE)

Mean square error [PRAT79] [JAYAn84] [JAIN89] of a decoded image is defined as

$$MSE = \frac{1}{n_r \times n_c} \sum_{i=0}^{n_r-1} \sum_{j=0}^{n_c-1} [x(i,j) - \tilde{x}(i,j)]^2 \tag{2.5}$$

where n_r and n_c represent the number of pixels in the row and column direction of an image respectively. x(i,j) and $\tilde{x}(i,j)$ represents the (i,j)th pixel elements of the original image and the reconstructed image respectively. MSE measures the absolute magnitude of the distortion between the original and reconstructed image and is independent of image data. For example, a decoded image of a transform coding system at 0.5 bpp has MSE normally less than 100.

2.2.1.2 Peak Signal to Noise Ratio (PSNR)

Peak signal to noise ratio [PRAT79] [JAYAn84] [JAIN89] of a decoded image is defined as

$$\mathrm{SNR} = 10\log_{10}\left[\frac{(\text{Peak value of signal})^2}{(\text{MSE})}\right] \text{ dB} \tag{2.6}$$

The peak value of the signal depends on the bit resolution of the pixel of the image. For an image of 8-bit resolution per pixel, the peak value of the image is then at most 255. The value of PSNR is also independent of image data and provides an indication to the absolute error magnitude of the decoded image in log scale. Typical values of PSNR for a decoded image of a transform coding system are somewhere around 28 to 32 dB when the bit-rate is 0.5 bpp.

2.2.1.3 Normalized Mean Square Error (NMSE)

Normalized mean square error [PRAT79] [JAYAn84] [JAIN89] is obtained by normalization of the mean square error with the total energy of an image and which is defined as

$$\begin{aligned}\mathrm{NMSE} &= \frac{\text{MSE}}{\text{Total Energy of Image}}\\ &= \frac{\text{MSE}}{\frac{1}{n_r \times n_c}\sum_{i=1}^{n_r}\sum_{j=1}^{n_c} x(i,j)^2}\end{aligned} \tag{2.7}$$

Unlike the criteria of MSE and PSNR, the value of NMSE is dependent on the energy of the image. It gives a relative value of the MSE with respect to different images. When two images have the same value of MSE but different energy. The one with larger energy will result smaller value of NMSE than other. This result concides with the fact that high activity images can tolerate larger errors than those of low activity. Therefore, NMSE gives relative indication of the error magnitude of decoded image. Typical values of NMSE for a decoded image of a transform coding system are somewhere around 10^{-2} to 10^{-3} when the bit-rate is 0.5 bpp.

2.2.2 Subjective Fidelity Criteria

Two pictures with the same MSE, PSNR or NMSE may not have the same visual qualities. Since the output images of an image coding system are to be observed by people, it is more appropriate to use a subjective criterion [PRAT79] [JAYAn84] [JAIN89] to measure how good the image is when viewed by human observers. An important characteristic of the human visual system is its logarithmic sensitivity to light intensity. Human observers are more sensitive to dark areas than light areas of an image. The subjective quality of an image can be evaluated by showing this image to different observers and averaging their evaluation. One commonly used subjective criterion is the overall goodness scale [JAIN89] as listed in Table 2.1.

Overall Goodness Scales

Excellent	(5)
Good	(4)
Fair	(3)
Poor	(2)
Unsatisfactory	(1)

Table 2.1 Image Goodness Scales

In this thesis, several standard test images are utilized to evaluate the performance of our proposed image coding algorithms using the above fidelity criteria. The statistics of the monochrome and color test images used in this thesis are shown in Appendices A and B.

2.3 TRANSFORM CODING THEORY

Compression of image data by transform coding is achieved by three basic processes. The first process is to transform highly correlated image data into weakly correlated coefficients. The next process is to quantize the transform coefficients so that they can be represented by fewer bits. Finally, the quantized coefficients should be encoded for efficient transmission. The block diagram of a basic transform coding system is shown in Figure 2.1.

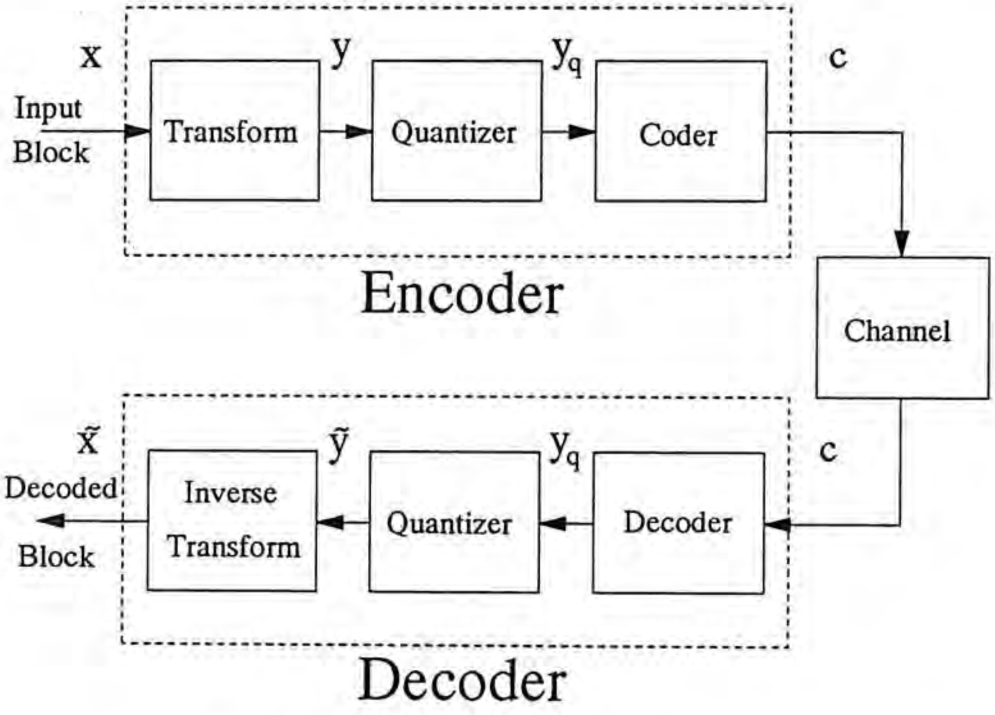


Figure 2.1 Block diagram of a basic transform coding system

2.3.1 Transformation

The primary objective of the transformation is to convert statistically dependent picture elements into an array of uncorrelated coefficients such that most of the energy is packed into a minimum number of coefficients. Normally, two-dimensional separable transform [JAIN81] [JAIN89] is preferred to one-dimensional transform because it reduces the pixel correlations in both vertical and horizontal directions, thereby achieving higher energy packing ability. A separable transform has simpler implementation than a non-separable because two-dimensional separable transformation can be achieved by sequentially applying a one-dimensional transformation in vertical and horizontal directions. Consider an image block [x] with size N x N passing through a two-dimensional separable transformation which utilizes an orthonormal transform matrix [T]. The block of data in transform domain [y] is given by

$$[y] = [T][x][T]^t \tag{2.8}$$

where the superscript t represents transpose of the matrix. There are several important properties for a two-dimensional separable orthonormal transform :

a. Energy Conservation

It can be proved easily [JAIN81] [JAIN89] that the total energy in spatial and transform domains are equal, that is

$$\sum_{p=0}^{N-1}\sum_{q=0}^{N-1} |\, y(p,q)\,|^2 = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |\, x(i,j)\,|^2 \tag{2.9}$$

where x(i,j) and y(p,q) are the elements of the original pixels and transform coefficients. On the other hand, the average reconstruction error variance is equal to the average quantization error variance in transform domain [JAIN89] [WANGg88], that is

$$\sigma_r^2 = \frac{1}{N \times N} \sum_{i,j=0}^{N-1} \sigma_r^2(i,j) = \frac{1}{N \times N} \sum_{i,j=0}^{N-1} \sigma_q^2(i,j) = \sigma_q^2 \tag{2.10}$$

where $\sigma_r^2(i,j) = E\{[x(i,j) - \tilde{x}(i,j)]^2\}$, i,j = 0, 1,..., N-1 is the reconstruction error variance of the $(i,j)^{th}$ pixel element;

$\sigma_q^2(i,j) = E\{[y(i,j) - \tilde{y}(i,j)]^2\}$, i,j = 0, 1,..., N-1 is the quantization error variance of the $(i,j)^{th}$ transform coefficient;

$x(i,j)$ = the $(i,j)^{th}$ pixel of the image;

$\tilde{x}(i,j)$ = the $(i,j)^{th}$ pixel of the reconstructed image;

$y(i,j)$ = the $(i,j)^{th}$ transform coefficient

$\bar{y}(i,j)$ = the $(i,j)^{th}$ quantized transform coefficient

In case of scalar quantization, the quantization error variance σ_q^2 is related to the input signal variance σ_x^2 by the following expression :

$$\sigma_q^2 = \varepsilon^2 2^{-2B} \sigma_x^2 \tag{2.11}$$

where ε^2 is the correction factor [WANGg88] of the quantizer which depends on the probability density function of the input signal and on the quantizer characteristics, and B is the number of bits assigned to the quantizer. The problem of optimizing a transform coding system using a scalar quantization scheme is then reduced to finding an allocation of bits B(i,j) assigned to the $(i,j)^{th}$ transform coefficient such that the average coefficient quantization error variance

$$\sigma_q^2 = \frac{1}{N \times N} \sum_{i,j=0}^{N-1} \sigma_q^2(i,j) = \frac{1}{N \times N} \sum_{i,j=0}^{N-1} \varepsilon^2(i,j)\, 2^{-2B(i,j)} \sigma_y^2(i,j) \tag{2.12}$$

where

$$\sigma_q^2(i,j) = \varepsilon^2(i,j)\, 2^{-2B(i,j)} \sigma_y^2(i,j) \tag{2.13}$$

is minimized with the constraint of a fixed average rate

$$B = \frac{1}{N \times N} \sum_{i,j=0}^{N-1} B(i,j) = \text{constant.} \tag{2.14}$$

Using the Lagrange multiplier method,

$$\frac{\partial}{\partial B(i,j)} \left[\sigma_q^2 - \lambda \left(B - \frac{1}{N \times N} \sum_{i,j=0}^{N-1} B(i,j) \right) \right] = 0 \tag{2.15}$$

$$i,j = 0, 1, 2, ..., N-1$$

The optimum bit allocation is then obtained as follows :

$$B(i,j) = B + \frac{1}{2}\log_2 \frac{\varepsilon^2(i,j)\sigma_y^2(i,j)}{\left[\prod_{p,q=0}^{N-1}\varepsilon^2(p,q)\sigma_y^2(p,q)\right]^{\frac{1}{N\times N}}} \qquad (2.16)$$

$$i,j = 0, 1, 2, \ldots, N\text{-}1$$

The corresponding average reconstruction error variance is given by

$$\min(\sigma_r^2) = \min(\sigma_q^2) = \varepsilon^2 2^{-2B}\left(\prod_{k,l=0}^{N-1}\sigma_y^2(k,l)\right)^{\frac{1}{N\times N}} \qquad (2.17)$$

Obviously, the optimum bit allocation scheme results in the average reconstruction error variance proportional to the geometric mean of the transform coefficient variances (GM_y), i.e.

$$GM_y = \left(\prod_{p,q=0}^{N-1}\sigma_y^2(p,q)\right)^{\frac{1}{N\times N}} \qquad (2.18)$$

It can be concluded that, in transform coding system, the optimum transform minimizes the geometric mean of the transform coefficient variances and keeps the arithmetic mean constant. Therefore, this statement can be used as a criterion to select an optimum transform in transform coding systems.

b. Energy Compaction and Decorrelation

Through orthonormal transformations, the average energy of the transform coefficients are non-uniform distributed with most energy packed into a few coefficients. Since the total energy is preserved, many of the transform coefficients will contain very little energy. Such coefficients have insignificant information contents and need not be coded for transmission. It has been shown that the higher the correlation between pixel elements is, the higher energy compaction will be achieved.

Consider an input vector x sampled from an one-dimensional, zero mean, unit-variance first-order Markov process with adjacent element correlation coefficient ρ, the covariance matrix of x is given by (2.1). The vector x is transformed into y by an orthogonal transform T, i.e. y=[T]x. The covariance matrix of the vector y in the transform domain $[COV_y(\rho)]$ is given as

$$\begin{aligned} [COV_y(\rho)] &= E[y\,y^t] \\ &= [T][COV_x(\rho)][T]^t \\ &= \begin{bmatrix} s_{0,0} & \cdots & \cdots & s_{0,N-1} \\ \cdot & \cdots & \cdots & \cdot \\ \cdot & \cdots & \cdots & \cdot \\ s_{N-1,0} & \cdots & \cdots & s_{N-1,N-1} \end{bmatrix} \end{aligned} \qquad (2.19)$$

When the off-diagonal terms of the covariance matrix $[COV_y(\rho)]$ are with small values, it means that the transform coefficients are weakly correlated. The Karhunen-Loeve transform (KLT), which used the eigenvectors of the covariance matrix of the input vector as its transform kernel, completely decorrelate the input vector. That is all off-diagonal terms in the covariance matrix $[COV_y(\rho)]$ are zero. Therefore, the KLT is said to be the optimum transform in terms of the properties of energy compaction and decorrelation.

2.3.2 Quantization

Before the coding and transmission operations, transform coefficients will be quantized to reconstructed levels with integer representation. This process introduces the unavoidable quantization error which is a significant factor of causing the degradation in decoded images. Normally, scalar quantization is adopted. A scalar quantizer maps a continuous variable u into a discrete variable û, which assumes values from a finite set $\{r_1, ..., r_L\}$ of numbers. The mapping is generally a staircase function as shown in Figure 2.2. The quantization rule is as follows : Define $\{t_k, k=1, ..., L+1\}$ as a set of increasing decision levels with t_1 and t_{L+1} as the minimum and maximum values of u respectively. If $t_k < u \leq t_{k+1}$ then it is mapped to r_k, the kth reconstruction level.

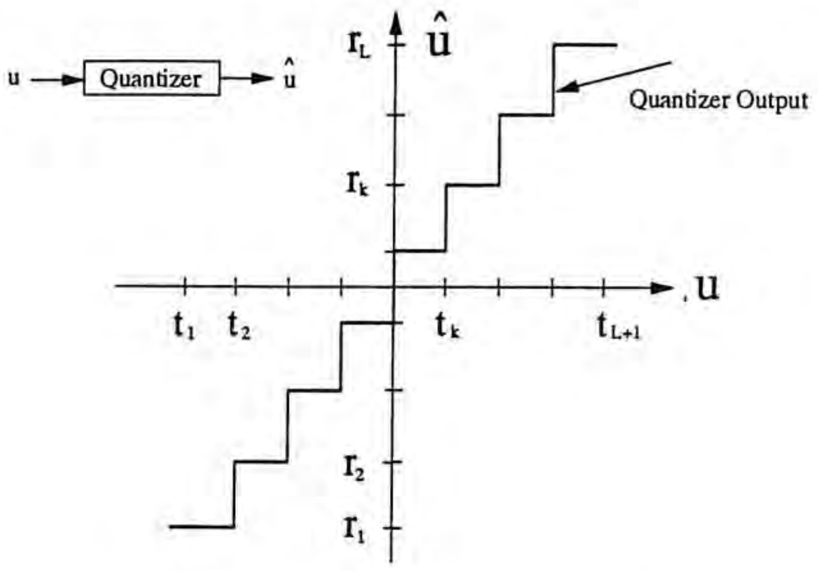


Figure 2.2 A quantizer

In 1960, Max [MAX60] derived the minimum mean square error (MMSE) quantizers for known input probability density functions and given number of quantization levels. Let u be a real scalar random variable with a continuous probability density function $p_u(u)$. It is desired to find the decision levels t_k and the reconstruction levels r_k for an L-level quantizer such that the mean square error

$$D = E\{(u-\hat{u})^2\} = \int_{t_1}^{t_{L+1}} (u-\hat{u})^2 p_u(u)\,du$$

$$= \sum_{i=1}^{L} \int_{t_1}^{t_{L+1}} (u-r_i)^2 p_u(u)\,du \tag{2.20}$$

is minimized. The results are obtained as follows :

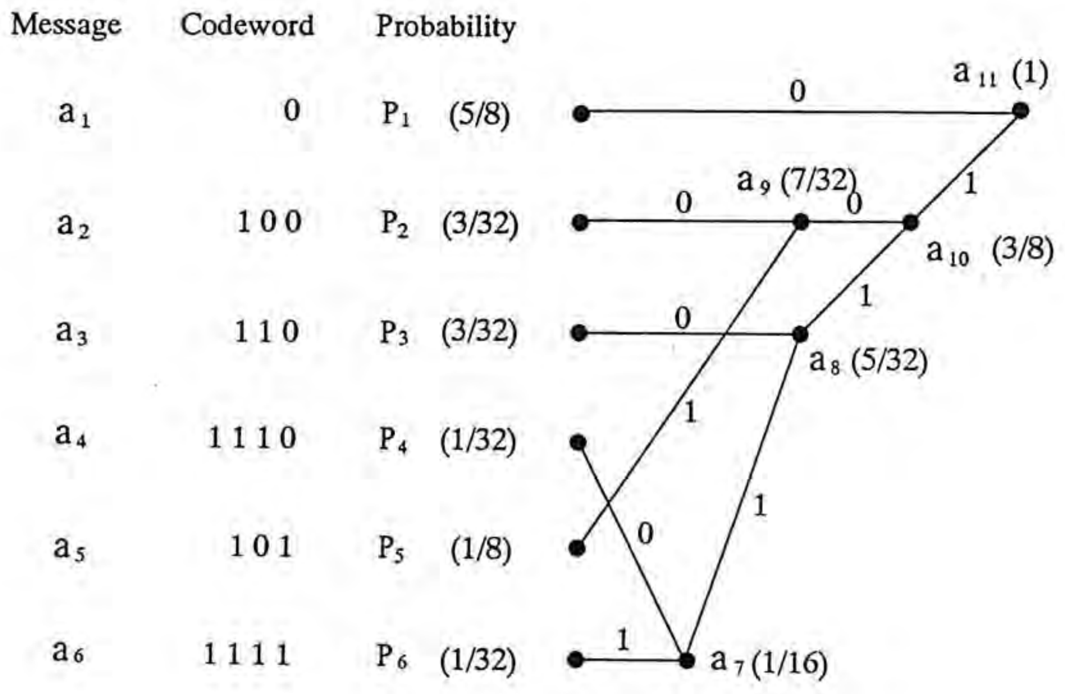
$$t_k = \frac{r_k + r_{k-1}}{2} \tag{2.21.a}$$

$$r_k = \frac{\int_{t_k}^{t_{k+1}} u\, p_u(u)\, du}{\int_{t_k}^{t_{k+1}} p_u(u)\, du} \qquad (2.21.b)$$

DC and AC transform coefficients have different characteristics, they are normally treated separately. Since DC coefficients contain the brightness information of image data, large error occurring in DC coefficients will probably cause visual discontinuity between blocks, that is so called blocking effect. To ensure reasonably good image quality, uniform quantizer is normally used for DC coefficients with fixed number of bits (8-bits) for each block of data. On the other hand, AC coefficients were found to have Laplacian distribution for most of the transforms [REINg83]. Hence, the Max quanitzer [MAX60] with Laplacian distribution [ADAMg78] [NOLLz79] could be used to quantize the AC coefficients. By using this quantizer, the decoded images will have the minimum distortion in terms of mean square error. As stated in section 2.2.2, the output images of an image coding system are to be viewed by human observers. Therefore, further improvement of picture quality can be obtained by taking the human visual system into account of the quantization process [BGGE83] [NGANls89] [JPEG90].

2.3.3 Coding

As the result of quantization, we obtain a specific reconstruction level for each transform coefficient. The next process is to assign each coefficient a codeword for transmission. A straightforward method is to assign each level with uniform length. Uniform codeword assignment, although simple, is not in general optimum in terms of the required bit rate. In most cases, there are some quantization levels are more likely to occur than others. Then if we assign shorter codewords to the more frequently occurring coefficients and longer codewords to the less frequently occuring coefficients, the overall bit-rate will be reduced. That is so called variable-length codeword assignment. Huffman has shown that Huffman coding is the optimum variable length coding scheme [HUFF52]. Many transform coding schemes [CHENp84] [NGANls89] [JPEG90] [WALL91] have utilized the Huffman coding for coding the quantized transform coefficients. Figure 2.3 shows an example of codeword generation in Huffman coding.



Message	Codeword	Probability
a_1	0	P_1 (5/8)
a_2	100	P_2 (3/32)
a_3	110	P_3 (3/32)
a_4	1110	P_4 (1/32)
a_5	101	P_5 (1/8)
a_6	1111	P_6 (1/32)

Figure 2.3 Illustration of codeword generation in Huffman coding [LIM90]

In the example, there are six messages and the probability of each message noted at each node as shown in Figure 2.3. In the first step of Huffman coding, we select the two messages that have the lowest probabilities, i.e. a_4 and a_6. We combine the probabilities of these two messages and form a new node with the combined probability. We now consider the two messages a_4 and a_6 as one message a_7 with a probability of (1/16). We now select another two messages having the lowest probabilities from among the five messages a_1, a_2, a_3, a_5, and a_7. The two messages chosen at that time are a_3 and a_7. They are again combined as one message, and '0' is assigned to one branch and '1' is assigned to the other. We can continue this process until we are left with one message with probability 1. To determine the specific codeword assigned to each message, we begin with the last node with probability 1, follow the branches that lead to the message of interest, and combine the 0s and 1s on the branches. For example, the message a_4 has the codeword 1110. Message with higher probability can clearly be seen to have shorter codewords, and the message with lower probability is with longer codewords.

2.3.4 JPEG International Standard

Recently, the Joint Photographic Experts Group (JPEG), which is a sub-working group of both the CCITT and ISO, is drafting an international standard for compression of still image. The proposed JPEG standard offers a continuous range of compression to reduce the transmission and storage requirements of large images by 10 to 100 times. The broad scope and the variety of targets to be achieved resulted in a three-part JPEG algorithm definition : the baseline system, the extended system, and the special function for lossless encoding. The baseline system is mandatory, while the extended system adds features such as sophisticated coding, lossless transmission, and progressive transmission. In this thesis, the term JPEG will indicate only the baseline system of JPEG scheme which utilizes the techniques of transform coding and DPCM coding.

JPEG scheme is intended to be used independently of the colour space. Each colour can be handled as a separate component. However, the best compression result can be achieved if the colour components are independent, such as YUV, where most of the information is concentrated in the luminance component (Y) and less in the chrominance components (U,V). Therefore, the R,G,B components of color images are first converted into Y,U,V using the transformation as follows [CCUBE90] :

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.3316 & 0.5 \\ 0.5 & 0.4186 & -0.0813 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.22}$$

Another advantage of using YUV colour space is that chrominance components (U and V) need not be specified as frequently as the luminance component (Y). Quality of an image is not much affected if the vertical resolution of the U and V components is reduced to half of that of Y. As a result, a further data reduction of 3/2 is obtained when transforming RGB into YUV. After colour space conversion, each component can then be independently processed by the JPEG system. A block diagram of the baseline JPEG system is shown in Figure 2.4.

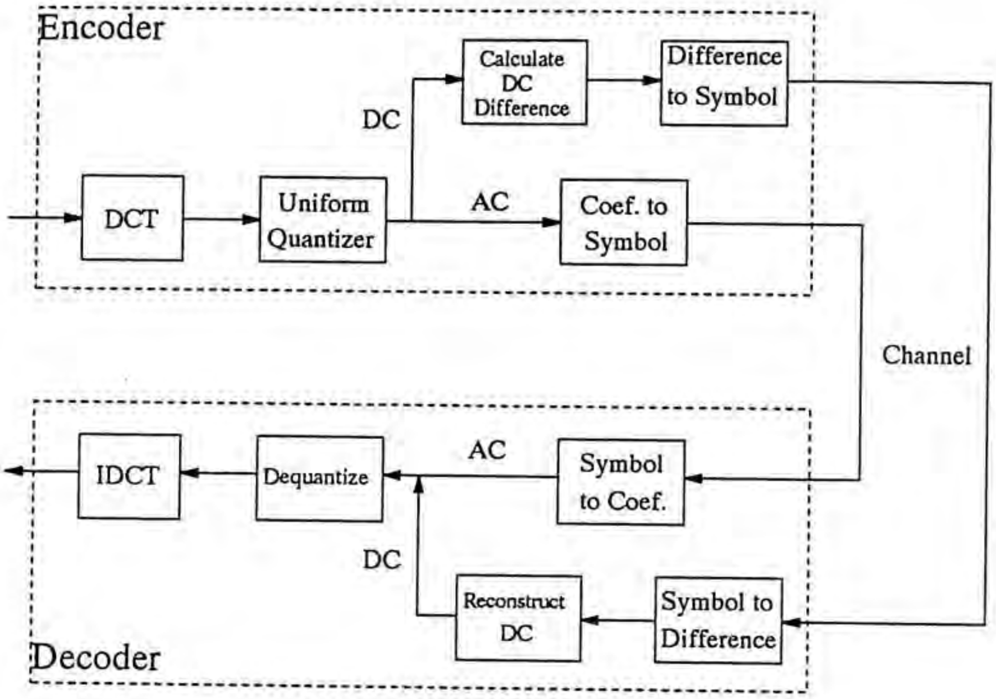


Figure 2.4 Block Diagram of the JPEG baseline System

Both JPEG encoder and decoder consist of three parts which are the DCT, quantization and Huffman coding. They are briefly described as follows :

a. Level Shift and Discrete Cosine Transform

At the encoder, each component of the input image is first divided into non-overlapping 8x8 blocks. All input pixels are level shifted by subtracting 2^{p-1}, where p is the precision of image pixel and usually equals to 8. This is to shift the representation of input block data from unsigned integers with range $[0,2^p-1]$ to signed integers with range $[-2^{p-1},2^{p-1}-1]$. Then each block undergoes two-dimensional discrete cosine transform to convert the pixel elements into coefficients. At the decoder, the dequantized coefficients will go through the inverse discrete cosine transform (IDCT) to reconstruct the image. Finally, 2^{p-1} are added to the pixels to obtain the reconstructed image.

b. Quantization and Dequantization:

After output from the two-dimensional DCT, each of the 64 transform coefficients is uniformly quantized in conjunction with a 64-element quantization matrix. Two default quantization matrices, one for the luminance component and other for chrominance component, are specified in the baseline JPEG scheme [JPEG90] and are as follow :

Q_1 = quantization matrix for luminance component

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

Q_2 = quantization matrix for chrominance component

$$= \begin{bmatrix} 17 & 18 & 24 & 47 & 66 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

Quantization is defined as division of each DCT coefficient y(p,q) by its corresponding quantizer step size, followed by rounding to the nearest integer :

$$y_q(p,q) = \left\lfloor \frac{y(p,q)}{Q(p,q)} \right\rfloor \qquad (2.23.a)$$

where $y_q(p,q)$ is the quantized DCT coefficients

$Q(p,q)$ is the quantization value for $(p,q)^{th}$ coefficient

$\lfloor\cdot\rfloor$ is the rounding operator

At the decoder, dequantization is similarly carried out to obtain the dequantized transform coefficients :

$$\tilde{y}(p,q) = y_q(p,q)Q(p,q) \tag{2.23.b}$$

where $\tilde{y}(p,q)$ is the dequantized DCT coefficient.

c. Coefficient Coding

At the encoder, after coefficient quantization, the two-dimensional array of the quantized AC coefficients is first rearranged into an one-dimensional array, using a common zigzag scan ordering as shown in Figure 2.5. Each non-zero quantized AC coefficients in the vector of zigzag ordered coefficients is then encoded in combination with the length, or so-called run-length, of preceding zero valued AC coefficients. The composite value of run-length and amplitude of the quantized AC coefficient is then Huffman coded. For the DC coefficients, an one-dimensional predictor is used. For blocks of each row, the predicted value of the DC coefficient of current $(k,l)^{th}$ block is simply the DC coefficient of previous $(k,l-1)^{th}$ block, i.e.

$$D\hat{C}_{k,l} = DC_{k,l-1} \tag{2.24}$$

The difference $d_{k,l}$ between predicted DC coefficient and quantized DC coefficient is then Huffman coded.

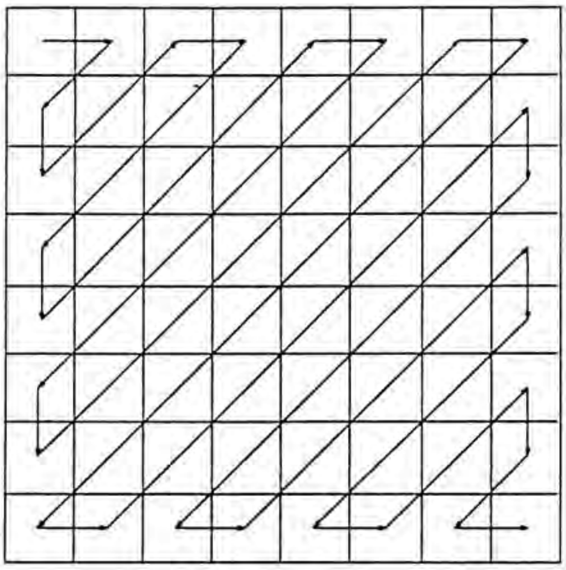


Figure 2.5 Zigzag scan path

2.4 VECTOR QUANTIZATION THEORY

The independent quantization of each signal value or parameter is called scalar quantization, while the joint quantization of a block of parameters is termed vector quantization. Let $x = (x_1, x_2, ..., x_k)$ be a k-dimensional input vector that consists of k real-valued, continuous-amplitude scalars x_p. A vector quantizer can be defined as a mapping Q of k-dimensional Euclidean space R^k into a finite subset Y of R^k. Thus

$$Q : R^k \rightarrow Y \tag{2.25}$$

where $Y = (y_i; i=1, 2, ..., M)$ is the set of codewords called codebook and M the number of codewords in Y. For waveform coding, the codewords are also k-dimensional real-valued vectors. A distortion measure between the input vector x and the codeword y_i, i.e. $d(x,y_i)$, should be used as a criterion to see how good the mapping is. Normally, Euclidean distortion between the two vectors is used. The block diagram of a simple vector quantizer is shown in Figure 2.6. At the encoder, the distortion between the input vector x and each codeword in the codebook should be calculated. The optimum encoding rule is the nearest neighbour rule, in which the index I is transmitted to the decoder if the codeword y_I yield the minimum distortion. However, the decoder simply looks up the I^{th} codeword y_I from a copy of the codebook Y, and then the output vector is y_I. In the encoding process, the computation grows exponentially with M and k so high-rate or high-dimensionality vector quantizers are impractical.

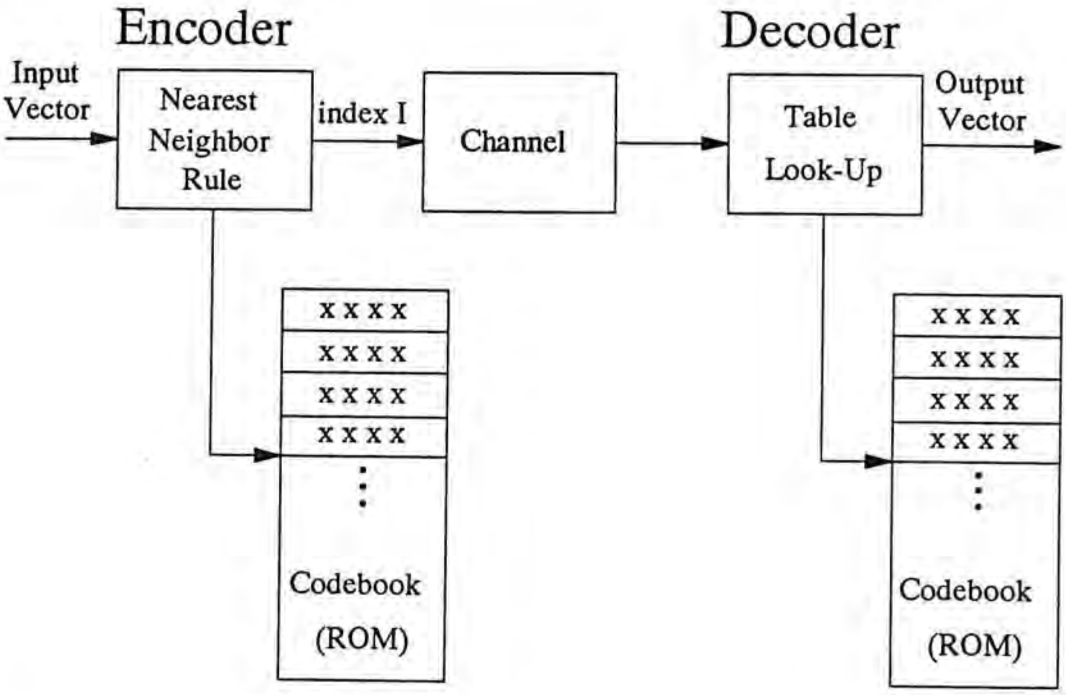


Figure 2.6 Block diagram of a simple vector quantization coding system

Vector quantization [GRAY84] [LIM90] can exploit the statistical dependence and non-linear dependence of the signal. When compared with the conventional scalar quantization scheme, vector quantization can lower the average distortion with the number of reconstruction levels held constant or can reduce the required number of reconstruction levels when the average distortion held constant. In other words, it has a better rate-distortion performance than the scalar quantization. Both vector quantization and transform coding are block-based image coding techniques. In comparison with transform coding, vector quantization has a simpler mapping process but a rather complicated quantization process at the encoder. For the mapping process, vector quantization requires only to convert a two-dimensional array into an one-dimensional one while transform coding needs to perform a two-dimensional transformation process. On the other hand, vector quantization system has simpler decoder design than that of transform coding. Similar system performance in terms of reconstruction error and compression ratio of the two coding methods are obtained.

2.4.1 Codebook Design and the LBG Clustering Algorithm

In vector quantization, the same codebook should be used at both encoder and decoder. The objective of designing an optimum codebook is to obtain a codebook consisting of M reproduction vectors, such that it minimizes the average distortion d(x,Q(x)), where

$$\begin{aligned} d(x,Q(x)) &= d(x,\tilde{x}) \\ &= [x-\tilde{x}][x-\tilde{x}]^t \end{aligned}$$

and $\tilde{x}$ or Q(x) is the reconstructed vector. The design of an optimum codebook is a non-linear problem. Attempts to solve this problem usually utilizes the following two necessary conditions [LIM90] for its solution.

Condition 1 For an input vector x to be quantized to one of the reconstructed levels, the optimum quantizer must choose the reconstructed levels y_i which has the minimum distortion to x :

$$Q(x) = y_i \quad \text{if and only if} \quad d(x,y_i) \le d(x,y_j) \quad \text{for} \quad i \ne j \text{ and } 1 \le j \le M \qquad (2.26)$$

Condition 2 Each reconstructed level y_i must minimize the average distortion D in the corresponding cell S_i :

$$\text{Minimize} \ \ E\ \{\ d(x,y_i)\ \mid\ x \in S_i\ \} \ \ \text{with respect to } y_i \qquad (2.27)$$

where S_i is the i^{th} Voronoi region in the training set. The level y_i that satisfies (2.27) is called the centroid of S_i.

The two necessary conditions suggest an iterative procedure for designing an optimum codebook. A well known iterative algorithm for designing the codebook is the LBG clustering algorithm [LINDbg82] which is depicted by a simplified flowchart as shown in Figure 2.7.

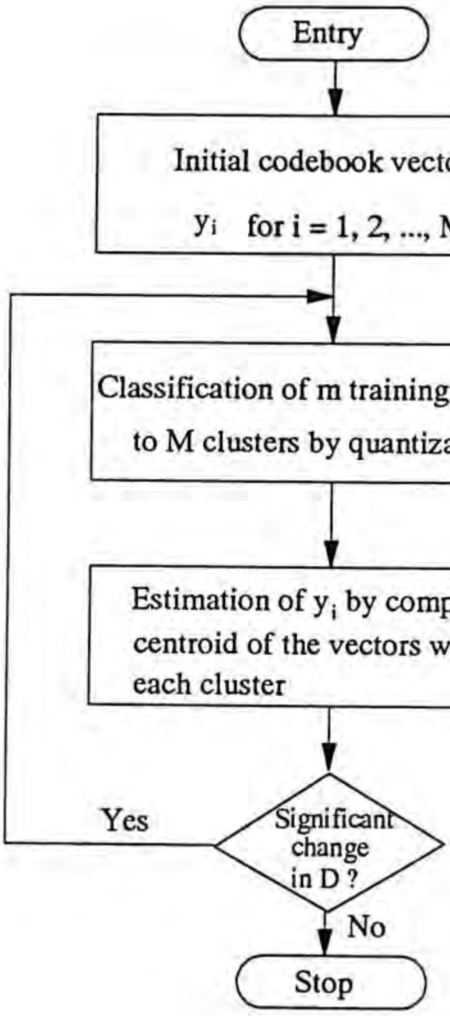

Entry
Initial codebook vectors
y_i for i = 1, 2, ..., M
Classification of m training vectors to M clusters by quantization
Estimation of y_i by computing centroid of the vectors within each cluster
Significant change in D ?
Yes
No
Stop

Figure 2.7 LBG Algorithm for codebook design

Suppose we have m training vectors denoted by x_i for $1 \leq i \leq m$. Since we estimate L reconstructed levels from m training vectors, we assume that $m \gg L$. The reconstructed levels y_i are determined by minimizing the average distortion D, given by

$$D = \frac{1}{m} \sum_{i=1}^{m} d(x_i, \tilde{x}_i) \tag{2.28}$$

In the LBG algorithm, an initial estimate of y_i is first made. Then the m training vectors are classified into L different groups or cluster corresponding to each reconstruction level by using condition 1 in (2.26). A new reconstruction level is determined from the vectors in each cluster. For convenience, suppose x_i for $1 \le i \le m_1$ are m_1 training vector quantized to the first reconstruction level. The new estimate of y_1 is obtained by finding the centroid of the m_1 training vectors as (2.27). A new estimate of all other reconstruction level y_i for $2 \le i \le L$ is similarly obtained. This completes one iteration of the algorithm, which can be stopped if the average distortion D does not change significantly between two consecutive iterations.

2.5 BLOCK TRUNCATION CODING THEORY

The block truncation coding (BTC) technique [DELPm79] for image compression has the advantage of simpler implementation complexity when compared with the two other block-based compression methods : transform coding and vector quantization. However, the compression ratio of BTC is less than those of transform coding and VQ. When using the BTC, each pixel in a block is truncated to one bit by thresholding and moment preserving selection of reconstruction levels. In addition to the one bit information which has to be transmitted, one also has to send information that would allow the receiver to reconstruct the data with the same moments as the original block. With moment preserving in local area of the image, it is possible to improve the visual quality of the decoded images especially in the edge areas. Consider a block of data x which is of size k=4x4. The two-dimensional block is then linear scanned to form an one-dimensional array. The mean or the first moment of the block is equal to

$$\overline{x} = \frac{1}{k} \sum_{i=1}^{k} x(i) \tag{2.29}$$

To generate the moment preserving reconstructed levels, we compare each pixel with the mean $\overline{x}$. If $x(i) < \overline{x}$, we set it to ***a***, while $x(i) \geq \overline{x}$, we set it to ***b***. The values of ***a*** and ***b*** are found in such a way that the first and second moment of the block are preserved. The second moment of the block is given by

$$\overline{x^2} = \frac{1}{k} \sum_{i=1}^{k} x(i)^2 \tag{2.30}$$

If q is the number of pixels larger or equal to the sample mean, the first and the second moments are then given by

$$\overline{x} = \frac{1}{k}[(k\text{-}q)a + qb] \tag{2.31}$$

$$\overline{x^2} = \frac{1}{k}[(k\text{-}q)a^2 + q\,b^2] \tag{2.32}$$

In order to preserve the first two moments of the block, we equate (2.31) with (2.29) and (2.32) with (2.30). The values of ***a*** and ***b*** are then obtained as

$$a = \overline{x} - \sigma_x \sqrt{\frac{q}{k\text{-}q}} \tag{2.33.a}$$

$$b = \overline{x} + \sigma_x \sqrt{\frac{k\text{-}q}{q}} \tag{2.33.b}$$

where σ_x is the standard deviation of the block x and is given by

$$\sigma_x = \sqrt{\overline{x^2} - (\overline{x})^2}$$

2.5.1 Optimum MSE Block Truncation Coding

Obviously, the above described moment-preserved two-level quantizer is not optimum in the sense of minimizing the mean square error (MSE). Recently, Hui [HUI89] developed an iterative algorithm for designing the optimum two-level BTC algorithm based on those algorithms [MAX60] [Lloyd82] [WU91] proposed for designing the minimum distortion scalar quantizer. Consider a two-level quantizer described by the output levels {a,b} and a threshold t such that the output of the quantizer is equal to ***a*** if x(i) < t and ***b*** if x(i) ≥ t, the expected distortion D using the square error criterion can be denoted as

$$D(\{a,b\}, t) = \sum_{x(i)<t} (x(i)-a)^2 + \sum_{x(i)\geq t} (x(i)-b)^2 \tag{2.34}$$

the optimum threshold t' for {a,b} is one that minimizes the distortion given in (2.34) and can be easily constructed by

$$t' = \frac{(a+b)}{2} \tag{2.35}$$

hence, for any threshold t,

$$D(\{a,b\}, t) \geq D(\{a,b\}, \frac{(a+b)}{2}) \tag{2.36}$$

On the other hand, when the threshold of quantizer is t, the optimum output quantization levels a' and b' can then be obtained by minimizing D({a,b}, t) with fixed t. The results are

$$a' = \frac{1}{k-q} \sum_{x(i)<t} x(i) \tag{2.37.a}$$

$$b' = \frac{1}{q} \sum_{x(i)\geq t} x(i) \tag{2.37.b}$$

This imply that

$$D\left(\{a,b\},t\right) \geq D\left(\{a',b'\},t\right) \tag{2.38}$$

According to the conditions in (2.36) and (2.38), we can design a two-level quantizer by iteratively optimizes the threshold t for the output level {a,b} and vice versa. The algorithm is illustrated in the flowchart shown in Figure 2.8. Initially, the minimum value x_{min} and the maximum value x_{max} of the block are used as initial values of the output levels. The initial threshold is simply equal to (a+b)/2. With this initial threshold, we can calculate the new values of the output levels according to (2.37). The new values of ***a*** and ***b*** are then compared with the old one to see whether they have stopped converging. If not, ***a*** and ***b*** will be used to compute a new threshold value t and the whole process is repeated iteratively until minimum mean-square error is achieved.

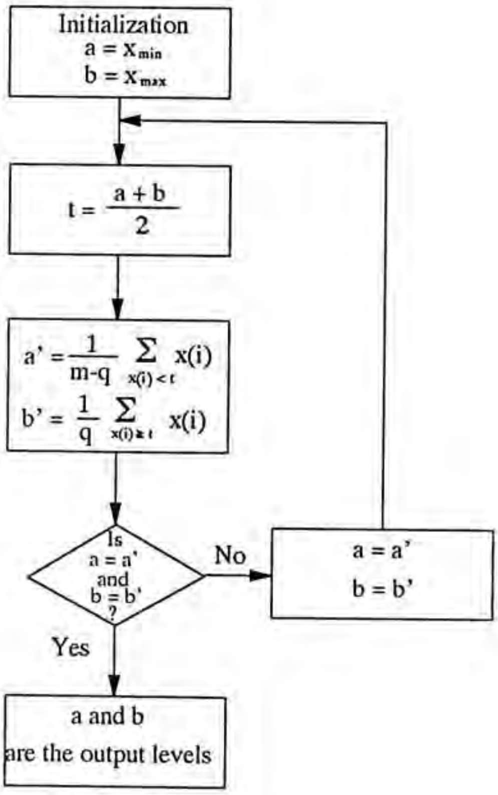

Initialization
$a = x_{min}$
$b = x_{max}$
$t = \frac{a+b}{2}$
$a' = \frac{1}{m-q} \sum_{x(i)<t} x(i)$
$b' = \frac{1}{q} \sum_{x(i)\geq t} x(i)$
Is a = a' and b = b' ?
No
Yes
a = a'
b = b'
a and b are the output levels

Figure 2.8 Algorithm for designing an optimum MSE two-level quantizer

CHAPTER 3 DEVELOPMENT OF NEW ORTHOGONAL TRANSFORMS

In this chapter, two new orthogonal transforms, called weighted cosine transform (WCT) and simplified cosine transform (SCT) are developed for transform coding of images. These two new transforms are derived with different aims. WCT is designed to have better performance than the discrete cosine transform (DCT), which is regarded as the industrial standard in transform coding systems, whilst maintaining similar implementation complexity. On the other hand, SCT is designed to have simpler structure than the DCT while maintaining close performance with the DCT.

3.1 INTRODUCTION

In transform coding of images, the primary objective of the orthogonal transformation is to convert statistically dependent picture elements into an array of uncorrelated coefficients such that most of the energy is packed into a minimum number of coefficients. Therefore, an orthogonal transform is said to be optimum in transform coding system when it completely decorrelates the image data. Karhunen-Loeve transform (KLT) is the optimum transform [ROSEk82] [JAIN89] and its basis vectors are eigenvectors of the covariance matrix of the incoming image data. KLT is image dependent since different images have different covariance matrices and so different KLTs. As the first-order Markov process of covariance matrix given by eqn. (2.1) is a good and simple mathematical model for image data, the KLT of this model received much attention in image compression. The explicit form of the KLT has been found [RAYd70], however, its fast computational algorithm is still not known. Because of the computational difficulty of the KLT, it is not used for a practical image coding system.

The problems of KLT mentioned before can be solved by the application of sub-optimal transforms which have fast computational algorithms. The first sub-optimal transform to be investigated for transform coding was two-dimensional discrete Fourier transform

[ANDRp68]. This was followed by the discovery that the Walsh transform could be utilized in place of the discrete Fourier transform with a considerable saving of computational loading [PRATk69]. Later, the Haar transform (HT) [ANDR70] and the Slant transform (ST) [PRATcw72] were developed by modifying the Walsh matrix. These transforms have simpler structure for implementation but lower energy packing ability than the KLT. The discovery of the DCT [AHMEnr74] in 1974 later was shown to be very important for image transform coding. The DCT is asymtotically equivalent to the KLT for the first-order Markov signals [HAMIp76] and Clarke has shown that when the adjacent element correlation coefficient is close to one, the KLT is in fact a DCT [CLAR81]. Since then, the trend seemed to be directed to the study of sinusoidal transforms. In 1976, Jain [JAIN76] proposed a sinusoidal transform called discrete sine transform (DST) which is the KLT of a first-order Markov process under the condition that the boundary of the process is known. In the following years, several sinusoidal transforms were proposed and investigated for image coding [JAIN79] [KITA80]. Wang [WANG84] provided a systematic classification for this family of trigonometric transforms. There are four types of cosine transforms and four types of sine transforms. Table 3.1 summarizes the two classification schemes of sinusoidal transforms proposed by Jain [JAIN76] and Wang [WANG84]. It is noted that EDCT-1 and DCT-II are the same transform, which is also the commonly used version of cosine transform. From now on, we shall use the classification and notation of Wang for representing the sinusoidal transforms. Also, the abbreviations DCT represents DCT-II.

Since then, many different kinds of orthogonal transforms have been proposed. When selecting an orthogonal transform for an image coding system, computational complexity and compression ability are the two most important criteria for evaluating the performance of the transform. A good transform should have low computational complexity and high compression ability which, however, usually cannot be achieved simultaneously. Consequently, there exist two different streams in the development of new orthogonal transforms. Considering that the advancement of in VLSI technology will continue to lessen the constraints in implementation, one stream is to search for transforms having performance better than the DCT. Such transforms, like the DCT, have fast computational algorithms but require more multiplication and addition operations. Typical examples of this stream are the phase shift cosine transform (PSCT) [WANG86] and the modified cosine transform (MDCT) [HUANkt90]. The other stream is to search for transforms requiring simple implementation and having similar performance to the DCT. Examples of transforms belonging to this stream are the C-matrix transform (CMT) [JONEhk78] [SRINr83] [KWAKsr83], the high correlation transform (HCT) and the low correlation transform (LCT) [CHAMc86], integer cosine transform (ICT) [CHAM89] [CHAMc91] and the Hadamard-structured discrete cosine transform (HDCT) [KOUm89]. Based on the above philosophies, two new orthogonal transforms called the weighted cosine transform (WCT) and the simplified cosine transform (SCT) are developed in this chapter.

Transform	Transform Kernel
Even Cosine - 1 (EDCT-1)	$\cos\left(i\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N-1]$
Even Cosine - 2 (EDCT-2)	$\cos\left(\left(i+\frac{1}{2}\right)\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N-1]$
Odd Cosine - 1 (ODCT-1)	$\cos\left(\left(i+\frac{1}{2}\right)\left(j+\frac{1}{2}\right)\frac{2\pi}{2N+1}\right)$ for $i,j \in [0,\ldots,N-1]$
Even Sine - 1 (EDST-1)	$\sin\left((i+1)(j+1)\frac{\pi}{N+1}\right)$ for $i,j \in [0,\ldots,N-1]$
Even Sine - 2 (EDST-2)	$\sin\left((i+1)\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N-1]$
Even Sine - 3 (EDST-3)	$\sin\left(\left(i+\frac{1}{2}\right)\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N-1]$
Odd Sine - 1 (ODST-1)	$\sin\left(\left(i+\frac{1}{2}\right)(j+1)\frac{2\pi}{2N+1}\right)$ for $i,j \in [0,\ldots,N-1]$
Odd Sine - 2 (ODST-2)	$\sin\left((i+1)(j+1)\frac{2\pi}{2N+1}\right)$ for $i,j \in [0,\ldots,N-1]$
Odd Sine - 3 (ODST-3)	$\sin\left((i+1)\left(j+\frac{1}{2}\right)\frac{2\pi}{2N+1}\right)$ for $i,j \in [0,\ldots,N-1]$

a. Jain's Classification [JAIN76]

Transform	Transform Kernel
Type I Cosine (DCT-I)	$\cos\left(ij\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N]$
Type II Cosine (DCT-II)	$\cos\left(i\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N-1]$
Type III Cosine (DCT-III)	$\cos\left(\left(i+\frac{1}{2}\right)j\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N-1]$
Type IV Cosine (DCT-IV)	$\cos\left(\left(i+\frac{1}{2}\right)\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N-1]$
Type I Sine (DST-I)	$\sin\left((i+1)(j+1)\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N-2]$
Type II Sine (DST-II)	$\sin((i+1)\left(j+\frac{1}{2}\right)\frac{\pi}{2N}$ for $i,j \in [0,\ldots,N-1]$
Type III Sine (DST-III)	$\sin\left(\left(i+\frac{1}{2}\right)(j+1)\frac{2\pi}{N}\right)$ for $i,j \in [0,\ldots,N-1]$
Type IV Sine (DST-IV)	$\sin\left(\left(i+\frac{1}{2}\right)\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right)$ for $i,j \in [0,\ldots,N-1]$

b. Wang's classification [WANG84]

Table 3.1 Different classification schemes for sinusoidal transforms

In section 3.2 and 3.3, development of the weighted cosine transform (WCT) and the simplified cosine transform (SCT) will first be described. The fast computational algorithms for computing these two new transforms will then be given in section 3.4. The performance of the two new transforms in transform coding of images will then be evaluated in section 3.5 by using both statistical model and real images. Finally, concluding remarks will be given in section 3.6.

3.2 WEIGHTED COSINE TRANSFORM (WCT)

Among the existing transforms in image coding, the DCT stands out to be the best sub-optimal transform. Over the past decade, with the fast development in VLSI technologies, the complexity of a codec in some cases becomes a less important factor than the performance. An interesting question thus arose. Can the performance of the DCT be further improved ? In other words, can the DCT be modified so that it approaches the optimum KLT performance faster for all values of adjacent element correlation coefficient ? Wang was the first to answer this question by introducing a little phase change to the even part of the DCT [WANG86]. The modified version for the DCT, called the phase shift cosine transform (PSCT), has better performance than the DCT for adjacent element correlation coefficient close to unity. The transform kernel $G_N(i,j)$ of the PSCT is given by

$$G_N(i,j) = \frac{2}{\sqrt{N}}\, k_i \cos\frac{i(2j+1)\pi}{2N} \sin\left[\frac{\pi}{4} - (-1)^i \left(\frac{N}{2} - j - \frac{1}{2}\right)\alpha\right] \qquad (3.1)$$

for i,j = 0, 1, ..., N-1

where $$k_i = \begin{cases} \frac{1}{\sqrt{2}} & i = 0 \\ 1 & i \neq 0 \end{cases}$$

Eqn. (3.1) shows that the PSCT matrix is obtained by weighting the DCT matrix with a sine function. With a few more multiplication and addition operations required, PSCT can also be computed via fast algorithms for the DCT. Recently, Huang et al. [HUANkt90] proposed the use of principle of dyadic symmetry to modify the DCT. This technique generates a number of order-8 transforms called the MDCT which also show improved performance over the DCT. Difference between MDCT and the DCT lies mainly in the odd part of the transform matrix. Fast computational algorithm for MDCT however is not known yet. Wang and Huang's works show that modification of either even or odd part of the DCT can result transforms of better performance. Hence, we propose to modify the DCT by weighting both the even and odd parts of the DCT with simple structured orthonormal matrices. The use of these simple matrices ensures that the weighted cosine transform (WCT) thus formed can be computed via existing fast DCT algorithms. A WCT of better performance than the DCT, PSCT and MDCT is found.

3.2.1 Development of the WCT

According to the classification of Wang [WANG84], an order-N DCT-II can be decomposed into two order-N/2 transforms, the odd part is DCT-IV, and the even part is DCT-II [WANG86] as shown in the following matrix equation.

$$[C_N^{II}] = [P_N]\begin{bmatrix} C_{N/2}^{II} & 0 \\ 0 & \bar{I}C_{N/2}^{IV}\bar{I} \end{bmatrix}[B_N] \tag{3.2}$$

where C_N^A represents the order-N type A cosine transform and

$$[B_N] = \frac{1}{\sqrt{2}}\begin{bmatrix} I_{N/2} & \bar{I}_{N/2} \\ \bar{I}_{N/2} & -I_{N/2} \end{bmatrix}$$

$$[P_N]^t = \begin{bmatrix} 1 & 0 & 0 & . & . & . & . & . & . \\ 0 & 0 & 1 & 0 & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & 0 & 1 & 0 \\ 0 & 1 & 0 & . & . & . & . & . & . \\ 0 & 0 & 0 & 1 & 0 & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & . & 0 & 1 \end{bmatrix}$$

I is the identity matrix and $\bar{I}$ is the anti-diagonal identity matrix. It is seen that a new orthonormal transform matrix can be obtained by replacing the DCT-II and DCT-IV matrix in eqn.(3.2) with two other orthonormal matrices. An order-N weighted cosine transform (WCT) is defined as

$$[WCT_N] = [P_N]\begin{bmatrix} G_{N/2} & 0 \\ 0 & \bar{I}U_{N/2}\bar{I} \end{bmatrix}[B_N] \tag{3.3}$$

G_M, the even part of the WCT matrix, is the transform matrix of PSCT [WANG86] whose transform kernal is given by eqn. (3.1).

U_M, the odd part of the WCT matrix, is obtained by weighting DCT-IV matrix with another sine function.

$$U_M(i,j)=\frac{2}{\sqrt{M}}\cos\frac{(2i+1)(2j+1)\pi}{4M}\sin\left[\frac{\pi}{4}-(-1)^i f_M(i)f_M(j)\left(\frac{M}{4}-\frac{1}{2}-\langle j\rangle_{M/2}\right)\beta\right] \qquad (3.4)$$

for $i,j = 0, 1, \ldots, M-1$ and $M = N/2$

where $\langle m\rangle_n$ means the residue of m mod n and

$$f_M(x)=\begin{cases}1, & x\le\frac{M}{2}-1\\ -1, & \textit{otherwise}\end{cases}$$

Case of order-8

For an order-8 WCT, where N=8 and M=4, we have

$$[G_4]=[C_4^{II}]\ominus\begin{bmatrix}a & b & c & d\\ d & c & b & a\\ a & b & c & d\\ d & c & b & a\end{bmatrix}$$

where $\ominus$ means direct multiplication of matrices and

$$a=\sin\left(\frac{\pi}{4}-\frac{3}{2}\alpha\right),\quad b=\sin\left(\frac{\pi}{4}-\frac{1}{2}\alpha\right),\quad c=\sin\left(\frac{\pi}{4}+\frac{1}{2}\alpha\right)\quad\text{and}\quad d=\sin\left(\frac{\pi}{4}+\frac{3}{2}\alpha\right)$$

$$[U_4]=[C_4^{IV}]\ominus\begin{bmatrix}e & f & f & e\\ f & e & e & f\\ f & e & e & f\\ e & f & f & e\end{bmatrix}$$

where

$$e=\sin\left(\frac{\pi}{4}-\frac{1}{2}\beta\right)\quad\text{and}\quad f=\sin\left(\frac{\pi}{4}+\frac{1}{2}\beta\right)$$

and the order-8 WCT matrix is then given by

$$[\mathrm{WCT}_8] = [\mathrm{P}_8]\begin{bmatrix} \mathrm{G}_4 & 0 \\ 0 & \bar{\mathrm{I}}\mathrm{U}_4\bar{\mathrm{I}} \end{bmatrix}[\mathrm{B}_8]$$

Case of order-16

For an order-16 WCT, where N=16 and M=8, we have

$$[\mathrm{G}_8] = [\mathrm{C}_8^{\mathrm{II}}] \ominus \begin{bmatrix} a & b & c & d & e & f & g & h \\ h & g & f & e & d & c & b & a \\ a & b & c & d & e & f & g & h \\ h & g & f & e & d & c & b & a \\ a & b & c & d & e & f & g & h \\ h & g & f & e & d & c & b & a \\ a & b & c & d & e & f & g & h \\ h & g & f & e & d & c & b & a \end{bmatrix}$$

$a = \sin\left(\frac{\pi}{4} - \frac{7}{2}\alpha\right)$, $b = \sin\left(\frac{\pi}{4} - \frac{5}{2}\alpha\right)$, $c = \sin\left(\frac{\pi}{4} - \frac{3}{2}\alpha\right)$, $d = \sin\left(\frac{\pi}{4} - \frac{1}{2}\alpha\right)$

$e = \sin\left(\frac{\pi}{4} + \frac{1}{2}\alpha\right)$, $f = \sin\left(\frac{\pi}{4} + \frac{3}{2}\alpha\right)$, $g = \sin\left(\frac{\pi}{4} + \frac{5}{2}\alpha\right)$ and $h = \sin\left(\frac{\pi}{4} + \frac{7}{2}\alpha\right)$

$$[\mathrm{U}_8] = [\mathrm{C}_8^{\mathrm{IV}}] \ominus \begin{bmatrix} p & q & r & s & s & r & q & p \\ s & r & q & p & p & q & r & s \\ s & r & q & p & p & q & r & s \\ p & q & r & s & s & r & q & p \\ p & q & r & s & s & r & q & p \\ s & r & q & p & p & q & r & s \\ s & r & q & p & p & q & r & s \\ p & q & r & s & s & r & q & p \end{bmatrix}$$

where

$p = \sin\left(\frac{\pi}{4} - \frac{3}{2}\beta\right)$, $q = \sin\left(\frac{\pi}{4} - \frac{1}{2}\beta\right)$, $r = \sin\left(\frac{\pi}{4} + \frac{1}{2}\beta\right)$ and $s = \sin\left(\frac{\pi}{4} + \frac{3}{2}\beta\right)$

then the order-16 WCT matrix is given by

$$[\mathrm{WCT}_{16}] = [\mathrm{P}_{16}]\begin{bmatrix} \mathrm{G}_8 & 0 \\ 0 & \bar{\mathrm{I}}\mathrm{U}_8\bar{\mathrm{I}} \end{bmatrix}[\mathrm{B}_{16}]$$

3.2.2 Determination of α and β

In the transform matrix of the WCT, there are two parameters α and ß as defined by eqns.(3.1) and (3.4) dependent on the order N and the correlation coefficient ρ of the signal to be transformed. Both α and ß are positive values and are in the range of [0, π / (N-2)]. The optimum values of α and ß for each value of N and ρ under the criterion of minimizing the geometric mean of the variance of the transform coefficients [WANGg88], which is proportional to the reconstruction error of an optimum bit allocation transform coding system, are found using exhaustive search and listed in Table 3.2. We found that the WCT optimized at ρ=0.95 maintains good performance for other values of ρ. From now on, WCT refers to the one with (α, ß) equal to (0.0237, 0.0231) and (0.021, 0.0184) for N equal to 8 and 16 respectively. WCT can achieve even better performance if the parameter (α, ß) adapts to the adjacent element correlation of input image data, however, such adaptation requires too much computation.

ρ	0.95	0.90	0.85	0.80
N=8	0.0237	0.0444	0.0621	0.0765
N=16	0.0210	0.0343	0.0407	0.0420

a. Values of α

ρ	0.95	0.90	0.85	0.80
N=8	0.0231	0.0453	0.0663	0.0860
N=16	0.0184	0.0329	0.0435	0.0502

b. Values of ß

Table 3.2 Best values of α and ß for the WCT

3.3 SIMPLIFIED COSINE TRANSFORM (SCT)

In most industrial applications, the hardware cost of an image transform coding system is an important factor. This motivates the works for finding simpler transforms as substitutes for the DCT at the expense of compression ability. In the last two decades, a number of non-sinusoidal transforms have been developed for image coding systems. Most non-sinusoidal transforms consist of only integer components but have lower compression ability than sinusoidal transforms. The C-matrix transform (CMT) [JONEhk78] [SRINr83] [KWAKsr83] has found application in image coding for its simpler structure than the DCT since only integers are contained in the transform matrix. Cham and Clarke generated two orthogonal transforms, the high correlation transform (HCT) and the low correlation transform (LCT), by using the principle of dyadic symmetry [CHAMc86]. Computations of these two transforms require only addition, substraction and binary shift operations. The compression ability of the two transforms both lied between the Walsh-Hadamard transform and that of the DCT. HCT and LCT can therefore be used to replace the Walsh-Hadamard transform in image coding systems. Since then, Cham extended his work and developed the integer cosine transform (ICT) [CHAM89] [CHAMc91] [KOHht91] from the DCT. It has been shown that the performance of ICT is better than those of the CMT, HCT and LCT and some order-8 ICTs even have higher transform efficiency than the DCT. In this section, a new orthogonal transform, called simplified cosine transform (SCT), is developed to have simpler computation complexity than the DCT while maintaining similar performance with the DCT and ICT. The transform kernel of SCT contains sinusoidal and non-sinusoidal parts. The odd part of the SCT matrix, which contains the sinusoidal part, is the same as the DCT-IV matrix while the even part, which contains the integer part, is a simple structured orthogonal matrix which can be computed using only addition and binary shift operations. It will be shown that SCT performs close to the DCT and even has better compression ability than ICT when the adjacent element correlation is equal to 0.95. SCT requires fewer operations than the DCT.

3.3.1 Development of the SCT

An order-N simplified cosine transform (SCT) is defined as

$$[\mathrm{SCT}_N] = [P_N]\begin{bmatrix} T_{N/2} & 0 \\ 0 & \bar{I}C^{IV}_{N/2}\bar{I} \end{bmatrix}[B_N] \tag{3.5}$$

where T is an orthogonal matrix defined as

$$[T_M] = [P_M]\begin{bmatrix} D_{M/2} & 0 \\ 0 & \bar{I}E_{M/2}\bar{I} \end{bmatrix}[B_M] \tag{3.6}$$

D is an orthogonal matrix generated by a recursive computation using the H product defined by Kou et Al. [KOU85] [KOUh86]. Similar to eqn. (3.1), $T_{N/2}$ and $D_{M/2}$ represent the even parts, $C^{IV}_{N/2}$ and $E_{M/2}$ represent the odd parts of SCT_N and T_M respectively.

Definition 2.1 : The H product of two matrices A and B is defined as

$$A \circledH B = \begin{bmatrix} A & A \\ B & -B \end{bmatrix}$$

where A and B are order-N square matrix.

Theorem 2.1 : If A and B are orthogonal matrices, then the H product of the two matrices is an order-2N orthogonal matrix.

Proof of theorem 2.1 can be found in ref. [KOUh86]. According to this theorem, we can construct an orthogonal matrix D as follows :

$$[D_2] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$[D_M] = [P_{D\,M}] \begin{bmatrix} D_{M/2} & D_{M/2} \\ E_{M/2} & -E_{M/2} \end{bmatrix} \tag{3.7}$$

where $P_{D\,M}$ is a permutation and scaling matrix which converts the matrix into sequency order and normalizes the basis vectors. E is an another orthogonal matrix generated by a recursive computation using the Kronecker product.

$$[E_2] = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & -1 \end{bmatrix}$$

$$[E_M] = [P_{E\,M}][E_{M/2}] \otimes [E_2] \tag{3.8}$$

where ⊗ represents the Kronecker product and $P_{E\,M}$ is another permutation and scaling matrix which converts the Hadamard order into sequency order and normalizes the basis vectors.

case of order-8

$$T_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$

the order-8 SCT matrix is then given by

$$[SCT_8] = [P_8]\begin{bmatrix} T_4 & 0 \\ 0 & \bar{I}C_4^{IV}\bar{I} \end{bmatrix}[B_8]$$

case of order-16

$$D_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -1 & -\frac{1}{2} \\ \frac{1}{2} & -1 & -\frac{1}{2} & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \qquad E_4 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{4} & -1 & -\frac{1}{2} \\ \frac{1}{4} & -\frac{1}{2} & -\frac{1}{2} & 1 \\ \frac{1}{2} & -1 & \frac{1}{4} & -\frac{1}{2} \end{bmatrix}$$

and

$$T_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & \frac{1}{2} & -1 & -\frac{1}{2} & -\frac{1}{2} & -1 & \frac{1}{2} & 1 \\ \frac{1}{2} & \frac{1}{4} & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & -\frac{1}{4} & -\frac{1}{2} \\ \frac{1}{2} & -1 & -\frac{1}{2} & 1 & 1 & -\frac{1}{2} & -1 & \frac{1}{2} \\ \frac{1}{4} & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{4} \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ \frac{1}{2} & -1 & \frac{1}{4} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{4} & 1 & -\frac{1}{2} \end{bmatrix}$$

the order-16 SCT matrix is then given by

$$[SCT_{16}] = [P_{16}]\begin{bmatrix} T_8 & 0 \\ 0 & \bar{I}C_8^{IV}\bar{I} \end{bmatrix}[B_{16}]$$

It is noted that the even part of the SCT matrix contains only the elements of 1/2, 1/4, 1/8 ..., which can be computed by using additions and binary shifts only.

3.4 FAST COMPUTATIONAL ALGORITHMS

3.4.1 Weighted Cosine Transform

The fast computational algorithm for WCT can be obtained by slight modification of the fast recursive computational algorithm for DCT defined by Hou [HOU87]. Eqn.(3.2) shows that an order-N WCT can be decomposed into $G_{N/2}$ and $U_{N/2}$. By using Wang's algorithm [WANG86], $G_{N/2}$ can be expressed by the following equation :

$$[G_M] = [C^{II}_M][V_M] \tag{3.9}$$

where M=N/2 and

$$[V_M] = \begin{bmatrix} \cos\left(\frac{M-1}{2}\alpha\right) & & & & & \sin\left(\frac{M-1}{2}\alpha\right) \\ & \cos\left(\frac{M-3}{2}\alpha\right) & & & \sin\left(\frac{M-3}{2}\alpha\right) & \\ & & \ddots & \iddots & & \\ & & \cos\left(\frac{\alpha}{2}\right) & \sin\left(\frac{\alpha}{2}\right) & & \\ & & -\sin\left(\frac{\alpha}{2}\right) & \cos\left(\frac{\alpha}{2}\right) & & \\ & & \iddots & \ddots & & \\ & -\sin\left(\frac{M-3}{2}\alpha\right) & & & \cos\left(\frac{M-3}{2}\alpha\right) & \\ -\sin\left(\frac{M-1}{2}\alpha\right) & & & & & \cos\left(\frac{M-1}{2}\alpha\right) \end{bmatrix}$$

Eqn. (3.9) can further be decomposed into two order-N/4 transforms, i.e.

$$[G_M] = [P_M]\begin{bmatrix} C^{II}_{M/2} & 0 \\ 0 & \bar{I}C^{IV}_{M/2}\bar{I} \end{bmatrix}[\overline{V}_M] \tag{3.10}$$

where

$$[\overline{V}_M] = \begin{bmatrix}
\sin\left(\frac{\pi}{4}-\frac{M-1}{2}\alpha\right) & & & & & \sin\left(\frac{\pi}{4}+\frac{M-1}{2}\alpha\right)\\
& \sin\left(\frac{\pi}{4}-\frac{M-3}{2}\alpha\right) & & & \sin\left(\frac{\pi}{4}+\frac{M-3}{2}\alpha\right) & \\
& & \ddots & \iddots & & \\
& & \sin\left(\frac{\pi}{4}-\frac{\alpha}{2}\right) & \sin\left(\frac{\pi}{4}+\frac{\alpha}{2}\right) & & \\
& & -\sin\left(\frac{\pi}{4}+\frac{\alpha}{2}\right) & \sin\left(\frac{\pi}{4}-\frac{\alpha}{2}\right) & & \\
& & \iddots & \ddots & & \\
& -\sin\left(\frac{\pi}{4}+\frac{M-3}{2}\alpha\right) & & & \sin\left(\frac{\pi}{4}-\frac{M-3}{2}\alpha\right) & \\
-\sin\left(\frac{\pi}{4}+\frac{M-1}{2}\alpha\right) & & & & & \sin\left(\frac{\pi}{4}-\frac{M-1}{2}\alpha\right)
\end{bmatrix}$$

Similarly, the matrix $[U_M]$ can be expressed in terms of the DCT-IV matrix as follows :

$$[U_M] = [C_M^{IV}][R_M] \tag{3.11}$$

where M=N/2 and

$$[R_M] = \begin{bmatrix} Q_{M/2} & W_{M/2} \\ -W_{M/2} & Q_{M/2} \end{bmatrix}$$

$$[Q_M] = \begin{bmatrix}
\cos\left(\frac{M-1}{2}\beta\right) & & & & & \frac{1}{\sqrt{2}}\sin\left(\frac{M-1}{2}\beta\right)\\
& \cos\left(\frac{M-3}{2}\beta\right) & & & \frac{1}{\sqrt{2}}\sin\left(\frac{M-3}{2}\beta\right) & \\
& & \ddots & \iddots & & \\
& & \cos\left(\frac{\beta}{2}\right) & \frac{1}{\sqrt{2}}\sin\left(\frac{\beta}{2}\right) & & \\
& & \frac{-1}{\sqrt{2}}\sin\left(\frac{\beta}{2}\right) & \cos\left(\frac{\beta}{2}\right) & & \\
& & \iddots & \ddots & & \\
& \frac{-1}{\sqrt{2}}\sin\left(\frac{M-3}{2}\beta\right) & & & \cos\left(\frac{M-3}{2}\beta\right) & \\
\frac{-1}{\sqrt{2}}\sin\left(\frac{M-1}{2}\beta\right) & & & & & \cos\left(\frac{M-1}{2}\beta\right)
\end{bmatrix}$$

$$[W_M] = \frac{1}{\sqrt{2}} \begin{bmatrix} \sin\left(\frac{M-1}{2}\beta\right) & & & & & & & \\ & \sin\left(\frac{M-3}{2}\beta\right) & & & & & & \\ & & \ddots & & & & & \\ & & & \sin\left(\frac{\beta}{2}\right) & & & & \\ & & & & -\sin\left(\frac{\beta}{2}\right) & & & \\ & & & & & \ddots & & \\ & & & & & & -\sin\left(\frac{M-3}{2}\beta\right) & \\ & & & & & & & -\sin\left(\frac{M-1}{2}\beta\right) \end{bmatrix}$$

The DCT-IV matrix in eqn.(3.11) can be expressed in terms of the DCT-II matrix by using the Hou's algorithm as follows :

$$\begin{aligned} [U_M] &= [C_M^{IV}][R_M] \\ &= [L_M][C_M^{II}]\,\mathrm{diag}[2\cos(\phi_m)][R_M] \\ &= [L_M][C_M^{II}][\hat{R}_M] \end{aligned}$$

where m = 0, 1, 2, ..., M-1

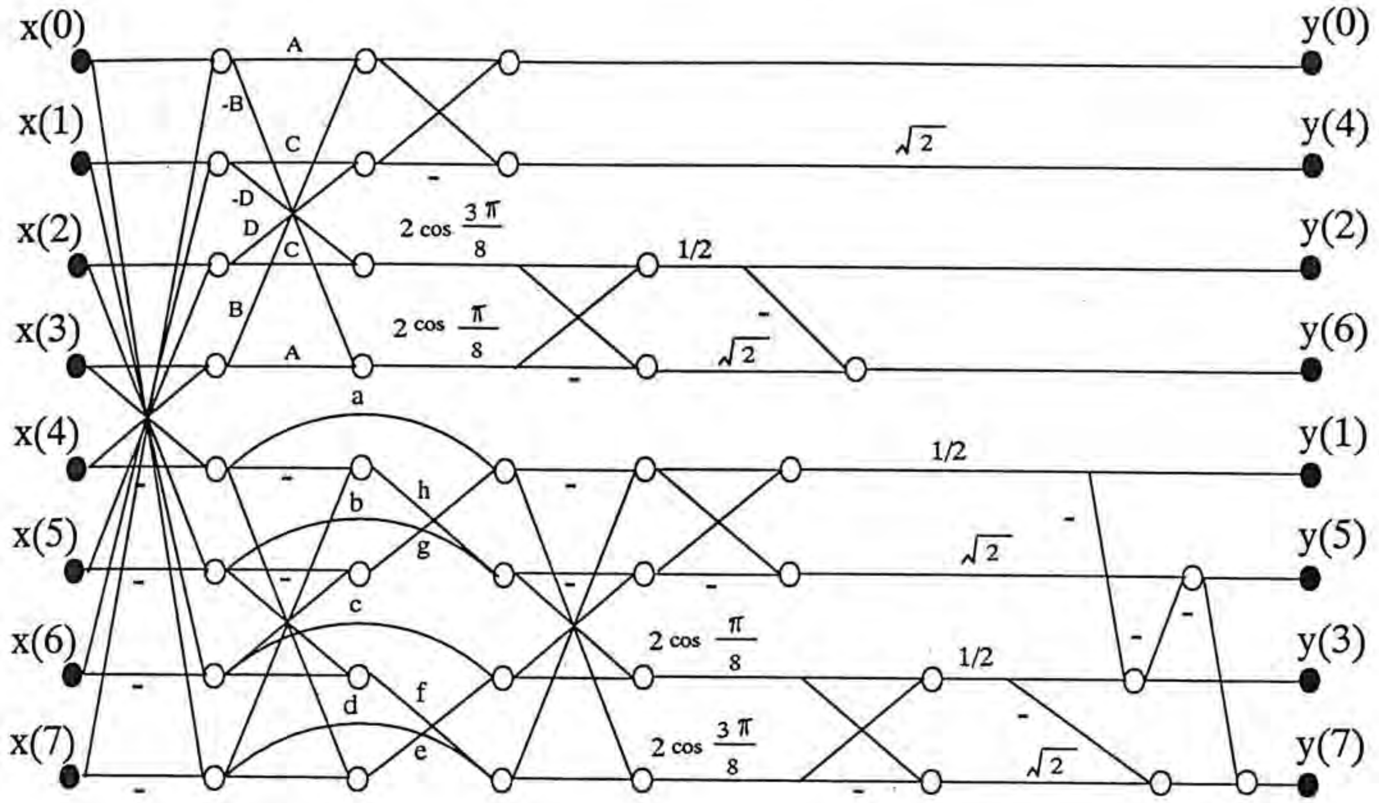
$$\phi_m = \frac{(2m+1)}{4M}$$

$$[L_M] = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & . & . & . & 0 \\ -\frac{1}{2} & 1 & 0 & 0 & . & . & . & 0 \\ \frac{1}{2} & -1 & 1 & 0 & . & . & . & 0 \\ -\frac{1}{2} & 1 & -1 & 1 & . & . & . & 0 \\ . & & & & & & & . \\ . & & & & & & & . \\ . & & & & & & & . \\ -\frac{1}{2} & 1 & -1 & 1 & . & . & . & 1 \end{bmatrix}$$

$$[\hat{R}_M] = \frac{1}{\sqrt{2}} \begin{bmatrix} 2\cos\phi_0 \hat{R}_0 \\ 2\cos\phi_1 \hat{R}_1 \\ . \\ . \\ . \\ 2\cos\phi_{M-1} \hat{R}_{M-1} \end{bmatrix}$$

$\hat{R}_i$ represents the ith row vector of the matrix $[\hat{R}_M]$.

Therefore an order-N WCT can be decomposed into two order-N/2 DCT-II which can be computed using a fast computational algorithm. Figure 3.1 shows the signal flow graph of the fast WCT algorithm for order-8. It is noted that computation of WCT is similar to DCT with a little modification to the existing fast DCT algorithms.



$$a = 2\cos\frac{7\pi}{16}\cos\frac{\beta}{2} \quad b = 2\cos\frac{5\pi}{16}\cos\frac{\beta}{2} \quad c = 2\cos\frac{3\pi}{16}\cos\frac{\beta}{2} \quad d = 2\cos\frac{\pi}{16}\cos\frac{\beta}{2}$$

$$e = -\sqrt{2}\cos\frac{3\pi}{16}\sin\frac{\beta}{2} \quad f = \sqrt{2}\cos\frac{\pi}{16}\sin\frac{\beta}{2} \quad g = \sqrt{2}\cos\frac{7\pi}{16}\sin\frac{\beta}{2} \quad h = -\sqrt{2}\cos\frac{5\pi}{16}\sin\frac{\beta}{2}$$

$$A = \sin\left(\frac{\pi}{4} - \frac{3}{2}\beta\right) \quad B = \sin\left(\frac{\pi}{4} + \frac{3}{2}\beta\right) \quad C = \sin\left(\frac{\pi}{4} - \frac{1}{2}\beta\right) \quad D = \sin\left(\frac{\pi}{4} - \frac{1}{2}\beta\right)$$

Figure 3.1 Signal flow diagram for the order-8 fast WCT algorithm

3.4.2 Simplified Cosine Transform

According to the definition of SCT given in eqn.(3.5), the odd part is the same as the DCT-II and which can be computed via HOU's fast DCT algorithm [HOU87] as described before. The even part of SCT can be computed recursively according to eqns. (3.7) and (3.8). Since the basis elements of the even part of SCT are the values of ±1, ±1/2, ±1/4, ±1/8 ... $\pm 1/2^{m-2}$ for an order-2^m transform, implementation requires no multiplications but additions and binary shifts. The signal flow diagram for an order-8 SCT is shown in Figure 3.2.

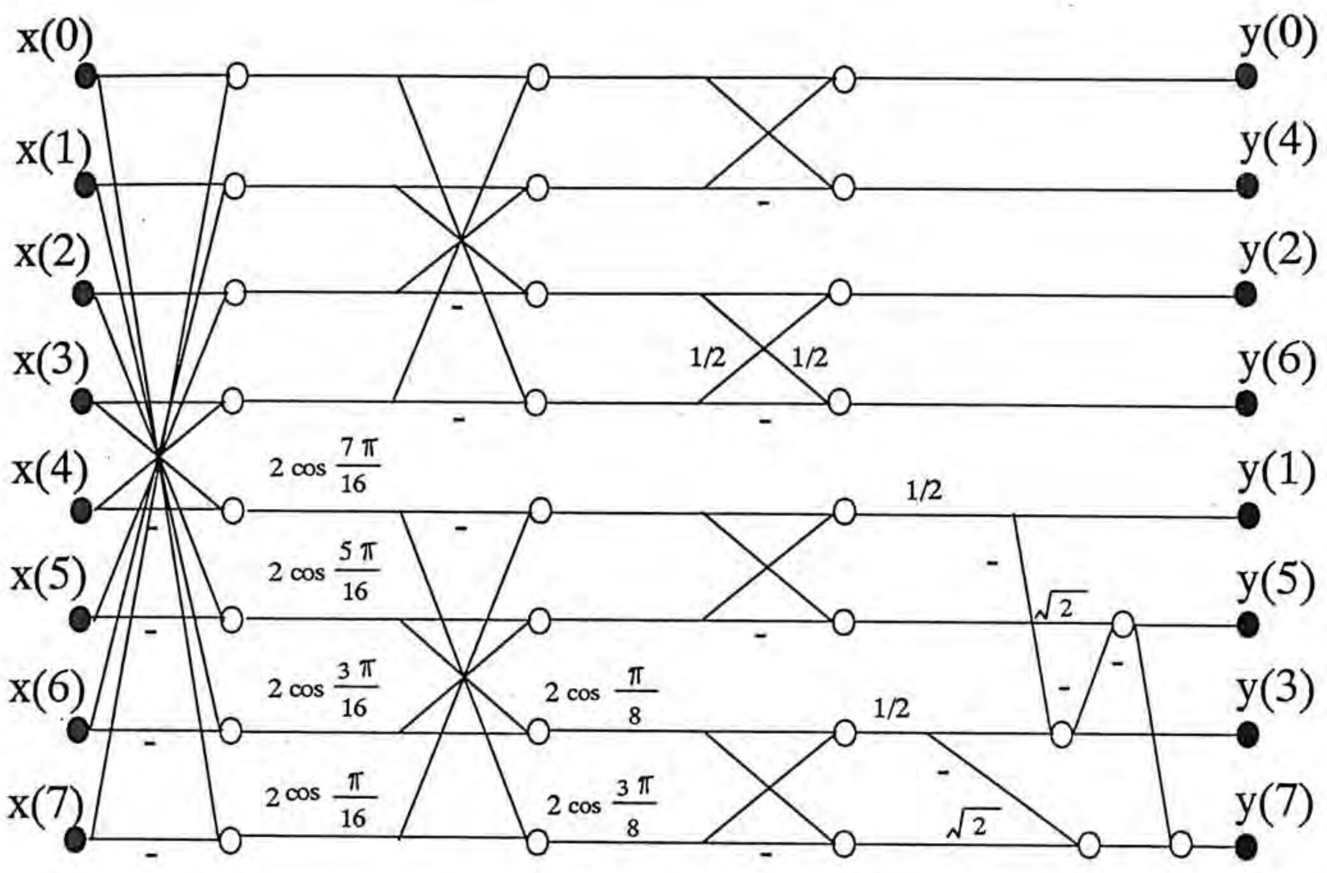


Figure 3.2 Signal flow diagram for the order-8 fast SCT algorithm

3.4.3 Computational Requirement

The computational requirement for computing the order-N DCT, WCT and SCT is shown in Table 3.3, where N is an integer power of 2. The actual number of operations required by those fast algorithms is also listed in Table 3.4.

	Number of Multiplications	Number of Additions
DCT	$\frac{1}{2}N\log_2 N$	$\frac{3}{2}N\log_2 N - N + 1$
WCT	$\frac{1}{2}N\log_2 N + \frac{3}{2}N$	$\frac{3}{2}N\log_2 N + 1$
SCT	$\frac{N}{4}(\log_2 N + 1)$	$\frac{5}{4}N\log_2 N - \frac{N}{4}$

Table 3.3 Computational requirements of fast DCT, WCT and SCT

N	DCT	WCT	SCT
4	4	10	3
8	12	24	8
16	32	56	20
32	80	128	48
64	192	288	112
128	448	640	256

a. Number of multiplications

N	DCT-II	WCT	SCT
4	9	13	9
8	29	37	28
16	81	97	76
32	209	241	192
64	513	577	464
128	1217	1345	1088

b. Number of additions

Table 3.4 Number of operations required by fast DCT, WCT and SCT

Table 3.3 and 3.4 reveals that implementation of an order-N WCT needs 1.5N multiplications and N additions more than the DCT. On the other hand, implementation of SCT requires $0.25N(\log_2 N-1)$ multiplications and $0.25N\log_2 N-0.75N+1$ additions less than the DCT, however, extra $0.25N(\log_2 N-1)$ binary shifts are required.

3.5 PERFORMANCE EVALUATION

In our experiments, both statistical model and real images have been utilized to evaluate the performance of WCT and SCT in image coding applications. We compare the performance of the two new transforms with the DCT, PSCT, MDCT, ICT, HCT and LCT to see how well the transforms are.

3.5.1 Evaluation Using Statistical Model

For an input vector x sampled from an one-dimensional, zero mean, unit-variance first-order Markov process with adjacent element correlation coefficient ρ, the covariance matrix of x is given by (2.1) in the last chapter. The vector x is transformed into y by an orthogonal transform T, i.e. y=[T]x. The covariance matrix of the vector y in the transform domain [$COV_y(\rho)$] is also given by (2.19). The performance of the new transforms is compared with DCT, PSCT, MDCT, ICT, HCT and LCT by using the criteria of transform coding gain [WANGg88], maximum reducible bits [WANG86], residue correlation [HAMIp76] and transform efficiency [CHAMc86]. In our experiments, ICT(10,9,6,2) [CHAM89] and ICT(81,80,74,67,53,40,23,13) [KOHht91] are chosen for the orders 8 and 16 ICT respectively.

3.5.1.1 Transform Coding Gain

Transform coding gain measures the energy compaction ability of the transform in transform coding system and it is given as follows

$$G_{TC} = \frac{\sum_{i=1}^{N} s_{ii}}{\left(\prod_{i=1}^{N} s_{ii}\right)^{1/N}}$$

The results of transform coding gain for WCT, SCT, DCT, PSCT, MDCT, ICT, HCT and LCT for orders 8 and 16 are listed in Table 3.5. It is seen that WCT and PSCT have larger transform coding gain than others. They performs very close but WCT is slightly better than PSCT. It implies that WCT has the best compression ability. On the other hand, SCT performs better than HCT and LCT and has similar values of transform coding gain as those of the DCT , ICT and MDCT. It is also noted that the order-8 SCT performs closer to the DCT than that of the order-16 SCT. For example, when the adjacent element coefficient is equal to 0.95, the transform coding gain for orders 8 and 16 SCT are 7.6128 and 8.1185, and those values of DCT are 7.6311 and 8.8216 respectively. This shows that the order-8 SCT results 99.76% of the DCT while the order-16 SCT achieves only 92.03% of the DCT. On the other hand, Table 3.5 also reveals that the order-8 SCT has larger transform coding gain than ICT when ρ=0.95.

ρ		0.80	0.85	0.90	0.95
N=8	WCT	2.4253	3.0518	4.2624	7.6601
	SCT	2.4121	3.0326	4.2330	7.6128
	DCT	2.4162	3.0386	4.2423	7.6311
	PSCT	2.4238	3.0502	4.2607	7.6588
	MDCT08	2.4192	3.0416	4.2445	7.6289
	ICT	2.4134	3.0341	4.2340	7.6105
	HCT	2.2825	2.8438	3.9292	6.9858
	LCT	2.2438	2.7683	3.7801	6.6298
N=16	WCT	2.5843	3.3030	4.7222	8.8518
	SCT	2.4080	3.0577	4.3465	8.1185
	DCT	2.5793	3.2942	4.7058	8.8216
	PSCT	2.5827	3.3010	4.7198	8.8486
	ICT	2.5139	3.2001	4.5539	8.4976
	HCT	2.2550	2.8186	3.9308	7.1756
	LCT	2.1970	2.6976	3.6761	6.5211

Table 3.5 Transform coding gain for orders 8 and 16 transforms

3.5.1.2 Maximum Reducible Bits

The maximum reducible bits (MRB) of an unitary transform T is defined as

$$\mathrm{MRB} = -\frac{1}{2N} \sum_{i=0}^{N-1} \log_2 s_{ii}$$

This criterion measures the geometric mean of the transform coefficient variances in log scale, which is equivalent to the maximum bits which can be reduced from each transform component. The greater is the MRB, the higher compression ability is the transform. However, value of MRB is not very sensitive to the change of transforms. Table 3.6 shows the MRB of WCT, SCT, DCT, PSCT, MDCT, ICT, HCT and LCT for N=8 and N=16.

ρ		0.80	0.85	0.90	0.95
N=8	WCT	0.6391	0.8048	1.0458	1.4687
	SCT	0.6352	0.8003	1.0408	1.4642
	DCT	0.6364	0.8017	1.0424	1.4660
	PSCT	0.6386	0.8045	1.0456	1.4685
	MDCT08	0.6371	0.8025	1.0429	1.4658
	ICT	0.6356	0.8006	1.0410	1.4640
	HCT	0.5849	0.7539	0.9871	1.4022
	LCT	0.5678	0.7345	0.9592	1.3645
N=16	WCT	0.6849	0.8619	1.1197	1.5730
	SCT	0.6339	0.8063	1.0599	1.5106
	DCT	0.6835	0.8600	1.1172	1.5705
	PSCT	0.6845	0.8615	1.1194	1.5727
	ICT	0.6619	0.8366	1.0918	1.5426
	HCT	0.5849	0.7461	0.9865	1.4211
	LCT	0.5678	0.7158	0.9309	1.3526

Table 3.6 Maximum Reducible Bits (MRB) for orders 8 and 16 transforms

Table 3.6 shows that values of MRB of WCT, SCT, DCT, MDCT and ICT are close and larger than those of HCT and LCT. MRB of WCT is the largest for all values of adjacent element correlation coefficients. It implies that WCT has the smallest values of geometric mean of transform coefficient variances and hence has the minimum reconstruction error at the output as stated in the section 2.3.1. On the other hand, SCT performs better than HCT and LCT and have similar values as those of the DCT, MDCT and ICT. It is seen that when N=8 and ρ=0.95, SCT is even better than the ICT. From Table 3.6, it is also shown that the order-8 SCT performs closer to the DCT than that of the order-16 SCT. For example, when the adjacent element coefficient is equal to 0.95, the values of MRB for orders 8 and 16 SCT are 1.4641 and 1.5106, and those of the DCT are 1.4660 and 1.5705 respectively. It is seen that the order-8 SCT results 99.87% of the DCT while the order-16 SCT achieves only 96.19% of the DCT.

3.5.1.3 Residue Correlation

The residue correlation (RC) of a sub-optimal transform T is defined by the following equation.

$$RC = \frac{\left[N \,|\, COV_x(\rho)|^2 - \sum_{i=0}^{N-1} |s_{ii}|^2 \right]}{N \,|\, COV_x(\rho)|^2 - N}$$

This criterion measures the proportional correlation left in by a sub-optimal transform T, the smaller is the residue correlation, the higher is decorrelation ability of the transform. Table 3.7 shows the RC for different transforms for orders 8 and 16.

Table 3.7 shows that WCT and PSCT perform outstandingly among other transforms since they result the minimum values of RC. WCT is slightly better than PSCT. It implies that WCT has the highest decorrelation ability. On the other hand, SCT performs better than HCT and LCT but has a little larger values than those of the DCT, MDCT and ICT. It is also noted that the order-8 SCT results closer to the DCT than that of the order-16 SCT. For example, when the adjacent element coefficient is equal to 0.95, the values of RC for orders 8 and 16 SCT are 0.1773 and 0.6290, and those of the DCT are 0.1768 and 0.5983. It is shown that the order-8 SCT results only 0.28% larger RC than that of the DCT while the order-16 SCT obtains 5.13% larger RC than that of the DCT. When ρ=0.95, the order-8 SCT has the same value of RC as ICT.

ρ		0.80	0.85	0.90	0.95
N=8	WCT	1.2559	0.6348	0.1910	0.0080
	SCT	2.1725	1.3538	0.6581	0.1772
	DCT	2.1600	1.3475	0.6557	0.1768
	PSCT	1.2984	0.6594	0.2020	0.0087
	MDCT08	2.1479	1.3242	0.6548	0.1770
	ICT	2.1276	1.3348	0.6534	0.1772
	HCT	2.9605	1.8259	0.8796	0.2350
	LCT	4.4192	2.8332	1.4063	0.3841
N=16	WCT	2.4738	1.5397	0.6011	0.0394
	SCT	4.9468	3.5211	1.9965	0.6290
	DCT	3.9883	3.0468	1.8256	0.5983
	PSCT	2.6150	1.6226	0.6329	0.0436
	ICT	3.8644	2.9872	1.8090	0.5985
	HCT	6.9251	4.9271	2.7796	0.8673
	LCT	12.590	9.1689	5.2038	1.6078

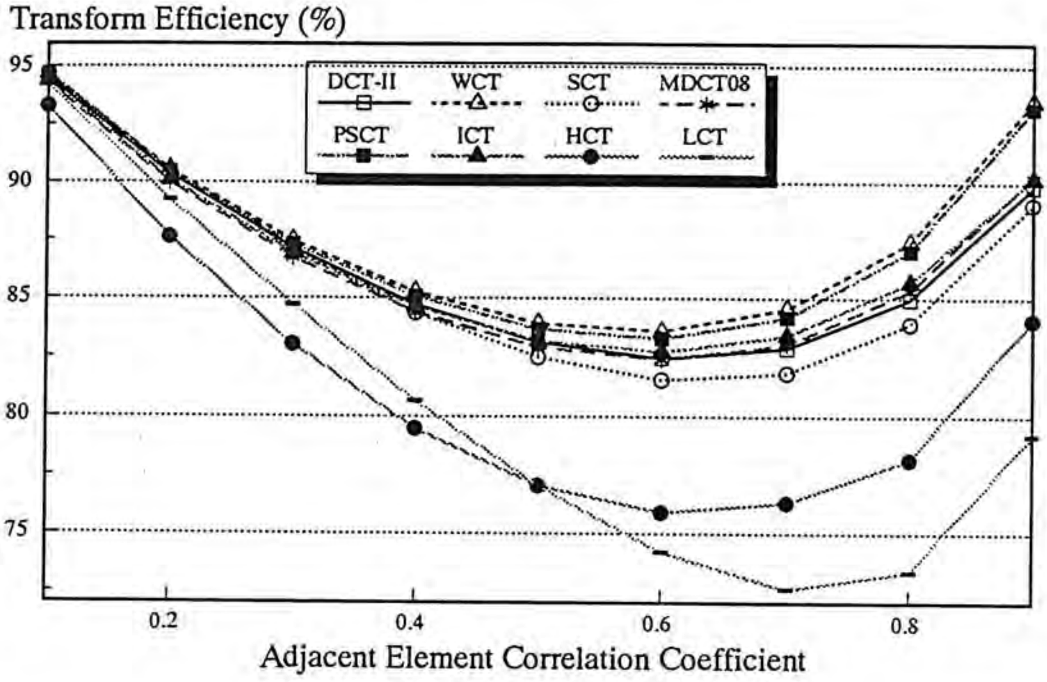
Table 3.7 Residue Correlation (RC) for orders 8 and 16 transforms

3.5.1.4 Transform Efficiency

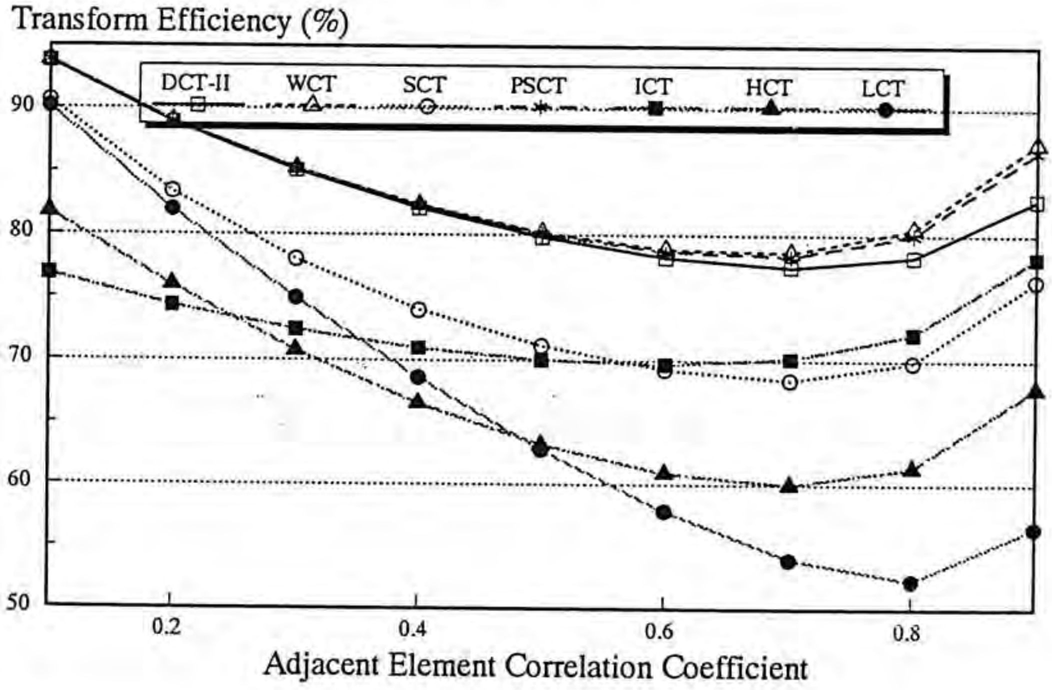
The transform efficiency measures the ability of an unitary transform T to decorrelate the input vector x and it is given as

$$\text{Transform Efficiency (\%)} = \frac{\sum_{i=0}^{N-1} |s_{ii}|}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |s_{ij}|} \times 100$$

The larger is the transform efficiency, the higher is decorrelation ability of the transform. The transform efficiency versus different correlation coefficients ρ for WCT, SCT, DCT, PSCT, MDCT, ICT, HCT and LCT with N=8 and N=16 is shown in Figure 3.3.



a. Order-8



b. Order-16

Figure 3.3 Transform Efficiency (%) for the DCT, WCT and SCT

As shown in Figure 3.3, the transform efficiency of WCT and PSCT is higher than other transforms. Transform efficiency of WCT is slightly higher than that of PSCT and it implies that WCT has the highest decorrelation ability among all the transforms being tested. On the other hand, transform efficiency of SCT is larger than those of HCT and LCT but slightly smaller than those of the DCT, MDCT and ICT. It is also noted that the order-8 SCT obtain the values of transform efficiency closer to DCT than that of the order-16 SCT.

The above tests show that WCT and PSCT have better performance than SCT, DCT, MDCT, ICT, HCT and LCT under the four well known criteria. WCT and PSCT perform very close but WCT is slightly better. Implementation of an order-N WCT and PSCT is also similar with WCT requiring 0.5N more multiplications and N more additions. On the other hand, it is also shown that SCT performs better than HCT and LCT but a little worse than the DCT and ICT. When the adjacent element correlation coefficient is equal to 0.95, the order-8 SCT performs even better than ICT. When the transforms are implemented using a general purpose signal processor or microprocessor via fast computational algorithms, an order-N SCT needs $0.25N(\log_2 N-1)$ multiplications and $0.25N\log_2 N-0.75N+1$ additions less but $0.25N(\log_2 N-1)$ binary shifts more than those of the DCT and ICT. The increase in the number of binary shift is moderate since complexity is dominated by multiplication, the over-all saving in the number of operations for SCT is still considerable. Therefore, SCT will require shorter computation time than those of the DCT and ICT. However, if the transforms are implemented using a dedicated hardware or chip, ICT will lead to a simpler IC structure and shorter computation time than the DCT and SCT becasue ICT involves only integer arithmetic and requires a smaller number of bit representation (4 bits for order-8 and 7-bits for order-16 in our experiments). Since SCT contains integer part in its even part of the transform matrix, SCT will also has simpler structure and faster computation time than the DCT.

3.5.2 Evaluation Using Real Images

Computer simulations using real images have been carried out to evaluate the performance of WCT and SCT. The three test images Lenna, Peppers and Sailboat are of size 256 x 256 with 8-bit resolution per pixel. In our experiments, an image is first divided into subsequent blocks of data with size NxN. The block of data undergoes a two-dimensional transformation and several high frequency coefficients are discarded. The retained coefficients are then inverse transformed to obtain the reconstructed image. Mean square error (MSE) as given in eqn. (2.5) is used to evaluate the performance of the transforms. Tables 3.8 and 3.9 show the MSE of WCT, SCT, DCT, PSCT, MDCT, ICT, HCT and LCT for different number of coefficients retained (N_r), test images and orders.

N_r	WCT	SCT	DCT	PSCT	MDCT08	ICT	HCT	LCT
2	476.15	458.89	458.89	476.14	458.98	459.05	461.57	472.72
6	158.41	159.71	159.42	158.43	159.70	159.83	172.19	223.10
10	109.97	110.89	110.05	110.17	110.66	110.60	127.30	137.23
18	72.60	73.04	72.68	72.64	73.16	73.16	93.23	89.77
26	42.53	42.77	42.56	42.63	42.98	42.91	60.20	47.99
34	31.16	31.21	31.18	31.21	31.58	31.56	43.23	34.12

a. Lenna

N_r	WCT	SCT	DCT	PSCT	MDCT08	ICT	HCT	LCT
2	541.89	528.69	528.69	541.54	529.12	529.47	535.93	565.90
6	183.34	186.12	184.67	183.86	184.69	185.51	207.10	285.49
10	115.62	118.31	116.74	115.60	116.23	116.38	154.99	165.31
18	62.50	64.25	62.64	62.55	62.23	62.27	98.03	92.77
26	36.88	37.75	37.00	36.97	36.91	36.83	61.98	47.12
34	23.61	23.79	23.66	23.66	23.93	23.88	39.19	28.56

b. Peppers

N_r	WCT	SCT	DCT	PSCT	MDCT08	ICT	HCT	LCT
2	714.47	696.36	696.36	714.78	696.26	696.39	706.40	736.30
6	345.84	349.39	347.97	345.97	347.63	347.64	370.41	433.35
10	238.31	241.15	239.60	238.56	239.21	239.04	283.45	284.67
18	139.76	141.54	139.95	139.95	140.04	139.96	179.83	168.03
26	88.00	89.18	88.10	88.19	88.41	88.29	118.57	99.69
34	53.94	54.18	53.97	53.97	54.42	54.42	70.02	60.68

c. Sailboat

Table 3.8 MSE performance for order-8 transforms

(N_r is the number of coefficient retained)

N_r	WCT	SCT	DCT	PSCT	ICT	HCT	LCT
2	790.20	735.69	735.69	790.14	735.76	740.24	758.84
6	336.87	344.34	339.80	336.94	340.16	365.13	461.58
10	255.80	263.00	256.58	255.98	257.09	293.35	368.44
32	130.12	150.14	130.29	130.34	130.73	168.22	218.62
64	75.48	97.08	75.63	75.60	76.24	105.51	124.74
128	40.25	49.93	40.38	40.44	46.64	55.21	64.25

a. Lenna

N_r	WCT	SCT	DCT	PSCT	ICT	HCT	LCT
2	955.10	924.19	924.19	954.87	924.68	935.33	978.96
6	397.58	406.78	404.21	397.60	405.34	445.76	602.97
10	293.01	302.77	297.63	293.70	298.81	363.96	476.46
32	130.56	155.35	131.94	131.06	132.83	203.43	261.79
64	70.63	97.87	70.78	70.09	71.54	126.03	143.44
128	32.66	41.07	32.91	32.90	41.13	54.09	59.91

b. Peppers

N_r	WCT	SCT	DCT	PSCT	ICT	HCT	LCT
2	1397.99	1372.24	1372.24	1397.74	1372.62	1385.40	1435.27
6	675.02	686.03	679.24	676.62	678.37	714.62	853.01
10	518.37	530.01	519.74	518.45	520.57	5882.8	702.69
32	267.24	306.58	268.11	267.88	268.67	350.03	408.51
64	158.19	203.45	158.25	158.26	159.28	225.68	246.66
128	70.55	87.41	70.83	70.87	90.95	101.62	111.07

c. Sailboat

Table 3.9 MSE performance for order-16 transforms

WCT has similar MSE performance as the DCT when the number of coefficients retained is about 3 and slightly better performance than the DCT when the number of coefficients retained is more than 5 for all three images. When the number of coefficients retained is small, such as the case of retaining only 2 coefficients, the DCT results smaller MSEs than WCT. For images coded using normal transform coding system, very few blocks contain less than 2 coefficients and so the use of WCT can result images of better quality. In progressive transmission, there is a moment when the decoded image contains very few coefficients in each block. However, such 'low-passed' image aims to provide a brief look of the image only and so should last for a very short moment. Therefore, WCT is still a better choice than the DCT in progressive transmission of images. On the other hand, the order-8 SCT has very close MSE performance as the DCT but not the order-16 SCT. Take the image Lenna as an example, when the number of coefficients retained is 10, orders 8 and 16 SCT have MSE 110.89 and 263, and those of the DCT are 110.05 and 256.58 respectively. Hence, we may conclude that the order-8 SCT can replace the order-8 DCT in image coding systems with no significant degradation in MSE performance.

3.6 CONCLUDING REMARKS

In this chapter, two new orthogonal transforms called the weighted cosine transform (WCT) and the simplified cosine transform (SCT) are proposed for image coding applications. The WCT is obtained by weighting the transform matrix of DCT with simple structured orthonormal matrices. WCT has a parameter (α, ß) which is fixed respectively to be (0.0237, 0.0231) and (0.021, 0.0184) for N equal to 8 and 16. The performance of WCT is better than DCT, PSCT and MDCT under several well known criteria for orders 8 and 16 based on a first-order Markov model. Simulations using real images also show that WCT has better energy packing ability. Implementation of an order-N WCT requires 1.5N multiplications and N additions more than the DCT. WCT can achieve even better performance if the parameter (α, β) adapts to the adjacent element correlation of input image data, however, such adaptation requires too much computation.

On the other hand, the SCT is obtained by replacing the even part of the DCT matrix with a simple structure orthogonal matrix which can be computed using only additions and binary shifts. Different tests show that the order-8 SCT performs very close to the DCT and ICT but the order-16 performs slightly worse than other transforms. When the adjacent element correlation coefficient is equal to 0.95, the order-8 SCT even performs better than ICT. Implementation of an order-N SCT needs $0.25N(\log_2 N-1)$ multiplications and $0.25N\log_2 N-0.75N+1$ additions less, and also $0.25N(\log_2 N-1)$ binary shifts more than those of the DCT and ICT. The increase in the number of binary shift is moderate, the overall saving in the number of operations for SCT is still considerable. Therefore, if transforms are implemented using a general purpose processor, SCT will need shorter computation time than those of the DCT and ICT. However, if the transforms are implemented using dedicated chip, ICT will lead to simpler IC structure and shorter computation time because it involves only integer arithmetic. To conclude, we provide an alternative for people to select an orthogonal transform in image transform coding systems. When performance is the most important criterion, we may use WCT as substitute for the DCT. On the other hand, when the cost of computation is the main consideration factor, we may use the order-8 SCT instead of the DCT and ICT.

The efficiency of the transformation process in a block-based image coding system can further be improved by incorporating pruning in such a process. In transform coding of images, most of the energy is packed into a few low frequency coefficients, therefore, it is not necessary to compute all coefficients. In the next chapter, we will consider pruning in transform coding system. Pruned fast computational algorithms for the DCT, WCT and SCT will be derived.

3.7 NOTE ON PUBLICATIONS

A paper entilted 'Image coding using weighted cosine transform' was presented at the 1990 IEEE region 10 conference on Computer and Communcation Systems (TENCON'90), Hong Kong, Sep. 1990. Another paper entitles 'A modified cosine transform' has been accepted for publication in Journal of Visual Communcation and Image Representation. These two papers were coauthored with Dr. W.K. Cham.

CHAPTER 4 PRUNING IN TRANSFORM CODING OF IMAGES

In this chapter, we consider pruning in the transformation process of a transform coding system. Pruning is a method to modify the signal flow in the computational algorithm of the transform such that certain operations having no contributions to the required outputs can be eleminated. Pruned fast algorithms for computing the DCT, WCT and SCT are derived.

4.1 INTRODUCTION

In the last chapter, we compared the performance and computational requirement of two new orthogonal transforms with some popular transforms. Comparison of computational requirement is based on the best fast algorithm of each transform. In fact, the implementation of even the simplest transforms using direct matrix multiplication still requires a lot of computations. That is why various fast algorithms have been proposed for reducing the computational complexity of the transformation process for different transforms. Since the DCT is widely accepted as the industrial standard in image coding applications, many fast algorithms have been developed. Some fast DCT algorithms [VETTn84] were derived via the fast Fourier transform algorithms. More efficient algorithms are derived using matrix factorisation [CHENsf77] [WANG84] [MORIhy85] [WANG91] or recursive computation [LEE84] [HOU87] [CHANh90] based on the intrinsic properties of the DCT matrix.

Among the existing algorithms, the Lee's method [LEE84], the Hou's method [HOU87], the direct method proposed by Chan & Ho [CHANh90] and the simple structured FCT (SFCT) algorithm proposed by Morikawa et al. [MORIhy85] and Wang [WANG91] all achieve the minimum computational complexity for the radix-2 DCT. To compute an order-N DCT, all four algorithms require $\frac{1}{2}$MN multiplications and 2MN-N+1 additions, where M=$\log_2$N. In addition to computational requirement, ease of implementation and fixed-point error properties are other two important criteria to access the performance of a fast DCT algorithm. The above-mentioned algorithms all have regular and simple structure for implementation. However, the Lee's algorithm which contains secant multipliers has larger round-off error than other three algorithms which all use cosines as multipliers. In Hou's method, an index mapping is required to transform the DCT to a phase-modulated DFT which may not be performed in-place. The SFCT and direct fast DCT algorithm are both in-place and numerically stable. Therefore, the direct fast DCT algorithm and the SFCT algorithm can be regarded as the two best algorithms for computing the radix-2 DCT. A major difference between the two algorithms is that the direct fast DCT algorithm requires index mapping from bit-reversal order to normal order whilst the SFCT algorithm requires index mapping from normal order to Hadamard order.

In transform coding of images, most of the energy of the spatial data is packed into a few low frequency transform coefficients, therefore, there is no need to compute all the transform coefficients since most of the high frequency coefficients can be discarded. This also implies that in the inverse transformation stage most of the input high frequency transform coefficients are zero. Hence, in both forward and inverse transformations a significant number of computations can be saved by incorporating pruning [WANG91] [NARAp91] in the fast DCT algorithms. Wang pruned the SFCT algorithm in the case that only the first P transform coefficients need to be computed for an order-N transform and has shown that the percentage of operation saving (POS) obtained by pruning is approximately equal to (1 - log P / Log N). In this chapter, a pruning algorithm for the Chan & Ho's direct fast DCT algorithm is derived. We shall show that the pruning algorithm requires less number of additions than the Wang's algorithm. Pruning is first derived when P is an integer power of 2 for the purpose of simplicity and finally extended to the general case for which P is any integer smaller than N. Owning to the orthogonal property of the DCT, the inverse of the DCT matrix is simply its transpose which can be obtained by reversing the direction of the signal flow in the flow diagram of the forward transformation. Therefore, the pruning algorithm for the forward transform can be easily changed into a pruning algorithm for the inverse transform with only the first P input coefficients being non-zero. Since the WCT and the SCT proposed in the last chapter can also be computed via the fast DCT algorithms, the direct fast algorithms and the corresponding pruning algorithms for computing the WCT and SCT are also derived in this chapter.

In section 4.2, the direct fast algorithms for computing the DCT, WCT and SCT will first be described. Pruning of these algorithms will be discussed in section 4.3 and the number of operations saved by using pruning will then be given in section 4.4, Finally, concluding remarks will be given in section 4.5.

4.2 DIRECT FAST ALGORITHMS FOR DCT, WCT AND SCT

4.2.1 Discrete Cosine Transform

The direct fast DCT algorithm proposed by Chan & Ho [CHANh90] is a variant of Hou's algorithm [HOU87]. The order-N DCT of a sequence {x(n):n = 0, 1, ..., N-1} is defined as

$$y(i) = \sqrt{\frac{2}{N}}\, k_i \sum_{j=0}^{N-1} x(j) \cos\left(i\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right) \qquad (4.1)$$

$$\text{for } i = 0, 1, \ldots, N-1$$

where

$$k_i = \begin{cases} \dfrac{1}{\sqrt{2}} & \text{for } i=0 \\ 1 & \text{otherwise} \end{cases}$$

and the inverse transform is given by

$$x(j) = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} y(i) k_i \cos\left(i\left(j+\frac{1}{2}\right)\frac{\pi}{N}\right) \qquad (4.2)$$

$$\text{for } n = 0, 1, \ldots, N-1$$

Considering the symmetry properties of the DCT matrix and ignoring the scaling coefficients, an order-N transform can be decomposed into two order-N/2 transforms as follows :

$$y(2i) = \sum_{j=0}^{N/2-1} [x(j) + x(N-j-1)] \cos \frac{i\,(2j+1)\pi}{2(N/2)} \tag{4.3.a}$$

$$y(2i+1) = \sum_{j=0}^{N/2-1} [x(j) - x(N-j-1)] \cos \frac{(2i+1)\,(2j+1)\pi}{2N} \tag{4.3.b}$$

By utilizing the trigonometric identity :

$$\cos(\alpha+\beta) = 2\cos\alpha\,\cos\beta - \cos(\alpha-\beta)$$

eqn.(4.3.b) can be written as

$$y(2i+1) = \sum_{j=0}^{(N/2)-1} [x(j) - x(N-j-1)]\, 2\cos\phi_j \cos \frac{i\,(2j+1)\pi}{2(N/2)} - y(2i-1) \tag{4.4}$$

where

$$\phi_j = \frac{(2j+1)\pi}{(2N)}$$

By using eqns. (4.3) and (4.4), an order-N DCT can then be calculated by recursively decomposing it to the lowest order, says order-2. The matrix representation of this factorisation is given by

$$C_N = P_N R_N \begin{bmatrix} C_{N/2} & 0 \\ 0 & C_{N/2} D_{N/2} \end{bmatrix} B_N \tag{4.5}$$

where P_N is a permutation matrix and B_N is to perform the input butterfly-like additions; R_N performs the recursive additions and $D_{N/2}$ is a diagonal matrix which performs the pre-multiplications given in eqn.(4.4). It is seen that this algorithm can be divided into two stages : the first one is the butterfly stage and the second is the permutation and recursive addition stage. The basic structure of the direct fast DCT algorithm is shown in Figure 4.1 and the flow diagram of the order-16 DCT algorithm is shown in Figure 4.2.

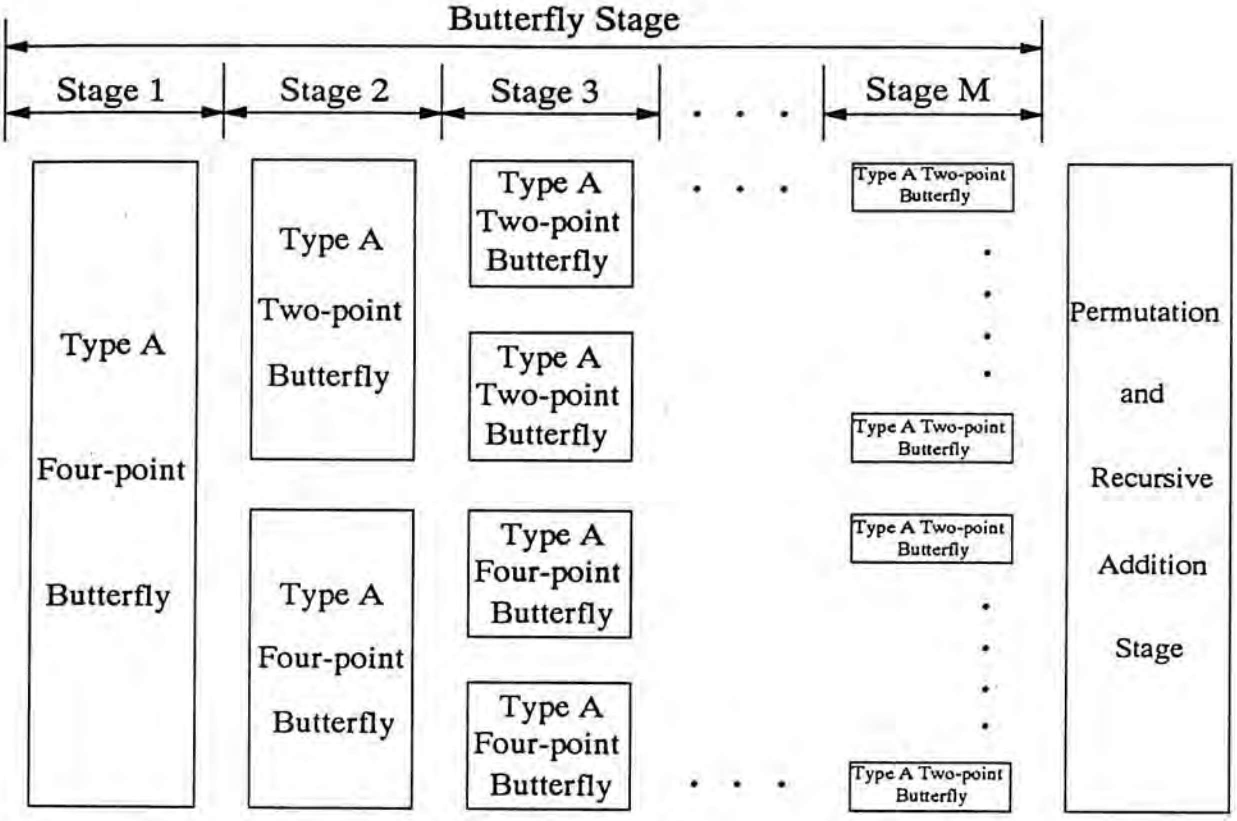

Butterfly Stage
Stage 1
Stage 2
Stage 3
Stage M
Type A Four-point Butterfly
Type A Two-point Butterfly
Type A Four-point Butterfly
Type A Two-point Butterfly
Type A Two-point Butterfly
Type A Four-point Butterfly
Type A Four-point Butterfly
Type A Two-point Butterfly
Type A Two-point Butterfly
Type A Two-point Butterfly
Type A Two-point Butterfly
Permutation and Recursive Addition Stage

Figure 4.1 Basic structure of the direct fast DCT algorithm

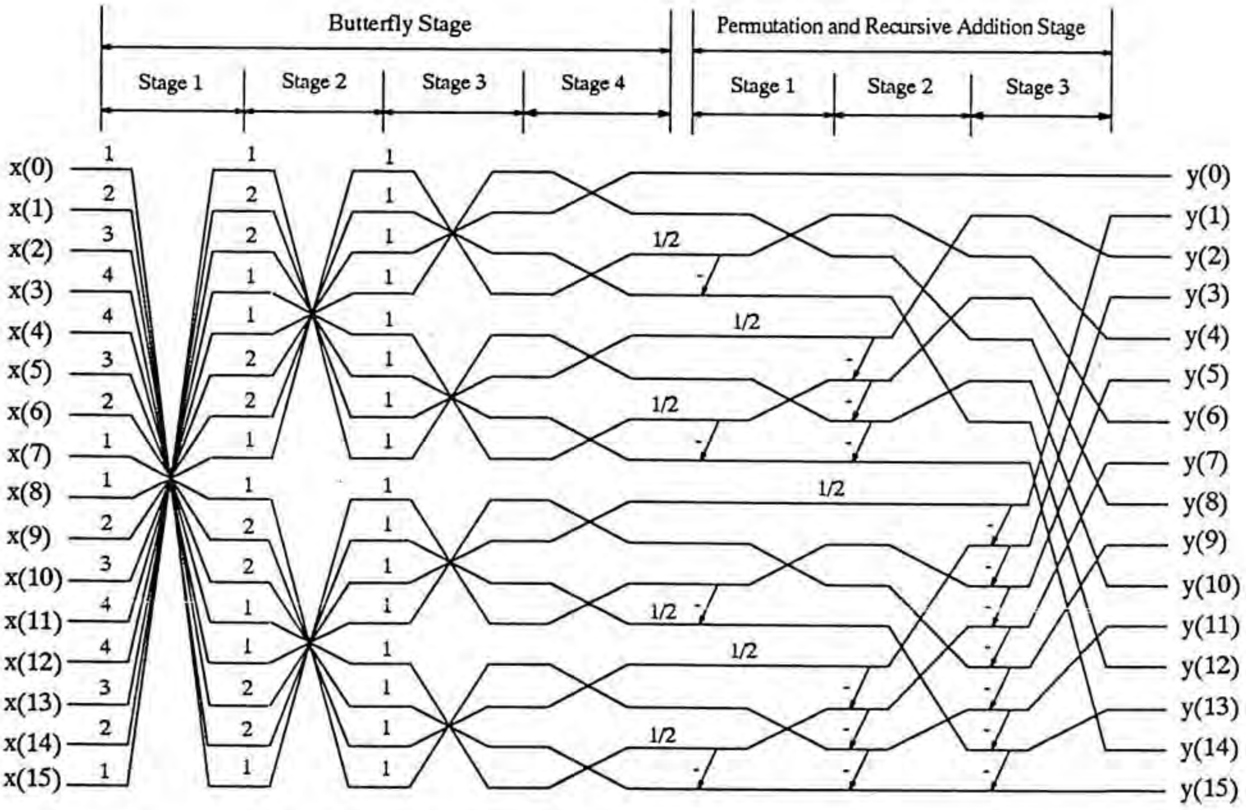
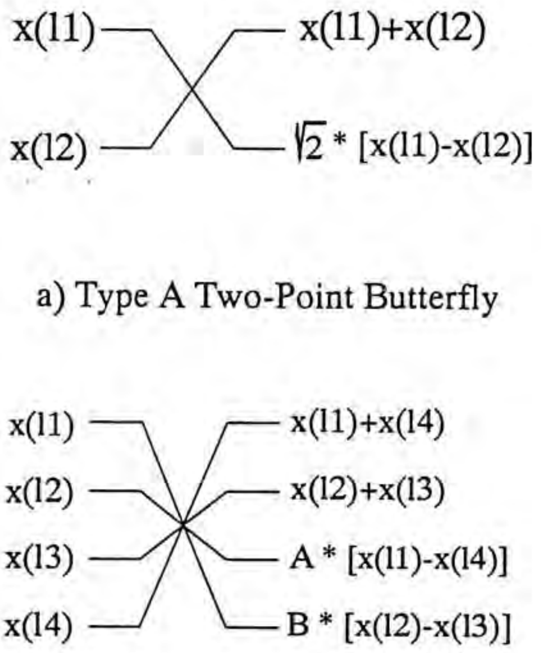


Figure 4.2 Signal flow diagram of the order-16 direct fast DCT algorithm (where the number in the butterfly represents the butterfly index in ith stage)

The structure of the direct fast DCT algorithm could be depicted using blocks as shown in Figure 4.1. The first butterfly stage represents a block, the second stage is divided into two blocks, the third into four blocks, and so on. Totally, there are M butterfly stages for an order-N DCT, where $N=2^M$. Four-point butterflies are employed in the first M-1 stages and followed by two-point butterflies in the final stage. This arrangement ensures that the output of the butterfly stage would be in bit-reversal order. The basic butterflies for the direct fast DCT algorithms, called type A four-point butterfly and type A two-point butterfly, are shown in Figure 4.3.



a) Type A Two-Point Butterfly

$$A = 2\cos\frac{(2j-1)}{2^{M+2-i}} \qquad B = 2\cos\frac{2^{M+1-i} - (2j-1)}{2^{M+2-i}}$$

for $i = 1, 2, \ldots, M-1$ and $j = 1, 2, \ldots, 2^{M-1-i}$

b) Type A Four-Point Butterfly

Figure 4.3 Basic butterflies in the direct fast DCT algorithm (where i is the butterfly stage index and j is the butterfly index in ith stage)

It can be seen that implementation of the two-point butterfly requires two additions and one multiplication and the four-point butterfly computation requires four additions and two multiplications. On the other hand, there are totally M-1 stages in the permutation and recursive stage which requires only additions.

4.2.2 Weighted Cosine Transform

The weighted cosine transform can be obtained by multiplying the DCT matrices with some simple-structured orthogonal matrices as given by eqns. (3.9) and (3.11). Therefore, the direct fast algorithm for the WCT is similar to the DCT with little modification. The basic structure of the direct fast WCT algorithm is shown in Figure 4.4. Figure 3.1 shows a signal flow diagram for the order-8 fast WCT.

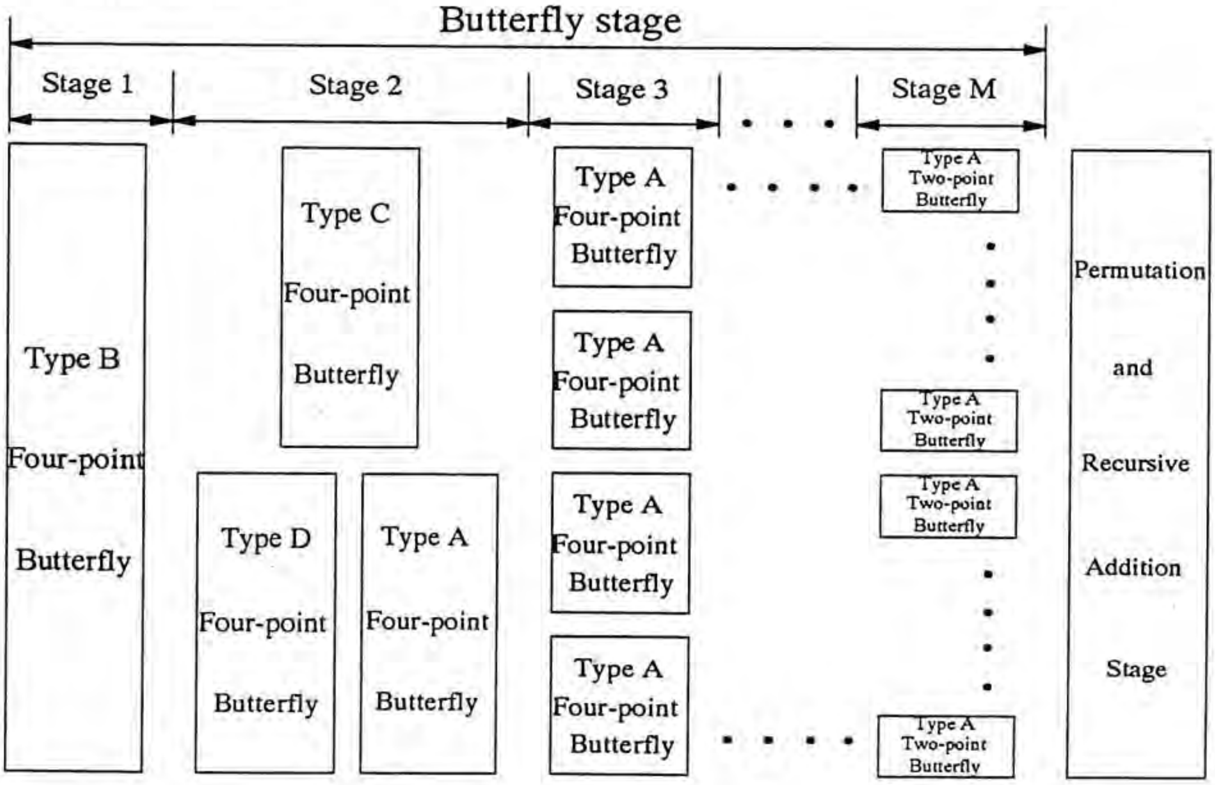
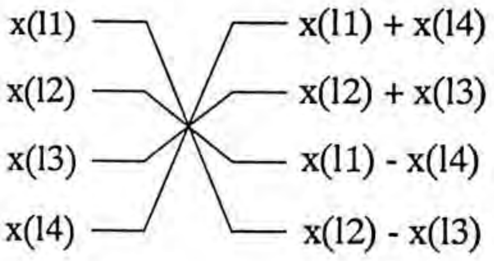
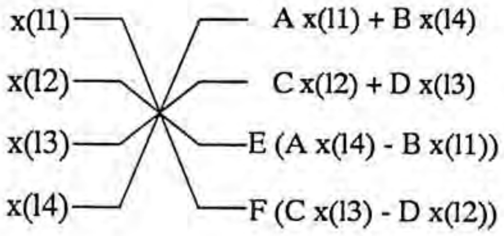

Butterfly stage
Stage 1
Stage 2
Stage 3
Stage M
Type B Four-point Butterfly
Type C Four-point Butterfly
Type D Four-point Butterfly
Type A Four-point Butterfly
Type A Four-point Butterfly
Type A Four-point Butterfly
Type A Four-point Butterfly
Type A Four-point Butterfly
Type A Two-point Butterfly
Type A Two-point Butterfly
Type A Two-point Butterfly
Type A Two-point Butterfly
Permutation and Recursive Addition Stage

Figure 4.4 Basic structure of the direct fast WCT algorithm

For an order-N WCT, where $N=2^M$, there are M butterfly stages and M-1 permutation and recursive addition stages. Compared with the DCT algorithm, the direct fast WCT algorithm has different structure in the first two butterfly stages. Three new four-point butterflies are used in the first two stages and they are shown in Figure 4.5.



a. Type B four-point butterfly

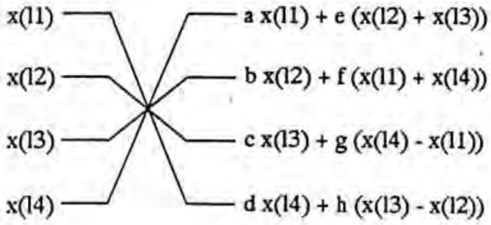


$$A = \sin\left(\frac{\pi}{4} - \frac{2^{M-1}-2j+1}{2}\alpha\right) \quad B = \sin\left(\frac{\pi}{4} + \frac{2^{M-1}-2j+1}{2}\alpha\right) \quad C = \sin\left(\frac{\pi}{4} - \frac{2j-1}{2}\alpha\right) \quad D = \sin\left(\frac{\pi}{4} + \frac{2j-1}{2}\alpha\right)$$

$$E = 2\cos\frac{(2j-1)}{2^M} \qquad F = 2\cos\frac{2^{M-1}-(2j-1)}{2^M}$$

for $i = 1, 2, ..., M-1$ and $j = 1, 2, ..., 2^{M-1-i}$

b. Type C four-point butterfly


x(l1)
x(l2)
x(l3)
x(l4)
a x(l1) + e (x(l2) + x(l3))
b x(l2) + f (x(l1) + x(l4))
c x(l3) + g (x(l4) - x(l1))
d x(l4) + h (x(l3) - x(l2))

$$a = 2\cos\frac{2^{M-2}-1}{2}\beta\ \cos\frac{2j-1}{2^{M+1}} \quad b = 2\cos\frac{2^{M-2}-1}{2}\beta\ \cos\frac{2^{M-1}-2j+1}{2^{M+1}} \quad c = 2\cos\frac{2^{M-2}-1}{2}\beta\ \cos\frac{2^{M-2}+2j-1}{2^{M+1}} \quad d = 2\cos\frac{2^{M-2}-1}{2}\beta\ \cos\frac{2^{M}-2j+1}{2^{M+1}}$$

$$e = \sqrt{2}\sin\frac{2^{M-2}-1}{2}\beta\ \cos\frac{2j-1}{2^{M+1}} \quad f = \sqrt{2}\sin\frac{2^{M-2}-1}{2}\beta\ \cos\frac{2^{M-1}-2j+1}{2^{M+1}} \quad g = \sqrt{2}\sin\frac{2^{M-2}-1}{2}\beta\ \cos\frac{2^{M-2}+2j-1}{2^{M+1}} \quad h = \sqrt{2}\sin\frac{2^{M-2}-1}{2}\beta\ \cos\frac{2^{M}-2j+1}{2^{M+1}}$$

for $i = 1, 2, ..., M-1$ and $j = 1, 2, ..., 2^{M-1-i}$

c. Type D four-point butterfly

Figure 4.5 Different types of four-point butterfly for the direct fast WCT algorithm

It can be seen that implementation of the type B four-point butterfly requires four additions, computation of the type C four-point butterfly requires ten multiplications and four additions and the type D four-point butterfly computation requires eight additions and eight multiplications. On the other hand, computation of the permutation and recursive stage requires only additions.

4.2.3 Simplified Cosine Transform

The odd part of the SCT matrix is the same as the DCT while the even part is a Walsh-liked simple-structured orthogonal matrix as given by eqn. (3.6). Therefore, the direct fast SCT algorithm is similar to the DCT with only difference in the even part. The basic structure of the algorithm is shown in Figure 4.6. Figure 3.2 shows a signal flow diagram for the order-8 fast SCT.

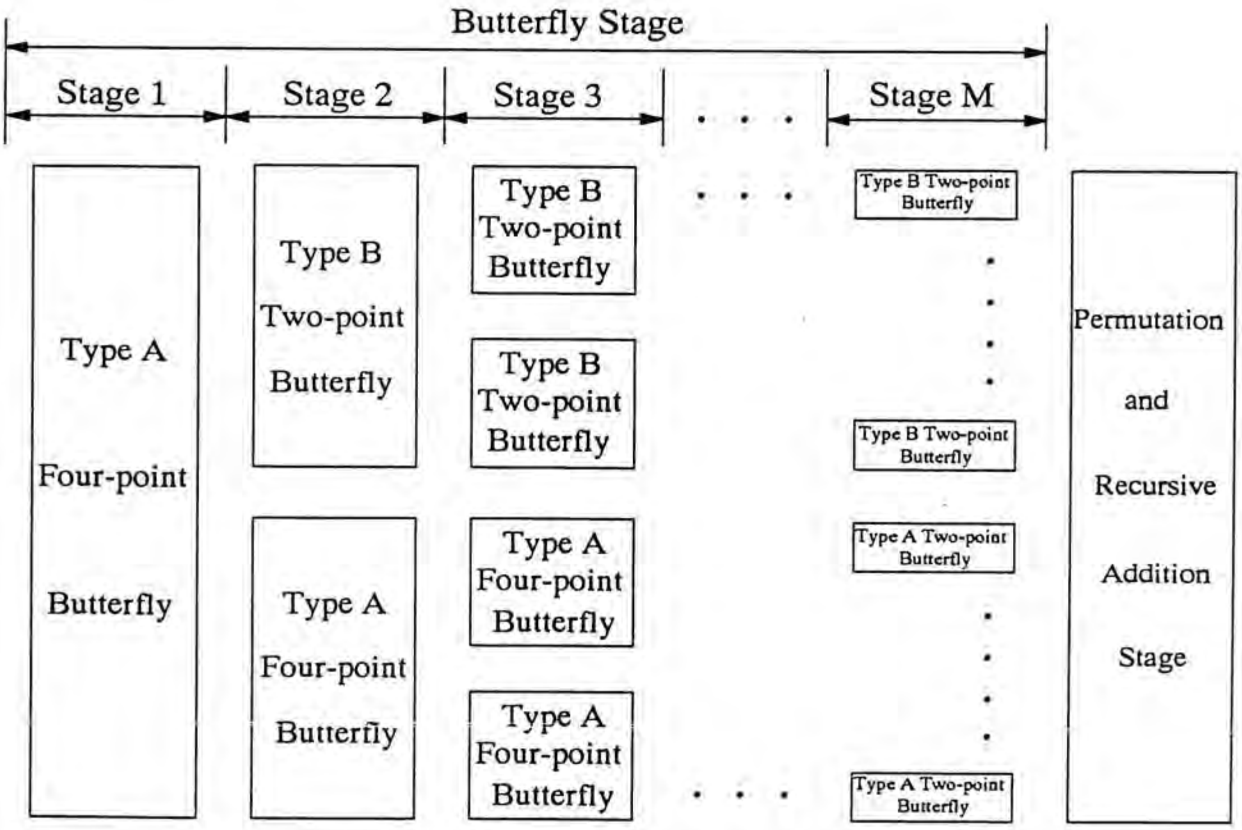

Butterfly Stage
Stage 1
Stage 2
Stage 3
Stage M
Type A Four-point Butterfly
Type B Two-point Butterfly
Type A Four-point Butterfly
Type B Two-point Butterfly
Type B Two-point Butterfly
Type A Four-point Butterfly
Type A Four-point Butterfly
Type B Two-point Butterfly
Type B Two-point Butterfly
Type A Two-point Butterfly
Type A Two-point Butterfly
Permutation and Recursive Addition Stage

Figure 4.6 Basic structure of the direct fast SCT algorithm

For an order-N SCT, where N=2^M, there are M butterfly stages in the flow diagram. Compared with the DCT algorithm, the WCT has different structure in the upper half (even part) from the second butterfly stage. A new two-point butterfly, called the type B two-point butterfly, is shown in Figure 4.7.

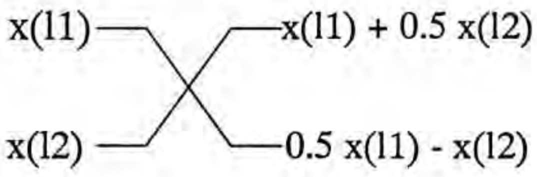


Figure 4.7 Type B two-point butterfly for the direct fast SCT algorithm

The multiplier of the type B two-point butterfly is 1/2 and which can be computed via a simple logical shift operation. Therefore, implementation of the type B two-point butterfly requires only two additions and no multiplications.

4.3 PRUNING IN FAST ALGORITHMS

In this section, we would consider pruning in the direct fast algorithms for DCT , WCT and SCT in the case that only the first P transform coefficients are required to be computed for an order N transform, where P=2^R, N=2^M and R<M.

4.3.1 Discrete Cosine Transform

Pruning is possible in both the butterfly stage and the permutation and recursive addition stage. For example, the signal flow diagram of the pruned order-16 direct fast DCT algorithm when only the first two coefficients are required to be computed is shown in Figure 4.8.

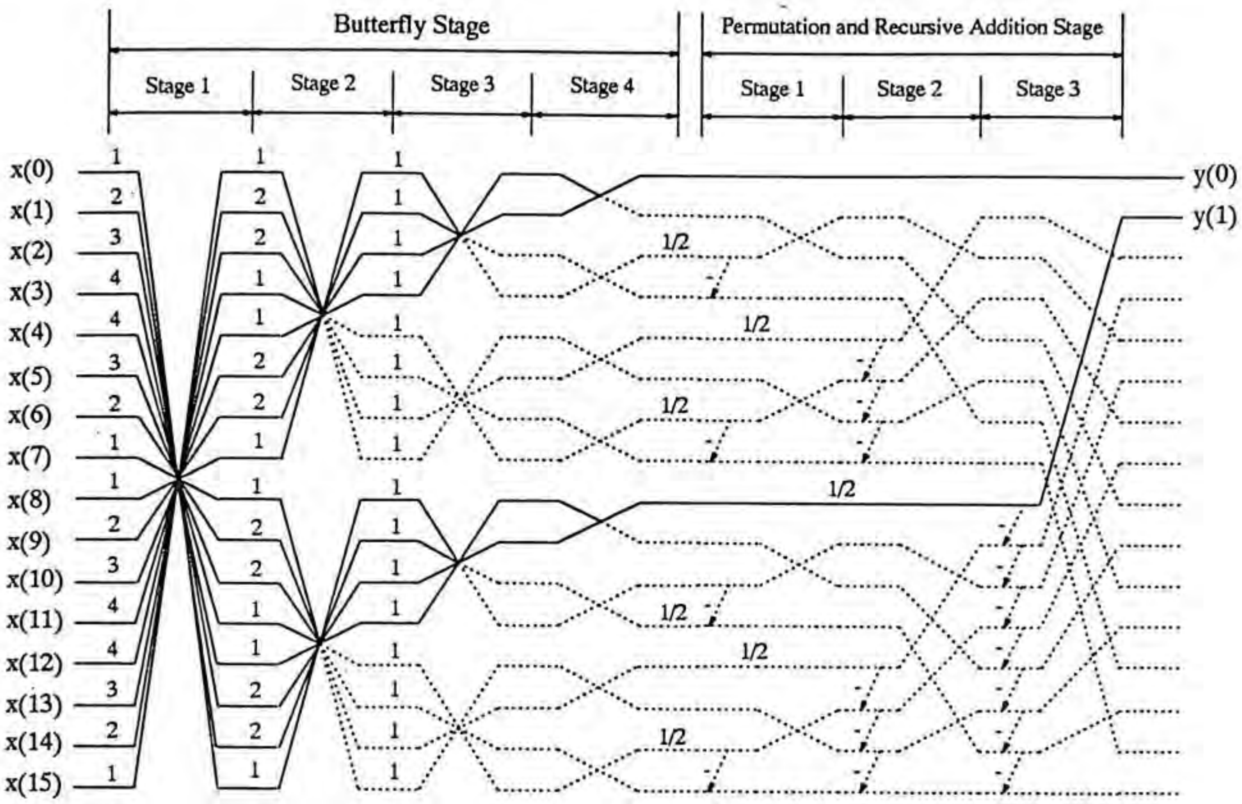


Figure 4.8 Signal flow diagram of the order-16 pruned direct fast DCT algorithm with only the first two coefficients are required to be computed

The solid lines represent the actual signal flow of the pruned algorithm and the dot lines represent the signal flow for the unpruned algorithm. It is clear that pruning is allowed at the last M-R stages of the butterfly stage and the possible pruned butterflies are given below :

a) Type A two-point butterflies without output :

In this case, no operations are required and the total number of such butterflies is equal to

$$NU_{Da} = \frac{N}{2} - P \qquad (4.6.a)$$

b) Type A two-point butterflies with the first output :

Only one addition is required and the total number of such butterflies is equal to

$$NU_{Db} = P \qquad (4.6.b)$$

c) Type A four-point butterflies without output :

Such butterflies require no operations and exist in the last M-R-2 stages only if R is smaller than M-2. At stage k (where k is the reverse order of i, i.e. k=M-i), there are 2^{M-2} - 2^{R+k-1} such butterflies. The total number of these pruned butterflies is equal to

$$NU_{Dc} = \sum_{k=1}^{M-R-2} (2^{M-2} - 2^{R+k-1})$$

$$= \begin{cases} \frac{N}{4}(M - R - 3) + P & \textit{for} \;\; R < M\text{-}2 \\ 0 & \textit{otherwise} \end{cases} \qquad (4.6.c)$$

d) Type A four-point butterflies with the first and second outputs :

Such butterflies require two additions and exist in the last M-R-1 stages. There are 2^{R+k-1} of such butterflies at stage k, therefore, the total number of these butterflies is then

$$NU_{Dd} = \sum_{k=1}^{M-R-1} 2^{R+k-1}$$

$$= \begin{cases} \frac{N}{2} - P & \textit{for} \;\; R < M\text{-}1 \\ 0 & \textit{otherwise} \end{cases} \qquad (4.6.d)$$

On the other hand, pruning is also possible at the permutation and recursive addition stage. Additions are required only for the last R stages. At the ith stage, it requires 2^{R-1} - 2^{i-1} additions, therefore, the number of additions required for computing the first P coefficients is equal to

$$NA_{Dr} = \sum_{i=i}^{R} (2^{R-1} - 2^{i-1})$$

$$= \begin{cases} P\left(\frac{R}{2} - 1\right) + 1 & \textit{for} \;\; R \geq 2 \\ 0 & \textit{otherwise} \end{cases} \qquad (4.7)$$

4.3.2 Weighted Cosine Transform

Similar to the DCT, pruning is possible at both the butterfly stage and the permutation and recursive addition stage. It is clear that pruning is allowed at the last M-R stages of the butterfly stage. The possible pruned butterflies are given below :

a) Type A two-point butterflies without output :

In this case, no operations are required and the total number of such butterflies is equal to

$$NU_{W_a} = \frac{N}{2} - P \tag{4.8.a}$$

b) Type A two-point butterflies with the first output :

Only one addition is required and the total number of such butterflies is equal to

$$NU_{W_b} = P \tag{4.8.b}$$

c) Type A four-point butterflies without output :

Such butterflies require no operations and exist in the last M-R-2 stages only if R is smaller than M-2. At stage k (where k is the reverse order of i, i.e. k=M-i), there are 2^{M-2} - 2^{R+k-1} such butterflies. The total number of these pruned butterflies is equal to

$$\begin{aligned} NU_{W_c} &= \sum_{k=1}^{M-R-2} (2^{M-2} - 2^{R+k-1}) \\ &= \begin{cases} \frac{N}{4}(M - R - 3) + P & \textit{for} \;\; R < M\text{-}2 \\ 0 & \textit{otherwise} \end{cases} \end{aligned} \tag{4.8.c}$$

d) Type A four-point butterflies with the first and second outputs :

Such butterflies require two additions and exist in the last M-R-1 stages. There are 2^{R+k-1} of such butterflies at stage k, therefore, the total number of these butterflies is then

$$NU_{W\,d} = \begin{cases} \frac{N}{8} - 1 & \textit{for} \quad R = 0 \\ \frac{3}{8}N - 2 & \textit{for} \quad R = 1 \\ \frac{N}{2} - P & \textit{for} \quad 1 < R < M\text{-}1 \\ 0 & \textit{otherwise} \end{cases} \qquad (4.8.d)$$

e) Type B four-point butterflies with the first and second outputs :

Such butterflies require two additions and exist only when R=0. The total number of such butterflies is equal to

$$NU_{W\,e} = 2^{M\text{-}2} \qquad (4.8.e)$$

f) type C four-point butterflies with the first and second outputs :

Such butterflies require two additions and exist only when R=0 and R=1. The total number of such butterflies is equal to

$$NU_{W\,f} = 2^{M\text{-}3} \qquad (4.8.f)$$

g) type D four point butterflies without output :

Such butterflies require no operations and exist only when R=0. The total number of such butterflies is equal to

$$NU_{W\,g} = 2^{M\text{-}3} \qquad (4.8.g)$$

On the other hand, it is seen that pruning is also possible at the permutation and recursive addition stage. The number of additions required for computing the first P coefficients is equal to those of the DCT given by eqn. (4.7), i.e. $NA_{W_r} = NA_{D_r}$.

4.3.3 Simplified Cosine Transform

Similar to those of the DCT and WCT, pruning is possible at both the butterfly stage and the permutation and recursive addition stage of the fast SCT algorithm. It is clear that pruning is allowed at the last M-R stages of the butterfly stage. The possible pruned butterflies are given below :

a) Type A two-point butterflies without output :

In this case, no operations are required and the total number of such butterflies is equal to

$$NU_{S\,a} = \frac{N}{4} - \frac{P}{2} \tag{4.9.a}$$

b) Type A two-point butterflies with the first output :

Only one addition is required and the total number of such butterflies is equal to

$$NU_{S\,b} = \frac{P}{2} \tag{4.9.b}$$

c) Type A four-point butterflies without output :

Such butterflies require no operations and exist in the last M-R-3 stages only if R is smaller than M-2. At stage k (where k is the reverse order of i, i.e. k=M-i), there are 2^{M-2} - 2^{R+k-1} such butterflies. The total number of these pruned butterflies is equal to

$$\begin{aligned} NU_{S\,c} &= \sum_{k=1}^{M\text{-}R\text{-}3} (2^{M-2} - 2^{R+k-1}) \\ &= \begin{cases} \frac{N}{8}(\mathrm{M - R - 3}) + \frac{P}{2} & \textit{for}\ \ \mathrm{R < M\text{-}2} \\ 0 & \textit{otherwise} \end{cases} \end{aligned} \tag{4.9.c}$$

d) Type A four-point butterflies with the first and second outputs :

Such butterflies require two additions and exist in the last M-R-2 stages. There are 2^{R+k-1} of such butterflies at stage k, therefore, the total number of these butterflies is then

$$\mathrm{NU}_{S\,d} = \sum_{k=1}^{M\text{-}R\text{-}2} 2^{R+k-1} = \begin{cases} \frac{N}{4} - \frac{P}{2} & \textit{for}\ \ R < M\text{-}1 \\ 0 & \textit{otherwise} \end{cases} \tag{4.9.d}$$

e) Type B two-point butterflies without outputs :

In this case, no operations are required and the total number of such butterflies is equal to

$$\mathrm{NU}_{S\,e} = \begin{cases} \frac{N}{4}(M - R - 3) + P & \textit{for}\ \ R = 0 \\ \frac{N}{4}(M - R - 3) + \frac{P}{2} & \textit{for}\ \ 0 < R < M\text{-}2 \\ 0 & \textit{otherwise} \end{cases} \tag{4.9.e}$$

f) Type B two-point butterflies with only the first output :

Such butterflies require one addition only and the total number of such butterflies is equal to

$$\mathrm{NU}_{S\,f} = \begin{cases} \frac{N}{2} - P & \textit{for}\ \ R = 0 \\ \frac{1}{2}(N - P) & \textit{for}\ \ 0 < R < M\text{-}1 \\ 0 & \textit{otherwise} \end{cases} \tag{4.9.f}$$

On the other hand, it is seen that pruning is also possible at the permutation and recursive addition stage. Additions are required only for the last R-1 stages. At the i^{th} stage, it requires 2^{R-1} - 2^{i-1} additions, therefore, the number of additions required for computing the first P coefficients is equal to

$$
\begin{aligned}
NA_{S\,r} &= \sum_{i=1}^{R-1}(2^{R-2} - 2^{i-1}) \\
&= \begin{cases} \frac{P}{4}(R-3)+1 & \textit{for} \;\; R \geq 2 \\ 0 & \textit{otherwise} \end{cases} \qquad (4.10)
\end{aligned}
$$

4.4 OPERATIONS SAVED BY USING PRUNING

4.4.1 Discrete Cosine Transform

For the butterfly stage, the number of additions saved ($NAS_{D\,B}$) and number of multiplications saved (NMS_D) are given by

$$
\begin{aligned}
NAS_{D\,B} &= 2\,NU_{D\,a} + NU_{D\,b} + 4\,NU_{D\,c} + 2\,NU_{D\,d} \\
&= N(\,M - R - 1) + P \qquad (4.11.a)
\end{aligned}
$$

$$
\begin{aligned}
NMS_D &= NU_{D\,a} + NU_{D\,b} + 2\,NU_{D\,c} + 2\,NU_{D\,d} \\
&= \frac{N}{2}(M\text{-}R) \qquad (4.11.b)
\end{aligned}
$$

For the permutation and recursive addition stage, the number of additions saved is equal to

$$
NAS_{D\,P} = \begin{cases} \frac{1}{2}(\,MN - RP) - (N - P) & \textit{for} \;\; R \geq 2 \\ \frac{1}{2}MN - N + 1 & \textit{otherwise} \end{cases} \qquad (4.11.c)
$$

The total number of additions saved in the whole pruned stages is then

$$\mathrm{NAS_D} = \mathrm{NAS_{D\,B}} + \mathrm{NAS_{D\,P}} \tag{4.12}$$

By considering the 'computation' as either addition or multiplication, the total number of computations is defined as the sum of the number of additions and multiplications. For the unpruned algorithm, the total number of computations (TC) is

$$\begin{aligned} \mathrm{TC_D} &= \text{number of multiplications} + \text{number of additions} \\ &= \left(\frac{1}{2}\mathrm{MN}\right) + \left(\frac{3}{2}\mathrm{MN} - \mathrm{N} + 1\right) \\ &= 2\mathrm{MN} - \mathrm{N} + 1 \end{aligned} \tag{4.13}$$

and the total number of computations saved is

$$\mathrm{TCS_D} = \mathrm{NAS_D} + \mathrm{NMS_D} \tag{4.14}$$

Let the percentage of operations saved be

$$\mathrm{POS_D} = \frac{\mathrm{TCS_D}}{\mathrm{TC_D}} \times 100\ \% \tag{4.15}$$

Figure 4.9 plots POS as a function of M and R. The FORTRAN implementation of the direct fast DCT algorithm with pruning is listed in Appendix C.

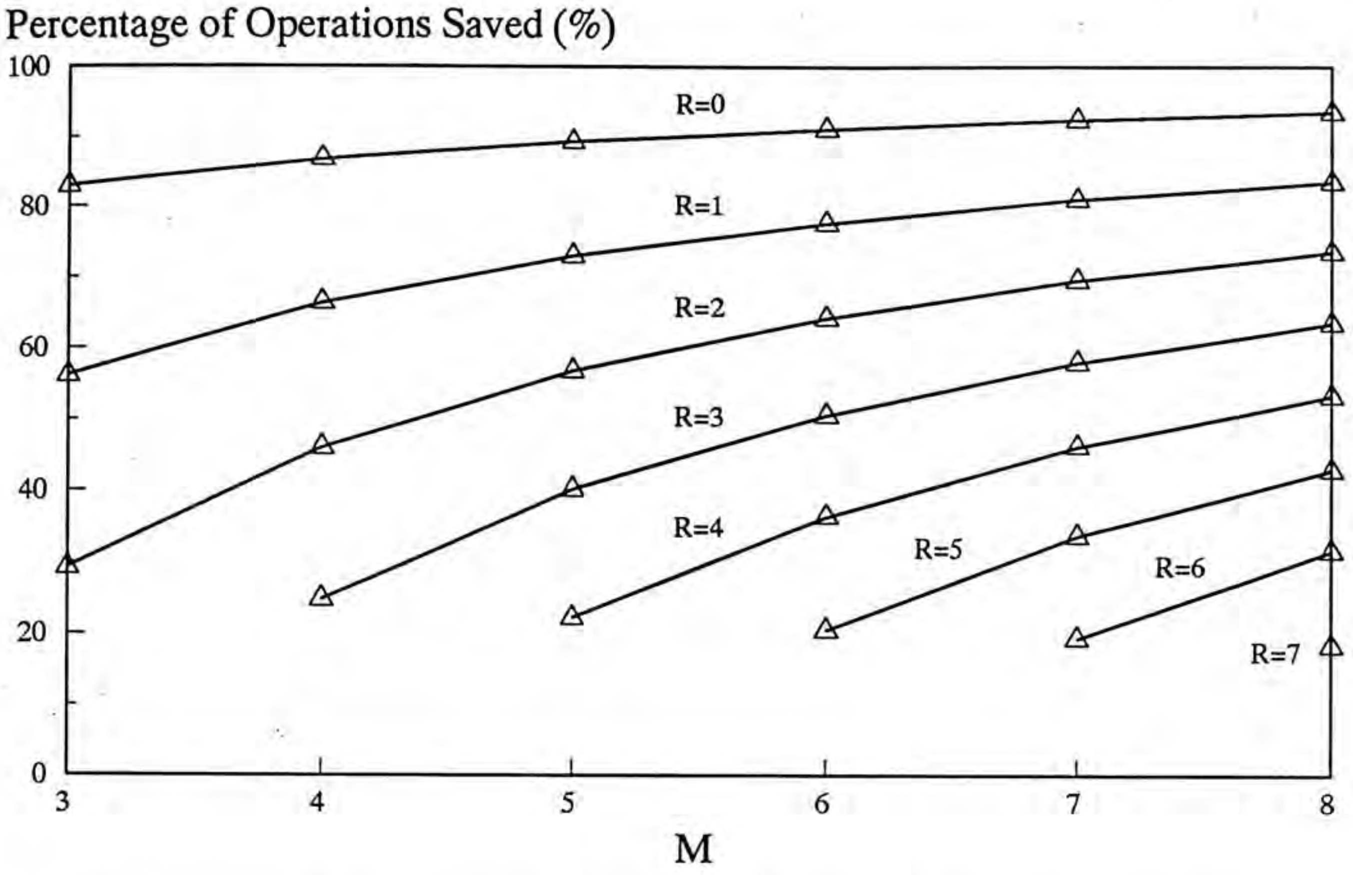


Figure 4.9 Percentage of operations saved when only the first 2^R transform coefficients are required to compute for an order-2^M DCT

In Figure 4.9, it is revealed that there is a significant saving of computations when pruning is incorporated into the direct fast DCT algorithm. For example, when there are only the first four coefficients to be computed for an order-8 transform, there is about 30% of the computations required by the unpruned algorithm can be saved. Table 4.1 lists the actual number of computations required by the proposed pruned direct fast DCT algorithm and the Wang's algorithm [WANG91].

	No. of Additions		No. of Multiplications	
P	Wang [WANG91]	This method	Wang [WANG91]	This method
1	15	15	0	0
2	30	30	8	8
4	48	45	16	16
8	66	61	24	24
16	81	81	32	32

a. order-16

	No. of Additions		No. of Multiplications	
P	Wang [WANG91]	This method	Wang [WANG91]	This method
1	31	31	0	0
2	62	62	16	16
4	100	93	32	32
8	140	125	48	48
16	178	161	64	64
32	209	209	80	80

b. order-32

Table 4.1 Computational complexity for the pruned fast DCT algorithms

In Table 4.1, it is shown that the number of multiplications required by both algorithms are the same, however, the proposed algorithm requires less number of additions than the Wang's method when P is larger than 2 and smaller than N.

4.4.2 Weighted Cosine Transform

For the butterfly stage, the number of additions saved ($NAS_{W\,B}$) and number of multiplications saved (NMS_W) are given by

$$NAS_{WB} = 2NU_{W_a} + NU_{W_b} + 4NU_{W_c} + 2NU_{W_d} + 2NU_{W_e} + 2NU_{W_f} + 8NU_{W_g}$$

$$= \begin{cases} NM + P & \textit{for} \quad R = 0 \\ N(M - R - 1) + P & \textit{otherwise} \end{cases} \tag{4.16.a}$$

$$NMS_W = NU_{W_a} + NU_{W_b} + 2NU_{W_c} + 2NU_{W_d} + 6NU_{W_f} + 8NU_{W_g}$$

$$= \begin{cases} \frac{N}{2}(M\text{-}R) + \frac{7}{4}N & \text{for} \quad R = 0 \\ \frac{N}{2}(M\text{-}R) + \frac{3}{4}N & \text{for} \quad R = 1 \\ \frac{N}{2}(M\text{-}R) & \text{otherwise} \end{cases} \tag{4.16.b}$$

For the permutation and recursive addition stage, the number of additions saved is equal to

$$NAS_{WP} = \begin{cases} \frac{1}{2}(MN - RP) - (N - P) & \textit{for} \quad R \geq 2 \\ \frac{1}{2}MN - N + 1 & \textit{otherwise} \end{cases} \tag{4.16.c}$$

The total number of additions saved in the whole pruned stages is then

$$NAS_W = NAS_{WB} + NAS_{WP} \tag{4.17}$$

Hence, the total number of computations saved is

$$TCS_W = NAS_W + NMS_W \tag{4.18}$$

For the unpruned algorithm, the total number of computation is equal to

$$TC_W = \left(\frac{1}{2}MN + \frac{3}{2}N\right) + \left(\frac{3}{2}MN + 1\right)$$

$$= 2MN + \frac{3}{2}N + 1 \tag{4.19}$$

The percentage of operations saved is then

$$POS_W = \frac{TCS_W}{TC_W} \times 100\ \% \tag{4.20}$$

Figure 4.10 plots POS of the WCT as a function of M and R.

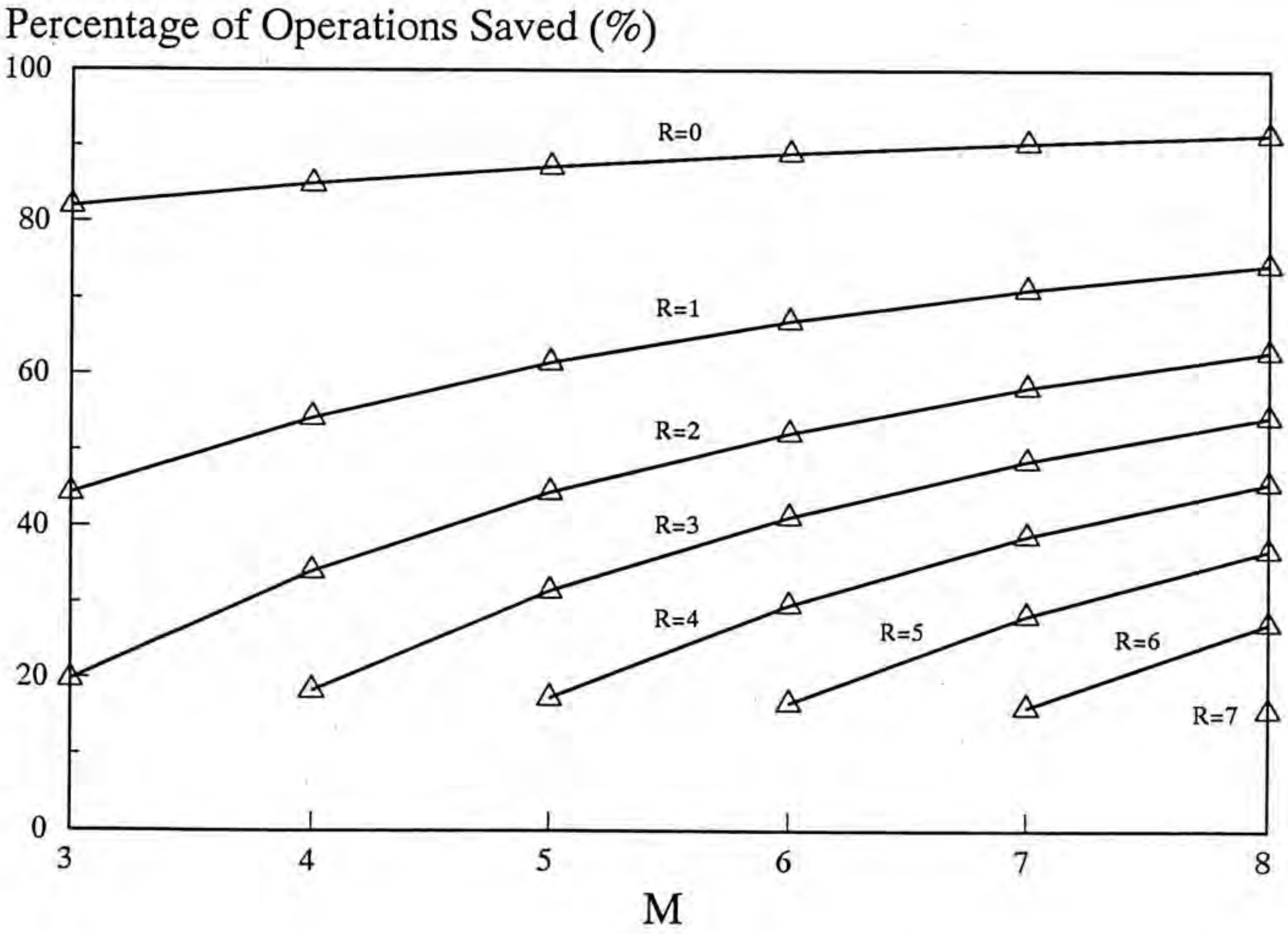


Figure 4.10 Percentage of operations saved when only the first 2^R transform coefficients are required to compute for an order-2^M WCT

Figure 4.10 reveals that the percentage saving of operations when using pruning for the WCT is less than those of the DCT. Since the WCT matrix is obtained by multiplying the

DCT matrices with several orthonormal matrices, such extra computations are occurred in the first two butterfly stages. When R is larger than one, pruning is not allowed in the first two butterfly stages and the percentage saving of operations will then become smaller when compared with that of the DCT.

4.4.3 Simplified Cosine Transform

For the butterfly stage, the number of additions saved (NAS_{SB}) and number of multiplications saved (NMS_S) are given by

$$\begin{aligned} NAS_{SB} &= 2NU_{Sa}+NU_{Sb}+4NU_{Sc}+2NU_{Sd}+2NU_{Se}+NU_{Sf} \\ &= N(M-R-1)+P \end{aligned} \tag{4.21.a}$$

$$\begin{aligned} NMS_S &= NU_{Sa}+NU_{Sb}+2NU_{Sc}+2NU_{Sd} \\ &= \begin{cases} \frac{N}{4}(M+1) & for \quad R=0 \\ \frac{N}{4}(M-R) & otherwise \end{cases} \end{aligned} \tag{4.21.b}$$

For the permutation and recursive addition stage, the number of additions saved is equal to

$$NAS_{SP} = \begin{cases} \frac{1}{4}(MN - RP - N + P) & for \quad R \geq 2 \\ \frac{1}{4}N(M-1) & otherwise \end{cases} \tag{4.21.c}$$

The total number of additions saved in the whole pruned stages is then

$$NAS_S = NAS_{SB} + NAS_{SP} \tag{4.22}$$

Hence, the total number of computations saved is

$$TCS_S = NAS_S + NMS_S \tag{4.23}$$

For the unpruned algorithm, the total number of computation is equal to

$$TC_S = \left(\frac{1}{4}N(M+1)\right) + \left(\frac{5}{4}MN - \frac{1}{4}N\right)$$

$$= \frac{3}{2}MN \tag{4.24}$$

The percentage of operations saved is then

$$POS_S = \frac{TCS_S}{TC_S} \times 100\ \% \tag{4.25}$$

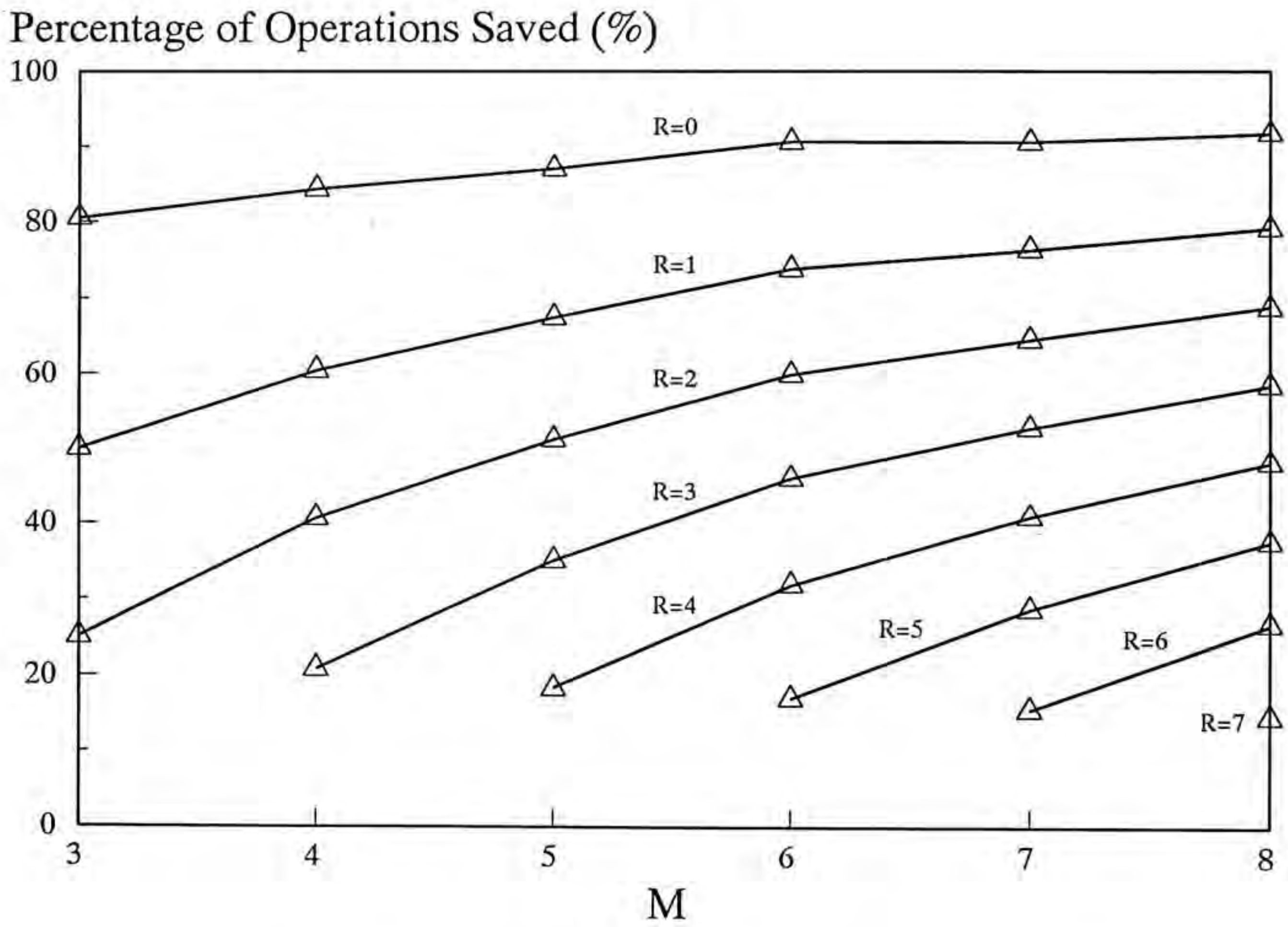


Figure 4.11 Percentage of operations saved when only the first 2^R transform coefficients are required to compute for an order-2^M SCT

Figure 4.11 plots POS of the WCT as a function of M and R. It is obviously that the percentage of operations saved by using pruning of SCT is larger than that of WCT and similar to that of the DCT.

4.4.4 Generalization for P is not an integer power of 2

In this section, we generalize the pruning algorithm for the DCT when P is an arbitrary integer. Let P be any integer smaller than N and P = 2^R + Q. There are totally M-R pruned stages in the butterfly stage. With Q being zero, the last pruned stage contains either four-point butterflies with the first two output or two-point butterflies with one output. In both situations, each extra output will lead to one extra multiplication and one extra addition. The extra number of multiplications required for Q not equal to zero is given by

$$ENM_D = 2^{M-R-1}Q \tag{4.26.a}$$

With Q=0, the first (M-R-1) pruned stages will contain four-point butterflies without output and two-point butterflies without output. In such butterflies, one extra output will need one extra addition. Therefore, the extra number of additions required when Q not equal to zero is given by

$$\begin{aligned} ENA_{D1} &= \sum_{i=1}^{M-R} 2^{i-1} Q \\ &= (2^{M-R} - 1)Q \end{aligned} \tag{4.26.b}$$

Consider the permutation and recursive addition stage. Let

$$K = \text{Integer}\left[\frac{Q}{4}\right]$$

$$L = Q \bmod 4$$

the extra number of additions required when Q is not equal to zero is

$$ENA_{D2} = 4[T(K\text{-}1) + V(K\text{-}1)]\ V(K) + [W(K{+}1)L + L - 1]V(L) \tag{4.26.c}$$

T(i) is a recursive function as follows :-

$$T(i) = T(i-1) + i$$

where T(0) = 1 and T(i) = 0 for i < 0

$$V(i) = \begin{cases} 0 & \text{if } i \le 0 \\ 1 & \text{otherwise} \end{cases}$$

and W(i) = j if T(j) ≤ i < T(j+1)

The total numbers of additions and multiplications saved by using pruning become

$$NAS_{D\,G} = NAS_D - ENA_{D\,1} - ENA_{D\,2} \tag{4.27.a}$$

$$NMS_{D\,G} = NMS\,D - ENM_D \tag{4.27.b}$$

The generalization of the pruning algorithm has been shown for the DCT. It is straightforward to develop the generalized pruning algorithms for the WCT and SCT by following the same procedure.

4.5 CONCLUDING REMARKS

The direct fast DCT algorithm proposed by Chan & Ho and the simple structure FCT (SFCT) algorithm proposed by Morikawa et al. and Wang are two of the most efficient algorithms for computing the radix-2 DCT. In this chapter, the direct fast algorithms for computing the WCT and SCT are first derived. The pruning algorithms for the direct fast algorithms of DCT, WCT and SCT are then proposed in the case where only the first P transform coefficients are required to be computed. The pruning algorithms save significant numbers of addition and multiplication operations. When comparing with the Wang's pruning algorithm for the SFCT, the proposed pruned fast DCT algorithm requires the same number of multiplications but less additions. On the other hand, the percentage of operations saved (POS) of WCT is less than that of the DCT while the POS of SCT is similar to that of the DCT.

In a transform coding system, AC and DC coefficients generated from the transform process are normally encoded and transmitted separately. Since DC coefficients contain the luminance information of the image, large error in coding the DC coefficients will lead to severely degradation of the reconstruction images. In the next chapter, a simple predictor called minimum edge difference (MED) predictor will be developed for efficient encoding of the DC coefficients in transform coding systems.

4.6 NOTE ON PUBLICATIONS

A paper entitled 'Pruning the fast discrete cosine transform' was presented at the 1991 International Symposium on Communications (ISCOM'91), Taiwan, December 1991. Another paper entitled 'Direct fast cosine transform pruning' was submitted to the IEE Proceedings (part-I) for publication. These two papers were jointly authored with Dr. W.K. Cham.

CHAPTER 5 EFFICIENT ENCODING OF DC COEFFICIENT IN TRANSFORM CODING SYSTEMS

In this chapter, a simple predictor, called minimum edge difference (MED) predictor, is proposed for efficient encoding of the DC coefficients in transform coding systems. The MED predictor predicts DC coefficients by minimizing the edge difference between the current and adjacent blocks of data. With this simple predictor, the compression ability of a transform coding system can be improved, for example by about 5% in case of JPEG baseline system.

5.1 INTRODUCTION

In the last two chapters, we developed two new orthogonal transforms and their pruned fast algorithms so as to increase the energy packing ability and also reduce the computation complexity of the transformation process. It has already been shown that in a transform coding system with optimum bit allocation, the reconstruction error variance is proportional to the geometric mean of transform coefficient variances. This implies that if we have some methods to reduce the geometric mean of the transform coefficient variances, then the reconstruction error of the overall system will be decreased. Since DC coefficients contain the most energy among all transform coefficients, reducing the variance of the DC coefficients will lead to a significant reduction of the geometric mean of transform coefficient variances. Therefore, efficient encoding of the DC coefficients in the mapping process can also improve the compression ability of a block-based image coding system.

Recently, the Joint Photographic Experts Group (JPEG), which is a sub-working group of both the CCITT and ISO, is drafting an international standard for compression of still images. The JPEG scheme [JPEG90] [CCUBE90] is basically a hybrid transform/DPCM coding system. At the encoder of the JPEG baseline system, each input 8 by 8 block of the image first undergoes a two-dimensional DCT and the transform coefficients are quantized using uniform quantizers with step-size depending on the sequency of the coefficients. The quantized AC coefficients are converted into a one-dimensional array by zigzag scanning and then encoded using a combination of run-length and Huffman coding. The quantized DC coefficients are encoded using DPCM with the prediction error Huffman coded. The JPEG scheme is a one-pass process and is relatively simple in comparison with other adaptive transform coding systems [CHENs77] [CHENp84] [NGANls89]. Normally, color images coded using the JPEG scheme have good quality at 0.75 bpp.

In JPEG scheme, differential pulse code modulation (DPCM) is used to minimize the variance of DC coefficient. The DPCM employs a simple previous element predictor which uses the DC coefficient of the previous block to predict the current one. We propose to enhance the JPEG scheme by using a new predictor called minimum edge difference (MED) predictor for more accurate prediction of DC coefficients. The MED predictor uses the edge difference between the current and adjacent blocks to predict a DC coefficient. Cham [CHAMc84] [NICOfcnc87] invented such edge difference to estimate the DC coefficients in the decoder to eliminate the need for transmission of DC coefficients. In this chapter, based on a two-dimensional Markov field, the variances of the prediction errors of the DC coefficients predicted using the MED predictor, the previous element predictor and other optimal predictors [JAYAn84] are derived for comparison. The performance upper bound of the MED predictor is also derived.

In section 5.2, the MED predictor will first be formulated and in section 5.3 its performance will then be compared with other predictors based on a Markov process described in section 2.1.2. Results of computer simulation of the JPEG scheme using the proposed and other predictors on several color test images will be given in section 5.4. Finally concluding remarks will be presented in section 5.5.

5.2 MINIMUM EDGE DIFFERENCE PREDICTOR

Consider a transform coding system which divides an image into non-overlapping blocks of size N by N. The image block is transformed into DC and AC coefficients which are then quantized and coded separately. Let the DC coefficient of the $(k,l)^{th}$ block be zero and $x_{k,l}(i,j)$ represents the $(i,j)^{th}$ pixel element of the zero DC block. Let $\vec{E}_{1\,k,l}$ and $\vec{E}_{2\,k,l}$ be the vertical and horizontal edge difference vectors respectively as shown in Figure 5.1, where A, B, C, D and E are DC coefficients.

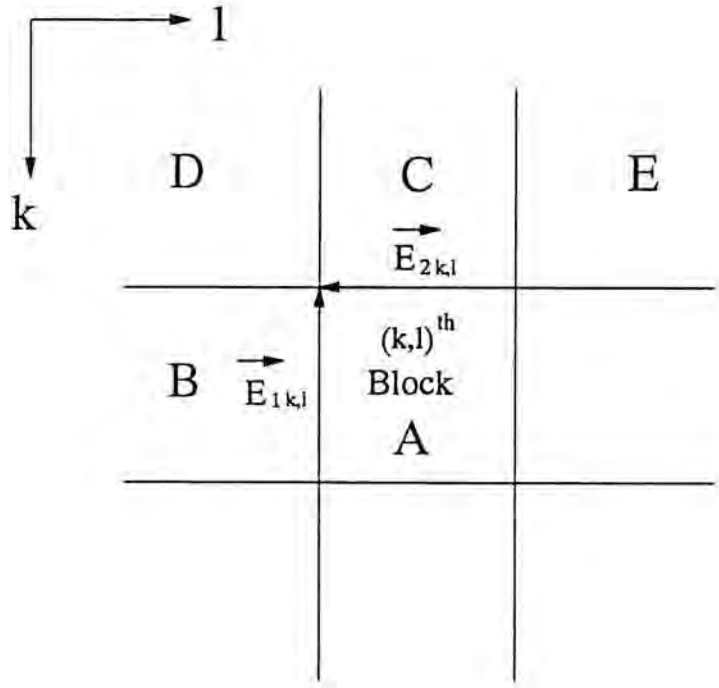


Figure 5.1 The two edge difference vectors considered in the predictor where A,B,C,D and E are the DC coefficients of the corresponding blocks

The two edge difference vectors are defined as :

$$\vec{E}_{1\,k,l} = B \times \vec{V} + \begin{bmatrix} x_{k,l-1}(0, N-1) & - & x_{k,l}(0, 0) \\ x_{k,l-1}(1, N-1) & - & x_{k,l}(1, 0) \\ x_{k,l-1}(2, N-1) & - & x_{k,l}(2, 0) \\ & . & \\ & . & \\ & . & \\ x_{k,l-1}(N-1, N-1) & - & x_{k,l}(N-1, 0) \end{bmatrix} \quad (5.1.a)$$

$$\vec{E}_{2\,k,l} = C \times \vec{V} + \begin{bmatrix} x_{k-1,l}(N-1, 0) & - & x_{k,l}(0, 0) \\ x_{k-1,l}(N-1\ 1) & - & x_{k,l}(0, 1) \\ x_{k-1,l}(N-1, 2) & - & x_{k,l}(0, 2) \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ x_{k-1,l}(N-1, N-1) & - & x_{k,l}(0, N-1) \end{bmatrix} \quad (5.1.b)$$

$$\text{for } k,l \in [1,n-1]$$

where n is the number of blocks in a row and $\vec{V}$ is a vector at the edge of the NxN DC basis picture, i.e.

$$\vec{V} = \left[\frac{1}{N}\ \frac{1}{N} \cdots \frac{1}{N} \right]^t$$

For simplicity of notation, the subscript (k,l) of the vector $\vec{E}_{1\,k,l}$ and $\vec{E}_{2\,k,l}$ will now be dropped.

If a predicted DC coefficient, $\hat{A}$, is added to the $(k,l)^{th}$ block, then the two edge difference vectors will be changed to the new vertical and horizontal edge difference vector $\vec{W}_1$ and $\vec{W}_2$:

$$\vec{W}_1 = \vec{E}_1 - \hat{A} \times \vec{V} \quad (5.2.a)$$

$$\vec{W}_2 = \vec{E}_2 - \hat{A} \times \vec{V} \quad (5.2.b)$$

The predicted value of the DC coefficient of the current $(k,l)^{th}$ block obtained from the minimum edge difference (MED) predictor is the one which minimizes the mean square edge difference (MSED) between the current block and adjacent blocks, i.e.

$$\text{MSED} = |\vec{W}_1|^2 + |\vec{W}_2|^2 \quad (5.3)$$

where $|\vec{W}_i|$ represents the norm of the vector $\vec{W}_i$. The predicted value of the DC coefficient $\hat{A}$ is therefore :

$$\hat{A}_{MED} = \frac{1}{2} \times \sum_{i=1}^{2} \sum_{j=0}^{N-1} e_i(j) \tag{5.4}$$

where $e_i(j)$ represents the j^{th} element of the edge difference vector $\vec{E}_i$. Prediction of a DC coefficient using eqn. (5.4) requires 2N additions and one right-shift operation for each NxN data block. Such MED predictor is independent of the adjacent element correlation coefficients ρ of the pixels. For the purpose of comparison, we will derive the MED predictor which is optimized with respect to ρ. The prediction formula given in eqn.(5.4) can be written in a more general form as follows :

$$\hat{A} = b \sum_{j=0}^{N-1} e_1(j) + c \sum_{j=0}^{N-1} e_2(j) \tag{5.5}$$

where b and c are the prediction coefficients and their values are selected such that the variance of the prediction error, i.e.

$$\sigma_d^2 = E\{(A - \hat{A})^2\}$$

is minimized. The optimal value of b and c can then be obtained by setting the derivatives of σ_d^2 with respect to b and c to zero. With the assumption that pixels are sampled from a zero-mean 2-D stationary random field with equal horizontal and vertical correlation coefficients, the optimal values of the prediction coefficients are found to be :

$$b_{OPT} = c_{OPT} = \frac{2 - p_{AQ}}{4 - 4p_{AQ} + p_{QQ}} \tag{5.6}$$

where

$$p_{AQ} = \frac{E\{AQ\}}{E\{A^2\}} \quad , \quad p_{QQ} = \frac{E\{Q^2\}}{E\{A^2\}} \quad \text{and} \quad Q = 2A - \sum_{i=1}^{2} \sum_{j=0}^{N-1} e_i(j)$$

Prediction of a DC coefficient using (5.5) and (5.6) requires to resolve the prediction coefficients by evaluating the covariance function of the incoming pixels. Therefore, it involves a lot of computations and also turns the coding algorithm into a two-pass process. In the next section, we will show that the MED predictor performs in fact very close to such optimal case when the adjacent element correlation coefficient of the pixels is high.

5.3 PERFORMANCE EVALUATION

In this section, we shall investigate the performance of the proposed MED predictor by using a statistical model. Assume an image which is sampled from a zero mean two-dimensional stationary random field with variance σ_x^2 and separable covariance function $COV_x(i,j,k,l)$ as given in section 2.1.2. For the purpose of simplicity, we also assume that the adjacent element correlation coefficients in both horizontal and vertical directions are equal, i.e. $\rho_h = \rho_v = \rho$. The variance of the prediction error when using the MED predictor on the image is equal to

$$\begin{aligned}\sigma_{d\,MED}^2 &= E\{(A-\hat{A}_{MED})^2\}\\ &= E\left\{\left(A-\frac{1}{2}\times\sum_{i=1}^{2}\sum_{j=0}^{N-1}e_i(j)\right)^2\right\}\\ &= \left[\;N + N\rho + \frac{1}{2}(1-\rho^N)^2 - \frac{2\rho(1-\rho^N)}{(1-\rho)}\;\right]\sigma_x^2 \end{aligned} \qquad (5.7)$$

When the optimal MED predictor as given in eqns.(5.5) and (5.6) is used, the variance of the prediction error becomes

$$\sigma_{d\,OptMED}^2 = (1-2b_{OPT})^2E\{A^2\} + 2b_{OPT}(1-2b_{OPT})E\{AQ\} + b_{OPT}^2E\{Q^2\} \qquad (5.8)$$

where

$$E\{A^2\} = \sigma_{DC}^2$$

$$= \frac{1}{N^2}\,\frac{[N(1-\rho^2)-2\rho(1-\rho^N)]^2}{(1-\rho)^4}\cdot\sigma_x^2$$

$$E\{\,AQ\,\} = \frac{2}{N}\left\{\frac{(1-\rho^N)[\,N(1-\rho^2)-2\rho(1-\rho^N)\,]}{(1-\rho)^2}\right\}\,\sigma_x^2$$

$$E\{\,Q^2\,\} = 4\left[\;N+N\rho+\frac{1}{2}(1-\rho^N)^2-\frac{2\rho(1-\rho^N)}{(1-\rho)}\;\right]\,\sigma_x^2$$

When the JPEG's simple previous element predictor is used, the variance of the prediction error is given by

$$\sigma_{d\,JPEG}^2 = E\{(A-\hat{A}_{JPEG})^2\}$$

$$= E\{(A-B)^2\}$$

$$= \frac{2}{N^2}\left[\frac{N(1-\rho^2)-2\rho(1-\rho^N)}{(1-\rho)^2}\right]\left[\frac{N(1-\rho^2)-2\rho(1-\rho^N)-\rho(1-\rho^N)^2}{(1-\rho)^2}\right]\sigma_x^2 \qquad (5.9)$$

To evaluate the performance of the proposed MED predictor, we compare it with that of JPEG by using the criterion of normalized error variance (NEV)

$$\mathrm{NEV} = \frac{\sigma_d^2}{\sigma_{DC}^2} \qquad (5.10)$$

where σ_d^2 and σ_{DC}^2 are the variances of the prediction error and the DC coefficient respectively. The lower is the NEV, the more efficient is the predictor. In addition to the JPEG's simple previous element predictor, performance of several other optimal predictors is listed for the purpose of comparison. The predicted DC coefficients and the variances of the corresponding prediction errors when using the optimum 1-element, 2-element and 3-element predictors [JAYAn84] are listed below :

Optimal 1-element predictor

$$\hat{A}_1 = \frac{E\{AB\}}{E\{A^2\}} A \quad \text{and} \quad \sigma^2_{d1} = \left[1 - \left(\frac{E\{AB\}}{E\{A^2\}}\right)^2\right] E\{A^2\} \tag{5.11}$$

Optimal 2-element predictor

$$\hat{A}_2 = \left[\frac{E\{AB\}}{E\{A^2\} + E\{BC\}}\right] (B + C) \text{ and } \sigma^2_{d2} = \left[\frac{E\{A^2\} - E\{BC\}}{E\{A^2\} + E\{BC\}}\right] E\{A^2\} \tag{5.12}$$

Optimal 3-element predictor

$$\hat{A}_3 = \frac{E\{AB\}}{E\{A^2\}} (B + C) - \frac{E\{BC\}}{E\{A^2\}} D \text{ and } \sigma^2_{d3} = \left(1 - \frac{E\{BC\}}{E\{A^2\}}\right)^2 E\{A^2\} \tag{5.13}$$

where

$$E\{AB\} = \left[\frac{\rho(1-\rho^N)^2}{N(1-\rho^2) - 2\rho(1-\rho^N)}\right] E\{A^2\}$$

$$E\{BC\} = \left[\frac{\rho^2(1-\rho^N)^4}{[N(1-\rho^2) - 2\rho(1-\rho^N)]^2}\right] E\{A^2\}$$

NEVs of the six predictors are plotted in Figure 5.2. It can be seen that all NEVs decrease as ρ increases. When ρ is larger than 0.65, the NEVs of the two MED predictors are smaller than those of other predictors and when ρ is larger than 0.8, the MED predictor performs very close to its optimal case. As ρ increases and approaches unity, NEVs of all predictors have very close values. Since ρ of pixels is larger than 0.65 in most cases, we may conclude that the MED predictor can encode the DC coefficients effectively and its optimization to ρ may not be necessary.

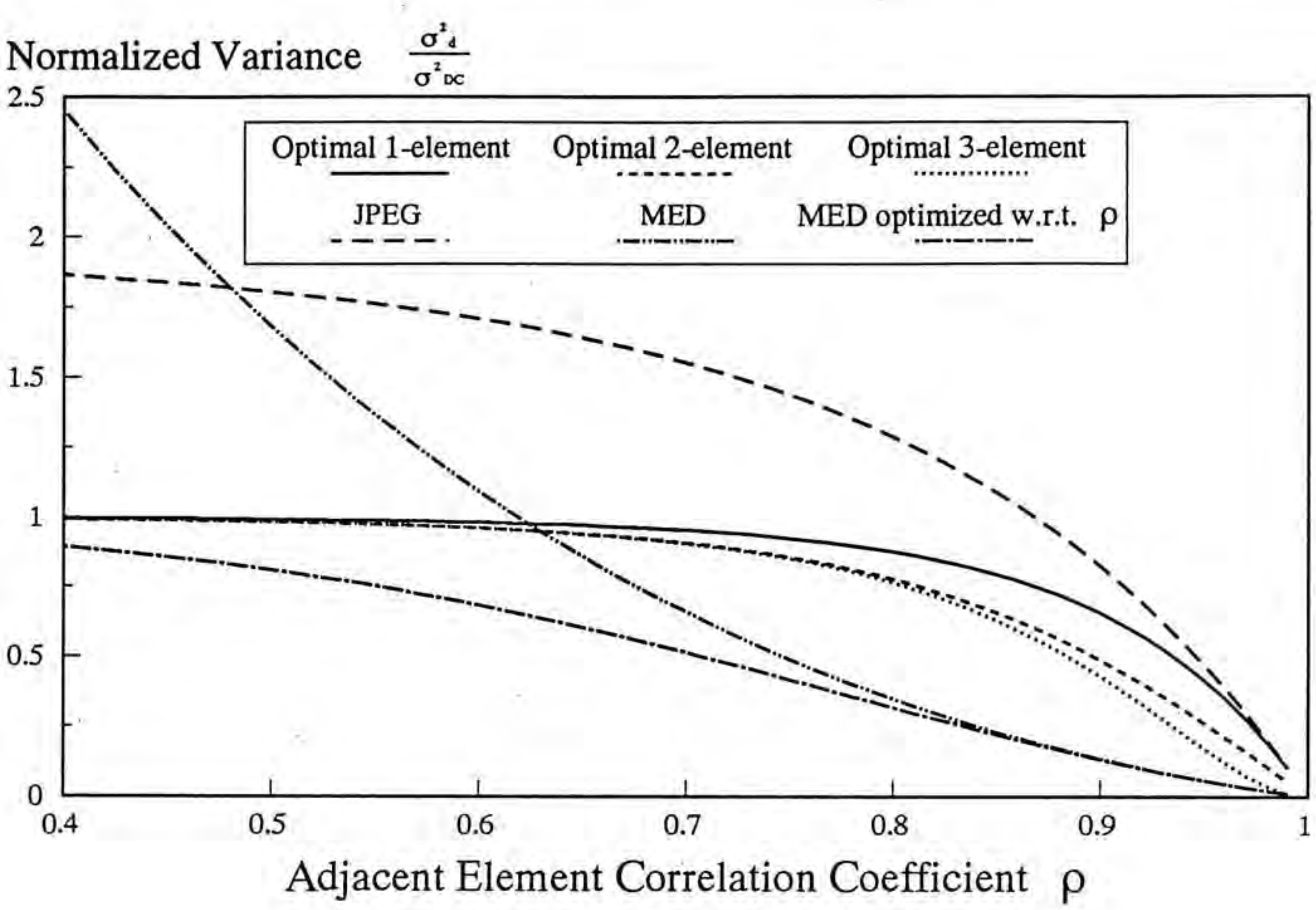


Figure 5.2 Normalized error variance for various predictors

5.4 SIMULATION RESULTS

Computer simulations of JPEG and the MED predictors have been carried out on three color images (Lenna, Peppers and Sailboat) which are of size 256x256 pixels and 8-bits resolution for each of the red (R), green (G) and blue (B) components. The block diagram of the system has already been shown in Figure 2.4 in section 2.3.4. JPEG scheme is intended to be used independently of the color space hence each color can be handled as separate components. However, the best compression result can be achieved if the color components are independent, such as YUV, where most of the information concentrates in the luminance component (Y) and less in the chrominance components (U, V). In our simulations, the input R, G, B components of the test images are first converted to Y, U, V components by using the transformation given in eqn. (2.22). After color space conversion, each component can then be coded independently by the JPEG baseline system. Appendix B shows the adjacent element correlation coefficient ρ of the Y, U, V components for the three test images.

In our experiments, the JPEG scheme together with one of the five predictors listed in Table 5.1 is applied on the test images. Whilst coding of AC coefficients are exactly the same as the JPEG baseline system, the DC coefficients are DPCM encoded using one of the five predictors. By using the same table designed for the simple previous element predictor, the prediction error of the DC coefficients for the five predictors are Huffman coded, hence the DC coefficient coding errors generated from the JPEG and the modified 'JPEG' systems are the same.

In order to demonstrate the suitability and effectiveness of the proposed MED predictor for encoding DC coefficients, we compare its performance with the JPEG's predictor, optimal 1-, 2- and 3- element predictors by using NEV described in section 5.3 as criterion. The NEV results for the Y, U, V components of the three test images are presented in Table 5.1. When JPEG scheme is applied on test images Lenna, Peppers and Sailboat at bit-rate 0.75 bpp, the mean square errors (MSEs), which are defined in the R, G, B domain, are respectively 101.8, 165.0 and 310.1 as shown on the right hand side of Table 5.1. Similarly, the MSEs are 147.1, 238.3 and 425.2 when the bit-rate is 0.5 bpp. Decoded images of the three test images are shown in Figures 5.3-5.5. It should be noted that when optimal 1-elment, 2-element, 3-element and MED predictors are employed instead of the JPEG predictor, the values of MSE remain unchanged but the values of NEV are reduced. Furthermore, Table 5.1 reveals that the MED predictor outperforms the other predictors in Y components of the images. As for the U, V components, the MED predictor always performs better than others except in some components with a high value of ρ. It agrees with the results obtained from the analytical method in the last section that when ρ is close to unity, the performance of all the predictors is very close.

To compare the relationship between the theoretical and experimental results, one may simply substitute the values of ρ listed in Appendix B into the expressions of the variances of the prediction error given in the last section. We have found that the experimental results are quite close to those of theoretical. Take the Y component of the image Lenna as an example, its ρ is equal to 0.953 as shown in Appendix B. Substitute this value into eqns.(5.7), (5.9), (5.11), (5.12) and (5.13), we obtain the values of NEV of the JPEG's predictor, optimal 1-, 2-, 3-element predictors and the MED predictor, which are 0.444, 0.394, 0.246, 0.156 and 0.054 respectively. They are close to but smaller than those in Table 5.1. In our experiments, prediction of DC coefficients is based on the quantized AC and DC coefficients. Generally, quantization errors results poorer prediction and so larger values of NEV.

Finally, to illustrate the bit rate reduction of the proposed MED predictor, we compare it with the JPEG's predictor and several well known predictors [NETRI80] [BRAIp90] which are originally used in pixel element prediction of luminance signal and constructed based on the criterion of minimizing the mean square prediction error. These predictors are listed in Table 5.2 and the tap nomenclature predictors has already been shown in Figure 5.1. Table 5.3 lists the number of bits required to code the quantized DC coefficients and the total bit-rate in bpp required for the system. It is clear that the MED predictor requires the least number of bits to encode the DC coefficients and of course the least bit-rate requirement among other predictors. Take the decoded image Lenna having MSE 147.1 as an example, the number of bits required to code the DC coefficient for the JPEG, P1, P2, P3, P4, P5, P6, P7 and MED predictors are equal to 8224, 7135, 7201, 6935, 7239, 6995, 7258, 7106 and 6480 respectively. For the JPEG baseline system, the images tested are coded at 0.5 and 0.75 bpp. By using the MED predictor, the same coded images can be represented using only about 0.47 and 0.71 bpp and that is equivalent to about 5% bit rate reduction over the standard JPEG scheme. The reduction in bit rate is due to the reduction in number of bits required for representing the DC coefficients. For example, image Lenna coded at 0.75 bpp using the MED predictor required only 7092 bits to represent DC coefficients, which is 73.5% of those required by the JPEG scheme.

	MSE = 147.1			MSE = 101.8		
Predictor	Y	U	V	Y	U	V
JPEG	0.672	0.689	1.270	0.666	0.705	1.220
Optimal 1-element	0.568	0.689	1.270	0.560	0.705	1.220
Optimal 2-element	0.255	0.210	0.662	0.254	0.202	0.524
Optimal 3-element	0.243	0.207	0.385	0.240	0.178	0.408
MED	0.055	0.205	0.635	0.040	0.139	0.386

a. Lenna

	MSE = 238.3			MSE = 165.0		
Predictor	Y	U	V	Y	U	V
JPEG	0.588	0.562	0.623	0.582	0.560	0.602
Optimal 1-element	0.503	0.507	0.623	0.496	0.508	0.602
Optimal 2-element	0.296	0.228	0.309	0.294	0.215	0.272
Optimal 3-element	0.263	0.222	0.305	0.259	0.198	0.261
MED	0.052	0.112	0.265	0.035	0.074	0.165

b. Peppers

	MSE = 425.2			MSE = 310.1		
Predictor	Y	U	V	Y	U	V
JPEG	0.396	0.397	0.459	0.394	0.367	0.403
Optimal 1-element	0.355	0.388	0.459	0.357	0.342	0.403
Optimal 2-element	0.210	0.251	0.383	0.207	0.227	0.264
Optimal 3-element	0.181	0.246	0.329	0.178	0.205	0.235
MED	0.060	0.204	0.312	0.047	0.135	0.199

c. Sailboat

Table 5.1 NEV of the Y, U, V components of the three color images when using different predictors

Predictor	b	c	d	e
JPEG	1	0	0	0
P1	1/2	0	1/2	0
P2	1	-1/2	1/2	0
P3	3/4	-3/8	5/8	0
P4	13/16	-5/16	1/2	0
P5	7/8	-1/2	1/2	1/8
P6	1/2	1/8	1/4	1/8
P7	3/4	-1/4	3/8	1/8

Table 5.2 Predictor coefficients of the eight predictors ($\hat{A}$ = b B + c C + d D + e E)

	Lenna				Peppers				Sailboat			
	MSE=147.1		MSE=101.8		MSE=238.3		MSE=165.0		MSE=425.2		MSE=310.1	
Predictor	N_{DC}	BR	N_{DC}	BR	N_{DC}	BR	N_{DC}	BR	N_{DC}	BR	N_{DC}	BR
JPEG	8224	0.50	9654	0.75	8162	0.50	9557	0.75	7202	0.50	8272	0.75
P1	7135	0.483	8165	0.727	7440	0.489	8550	0.735	6750	0.493	7596	0.740
P2	7201	0.484	8240	0.728	7537	0.491	8686	0.737	6788	0.494	7620	0.740
P3	6935	0.480	7967	0.724	7343	0.488	8457	0.733	6739	0.493	7554	0.739
P4	7239	0.485	8250	0.729	7656	0.492	8790	0.738	7148	0.499	7990	0.746
P5	6995	0.481	7990	0.725	7410	0.489	8516	0.734	6779	0.494	7595	0.740
P6	7258	0.485	8434	0.731	7543	0.491	8666	0.736	6760	0.493	7661	0.741
P7	7106	0.483	8175	0.727	7446	0.489	8274	0.730	6768	0.493	7580	0.739
MED	6480	0.473	7092	0.711	6483	0.474	7089	0.712	6308	0.486	6827	0.728

Table 5.3 Number of bits required to code the DC coefficients (N_{DC}) and the total bit-rate in bpp (BR) required to code the whole image

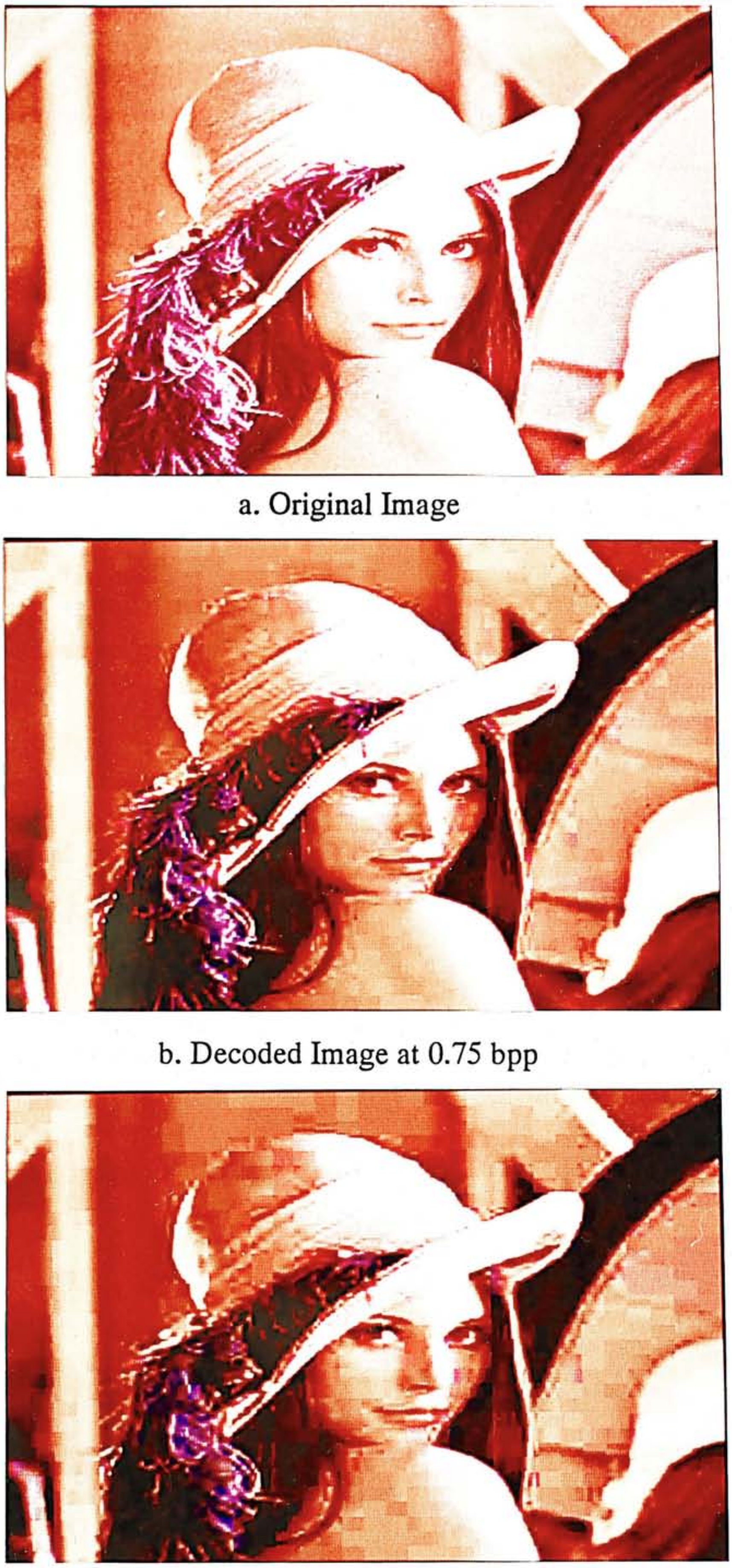
5.5 CONCLUDING REMARKS

In this chapter, a minimum edge difference (MED) predictor is proposed to enhance the performance of the JPEG baseline system. Analytical method based on the first order Markov process shows that when the adjacent element correlation coefficient of the pixel element is larger than 0.65, the MED predictor results the smallest values of the prediction error among all predictors tested and performs very close to its optimum case. Simulation results based on real images show that about 5% bit-rate reduction can be achieved when the MED predictor is adopted in the JPEG baseline system. The MED predictor requires only extra 2N addition and one right shift operations for each N by N block of the image. Therefore, it can be implemented by JPEG chip manufacturers easily as an optional feature.

When the pixels of the image are mapped into a set of coefficients in a block-based image coding system, the next process is to quantize the coefficients into different output levels. Quanitzation can either be in scalar or vector. Vector quantization has proved to have better rate-distortion performance than the scalar quantization especially at low bit-rate applications [GRAY84]. The main drawback of VQ is its computational complexity in the encoding process. To tackle this problem, we will propose two fast encoding methods for vector quantization of images in the next chapter.

5.6 NOTE ON PUBLICATIONS

A paper entitled 'Efficient encoding of DC coefficients in transform coding of images using JPEG scheme' was presented at the 1991 International Symposium on Circuits and Systems (ISCAS'91), Singapore, June 1991. Another paper entitled 'Minimum edge difference predictor for encoding the DC coefficients in transform coding of images' was submitted to IEEE Transaction on Communications for publication. These two papers were co-authored with Dr. W.K. Cham and Mr. C.T. See.

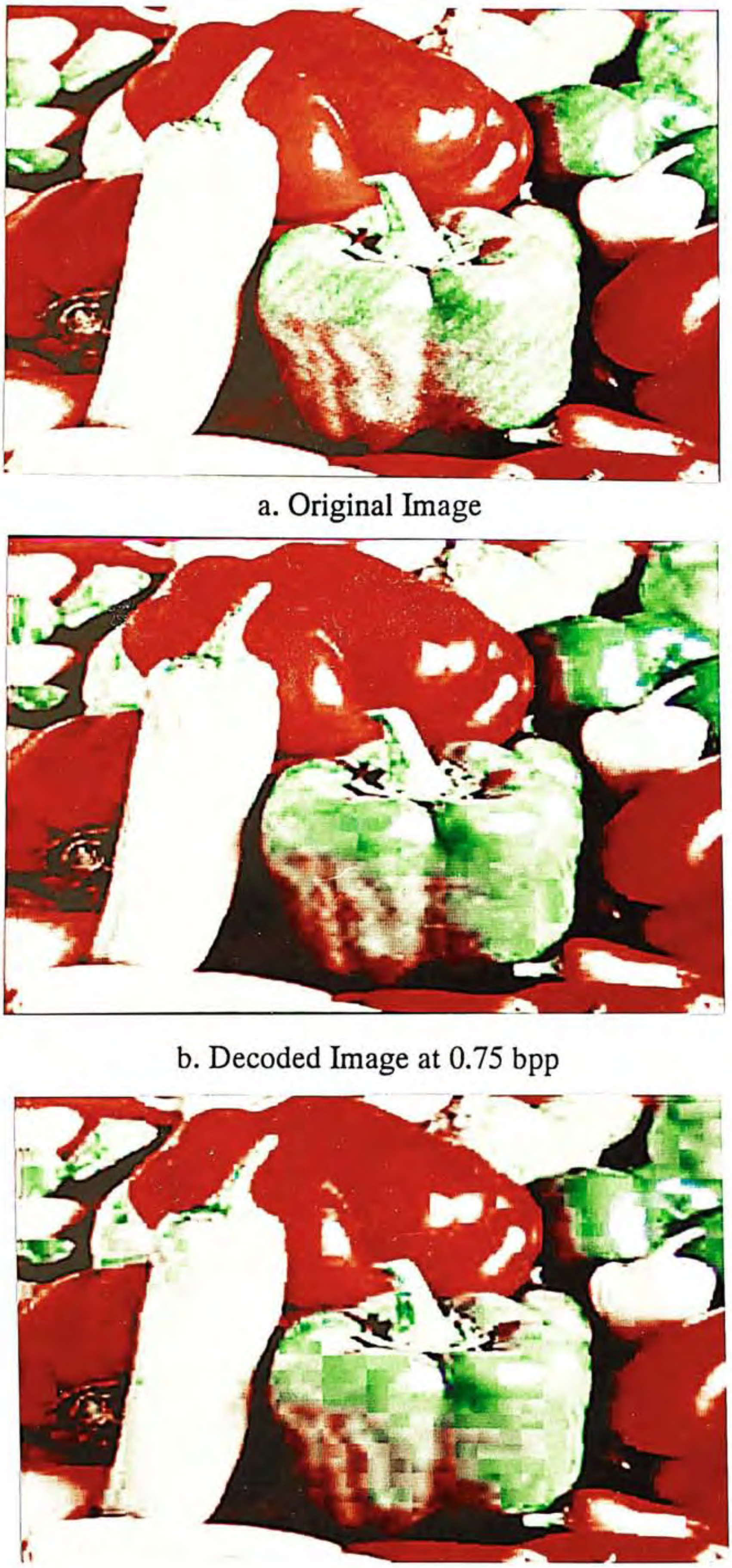


a. Original Image

b. Decoded Image at 0.75 bpp

c. Decoded Image at 0.5 bpp

Figure 5.3 Decoded images of JPEG baseline system for the test image Lenna

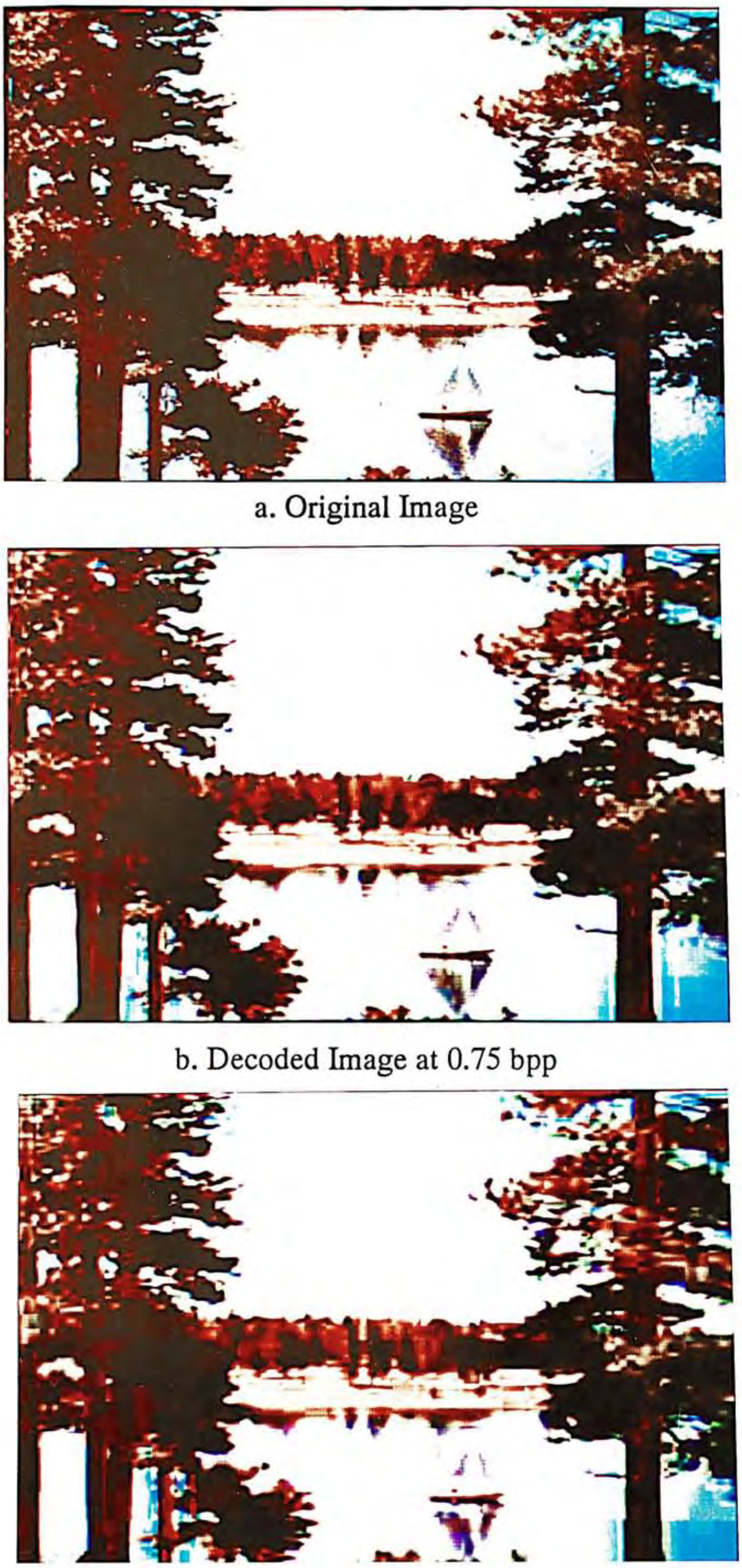


a. Original Image

b. Decoded Image at 0.75 bpp

c. Decoded Image at 0.5 bpp

Figure 5.4 Decoded images of JPEG baseline system for the test image Peppers



a. Original Image

b. Decoded Image at 0.75 bpp

c. Decoded Image at 0.5 bpp

Figure 5.5 Decoded images of JPEG baseline system for the test image Sailboat

CHAPTER 6 EFFICIENT ENCODING ALGORITHMS FOR VECTOR QUANTIZATION OF IMAGES

In this chapter, a fast VQ encoding method called sub-codebook searching (SCS) algorithm is first developed for image coding. This algorithm allows searching only a portion of the codebook to find the minimum distortion codeword of an input vector. In comparison with other existing VQ encoding algorithms, the SCS algorithm requires the minimum number of total operations. The SCS algorithm is then extended to form an image coding algorithm called predictive sub-codebook searching (PSCS) algorithm which not only has less computation requirement but also improved compression ratio when comparison with standard VQ scheme.

6.1 INTRODUCTION

In the last three chapters, we have discussed several methods to increase the efficiency of the mapping process in block-based image coding systems. In additions to the mapping operation, quantization is also an important factor to affect the efficiency of the coding system. In the past decade, vector quantization (VQ) has widely been used in image coding systems [GRAY84] [NASRk88] for its superior rate-distortion performance over traditional scalar quantization schemes. In vector quantization, an image is first divided into subsequent block of data with size k=NxN. The block of data is then linearly scanned to form a k-dimensional vector. A set of vectors from different images is then chosen as the training sequence from which a codebook of representative vectors is generated using iterative clustering methods [LINDbg80] [VAISg88] [FLANmfrn89] [EQUI89]. To encode an input vector, the codebook is searched for the closest match representative vector or codeword. Compression is achieved by representing the codeword using the index, which has a shorter length than that of the input vector. At the decoder, the image is reconstructed by simple table look-up method using the index as the address to a table containing the codewords. Hence, computation of VQ system concentrates at the encoder. For encoding an input vector, distortion between the vector and each codeword in the codebook is usually calculated to find the minimum distortion codeword. When the codebook size is M and vector dimension is k, the encoding process will require M Mk multiplications and (2k-1)*M additions for each pixel if Euclidean distortion is used. Such complexity increases exponentially with M and k and thus limits the use of large codebook sizes and vector dimensions.

To tackle the problem of computational complexity at the encoder of VQ systems, Bei and Gray [BEIg85] developed a partial distance search algorithm (PDS) which allows early termination of the distortion calculation of the input vector and codewords by incorporating a premature exit condition in the searching process. Later on, many different kinds of fast VQ encoding algorithms have been proposed. One kind of works is to improve the efficiency of PDS algorithm by adding pre-computation and extra memory storage. Paliwal and Ramasubramanian [PALIr89] reduced the computations of PDS algorithm by sorting the codebook according to the descending order of interblock transition probability of the codewords. Huang and Chen [HUANc90] utilized the triangular inequality elimination (TIE) method to eliminate the distortion calculation of several unlikely codewords to speed up the PDS algorithm. Recently, Ndifor and Ellis [NDIFe91] used the adjacent blocks to predict the current one to increase the efficiency of PDS algorithm. On the other hand, several new premature exit conditions have been developed to further increase the efficiency of PDS algorithm. Soleymani and Morgera [SOLEm89] developed an exit condition based on the absolute error associated with the element of the codewords. Later, Ra and Kim [RAk91] introduced another premature exit condition based on the sum of absolute error. All these methods maintained the same fidelity of decoded images of the standard VQ scheme. However, all these methods search every codeword in the codebook for the minimum distortion codeword of the input vector. They do not make use of the fact that image data of two adjacent blocks are correlated and so the minimum distortion codewords are probably in the vicinity of each other.

Unlike the above method, another kind of works is to reduce the computation with scarifying quality of decoded images of the standard VQ scheme, typical examples are tree-searched vector quantization (TSVQ) [GRAY84] and finite state vector quantization (FSVQ) [FOSTgd85] [DUNHg85] [ARAVg86] [KIM88]. In FSVQ scheme, an input vector first has its present state determined according to the previously encoded blocks based on a specific state function. This state function can also be updated [NASRf90a] during the coding process. After the state of the input vector is decided, the corresponding sub-codebook will be searched to find its representing vector. The sub-codebook is a sub-set of the whole codebook and is formed together with the whole codebook during the training phase. However, the minimum distortion vector of the input vector in the whole codebook may not be fallen in the sub-codebook, therefore an extra coding error will be introduced. This extra error may be compensated by the use of smaller codebooks because fewer bits will be required to represent the index of the sub-codebook, hence the overall efficiency of FSVQ scheme can be improved. In this chapter, a fast VQ encoding method called sub-codebook searching (SCS) algorithm is first proposed. The SCS algorithm, which exploits the correlation of adjacent blocks, allows searching only a portion of the codebook to find the minimum distortion codeword of the input vector. Unlike the FSVQ, the SCS algorithm generates decoded images having the same fidelity as the standard VQ scheme. The SCS algorithm is then extended to form an image coding algorithm called predictive sub-codebook searching (PSCS) algorithm, which not only requires less computational load in the encoding process but also has higher compression ability in comparison with standard VQ scheme.

In section 6.2, the sub-codebook searching (SCS) algorithm for fast VQ encoding of images will first be formulated and the predictive sub-codebook searching (PSCS) algorithm will then be described in section 6.3. The results of computer simulations using real images will be given in section 6.4. Finally, concluding remarks will be given in section 6.5.

6.2 SUB-CODEBOOK SEARCHING ALGORITHM

In vector quantization of images, data is first divided into subsequent blocks of size k=NxN. Each block is then linearly scanned to form a k-dimensional vector. Let Y = { y_i, i=1,2, ..., M } be a codebook of size M where codeword y_i = (y_{i1}, y_{i2}, ..., y_{ik}) is a k-dimensional vector. For an input vector x = (x_1, x_2, ..., x_k), it is required to find the minimum distortion codeword from the codebook Y under the Euclidean distortion measure defined as

$$d(x, y_i) = \sum_{j=1}^{k}(x_j - y_{ij})^2$$

For an image block x, its minimum distortion codeword is likely to be in the vicinity of y_c, which is the codeword of one of its adjacent blocks as shown in Figure 6.1 having the smallest distortion to x, since image data of two adjacent blocks are correlated. Therefore, those codewords in the codebook with small distance to y_c should be searched first when y_c is known. For each input vector x, we can define a control codeword y_c, a sub-codebook Y_c and its decision distortion d_c.

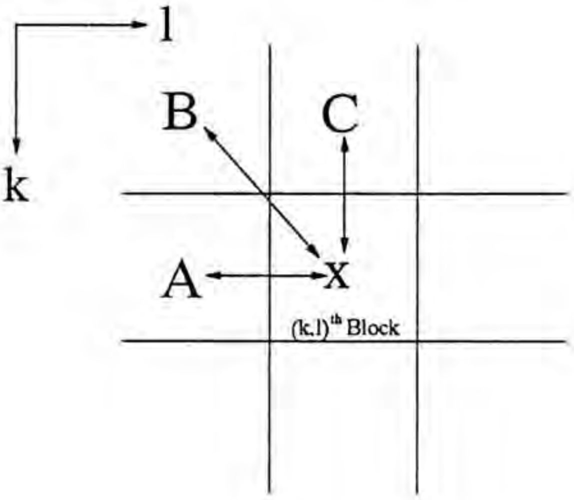


Figure 6.1 Adjacent blocks for determination of control codeword

Definition 6.1 : The control codeword of x, denoted as y_c, is the codeword of one of the three adjacent blocks of x as shown in Figure 6.1, which has the smallest distortion to x.

Definition 6.2 : The sub-codebook Y_c of a control codeword y_c is a subset of Y and elements of Y_c, denoted as y_i, are those of Y that satisfy

$$d(y_i, y_c) \leq 4\ d_c.$$

Theorem 6.1 : For any vector x, if the Euclidean distortion between x and its control codeword y_c is smaller than or equal to the decision distortion of Y_c, i.e.

$$d(x, y_c) \leq d_c$$

then the minimum distortion codeword of x must be in Y_c.

[Proof]

In Figure 6.2, the outer circle, which has radius of $2\sqrt{d_c}$, represents the sub-codebook Y_c and the origin is the location of the codeword y_c. The inner circle, which has radius of $\sqrt{d_c}$, can be regarded as the decision circle for the sub-codebook Y_c. Theorem 6.1 states that if the input vector x is on or inside the inner circle, then the minimum distortion codeword of x must be in Y_c and so searching needs only to be performed inside Y_c. Let's consider the case when x is on the inner circle as shown in Figure 6.2 and there is a codeword y_w just outside Y_c. Therefore, we have

$$d(x, y_c) = d_c \quad \text{and} \quad \sqrt{d(y_c, y_w)} > 2\sqrt{d_c} \tag{6.1}$$

According to the triangular inequality, we have

$$\sqrt{d(y_c, y_w)} < \sqrt{d(x, y_c)} + \sqrt{d(x, y_w)} \tag{6.2}$$

(6.1) and (6.2) imply

$$2\sqrt{d_c} < \sqrt{d_c} + \sqrt{d(x, y_w)}$$

$$\text{or} \quad d(x, y_c) < d(x, y_w) \tag{6.3}$$

Hence, the distance between x and y_c is always smaller than the distance between x and y_w. Obviously, $d(x,y_c)$ remains smaller than $d(x,y_w)$ if x moves towards y_c or y_w moves away from y_c. Therefore, if x is on or inside the inner circle, then no codeword outside Y_c is closer to x than y_c. In other words, the closest distance codeword of x must be inside Y_c.

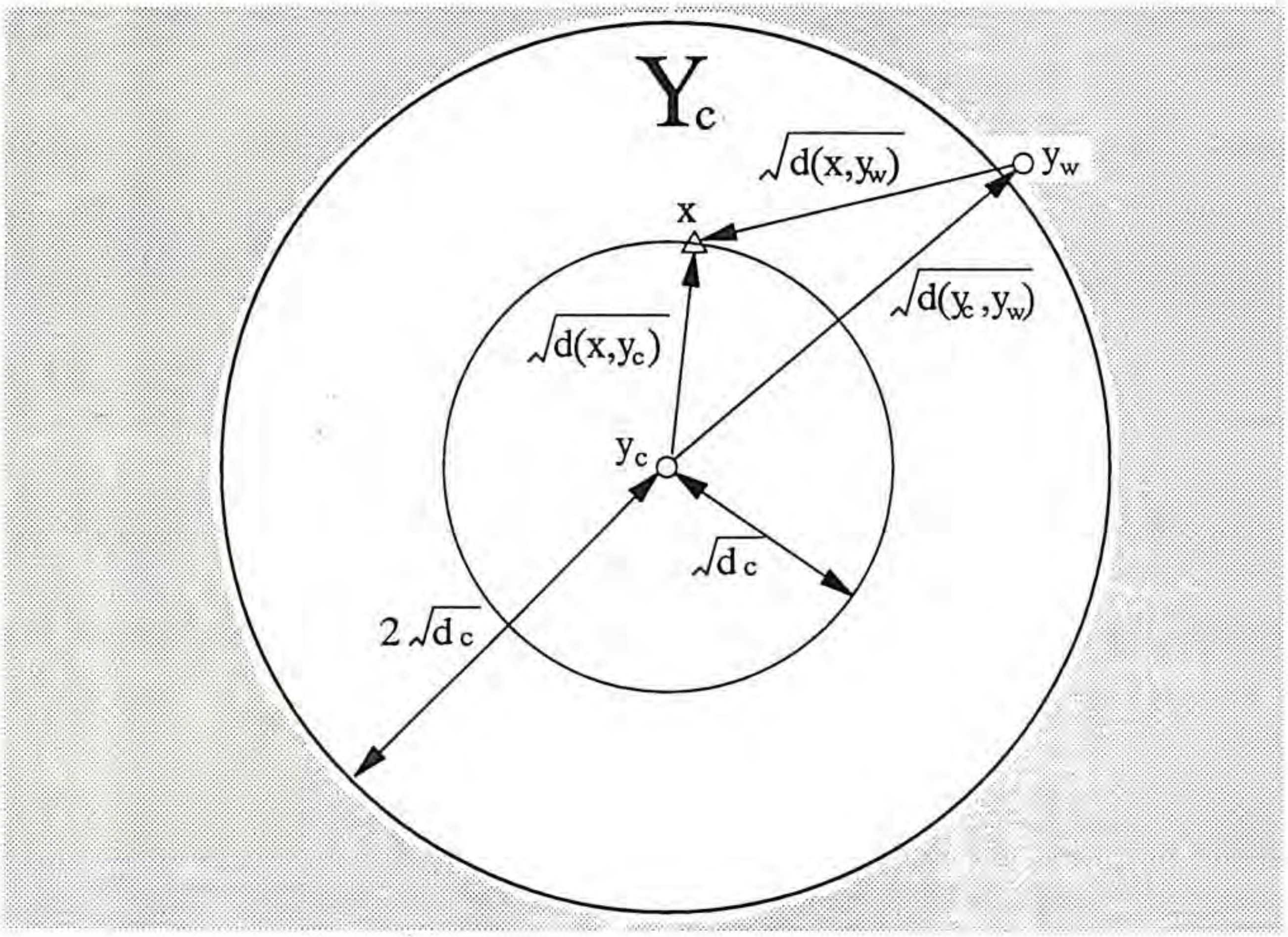


Figure 6.2 Sub-codebook Y_c in Euclidean plane

To make use of the above property to effect fast encoding, we need to construct the sub-codebook Y_c for each codeword y_c in Y in the stage of codebook generation using the training set data.

6.2.1 Formation of the sub-codebook

The sub-codebook should be formed early in the training phase. From Definitions 6.1 and 6.2, we can see that the size and the decision distortion are the two major factors in designing the sub-codebook. Two methods are proposed to construct the sub-codebook. One determines the decision distortion of each sub-codebook first while the other decides the size of the sub-codebook first.

6.2.1.1 Method I

First of all, we need to determine the decision distortion d_c for each sub-codebook Y_c. Let $P_{y_c}(y_i)$ be the probability that the best match to x is the codeword y_i when the control codeword is y_c. This probability can be determined by going through the training set sequence and calculating the frequency of occurrence of each codeword y_i when the control codeword y_c is determined. The decision distortion is then computed using the following equation :

$$d_c = \sum_{j=1}^{M} P_{y_c}(y_j)\ d(y_j, y_c) \tag{6.4}$$

Therefore, d_c can also be regarded as the average distortion of the codewords in Y to the control codeword y_c. Once the decision distortion d_c is determined, the sub-codebook Y_c can be constructed by grouping those codewords in Y having distortion to y_c smaller than or equal to $4d_c$.

6.2.1.2 Method II

From the training set of images, the sub-codebook Y_c can be found by grouping those codewords with small distortion to y_c. The number of codewords in Y_c is determined by the probability $P(y_c)$, which is the probability that y_c is the control codeword. This probability can be obtained by going through the training set sequence and calculating the frequency of occurrence of each codeword in Y. Fewer bits will be assigned to those with larger probability. Once the size of the sub-codebook is decided, the decision distortion of Y_c, d_c, will then be found by taking the maximum value of the distortion between the codewords in Y_c and the control codeword y_c.

6.2.1.3 Extra Memory Storage for the SCS Algorithm

Two tables are required at the encoder : one contains the decision distortion for each sub-codebook and the other contains the information about the mapping of the index of the sub-codebook Y_c to the index of Y. Since each codeword in the codebook Y has its corresponding sub-codebook, the number of sub-codebook is equal to the codebook size M. Let N_c be the number of bits required to store the distortion, then the total number of bits required to store the decision distortion table is equal to M N_c. Normally, N_c is taken as 16 bits. The number of bits required to store the index mapping table of method I is equal to

$$\log_2 M \sum_{c=1}^{M} Q_c$$

where Q_c is the number of codewords in the sub-codebook Y_c. The number of bits required to store the index mapping table of method II is equal to

$$\log_2 M \sum_{c=1}^{M} 2^{B_c}$$

where B_c is the number of bits assigned to the sub-codebook Y_c.

6.2.2 Premature exit conditions in the searching process

The efficiency of the search in both Y_c and Y can be improved by incorporating premature exit conditions. In the past few years, several premature exit conditions have been proposed. The first one is the well-known PDS algorithm developed by Bei and Gray in 1985 [BEIg85]. Consider an input vector x, let the smallest distortion found before checking y_i is d_{min}. If

$$\sum_{j=1}^{l}(x_j - y_{ij})^2 > d_{min} \qquad \text{for } l \in [1,2,\ldots,k] \tag{6.5}$$

then the codeword y_i will not be the best match to x and the remaining elements of the vector need not be calculated. This exit condition eliminates certain computations but the efficiency is not high since it still involves multiplications in the distortion calculation of those rejected codewords. Several exit conditions with only additions required have been proposed to further improve the efficiency of the PDS algorithm. In 1987, Soleman and Morgera [SOLEm87a] developed an exit condition based on the absolute error. For an input vector x, y_i cannot be a better match than the previous one if

$$|x_j - y_{ij}| > \sqrt{d_{min}} \qquad \text{for } j \in [1,2,\ldots,k] \tag{6.6}$$

This condition requires only one addition and one comparison for checking each element of the vector. Later, they extended this idea and proposed another exit condition based on the sum of absolute error [SOLEm87b]. For an input vector x, y_i will not be the best match to x if

$$\sum_{j=1}^{l} |x_j - y_{ij}| > \sqrt{k\, d_{min}} \qquad \text{for } l \in [1,2,\ldots,k] \tag{6.7}$$

This condition requires two additions and one comparisons for checking each element of the vector. In 1989, Soleymani and Morgera developed another premature exit condition [SOLEm89]. For each codeword y_i in the codebook Y, let $r_i = \sqrt{e_i}$ where

$$e_i = \max_{x \in S_i} d(x, y_i) \qquad \text{for } i \in [1,2,\ldots,M] \tag{6.8}$$

and S_i is the i^{th} Voronoi region in the training set. For an input vector x, if

$$|x_j - y_{ij}| > r_i \qquad \text{for } j \in [1,2,\ldots,k] \tag{6.9}$$

then y_i is not the nearest codeword to x. This condition requires one addition and one comparison for checking each element of the input vector. Recently, Ra and Kim developed another premature exit condition based on the absolute error of the mean of the vector. Consider an input vector, y_i will not be the best match to x if

$$\left| \sum_{j=1}^{k} x_j - \sum_{j=1}^{k} y_{ij} \right| > \sqrt{k\, d_{min}} \tag{6.10}$$

This condition requires one addition and one comparison for checking each element of the vector. Normally, the exit conditions given in (6.6)-(6.10) will be checked prior to the exit condition of PDS given in (6.5). If all the above exit conditions are tested, will the search be the most efficient ? The answer is obviously no. The condition given in (6.7) requires two additions for checking each element of the vector, this overhead may not be compensated by the reduction of multiplication. On the other hand, the decision level r_i of the exit condition given in (6.9) is obtained from the training set data only and may not be suitable for all kinds of images. Therefore, there is a chance for getting the wrong minimum distortion codeword of the input vector. However, the exit condition in (6.10) will be the same as the condition in (6.6) when it is performed in the transform domain. Due to the above reasons, we utilize only the exit conditions in (6.5) and (6.6) to speed up the search. Each element of an input vector will first be checked using condition (6.6), if the test is invalid then condition (6.5) will be tested.

Obviously, more computations will be saved if those exit conditions valid as early as possible. Therefore, those exit conditions can be applied in the transform domain with further benefits. Since most of the energy of the spatial data is packed into a few low frequency transform coefficients, most of the unlikely codewords can be rejected earlier than in the spatial domain by using the above exit conditions. Walsh-Hadamard transform [PRATk69] is used in our algorithm for its simplicity. Each incoming block of data first undergoes a two dimensional Walsh-Hadamard transform and the coefficients are then rearranged into a k-dimensional vector by using the zigzag scan ordering. When performing in transform domain, a question is thus arose. Is it necessary to check the exit condition given in (6.6) for all elements of the vector ? The answer is no. Since the spatial energy concentrates on a few low frequency coefficients, the exit condition in (6.6) will probably become valid in those coefficients. Experiments using several images show that this exit condition always valid in the first four low frequency coefficients. Therefore, we check this exit condition only for the first four low frequency coefficients in order to reduce the computation overhead.

6.2.3 Sub-codebook searching algorithm

To encode an input vector x, we first perform a two-dimensional Walsh-Hadamard transform to it and then determine its control codeword y_c and the distortion $d(x,y_c)$. By checking $d(x,y_c)$ according to theorem 6.1, we can determine whether the search should be performed in Y_c or Y. The minimum distortion codeword of x will then be searched by incorporating the two exit conditions given in (6.5) and (6.6). Details of the sub-codebook searching algorithm are presented in the flowchart shown in Figure 6.3 and the searching process in either the sub-codebook Y_c or the codebook Y is illustrated in the following pesudo Pascal code :

```
Procedure Searching;
Begin
    zig_zag;                                  (* perform the zigzag scanning *)
    for i:=1 to M do                          (* main search process begin *)
    begin
        d:=0;
        for j:=1 to 4 do
            if abs($x_j$-$y_{ij}$) > $\sqrt{d_{min}}$ then exit      (* first exit condition *)
        for j:=1 to k do
        begin
            d:=d+$(x_j-y_{ij})^2$;
            if d > $d_{min}$ then exit;                  (* second exit condition *)
        end;
        $d_{min}$:=d;
    end;
end;
```

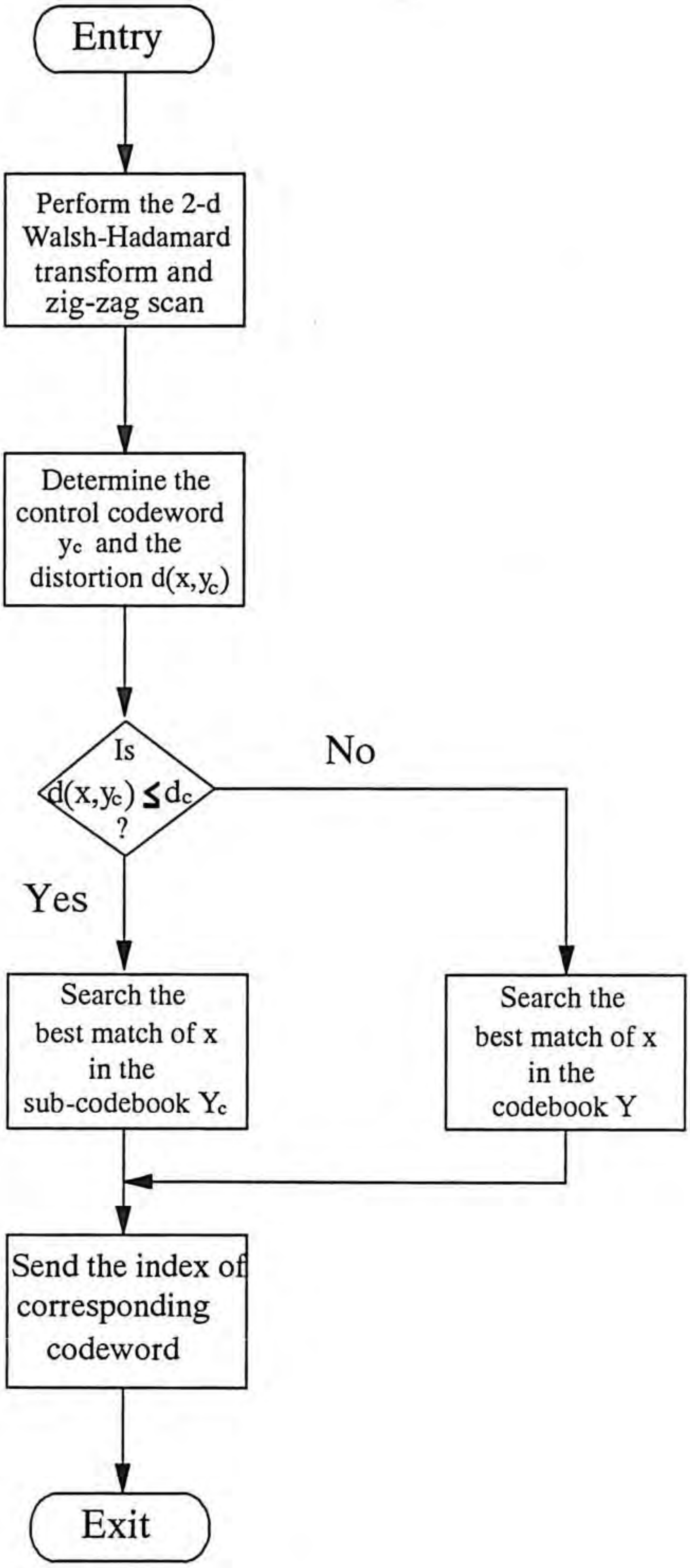

Entry
Perform the 2-d Walsh-Hadamard transform and zig-zag scan
Determine the control codeword y_c and the distortion $d(x,y_c)$
Is $d(x,y_c) \leq d_c$?
No
Yes
Search the best match of x in the sub-codebook Y_c
Search the best match of x in the codebook Y
Send the index of corresponding codeword
Exit

Figure 6.3 The sub-codebook searching algorithm at the encoder

6.3 PREDICTIVE SUB-CODEBOOK SEARCHING ALGORITHM

For VQ employing the SCS algorithm, if each codeword is transmitted using an index related to codeword Y, then the bit-rate required for the SCS algorithm is the same as the standard VQ scheme no matter whether the search occurs in Y or Y_c. In this section, we propose a predictive SCS algorithm, which allows the encoder to send a codeword using an index related to the sub-codebook Y_c. How does the decoder know which codebook is used for reconstructing the image ? To solve this problem, we use a simple predictor at both encoder and decoder to determine the control codeword of any input vector x. For an input vector x, several adjacent blocks as shown in Figure 6.4 are utilized to predict the control codeword of x.

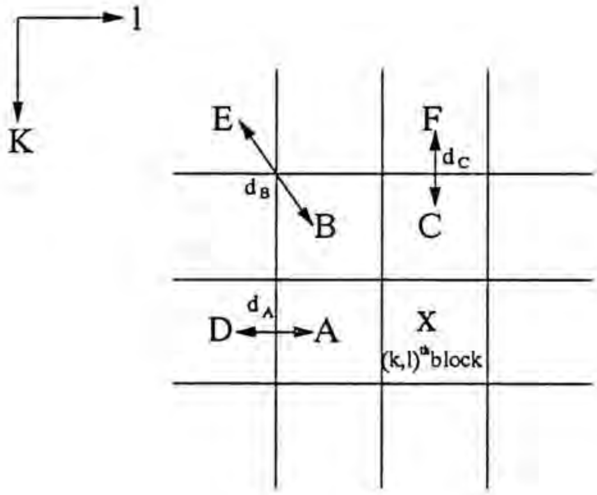


Figure 6.4 Adjacent blocks for prediction of control codeword

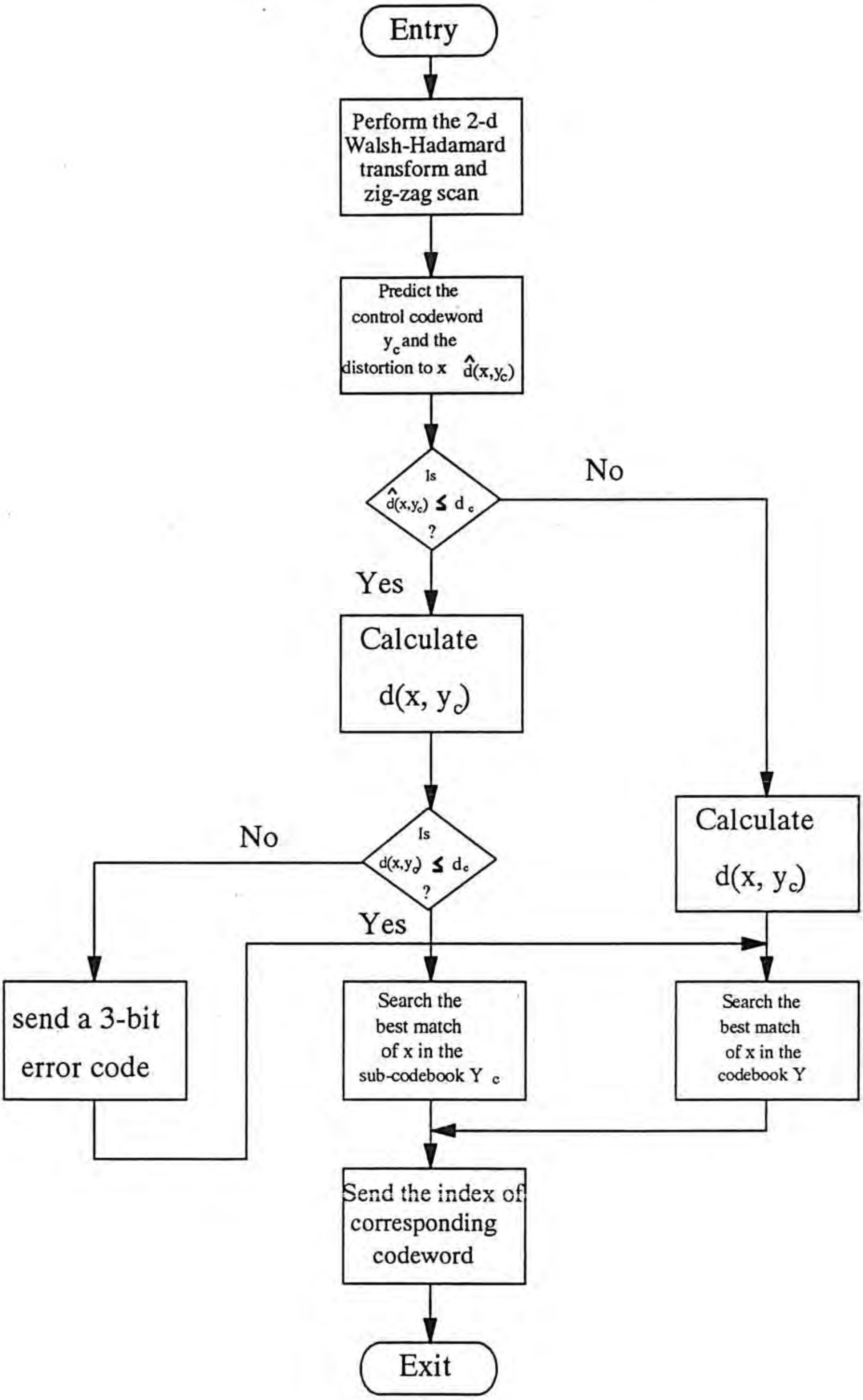
(where A,B,C,D,E and F represent the codewords of the corresponding blocks)

Distortions d_A, d_B and d_C, which are the interblock distortions as depicted in Figure 6.4, are determined simply by method of table look-up. The table contains the inter-codeword distortion of the codeword in codebook Y, $d(y_i,y_j)$ for $i,j \in [1,2, ..., k]$, are first calculated in the training phase and stored at both the encoder and decoder. To determine the control codeword of x, we choose the one with minimum corresponding distortion and the maximum value of the distortions will be acted as the prediction of the distortion between x and the control codeword y_c, $\hat{d}(x, y_c)$, which is used to determine whether the best match of x is in the sub-codebook Y_c or not. Both the encoder and decoder perform the same operations. The prediction described before will cause an error when the predicted distortion $\hat{d}(x, y_c)$ is smaller than the decision distortion y_c and the actual distortion $d(x,y_c)$ is not. When this case occurred, the minimum distortion codeword of x may not be found and would cause an extra coding error for the system. This prediction error can be detected by a simple comparison. When $\hat{d}(x, y_c)$ is smaller than d_c, compare $d(x, y_c)$ with the d_c. If $d(x, y_c)$ is larger than d_c, then an error will occur. When an error is detected, an 3-bit code will be sent to the decoder to signal an error occurred.

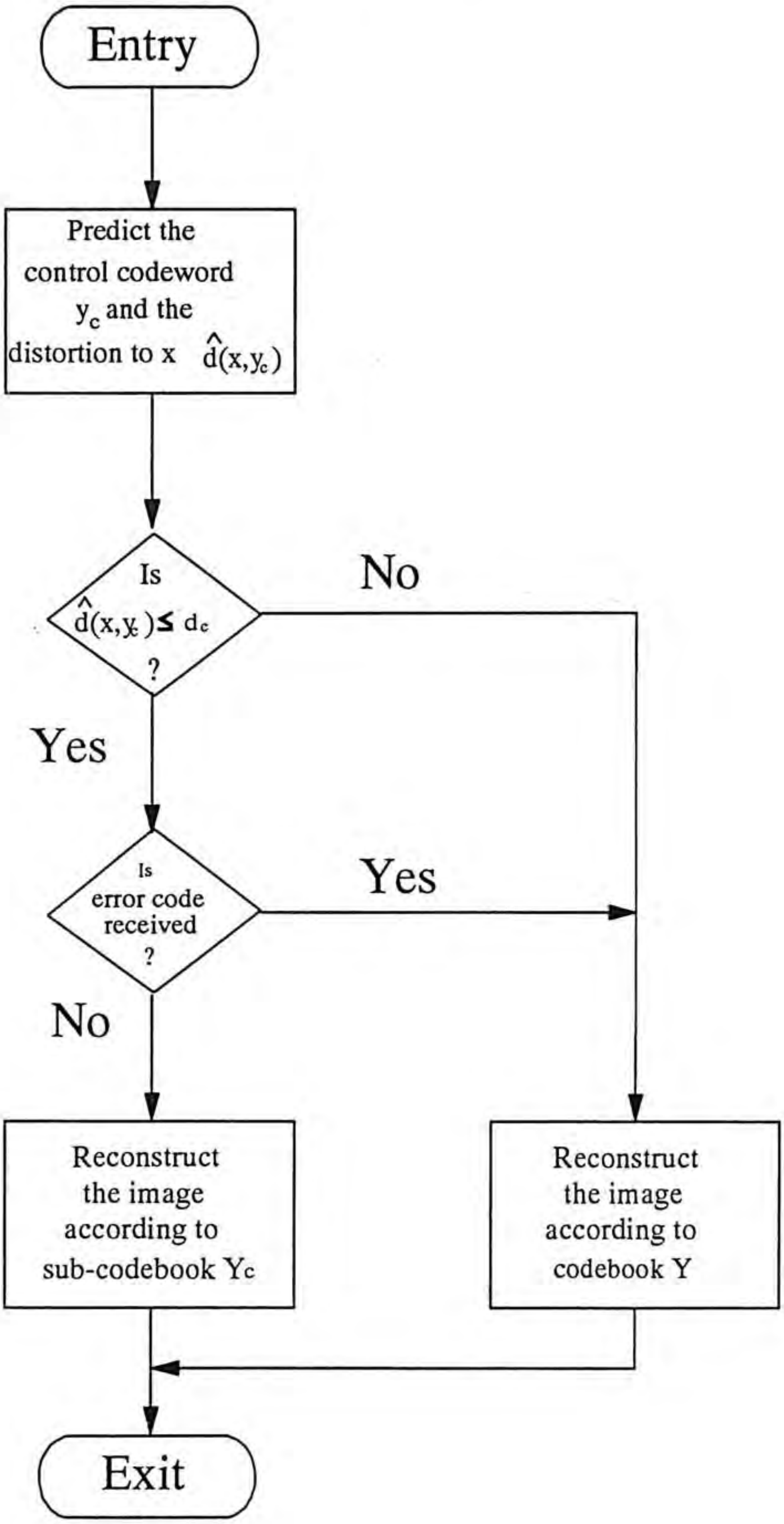
In the training phase, the sub-codebook Y_c should be constructed for each codeword y_c in Y. Method II given in section 6.2.1 is utilized to construct the sub-codebook. Since a 3-bit error code is used, some contents in the sub-codebook should be left blank. If the number of bits assigned to the sub-codebook Y_c is B_c, then the actual number of codewords of the sub-codebook Y_c is then equal to $2^{B_c} - 2^{B_c-3}$.

Predictive sub-codebook searching algorithm

To encode an input vector x, the control codeword and the corresponding distortion are first predicted by a simple predictor. If the predictor predicts y_c correctly, there is no need to send any overhead information to tell the decoder which codebook is being used because the same predictor is used in the decoder. When the predictor predicts wrongly, an 3-bit error code will be sent to the decoder. Then we can determine according to theorem 6.1 whether the best match of x is in the sub-codebook Y_c or not. If the best match of x is in Y_c, then searching is only performed in Y_c and the index of Y_c which requires less number of bits than those of Y will be sent to the decoder. When the interblock correlation is high enough, the probability of the best match of x fallen into the sub-codebook Y_c will be quite high, then a large number of bits and computations can be saved. Details of the sub-codebook searching algorithm are presented in the flowchart shown in Figure 6.5.


Entry
Perform the 2-d Walsh-Hadamard transform and zig-zag scan
Predict the control codeword y_c and the distortion to x $\hat{d}(x,y_c)$
Is $\hat{d}(x,y_c) \leq d_c$?
No
Yes
Calculate $d(x, y_c)$
Is $d(x,y_c) \leq d_c$?
No
Yes
Calculate $d(x, y_c)$
send a 3-bit error code
Search the best match of x in the sub-codebook Y_c
Search the best match of x in the codebook Y
Send the index of corresponding codeword
Exit

a. Encoder


Entry
Predict the control codeword y_c and the distortion to x $\hat{d}(x,y_c)$
Is $\hat{d}(x,y_c) \leq d_c$?
No
Yes
Is error code received ?
Yes
No
Reconstruct the image according to sub-codebook Yc
Reconstruct the image according to codebook Y
Exit

b. decoder

Figure 6.5 The predictive sub-codebook searching algorithm

6.4 SIMULATION RESULTS

Computer simulation using real images has been carried out to test the efficiency of the proposed SCS and PSCS algorithms for vector quantization of images. For the SCS algorithms, the two methods of constructing the sub-codebook given in section 6.2.1 are both simulated. The SCS algorithm using method I for generating the sub-codebook is denoted as SCS-I and the other is denoted as SCS-II. In our experiments, the codebook is generated by the well known LBG algorithm [LINDbg82] and is of size M=256 (8 bits) and the vector dimension k is equal to 16. The training set data consists of eight standard images which are of size 256x256 with 8-bits resolution per pixel and have shown in Appendix C. The test images Lenna and Sailboat are outside the training set while another test images Woman and Peppers are inside the training set. The sub-codebook size of SCS-II ranges from 16 (4 bits) to 128 (7 bits) while that of PSCS ranges from 14 (4 bits) to 112 (7 bits). We compare the proposed algorithms with several existing methods [BEIg85] [SOLEm89] [PALIr89] [HUANc90] [NDIFe91] [RSk91] in terms of the number of computations and the bit-rate requirement. The results for the test images are presented in Table 6.1 and 6.2.

Methods	Multiplications	Additions	Comparisons	Total
Full Search	256	496	15.94	767.94
[BEIg85]	43.61	71.21	43.61	158.43
[PALIr89]	31.48	46.95	31.48	109.91
[SOLEm89]	17.65	74.35	76.71	168.71
[HUANc90]	9.35	17.33	9.69	36.37
[NDIFe91]	28.53	41.01	27.53	97.07
[RAk91]	20.71	30.63	20.57	71.91
SCS-I	4.21	18.76	8.74	31.71
SCS-II	4.23	19.02	8.83	32.08
PSCS	3.16	21.67	12.77	37.60

a. Woman

Methods	Multiplications	Additions	Comparisons	Total
Full Search	256	496	15.94	767.94
[BEIg85]	52.90	89.81	52.90	195.61
[PALIr89]	44.73	73.46	44.73	162.92
[SOLEm89]	22.78	95.03	98.03	215.84
[HUANc90]	18.61	34.42	20.47	73.50
[NDIFe91]	39.43	62.82	38.43	140.68
[RAk91]	24.43	37.27	25.08	86.78
SCS-I	7.80	28.77	17.40	53.97
SCS-II	7.82	29.33	17.54	54.69
PSCS	7.04	33.82	22.42	63.28

b. Lenna

Methods	Multiplications	Additions	Comparisons	Total
Full Search	256	496	15.94	767.94
[BEIg85]	50.56	85.12	50.56	186.24
[PALIr89]	45.16	74.32	45.16	164.64
[SOLEm89]	19.92	86.20	88.82	194.94
[HUANc90]	17.98	33.02	19.99	70.99
[NDIFe91]	38.43	62.82	38.43	139.68
[RAk91]	23.57	35.65	24.12	83.34
SCS-I	7.03	27.64	16.61	51.28
SCS-II	7.05	28.09	16.71	51.85
PSCS	6.64	33.53	22.49	62.66

c. Peppers

Methods	Multiplications	Additions	Comparisons	Total
Full Search	256	496	15.94	767.94
[BEIg85]	56.78	97.56	56.78	211.12
[PALIr89]	59.02	102.03	59.02	220.27
[SOLEm89]	24.07	101.41	104.62	230.10
[HUANc90]	29.93	55.28	33.64	118.85
[NDIFe91]	48.55	81.08	47.55	177.18
[RAk91]	27.23	42.23	28.52	97.98
SCS-I	11.22	37.53	25.44	74.19
SCS-II	11.22	38.03	25.25	74.50
PSCS	10.83	42.75	28.94	82.52

d. Sailboat

Table 6.1 Computational complexity of different VQ encoding methods for the test images

	Woman	Lenna	Peppers	Sailboat
Total no. of blocks	4096	4096	4096	4096
No. of blocks searched in sub-codebook	2196	1418	1287	1422
No. of error blocks	142	241	214	186
Total no. of bits required	26423	30559	31058	31681
Bit-rates (bits per pixel bpp)	0.403	0.466	0.474	0.483

Table 6.2 Bit-rate requirement for the predictive sub-codebook searching algorithm

Table 6.1 reveals that the SCS and PSCS algorithm require the least number of multiplications for all test images when compared with other existing encoding algorithms. They perform very close but PSCS is slighly better. For example, when coding the image Lenna, the numbers of multiplications required by SCS-I, SCS-II, PSCS and full search algorithm are 7.8, 7.82, 7.04 and 256 respectively. This show that the PSCS algorithm needs only 2.75% of that required by the full search algorithm. When considering the total number of operations, the PSCS algorithm requires a little bit more than the SCS algorithms but less than other methods. Take the image Sailboat as an example, the total operations required by SCS-I, SCS-II, PSCS and full search algorithm are 74.19, 74.50, 82.52 and 767.94 respectively. It is noted that the SCS-I and SCS-II algorithm requires only 9.66% and 9.7% of that required by the full-search method. When compared SCS-I and SCS-II, Table 6.1 shows that the SCS-I and SCS-II algorithm have similar results for all test images and SCS-I is slightly better. On the other hand, the bit-rate required by all the methods except the PSCS algorithm is fixed and equal to $\log_2 M/k=0.5$ bpp. When considering the bit-rate requirement as shown in Table 6.2, the bit-rates required by the PSCS algorithm for the image Woman, Lenna, Peppers and Sailboat are 0.403, 0.466, 0.474 and 0.483 respectively, which is equal to about 4 to 20% bit-rate reduction. The amount of bit-rate reduction is almost independent of whether the image is inside the training set or not. Instead it depends mainly on the activity of the image itself. When the correlation between pixels of the image is small, or in other words the image is of high activity, the number of blocks encoded by the sub-codebook Y_c will become less and then the bit-rate reduction will be small.

6.5 CONCLUDING REMARKS

In this chapter, a new encoding method called sub-codebook searching (SCS) algorithm is first proposed for vector quantization of images. This algorithm is characterized that it allows only searching part of the codebook to find the best match codeword of the input vector by using two tables at the encoder. Based on the SCS algorithm, a predictive sub-codebook (PSCS) algorithm is formulated. The PSCS algorithm not only reduces the computation load but also improves the compression ratio of standard VQ scheme. Computer simulations using real images show that the proposed algorithms are very efficient since they require the least number of multiplications as well as the number of total operations in comparison with other existing VQ encoding algorithms. However, extra computations and memory storage are required in both the training stage and the encoder/decoder for the SCS and PSCS algorithm. In the training stage, extra operations are required to form the sub-codebook for both algorithms. Furthermore, the PSCS algorithm requires several extra operations to predict the control codeword of the input vector at both encoder and decoder. Extra memory storage are needed to store the decision distortion table and the sub-codebook mapping table at the encoder for both algorithms and at the decoder for the PSCS algorithm only. Beside, the inter-codeword distortion table are required to store at both encoder and decoder for the PSCS algorithm to predict the control codeword of the input vector.

In vector quantization of images, computational complexity and compression ratio are the two important factors that affect the efficiency of a coding system. In this chapter, we have proposed schemes to improve the efficiency of the standard VQ scheme by reducing the computational complexity and increasing compression ratio while keeping the fidelity of the decoded images unchanged. In the next chapter, we shall describe a new coding algorithm, which has low computational complexity and high compression ratio, for vector quantization of images.

6.6 NOTE ON PUBLICATIONS

A paper entitled 'An efficient encoding algorithm for vector quantization of images' has been accepted for presentation at the Sixth European Signal processing Conference, Belgium, August 1992. Another paper entitled 'An efficient VQ encoding algorithm for image coding' has also been accepted for presentation at the Second International Conference on Image Processing, Singapore, September 1992. Also, a paper entitled 'Sub-codebook searching algorithm for efficient VQ encoding of images' has been accepted for publication in Electronics Letters. On the other hand, a paper called 'A new fast VQ encoding algorithm for image compression' was submitted for publication at IS&T/SPIE Symposium on Electronic Imaging : Science and Technology, California USA, Jan. 1993. All these papers were jointly authored with Dr. W.K. Cham.

CHAPTER 7 PREDICTIVE CLASSIFIED ADDRESS VECTOR QUANTIZATION OF IMAGES

In this chapter, a new coding scheme called predictive classified address vector quantization (PCAVQ) is proposed for image compression. In this scheme, a new two-stage classification method based on the three-level block truncation coding technique is developed to classify blocks of an image into different classes so as to improve the edge quality of decoded images. In the proposed system, the predictive mean removal VQ technique [NASRf90c] is applied to reduce the blocking effect of decoded images and a new simplified address VQ technique is also developed to reduce the bit-rate requirement.

7.1 INTRODUCTION

In vector quantization of images, computational effectiveness and compression ratio are the two main criteria in evaluating the performance of a coding system. A good vector quantization system should have low computation complexity and high compression ratio. Normally, they cannot be achieved at the same time. In the last section, we have described two efficient VQ encoding algorithms which improve both the computational effectiveness and compression ratio while maintaining the same fidelity of decoded images of standard VQ scheme. When considering the fidelity of decoded images, the main problem of standard VQ scheme is the visual degradation of edges. The poor edge reproduction ability is mainly due to the use of an universal codebook for all blocks. Since an image contains different kinds of blocks, such as edges, shade, or mixture of both, a single codebook can not represent each type of blocks well. However, edges are very important portion of the perceptual information of an image, degradation of edges is very visually annoying. To solve this problem, it is necessary to encode different kinds of blocks using different codebooks such that each kind of sub-sources can well be represented in its corresponding codebook. Ramamurthi and Gersho were the first to tackle this problem by developing a classified VQ scheme [RAMAg86]. In their classified VQ scheme, an image is first divided into different sub-sources with similar local statistics and each sub-source is encoded by a particular codebook. The edge classification of this scheme is based on the gradient computation in the spatial domain. The edge detection scheme is quite complicated and is difficult to decide threshold values for different kinds of images. Following the work of Ramamurthi and Gersho, different types of classified VQ schemes haven been proposed. Davignon developed a simple classification scheme based on binary vector quantization [DAVI90]. The main drawback of this algorithm is the occurence of the 'staircase effect' in the decoded image especially in the areas with smooth edges. Recently, Kim and Lee proposed an efficient edge oriented classifier in the DCT domain [KIMl89] [KIMl91] [KIMl92]. Meanwhile, Ngan et al. has also developed a classified hybrid image coder [NGANk89] [NGANkw91], which is also based on the classification of images in DCT domain, for image sequence coding. However, implementation of a two-dimensional DCT requires a lot of extra computations. On the other hand, Po and Chan [POc91] proposed a directionally edge classifier in Walsh-Hadamard domain in order to reduce the computation overhead in the classification process. In this chapter, a simple edge oriented classifier using the three-level block truncation coding (BTC) [DELPm79] [EFRAlm91] is proposed for efficient classification of edges so as to improve the visual qualtiy of the edges of decoded images.

On the other hand, while considering the compression ratio of a coding system, the address VQ technique recently proposed by Nasrabadi and Feng [FENGn88] [FENGn89] [NASRf90a] [NASRf90b] [FENGn91] [DIXITf91] was a very efficient method to reduce the bit-rate requirement of the standard VQ scheme. This method is based on exploiting the interblock correlation by encoding a group of blocks together using an address codebook. The address codebook consists of a set of address codevectors where each codevector represents a combination of addresses of group of blocks. Each element of this codevector is actually an address of an entry in the LBG codebook. However, their method involves a lot of computations to build up the interblock transition probability matrices such that to dynamically update the content of the active region of the address codebook for each block of data. Such complexity makes it impractical. In our work, we propose a simple method to build up the address codebook in the training phase and develop a fast searching algorithm for large address codebook. In our proposed algorithm, there is no need to update the address codebook during encoding. The new address VQ technique is efficient since it requires only a few extra comparisons.

In section 7.2, the optimum three-level block truncation coding will be reviewed and then the predictive classified address vector quantization (PCAVQ) will be formulated in section 7.3. The results of computer simulations using real images will be discussed in section 7.4. Finally, concluding remarks will be given in section 7.5.

7.2 OPTIMAL THREE-LEVEL BLOCK TRUNCATION CODING

Consider a three-level quantizer described by the output levels {a,b,c} and two thresholds $\{t_1,t_2\}$ such that the output of the quantizer is equal to a if $x_i < t1$, b if $t_1 \leq x_i < t_2$ and c if $x_i \geq t_2$, the expected distortion D using the square error criterion can be denoted as

$$D(\ \{a,b,c\},\{t_1,t_2\}) = \sum_{x_i < t_1} (x_i - a)^2 + \sum_{t_1 \leq x_i < t_2} (x_i - b)^2 + \sum_{x_i \geq t_2} (x_i - c)^2 \tag{7.1}$$

The optimum thresholds t_1'and t_2' for {a,b,c} is one that minimizes the distortion given in (7.1) and can be easily found to be

$$t'_1 = \frac{(a+b)}{2} \qquad t'_2 = \frac{(b+c)}{2} \tag{7.2}$$

Hence, for any thresholds t_1 and t_2,

$$D(\ \{a,b,c\},\{t_1,t_2\}) \geq D(\ \{a,b,c\},\left\{\frac{(a+b)}{2},\frac{(b+c)}{2}\right\}) \tag{7.3}$$

On the other hand, when the thresholds of quantizer are t_1 and t_2, the optimum output quantization levels a', b' and c' can then be obtained by minimizing D({a,b,c}, $\{t_1,t_2\}$) with fixed t_1 and t_2. The results are

$$a' = \frac{1}{u} \sum_{x_i < t_1} x_i \tag{7.4.a}$$

$$b' = \frac{1}{k\text{-}u\text{-}v} \sum_{t_1 \leq x_i < t_2} x_i \tag{7.4.b}$$

$$c' = \frac{1}{v} \sum_{x_i \geq t_2} x_i \tag{7.4.c}$$

where u and v are the number of pixels smaller than the lower threshold t_1 and larger or equal to the upper threshold t_2 respectively. This implies that

$$D\left(\{a,b,c\},\{t_1,t_2\}\right) \geq D\left(\{a',b',c'\},\{t_1,t_2\}\right) \tag{7.5}$$

According to the conditions in (7.3) and (7.5), we can design a three-level quantizer by iteratively optimizing the thresholds t_1 and t_2 for the output level {a,b,c} and vice versa. The algorithm is illustrusted in the flowchart as shown in Figure 7.1. The minimum value x_{min}, the mean value x_{mean} and the maximum value x_{max} of the block are used as initial values of the output levels. The inital thresholds are simply equal to (a+b)/2 and (b+c)/2. With these initial thresholds, we can calculate the new values of the output levels according to (7.4). The new values of a, b and c are then compared with the old one to see whether they have stopped converging. If not, a, b and c will be used to compute two new threshold values t_1 and t_2, and the whole process is repeated iteratively until minimum mean-square error is achieved. After encoding, two kinds of information are necessary to be sent to the decoder. The first one is the values of the three output levels a, b and c, and the other is the three-level bit-map information TL(i) for each pixel :

$$TL(i) = \begin{cases} -1 & \text{if} \quad x_i < t_1 \\ 0 & \text{if} \quad t_1 \leq x_i < t_2 \\ +1 & \text{if} \quad x_i \geq t_2 \end{cases} \tag{7.6}$$

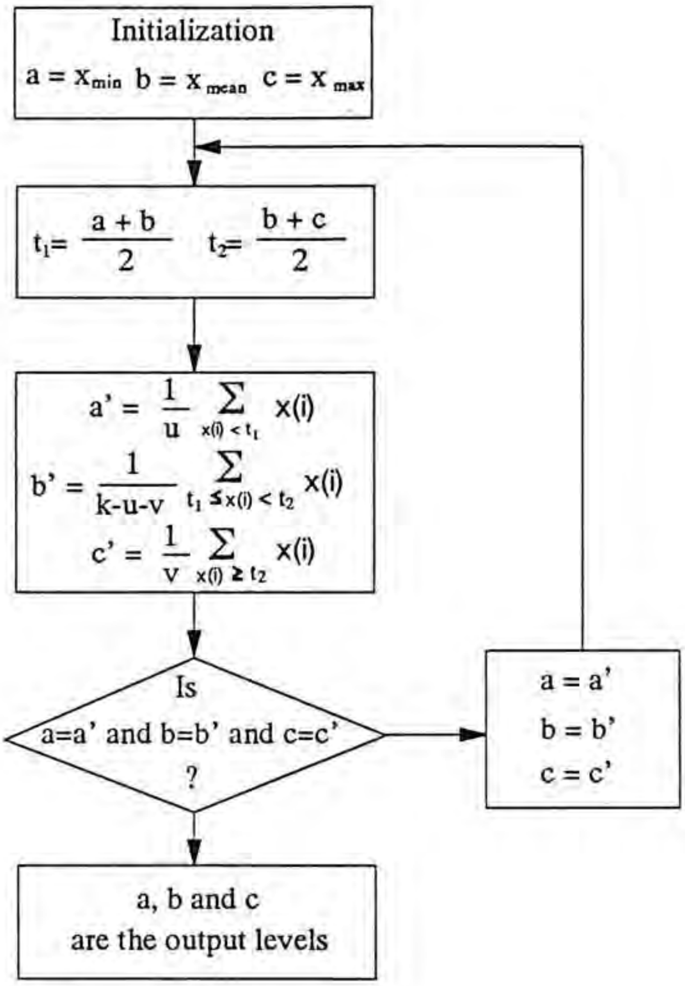

Initialization
$a = x_{min}$ $b = x_{mean}$ $c = x_{max}$
$t_1 = \frac{a+b}{2}$ $t_2 = \frac{b+c}{2}$
$a' = \frac{1}{u} \sum_{x(i) < t_1} x(i)$
$b' = \frac{1}{k-u-v} \sum_{t_1 \leq x(i) < t_2} x(i)$
$c' = \frac{1}{v} \sum_{x(i) \geq t_2} x(i)$
Is a=a' and b=b' and c=c' ?
a = a'
b = b'
c = c'
a, b and c are the output levels

Figure 7.1 Algorithm for designing an optimum MSE three-level quantizer

7.3 PREDICTIVE CLASSIFIED ADDRESS VECTOR QUANTIZATION (PCAVQ)

The objective of the proposed predictive classified address vector quantization (PCAVQ) scheme is to develop an accurate algorithm to divide an image into different kinds of blocks having similar characteristics such that each kind of blocks can well be represented in the corresponding codebook. The block diagram of the PCAVQ scheme is shown in Figure 7.2. Basically, the proposed PCAVQ scheme can be divided into three main parts : classification of images, predictive mean-removal and address VQ of non-edge classes.

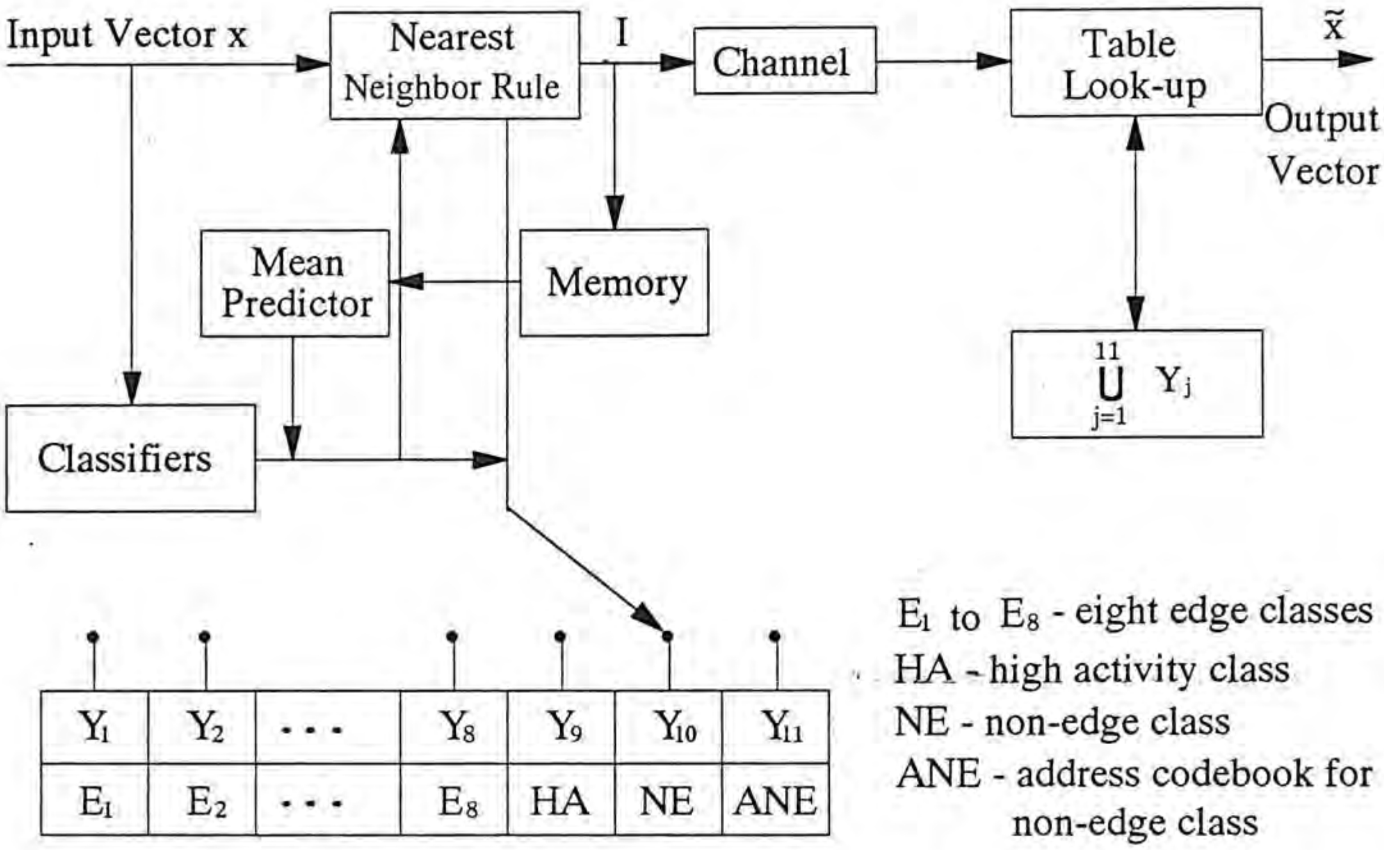


Figure 7.2 Block diagram of PCAVQ scheme

7.3.1 Classification of Image using Three-level BTC

Classification of image is performed in the spatial domain and it is divided into two stages. Classifier I first determine whether the input block of image is a non-edge block, an edge block or a high activity block. Then the edge block will further be classified into one of eight edge classes using classifier II.

7.3.1.1 Classifier I

Classifier I is used to classify an input block x into either a non-edge block, an edge block or a high activity block by calculating the standard deviation σ of x as follows

$$\text{x is} \begin{cases} \text{non-edge block} & \text{if} \quad \sigma < \text{Thd}_1 \\ \text{edge block} & \text{if} \quad \text{Thd}_1 \le \sigma < \text{Thd}_2 \\ \text{high activity block} & \text{if} \quad \sigma \ge \text{Thd}_2 \end{cases} \tag{7.7}$$

where Thd_1 is the lower threshold and Thd_2 is the upper threshold. The reason for using the upper threshold Thd_2 to further classify the image into edge blocks and high activity blocks is to enable a better adaptation to those high activity blocks. When the standard deviation of an image block is larger than Thd_2, there is likely more than one edge in the block. Therefore, this arrangement can improve the edge reproduction ability in the codebooks and make them robust to different kinds of images. Extra bits are required to send to the decoder to indicate which class the block is. As most blocks are non-edge blocks, so one bit is used to represent non-edge class and two bits are used used to represent edge and high activity classes.

7.3.1.2 Classifier II

In the second stage, edge blocks are subdivided into different types of edge classes by using Classifer II. For an edge block x, it is first passed through an optimum three-level BTC coder and a three-level bit-map as given in (7.6), which contains the values of +1, 0 and -1, will be generated. The Euclidean distance between the bit-map of x and the eight prescribed edge classes as shown in Figure 7.3 is then calculated. x is classified as the class which has the minimum distance to x. Different edge classes will then be vector quantized by using different codebooks. Three extra bits are required for each edge class to indicate which codebook is used.

+1	+1	+1	+1
0	0	0	0
0	0	0	0
-1	-1	-1	-1

Class 1

Horizontal

+1	0	0	-1
+1	0	0	-1
+1	0	0	-1
+1	0	0	-1

Class 3

Vertical

0	0	-1	-1
0	0	0	-1
+1	0	0	0
+1	+1	0	0

Class 5

Diagonal (135°)

+1	+1	0	0
+1	0	0	0
0	0	0	-1
0	0	-1	-1

Class 7

Diagonal (45°)

-1	-1	-1	-1
0	0	0	0
0	0	0	0
+1	+1	+1	+1

Class 2

-1	0	0	+1
-1	0	0	+1
-1	0	0	+1
-1	0	0	+1

Class 4

0	0	+1	+1
0	0	0	+1
-1	0	0	0
-1	-1	0	0

Class 6

-1	-1	0	0
-1	0	0	0
0	0	0	+1
0	0	+1	+1

Class 8

Figure 7.3 The eight edge classes

7.3.2 Predictive Mean Removal Technique

Apart from the edge degradation in the decoded images, blocking effect is another problem in standard VQ scheme. Blocking effect is more visually annoying in smooth areas of an image. In order to reduce the blocking effect, Baker and Gray [BAKEg82] proposed a technique called mean removal vector quantization (MRVQ). In their system, the sample mean of a block is scalar quantized and the residue block obtained by substracting the sample mean with each element of the block is vector quantized. Two kind of information should be sent to the decoder : scalar quantized mean and the vector quantized residue block. This technique was extended by Nasrabadi and Feng [NASRf90b] in their predictive mean removal VQ (PMRVQ) algorithm. In PMRVQ, the sample mean is predicted at both encoder and decoder, therefore, there is no need to send the mean information. The predicted value of the sample mean of the block is computed by using the previously encoded vertical and horizontal pixels bordering the current block as shown in Figure 7.4. The predicted mean of the block is given by

$$\hat{x}_{mean} = \frac{1}{9}\sum_{i=1}^{9} a_i \qquad (7.8)$$

where the a_i's are the encoded vertical and horizontal pixels bordering the current block. In our proposed PCAVQ system, we apply the predictive mean-removal technique to the non-edge blocks in order to reduce the blocking effect in decoded images.

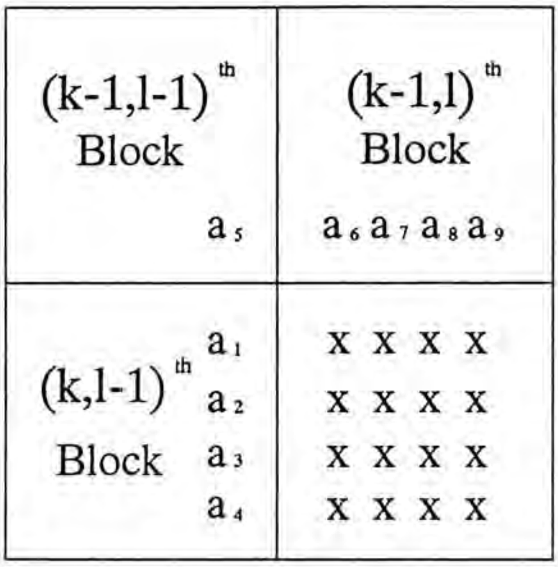


Figure 7.4 Pixels used to predict the mean of the block x

7.3.3 Simplified Address VQ Technique

The address VQ (AVQ) technique is an efficient way to increase the compression ratio of the standard VQ system by exploiting the inter-block correlation of the image. The inter-block redundancy is removed by grouping the addresses resulted from vector quantization of four-neighbor blocks when using a standard VQ codebook (LBG codebook). An address codebook consists of a set of codevectors. The element of each codevector, which represents the address of one of the codevectors in the LBG codebook, is used to transmit a codeword representing the block combination rather than vector quantizing each individual block by the LBG codebook. Since the number of codevectors in the address codebook is less than the total number of block combinations, fewer bits is required for AVQ than that of standard VQ scheme. Therefore, further reduction of bit-rate requirement can be achieved. The address codebook is illustrated in Figure 7.5.

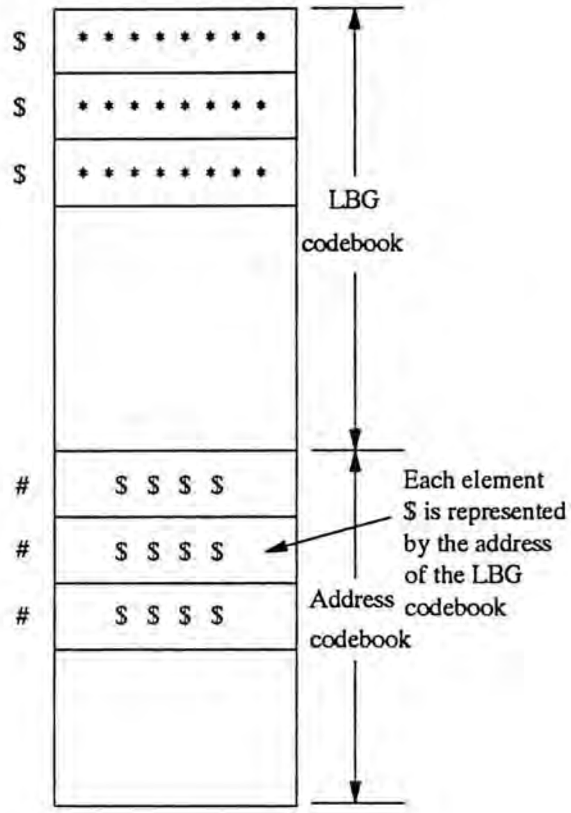


Figure 7.5 The LBG codebook and the address codebook

In their proposed AVQ scheme, the address codebook is obtained according to the training images. First of all, all blocks in the training set are vector quantized by obtaining the address of the corresponding codevector in the LBG codebook. Then all of the possible address combinations of four neighboring blocks occurring in the training set are then extracted to form the address codebook. Since an image is of high correlation in most cases, the size of an address codebook will be much smaller than the total possible block combinations. However, the address codebook is still very large so it is divided into two regions : active region and non-active region. Only the active region of the address codebook is used by both encoder and decoder. Since different images have different characteristics, the active region of the address codebook should be updated all the times to cope with the change of local statistics of images. The block transition probability matrices, which contain the conditional probability (frequency) of a codevector occurring given one of its three neighboring blocks, are used to update the active region of the address codebook when coding a block. Such process will be performed in both encoder and decoder. Therefore, a lot of computations will be required and this will make the system impractical. In this section, we propose a simplified address VQ technique which does not need to update the address codebook.

In the original AVQ scheme, each group of four neighbor blocks are required to be searched to find a match in the address codebook. However, only those highly correlated blocks will probably be in the address codebook. In our simplified AVQ scheme, each group of four neighboring blocks will first be checked by a boolean condition CN given in (7.9). If the condition is satisfied, then searching in the address codebook will be continued, otherwise searching will be skipped.

$$\mathrm{CN} = \begin{cases} \text{true} & \text{if } \sigma_{\max} \leq t_a \\ \text{false} & \text{otherwise} \end{cases} \tag{7.9}$$

where $\sigma_{\max}$ is the maximum standard deviation of the four encoded neighboring blocks and t_a is a threshold value obtained in the stage of formation of the address codebook. Large value of $\sigma_{\max}$ implies the four adjacent blocks are less correlated than those with smaller $\sigma_{\max}$. With such an arrangement, a large number of operations will be saved because most of the low correlated blocks will probably not be in the entries of the address codebook.

7.3.3.1 Formation of Address Codebook

In this method, the address codebook is formed according to the images in the training set data. Details of the procedure are shown below :

Step 1 All blocks in the training set are vector quantized by using the LBG codebook;

Step 2 All the possible address combinations of four neighboring blocks occurring in the training set data are extracted;

Step 3 Calculate the value of σ_{max} for each extracting combination;

Step 4 Sort the extracting combinations with the ascending order of the value of σ_{max};

Step 5 Select the first N_{AC} combinations in the sorted combinations, where the N_{AC} is the prescribed size of the address codebook;

Step 6 Take the maximum value of σ_{max} in the address codebook as the threshold value t_a.

7.3.3.2 Fast Address Codebook Searching Algorithm

In the proposed searching algorithm, the addresses of four adjacent blocks to be encoded are first arranged according to the order shown in Figure 7.6.

1	2
3	4

Figure 7.6 Searching order in address codebook

Each codevector in the address codebook is also arranged in such order. Those codevectors having the same value for the first element will first be grouped together. Then the address codebook is sorted according to the ascending order of the value of the first element. A table containing the grouping information is necessary at both encoder and decoder. Each entry of the table should have the following format

Starting Address	Length

Each entry contains two fields : the starting address and the length of the codevectors having the same value of the first element of the codevector in the address codebook. For an input address vector, we first look up the above table to see the starting address of those codevectors having the same value of the first element. Search is then performed over the following codevectors and the number of codevectors to be searched is prescribed by the Length field of the table. With this arrangement, only'part of the address codebook to be searched for the best match of the input address vector, therefore, a significant saving of operations will be resulted. When the total number of codevectors in address codebook is equal to N_{AC} and the LBG codebook is of size M, the total number of bits required to store the table information in either encoder or decoder is then equal to 2M $\log_2 M_{AC}$. On the other hand, it is also necessary to send one bit to the decoder to indicate whether the four blocks is in the address codebook or not.

7.3.4 Encoding Process of PCAVQ

The flow diagram of the encoding process of PCAVQ scheme is shown in Figure 7.7. First of all, an input vector is passed through classifier I to determine whether it is a non-edge vector, an edge vector or a high activity vector. For a high activity vector, it is then vector quantized using its corresponding codebook. For an edge vector, it is further classified into one of eight prescribed edge classes by passing through classifier II. Each edge class is then vector quantized by different codebooks. On the other hand, a non-edge vector is first passed through a predictive mean removal VQ coder and the residue block is then address vector quantized. The index of different codebooks and the extra information to indicate which codebook is used are then sent to the decoder.

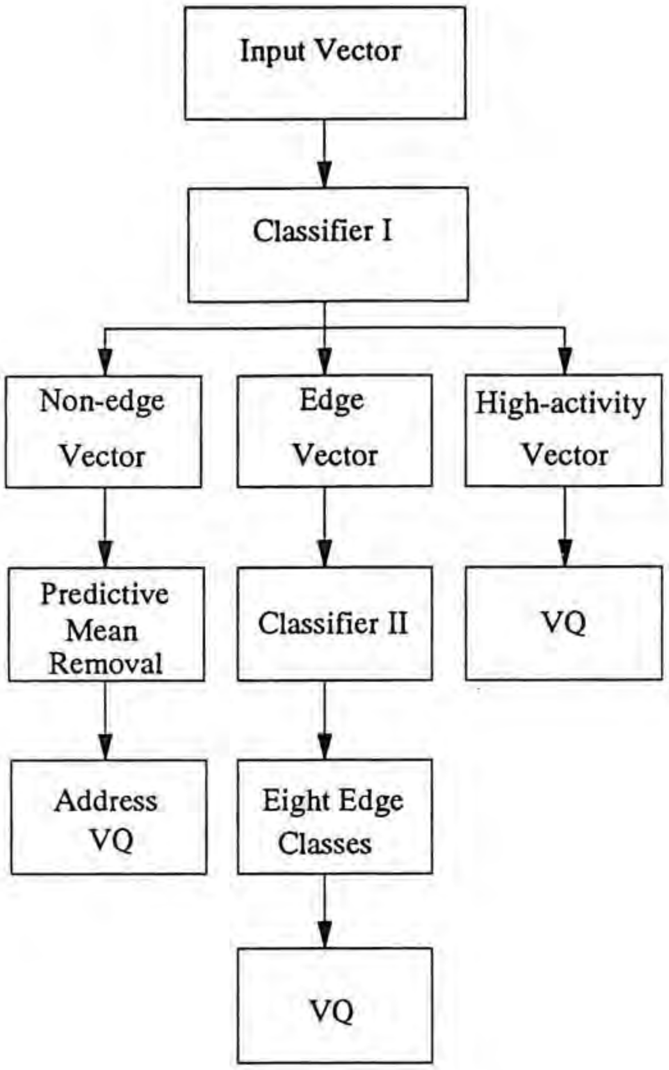


Figure 7.7 Flow diagram of the encoding process of PCAVQ

7.4 SIMULATION RESULTS

Computer simulations using real images has been carried out to evaluate the performance of the proposed PCAVQ scheme. In our experiments, the two threshold values of classifier I, Thd_1 and Thd_2, are set to 18 and 28 respectively. The size of the LBG codebook is 256 and those for the non-edge block, edge block and high activity block are equal to 64, 16 and 16 respectively. Full search method is used for searching in each codebook. The threshold value t_a of the simplified address VQ scheme is equal to 11.7 and the size of the address codebook is 4096. The training set data consists of eight standard images which are of size 256x256 with 8-bit resolution per pixel and have shown in Appendix D. The test images Lenna and Sailboat are both outside the training set while another test images Woman and

Peppers are inside the training set. The MSE performance and the bit-rate requirement of the test images for different systems are presented in Table 7.1. The computational requirements of PCAVQ and standard VQ scheme are also listed in Table 7.2. On the other hand, the decoded images for PCAVQ scheme, JPEG baseline system and standard VQ scheme are shown in Figures 7.8-14.

Picture	MSE	Bit-rate (bpp)
Woman	37.22	0.5
Lenna	133.61	0.5
Peppers	118.30	0.5
Sailboat	271.79	0.5

a. Standard VQ scheme

Picture	MSE	Bit-rate (bpp)
Woman	11.61	0.5
Lenna	73.58	0.5
Peppers	74.77	0.5
Sailboat	190.89	0.5

b. JPEG baseline system

Picture	MSE	N_{NE}	N_E	N_{HA}	N_{AVQ}	Bit-rate (bpp)
Woman	23.27	3743	210	143	3296	0.297
Lenna	58.62	3275	403	418	632	0.428
Peppers	56.01	3250	323	523	2388	0.344
Sailboat	79.36	2605	572	919	328	0.434

b. PCAVQ scheme

Table 7.1 Simulation results for the test images

(where N_{NE}, N_E, N_{HA} and N_{AVQ} are the number of non-edge vectors, edge vectors, high activity vector and address VQ coded vectors respectively)

Picture	NM_{class}	NM_{AVQ}	NM_{VQ}	NM_{PCAVQ}	$NM_{standard\ VQ}$
Woman	1.59	0	59.86	61.45	256
Lenna	2.03	0	54.38	56.41	256
Peppers	1.84	0	54.09	55.93	256
Sailboat	2.41	0	46.53	48.94	256

a. Number of multiplications

Picture	NA_{class}	NA_{AVQ}	NA_{VQ}	NA_{PCAVQ}	$NA_{standard\ VQ}$
Woman	2.02	0	115.99	118.01	496
Lenna	2.92	0	105.40	108.32	496
Peppers	2.55	0	104.79	107.34	496
Sailboat	3.72	0	90.15	93.87	496

b. Number of additions

Picture	NC_{class}	NC_{AVQ}	NC_{VQ}	NC_{PCAVQ}	$NC_{standard\ VQ}$
Woman	0.33	0.70	3.68	4.71	15.94
Lenna	0.65	1.26	3.34	5.25	15.94
Peppers	0.51	0.70	3.32	4.53	15.94
Sailboat	0.94	0.82	2.85	4.61	15.94

c. Number of comparisons

Table 7.2 Computational requirements per pixel for the test images (where the subscripts class, AVQ, VQ, PCAVQ and standard VQ represent the corresponding operations in the classification process, address VQ process, VQ process, the whole PCAVQ algorithm and the standard VQ scheme respectively)

As shown in Table 7.1, the proposed PCAVQ achieves a very good MSE performance and has the minimum values for the images Lenna, Peppers and Sailboat in comparison with other systems. Consider the image Lenna, the values of MSE for PCAVQ, JPEG baseline system and standard VQ scheme are 58.67, 73.58 and 133.61 respectively. For the image Woman, JPEG baseline system results a lower MSE value but requires a higher bit-rate than the PCAVQ system. When considering the bit-rate requirement, the proposed method is also less than those of JPEG baseline system and standard VQ scheme. Take the image Lenna as an example, the bit-rate required for PCAVQ scheme is 0.428 bpp while 0.5 bpp is needed for other two systems. This implies that PCAVQ scheme has about 15% reduction in bit-rate requirement. Simulation results in Table 7.1 also show that there are 40% bit-rate reduction for Woman and Peppers and 15% bit-rate reduction for Lenna and Sailboat over the other two systems when using the proposed PCAVQ scheme. It implies that the amount of bit-rate reduction is dependent on whether the test image is inside the training set or not. The reason is that most of non-edge blocks will be coded by the address codebook if the test image is inside the training set. Consider the image Woman, there is 3296 non-edge blocks which is equal to 88% of total non-edge blocks coded by the address codebook.

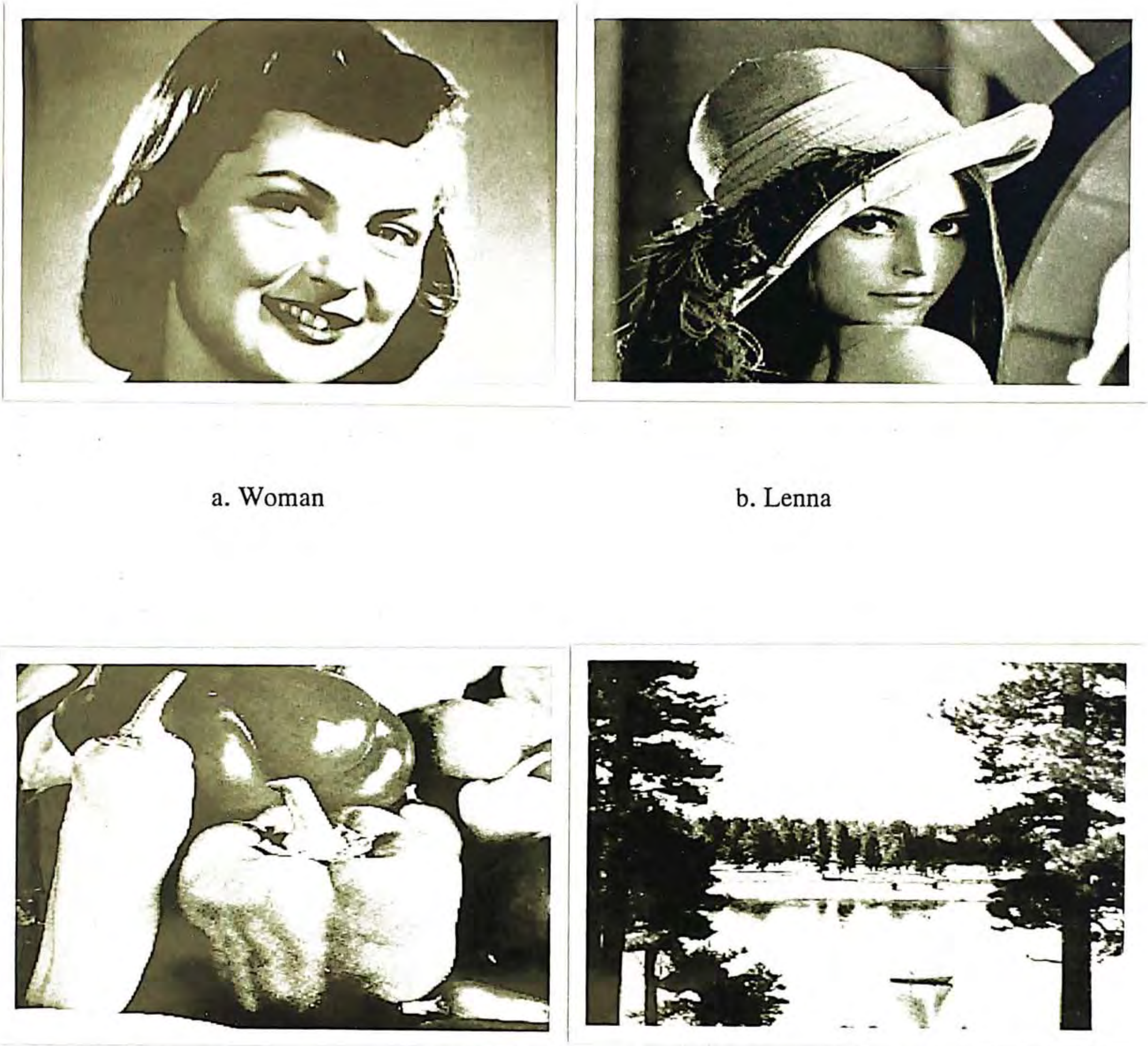
Considering the subjective performance of decoded images of the proposed system. Figures 7.8-14 show that decoded images of PCAVQ and JPEG baseline system are better than those of standard VQ scheme for all test images in all aspects such as edge quality and blocking effect. In comparison with JPEG baseline system, the proposed PCAVQ scheme results better edge quality in the decoded images and also has less blocking effect especially in the smooth areas of the image. However, JPEG baseline system has better visual performance in the high activity areas of the image, for example, the eyes of the image Lenna. As shown in Table 7.2, the computational requirement of the PCAVQ scheme is also less than those of the standard VQ scheme. For examples, when coding the image Lenna, the PCAVQ requires 56.41 multiplications, 108.32 additions and 5.25 comparisons which is equal to about 22.13 % of the total operations required by those of standard VQ scheme. These figures can also be reduced when incorporating fast encoding algorithms described in the last chapter. It is also noted that the extra computations required by the classification process and the simplified address VQ technique are limited. For the image Lenna, only 2.03 multiplication, 2.92 addition and 0.65 comparison operations are required for the classification process. On the other hand, only 1.26 extra comparions are needed when performing the proposed simplified address VQ technique. Therefore, the proposed PACVQ scheme not only improves both the subjective and objective performance of the decoded images but also reduces the computational complexity and the bit-rate requirements of standard VQ scheme.

7.5 CONCLUDING REMARKS

A new vector quantization system called predictive classified address vector quantization (PCAVQ) is proposed for image compression in this chapter. In PCAVQ algorithm, the techniques of classified VQ, predictive mean-removal VQ and address VQ have been utilized. A new two-stage image classification scheme is proposed and the part of edge classification is based on an optimum three-level block truncation coding technique. A simplified address VQ technique is also developed to reduce the bit-rate requirement of the system. Simulation results that very good visual quality with no detectable edge degradation is obtained in the decoded images of the PCAVQ scheme. It has also been shown that an excellent MSE performance is obtained for the PCAVQ scheme at bit-rates around 0.4 bpp. The compututional complexity of the overall system of PCAVQ is also less than that of standard VQ scheme. Therefore, the proposed PCAVQ algorithm is very efficient since it not only improve both the subjective and objective fidelity of the decoded images but also reduces the computational complexity and bit-rate requirement of standard VQ scheme.

7.6 NOTE ON PUBLICATIONS

A paper entitled 'Classified vector quantization of images using three-level block truncation coding,' was accepted for presentation at the Third International Symposium on Signal Processing and Its Applications, Australia, August 1992. This paper was co-authored with Dr. W.K. Cham.



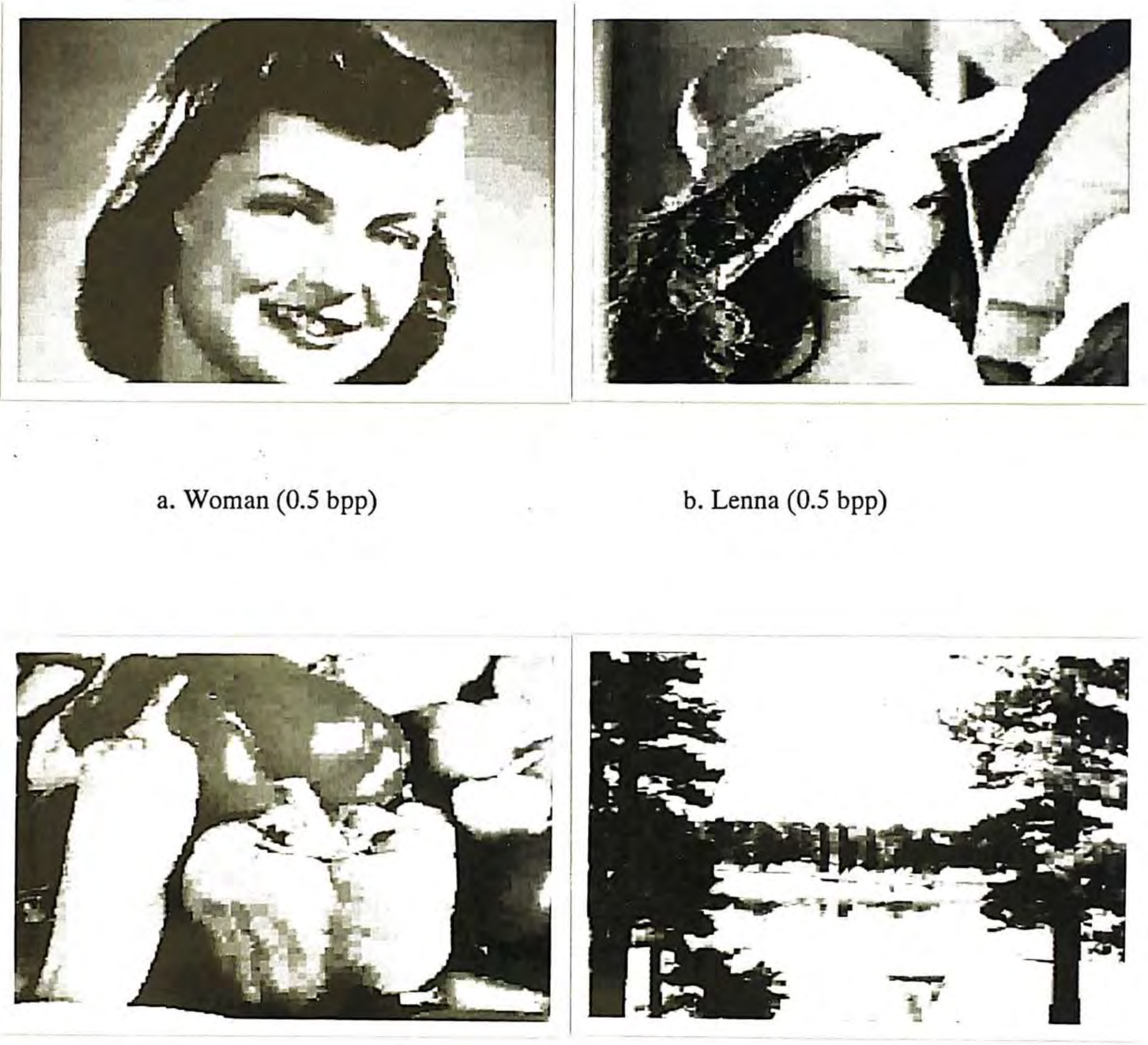
a. Woman

b. Lenna

c. Peppers

d. Sailboat

Figure 7.8 Original test images



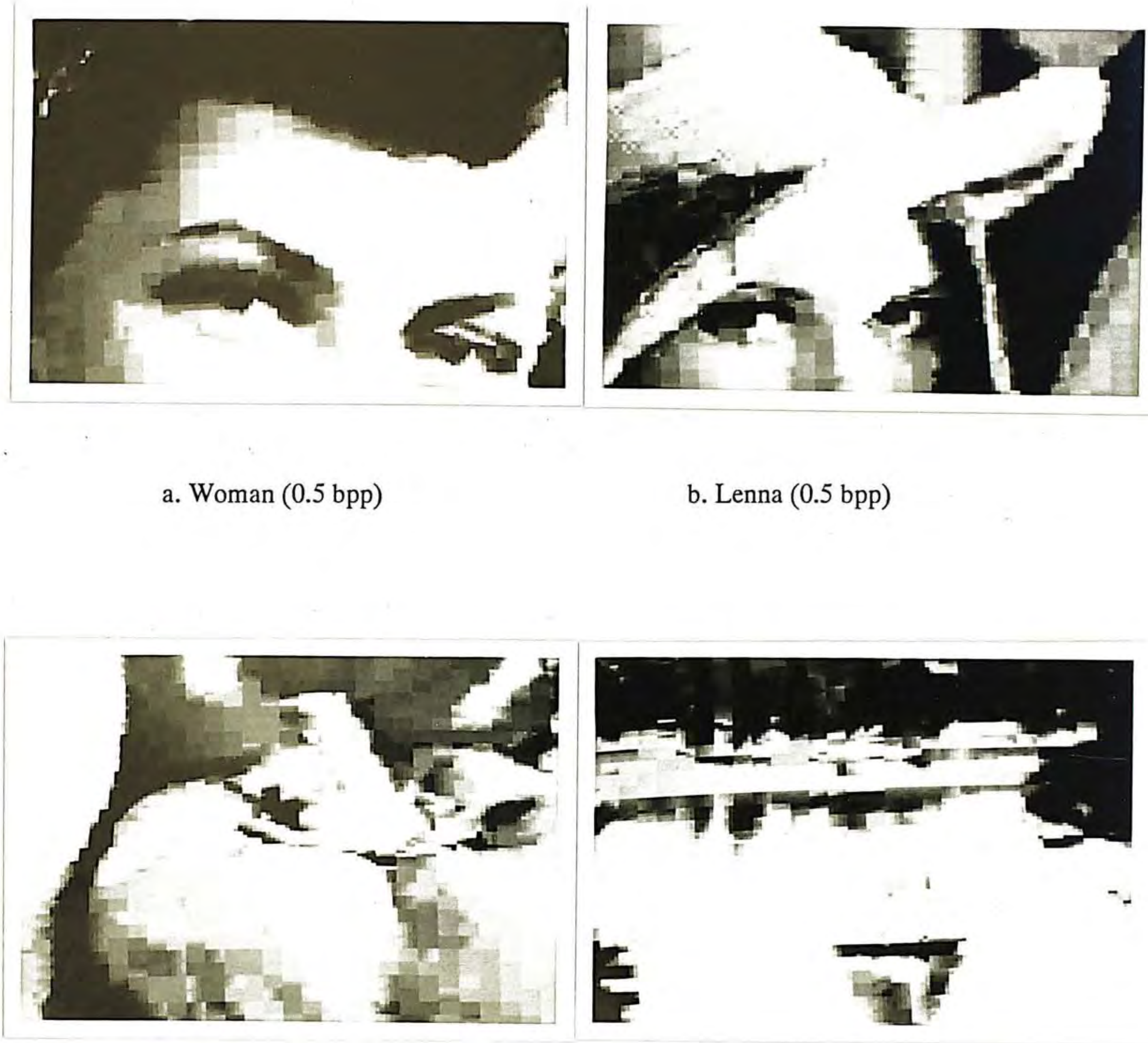
a. Woman (0.5 bpp)

b. Lenna (0.5 bpp)

c. Peppers (0.5 bpp)

d. Sailboat (0.5 bpp)

Figure 7.9 Decoded images of standard VQ scheme



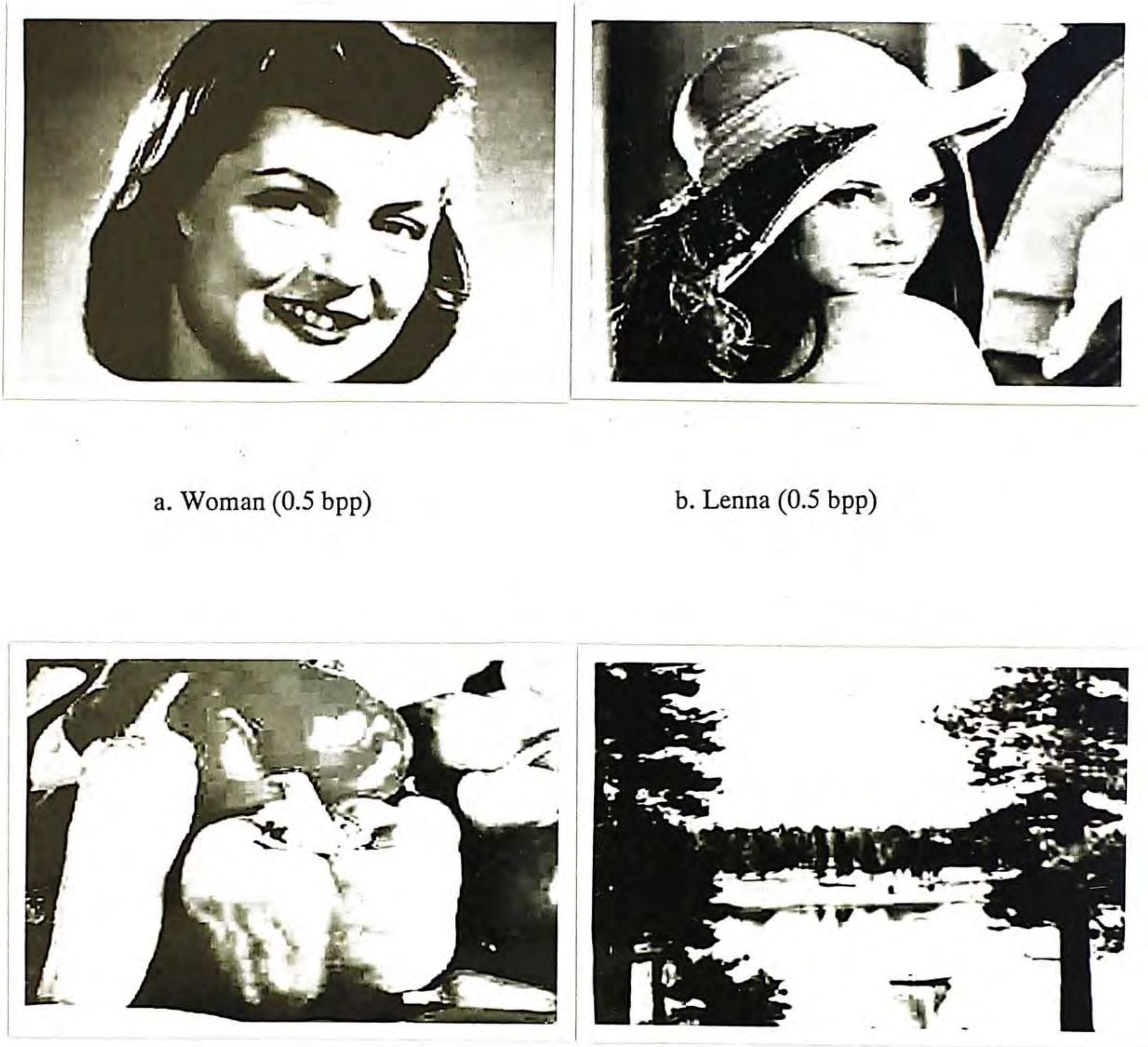
a. Woman (0.5 bpp)

b. Lenna (0.5 bpp)

c. Peppers (0.5 bpp)

d. Sailboat (0.5 bpp)

Figure 7.10 Enlarged decoded images of standard VQ scheme



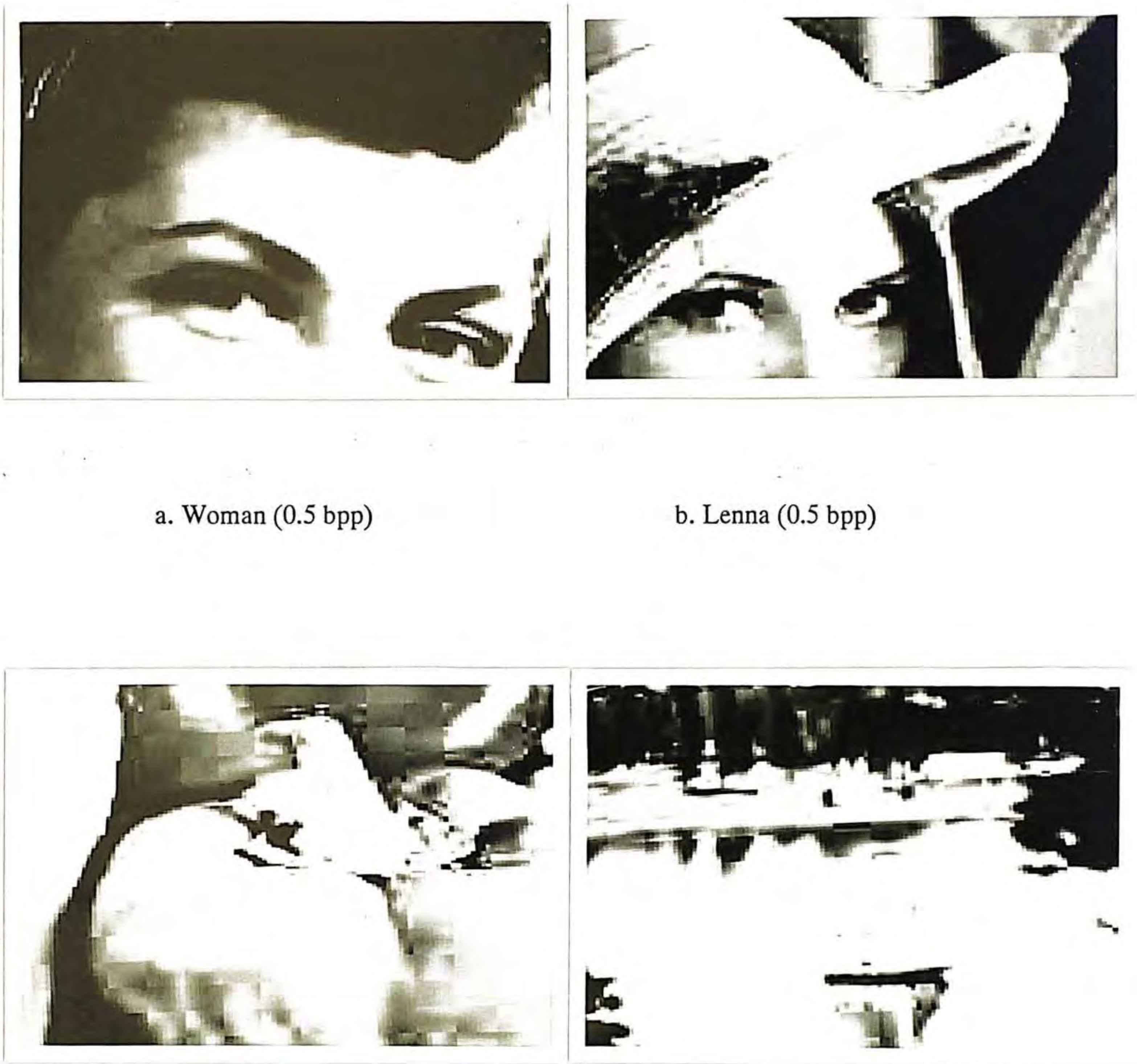
a. Woman (0.5 bpp)

b. Lenna (0.5 bpp)

c. Peppers (0.5 bpp)

d. Sailboat (0.5 bpp)

Figure 7.11 Decoded images of JPEG baseline system



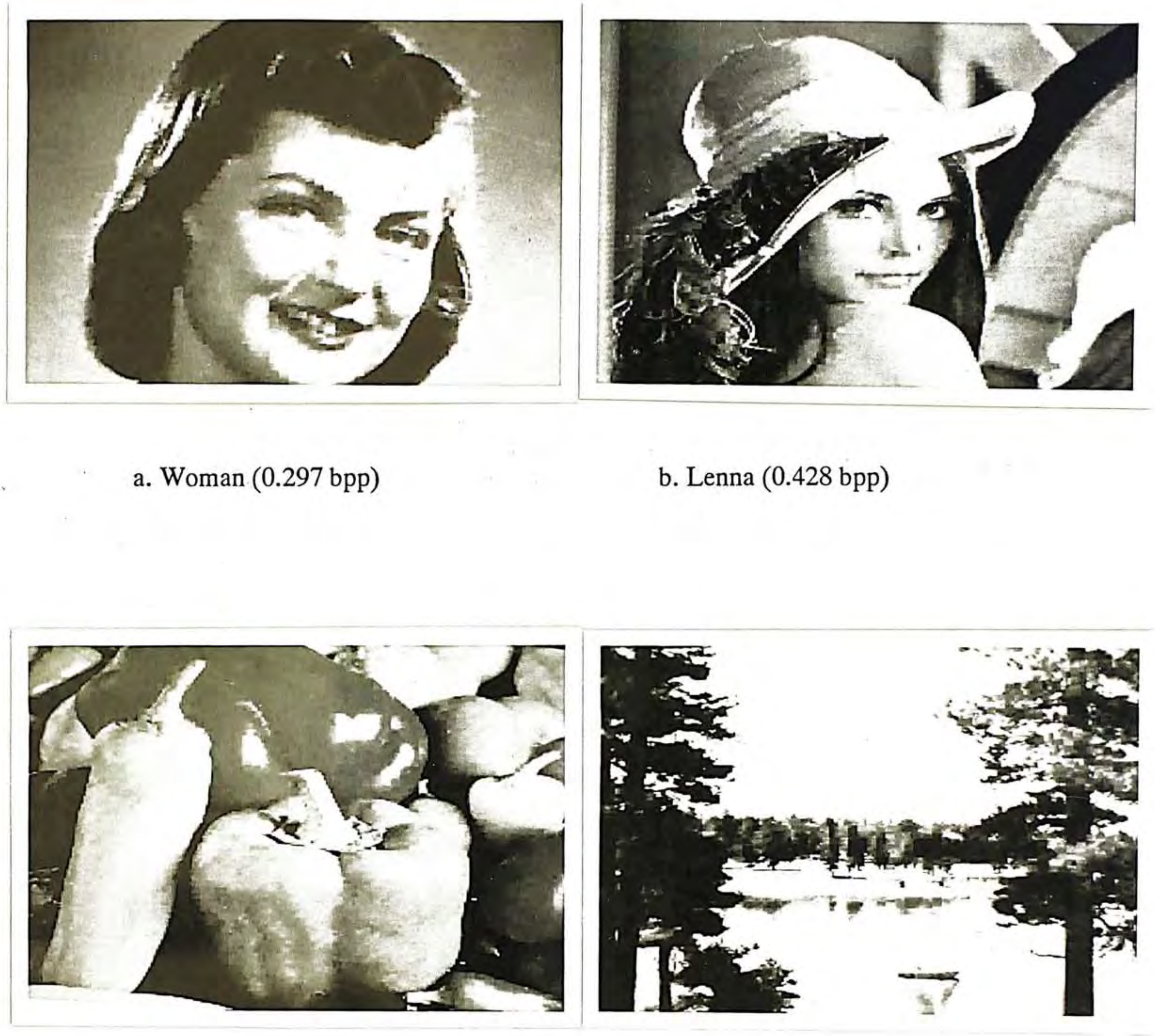
a. Woman (0.5 bpp)

b. Lenna (0.5 bpp)

c. Peppers (0.5 bpp)

d. Sailboat (0.5 bpp)

Figure 7.12 Enlarged decoded images of JPEG baseline system



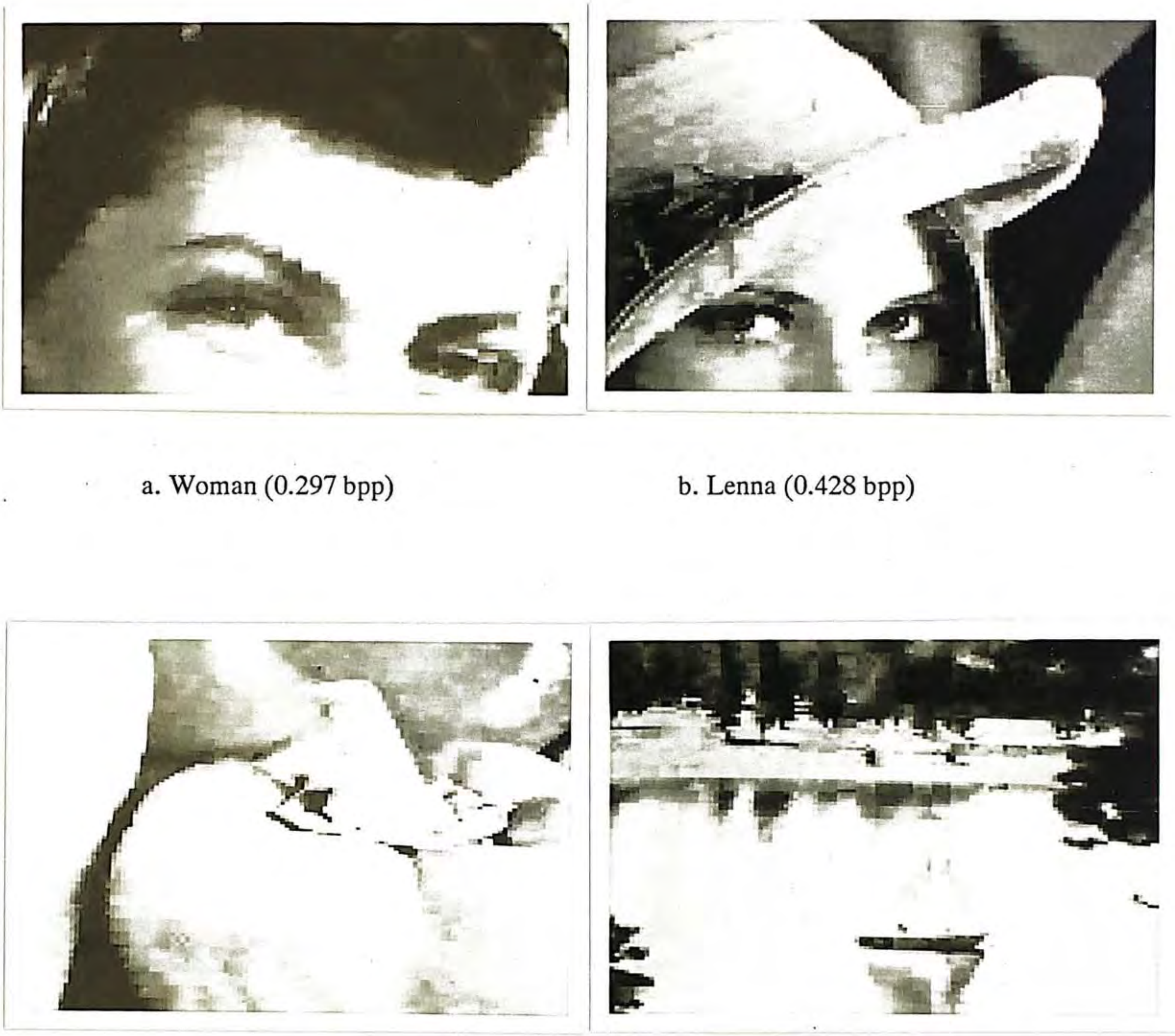
a. Woman (0.297 bpp)

b. Lenna (0.428 bpp)

c. Peppers (0.344 bpp)

d. Sailboat (0.434 bpp)

Figure 7.13 Decoded images of PCAVQ scheme



a. Woman (0.297 bpp)

b. Lenna (0.428 bpp)

c. Peppers (0.344 bpp)

d. Sailboat (0.434 bpp)

Figure 7.14 Enlarged decoded images of PCAVQ scheme

CHAPTER 8 RECAPITULATION AND TOPICS FOR FURTHER INVESTIGATION

8.1 RECAPITULATION

As stated in chapter 1, our goal of work is to build up a high efficiency block coding technique for image compression, that is a method having low computational complexity and high compression ability. This thesis essentially contains several contributions to the development of a high efficiency block-based image coding system. Our works were concentrated on the mapping and the quantization process of such a system.

In chapter 3, we developed two new orthogonal transforms, namely weighted cosine transform (WCT) and simplified cosine transform (SCT), for transform image coding. The two transforms were derived based on two different approaches. WCT was designed in order to improve the performance of the DCT and maintain the similar complexity of the DCT. WCT was obtained by weighting the transform matrix of the DCT with simple structured orthonormal matrices. Different tests showed that WCT performs better than the DCT and other transforms being tested for orders 8 and 16. On the other hand, SCT was developed for the purpose of reducing the computation requirement of the DCT while maintaining the similar output performance of the DCT. SCT was generated by replacing the even part of the DCT matrix with a simple structured orthogonal matrix which can be computed using only addition and binary shift operations. SCT performed better than HCT and LCT, and has similar performance as those of the DCT, ICT and MDCT. When the adjacent element correlation coefficient is equal to 0.95, the order-8 SCT is even better than the ICT. The works of this chapter provide an alternative for people to select an orthogonal transform in a image transform coding system. When the performance is the most important factor, we may use WCT as substitute for the DCT. Whilst the cost of computation is of important, we may use the order-8 SCT to replace the DCT and ICT.

We derived in chapter 4 the pruned fast computational algorithms for the DCT, WCT and SCT. Detail analysis of such algorithms had been shown that a significant saving of operations can be obtained when the number of coefficients required to be computed is smaller than the number of input pixels. This case is often occurred in a transform coding system since most of the spatial energy is concentrated on a few number of low frequency coefficients after transformation. Therefore, the computation effectiveness of the overall system can be increased when incorporating pruning in the transformation process.

In chapter 5, we proposed a simple predictor called minimum edge difference (MED) predictor for efficient encoding of the DC coefficients in a transform coding system. The MED predictor predicted DC coefficients by minimizing the edge difference between the current and adjacent blocks of data. Analytical method was utilized to evaluate the performance of the proposed predictor. It was shown that when the adjacent element correlation coefficient of the pixel element is larger than 0.65, the MED predictor resulted the smallest values of the prediction error variance among several other predictors and performed very close to its optimum case. Computer simulations using real images showed that there is about 5% bit-rate reduction when the MED predictor is adopted in the JPEG baseline system. Extra 2N additions and one binary shift were required for the MED predictor.

We developed two efficient encoding algorithms for vector quantization of images in chapter 6. The sub-codebook searching (SCS) was first developed and then extended to form another coding algorithm called predictive sub-codebook searching (PSCS) algorithm. The SCS algorithm allowed searching only a portion of the codebook to find the minimum distortion codeword of the input vector and transmitting the index of the codeword of the whole codebook. Since searching was performed in a rather smaller space, a significant saving of computation was resulted. On the other hand, the PSCS algorithm not only allowed to search the sub-codebooks but also to transmit the index of the codewords in the sub-codebook. When considering the computation complexity, the two algorithms are very efficient since both algorithms required the least number of multiplications as well as the total operations when compared with other existing fast VQ encoding algorithm by using computer simulations on real images. On the other hand, simulation results also showed that PSCS has about 4-20 percent of bit-rate reduction over standard VQ scheme.

Following chapter 6, we proposed a new vector quantization system called predictive classified address vector quantization (PCAVQ). With the PCAVQ scheme, blocks of an image were first classified into non-edge blocks, edge blocks and high-activity blocks. The high activity block was then vector quantized. For the edge block, it was further divided into one of eight prescribed edge classes by using an optimum three-level block truncation coding technique. Each edge class was then vector quantized by its corresponding codebook. On the other hand, the non-edge block was passed through a predictive mean removal VQ coder. The residue block was then encoded by a proposed simplified address VQ technique in order to reduce the bit-rate requirement. Experiment results showed that the PCAVQ not only improves the visual quality and objective MSE performance of decoded images, but also reduces the computational complexity and bit-rate requirement of standard VQ scheme.

8.2 TOPICS FOR FURTHER INVESTIGATION

In this section, we highlight some possible research works related to the work in this thesis for further investigation.

A. Generation of New Transforms

1. Based on the philosophy of designing the WCT, look for other transforms having better performance than the DCT and WCT.

2. According to the technique of developing the odd part of the SCT, generate a new transform which contains only simple orthogonal matrices in both even and odd part of the transform matrix. The new transform is expected to have compression ability lie between the DCT and those of HCT and LCT.

3. Implementation of the pruned fast algorithms for the DCT. WCT and SCT. Since the structures of those algorithms are quite regular, they are suitable for VLSI implementation using existing processors or dedicated hardware.

B. Efficient Encoding of DC Coefficients

Apply the proposed MED predictor to the two other block-based image coding systems - vector quantization and block truncation coding. However, we cannot apply the MED predictor directly since the MED predictor requires the information of the encoded AC coefficients of the current blocks which is not available in the two systems. To solve this problem, we can modify the structure of the MED predictor by considering only the pixels in the previously encoded blocks instead of the current one. A further improvement of the efficiency of those systems will be produced.

C. Vector Quantization

1. Find other efficient methods to predict the control codeword and its corresponding distortion for the PSCS algorithm to further improve the bit-rate reduction ability of the system.

2 Develop a new simple dynamic address VQ technique to reduce the bit-rate requirement more efficient than the proposed one when maintaining its simplicity.

REFERENCES

[ADAMg78] W.C. Adams and C.E. Giesler, 'Quantizing characteristics for signals having Laplacian amplitude probability density function,' *IEEE Trans. Commun.*, vol. COM-26, pp. 1295-1297, Aug. 1978.

[AHMEnr74] N. Ahmed, T. Natarajan and K.R. Rao, 'Discrete cosine transform,' *IEEE Trans. Computer*, Vol. C-23, pp.90-93, Jan. 1974.

[ALEXr82] S.T. Alexander and S.A. Rajala, 'Analysis and simulation of an adaptive coding system using the LMS algorithm,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.1219-1224, 1982.

[ALEXr85] S.T. Alexander and S.A. Rajala, 'Image compression results using LMS adaptive algorithm,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-33, pp. 712-717, Mar. 1985.

[ANDR70] H.C. Andrew, *Computer Techniques in Image Processing*, Academic Press, 1970.

[ANDRp68] H.C. Andrew and W.K. Pratt, 'Fourier transform coding of images,' *Proc. International Conf. on System Science*, Hawaii, pp.677-679, Jan. 1968.

[ARAVg86] R. Aravind and A. Gersho, 'Low-rate image coding with finite-state vector quantization,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.137-140, Apr. 1986.

[BAKEg82] R.L. Baker and R.M. Gray, 'Image compression using non-adaptive spatial vector quantization,' *Proc. Conf. Rec. Sixteenth Asilomar Conf. circuits, Syst, Comput.*, pp.55-61, Oct. 1982.

[BEIg85] C. Bei and R.M. Gray, 'An improvement of the minimum distortion encoding algorithm for vector quantization,' *IEEE Trans. Commun.*, vol. COM-33, pp.1132-1133, Oct. 1985.

[BERG71] T. Berger, *Rate Distortion Theory : A Mathematical Basis for Data Compression*, Prentice-Hall Inc., 1971.

[BGGE83] J.D. Bggerton, 'Optimal visual quantizrs for the transform source encoding of images,' *Ph.D. Thesis*, Southern Methodist University, Nov. 1983.

[BRAIp90] R.C. Brainard and A. Puri, 'Compact coder for component color television,' *IEEE Trans. Commun.*, Vol. COM-38, pp. 223-232, Feb. 1990.

[BRAN85] A.V. Brandt, 'Subband coding of videoconference signal using quadrature mirror filters,' *Proc. IASTED*, pp.212-215, June 1985.

[CASSsj89] P.M. Cassereau, D.H. Staelin and G.D. Jager, 'Encoding of images based on a lapped orthogonal transform,' *IEEE Trans. Commun.*, Vol. COM-37, pp. 189-193, Feb. 1989.

[CCUBE90] 'Preliminary data book of CL550A JPEG image compression processor,' C-Cube Microsystems, Feb. 1990.

[CHAM83] W.K. Cham, 'Transform coding of pictorial data,' *Ph.D. Thesis*, Loughborough University of Technology, Sep. 1983.

[CHAM89] W.K. Cham, 'Development of integer cosine transforms by the principle of dyadic symmetry,' *Proc. IEE*, vol.136, pt.I, no.4, pp.276-282, Aug. 1989.

[CHAMc84] W.K. Cham and R.J. Clarke, 'DC coefficient restoration in transform image coding,' *Proc. IEE*, vol.131, part F, pp. 709-713, Dec. 1984.

[CHAMc86] W.K. Cham and R.J. Clarke, 'Application of the principle of dyadic symmetry to the generation of orthogonal transforms,' *Proc. IEE*, vol.133, pt.F, no.3, pp.264-270, June 1986.

[CHAMc91] W.K. Cham and Y.T. Chan, 'An order-16 integer cosine transform,' *IEEE Trans. Signal Processing*, vol. 39, pp.1205-1208, May 1991.

[CHANG89] S.K. Chang, *Principles of Pictorial Information Systems Design*, Prentice - Hall, 1989.

[CHANh90] S.C. Chan and K.L. Ho, 'Direct methods for computing discrete sinusoidal transforms,' *Proc. IEE*, vol.137, pt.F, pp.433-442, Dec. 1990.

[CHENp84] W.H. Chen and W.K. Pratt, 'Scene adaptive coder,' *IEEE Trans. Commun.*, vol. COM-32, pp. 225-232, Mar. 1984.

[CHENs77] W.H. Chen and C.H. Smith, 'Adaptive coding of monochrome and color images,' *IEEE Trans. Commun.*, Vol. COM-25, pp. 1285-1292, Nov. 1977.

[CHENsf77] W.H. Chen, C.H. Smith and S.C. Fralick, 'A fast computational algorithm for the discrete cosine transform,' *IEEE Trans. Commun.*, vol. COM-25, pp. 1004-1009, Sep. 1977.

[CHENw91] L.H. Chen and S.F. Weng, 'A new image compression method using dynamic predictor based on current context,' *Siganl Processing*, vol.24, no.1, pp.43-59, July 1991.

[CLAR81] R.J. Clarke, 'Relation between the Karhunen-Loeve and cosine transforms,' *Proc. IEE*, vol. 128, pt. F, pp.359-360, Nov. 1981.

[CLAR84] R.J. Clarke, 'Hybrid intraframe transform coding of image data,' *Proc. IEE*, vol.131, pt.F, PP.2-6, Feb. 1984.

[CLAR85] R.J. Clarke, *Tansform Coding of Images*, Academic Press, 1985.

[DAVI72] L.D. Davision, 'Rate distortion theory and its applications,' *Proc. IEEE*, vol.60, no.7, pp.800-808, July 1972.

[DAVI90] A. Davignon, 'Block classification scheme using binary vector quantization for image coding,' *International Journal of Electronics*, vol.68, pp.667-673, May 1990.

[DELPm79] E.J. Delp and O.R. Mitchell, 'Image compression using block truncation coding,' *IEEE Trans. Commun.*, vol. COM-27, pp. 1335-1342, Sep. 1979.

[DIXIf91] S.S. Dixit and Y. Feng, 'Hierarchical address vector quantization for image coding,' *Graphical Models and Image Processing*, vo.53, no.1, pp.63-70, Jan. 1991.

[DUNHg85] M.O. Dunham and R.M. Gray, 'An algorithm for the design of labeled-transition finite-state vector quantization,' *IEEE Trans. Commun.*, vol. COM-33, pp. 83-89, Jan. 1985.

[EFRAlm91] N. Efrati, H. Liciztin and H.B. Mitchell, 'Classified block truncation coding - vector quantization: an edge sensitive image compression algorithm,' *Image Communication*, vol.3, pp.275-283, June 1991.

[EQUI89] W.H. Equitz, 'A new vector quantization clustering algorithm,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-37, pp. 1568-1575, Oct. 1989.

[FARR90] P.M. Farrelle, *Recursive Block Coding for Image Data Compression*, Springer - Verlag, 1990.

[FARRj86] P.M. Farrelle and A.K. Jain, 'Recursive block coding - a new approach to transform coding,' *IEEE Trans. Commun.*, Vol. COM-34, pp. 161-179, Feb. 1986.

[FENGn88] Y. Feng and N.M. Nasrabadi, 'Address-vector quantization : an adaptive vector quantization scheme using interblock correlation,' *Proc. SPIE Conf. Visual Commun. and Signal Proc.*, pp.214-222, 1988.

[FENGn89] Y. Feng and N.M. Nasrabadi, 'A dynamic address-vector quantization algorithm based on inter-block and inter-color correlation for color image coding,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.1755-1758, Apr. 1988.

[FENGn91] Y. Feng and N.M. Nasrabadi, 'Dynamic address-vector quantization of RGB colour images,' *Proc. IEE*, part-I, vol.138, pp.225-231, Aug. 1991.

[FLANmfrn89] J.K. Flanagan, D.R. Morrell, R.L. Frost, C.J. Read and B.E. Nelson, 'Vector quantization codebook generation using simulated annealing,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.1759-1762, 1989

[FORCk89] R. Forchheimer and T. Kronander, 'Image coding - from waveforms to animation,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-37, pp. 2008-2023, Dec. 1989.

[FOSTgd85] J. Foster, R.M. Gray and M.O. Dunham, 'Finite-state vector quantization for waveform coding,' *IEEE Trans. Information Theory*, vol. IT-31, pp.348-359, May 1985.

[FOX91] E.A. Fox, 'Advances in interactive digital multimedia systems,' *IEEE Computer Magazine*, pp.9-21, Oct. 1991.

[GALL75] M.M. Galloway, 'Texture analysis using gray level run-lengths,' *Computer Graphics and Image Processing*, no.2, pp.172-179, June 1975.

[GHARt86] H. Gharavim and A. Tabatabai, 'Subband coding of digital images using two-dimensional quadrature mirror filtering,' *Proc. SPIE Conf. Visual Commun. and Signal Proc.*, pp.51-61, Sep. 1986.

[GHARt88] H. Gharavi and A. Tabatabai, 'Sub-band coding of monochrome and color images,' *IEEE Trans. Circuits and Systems*, vol.CAS-35, pp.207-214, Feb. 1988.

[GONZw87] R.C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, 1987.

[GRAY84] R.M. Gray, 'Vector Quantization,' *IEEE ASSP Mag.*, pp.4-29, Apr. 1984.

[GRIShw87] N.C. Griswold, D.R. Halverson and G.L. Wise, 'A note on adaptive block truncation coding for image processing,' *IEEE Trans. Acoust. speech and Signal Proc.*, vol. ASSP-35, pp.1201-1203, Aug. 1987.

[HABE91] W. Habermann, 'HDTV standards,' *IEEE Communications Magazine*, vol.29, pp.10-12, Aug. 1991.

[HABI77] A. Habibi, 'Survey of adaptive image coding techniques,' *IEEE Trans. Commun.*, Vol. COM-25, pp. 1275-1284, Nov. 1977.

[HALVgw84] D.R. Halverson, N.C. Griswold and G.L. Wise, 'A generalized block truncation coding algorithm for image compression,' *IEEE Trans. Acoust. speech and Signal Proc.*, vol. ASSP-32, pp. 664-668, June 1984.

[HAMIp76] H. Hamidi and J. Peral, 'Comparison of the cosine and Fourier transforms of Markov-1 signals,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-24, pp. 428-429, Oct. 1976.

[HOU87] H.S. Hou, 'A fast recursive algorithm for computing the discrete cosine transform,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol.ASSP-35, pp.1455-1461, Oct. 1987.

[HUANc90] S.H. Huang and S.H. Chen, 'Fast encoding algorithm for VQ-based image coding,' *Electronics Letters*, vol.26, pp.1618-1619, Sep. 1990.

[HUANkt90] S.J. Huang, S.N. Koh and H.K. Tang, 'Algorithm for generating new orthogonal transforms,' *Electronics Letters*, vol.26, no.18, pp.1489-1490, Aug. 1990.

[HUFF52] D.A. Huffman, 'A method for the construction of minimum redundancy codes,' *Proc. IEEE*, vol.40, pp.1098-1101, Sep. 1952.

[HUI89] L. Hui, 'Optimum mean-square block truncation coding of monochrome images,' *Proc. International Conference on Image Proc.*, pp. 501-504, Singapore, Sep. 1989.

[HUI90] L. Hui, 'An adaptive block truncation coding algorithm for image compression,' *Proc. ICASSP*, pp.2233-2236, Apr. 1990.

[JAIN76] A.K. Jain, 'A fast KLT for a class of random processes,' *IEEE Trans. Commun.*, vol. COM-24, pp.1023-1029, Sep. 1976.

[JAIN79] A.K. Jain, 'A sinusoidal family of unitary transforms,' *IEEE Trans. Pattern Analy. Mach. Intell.*, vol. PAMI-1, pp.356-365, Oct. 1979.

[JAIN81] A.K. Jain, 'Image data compression : a review,' *Proc. IEEE*, Vol. 69, No. 3, pp. 349-389, Mar. 1981.

[JAIN89] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice - Hall, 1989.

[JAYAn84] N.S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice - Hall, 1984.

[JONEhk78] H.W. Jones, D.N. Hein and S.C. Knauer, 'The Karhunen-Loeve discrete cosine and related transforms obtained via the Hadamard transform,' *Int. Telemetering Conf.*, Los Angeles, Nov. 1978.

[JPEG90] 'JPEG technical specification, Revision 5,' JPEG-8-R5, Jan. 1990.

[KAMEsg91] M. Kamel, C.T. Sun and L. Guan, 'Image compression by variable block truncation coding with optimal threshold,' *IEEE Trans. Signal Processing*, vol. 39, pp. 208-212, Jan. 1991.

[KARLv88] G. Karlsson and M. Vetterli, 'Three-dimensional sub-band coding of video,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.1100-1104, Apr. 1988.

[KIM88] T. Kim, 'New finite state vector quantizers for images,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.1180-1183, Apr. 1988.

[KIMbsm88] C.S. Kim, J. Bruder, M.J.T. Smith and R.M. Mersereau, 'Subband coding of color images using finite state vector quantization,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.753-756, Apr. 1988.

[KIMl89] J.W. Kim and S.U. Lee, 'Discrete cosine transform - classified VQ technique for image coding,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.1831-1834, Apr. 1989.

[KIMl91] D.S. Kim and S.U. Lee, 'Image vector quantizer based on a classification in the DCT domain,' *IEEE Trans. Commun.*, vol.COM-39, pp.549-556, Apr. 1991.

[KIMl92] J.W. Kim and S.U. Lee, 'A transform domain classified vector quantizer for image coding,' *IEEE Trans. Circuits and Systems for Video Tech.*, vol.2, pp.3-14, Mar. 1992.

[KIMsm89] C.S. Kim, M.J.T. Smith and R.M. Mersereau, 'An improved SBC/VQ scheme for color image coding,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.1941-1944, Apr. 1989.

[KITA80] H. Kitajima, 'A symmetric cosine transform,' *IEEE Trans. Computer*, vol. C-29, pp.317-323, Apr. 1980.

[KOHht91] S.N. Koh, S.J. Huang and H.K. Tang, 'Development of order-16 integer transforms,' *Signal Processing*, vol.24, no.3, pp.283-289, Sep. 1991.

[KOU85] W.D. Kou, 'New orthogonal transform for digital signal processing,' *Electronics Letters*, vol.21, no.16, pp.666-667, Aug. 1985.

[KOUh86] W.D. Kou and Z. Hu, 'Several methods of constructing orthogonal transforms,' *ACTA Electronica Sinica*, (in Chinese), vol.14, no.3, pp.95-101, May 1986.

[KOUm89] W.D. Kou and J.W. Mark, 'A new look at DCT-type transforms,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-37, pp.1899-1908, Dec. 1989.

[KRISf86] P. Krishnasami, 'DPCM picture coding using two-dimensional quantizer,' *Signal Processing*, vol.10, pp.315-322, Apr. 1986.

[KUNTbl87] M. Kunt, M. Benard and R. Leonardi, 'Recent results in high-compression image coding,' *IEEE Trans. Circuits and Systems*, vol. CAS-34, pp.1306-1336, Nov. 1987.

[KUNTik85] M. Kunt, A. Ikonomopoulos and M. Kocher, 'Second-generation image coding techniques,' *Proc. IEEE*, vol.73, pp.549-574, Apr. 1985.

[KWAKsr83] H.S. Kwak, R. Srinivasan and K.R. Rao, 'C-matrix transform,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-31, pp.1304-1307, Oct. 1983.

[LEE84] B.G. Lee, 'A new algorithm for the discrete cosine transform,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-32, pp. 1243-1245, Dec. 1984.

[LEGAt88] D. LeGall and A. Tabatabai, 'Subband coding of digital images using short kernel filters and arithmetic coding techniques,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.761-765, Apr. 1988.

[LEMAm84] M.D. Lema and O.R. Mitchell, 'Absolute moment block truncation coding and its application to color images,' *IEEE Trans. Commun.*, vol. COM-32, pp. 1148-1157, Oct. 1984.

[LIM90] J.S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice - Hall, 1990.

[LINDbg80] Y. Linde, A. Buzo and R.M. Gray, 'An algorithm for vector quantizer design,' *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[LLOY82] S.P. Lloyd, 'Least squares quantization in PCM,' *IEEE Trans. Information Theory*, vol. IT-28, pp.129-137, Mar. 1982.

[MAKHrg85] J. Makhoul, S. Roucos and H. Gish, 'Vector quantization in speech coding,' *Proc. IEEE*, vol.73, no.11, pp.1551-1588, Nov. 1985.

[MALV90] H.S. Malvar, 'Lapped transforms for efficient transform/subband coding,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-38, pp.969-978, June 1990.

[MALVs88] H.S. Malvar and D.H. Staelin, 'Reduction of blocking effects in image coding with a lapped orthogonal transform,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.781-784, Apr. 1988.

[MALVs89] H.S. Malvar and D.H. Staelin, 'The LOT : transform coding without blocking effects,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-37, pp.553-559, Apr. 1989.

[MAX60] J. Max, 'Quantizing for minimum distortion,' *IRE Trans. Information Theory*, vol.6, pp.7-12, Mar. 1960.

[MORIhy85] Y. Morikawa, H. Hamada and N. Yamane, 'A fast algorithm for the cosine transform based on successive order-reduction of the Tchebycheff polynomial,' *Trans. Inst. Elec. Commun. Eng. (Japan)*, vol.J68-A, pp.173-180, 1985.

[MUSMpg85] H.G. Musmann, P. Pirsch and H.J. Grallert, 'Advances in picture coding,' *Proc. IEEE*, vol.73, no.4, pp.523-548, Apr. 1985.

[NARAp91] S.B. Narayanan and K.M.M. Prabhu, 'Fast Hartley transform pruning,' *IEEE Trans. Signal Processing*, vol.ASSP-39, pp.230-233, Jan. 1991.

[NASRc90] N.M. Nasrabadi and C.Y. Choo, 'Hierarchical block truncation coding of digital HDTV images,' *IEEE Trans. Consumer Electronics*, vol. 36, pp.254-261, Aug. 1990.

[NASRf90a] N.M. Nasrabadi and Y. Feng, 'A dynamic finite-state vector quantization scheme,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.2261-2264, Apr. 1990.

[NASRf90b] N.M. Nasrabadi and Y. Feng, 'A multilayer address vector quantization technique,' *IEEE Trans. Circuits and Systems*, vol. CAS-37, pp.912-921, July 1990.

[NASRf90c] N.M. Nasrabadi and Y. Feng, 'Image compression using address vector quantization,' *IEEE Trans. Commun.*, vol. COM-38, pp.2166-2173, Dec. 1990.

[NASRk88] N.M. Nasrabadi and R.A. King, 'Image coding using vector quantization : a review,' *IEEE Trans. Commun.*, vol. COM-36, pp. 957-971, Aug. 1988.

[NDIFe91] J. Ngwa-Ndifor and T. Ellis, 'Predictive partial search algorithm for vector quantization,' *Electronics Letters*, vol.27, pp.1722-1723, Sep. 1991.

[NETR77] A.N. Netravali, 'On quantizers for DPCM coding of picture signal,' *IEEE Trans. Information Theory*, vol.IT-23, pp.360-370, May 1977.

[NETRh88] A.N. Netravali and B.G. Haskell, *Digital Pictures - Representation and Compression*, Plenum Press, 1988.

[NETRl80] A.N. Natravali and J.O. Limb, 'Picture coding : a review,' *Proc. IEEE*, vol. 68, pp. 366-406, Mar. 1980.

[NGAN82] K.N. Ngan, 'Adaptive transform coding of video signals,' *Proc. IEE*, vol.129, pt.F, pp.28-40, Feb. 1982.

[NGANk89] K.N. Ngan and H.C. Koh, 'Classified hybrid image coder,' Proc. *Int. Conf. Image Processing*, pp.45-48, Sep. 1989.

[NGANkw91] K.N. Ngan, H.C. Koh and W.C. Wong, 'A hybrid image coding scheme using CVQ and DCT,' *Proc. Int. Symposium Circuits and Systems*, pp.408-411, June 1991.

[NGANls89] K.N. Ngan, K.S. Leong and H. Singh, 'Adaptive cosine transform coding of images in perceptual domain,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-37, pp.1743-1750, Nov. 1989.

[NICOfcnc87] R.C. Nicol, B.A. Fenn, R.J. Clarke, K.N. Ngan and W.K. Cham, 'Method and apparatus for transmitting an image,' *European Patent Office*, Patent no. : 8230382502, July 1987.

[NINO91] Y. Ninomiya, 'HDTV broadcasting systems,' *IEEE Communications Magazine*, vol.29, pp.15-22, Aug. 1991.

[NOLLz79] P. Noll and R. Zelinski, 'Comments on "Quantizing characteristics for signals having Laplacian amplitude probability density function",' *IEEE Trans. Commun.*, vol. COM-27, pp. 1259-1260, Aug. 1979.

[PALIr89] K.K. Paliwal and V. Ramasubramanian, 'Effect of ordering the codebook on the efficiency of the partial distance search algorithm for vector quantization,' *IEEE Trans. Commun.*, vol. COM-37, pp.538-540, May 1989.

[PIRS82] P. Pirsch, 'Adaptive intra interframe DPCM coder,' *Bell System Technical Journal*, vol.61, pp.747-764, 1982.

[POc91] L.M. Po and C.K. Chan, 'Directionally classified subspace image vector quantization,' *Proc. Int. Conf. on Circuits and Systems*, pp.336-339, June 1991.

[POLYr82] A. Polysongsang and K.R. Rao, ’DCT/DPCM processing of NTSC composite video signal,’ *IEEE Trans. Commun.*, Vol. COM-30, pp. 541-549, Mar. 1982.

[PRAT79] W.K. Pratt, *Digital Image Processing*, Wiley-Interscience, 1979.

[PRATcw72] W.K. Pratt, W.H. Chen and L.R. Welch, ’Slant transform for image coding,’ *Proc. Symposium Appl. of Walsh Functions*, Mar. 1972.

[PRATk69] W.K. Pratt and J. Kane, ’Hadamard transform image coding,’ *Proc. IEEE*, vol. 57, pp.58-68, Jan. 1969.

[RAk91] S.W. Ra and J.K. Kim, ’Fast weight-ordered search algorithm for image vector quantization,’ *Electronics Letters*, vol.27, pp.2081-2083, Oct. 1991.

[RAMAg86] B. Ramamurthi and A. Gersho, ’Classified vector quantization of images,’ *IEEE Trans. Commun.*, vol. COM-34, pp. 1105-1115, Nov. 1986.

[RAYd70] W.D. Ray and R.M. Drive, ’Further decomposition of the Karhunen-Loeve series representation of a stationary random process,’ *IEEE Trans. Information Theory*, vol. IT-16, pp.845-850, Nov. 1970.

[REINg83] R.C. Reininger and J.D. Gibson, ’Distribution of the two-dimensional DCT coefficients for images,’ *IEEE Trans. Commun.*, vol. COM-31, pp. 835-839, Nov. 1986.

[ROSEhd90] K. Rose, A. Heiman and I. Dinstein, ’DCT/DST alternate-transform image coding,’ *IEEE Trans. Commun.*, Vol. COM-38, pp. 94-101, Jan. 1990.

[ROSEk82] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, volumn 1, Academic Press, 1982.

[SOLEm87a] M.R. Soleymani and S.D. Morgera, ’An efficient nearest neighbor search method,’ *IEEE Trans. Commun.*, vol. COM-35, pp.677-679, June 1987.

[SOLEm87b] M.R. Soleymani and S.D. Morgera, ’A high-speed search algorithm for vector quantization,’ *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.1946-1948, 1987.

[SOLEm89] M.R. Soleymani and S.D. Morgera, ’A fast MMSE encoding technique for vector quantization,’ *IEEE Trans. Commun.*, vol. COM-37, pp.656-659, June 1989.

[SRINr83] R. Srinivasan and K.R. Rao, ’An approximation to the discrete cosine transform for N=16,’ *Signal Processing*, vol.5, no.1, pp.81-85, Jan. 1983.

[UDPIr85] V.R. Udpikar and J.P. Raina, 'Modified algorithm for block truncation coding of monochrome images,' *Electronics Letter*, vol. 21, pp. 900-902, Sep. 1985.

[UDPIr87] V.R. Udpikar and J.P. Raina, 'BTC image coding using vector quantization,' *IEEE Trans. Commun.*, vol. COM-35, pp. 352-356, Mar. 1987.

[VAISg88] J. Vaisey and A. Gersho, 'Simulated annealing and codebook design,' *Proc. IEEE Int. Conf. Acoust. Speech and signal Proc.*, pp.1176-1179, 1988.

[VETT84] M. Vetterli, 'Multi-dimensional sub-band coding : some theory and algorithms,' *Signal Processing*, vol.6, pp.97-112, Jan. 1984.

[VETTn84] M. Vetterli and H. Nussbaumer, 'Simple FFT and DCT algorithms with reduced number of operations,' *Signal Processing*, vol.6, no.4, pp.267-278, Aug. 1984.

[WALL92] G.K. Wallace, 'The JPEG still picture compression standard,' *IEEE Trans. Consumer Electronics*, vol.38, pp.xviii-xxxiv, Feb. 1992.

[WANG84] Z. Wang, ' Fast algorithm for the discrete W transform and for the discrete cosine transform,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-32, pp. 803-816, Aug. 1984.

[WANG86] Z. Wang, 'The phase shift cosine transform,' *ACTA Electronica Sinica*, Vol.14, No.6, pp.11-19, Nov. 1986.

[WANG91] Z. Wang, 'Pruning the fast discrete cosine transform,' *IEEE Trans. Commun.*, vol.COM-39, pp.640-643, May 1991.

[WANGg88] L. Wang and M. Goldberg, 'Progressive image transmission by transform coefficient residual error quantization,' *IEEE Trans. Commun.*, Vol. COM-36, pp. 75-87, Jan. 1988.

[WANGr91] Y. Wang and V. Ramamoorthy, 'Image reconstruction from partial subband images and its application in packet video transmission,' *Image Communcation*, vol.3, pp.197-229, June 1991.

[WONGs81] W.C. Wong and R. Steele, 'Adaptive discrete cosine transformatio of pictures using an energy distribution logarithmic model,' *The Radio and Electronic Engineer*, vol.51, no.11/12, pp.571-578, Nov. 1981.

[WOOD91] J.W. Woods, *Subband Image Coding*, Kluwer International, 1991.

[WOODn86] J.W. Woods and S.D. O'Neil, 'Subband coding of images,' *IEEE Trans. Acoust. Speech and Signal Proc.*, vol. ASSP-34, pp. 1278-1288, Oct. 1986.

[WU91] X. Wu, 'Optimal bi-level quantization and its application to multilevel quantization,' *IEEE Trans. Information Theory,* vol. IT-37, pp.160-163, Jan. 1991.

APPENDIX A Statistics of Monochrome Test Images

Picture	Size	Mean	Variance	ρ
Woman	256 x 256	95.81	1666.84	0.979
Lenna	256 x 256	123.57	2296.66	0.943
Peppers	256 x 256	119.62	2922.48	0.949
Sailboat	256 x 256	124.54	4315.27	0.936
Baboon	256 x 256	129.48	1769.23	0.685
Ship	256 x 256	134.23	2873.73	0.961
Bluetit	256 x 256	73.07	797.88	0.969
Lee	256 x 256	61.96	945.77	0.978
Railway	256 x 256	91.31	1131.30	0.881
CU	256 x 256	116.79	912.42	0.959

APPENDIX B Statistics of Color Test Images

Picture	Mean	Variance	ρ
Lenna	124.21	2298.55	0.953
Peppers	120.12	2929.34	0.966
Sailboat	125.24	4338.34	0.965

a. Luminance Y

Picture	Mean	Variance	ρ
Lenna	46.37	296.52	0.986
Peppers	36.34	1232.51	0.966
Sailboat	7.39	723.29	0.912

b. Chrominance U

Picture	Mean	Variance	ρ
Lenna	45.83	45.63	0.996
Peppers	23.25	214.71	0.971
Sailboat	32.08	189.04	0.977

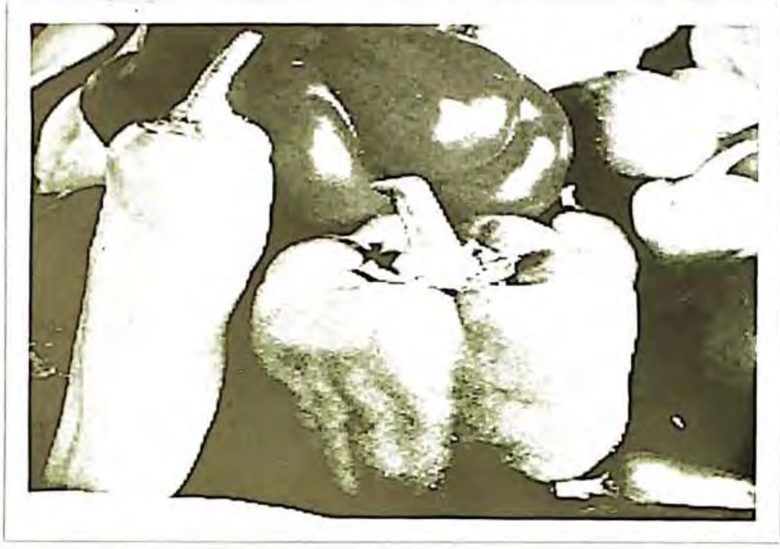
c. Chrominance V

APPENDIX C Fortran Program Listing for the Pruned Fast DCT Algorithm

```
C     DIRECT FAST DCT WITH PRUNING
C     BY KWOK-TUNG LO
C
C     PARAMETER
C     X - INPUT AND OUTPUT ARRAY
C     N - ORDER OF THE TRANSFORM
C     M - N = 2**M
C     R - NO. OF THE FIRST 2**R COEFFICIENTS TO BE COMPUTED
C
      SUBROUTINE PFCT(X,N,M,R)
C     4-POINT BUTTERFLY
      G=0
      DO 10 I=1,M-1
            N2=2**(M-1-I)
            N3=2**(M+1-I)
            IF (I.LT.R+1) THEN
                  N1=2**(I-1)
                  DO 20 K=1,N1
                        DO 20 L=1,N2
                              L1=L+(K-1)*N3
                              L2=N3/2-L+(K-1)*N3+1
                              L3=N3/2+L+(K-1)*N3
                              L4=N3-L+(K-1)*N3+1
                              C1=2*COS((N3-2*L+1)*PI/(2*N3))
                              C2=2*COS((2*L-1)*PI/(2*N3))
                              T1=X(L1)
                              T2=X(L2)
                              T3=T(L3)
                              X(L1)=X(L1)+X(L4)
                              X(L2)=X(L2)+X(L3)
                              X(L3)=C2*(T1-X(L4))
                              X(L4)=C1*(T2-T3)
20                      CONTINUE
            ELSE
                  N1=2**R
                  DO 30 K=1,N1
                        N4=(K-1)*(2**G)
                        DO 40 L=1,N2
                              L1=L+N4*N3
                              L2=N3/2-L+N4*N3+1
                              L3=N3/2+L+N4*N3
                              L4=N3-L+N3*N4+1
                              X(L1)=X(L1)+X(L4)
                              X(L2)=X(L2)+X(L3)
40                      CONTINUE
                        G=G+1
30                CONTINUE
            ENDIF
10    CONTINUE
C     2-POINT BUTTERFLY
      IF (R.EQ.0) THEN
            X(1)=X(1)+X(2)
      ELSE
            H=M-R
            N1=2**(R-1)
            N2=2**(H+1)
            IF (R.LT.M) THEN
```

```
                N1=N1*2
                N2=N2/2
            ENDIF
                DO 50 I=1,N1
                    T1=X((I-1)*N2+1)
                    X((I-1)*N2+1)=X((I-1)*N2+1)+X((I-1)*N2+2)
                    IF (R.EQ.M) X((I-1)*N2+2)=(T1-X((I-1)*N2+2))/SQRT(2)
50              CONTINUE
        ENDIF
C       PERMUTATION AND RECURSIVE ADDITION
        IF (R.GT.0) THEN
            IF (R.EQ.M) THEN
                N3=2
                N4=H+1
            ELSE
                N3=1
                N4=H-1
            ENDIF
            DO 60 I=1,(M-1)
                N1=2**(M-I-1)
                N2=2**I
                DO 70 K=1,N1
                    X(N2+1+2*(K-1)*N2)=X(N2+1+2*(K-1)*N2)/2
                    IF (N3.GT.1) THEN
                        DO 80 L=1,(N2-1)
                            L1=N2+L+1+2*(K-1)*N2
                            X(L1)=X(L1)-X(L1-1)
80                      CONTINUE
                        DO 90 L=1,(2*N2-1)
                            T(L+1)=X(L+1+2*(K-1)*N2)
90                      CONTINUE
                        DO 100 L=1,(N2-1)
                            X(2*L+1+2*(K-1)*N2)=T(L+1)
                            X(2*L+2*(K-1)*N2)=T(N2+L)
100                     CONTINUE
                    ENDIF
70              CONTINUE
                N3=N3*2
60          CONTINUE
        ENDIF
C       COEFFICIENT SCALING
        X(1)=X(1)/SQRT(N)
        DO 110 I=2,(2**R)
            X(I)=X(I)/SQRT(N/2)
110     CONTINUE
        RETURN
        END
```

APPENDIX D Training set images for building the codebook for standard VQ scheme



a. Peppers

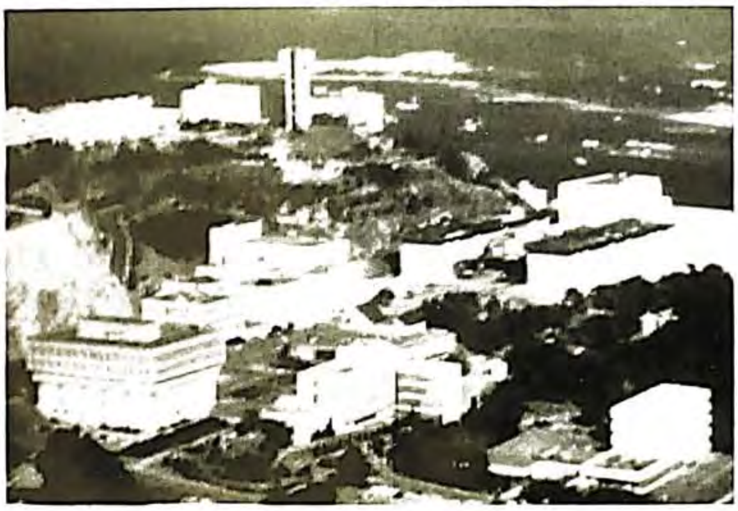
b. Baboon

c. Woman

d. Ship

e. Lee

f. Bluetit



g. CU

h. Railway

APPENDIX E LIST OF PUBLICATIONS

[1] K.T. Lo and W.K. Cham, 'Image coding using weighted cosine transform,' *Proceeding TENCON'90 on Computer and Communication System*, pp.464-468, Hong Kong, Sep. 1990.

[2] C.T. See, K.T. Lo and W.K. Cham, 'Efficient encoding of DC coefficients in transform coding of images using JPEG scheme,' *Proceeding of the International Symposium on Circuits and Systems*, pp.404-107, Singapore, June 1991.

[3] K.T. Lo and W.K. Cham, 'Pruning the fast cosine transform,' *Proceeding of the International Symposium on Communications*, pp.584-587, Taiwan, Dec. 1991.

[4] K.T. Lo and W.K. Cham, 'An efficient encoding algorithm for vector quantization of images,' *Proceeding of the Sixth European Signal Processing Conference*, Belgium, Aug. 1992.

[5] K.T. Lo and W.K. Cham, 'Classified vector quantization of images using three-level block truncation coding,' *Proceeding of the Third International Symposium on Signal Processing and its Applcations (ISSPA 92)*, pp.315-318, Gold Coast, Australia, Aug. 1992.

[6] K.T. Lo and W.K. Cham, 'An efficient VQ encoding algorithm for image coding,' *Proceeding of the second International Conference on Image Processing*, Singapore, Sep. 1992.

[7] K.T. Lo and W.K. Cham, 'Sub-codebook searching algorithm for efficient VQ encoding of images,' *Electronics Letters* (in press).

[8] K.T. Lo and W.K. Cham, 'A modified cosine transform,' *Journal of Visual Communication and Image Representation* (in press).

[9] K.T. Lo, W.K. Cham and C.T. See, 'Minimum edge difference predictor for encoding the DC coefficients in transform coding of images,' submitted for publication, IEEE Trans. Commun., Sep. 1991.

[10] K.T. Lo and W.K. Cham, 'Direct fast cosine transform pruning,' submitted for publication, IEE Proceedings (part-I), Mar. 1992.

[11] K.T. Lo and W.K. Cham, 'A fast VQ encoding algorithm for image compression,' Submitted for publication, IS&T/SPIE Symposium on Electronic Imaging : Science and Technology, California USA, Jan. 1993.