INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI®

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Structured Vector Quantizers in Image Coding

Manijeh Khataie

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

December 1999

ABSTRACT

Structured Vector Quantizers in Image Coding

Manijeh Khataie, Ph. D.

Concordia University, 1999

Data compression has become an essential part of modern digital communication, video signal processing, and storage systems. Although the bandwidth of communication networks has been increasing continuously, the introduction of new services and the expansion of the existing ones demand an even higher bandwidth. Image data compression is concerned with the minimization of the volume of data used to represent an image.

For a typical image, the values of adjacent pixels are highly correlated. The transform and predictive codings use this correlation between the neighbors to achieve a high degree of compression. The goal of transform coding is to decorrelate the pixel values and redistribute the signal energy among only a small set of transform coefficients. For most images, the Discrete cosine transform (DCT) is very close to an optimum transform.

In recent years, image compression algorithms using Vector Quantization (VQ) have been receiving considerable attention. Unstructured vector quantizers, i.e., those with no restriction on the geometrical structure of the codebook, suffer from two basic drawbacks, viz., the codebook search complexity and the large storage requirement. This explains the interest in the structured VQ schemes, such as lattice-based VQ and multi-stage VQ.

The objective of this thesis is to devise techniques to reduce the complexity of vector quantizers. In order to reduce the codebook search complexity and memory requirement, a universal Gaussian codebook in a residual VQ or a lattice-based VQ is used. To achieve a better performance, a part of work has been done in the frequency domain. Specifically, in order to retain the high-frequency coefficients in transform coding, two methods are suggested. One is developed for moderate to high rate data compression while the other is effective for low to moderate data rate.

In the first part of this thesis, a residual VQ using a low rate optimal VQ in the first-stage and a Gaussian codebook in the other stages are introduced. From rate distortion theory, for most memoryless sources and many Gaussian sources with memory, the quantization error under MSE criterion, for small distortion, is memoryless and Gaussian. For VQ with a realistic rate, the error signal has a non-Gaussian distribution. It is shown that the distribution of locally normalized error signals, however, becomes close to a Gaussian distribution.

In the second part, a new two-stage quantizer is proposed. The function of the first stage is to encode the more important low-pass components of the image and that of the second is to do the same for the high-frequency components ignored in the first stage. In one scheme, a high-rate lattice-based vector quantizer is used as the quantizer for both stages. In another scheme, the standard JPEG with a low rate is used as the quantizer of the first stage, and a lattice-based VQ is used for the second stage. The resulting bit rate of the two-stage lattice-based VQ in either scheme is found to be considerably better than that of JPEG for moderate to high bit rates.

In the third part of the thesis, a method to retain the high-frequency coefficients is proposed by using a relatively huge codebook obtained by truncating the lattices with a large radius. As a result, a large number of points fall inside the boundary of the codebook, and thus, the images are encoded with high quality and low complexity. To reduce the bit rate, a shorter representation is assigned to the more frequently used lattice points. To index the large number of lattice points which fall inside the boundary, two methods that are based on grouping of the lattice points according to their frequencies of occurrence are proposed. For most of the test images, the proposed methods of retaining high-frequency coefficients is found to outperform JPEG.

ACKNOWLEDGEMENTS

I am deeply indebted to the many people who have supported me in making the completion this thesis. First I would like to express my special appreciation to my supervisor, Professor M. Reza Soleymany for his well understanding, constant encouragement and guidance during my study and research. I owe a great deal of thanks to my other supervisor, Professor M. Omair Ahmad, who read the first draft of my thesis very carefully and made numerous corrections and valuable suggestions for improvements. I also wish to thank my manager in Harris, Mary Shields, for providing a flexible schedule to make the completion of my thesis easier.

Finally, I would like to give special thanks to my parents who devoted their lives to support their children and also to my husband and my children for their continuing love and support throughout this study.

This work has been supported by Natural Science and Engineering Research Council (NSERC) grants OGP0036575 and OGPIN011.

To my mother and my late father

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

DCT	Discrete cosine transform
VQ	Vector quantizer/ quantization
DPCM	Differential pulse code modulation
PCM	Pulse code modulation
KLT	Karhunen-Loeve transform
DFT	Discrete Fourier transform
DHT	Discrete Hadamard transform
DWHT	Discrete Walsh Hadamard transform
DST	Discrete sine transform
JPEG	Joint Photographic Expert Group
CCITT	International Telegraph and Telephone Consultative Committee
ISO	International Organization for Standardization
IEG	International Electrotechnical Commission
2D-DCT	Two-dimensional DCT
MSE	Mean squared error
GLA	Generalized Lloyd algorithm
TSVQ	Tree-search vector quantizer
FSVQ	Finite-state vector quantizer
LBVQ	Lattice-based vector quantizer
RVQ	Residual vector quantizer
i.i.d.	Independent, identically distributed
AEP	Asymptotic equipartition property
d.m.s.	Discrete memoryless source
bpb	Bit per block
bps	Bit per sample

PSNR	Peak signal to noise ratio
SNR	Signal to noise ratio
RQ	Residual quantizer

Chapter 1

INTRODUCTION

1.1 GENERAL

The bandwidth of the communication networks has been increasing continuously as a result of technological advances. However, the introduction of new services and the expansion of the existing ones have resulted in an even higher demand for the bandwidth. This explains the many efforts currently being invested in the area of data compression. The primary goal of these works is to develop techniques of coding information sources such as speech, image and video so as to reduce the number of bits required to represent a source without significantly degrading its quality. Image and video compression is essential for image transmission applications such as TV transmission, video conferencing, remote sensing via satellite, aircraft, radar or sonar and facsimile transmission of printed materials as well as where pictures are stored in databases, such as archiving medical images, finger prints, educational and business documents and drawings.

1.2 DATA COMPRESSION

Data compression techniques can be classified into two categories, lossless and lossy. Lossless data compression techniques permit perfect reconstruction of the original information, whereas the lossy schemes do not guarantee perfect reconstruction. However, they offer better compression ratios.

In many applications, such as computerized tomography and satellite remote sensing of images, where image data is constantly produced for archival storage, no information should be lost during the process. Therefore, in these cases, one has to use a lossless scheme. In lossless compression, the shorter indices are assigned to gray levels that occur more often. Huffman coding [2] and arithmetic coding [3] are two examples of lossless compression.

Lossy compression techniques reduce the number of bits required for the reconstruction of the source by introducing some distortion in the data. For a given source, the amount of distortion depends on the degree of compression. Typically, images have a high degree of correlation between the adjacent pixels. Most compression techniques use this correlation between the neighboring pixels in order to achieve a considerable compression. These methods exploit a set of uncorrelated parameters that represent a picture and from which the picture can be reproduced. Transform coding [4] expands a picture in terms of a family of orthonormal functions and takes the coefficients of the expansion as a representation of the picture. If we do not limit to linear orthogonal transformation, there are other techniques that achieve the same result. One such technique is the predictive compression [4]. Because of the strong correlation between the pixels of an image, it is possible to derive an estimate or prediction, $\hat{x}(m,n)$, for a given element $x(m,n)$ in terms of its neighboring picture elements. The difference $e(m,n) = x(m,n) - \hat{x}(m,n)$ is the estimation error for the picture elements. It is reasonable to expect that the random variable $e(m,n)$ should be less correlated than the elements in the original picture.

Vector Quantization (VQ) [5] is another example of a lossy data compression techniques. In a vector quantizer, the data sequence is quantized in groups (blocks) instead of individually. It is well known that the vector quantization always results in a better performance than the scalar quantization [6], [7]. Although the performance of an optimum Vector Quantizer(VQ) is good, the quantization and encoding steps are complex. Lack of a structure in an optimum vector quantizer is the reason for its complexity. This explains the interest in VQ schemes with structured codebooks, such as tree searched [8], residual (multi-stage) [9], gain/shape [10], and lattice-based vector quantizers [11]. Due to the superior performance of VQ in comparison to scalar quantization, use of VQ in conjunction with a predictive or transform coding technique usually yields a better performance.

One of the results of any transformation is that the signal energy is distributed among a small set of transform coefficients. Most of the compression in transform coding is a result of dropping small-valued coefficients and coarsely quantizing the others. Optimal bit allocation [4] is a complex strategy, especially if it is adaptive. It involves quantizers with different number of levels and reassignment procedures. This explains the reason for interests in non-optimal techniques. In zonal coding [4], the coefficients with index less than a predefined value are retained and the rest are set to zero. The zonal coding has been improved by proposing a classified transformer [4], in which depending on the activity content of the block, different bit assignment matrices are used. In some other methods, the transform matrix is divided into different zones and each zone is quantized with different quantizers [12], [13]. In [12], a scalar quantizer has been used for low-frequency coefficients while high-frequency coefficients are vector quantized. In [13], the transform coefficients are grouped in a zig-zag order, and each group are vector quantized. In [14],[15], using quantization table or weighted pyramid VQ, the high frequency-coefficients are given some small weights. In most of these methods, the high-frequency coefficients are almost neglected. Although the energy of these coefficients are small, retaining them could result in a better performance.

Lattice-Based Vector Quantizer (LBVQ) is a structured VQ technique in which the lattice points are used as a codebook of VQ. The lattice-based vector quantizer proposed in [16] and [17], has been extensively studied by many researchers [11], [18], [19]. Because of the regular structure of the LBVQ, its use results in a drastic reduction in the complexity in comparison to an optimum VQ for the same rate and vector dimension. Codebook storage is eliminated, since lattices are easily generated and mapping between lattice points and binary words are known. Since a lattice is a set of points which are uniformly distributed[20], using LBVQ is optimum for uniformly distributed sources. However, LBVQ has also been used for Gaussian and

Laplacian sources, showing a good performance [19], [21]. Usually in a lattice-based vector quantizer, the lattice is truncated such that the desired number of lattice points fall inside the boundary. For a source with a given probability density function (pdf), only a few of these lattice points are used. To take advantage of the source regularities, geometric vector quantizer has been suggested [15], [22]. Efficient algorithms exist for implementing a lattice quantizer with an N-dimensional hypercube boundary. However, for other desirable boundaries, such as sphere or pyramid, indexing is still a problem. In the existing methods, indexing requires excessive storage or complex enumeration algorithms [15], [23].

1.3 SCOPE AND ORGANIZATION OF THE THESIS

The objective of this thesis is to devise techniques to reduce the complexity of vector quantizer. In order to reduce the codebook search complexity and memory requirement, a universal Gaussian codebook in a residual VQ or a lattice-based VQ is suggested. Since for all images only one codebook is needed in different stages of a residual VQ, different structures and mapping techniques can be developed to reduce the search complexity. The effect of high-frequency coefficients in transform coding is also investigated by taking into account the indexing problem in lattice-based vector quantization. Based on this study, a technique is developed to include the quantized high-frequency coefficients in order to improve the quality of the reconstructed images without significantly increasing the bit rates.

This thesis is organized as follows. Chapter 2 reviews the necessary background material to carry out the proposed investigation. The basic element of data compression such as Transform coding, different kinds of transformations, predictive

coding, entropy coding, and vector quantization are discussed.

In Chapter 3 a Gaussian codebook to quantize error samples in the residual VQ is presented. The scheme is based on a multi-stage residual VQ. A well known result of rate-distortion theory states that, under broad conditions, the quantization error has a Gaussian distribution. It is also known that a Gaussian memoryless source is successively refinable. Since the use of codebooks designed for a generic Gaussian source for different stages of a residual vector quantizer does not result in loss of performance, a residual vector quantizer using an optimal vector quantizer in the first stage and a Gaussian codebook in the other stages have been introduced. The closeness of the distribution of the error signals to the Gaussian distribution is examined and the loss in optimality of the codebook for the error signal when the rate is not high is also studied.

In Chapter 4, two-stage residual image coding technique that uses transform coding and the lattice based VQ is presented. To exploit most of the memory sources, a transform coding with a lattice-based VQ is used. The imposition of additional structure on the multi-stage VQ makes the code more submissive to a sequential search. In the proposed method, the second stage is added to retain the information lost in the first stage. A standard JPEG or a DCT transform coding is used for the first stage, and an optimum VQ, a lattice-based VQ and a Gaussian codebook is used as the quantizer for the second stage. The effect of adding the second stage in improving the performance of the quantization in terms of the compression ratio and the image quality is studied.

Chapter 5 concentrates on the indexing of the lattice points used as a codebook for image transform coding. In order to improve the quality of a compressed image, a large number of lattice points must be selected as codewords to represent

the coefficients with small energy in transform coding. However, the large number of lattice points results in a high bit rate. To reduce the bit rate, a shorter representation with appropriate indexing must be assigned to the more frequently used lattice points. In this chapter, two methods to index the large number of lattice points that fall inside the prescribed boundary, are proposed. Both these methods are based on grouping of the lattice points according to their frequencies of occurrence. In the first method, these points are grouped based on the non-zero elements of the quantized scaled DCT coefficients. In the second one, the grouping is carried out according to the radial parameter.

Chapter 6 highlights the important findings of the investigation carried out in thesis and gives suggestion for further study.

Chapter 2

BACKGROUND

2.1 INTRODUCTION

Transform coding and predictive coding are two well-known methods for redundancy reduction in image coding. Both techniques remove the linear dependency between the neighbor pixels. In transform coding we use only the linear orthogonal transformation. A variety of techniques can be used to quantize the transformed coefficients or error signals. Scalar quantizer is simple to implement, and vector quantization performs better but is more complex. Lattice-based vector quantizer reduces this complexity using the structured lattice points as a codebook. Entropy coding is an efficient method for encoding the predicted or transformed image information. This chapter is a brief review of these techniques which have also been used and refereed to in this study.

2.2 PREDICTIVE CODING

Among the many different predictive coding methods, the Differential Pulse Code Modulation (DPCM) is the most common one. In this method, error signal, the difference between the previously quantized samples and the new samples, are quantized and encoded. Figure 2.1 shows the block diagram of the encoding and decoding operation involved with a DPCM. The correlation between the different samples of

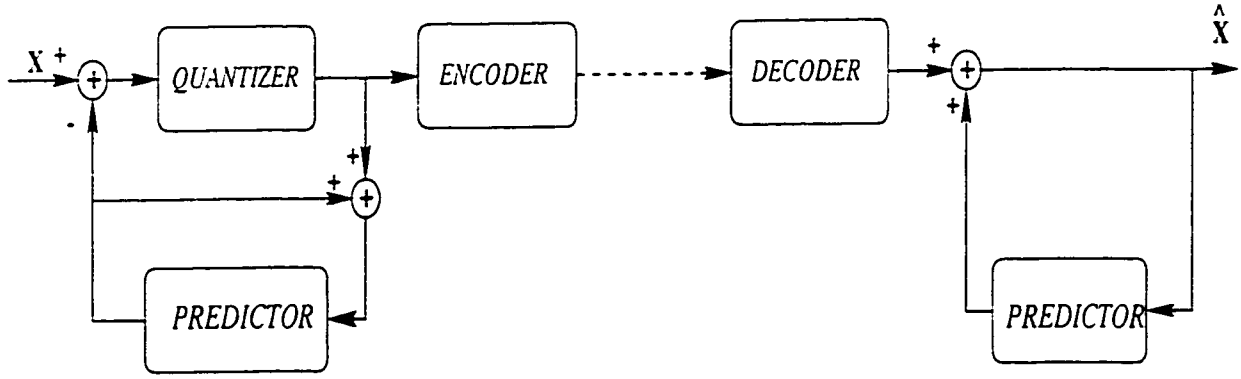

Figure 2.1: DPCM block diagram.

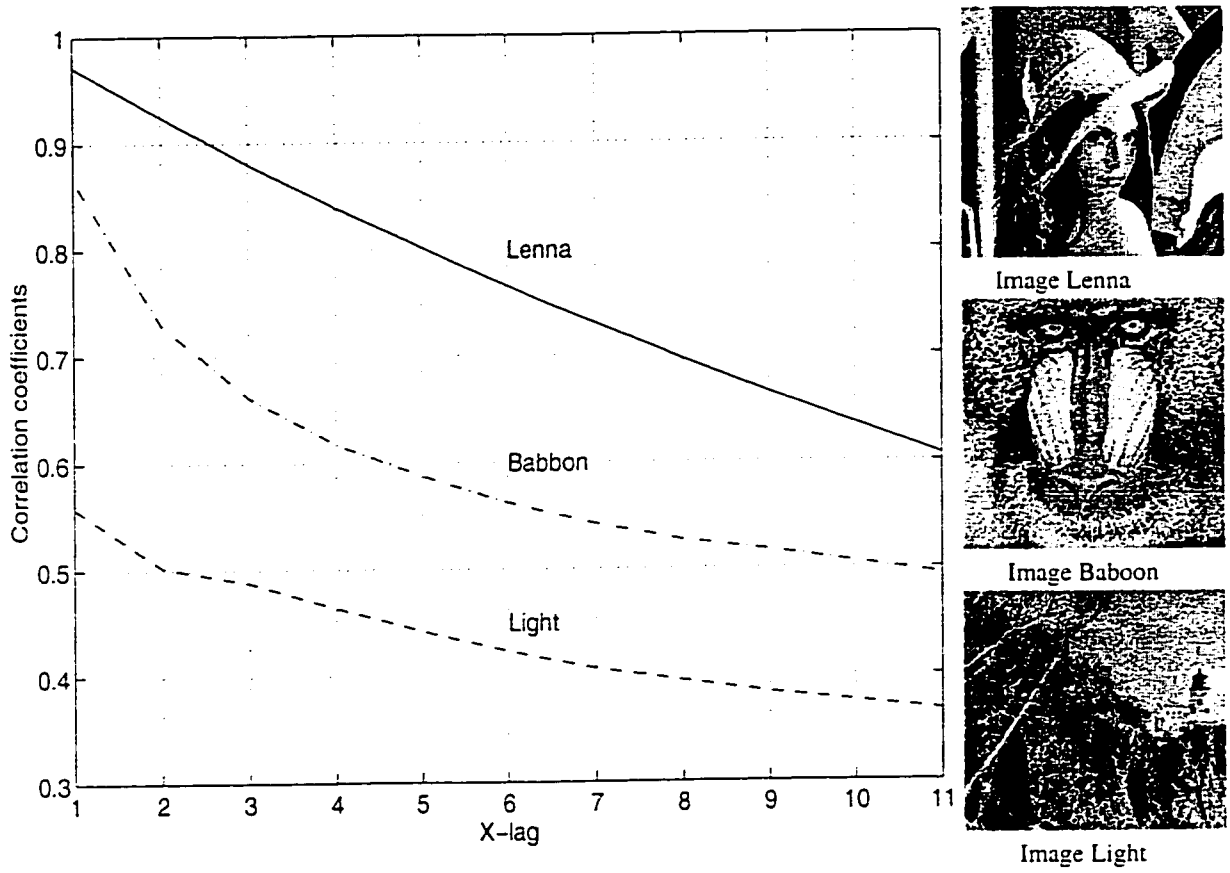

Figure 2.2: Correlation coefficient of some images.

the error signal is much less than the correlation between the original signal samples. In other words, the redundancy of the quantized samples is reduced. In this way the image can be quantized more efficiently. The correlation of the adjacent pixels for different images are shown in Figure 2.2. Figure 2.3 compares the correlation of pixels of the error signals and original samples. As it can be seen, the error signals are less correlated than the original image. The distribution function of the error signal is shown in Figure 2.4. As expected, the dynamic range of the error signal is smaller than the original one. For example, for the image *Lenna* the samples' amplitudes are between 0 and 255; however, as it can be seen in the Figure 2.4, the error samples are almost between -30 and 30. Hence, less number of bits are needed to encode an error signal.

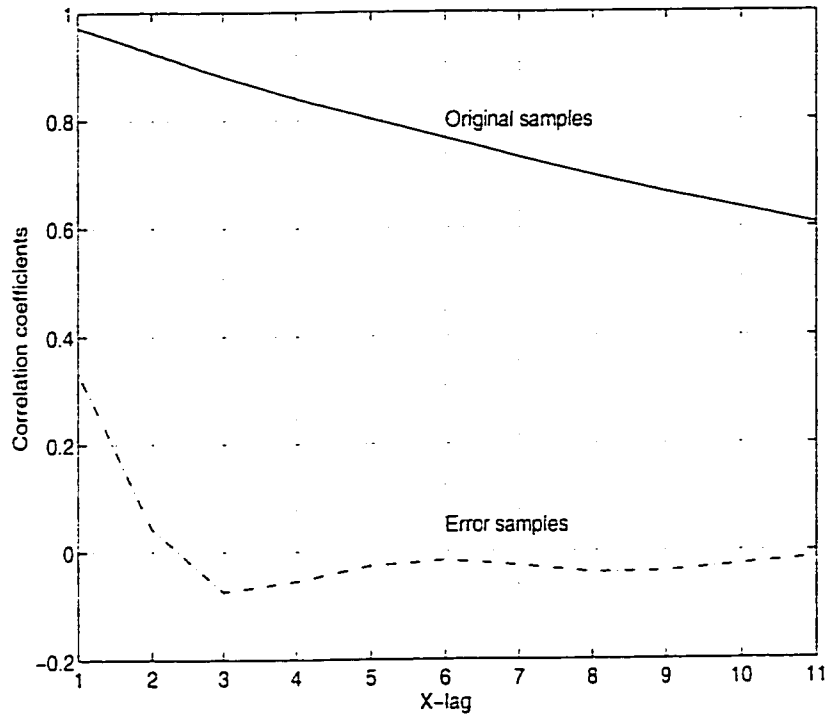

Figure 2.3: Comparison of correlation coefficient of error image and original image of the image *Lenna*.

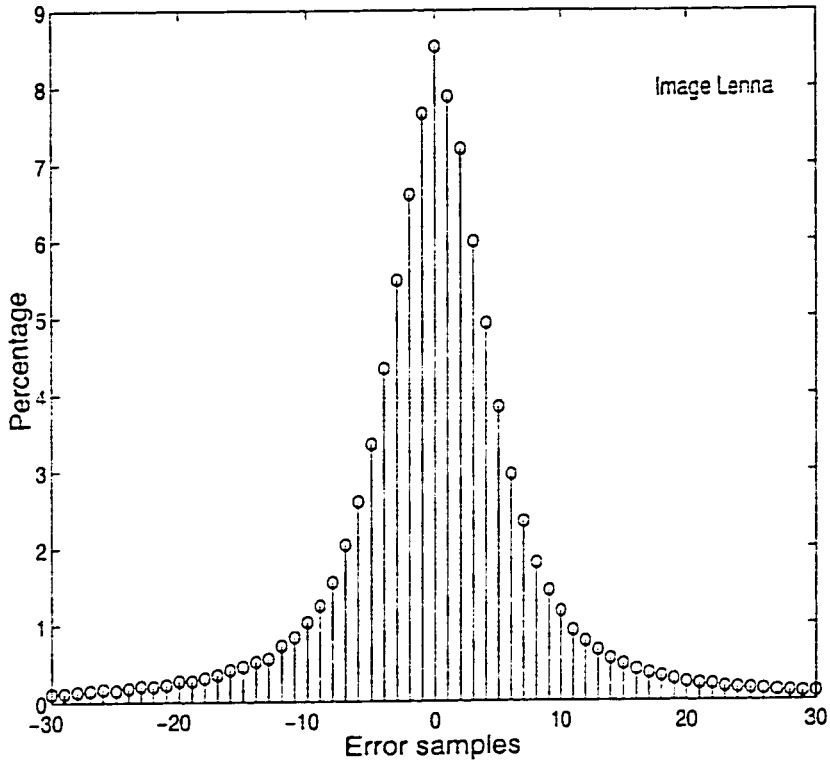

Figure 2.4: Distribution of error signal.

In general, for a linear prediction, a sample can be predicted as

$$\hat{x}(n) = \sum_{j=1}^{N} h_j x(n-j), \tag{2.1}$$

where h_j is the prediction coefficients. If the prediction gain is defined by

$$G_p = \sigma_x^2/\sigma_e^2, \tag{2.2}$$

it can be shown that

$$SNR|_{DPCM} = SNR|_{PCM} + 10\ log\ G_p, \tag{2.3}$$

and the reduction in bit rate for DPCM compared to PCM is given by

$$R|_{PCM} - R|_{DPCM} = \frac{1}{2}\log_2(\sigma_x^2/\sigma_e^2), \tag{2.4}$$

where σ_x^2 is the input variance and σ_e^2 is error signal variance.

For the first-order prediction, $\hat{x}(n) = h_1 x(n-1)$, the error signal is given by

$$e(n) = x(n) - h_1 x(n-1). \tag{2.5}$$

In order to have a maximum gain, the energy of $e(n)$ has to be minimized. It can be shown that for this purpose, the prediction coefficient h_1 has to be equal to the correlation coefficient ρ_1 defined by [4]

$$\rho_1 = \frac{E[X(n)X(n-1)]}{E[X^2(n)]}. \tag{2.6}$$

As a result, the gain for the first-order prediction is

$$G_{P_{max}} = \frac{1}{1-\rho_1^2}, \tag{2.7}$$

and the bit rate reduction is given by

$$R|_{PCM} - R|_{DPCM} = -\frac{1}{2}\log_2(1-\rho_1^2). \tag{2.8}$$

As an example, for $\rho = 0.97$, SNR of a 6-bit PCM can be achieved by a 4-bit DPCM.

For the second-order prediction, $\hat{x}(n)$ is defined by

$$\hat{x}(n) = h_1 x(n-1) + h_2 x(n-2). \tag{2.9}$$

In this case, the optimum prediction coefficients are given by [4]

$$h_{1_{opt}} = \rho_1(1-\rho_2)/(1-\rho_1^2) \tag{2.10}$$

and

$$h_{2_{opt}} = (\rho_2 - \rho_1^2)/(1-\rho_1^2), \tag{2.11}$$

where

$$\rho_2 = \frac{E[X(n)X(n-2)]}{E[X^2(n)]}. \tag{2.12}$$

2.2.1 Two-dimensional prediction

The idea of the DPCM can be extended to the two-dimensional space. In this case, a pixel can be predicted using its adjacent pixels in two dimensions, i.e.,

$$\hat{x}(m,n) = \sum_{(i,j)\in U} \sum h_{i,j} x(m-i, n-j), \tag{2.13}$$

where U is a two-dimensional prediction region and $h_{i,j}$'s are the prediction coefficients. Prediction can be causal or non-causal. Examples of causal and non-causal predictions are shown in Figure 2.5. In a causal prediction, the prediction of a sample depends only on the previous samples, but in a non-causal prediction, some future pixels also used in the prediction.

It has been shown that for typical images using more than four nearest pixels for the prediction of a sample is not useful and cannot increase the prediction gain [1], [24] . Thus, a sample in a two-dimensional DPCM can be predicted as

$$\hat{x}(m,n) = h_1 x(m-1,n-1) + h_2 x(m-1,n) + h_3 x(m,n-1) + h_4 x(m-1,n+1). \tag{2.14}$$

Maximizing the prediction gain requires the minimization of the error variance. Minimizing the error variance, in the special case of a separable correlation function, results in the following relations [1],

$$h_2 = \rho_v \qquad h_3 = \rho_h \qquad h_1 = -\rho_v \rho_h \qquad h_4 = 0 \tag{2.15}$$

where ρ_h and ρ_v are horizontal and vertical correlation coefficients as given by

$$\rho_v = R_{xx}(1,0)/\sigma^2 \qquad \rho_h = R_{xx}(0,1)/\sigma^2. \tag{2.16}$$

A separable model for covariance function is defined as

$$R_{xx}(m,n) = \sigma^2 \rho_v^{-|m|} \rho_h^{-|n|}. \tag{2.17}$$

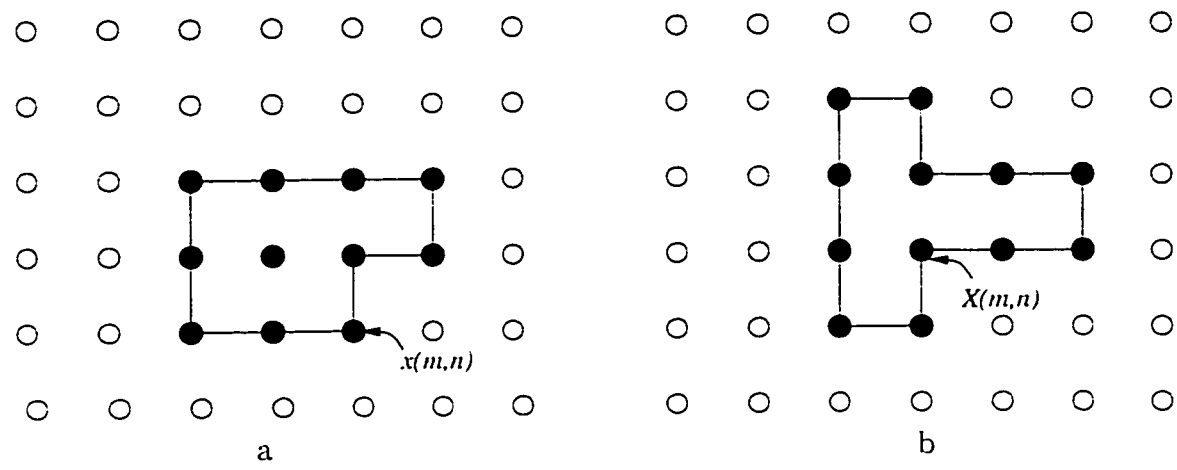

Figure 2.5: Examples of two-dimensional prediction. (a) Causal prediction. (b) Non causal prediction.

2.2.2 Drawbacks of DPCM

Although the DPCM is a simple scheme and results in a better performance compared to the PCM, three types of degradation are common in a DPCM quantizer design: granularity, slope overload and edge-busyness [4]. Granularity is because of the step-like nature of the output where the input signal is almost constant. Slope overload happens when there is a sharp change in the input signal (edges). In this case the quantized output cannot follow the input and a few steps are needed to match the output with the input. Edge-busyness is caused at less sharp edges when the input in the adjacent lines are quantized into different levels. Another drawback of the DPCM is its sensitivity to channel noise and image statistics. Adaptive techniques have been used to compensate these drawbacks.

2.3 TRANSFORM CODING

For a typical image, the correlation between the adjacent pixels is high. Transform coding uses this correlation in order to achieve a high compression ratio. To show

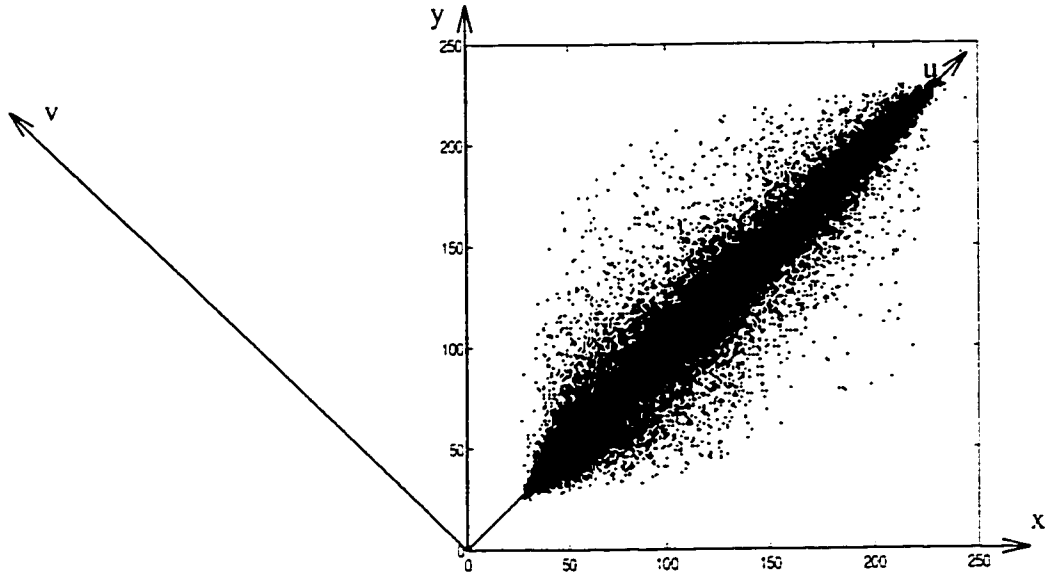

Figure 2.6: The correlation of adjacent pixels for the image *Lenna*.

this correlation, we group two consecutive pixels of image *Lenna* as a vector (x,y), and the dependency of y on x is presented in Figure 2.6. It can be seen that most of these points are concentrated near bisector y=x, as indicated dense area.

Quantizing any two consecutive samples independently results in an inefficiency, since the quantization levels for both dimensions are the same. For example, the quantizer allocates the same bit rate to the upper-left as to the dense area. However, the probability of a vector being in this area is very low. To improve the quantizer efficiency, after grouping the samples, the coordinate system can be rotated by a certain angle such that one of the axes is placed in the middle of the dense area as shown in Figure 2.6. In this case, more bits can be allocated to the u-axis and less to the v-axis. Hence, with the same average bit rate, better precision is achieved. After quantization and encoding, the inverse of this rotation is carried out in the decoder. The main idea of all image transformations in coding is to convert the original samples to new coefficients such that the new coefficients are less correlated than the original samples. Furthermore, these transformations have a

tendency to pack a large amount of energy into a few transform coefficients. The optimum transform which has the best "input-decorrelating" and "variance-ordering" properties is called the Karhunen Loeve Transform (KLT). The KLT completely decorrelates all pixels. The problem is that it depends on the statistics of the input samples, and it is hard to implement. Other transforms includes Discrete Fourier Transform (DFT), Discrete Walsh Hadamard Transform (DWHT), Discrete Sine Transform (DST) and Discrete Cosine Transform (DCT) [4], [1]. For image comparison, the DCT transform is very close to an optimum transform. Furthermore, it is signal-independent and it can be implemented using Fast Fourier Transform (FFT). As a result, it is the most popular transform used for image and video compression. Wavelet transform, which is a generalization of the conventional transforms has also been used for image compression [25].

In addition to the type of transform used, the bit allocation for the coefficients plays an important role in the performance of a compression scheme. There are many adaptive and non-adaptive methods for bit allocation. These includes optimum bit allocation, zonal sampling, threshold sampling and switched bit allocation [26], [27].

In general, a transform coding scheme has three major blocks: transformer, quantizer and lossless encoder. For an optimum bit allocation, the set of N transform coefficients usually needs N different quantizers. Figure 2.7 shows an image transform encoder and decoder.

Many efforts have been made for improving the quantizer and noiseless encoder. The Joint Photographic Expert Group (JPEG) [14] is a result of these efforts. The JPEG is accepted as a standard for compression techniques by the

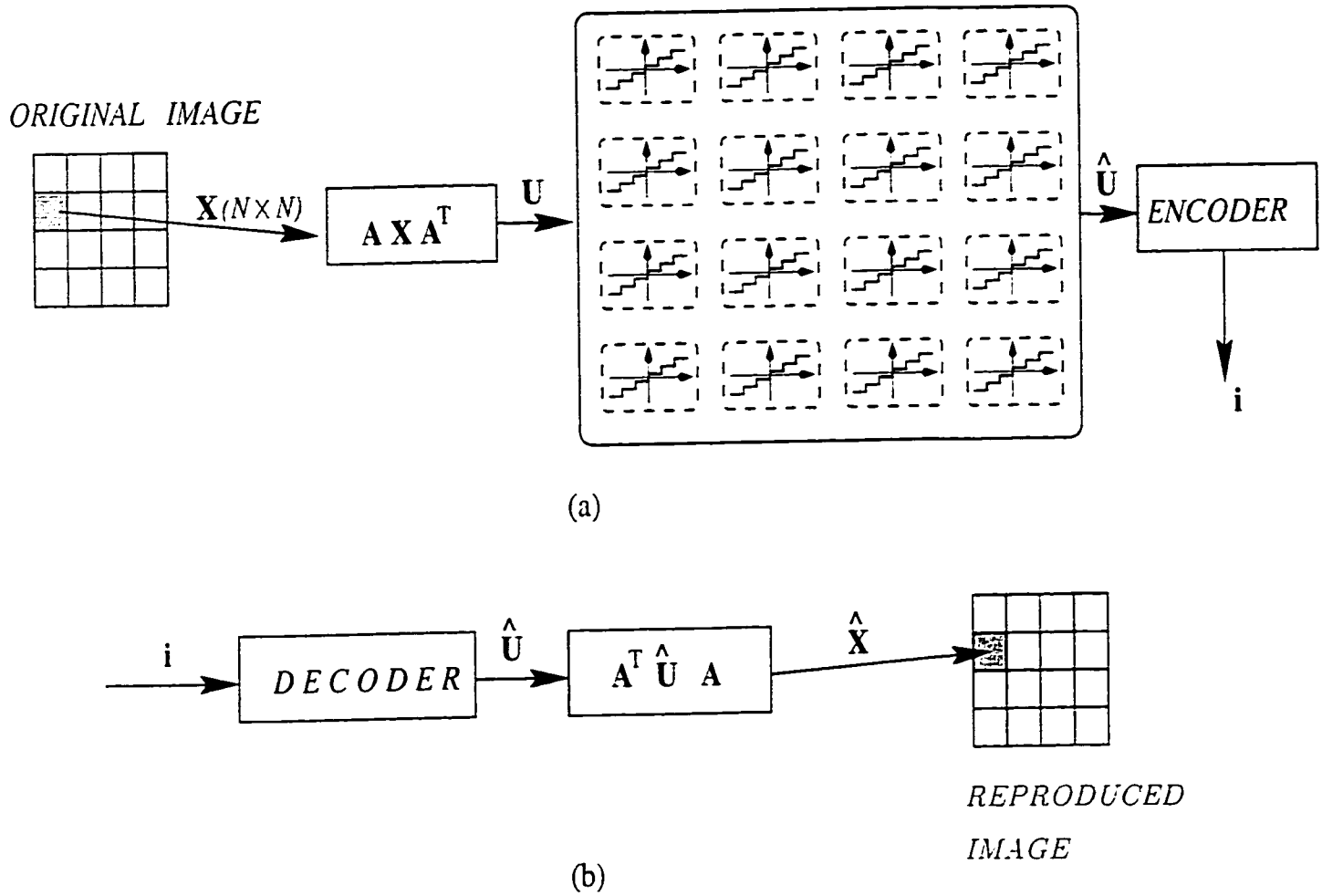

Figure 2.7: The block diagram of a transform coding. (a) Transform encoder. (b) Transform decoder.

International Telegraph and Telephone Consultative Committee (CCITT), International Organization for standardization (ISO) and International Electrotechnical Commission (IEG).

2.3.1 Linear Transform

For a one-dimensional sequence $\mathbf{X}^T = \{x(n) : 0 \leq n \leq N-1\}$, a transformation can be written as

$$\mathbf{U} = \mathbf{A}\,\mathbf{X}, \qquad u(k) = \sum_{n=0}^{N-1} a(k,n)x(n) \tag{2.18}$$

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{U}, \qquad x(n) = \sum_{k=0}^{N-1} u(k)a^*(k,n) \tag{2.19}$$

where $\mathbf{U}^T = \{u(n) : 0 \leq n \leq N-1\}$ are transform coefficients and

$$\mathbf{A}^T = \{\mathbf{a}(0), \mathbf{a}(1), ..., \mathbf{a}(N-1)\}^T$$

is an $N \times N$ transform matrix. $\mathbf{a}^*(k) = \{a^*(k,n), 0 \leq n \leq N-1\}$ are called the basis vectors.

For example, for $N = 2$, the transformation matrix $\mathbf{A}$, resulting from 45° rotation of the coordinate system, is given by

$$\mathbf{A} = \mathbf{A}^{-1} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.20}$$

The basis vectors are

$$\mathbf{a}(1)^T = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}], \tag{2.21}$$

$$\mathbf{a}(2)^T = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]. \tag{2.22}$$

For orthonormal transform, the transformation matrix satisfies the property $\mathbf{A}^{-1} = \mathbf{A}^T$. The basis vectors of some of the popular transformations are shown in Figure 2.8. Among these transformations, only the basis vectors of the KLT is defined by the statistics of the source.

Two-dimensional linear transform is defined by

$$u(l,k) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)a(k,l,m,n) \tag{2.23}$$

$$x(m,n) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} u(k,l)a^*(k,l,m,n) \tag{2.24}$$

If a separable and unitary transformation matrix is chosen, i.e., $a^*(k,l,m,n) = a_v(k,m)a_h(l,n)$, Eqn 2.23 becomes

$$u(k,l) = \sum_{m=0}^{N-1} a_v(k,m) \sum_{n=0}^{N-1} x(m,n)a_h(l,n) \tag{2.25}$$

where $\mathbf{a}_v$ and $\mathbf{a}_h$ are the column and row transform basis vectors. The above equation can be re-written as

$$\mathbf{U} = \mathbf{A_v X A_h^T} \tag{2.26}$$

In the case of symmetric kernels, $\mathbf{A_v} = \mathbf{A_h} = \mathbf{A}$ and the transformation equations can be written as

$$\mathbf{U} = \mathbf{AXA^T} \qquad u(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a(k,m)x(m,n)a(l,n) \tag{2.27}$$

$$\mathbf{X} = \mathbf{A^T UA} \qquad x(m,n) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} a^*(k,m)u(k,l)a^*(l,n) \tag{2.28}$$

As a result of this separable transform, the image X becomes to be a superposition of a series of representations for the image called the "basis images", as given by

$$\mathbf{X} = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} u(k,l)\mathbf{A_{k,l}} \tag{2.29}$$

$$\mathbf{A_{k,l}} = \mathbf{a}_k^* . \mathbf{a}_l^{*T} \tag{2.30}$$

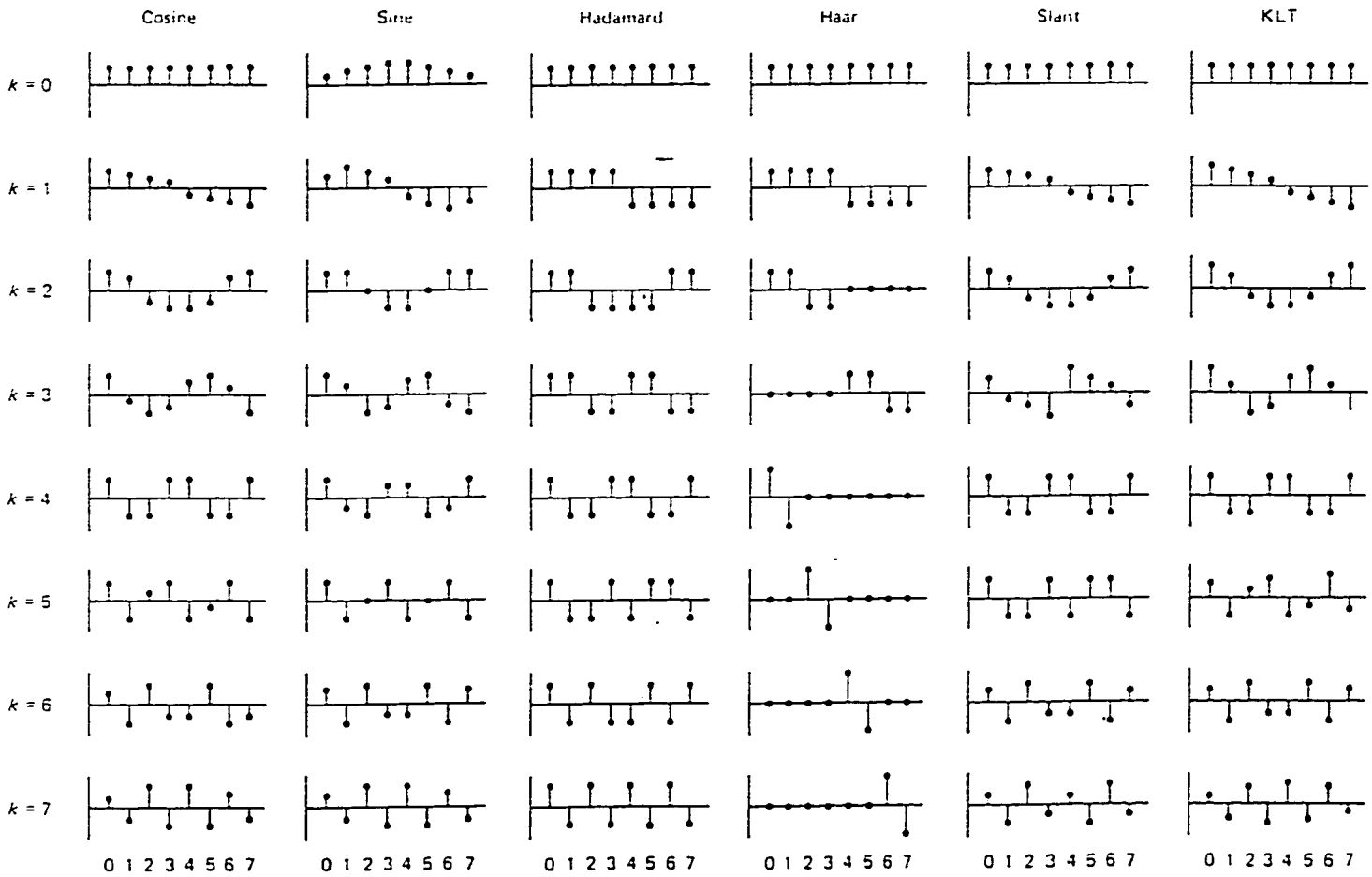

Figure 2.8: Basis vectors of some transforms with N=8, reproduced from [1].

where $\mathbf{a}_k$ is a vector corresponding to kth column of the matrix $\mathbf{A}$. In other words, each image is reconstructed by the superposition of the basis images weighted by the transform coefficients. Figure 2.9 shows these basis images for different transformations.

2.3.2 Some well-known transforms

Karhunen-Loeve Transform (KLT): The Karhunen-Loeve transform, which is also called eigenvector or Hotelling transform, is defined by the eigenvectors of the correlation matrix of the input samples. The correlation function is defined as

$$R_{xx}(k) = E[X(n)X(n-k)], \tag{2.31}$$

and the correlation matrix is given by

$$\mathbf{R}_{xx} = \{R_{xx}(|k-l|)\} \qquad k, l = 0, 1, ..., N-1. \tag{2.32}$$

The correlation matrix $\mathbf{R}_{xx}$ has a set of eigenvalues λ_i's and eigenvectors $\mathbf{I}'_i$'s defined by

$$\mathbf{R}_{xx}\mathbf{I}'_i = \lambda_i \mathbf{I}'_i. \tag{2.33}$$

Here $\mathbf{R}_{xx}$ is a real symmetric matrix, and thus its eigenvalues are real and there are exactly N eigenvectors which are orthogonal and can be normalized to form an orthonormal set $\mathbf{I}_i$, $i = 0, 1, ..., N-1$, that is,

$$\mathbf{I}_k^T . \mathbf{I}_l = \delta_{kl}. \tag{2.34}$$

The transform matrix of KLT is composed of eigenvectors of $\mathbf{R}_{xx}$. In other words, the basis vectors are eigenvectors of $\mathbf{R}_{xx}$, that is,

$$\mathbf{a}_k = \mathbf{I}_k, \tag{2.35}$$

thus (2.19) takes a form given by

$$\mathbf{X} = \mathbf{A}^\mathbf{T}\mathbf{U} = \sum_{k=0}^{N-1} u(k)\mathbf{I}_k, \tag{2.36}$$

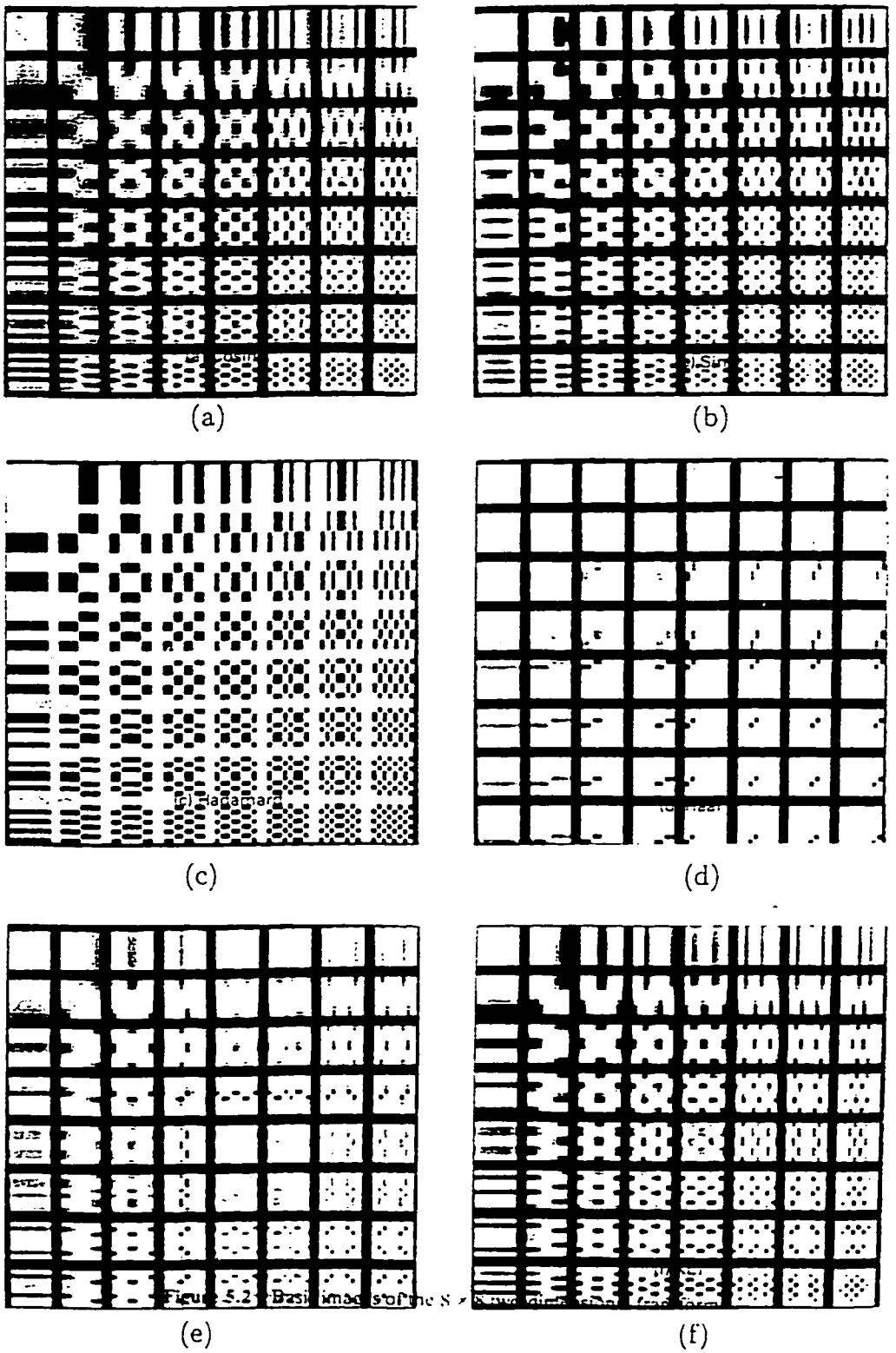

Figure 2.9: Basis images of some transforms, reproduced from [1]. (a) Cosine. (b) Sine. (c) Hadamard. (d) Haar. (e) Slant. (f) KLT.

implying that, the input signal X is a superposition of weighted eigenvectors which are derived from the correlation matrix of X. Therefore, the basis vectors depend on the statistics of the input samples. The correlation matrix of the transformed coefficients is a diagonal matrix with the eigenvalues of the correlation matrix of the input samples as its diagonal elements, that is

$$\mathbf{R}_{uu} = \begin{bmatrix} \lambda_0 & 0 & 0 & . & . & 0 \\ 0 & \lambda_1 & 0 & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & 0 & \lambda_{N-1} \end{bmatrix} \tag{2.37}$$

Discrete Hadamard Transform (DHT): In this transform, the transform matrix is constructed by a recursive operation, given by

$$\mathbf{U} = \mathbf{HX} \qquad {}^{2}\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad {}^{2N}\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} {}^{N}\mathbf{H} & {}^{N}\mathbf{H} \\ {}^{N}\mathbf{H} & -{}^{N}\mathbf{H} \end{bmatrix} \tag{2.38}$$

where ${}^{N}\mathbf{H}$ represents the transform matrix with a dimension of $N \times N$. The coefficient variance of this transform does not monotically decrease with the coefficient index. To have ordered coefficients, the ordering of rows in the transform matrix is changed. This new transform is called the Discrete Walsh Hadamard Transform (DWHT). To get the DWHT transform matrix, the DHT matrix is "sequency" ordered. The term "sequency" of a basis vector is defined by the number of sign changes in the vector. The concept of sequency for basis vectors is similar to the frequency in DCT or DFT. The transform matrix for a DHT and a DWHT for $N = 4$, for example, are given by

$$
{}^4\mathbf{H}(DHT) = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{matrix} 0 \\ 3 \\ 1 \\ 2 \end{matrix} \qquad {}^4\mathbf{H}(DWHT) = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \tag{2.39}
$$

the numbers beside the transform matrix are the sequency of the basis vectors. Although this transform is very easy to implement, it is not optimum, i.e. it does not diagonalize the covariance matrix.

There are some other fast transforms such as the Haar transform which is suitable for feature extraction, or the Slant transform which has a very good energy compaction property for images [1]. The basis vector and basis images of these transforms are shown in Figures 2.8 and 2.9, respectively.

Discrete Fourier Transform (DFT): The discrete Fourier transform of a sequence $\{x(n), n = 0, 1, ..., N-1\}$ is defined by

$$
u(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \qquad k = 0, 1, ..., N-1 \tag{2.40}
$$

$$
x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} u(k) e^{j2\pi kn/N} \qquad n = 0, 1, ..., N-1 \tag{2.41}
$$

Thus, the transform matrix of the DFT is given by

$$
\mathbf{F} = \left\{ \frac{1}{\sqrt{N}} e^{-j2\pi kn/N} \right\}_{k,n=0,1,...,N-1} . \tag{2.42}
$$

The most important point in the DFT is that it can be implemented using some fast methods called the Fast Fourier Transforms (FFT). With these methods the complexity of operation is reduced from N^2 to $(N \log N)$ [28]. The problem with the DFT is that it is not an optimal transformation, since it does not diagonalize the covariance matrix. In addition, the inverse DFT generates samples which are

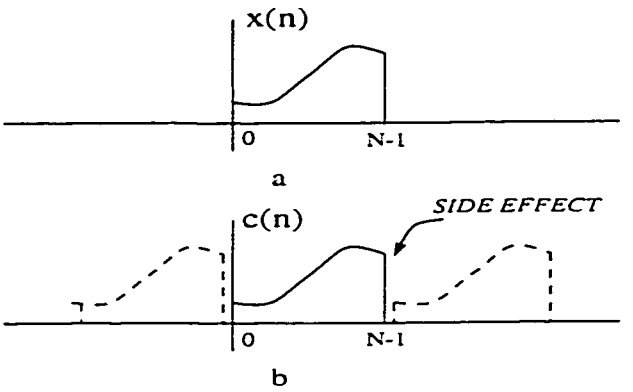

Figure 2.10: Side effect in DFT.

periodic extension of the first N samples, that is,

$$c(n) = x(n) \qquad n = 0, 1, ..., N-1 \tag{2.43}$$

and

$$c(n) \neq x(n) \qquad n = N, N+1, ... \tag{2.44}$$

$$c(n+N) = c(n) \tag{2.45}$$

This periodicity in DFT causes discontinuities at the beginning and end of each block. This effect can be seen in Figure 2.10.

Discrete Cosine Transform (DCT): Among the different transforms, the DCT has the decorrelation property very close to that of the KLT for most images. The discrete cosine transform is defined by

$$u(k) = \sqrt{\frac{2}{N}}\alpha(k) \sum_{n=0}^{N-1} x(n) \cos \frac{(2n+1)k\pi}{2N} \qquad k = 0, 1, ..., N-1 \tag{2.46}$$

$$\alpha(0) = \frac{1}{\sqrt{2}} \qquad \text{and} \qquad \alpha(k) = 1 \qquad k \neq 0$$

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \alpha(k)u(k) \cos \frac{(2n+1)k\pi}{2N} \qquad n = 0, 1, ..., N-1. \tag{2.47}$$

In the matrix form, this transformation can be written as

$$\mathbf{U} = \mathbf{CX} \qquad \mathbf{X} = \mathbf{C}^{-1}\mathbf{U}. \tag{2.48}$$

The DCT basis vectors can be obtained from

$$\mathbf{b}_k^T = \left\{ \sqrt{\frac{2}{N}}\alpha(k)\cos\frac{(2n+1)k\pi}{2N}, \quad n = 0, 1, ..., N-1 \right\} \qquad \text{for } k = 0, 1, ..., N-1 \tag{2.49}$$

It can be shown that the cosine transform is very close to the KLT for a first-order stationary Markov sequence when the correlation parameter ρ is close to 1. Because of strong correlation between adjacent pixels of a typical image, this transform is very close to the optimum transform for most images. The DCT has an excellent energy compaction property for highly correlated data. Further, it can be easily implemented using fast implementation methods. These properties make the DCT a popular transform for image coding. The energy distribution of DCT coefficients of a typical image, *Lenna*, is examined. The 8×8 DCT coefficients are scanned in a zig-zag order as shown in Figure 2.12, starting from the lowest to highest frequency. It is seen that most of energy is contained in low-frequency coefficients and energy, in general, decreases very rapidly as the frequency increased, as it can be seen in figure 2.11. The implementation of the DCT causes the input block of N-samples to extend into the blocks of 2N samples with an even symmetry, that is

$$c(n) = \begin{cases} x(n) & n = 0, 1, ..., N-1 \\ x(2N-1-n) & n = N, N+1, ..., 2N-1 \end{cases}. \tag{2.50}$$

This periodic extension has smaller end-effects than the DFT operation (Figure 2.13).

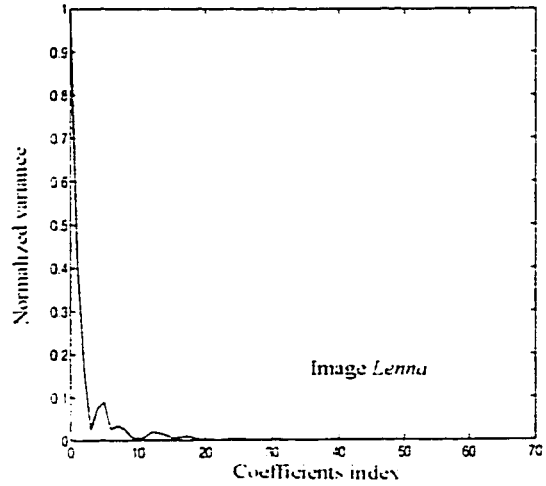

Figure 2.11: The energy distribution of DCT coefficients of image *Lenna*.

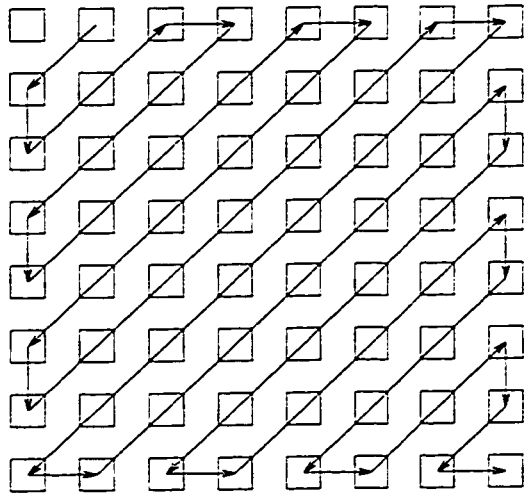

Figure 2.12: Zig-zag order of transform coefficients.

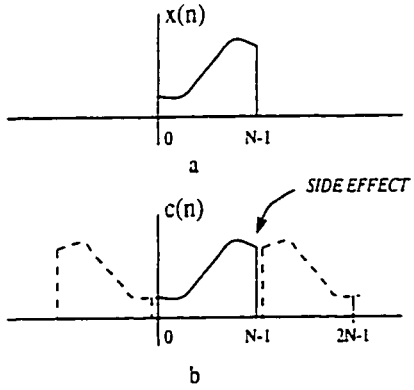

Figure 2.13: Side effect in DCT.

Two-dimensional DCT: Two-Dimensional DCT (2D-DCT) is defined by

$$u(k,l) = \frac{2}{N}\alpha(k)\alpha(l)\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)\cos\frac{\pi k(2m+1)}{2N}\cos\frac{\pi l(2n+1)}{2N} \tag{2.51}$$

$$x(m,n) = \frac{2}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} \alpha(k)\alpha(l)u(k,l)\cos\frac{\pi k(2m+1)}{2N}\cos\frac{\pi l(2n+1)}{2N} \tag{2.52}$$

$$\alpha(0) = \frac{1}{\sqrt{2}} \qquad \alpha(k) = 1 \quad k \neq 0 \tag{2.53}$$

where $x(m,n)$ is an $N \times N$ block and $k,l,m,n = 0,1,...,N-1$.

Figure 2.14 compares the KLT, DCT and DFT transform coding gains versus block length for a first-order Gauss-Markov source with $\rho = 0.95$. As the graph shows, for a Gauss-Markov source, the gain of DCT almost equals that of KLT, with a difference of less than 0.1 dB. In this figure G_p is the maximum gain achievable by a transform coding. It can be shown that $G_p = (1-\rho^2)$, and for $\rho = 0.95$, G_p is equal 10.11 dB. This is the upper limit for any transform. As it can be seen, DFT is asymptotically optimum.

Table 2.1 compares the SNR of different transforms for an image modeled by isotropic covariance function with $\rho = 0.95$. This table also shows that the DCT is very close to an optimum transform when the correlation of the adjacent pixels is high ($\rho = 0.95$). An isotropic or circularly symmetric function must satisfy the property

$$R_{xx}(m,n) = \sigma_x^2\rho^d \tag{2.54}$$

$$d = \sqrt{m^2+n^2} \qquad \rho = \exp(-|\alpha|) \qquad \text{when} \quad \alpha_1 = \alpha_2 = \alpha$$

where α_1 and α_2 are the correlation factor in the horizontal and vertical directions respectively.

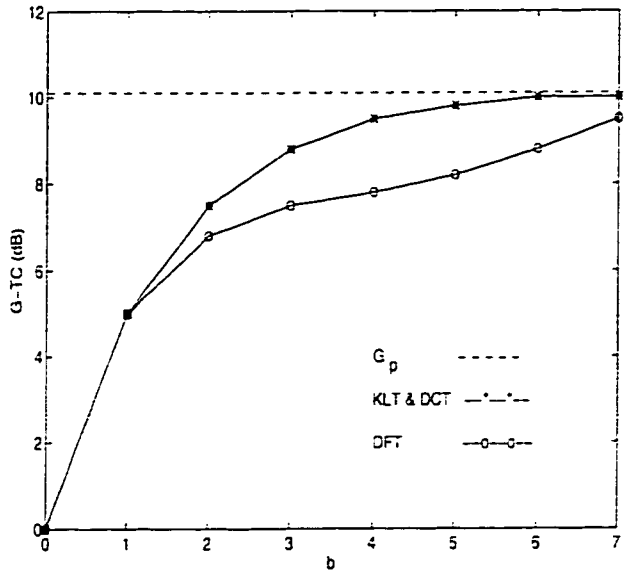

Figure 2.14: Transform coding gains versus $b = \log_2 N$ for a first-order Gauss-Markov source with $\rho = 0.95$.

Table 2.1: SNR comparison of various transform coders for random fields with isotropic covariance function $\rho = 0.95$, reproduced from [1]

Block size	Rate bits/pixels	SNR(dB)				
		KLT	DCT	DST	DFT	Hadamard
8 ×8	0.25	11.74	11.66	9.08	10.15	10.79
	0.5	13.82	13.76	11.69	12.27	12.65
	1.00	16.24	16.19	14.82	14.99	15.17
	2.00	20.95	20.80	19.53	19.73	19.86
	4.00	31.61	31.54	30.17	30.44	30.49
16 × 16	0.25		12.35	10.37	10.77	10.99
	0.5		14.25	12.82	12.87	12.78
	1.00		16.58	15.65	15.52	15.27
	2.00		21.26	20.37	20.24	20.01
	4.00		31.9	31.00	30.88	30.69

2.3.3 Bit allocation in transform coding

After having chosen a suitable transform, the next step is to allocate bits to different coefficients. Since the variances of different coefficients are not equal, they need different number of bits. A given number of bits should be distributed between coefficients such that the overall distortion is minimized. All orthogonal transforms preserve the variance. To show this, consider a source with variance σ_x^2 and its transform coefficients with variances σ_k^2. The total energy of the coefficients can be expressed as

$$\begin{aligned}
\tfrac{1}{N}\textstyle\sum_{k=0}^{N-1}\sigma_k^2 &= \tfrac{1}{N}\textstyle\sum_{k=o}^{N-1}E[u^2(k)] \\
&= \tfrac{1}{N}\mathbf{E}[\mathbf{U^T U}] \\
&= \tfrac{1}{N}\mathbf{E}[\mathbf{X^T A^T A X}] \\
&= \tfrac{1}{N}\mathbf{E}[\mathbf{X^T X}] \\
&= \tfrac{1}{N}\textstyle\sum_{k=0}^{N-1}E[X^2(k)] \\
&= \tfrac{1}{N}\textstyle\sum_{k=0}^{N-1}\sigma_x^2(k) \\
&= \sigma_x^2
\end{aligned} \tag{2.55}$$

Furthermore, for orthogonal transforms, the reconstruction error variance in transform coding equals that introduced by the set of quantized coefficients, as given by

$$\sigma_r^2 = \sigma_q^2 = \frac{1}{N}\sum_{k=0}^{N-1}\sigma_{qk}^2 \tag{2.56}$$

where σ_r^2 is the reconstructed error variance, σ_q^2 is the quantization error variance and σ_{qk}^2 is the variance of the quantization error of the kth coefficients. After finding

an orthogonal transform matrix, the problem is to minimize σ_q^2 with the constraint of a given average bit rate defined by

$$R = \frac{1}{N}\sum_{k=0}^{N-1} R_k \tag{2.57}$$

where R_k is the bit rate for the kth coefficient. This requires solving the following equation which is obtained from using the Lagrange multiplier method,

$$\frac{\partial}{\partial R_k}[\sigma_q^2 - \lambda(R - \frac{1}{N}\sum_{k=0}^{N-1} R_k)] = 0 \qquad k = 0, 1, ..., N-1 \tag{2.58}$$

An optimum bit allocation is thus achieved as

$$R_k = R + \frac{1}{2}\log_2 \frac{\sigma_k^2}{\left[\prod_{j=0}^{N-1}\sigma_j^2\right]^{1/N}}. \tag{2.59}$$

Optimum bit allocation for each coefficient depends on the distribution of coefficient variances. For example, for N=2, the bit allocation is given by

$$R_0 = R + \frac{1}{2}\log_2 \frac{\sigma_0}{\sigma_1}, \tag{2.60}$$

$$R_1 = R - \frac{1}{2}\log_2 \frac{\sigma_0}{\sigma_1}. \tag{2.61}$$

In the case of equal variance , we have $R_k = R$ for $k = 0, 1, ..., N-1$. The dependency of bit allocation on the variance is illustrated in Figure 2.15. For a uniform quantizer to have the equal quantization error variance, all quantizers have to have equal step size. Since the dynamic ranges of the coefficients are different, the coefficient with a higher variance needs more quantization levels than a coefficient with a lower variance. For example, in Figure 2.15, the coefficient A can be quantized by 16 levels or 4 bits; however, for the coefficient B, 2 bits are enough.

For practical considerations, the second term in (2.59), for the small values of σ_k, could be negative with a magnitude greater than R. This will cause a negative

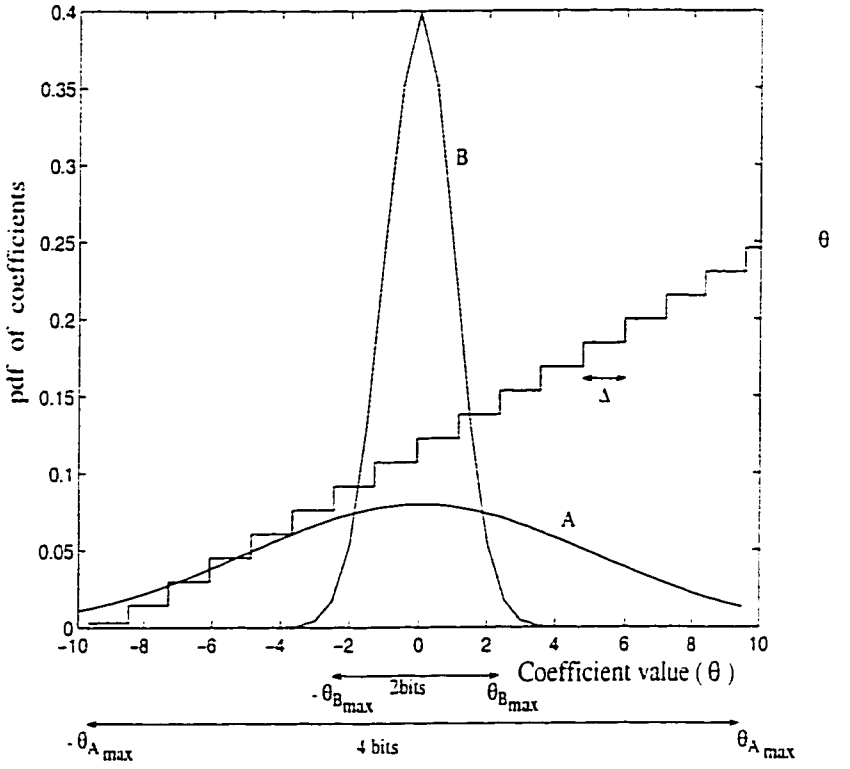

Figure 2.15: The dependency of bit allocation on variance.

$$\begin{bmatrix}
8 & 7 & 6 & 5 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
7 & 6 & 5 & 4 & 3 & 3 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
6 & 5 & 4 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
5 & 4 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
3 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
3 & 3 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Figure 2.16: Bit allocation for 16 ×16 DCT of image modeled by an isotropic covariance function with $\rho = 0.95$ with an average bit rate of 1 bps, reproduced from [1].

bit allocation. To avoid negative values of R_k, the equation can be modified as

$$R_k = max\{0, R + \frac{1}{2}\log_2 \frac{\sigma_k}{D}\} \tag{2.62}$$

where D is the geometrical average of the variances of the coefficients. Figure 2.16 demonstrates the application of this bit allocation method to a 16 × 16 block DCT coding of an image modeled by an isotropic covariance function with $\rho = 0.95$ and an average bit rate of 1 bit per sample.

Zonal coding: Figure 2.16 shows that only a small zone of the transformed image contains elements with non-negligible values. This is the main idea behind zonal coding. In zonal coding the coefficients with the index less than a specified value are retained and the rest are set to zero. In other words, the coefficients are masked

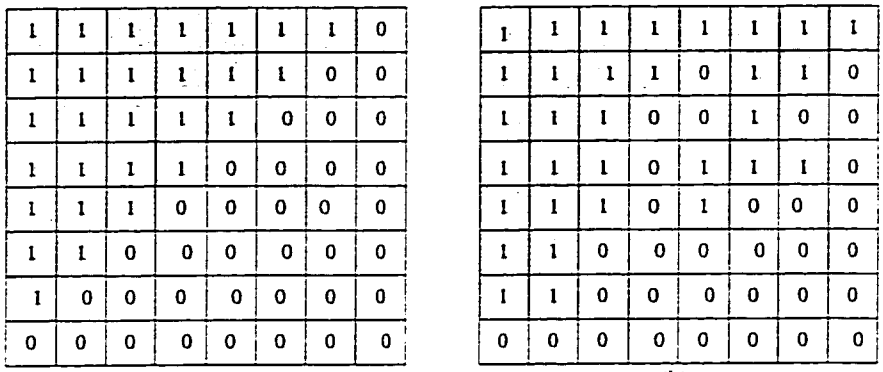

Figure 2.17: A typical mask for (a) Zonal coding. (b) Threshold coding.

with a zonal mask defined by

$$m(k,l) = \begin{cases} 1 & k < K \ , \ l < L \\ 0 & otherwise \end{cases} \tag{2.63}$$

Figure 2.17(a) shows a typical mask for zonal coding.

Threshold coding: In threshold coding the variance of the coefficients rather than their indices are considered for masking. The mask for a threshold coding is defined by

$$m(k,l) = \begin{cases} 1 & \sigma^2(k,l) > \eta \\ 0 & otherwise \end{cases} \tag{2.64}$$

The threshold η is chosen to get a desirable bit rate. Figure 2.17(b) shows a typical mask for the threshold coding.

2.3.4 Image transform coding

The closeness of the DCT to the optimum transform makes it the popular transform for image coding. In the DCT, the first coefficient is the dc coefficient and remaining coefficients are ac coefficients. Usually, the dc coefficient of the transform coding is coded separately using the DPCM. As shown in Figure 2.18, encoding the ac coefficients involves two steps: first, quantizing and then, indexing the output points of

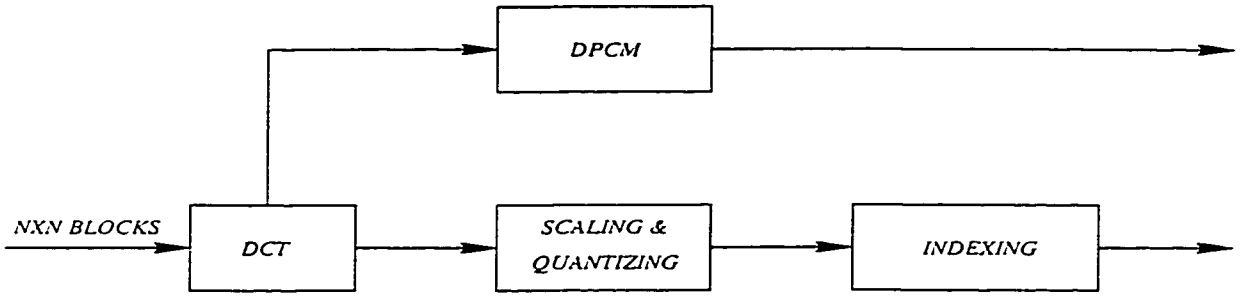

Figure 2.18: The block diagram of transform coding.

the quantizer. For the first step, many lossy scalar and vector quantizer techniques have been designed. Depending on the quantizer, different noiseless coding schemes have been used to index the output points of the quantizer.

JPEG [14] partitions each image into 8×8 blocks. DCT is computed over these blocks. After the transformation, the DCT coefficients are scaled and truncated in order to reduce the dynamic range of the data. The scaled DCT coefficients are ordered into a zig-zag sequence. The non-zero amplitudes of this one-dimensional sequence and the runlength of zeros are entropy coded.

Due to the regular structure of lattices, many researchers have used the Lattice-Based Vector Quantizer (LBVQ) for quantizing the DCT coefficients , but only a few methods have been suggested for indexing the output points [15]. Fischer [29] has combined an lattice-based vector quantizer with a noiseless code to encode the DCT coefficients of images. The output lattice points are labeled by using an enumeration method for a Laplacian source, and it is shown that the combination of the LBVQ and noiseless code outperforms the uniform scalar quantizer combined with noiseless coding for each coefficient.

2.4 PRINCIPLES OF JPEG STANDARD

Like any transform coding scheme, the block diagram of JPEG scheme consists of two basic blocks, a DCT based compression followed by a lossless variable length coding (a special case of Huffman coding). Each 8×8 block of input goes through the processing steps giving a stream of compressed data at the output. In the first step, each block is converted into 64 DCT coefficients whose values are uniquely determined by the 64 input pixels. The DCT coefficients are quantized by a set of uniform scalar quantizers defined in a quantization table. The goal of this step is to omit information which is not visually important. Several quantization tables have been defined and the quality of a coded image is controlled by these tables.

After quantization, the scaled DCT coefficients are coded. In this step, because of the strong correlation between the DC coefficients of successive blocks, they are coded differentially (Differential Pulse Code Modulation DPCM). The AC coefficients are scanned in a zig-zag sequence, as shown in Figure 2.12. This ordering places the low-frequency coefficients before the high-frequency coefficients which are usually zero (after scaling). The last step in JPEG is entropy coding. Two entropy coding schemes is used in JPEG: Huffman coding [2] or arithmetic coding [3]. After scaling there are only a few non-zero elements in the quantized AC coefficients. Each of these non-zero elements is represented in combination with runlength, the number of consecutive zero-valued coefficients which precede the non-zero coefficients. Two symbols are used to show the combination of runlength and non-zero-coefficients. The first symbol represents two pieces of information, the runlength and the size. The second symbol represents the amplitude of the non-zero coefficients. The runlength is the number of consecutive zero-valued AC coefficients and the size is the number of bits used to encode the amplitude of a non-zero coefficient. A special codeword is generated for the End Of Block, symbol EOB, which is viewed as the terminator of an 8×8 sample block. For the DC coefficients, two symbols are also

used. However, the first symbol represents only the number of bits used to encode the amplitude of the DC coefficient, size of the symbol. The second symbol represents the amplitude of the difference signal. Finally, these symbols are encoded using a variable-length code.

2.5 VECTOR QUANTIZATION

A fundamental result of Shannon's rate-distortion theory [6], a branch of information theory devoted to data compression, is that a better performance can be achieved by coding vectors instead of scalars. This holds even if the data source is memoryless, i.e., the sequence of source samples are independent. However, a greater performance improvement can be achieved if the source samples are correlated. It has been proven that vector quantizer is asymptotically the optimal structure for source coding when the vector dimension tends to infinity [7]. Before 1980, this theory had a limited impact on system design, because it did not provide constructive design techniques for encoders. After the publication of the paper by Linde et al. [30], in which the Lloyd algorithm [31], an algorithm for the design of an optimal scalar quantizer, was generalized to vector space, vector quantization gained popularity.

Formally, a Vector Quantizer (VQ) can be defined as a mapping Q(.) of the N-dimensional Euclidean space $\Re^N$ into a finite subset Y of $\Re^N$, i.e.

$$Q(\mathbf{x}) : \Re^N \rightarrow Y \tag{2.65}$$

where $\mathbf{x} = \{x_1, x_2, ..., x_N\} \in \Re^N$ is an input vector and $Y = \{\mathbf{y}_i \in \Re^N; i = 1, 2, ..., M\}$ is the codebook, and its elements $\{\mathbf{y}_i\}$ are called code-vectors or reproduction vectors. Vector quantization is a combination of two functions, an encoding and a decoding. The encoder receives the N-dimensional input vector $\mathbf{x}$ and searches through the codebook to find the address of a reproduction vector $\hat{\mathbf{x}} = \mathbf{y}_i$, which is closest to the input vector. The index of this reproduction vector is transmitted and

the decoder uses this address to look up for the reproduction vector. For choosing the address of $\hat{\mathbf{x}}$, a distortion measure $d(\mathbf{x}, \hat{\mathbf{x}})$ has to be defined. It represents the penalty associated with reproducing $\mathbf{x}$ by $\hat{\mathbf{x}}$. One simple distortion measure is the Mean Squared Error (MSE), defined as

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \| \mathbf{x} - \hat{\mathbf{x}} \|^2 = \sum_{i=0}^{N-1} (x_i - \hat{x}_i)^2 \tag{2.66}$$

An optimum VQ is a quantizer which generates the reproduction vectors, minimizing the expected distortion, defined as

$$D = E\{d(\mathbf{x}, \hat{\mathbf{x}})\} \tag{2.67}$$

The optimum VQ may be generated from the training images using the clustering technique introduced by Linde et al. in 1980 [30]. This technique, called the Generalized Lloyd Algorithm (GLA), is a generalization of the Lloyd's scalar algorithm [31] to vector space. This technique begins with an initial codebook and an iteration process comprising the following two steps yields an optimum codebook. The first step is to encode the training sequence and to calculate the average distortion. In the next step, each codeword is replaced by the centroid of the input vectors encoded into it. The size of a VQ codebook is usually a power of 2, i.e., $M = 2^b$, so that the index of reproduction vector can be represented using b bits.

The initial codebook in the GLA algorithm is very important, because regarding the initial codebook the method results in different locally optimum codebook. One of the method to generate the initial codebook is the spiliting method. It starts with the average of training sequence and then with a small change in the average and using the GLA algorithm constructs a codebook with size 2. In the same way with a small changes in the code-vectors a double size codebook is constructed until a desirable size codebook is achieved. The codebook is used as an initial codebook for GLA algorithm.

The encoding complexity for an optimum VQ, where each input vector is compared with all the vectors in the codebook, increases exponentially with the rate R and dimension N. This complexity and the memory requirement of an optimum VQ can be greatly reduced by imposing a structure on the codebook. Several schemes have been proposed for reducing the complexity of a full search VQ. These methods include Tree-Search Vector Quantizer (TSVQ) [8], Lattice-Based Vector Quantizer (LBVQ) [11] and Finite-State Vector Quantizer (FSVQ).

In addition, techniques such as classified VQ have also been proposed to match the codebook to certain properties of the source in order to improve the performanc. Many other efforts have been made to improve vector quantization techniques. These include adaptive VQ and variable-dimension VQ. Some techniques such as gain/shape VQ [11], predictive VQ and transform coding do some preprocessing on the input vector before encoding. Gray [5] has presented a good review of these techniques.

2.5.1 Lattice-based vector quantizer

Because of the regular structure of the lattice-based VQ, its use can result in a drastic reduction in the complexity in comparison to the GLA algorithm for the same rate and dimension. It is optimum for uniformly distributed sources. Hence, it may not give a good performance for other sources. Some works [19], [21] have been reported in which the lattice-based VQ is used for Gaussian and Laplacian sources. Jeong and Gibson [19], [32] have used a lattice-based VQ to encode the 2D-DCT coefficients of images.

A lattice is an infinite regular array that covers N-dimensional space uniformly. A lattice can be defined as a set of vectors

$$\Lambda = \{\lambda : \lambda = u_1\mathbf{a}_1 + u_2\mathbf{a}_2 + \cdots + u_N\mathbf{a}_N\} \tag{2.68}$$

where $\{\mathbf{a}_i : i = 1, 2, ..., N\}$ is the set of basis vectors of the lattice and u_i's are integers. Matrix $\mathbf{G}$ with its rows composed of the basis vectors $\mathbf{a}_i$'s is called the generator matrix. The determinant of the lattice Λ is defined as,

$$det\Lambda = |det(\mathbf{G}.\mathbf{G}^T)|^{1/2} \tag{2.69}$$

If G is a square matrix then $det\Lambda = det\mathbf{G}$.

Any N-dimensional lattice Λ has a dual lattice Λ^*, given by

$$\Lambda^* = \{\lambda^* \in \Re^N : \lambda.\lambda^* \in \mathbf{Z} \quad \forall\lambda \in \Lambda\}, \tag{2.70}$$

where (.) is the inner product , and $\mathbf{Z}$ is the set of integer. Voronoi region, $R_v(\Lambda)$, is the set of points x in N-dimensional space that are closer to the origin than to any other lattice point, i.e.,

$$R_v(\Lambda) = \{\mathbf{x} :\| \mathbf{x} \|^2 \leq \| \mathbf{x} - \lambda \|^2 \quad \forall\lambda \in \Lambda\}. \tag{2.71}$$

The fundamental volume of Λ, $v(\Lambda)$, is the volume of its voronoi region $R_v(\Lambda)$. The determinant of a lattice determines the volume of its voronoi region.

Lattices frequently used in image coding include the cubic lattice Z^N and the root lattices A_N, D_N, E_N [32]. These lattices can be defined using their generator matrices.

The lattice-based VQ encodes the source vectors by mapping them into the lattice points.

$$LQ(\mathbf{x}) : \mathbf{x} \rightarrow \mathbf{y_i} \in \Lambda \quad \text{if } \mathbf{x} \in R_{v_i}(\Lambda) \tag{2.72}$$

where $R_{v_i}(\Lambda)$ is the voronoi region when the origin is translated to $\mathbf{y}_i$.

Using a lattice as a codebook involves three steps: truncating, scaling, and arranging the lattice points which are outside of the truncated region. For a given

dimension N and bit rate R, 2^{NR} is the number of lattice points used. The lattice is truncated in such a way that the desired number of output points fall inside the boundary. Hence, in truncating lattice points, two parameters should be defined: the shape of the boundary and the radial parameter. For minimum distortion, the shape of the boundary is the shape of the contour of constant probability density function (pdf) [21]. This contour is spherical for Gaussian source and a pyramid for Laplacian source. To determine the radial parameter, we can use the *theta function* of a lattice, which specifies the number of lattice points at a certain distance from the origin [20]. Theta function of the lattice Λ is defined as

$$\Theta_{\Lambda}(z) = \sum_{\Lambda} q^{\mathbf{x}\cdot\mathbf{x}} = \sum_{m=0}^{\infty} n(m) q^{m} \tag{2.73}$$

where $n(m)$ is the number of lattice vectors with norm squared m (i.e., the number of lattice points at a distance m from the origin), z is a real number and $q = e^{\pi i z}$. Conway and Sloane have investigated the theta functions of several lattices [20]. Theta functions of some lattices can be expressed in terms of the Jacobi theta functions. For example, consider the Jacobi theta function $\theta_3(z)$ which is useful for expressing the theta function of cubic lattice:

$$\theta_3(z) = \sum_{m=-\infty}^{\infty} q^{m^2} = 1 + 2q + 2q^4 + 2q^9 + 2q^{16} + \ldots \tag{2.74}$$

In this case, the theta function is given by

$$\Theta_{Z^N}(z) = [\theta_3(z)]^N \tag{2.75}$$

For some dimensions, explicit expressions for defining the coefficients of q^m in the theta function are obtained [20]. For cubic lattice, the expressions for dimension 2,4,8, for example, for these coefficients are given by:

$$n_2(m) = 4 \sum_{i=1}^{\lfloor \frac{(m+1)}{2} \rfloor} (-1)^{i+1} \lfloor \frac{m}{2i-1} \rfloor,$$

$$n_4(m) = \begin{cases} 8\sum_{d|m} d & \text{odd m} \\ 24\sum_{d|m,\text{odd d}} d & \text{even m} \end{cases},$$

$$n_8(m) = 16\sum_{d|m}(-1)^{m-d}d^3,$$

where $n_N(m)$ is the coefficient of q^m for dimension N, $\lfloor x \rfloor$ means the greatest integer less than or equal x, and $\sum_{d|m} d$ represents the summation of these integers from 1 to m that can divide m. For $N = 16$, no explicit formula is known, and the direct expansion of $(\theta_3(z))^{16}$ is used to compute $n_{16}(m)$.

The truncated lattice points must be scaled to achieve minimum distortion. The best scaling is found by repeated experiments. Although, this method is not a precise procedure, it is the best method when the pdf of the source is unknown. Jeong and Gibson [19] have developed an analytical solution for the scaling of independent, identically distributed, i.i.d., Gaussian and Laplacian sources.

For finding the nearest lattice point, a fast quantization algorithm has been devised by Conway and Sloane [33]. This method is appropriate for root lattices A_n, D_n, E_n and their duals. First, for a real number x, they have defined a simple function $f(x)$ having an integral value closest to x. In the case of a tie, the integer with the smallest absolute value is chosen. For a vector $\mathbf{x} = (x_1, x_2, ...x_N) \in R^N$, $f(\mathbf{x})$ is defined as

$$f(\mathbf{x}) = (f(x_1), f(x_2), ..., f(x_N)). \tag{2.76}$$

A function $g(\mathbf{x})$ is defined in the same manner as $f(\mathbf{x})$ except that the worst component of $\mathbf{x}$, that is the element of x that is farthest from its corresponding integer, is rounded the wrong way. If $w(x)$ denotes a real number x rounded the wrong way,

then

$$w(x) = \begin{cases} k+1 & \text{if } k \le x \le k+\frac{1}{2} \\ k & \text{if } k+\frac{1}{2} < x < k+1 \\ -k-1 & \text{if } -k-\frac{1}{2} \le x \le -k \\ -k & \text{if } -k-1 < x < -k-\frac{1}{2} \end{cases} \tag{2.77}$$

where k is a non-negative integer. The nearest lattice point to a point in $\Re^N$ is found using $f(\mathbf{x})$ and $g(\mathbf{x})$. For example, for a cubic lattice, $f(\mathbf{x})$ is the nearest lattice point to $\mathbf{x} \in \Re^N$; however, quantizing with lattice D_N requires calculating $f(x)$ and g(x) and choosing the one with an even coordinate sum [34].

The last step in the lattice-based quantization is encoding the input points which fall outside the truncated region. These points are reflected on the contour surface along their radial line. The nearest lattice point which lies inside the lattice region is selected as the output.

2.6 ENTROPY CODING

A discrete-amplitude source is a source taking values from a finite set, i.e., $x(n) \in \mathcal{X} = \{x_1, x_2, ..., x_K\}$. The source alphabet $\mathcal{X}$ is associated with a set of probabilities $\{p_1, p_2, ..., p_K\}$, where $p_i = Pr.\{X(n) = x_i\} = p(x_i)$, $x_i \in \mathcal{X}$. The source is called a memoryless source if its samples are statistically independent.

The entropy of a discrete random variable X is defined by

$$H(X) = -\sum_{k=1}^{K} p(x_k) \log p(x_k) = \mathbf{E}[\log \frac{1}{p(X)}]. \tag{2.78}$$

If a base-2 logarithm is taken, the entropy is expressed in bits. Entropy is a positive number with the following boundaries:

$$0 \le H(X) \le \log_2 K. \tag{2.79}$$

An entropy $H(X) = 0$ means that there is no uncertainty and the source is totally predictable. This condition happens only if all the source alphabet values have the probability of zero except one of them. An entropy $H(X) = \log_2 K$ corresponds to the case when all probabilities are equal.

If there is an statistical dependency between the samples, the source has memory. To take advantage of this dependency, N successive samples $(x(n), x(n+1), ..., x(n+N-1))$ are arranged in a block designated as vector $\mathbf{X}$. The probability of a specific block is $p(\mathbf{x})$, and the entropy per symbol of this vector is given by

$$\begin{aligned} H_N(\mathbf{X}) &= \tfrac{1}{N}\mathbf{E}[-\log_2 p(\mathbf{X})] \\ &= -\tfrac{1}{N}\sum\sum_{all\ \mathbf{x}}\cdots\sum p(\mathbf{x})\log_2 \mathbf{x} \\ H(X) &= \lim_{N\to\infty} H_N(\mathbf{X}), \end{aligned} \tag{2.80}$$

and for a memoryless source

$$H_N(\mathbf{X}) = H(X). \tag{2.81}$$

2.6.1 The Asymptotic Equipartition Property

The weak law of large numbers [35] states that for independent, identically distributed (i.i.d.) random variables, $\frac{1}{n}\sum_{i=1}^{n} x_i$ is close to the expected value of X for large values of n. The law of large number in information theory is the Asymptotic Equipartition Property (AEP). This property is formalized in the following theorem [2].

AEP Theorem: If $X_1, X_2, ...$ are i.i.d. with the probability of observing $p(x)$, then in probability

$$-\frac{1}{N}\log p(X_1, X_2, ..., X_N) \to H(X). \tag{2.82}$$

This theorem suggests dividing each sequence into two sets, the typical set and the non-typical set.

Definition: The typical set $A_\epsilon^{(N)}$ with respect to $p(x)$ is the set of sequences

$(X_1, X_2, ..., X_N) \in \mathcal{X}^N$ with the following property

$$2^{-N(H(X)+\epsilon)} \leq p(x_1, x_2, ..., x_N) \leq 2^{-N(H(X)-\epsilon)} \tag{2.83}$$

This set has the following properties:

1. If $(x_1, x_2, ..., x_N) \in A_\epsilon^{(N)}$, then

$$H(X) - \epsilon \leq -\frac{1}{N} \log p(x_1, x_2, ..., x_N) \leq H(X) + \epsilon \tag{2.84}$$

2. $p(A_\epsilon^{(N)}) \geq 1 - \epsilon$ for sufficiently large N.

3. $|A_\epsilon^{(N)}| \leq 2^{-N(H(X)+\epsilon)}$, where $|A|$ denotes the number of elements in the set A.

4. $|A_\epsilon^{(N)}| \geq (1 - \epsilon)2^{(H(X)-\epsilon)}$ for sufficiently large N .

Hence, a typical set has a probability close to 1, and all of its elements are nearly equiprobable with the probability $2^{-NH(X)}$. Figure 2.19 shows the typical and non-typical sets. If this set is found then a special code can be defined. All elements in the typical set can be coded using $N(H + \epsilon) + 1$ bits and all elements of the non-typical set can be expressed using $\log_2 |\mathcal{X}^N| = N \log_2 |\mathcal{X}|$. We can use one prefix bit to show whether or not the vector belongs to the typical set. For example, a 0 as the first bit indicates that the code belongs to the typical set and the code length is $N(H + \epsilon) + 1$. On the other hand, a 1 as the first bit shows that the code belongs to the non-typical set and the code length is longer.

2.7 RATE DISTORTION FUNCTION

The rate distortion function, $R(D)$, specifies the minimum rate at which one must receive the information about the source output in order to be able to reproduce it with an average distortion that does not exceed a given D. To find an expression for $R(D)$, first the notation is defined, then a brief introduction to information theory

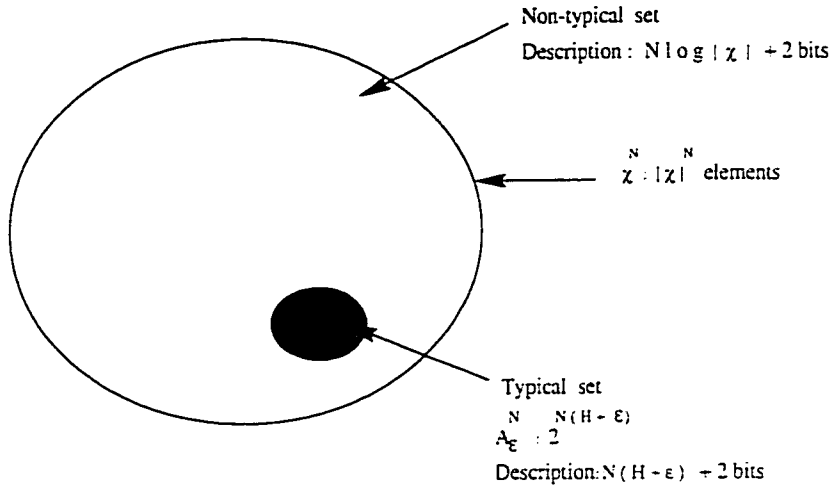

Figure 2.19: Typical and non-typical sets.

is presented [36].

A source with alphabet $\mathcal{X} = \{x_1, x_2, ..., x_K\}$, and a set of the associated probabilities $P = \{p_1, p_2, ..., P_K\}$, is denoted as $(\mathcal{X}, P)$. For convenience, the random variable $X(.)$ and the probability $P(X(.))$ are denoted as

$$X(j) = j \qquad \text{and} \qquad P(X(j)) = P(j) = P_j.$$

With these notational definitions, the entropy of a discrete random variable can be written as

$$H(X) = \mathbf{E}[\log \frac{1}{P_j}] = -\sum_{j=1}^{K} P_j \log P_j \tag{2.85}$$

where H(X) is the average uncertainty as to value X will assume. Let $\mathcal{X}$ and $\mathcal{Y}$ be two alphabets and P_{ij} be the joint distribution defined on the product space of random variables $X(j)$ and $Y(k)$, and P_j and Q_k be the marginal distributions. The conditional entropy is the amount of uncertainty that remains as to the a value X will assume, if the value of Y has been specified. Formally, the conditional entropy is given by

$$H(X|Y) = -\sum_{j,k} P_{jk} \log P_{j|k}. \tag{2.86}$$

The mutual information, $I(X;Y)$ is the amount of information that the knowledge of Y provides about the value assumed by X. The mutual information can thus be written as $I(X;Y)$ is defined as

$$I(X;Y) = -\sum_{j,k} P_{jk} \log \frac{P_{jk}}{P_j Q_k}. \tag{2.87}$$

It can be shown that is

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X). \end{aligned} \tag{2.88}$$

Distortion measure: The cost function $\rho(\mathbf{X}, \mathbf{Y})$ which specifies the penalty charged for reproducing the source word $\mathbf{X}$ by vector $\mathbf{Y}$ is called word distortion measure. Let $\{x_t, t = 0, \pm 1, \pm 2, ...\}$ be a time-discrete stationary source. A sequence of word distortion measures, called the fidelity criterion, is given by

$$F_\rho = \{\rho_n(\mathbf{X}, \mathbf{Y}), 1 \leq n < \infty\},$$

where

$$\rho_n(\mathbf{X}, \mathbf{Y}) = \frac{1}{n} \sum_{l=1}^{n} \rho(X_l, Y_l).$$

For the magnitude-error criterion $\rho(X, Y) = |X - Y|$, and in the case of the squared-error criterion $\rho(X, Y) = (X - Y)^2$. The average distortion associated with the conditional distribution Q is denoted by

$$d(Q) = \sum_{j,k} P_j Q_{k|j} \rho_{jk}, \tag{2.89}$$

where $p_j Q_{k|j} = P_{jk}$ is the joint distribution. The conditional probability is said to be D-admissible iff $d(Q) \leq D$.

For a fixed D the rate distortion function with respect to a specified fidelity criterion F_ρ is defined as

$$R(D) = \min_{Q \in Q_D} I(Q), \tag{2.90}$$

where

$$Q_D = \{Q_{k|j} : d(Q) \leq D\}.$$

In other words, the rate distortion function is the least information about the source that must be conveyed to the user in order to achieve a prescribed fidelity. Let D_{max} be the minimum value that $d(Q)$ as given by (2.89) can assume. In general, $R(D)$ is a continuous, monotonically decreasing, convex $\cup$ function in the interval $D = 0$ to $D = D_{max}$ and $R(D) = 0$ for $D > D_{max}$. It can be shown that $R(D)$ always exists and $0 \leq R(D) \leq \log K$, where K is the size of the source alphabet. For all cases $R(0) \leq H(X)$ and the equality holds if reproducing alphabet images the source alphabet in the sense that for each source letter there is a unique reproducing letter such that $\rho(j, k) = 0$.

2.7.1 The application of $R(D)$

Let $\rho_n(X, Y)$ be the distortion measure for words of length n, and $B = \{y_1, ..., y_M\}$ be a codebook of size M and block length n. If $\rho(B) = \mathbf{E}[\rho_n(X|B)] \leq D$, then B is a D-admissible code. The smallest size of any D-admissible code is denoted by $M(n, D)$.

The fundamental source coding theorem establishes that for any $\epsilon > 0$ and $D \geq 0$, an integer n can be found such that there exists a $(D + \epsilon)$-admissible code of block length n with rate $R < R(D) + \epsilon$. In other words

$$\frac{1}{n} \log M(n, D + \epsilon) < R(D) + \epsilon \quad \text{for sufficiently large n.} \tag{2.91}$$

The converse of this theorem states that no D-admissible source code has a rate less than $R(D)$.

These theorems show that with given fidelity the rate distortion function is a lower bound to encode any discrete memoryless source (d.m.s.). As a consequence

of the source coding theorem and its converse, we have that for all $D > 0$

$$\lim_{n\to\infty} \frac{1}{n} \log M(n, D) = R(D), \tag{2.92}$$

which sometimes is referred to a definition of $R(D)$. It can also be proved (information transmission theorem) that it is impossible to reproduce a d.m.s. with fidelity D at the receiving end of any discrete memoryless channel of capacity $C < R(D)$ bits per source letter.

These theorems also provide the practical significance of the rate distortion function for communications.

2.7.2 Continuous amplitude stationary sources

All the definitions that were given for discrete sources can also be extended to continuous-amplitude or analog sources.

Let X be a random variable with cumulative distribution $P(x) = Pr(X \leq x)$. If $P(x)$ is continuous, X is called continuous random variable. Let $p(x) = P'(x)$ be the probability density function for X. The differential entropy $h(X)$ is defined as

$$h(X) = \int_S p(x) \log p(x) dx, \tag{2.93}$$

where S is the support set of the random variable X. The differential entropy for a continuous random vector $\mathbf{X} = (X_1, ..., X_n)$ is given by

$$h(\mathbf{X}) = \mathbf{E}[-\log P(\mathbf{X})] = -\int p(\mathbf{x}) \log p(\mathbf{x}) \mathbf{dx} \tag{2.94}$$

where $\mathbf{dx} = dx_1 dx_2 ... dx_n$. For two continuous random variables X and Y, the conditional differential entropy is defined by

$$h(X|Y) = -\int \int p(x, y) \log p(x, y) dx dy \tag{2.95}$$

and the average mutual information is given

$$\begin{aligned} I(X;Y) &= \int\int p(x,y)\log\frac{p(x,y)}{p(x)q(y)}dxdy \\ &= h(X) - h(X|Y). \end{aligned} \tag{2.96}$$

For any one-to-one transformation of coordinates, the differential entropy changes by an amount equal to the expected value of the log of magnitude of the Jacobian. In the new coordinate $[z_i = f(x_i)\ , 1 \leq i \leq n]$, the differential entropy is given by

$$h(Z) = h(X) + \mathbf{E}\left[\log|J|\right], \tag{2.97}$$

where J is the Jacobian of the transformation. Since $I(X;Y)$ is the difference of two differential entropies, it is not changed under a one-to-one transformation.

Rate distortion function for continuous source: For a continuous source, $\rho(x)$ is defined the measure of accuracy of the reproduction source. The average distortion and the average mutual information assigned to any conditional density $q(Y|X)$, are defined as

$$d(q) = \int\int p(x)q(x)q(y|x)\rho(x,y)dxdy \tag{2.98}$$

$$I(q) = \int\int p(x)q(y|x)\log\frac{q(y|x)}{q(y)}dxdy. \tag{2.99}$$

The rate distortion function of a source with respect to a fidelity criterion F_ρ is defined by

$$R(D) = \inf_{q\in Q_D}(I(q)), \tag{2.100}$$

where $Q_D = \{q(y|x) : d(q) \leq D\}$. With some mathematical operations the minimum can be achieved for

$$q(y|x) = \lambda(x)q(y)e^{s\rho(x,y)}, \tag{2.101}$$

where λ is given

$$\lambda(x) = \left[\int q(y)e^s\rho(x,y)dy\right]^{-1}, \tag{2.102}$$

and

$$R(D) = sD + \int p(x) \log \lambda_x dx, \tag{2.103}$$

$$D = \int \int \lambda(x) p(x) e^{s\rho(x,y)} dx dy. \tag{2.104}$$

It can be shown that

$$\begin{aligned} c(y) &\equiv \left[\int q(y) e^{s\rho(x,y)} dy \right]^{-1} \\ &= \begin{cases} 1 & \text{for } q(y) > 0 \quad \forall y \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2.105}$$

2.7.3 Shannon lower bound

The distortion measurement is called a difference distortion, if $\rho(x) = \rho(x - y)$. In the case of difference distortion measures , it can be proved that

$$R(D) \geq h(p) + sD - \log \int e^{s\rho(z)} dz = R_L(D), \tag{2.106}$$

where $R_L(D)$ is called the Shannon lower bound.

The following theorem gives the condition that a rate distortion function equals to its lower bound.

Given any $s < 0$, $R(D_s) = R_L(D_s)$, if and only if the source x can be expressed as the sum of two statistically independent random variables one of which is distributed according to the probability density function $g_s(.)$ given by

$$g_s(X) = \frac{e^{s\rho(x)}}{\int e^{s\rho(z)dz}}. \tag{2.107}$$

For magnitude error distortion measure $\rho(x-y)=|x-y|$, this probability density function and the Shannon lower bound is given by

$$g_s(X) = \frac{|s|}{2}e^{s|X|} \tag{2.108}$$

$$R_L(D) = h(p) - \log(2eD) \tag{2.109}$$

where

$$D = \frac{1}{|s|}.$$

On the other hand, for squared-error distortion measure $\rho(x-y)=(x-y)^2$, the probability density function and the lower bound can be obtained by

$$g_s(X) = \sqrt{\frac{|s|}{\pi}}\, e^{sX^2} \tag{2.110}$$

$$R_L(D) = h(p) - \log(2\pi eD) \tag{2.111}$$

where

$$D = \frac{1}{2|s|}$$

An important special case is when $p(.)$ is the normal density,

$$N(\mu,\sigma^2) = (2\pi\sigma^2)^{-1/2}\exp\left[\frac{-(x-\mu)^2}{2\sigma^2}\right]. \tag{2.112}$$

In this case, it can be shown that

$$q(y) = \left[2\pi(\sigma^2-D)\right]^{-1/2}\exp[\frac{-(y-\mu)^2}{2(\sigma^2-D)}\Big], \tag{2.113}$$

that is, the output has a normal distribution, $N(\mu,\sigma^2-D)$. We can deduce that for a memoryless Gaussian source and squared error criterion, the difference between the input source and its reproduction, $Z=X-Y$, is normal with variance D. Also for normally distributed source

$$R_L(D) = R(D) \;\;,\;\; 0 < D \le \sigma^2 = D_{max},$$

and

$$R(D) = \begin{cases} \frac{1}{2}\log\frac{\sigma^2}{D} & 0 \le D \le \sigma^2 \\ 0 & D \ge \sigma^2 \end{cases} \tag{2.114}$$

It has been proved that the upper bound of the rate distortion function of any source with zero mean and variance σ^2 is the rate distortion function of a Gaussian source, that is,

$$R(D) \leq \frac{1}{2} \log \frac{\sigma^2}{D}, \tag{2.115}$$

with equality sign holding iff $p(x)$ is normal.

Chapter 3

RESIDUAL VECTOR QUANTIZER

3.1 INTRODUCTION

Optimum vector quantizers, designed using generalized Lloyd algorithm [30], can be used for a variety of sources. However, their practical applications are limited by the complexity of codebook search and codebook storage. Lack of a structure in an optimum Vector Quantizer (VQ) is the reason for the complexity. This explains the interest in VQ schemes with structured codebooks, such as tree searched [8], residual (multi-stage) [9], gain/shape [10], and lattice-based vector quantizers [11]. In order to reduce the search complexity, Buzo *et al.* [8] have proposed a tree searched encoder. In their method, the encoder searches a sequence of small codebooks instead of a large one. In this way, the complexity of search is reduced with a small increase in the distortion, but the codebook storage requirement is greater than that in the full-search VQ. Multi-stage VQ [9] divides the quantization task into several successive stages, resulting in a reduction of codebook search and storage complexity, but it increases the encoding distortion. For example, in a two-stage VQ, after the input vector $\mathbf{X}$ is quantized by the first stage, the error is quantized by the second stage, and the final reproduction of $\mathbf{X}$ is the summation of the two quantized levels. If the two-stage VQ has M_1 code-words in the first stage and M_2 code-words in the second one, it requires $M_1 + M_2$ distance computation, Where as the corresponding single-stage VQ would have required $M_1 \times M_2$ memory space and $M_1 \times M_2$ distance computations. Thus, with the same rate and dimension, the complexity of a two-stage VQ is much less than that of a single-stage VQ. This reduction in the complexity comes at the expense of an increased distortion.

The point in the multi-stage quantizers is to find the condition under which the source can be successively reconstructed without loss of optimality [37], [38]. Several researchers have investigated the problem of successive refinement of information. The goal of these studies is to achieve an optimal description at each stage to ensure that the on going description is optimal whenever it is interrupted. Equitz *et al.*

[37] have shown that a source is successively refinable if and only if the individual solution of the rate distortion problem for the source can be written as a Markov chain. Since it can be shown that there exists a Markov chain for a Gaussian distributed signal under the MSE criterion, a Gaussian source is successively refinable [37].

From the rate distortion theory, for most memoryless sources and many Gaussian sources with memory, the ideal encoding noise under MSE criterion, for small distortion is memoryless and Gaussian. Based on the modeling assumption of a Gaussian distributed first-stage VQ encoding error, Pan and Fischer[39] introduced a two-stage quantizer with a lattice vector quantizer with a spherical codebook for the second-stage for memoryless sources.

In [40], Lee *et al.* have shown that if the source density is smooth and the first-stage is a high-rate VQ, then it can be assumed that the first-stage error is uniform over each quantization cell. They have also shown that the overall encoding distortion approaches asymptotically to that of a single stage VQ as the size of the first-stage codebook approaches infinity.

However, the assumption of a Gaussian quantization error cannot be extended to sources with memory, such as images. In this chapter, it is shown that the residual vectors normalized by the zonal energy have a distribution close to a Normal distribution. Therefore, a quantizer designed for a Gaussian source is almost optimal for these normalized error samples.

This adaptation is particularly efficient, since for a fixed compression ratio, the same codebook is used for any residual samples of images. This method is also applicable for the raw Synthetic Aperture Radar (SAR) data, since the raw data

statistics is Gaussian with zero mean and also they are uncorrelated [41].

This chapter is organized as follows. Section 3.2 gives a brief introduction to multi-stage residual VQ and discusses the problem of successive refinement. Section 3.3 describes the distribution of error signal. The problem of the mismatch of distribution of the codebook and the distribution of the source is investigated in Section 3.4. Section 3.5 presents the Kolmogorov-Smirnov test (KS) [42] which is a test for goodness of fit of distribution to the different distributions. Finally, Section 3.6 presents the results of simulation. Section 3.7 gives a summary of study carried out in this chapter.

3.2 MULTI-STAGE VQ

A Multi-stage Residual Quantizer (RQ) consists of a cascade of quantizer stages, each operating on the residue of the previous stage. The block diagram of a residual quantizer is shown in Figure 3.1. In a residual quantizer the total distortion is the distortion of the final stage. For probability mass function $p(x)$ and conditional probability mass function of $q(y|x)$, in a K-stage quantizer, the total rate and distortion are, respectively, given by

$$D = D_K = \int \int p(x_K) q(y_K|x_K) d(x_K, y_K) dx_K dy_K \tag{3.1}$$

and

$$R = R_1 + R_2 + ... + R_K, \tag{3.2}$$

where X_K and Y_K are the input and output of the last stage, and R_i and D_i are the rate and distortion of the ith stage. The residue of each stage is the input to the next stage. For example, the input of stage i is given by

$$X_i = X_{i-1} - Y_{i-1}.$$

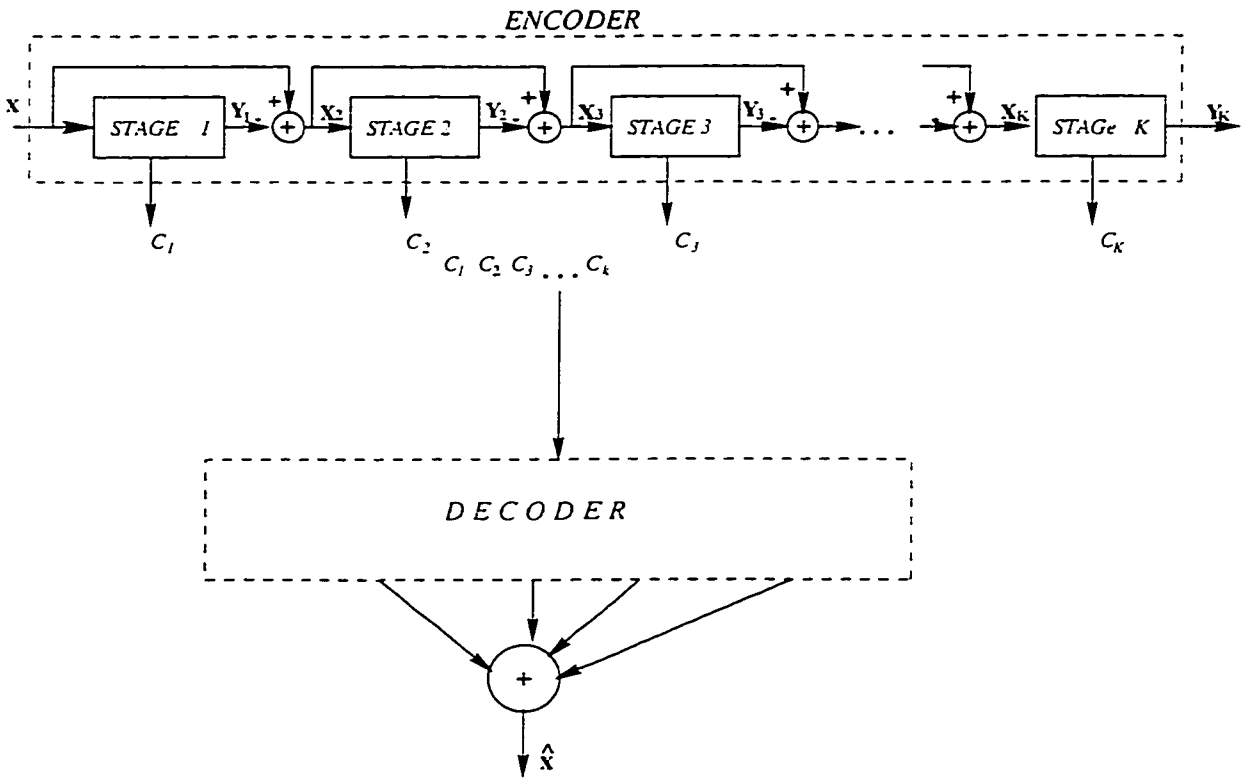

Figure 3.1: The block diagram of a K-stage residual quantizer.

For an unknown source, the multi-stage quantizer is not optimum in the sense of rate and distortion. Several researchers have published results on the condition under which a multi-stage quantizer can be an optimum [37] [38]. All these investigations are based on the jointly good description [38]. In the jointly good description the goal is that by sending two descriptions of the source, each describing it well, at the receiver the combination of the descriptions can give the maximum possible information.

Consider a stochastic process $X_1, X_2, \ldots$, where each X_i is an independent, identically distributed, i.i.d., random variable with a known distribution $p(x)$. X is encoded twice with rates R_1 and R_0 bits per symbol. Given three single letter

distortion measures, d_1, d_2 and d_0, the problem is to find the information that should be sent at rate R_1 and R_0 so that a receiver given only R_1 can reconstruct X with distortion D_1, given only R_0 can recover X with distortion D_0, and given both descriptions can recover X with distortion D_2. The problem of multiple descriptions was posed by Witsenhausen [43], Wolf *et al.* [44], and Ozarow [45]. Gamal and Cover [38] in their work exhibit an achievable rate region of (R_0, R_1) pairs as a function of the distortion vector $\mathbf{D} = (D_0, D_1, D_2)$.

Consider a sequence of blocks $\mathbf{X} = (X_1, X_2, ...X_n)$, where X_i's are i.i.d. random variables with a known distribution $p(x)$. By definition, the achievable rate for distortion $\mathbf{D} = (D_0, D_1, D_2)$ is (R_0, R_1), if there exist a sequence of descriptions $i(x) \in \{1, 2, ..., 2^{nR_1}\}$ and $j(x) \in \{1, 2, ..., 2^{nR_0}\}$, and reconstruction functions $\hat{x}_1(i)$, $\hat{x}_0(j)$, $\hat{x}_2(i, j)$ such that for a sufficiently large n

$$E[d_m(\mathbf{X}, \hat{\mathbf{X}}_\mathbf{m})] \leq D_m, \qquad m = 0, 1, 2, \tag{3.3}$$

where $d_m(.,.)$ is the distortion measure defined by the average per-letter distortion,

$$d_m(\mathbf{X}, \hat{\mathbf{X}}_\mathbf{m}) = \frac{1}{n} \sum_{i=1}^{n} d_m(x_i, \hat{x}_{mi}),$$

and $\hat{\mathbf{X}}$ is the sequence of the reconstruction vectors. The rate distortion region is the closure of the set of achievable rate pairs (R_0, R_1) inducing a distortion less than or equal $\mathbf{D}$. An achievable rate region is any subset of the rate distortion region. Gamal and Cover [38] proved that the achievable rate region for distortion $D = (D_0, D_1, D_2)$ is given by the convex hull of all (R_0, R_1) pairs such that

$$\begin{aligned} R_0 &> I(X; \hat{X}_0) \\ R_1 &> I(X; \hat{X}_1) \\ R_0 + R_1 &> I(X; \hat{X}_0, \hat{X}_1, \hat{X}_2) + I(\hat{X}_0; \hat{X}_1) \end{aligned} \tag{3.4}$$

if there exists a probability mass function $p(x, \hat{x}_0, \hat{x}_1, \hat{x}_2) = p(x)p(\hat{x}_0, \hat{x}_1, \hat{x}_2|x)$ such that

$$D_m \geq E[d_m(X, \hat{X}_m)] \qquad m = 0, 1, 2 \tag{3.5}$$

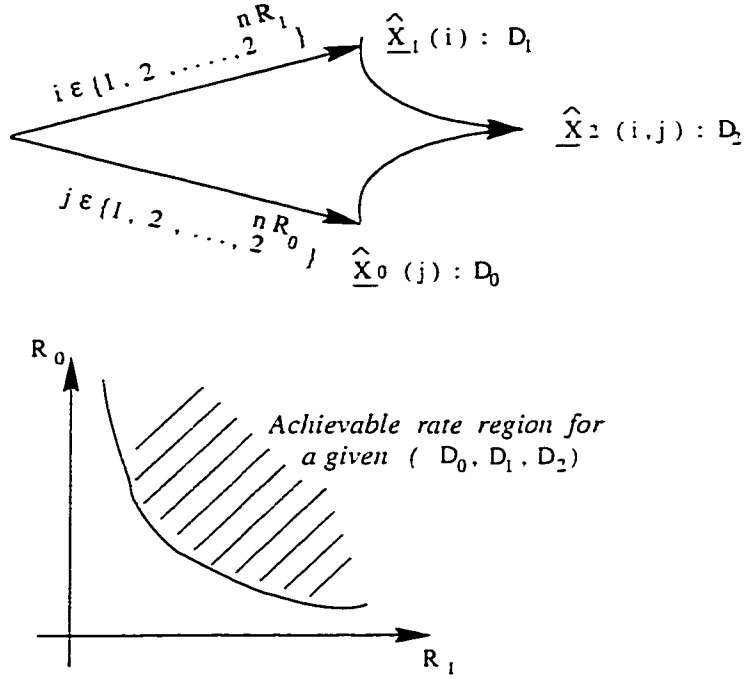

Figure 3.2: Multiple description and achievable rate region.

where $I(0)$ denotes Shannon mutual information. Ahlswede [46] showed that the above conditions are both necessary and sufficient in the "no-excess rate case," i.e. $R_0 + R_1 = \Re(D_2)$. For clarity, we use R for the rate and $\Re(.)$ for rate distortion function. Figure 3.2 shows the case where two receivers receive individual descriptions and the third has access to both descriptions. The lower diagram in this figure shows the achievable rate region.

The successive refinement problem which is shown in Figure 3.3 is a special case of the multiple description problem. In this case, there is no constraint on $D_0 = E[d_0(\mathbf{X}, \hat{\mathbf{X}}_0)]$ and we require $R_1 = \Re(D_1)$ and $R_2 = R_0 + R_1 = \Re(D_2)$. In general, the successive refinement from distortion D_1 to distortion D_2 is achievable if there exists a sequence of encoding function $i : \mathcal{X}^n \to \{1, 2, ..., 2^{nR_1}\}$ and $j : \mathcal{X}^n \to \{1, 2, ..., 2^{n(R_2 - R_1)}\}$ and reconstruction functions $q_1 : \{1, 2, ...2^{nR_1}\} \to \hat{\mathcal{X}}_1^n$ and

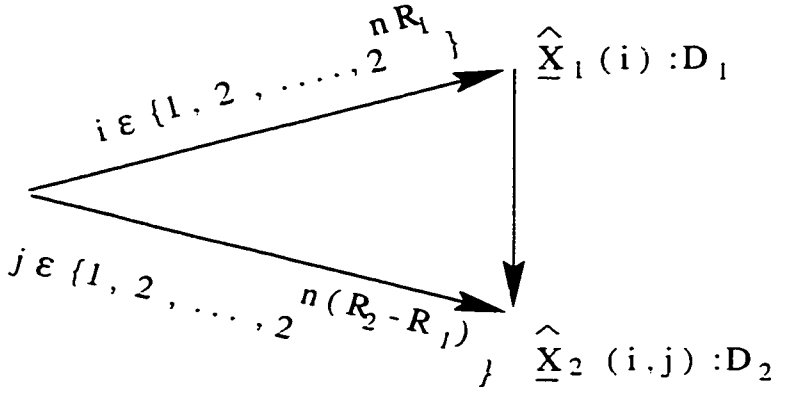

Figure 3.3: The successive refinement.

$q_2 : \{1, 2, \ldots 2^{nR_1}\}\{1, 2, \ldots 2^{n(R_2-R_1)}\} \rightarrow \hat{X}_2^n$, such that for

$$\hat{X}_1^n = q_1(i(X^n)) \tag{3.6}$$

and

$$\hat{X}_2^n = q_2(i(X^n), j(X^n)), \tag{3.7}$$

we have

$$\lim_{n\to\infty} \sup E[d(X^n, \hat{X}_1^n)] \leq D(R_1) \tag{3.8}$$

and

$$\lim_{n\to\infty} \sup E[d(X^n, \hat{X}_2^n)] \leq D(R_2). \tag{3.9}$$

In 3.8 and 3.9, $D(R)$ is the distortion rate function defined by

$$D(R) = \min_{p(\hat{X}|X)} E[d(X, \hat{X})] \tag{3.10}$$

In other words, the sequence $X_1, X_2, ..., X_n$ is successively refined if $R_1 = \Re(D_1)$ and $R_2 = \Re(D_2)$, i.e., the rate distortion limit in each of the two stages is achieved.

The successive refinement for the quantization of a single variable is not achievable. However, if long blocks of i.i.d. variables were considered , the successive refinement in some cases is possible. For example, for long blocks of i.i.d. Gaussian

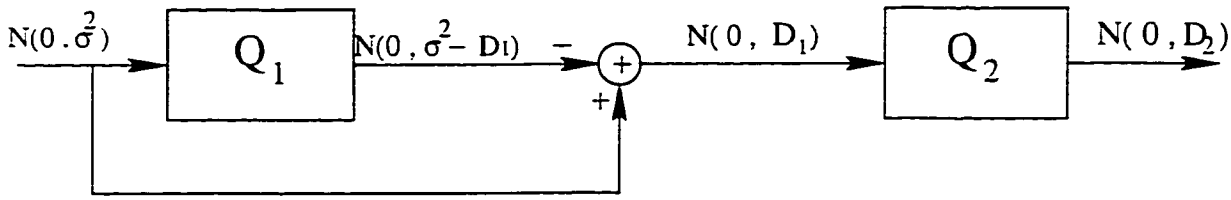

Figure 3.4: The block diagram of a two-stage residual quantizer for a Gaussian source.

random variables, the successive refinement is always possible.

Equitz and Cover [37] have proved that successive refinements from description $\hat{X}_1$ with distortion D_1 to description $\hat{X}_2$ with distortion $D_2 \leq D_1$ is achievable if and only if there exists a conditional distribution $p(\hat{x}_1, \hat{x}_2|x)$ such that $X, \hat{X}_1, \hat{X}_2$ can be written as a Markov chain $X \rightarrow \hat{X}_2 \rightarrow \hat{X}_1$. In this case the joint conditional distribution becomes

$$p(\hat{x}_1, \hat{x}_2|x) = p(\hat{x}_2|x)p(\hat{x}_1|\hat{x}_2). \tag{3.11}$$

As an example, consider the random variable $N(0, \sigma^2)$. Under the MSE criterion, the error signal is Gaussian with the variance D. It means that if X is $N(0, \sigma^2)$ then $p(\hat{x}) = N(0, \sigma^2 - D)$ and $p(x|\hat{x}) = N(\hat{x}, D)$. It can be shown that the source is refinable. For the two-stage residual quantizer,shown in Figure 3.4, we can write

$$p(\hat{x}_1) = N(0\ , \sigma^2 - D_1) \tag{3.12}$$

$$p(x|\hat{x}_2) = N(\hat{x}_2\ , D_2) \tag{3.13}$$

$$p(\hat{x}_2|\hat{x}_1) = N(\hat{x}_1\ , D_1 - D_2) \tag{3.14}$$

It can be shown that 3.13-3.14 yield a joint function

$$p(x, \hat{x}_1, \hat{x}_2) = p(\hat{x}_1)p(\hat{x}_2|\hat{x}_1)p(x|\hat{x}_2). \tag{3.15}$$

implying that X, $\hat{X}_1$, $\hat{X}_2$ can be written as a Markov chain, $X \rightarrow \hat{X}_2 \rightarrow \hat{X}_1$. Since there exists a Markov chain satisfying the Equitz and Cover theorem [37], the Normal source with MSE criterion is successively refinable. The existence of the

Markov chain also guarantees the achievebility of

$$(R_1, R_2) = (\Re(D_1), \Re(D_2)) = (\frac{1}{2}\log\frac{\sigma^2}{D_1}, \frac{1}{2}\log\frac{\sigma^2}{D_2}). \tag{3.16}$$

In the following, we present an alternative proof showing that the Gaussian source under MSE criterion is refinable and $\Re(D) = \Re_1(D_1) + \Re_2(D_2)$.

Under the mean-squared error criterion, the rate distortion function of the first stage with a Gaussian input $N(0, \sigma^2)$ is given by [36],

$$\Re_1(D_1) = \frac{1}{2}\log\frac{\sigma^2}{D_1}. \tag{3.17}$$

Since the input to the second-stage is also Normal with variance D_1, the rate distortion function for the second stage is given by

$$\Re_2(D_2) = \frac{1}{2}\log\frac{D_1}{D_2}. \tag{3.18}$$

For the two stage, $D = D_2$. Thus,

$$\Re_2(D_2) = \frac{1}{2}\log\frac{D_1}{D}. \tag{3.19}$$

For the two-stage residual quantizer, the rate distortion function is given by

$$\begin{aligned} \Re(D) &= \Re_1(D_1) + \Re_2(D_2) && (3.20)\\ &= \frac{1}{2}\log\frac{\sigma^2}{D_1} + \frac{1}{2}\log\frac{D_1}{D} && (3.21)\\ &= \frac{1}{2}\log\frac{\sigma^2}{D}. && (3.22) \end{aligned}$$

Extending this result to a K-stage quantizer we have

$$\begin{aligned} \Re(D) &= \Re_1(D_1) + \Re_2(D_2) + ... + \Re_K(D_K) && (3.23)\\ &= \frac{1}{2}\log\frac{\sigma^2}{D_1} + \frac{1}{2}\log\frac{D_1}{D} + ... + \frac{1}{2}\log\frac{D_{K-1}}{D} && (3.24)\\ &= \frac{1}{2}\log\frac{\sigma^2}{D}. && (3.25) \end{aligned}$$

It can also be shown that the Laplacian source under the absolute error criterion, $d(x, \hat{x}) = |x - \hat{x}|$, is refinable [37].

As a result, dividing the quantization task into several successive stages does not affect the accuracy of the quantizer. In addition, the residual quantizer can achieve a great deal of savings in terms of storage and computational complexity.

According to the fundamental source coding theorem, for a block quantizer with dimension n and codebook size K, the average rate of any D-admissible code is $\Re(D)$, that is,

$$\lim_{n\to\infty} \frac{1}{n} \log(K, D) = \Re(D). \tag{3.26}$$

Thus, using a block coding with a large dimension, the multi-stage quantizer for Normal distribution is successively refinable.

3.3 DISTRIBUTION OF ERROR SAMPLES

Under the MSE criterion, the rate-distortion function of a wide class of memoryless sources for small distortions (equivalently, for high rates), approaches the Shannon lower bound. As a result, for high rate, the ideal encoding noise is memoryless and Gaussian. For many Gaussian sources with memory, a critical rate exists such that for the rates larger than this critical rate, the encoding error becomes white and Gaussian [36]. For example, for a Gauss-Markov source with parameter ρ, the rate distortion function is given by

$$R(D) = \frac{1}{2} \log_2 \frac{1-\rho^2}{D} \qquad D \leq \frac{1-\rho}{1+\rho}. \tag{3.27}$$

For this source the critical rate corresponding to $D = (1-\rho)/(1+\rho)$ is

$$R_c = \log_2(1+\rho). \tag{3.28}$$

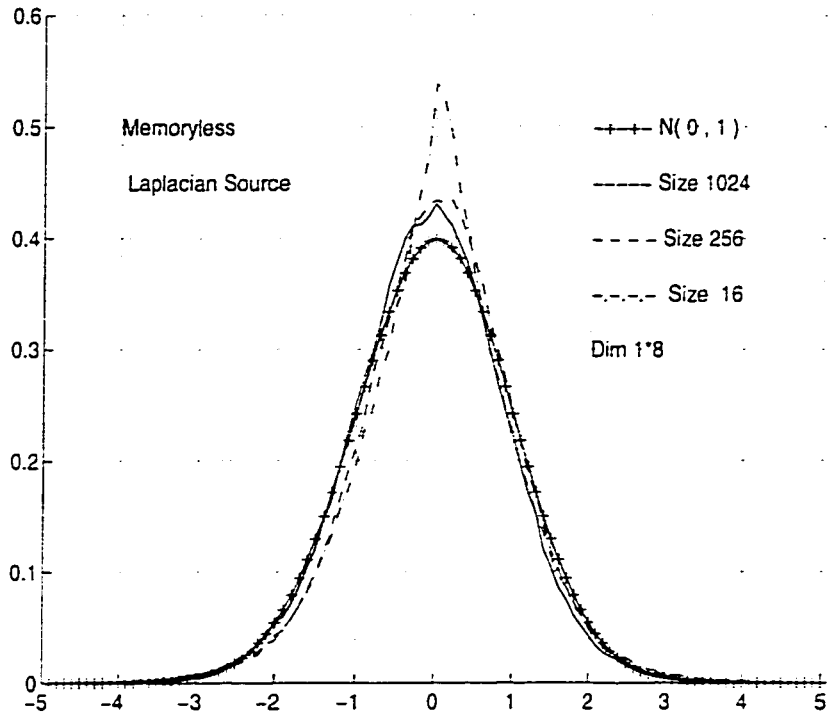

Figure 3.5: Normalized histogram of the VQ encoding error for memoryless Laplacian source with dimension 8 and codebook sizes ranging from 16 to 1024.

In other words, if the source encoding rate is $R \geq R_c$, then the optimum quantization noise is white and Gaussian. In the case $R < R_c$, the optimum encoding noise is not Gaussian. As a consequence, as the rate increases, the error signal tends to be memoryless Gaussian, so that it becomes successively refinable. This justifies the use of a multi-stage residual quantizer for the error signal without significant loss of optimality. Figure 3.5 shows the histogram of error signals corresponding to memoryless Laplacian sources for dimension $L = 8$ and for various codebook sizes $M = 2^{LR}$. As Figure 3.5 shows, for large codebook, the error signal is close to a Gaussian source with the same variance.

For Gaussian sources with memory, the effectiveness of an encoding method is dependent on the feasibility of using a large enough first-stage vector quantizer

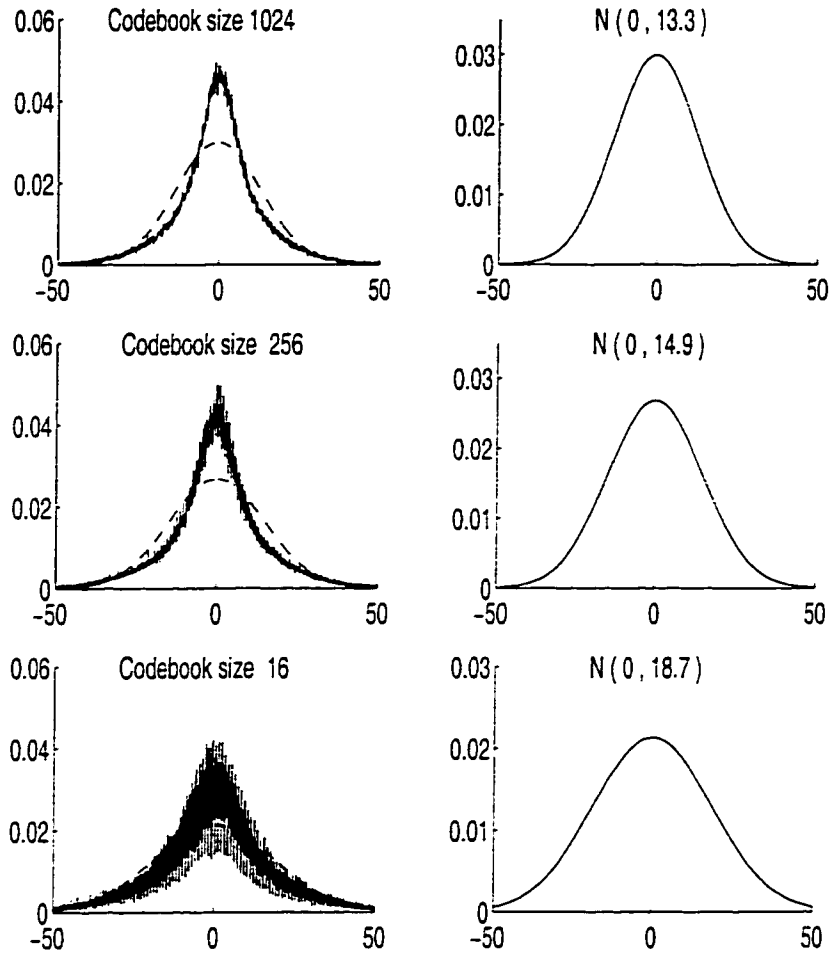

Figure 3.6: Normalized histogram of the VQ encoding error for image *Baboon* with dimension 4 and different codebook sizes.

codebook to exploit most of the source memory. We have studied the effect of the increase in the rate for a constant dimension and the effect of increase in the dimension for a constant rate for different images on the distribution of the error signal. These results show that for the implementable rates, the idea cannot be extended to the sources with memory like images. For example, Figure 3.6 shows the comparison of the distribution of error signal for image *Baboon* with that of a Gaussian source. It can be seen even for bit rate 2.5 bps, the distribution is not a good fit to a Gaussian distribution.

For a multi-stage VQ with a reasonable rate quantization in the first-stage, the error signal is far from having a global distribution. This is due to the different statistical parameters in different regions of the image. Also, by using a low rate quantizer in the first stage, the errors samples follow the same distribution. The normalized histogram of the error signals with a low bit rate VQ in the first stage for different images is shown in Figure 3.7. It can be seen that the distributions of the error signal for different images are different and cannot be quantized by a single quantizer. However, by normalizing the error vectors by the zonal energy, their distribution become close to a Normal distribution. In order to carry out this normalization, the two dimensional error signal is divided into different zones and the average energy of each zone is calculated. The error samples in each zone are then divided by the energy of the corresponding zone. The energy of a zone is defined as

$$\varsigma_{z_i}^2 = \frac{1}{||z_i||} \sum_{all\ x\ in\ z_i} x_i^2 \tag{3.29}$$

where z_i refers to the ith zone and $||.||$ denotes the cardinality given by

$$\sum_{all\ zones} ||z_i|| = n_s \ , \tag{3.30}$$

and n_s is the total number of samples. The locally normalized error signal has a variance of unity. If x is a locally normalized error signal, then its variance is given by

$$\sigma^2 = \frac{1}{n_s} \sum_{all\ samples} x^2 \tag{3.31}$$

$$= \frac{1}{n_s} \sum_{all\ zones} \sum_{all\ samples\ in\ z_i} \frac{x_{ij}^2}{\varsigma_{z_i}^2} \tag{3.32}$$

$$= \frac{1}{n_s} \sum_{all\ zones} \frac{1}{\varsigma_{z_i}^2} \sum_{all\ samples\ in\ z_i} x_{ij}^2 \tag{3.33}$$

$$= \frac{1}{n_s} \sum_{all\ zones} \frac{||z_i||\ \varsigma_{z_i}^2}{\varsigma_{z_i}^2} \tag{3.34}$$

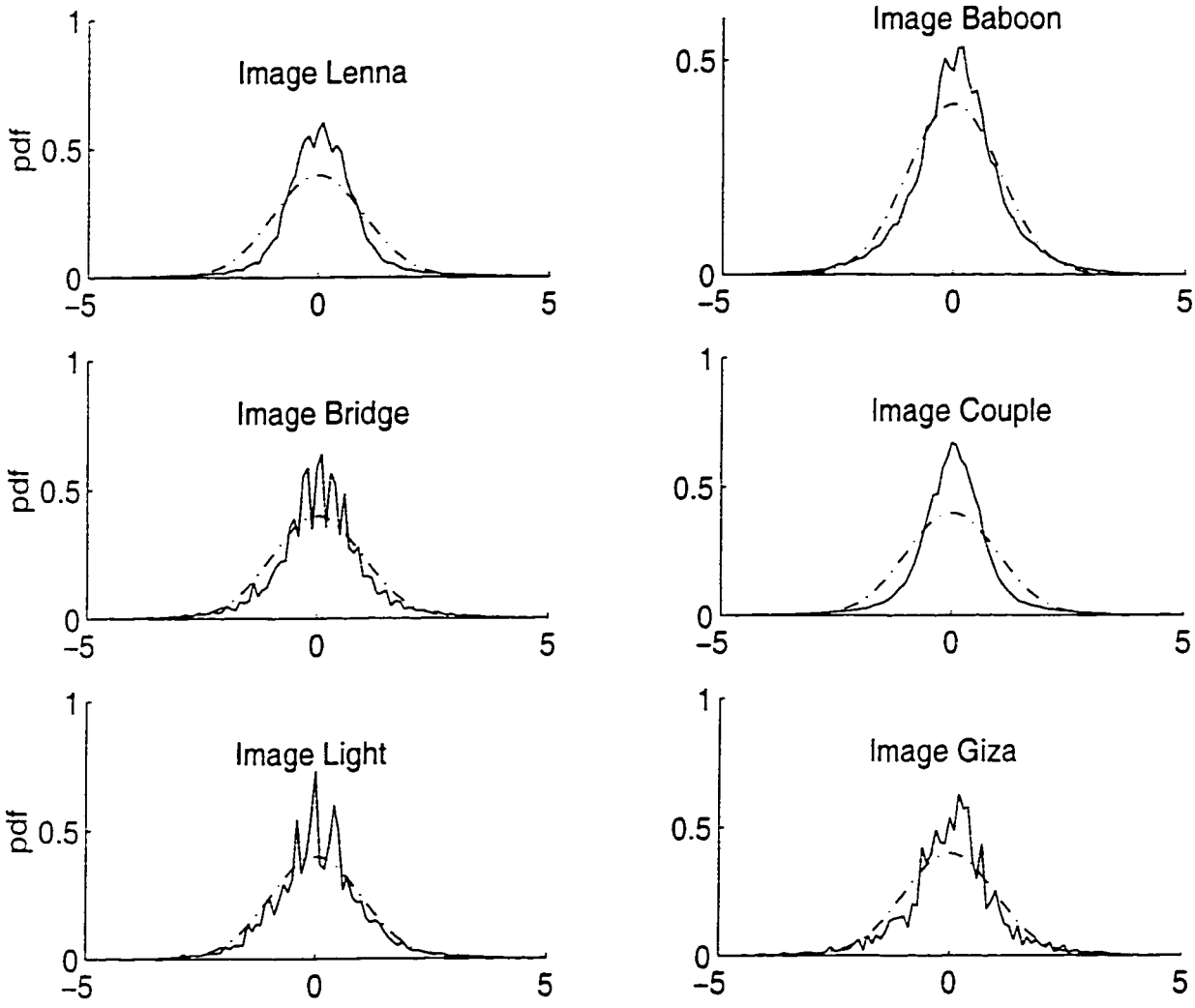

Figure 3.7: Normalized histogram of the VQ encoding error for the different images with dimension 4 and codebook size 16 (1bps).

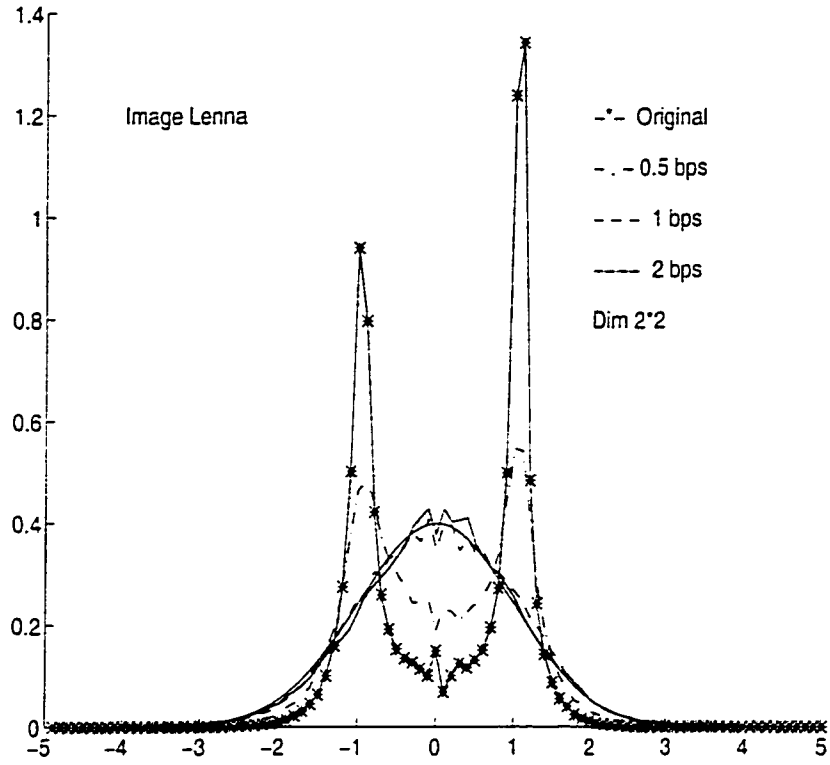

Figure 3.8: Normalized histogram of the VQ encoding error for image *Lenna* with dimension 4 and codebook sizes ranging from 4 to 256.

$$= \frac{1}{n_s} \sum_{all\ zones} ||z_i|| \tag{3.35}$$

$$= 1. \tag{3.36}$$

To obtain a distribution close to a Normal distribution, the bit rate in the first-stage should not be too low. For example, for image *Lenna*, when quantized by 0.5 bps optimum VQ, the normalized distribution of error samples follow the distribution of the original image. By increasing the bit rate in the first stage, this can be changed. Figure 3.8 shows the effect of bit rate in the first stage for the image *Lenna*.

The histograms of the locally normalized error samples for different images are shown in Figures 3.9. The effect of normalizing the error samples locally can be seen from this figure. As seen from figure the error signal has a distribution very

close to Normal. Figure 3.10 shows the comparison of the histograms of error sign and one of the locally normalized error signals.

Since the locally normalized curves are close to a Normal distribution, we co clude that a quantizer designed for a Gaussian source can be considered as almo optimal for these error signals. In the next section, it will be shown that in the ca of a mismatch between the actual distribution and a normal one, the distortion less than the case where the locally normalized error signal is actually Gaussia even though the distortion that one gets is generally more than in the case whe the codebook is optimally designed for the source.

3.4 MISMATCH

It is well known [36] that for all sources with a given second moment σ^2, the sourc that is most difficult to describe within a mean square error distortion D is the mem oryless zero-mean Gaussian source. If the rate distortion function for a memoryle Gaussian source is given by $\Re_g(D)$ and for a general source having the same secon moment σ^2 is $\Re(D)$, then

$$\Re(D) \leq \Re_g(D). \tag{3.37}$$

Let us now assume that a Gaussian codebook of rate $\Re_g(D)$ is used to compress source that is not Gaussian or memoryless. In [47], Sakrison has shown that using codebook designed for a memoryless Gaussian source with a given second momer to compress a non-Gaussian source with the same second moment does not resu in a distortion higher than the distortion corresponding to the original Gaussia source. Lapidoth [48] has shown that the resulting distortion is also no smaller tha the distortion corresponding to the Gaussian source. These results demonstrat that the distortion that one can expect due to the use of a Gaussian codebook fc

close to Normal. Figure 3.10 shows the comparison of the histograms of error signal and one of the locally normalized error signals.

Since the locally normalized curves are close to a Normal distribution, we conclude that a quantizer designed for a Gaussian source can be considered as almost optimal for these error signals. In the next section, it will be shown that in the case of a mismatch between the actual distribution and a normal one, the distortion is less than the case where the locally normalized error signal is actually Gaussian, even though the distortion that one gets is generally more than in the case when the codebook is optimally designed for the source.

3.4 MISMATCH

It is well known [36] that for all sources with a given second moment σ^2, the source that is most difficult to describe within a mean square error distortion D is the memoryless zero-mean Gaussian source. If the rate distortion function for a memoryless Gaussian source is given by $\Re_g(D)$ and for a general source having the same second moment σ^2 is $\Re(D)$, then

$$\Re(D) \leq \Re_g(D). \tag{3.37}$$

Let us now assume that a Gaussian codebook of rate $\Re_g(D)$ is used to compress a source that is not Gaussian or memoryless. In [47], Sakrison has shown that using a codebook designed for a memoryless Gaussian source with a given second moment to compress a non-Gaussian source with the same second moment does not result in a distortion higher than the distortion corresponding to the original Gaussian source. Lapidoth [48] has shown that the resulting distortion is also no smaller than the distortion corresponding to the Gaussian source. These results demonstrate that the distortion that one can expect due to the use of a Gaussian codebook for

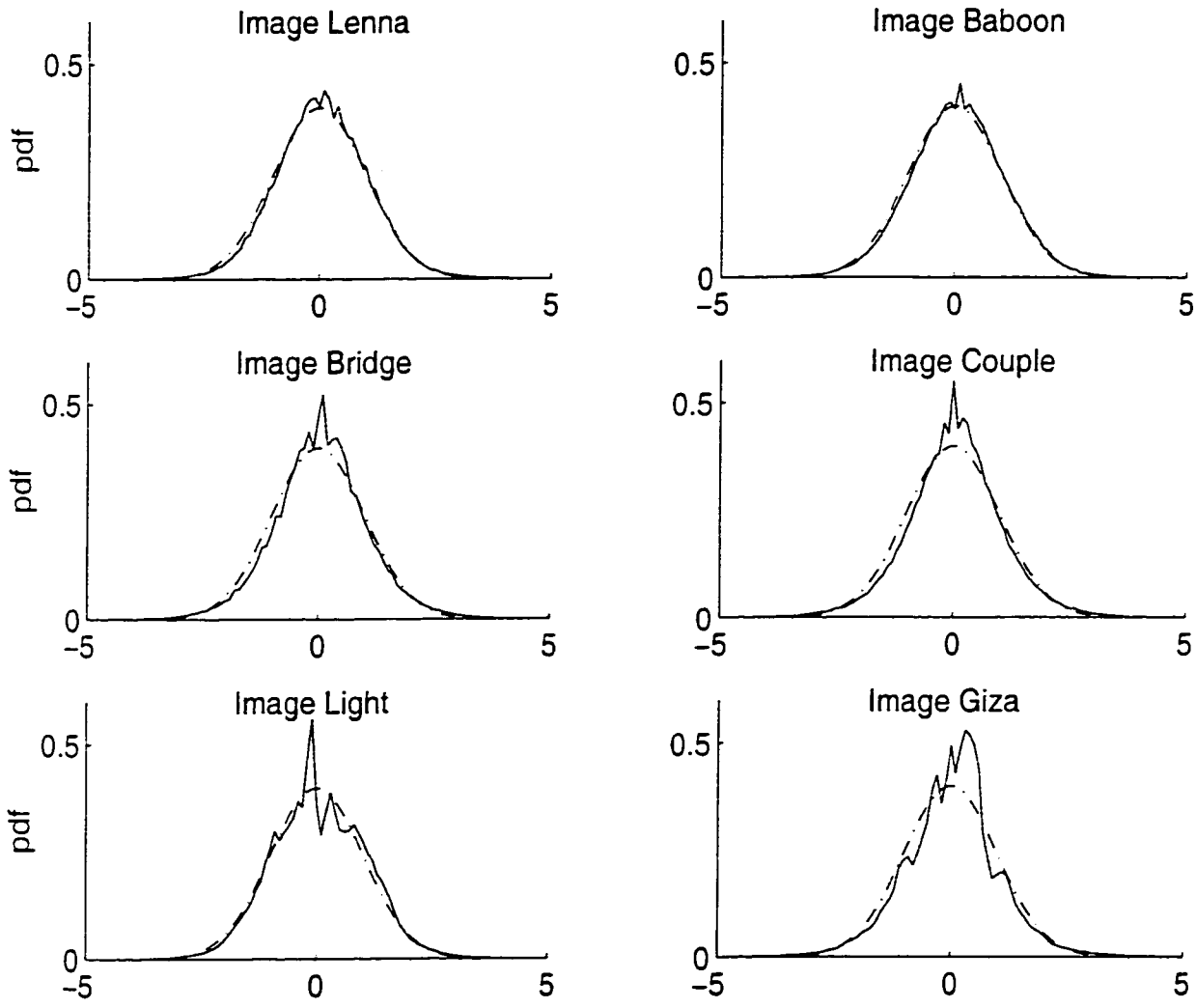

Figure 3.9: Normalized histograms of the locally normalized error for the different images with dimension 4 and codebook size 16 (1bps).

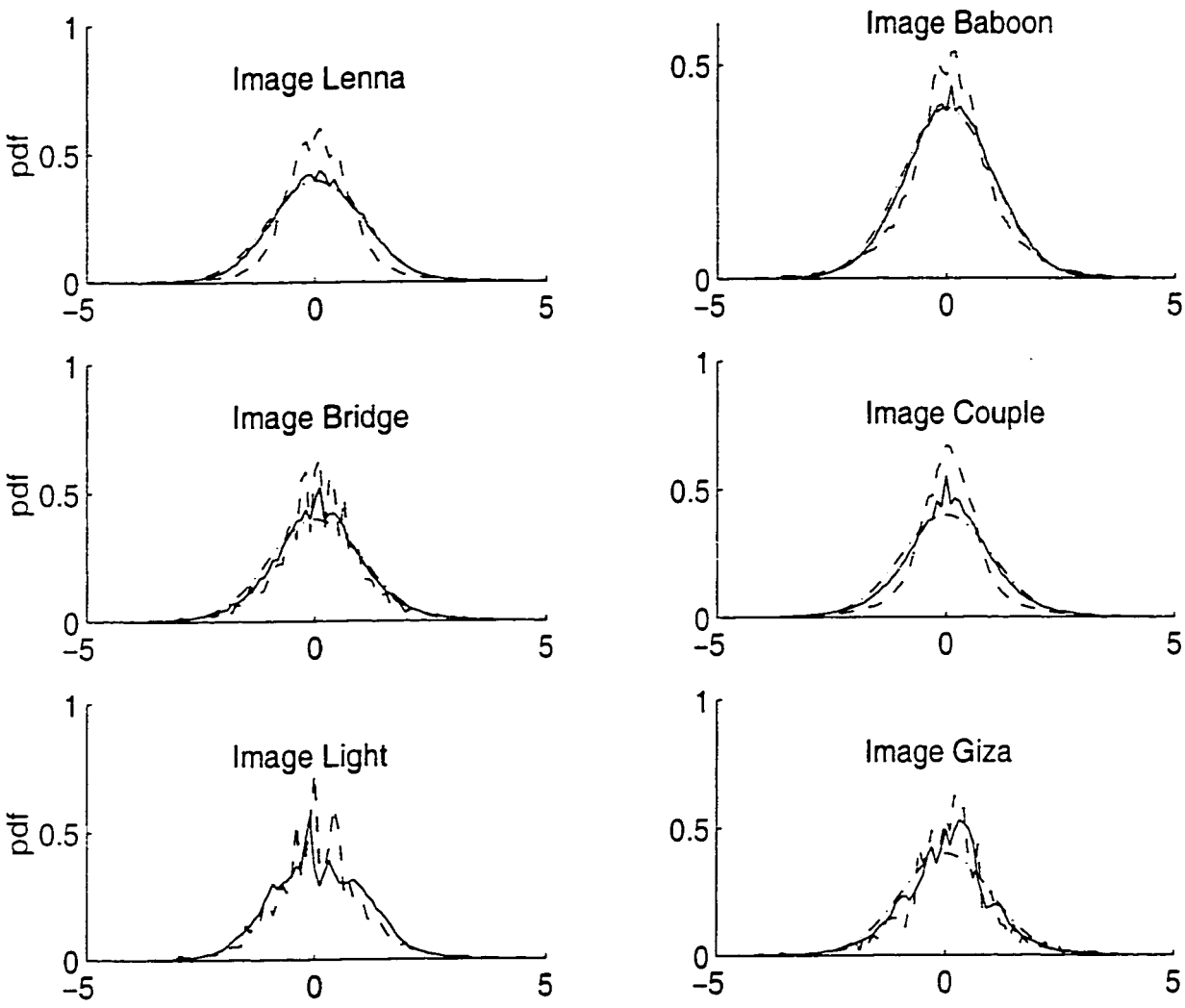

Figure 3.10: Comparison of the histograms of the error signal and the histogram of the locally normalized error for the different images.

a non-Gaussian source with the same variance is exactly the same as the distortion one would get if the non-Gaussian source were actually Gaussian. Indeed, the loss in performance due to the use of a Gaussian (non-optimal) codebook is exactly offset by the use of excess rate. To state this result Lapidoth proved the following theorem [48].

Theorem: Consider a random codebook whose 2^{nR} code words are drawn independently and uniformly over the n-dimensional sphere of radius r_n centered around the origin. Let x be an n-tuple of source samples generated by an ergodic source with a second moment σ^2, and let $0 < D < \sigma^2$.

a) If $R < \frac{1}{2}\log(\sigma^2/D)$ then irrespective of the radii

$$Pr(\exists \hat{x} \in C : ||x - \hat{x}||^2 \leq nD) \stackrel{n\to\infty}{\longrightarrow} 0.$$

b) If $R > \frac{1}{2}\log(\sigma^2/D)$ and $r_n = \sqrt{n(\sigma^2 - D)}$, then

$$Pr(\exists \hat{x} \in C : ||x - \hat{x}||^2 \leq nD) \stackrel{n\to\infty}{\longrightarrow} 1.$$

This result shows that using a universal Gaussian codebook for the multi-stage quantizer is a promising method to achieve a given distortion with low complexity.

To show how much locally normalized error signal is close to the Gaussian source, we have performed the well known Kolmogorov-Smirnov(KS) [42] test for goodness of fit of the distribution.

3.5 KOLMOGOROV-SMIRNOV TEST

The Kolmogorov-Smirnov test (KS) [42] is a test for goodness of fit of a given distribution to various well known distributions. The test statistic is based on a distance measure between the sample distribution function and a well defined

distribution or a test distribution. Let $X = (x_1, ..., x_M)$ be a given set of data. The KS test compares the sample distribution $F_X(.)$ to a given distribution function $F(.)$. If $y_n, n = 1, 2, .., M$ are the order statistic of the data X, then the sample distribution is given by

$$F_X(z) = \begin{cases} 0 & z < y_1 \\ \frac{n}{M} & y_n \leq z \leq y_{n+1} \quad n = 1, 2, ..., M \\ 1 & z \geq y_M. \end{cases} \tag{3.38}$$

The KS test is defined by

$$t = \max_{i=1,2,...,M} |F_X(x_i) - F(x_i)|. \tag{3.39}$$

When different distributions are tested, the one that yields the smallest KS statistic, t, is the best fit for the data. The result of this test for the error signal and the normalized error signal for some images are shown in Table 3.1. The normalized error signal in all cases yields a smaller KS statistic for Gaussian distribution function than the other tested distributions.

3.6 SIMULATION AND RESULTS

The proposed method was investigated in the context of coding of 8-bit monochrome images of size 512 × 512 with different contexts, face and scenery. Also images from Canadian Remote Sensing Satellite, Radarsat, are tested. A set of 2^{18} normally distributed samples were generated and the generalized Lloyd algorithm [30] was used to generate an optimum codebook. The error samples are divided into 16 × 16 vectors (zones), and the samples in each zone were normalized to the magnitude of that zone. Eight bits were used to encode the energy of each zone, i.e., 1/32 bits per

Table 3.1: Kolmogorov-Smirnov test for the error signals of some images

image	Scheme	Gaussian	Laplacian	Cauchy
Lenna	Error signal	0.0175	0.0049	82.41
	Normalized error signal	0.0012	0.0038	83.41
Bridge	Error signal	0.0078	0.0011	83.1837
	Normalized error signal	0.0018	0.0029	83.4436
Baboon	Error signal	0.01	0.0018	83.37
	Normalized error signal	3.54e-04	0.0067	83.71

sample (a negligible rate with respect to the total bit rate). The normalized error samples were encoded using the codebook generated for the Gaussian source. The distortion of the quantized images were compared with the distortion of the images quantized using an optimum VQ. The objective measure for the coder performance used in this study is the mean square criterion. It refers to the average of the squares of the error between the original image and the reconstructed one. That is,

$$D = E[||\mathbf{x} - \hat{\mathbf{x}}||^2]. \tag{3.40}$$

The mean square error is expressed in terms of the Peak Signal-to-Noise Ratio (PSNR) which, for images with 8-bit pixel values, is defined as

$$PSNR = 10 \log \frac{(255)^2}{D}. \tag{3.41}$$

The result of some of tests are shown in Table 3.2 and 3.3. In most cases the differences were found to be less than 1 dB. For example, for the image *Lenna*, if

the first stage is an optimum VQ with dimension 2×2 and codebook size 1024, the difference is 0.1 dB. Similar results were observed for other images. To encode the images in the first stage, an optimum vector quantizer was used.

Three different examples of the reconstructed images using the two-stage optimum VQ and the Gaussian codebook shown in Figures 3.11 to Figure 3.13. For the image *Lenna* in Figure 3.11, a 4 dimensional VQ is used in the first-stage and the size of the codebook is 64. Figure 3.11a shows the reconstructed image when a universal Gaussian codebook is used in the second stage. Figure 3.11b shows the result of using an optimum VQ in the second stage. The dimension of vectors for both cases is 4×4 and the size of codebook in this stage is 256. For the image *Bridge* and the image *Lansat3* shown in Figures 3.12 and 3.13, the codebook size in the first-stage is 64. The codebook for the second-stage is the same as the one used for the image *Lenna*. As it can be observed, the images reconstructed by the two methods are very close.

3.7 MERITS

In the proposed method, a universal Gaussian codebook is used for the second stage. For quantizers with more than two-stages, the distortion is smaller in the later stages and the distribution of the error samples is closer to a normal distribution. Hence, for later stages the Gaussian codebook is even closer to an optimum codebook, and only a single generalized codebook needs to be designed and used for all images in different stages. This is a significant advantage of the proposed method in which, almost without loss of optimality, one universal codebook can be used in different stages for different sources.

Since the codebook designed for a Gaussian source is fixed, different structures

Table 3.2: Comparison of using an optimum codebook for second stage of a two-stage vector quantizer with universal Gaussian codebook for different images

Images	First stage			Second stage			
						Locally normalized	
	PSNR	bps	Dim	bps	Dim	Gaussian PSNR	Optimum PSNR
Lenna	33.83	1.5	2×2	0.5	2×2	35.32	36.02
	36.79	2.0	2×2	0.5	4×4	39.93	40.47
	39.69	2.5	2×2	0.5	4×4	41.29	42.18
Baboon	25.31	0.5	4×4	0.5	4×4	27.83	28.59
	28.95	1.5	2×2	1.0	2×2	33.79	33.92
	31.84	2.0	2×2	1.0	2×2	36.79	36.98
Bridge	28.5	1.5	2×2	0.5	4×4	31.06	31.68
	31.33	2.0	2×2	1.0	2×2	36.00	36.17
Giza	25.09	1.0	2×2	0.5	4×4	27.52	28.15
	28.69	1.5	2×2	0.5	4×4	31.25	31.78
Lansat3	27.28	1.5	2×2	0.5	4×4	29.87	30.18
	30.23	2.0	2×2	0.5	4×4	32.97	33.24

(a)

(b)

Figure 3.11: Comparison of the results for image *Lenna*, for bit rate 2 bps. (a) Reconstructed image quantized by the universal Gaussian codebook. (b) Reconstructed image quantized by an optimum codebook.

(a)

(b)

Figure 3.12: Comparison of the results for image *Bridge*, for bit rate 2 bps. (a) Reconstructed image quantized by the universal Gaussian codebook. (b) Reconstructed image quantized by an optimum codebook.

(a)

(b)

Figure 3.13: Comparison of the results for image *Lansat3*, for bit rate 2 bps. (a) Reconstructed image quantized by the universal Gaussian codebook. (b) Reconstructed image quantized by an optimum codebook.

Table 3.3: Comparison of using an optimum codebook for second and third stage of a residual vector quantizer with universal Gaussian codebook for different images

Images	First stage			Second stage				Third stage			
	PSNR	bps	Dim	bps	Dim	Gaus. PSNR	Opt. PSNR	bps	Dim	Gaus. PSNR	Opt. PSNR
Lenna	36.79	1.5	2× 2	0.5	4×4	39.93	40.47	0.5	4×4	40.51	41.4
Baboon	28.95	1.5	2× 2	0.5	4×4	32.12	32.48	0.5	4×4	33.52	33.76
Bridge	28.5	1.5	2× 2	0.5	4×4	31.06	31.68	0.5	4×4	33.20	33.76

can be imposed for reducing the complexity of the encoder. For instance, the code-vectors can be localized and the search can be started from the code-vectors which are closer to the origin, or a mapping can be carried out based on the energy of the vectors. Having one generalized codebook for all stages gives an opportunity to find some mathematical mapping between the source vectors and the codevectors which can considerably reduce the complexity of search. The distribution of the codebook is well known and this makes it possible to define a lossless entropy coding. Use of the entropy coding can reduce bit rate and makes this method more efficient.

3.8 SUMMARY

In this chapter, the idea of using a universal codebook for a multi-stage vector quantizer for image compression has been presented. It has been shown that the locally normalized error vectors of an image have a distribution close to a normal distribution. Since a memoryless Gaussian source is successively refinable, the error signal is successively refinable as well. As a consequence, the codebook designed for a memoryless Gaussian source can be used in different stages of a multi-stage

VQ to quantize the image error samples. An optimum codebook designed for a normally distributed source has been used to quantize the error samples of different images, and the results were compared with the reconstructed images quantized by an optimum VQ. The results were very close. In some cases the difference is less around 0.1 dB, but in general, the difference was found to be less than 1 dB. Since with the proposed method only one codebook is needed in different stages of a residual VQ, different structures and mapping techniques can be used to reduce the search complexity.

Chapter 4

TWO-STAGE RESIDUAL LATTICE-BASED VECTOR QUANTIZER

4.1 INTRODUCTION

As explained in the previous chapter, Vector Quantization (VQ) theory aims at achieving the highest VQ performance as a function of rate and dimension, but the application of a VQ is concerned with obtaining a high level of VQ performance at an affordable cost. The memory and computation which are required for VQ implementation, depend on the VQ rate, vector dimension, and the constraint imposed on the quantizer's structure. Imposing carefully selected structural constraints can reduce the complexity of a VQ.

A class of structured quantizers that reduce both memory and computation is the product code vector quantization. A product code vector quantizer is a structured VQ in which different components of the VQ quantize different features of the source. The gain-shape VQ and the residual VQ are two examples of product code VQ.

A residual vector quantizer is a simple product code VQ with a direct sum codebook structure and a sequential search procedure. The quantizer has a sequence of encoder stages where each stage encodes the residual vector of the previous stage. Residual VQ, similar to other structured VQs, is not able to provide performance as good as that of the unstructured VQ for a given rate and vector dimension, but it provides a better performance for a given complexity.

In Chapter 3, we discussed the condition of optimality of a multi-stage VQ. It was shown that in the limit for a class of memoryless sources and sources with memory, multi-stage VQ's codebook can be optimally designed, or in other words, the sources are successively refinable. It was also shown that for sources with memory, like images, the locally normalized error samples, defined by normalizing the residual samples by the magnitude of the zone to which the sample belongs, are

successively refinable. The effectiveness of the encoding method in Chapter 3 is dependent on the feasibility of using a large enough vector quantizer codebook in the first stage to obtain for low distortion a rate that is close to the lowest rate achievable by the rate-distortion theory. The computation and memory complexity required for unstructured VQ implementation to achieve this requirement limits the application of the proposed method.

Imposing an additional structure on the product code makes the code more amenable to sequential searches. Multi-stage VQ with a lattice structured codebook is such an example.

The lattice-based VQ, which is an extension of the uniform scalar quantization to the multi-dimensional case, offers some advantages over the classical vector quantization. It reduces the computational time for comparable performances and no memory is required to store the codebook. Due to the relative ease of lattice vector quantization, optimum encoding is feasible for moderate to large values of rates and vector dimensions.

To exploit most of the source memory, transform coding in the first stage can be used. Transform coding decorrelates the pixel values and distributes the energy among a small set of transform coefficients.

The work presented in this chapter is based on a two-stage residual lattice VQ. Two different schemes are presented. In the first one, each block is converted into DCT coefficients. The low-frequency coefficients are quantized using a high-rate Lattice-Based Vector Quantizer (LBVQ) in the first stage. In the second scheme, we use a low-rate JPEG encoder, for the first stage. For both schemes, in the second stage, the difference of the quantized image in the first stage and the original one is

quantized with an optimum VQ, an LBVQ, or a codebook designed for a memoryless Gaussian source. The results of three methods are compared. Although using LBVQ reduces the complexity of quantizer, encoding of the lattice points when the vector dimension increases is not a trivial task. Some efforts have been made in this direction, but still indexing of these points for boundaries other than cubical is still difficult. The enumeration method [15] introduced for indexing needs too many recursive computations. The indexing problem is discussed in Chapter 5.

In this chapter the energy of different coefficients, for some images are also presented and they are compared with the DCT coefficients of the error samples.

4.2 TWO-STAGE RESIDUAL LATTICE-BASED VQ

By dividing the quantization task into several successive stages, residual vector quantization achieves a great deal of savings in terms of storage and computational complexity. A residual vector quantizer consists of a cascade of VQ stages, where each stage operates on the residue of the previous stage. The codebook design suggested in [49] is based on a sequential design of each stage by using GLA. This method has been reported to provide a poor reproduction quality when the number of stages exceeds two [9]. Some algorithms have been introduced to improve the performance of residual VQ [9].

The block diagram of a residual VQ encoder is shown in Figure 3.1. It is based on successive quantizations of residual signals. A K-stage residual VQ, each with the codebook size M, can be uniquely represent M^K vectors with only $M.K$ codewords. This structure results in tremendous reduction in the codebook search and

the storage complexity. The overall encoding rate is $\frac{1}{N}\lceil K \log_2 M \rceil$ bits per sample, where $\lceil b \rceil$ denotes the smallest integer larger than b.

The codebook for the residual VQ can be designed in two ways. In the first method, the codebook in each stage is designed separately. For example, a generalized Lloyd algorithm [30] can be used to design each stage. Let $\mathbf{x}$ be the input vector, and $\hat{\mathbf{x}}_l$ the quantized error vector from stage l. The input to stage $l+1$ is given by

$$\mathbf{e}_l = \mathbf{x} - \sum_{i=1}^{l} \hat{\mathbf{x}}_i \tag{4.1}$$

Then, $(l+1)$th stage has to choose $\hat{\mathbf{x}}_{l+1} = \mathbf{y}_j$ to minimize the squared error given by

$$d_{l+1} = \|\mathbf{e}_l - \mathbf{y}_j\|^2 \qquad j = 1, 2, ..., M_l. \tag{4.2}$$

In the second method, (jointly optimum encoding), the indices of quantized error vectors are jointly selected.

For reducing the complexity of the encoder, a lattice-based VQ can be used in each stage. Because of the regular structure of an LBVQ, its use, in general, results in a drastic reduction in the complexity in comparison to an optimum VQ for the same bit rate and vector dimension. Although an LBVQ, similar to other types of structured VQs, is incapable of providing a performance as good as that of an optimum VQ for a given rate and dimension, it provides a good performance for a given memory and computational complexity. One reason for this is that by using structured quantizers, one can implement codes with large vector dimensions.

The lattice points form a subset of the Euclidean space $\Re^N$ which are uniformly distributed. Hence, using an LBVQ is optimum for uniformly distributed sources. However, LBVQs have also been used for Gaussian and Laplacian sources,

showing good performance [21] [16]. In [21], it has been shown that by an appropriate shaping of the support region of the codebook, an LBVQ offers the granular gain of the lattice codebook as well as the boundary gain.

The use of Lattice structured codebooks for non-uniform sources has been the subject of several investigations. For example, piecewise uniform LBVQ has been designed to produce the codebooks for a Gaussian and a Laplacian sources [19]. In this method, the scale-factor (step-size) is defined by the density of the input points in different areas.

The use of lattices with variable step-size has also been suggested in [23], [50]. A Scalar Vector Quantizer(SVQ) [23] is a fixed-rate entropy-coded scalar quantizer. An SVQ combined with trellis-coded quantization [50] provides an excellent fixed-rate encoding performance. Similar to other trellis encoding techniques, it involves a considerable encoding delay.

In image compression, the use of the above-mentioned quantization techniques in conjunction with a transformation yields a better performance. For a typical image, the values of the adjacent pixels are highly correlated. Transform coding uses this correlation between the neighboring pixels to achieve a considerable compression. The goal of transform coding is to decorrelate the pixel values. The result of transformation on the correlated image samples is that the signal energy is distributed among a small set of transform coefficients. Hence, in transform coding many coefficients with negligible information content are neglected. The number of the retained coefficients is a trade-off between distortion (quality of the retrieved image) and the compression rate. For most images, the Discrete Cosine Transform (DCT) is very close to an optimum transform (Karhunen-Loeve transform.) The DCT consists of cosine terms of different frequency components and results in a

spectral decomposition of the original image. Using Kolmogorov-Smirnov test, it has been shown that the DCT of an image has Gaussian dc components and Laplacian ac components [51]. This makes the combination of transform coding and geometric coding an efficient source coding scheme. Figures 4.1 - 4.2 show the distribution of some of the coefficients for different images. These coefficients are normalized by energy of each coefficient. For example, for the image *Bridge*, the distribution of the coefficients are quite close to the distribution of a Laplacian source.

Since most images have a low-pass power spectrum, the low-frequency coefficients are usually retained while the high-frequency coefficients are omitted. A major drawback of this method is that it is possible that some of the coefficients which are not in the coefficients retention set can have non-negligible energy, and by neglecting them, a great deal of information could be lost.

Tables 4.1 - 4.3 show the energy of the coefficients of images *Lenna*, *Bridge* and *Light* respectively. For some images with a plain background, the energy of the high-frequency coefficients is not too large. As it can be seen from Table 4.1, for image *Lenna*, the ratio of the energy of the low-frequency coefficients and that of the high-frequency coefficients is more than 1000. However, for the image *Light* (Table 4.3), this ratio is reduced to less than 7. For these images, neglecting the high-frequency transform coefficients is not effective. It can be observed that even if a lossless quantizer is used for encoding of a fixed number of low-frequency coefficients, there is a limitation for improving the quality of the image. For example, even in an 8-bit image *Lenna*, if 15 lowest-frequency coefficients are chosen from an 8×8 blocks, the maximum achievable PSNR is 33.38 dB.

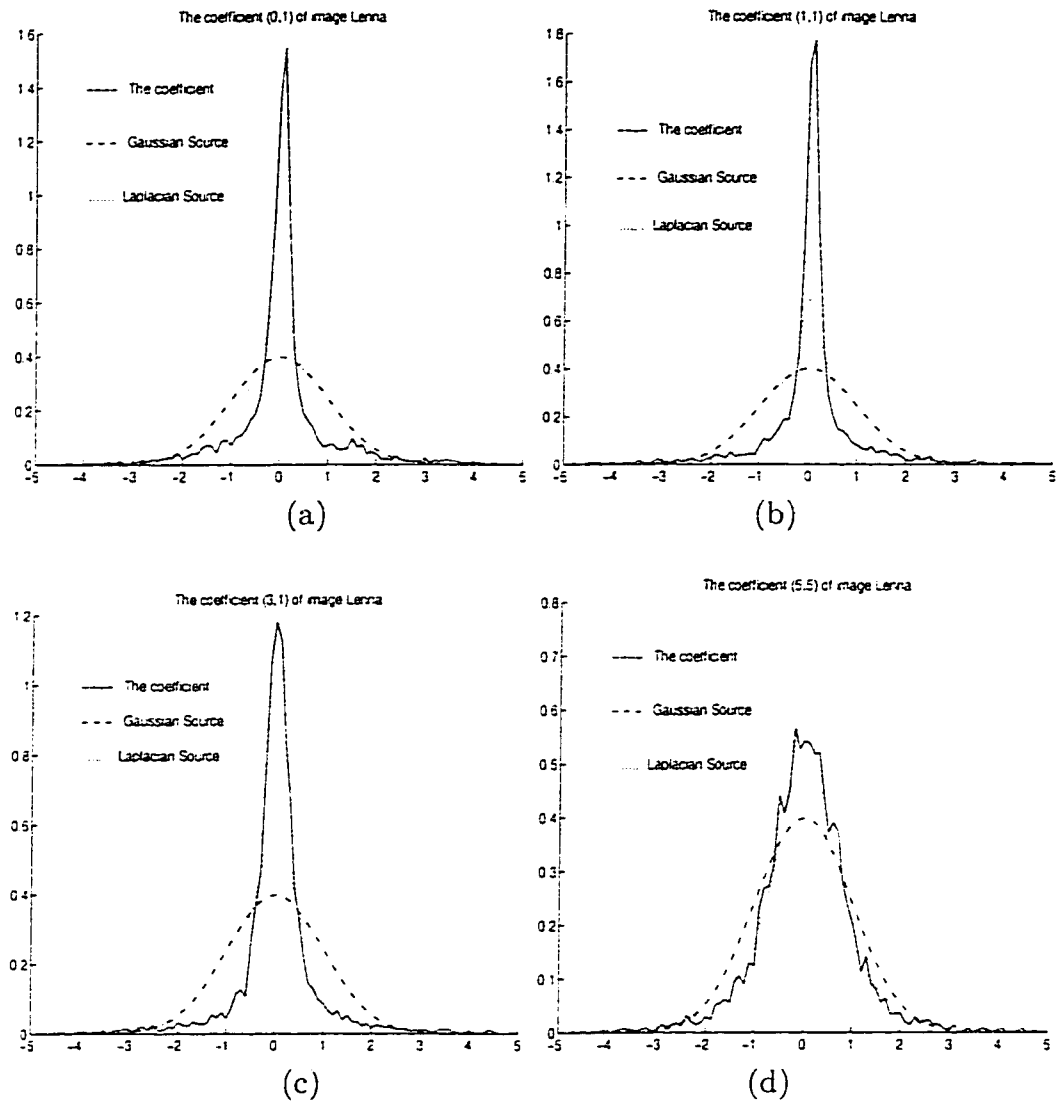

Figure 4.1: Distribution of some DCT coefficients of the image *Lenna*. (a) Coef.(0,1). (b) Coef.(1,1) (c) Coef.(3,1). (d) Coef.(5,5).

Table 4.1: The energy of the coefficients for the image *Lenna*

-0.0	7371.3	1599.3	472.7	212.4	96.0	44.3	25.8
2844.3	1366.9	620.0	300.7	114.6	61.1	34.1	19.3
447.6	459.0	354.4	173.8	84.1	48.7	22.8	15.2
136.6	133.2	123.1	86.1	53.2	29.4	18.7	11.6
47.2	46.0	44.9	43.3	28.9	18.6	11.8	9.8
21.4	20.9	21.2	19.5	15.6	11.4	9.6	8.3
12.2	11.8	10.7	11.1	10.3	8.5	7.3	6.6
9.2	8.7	7.7	7.7	7.5	7.2	6.3	5.7

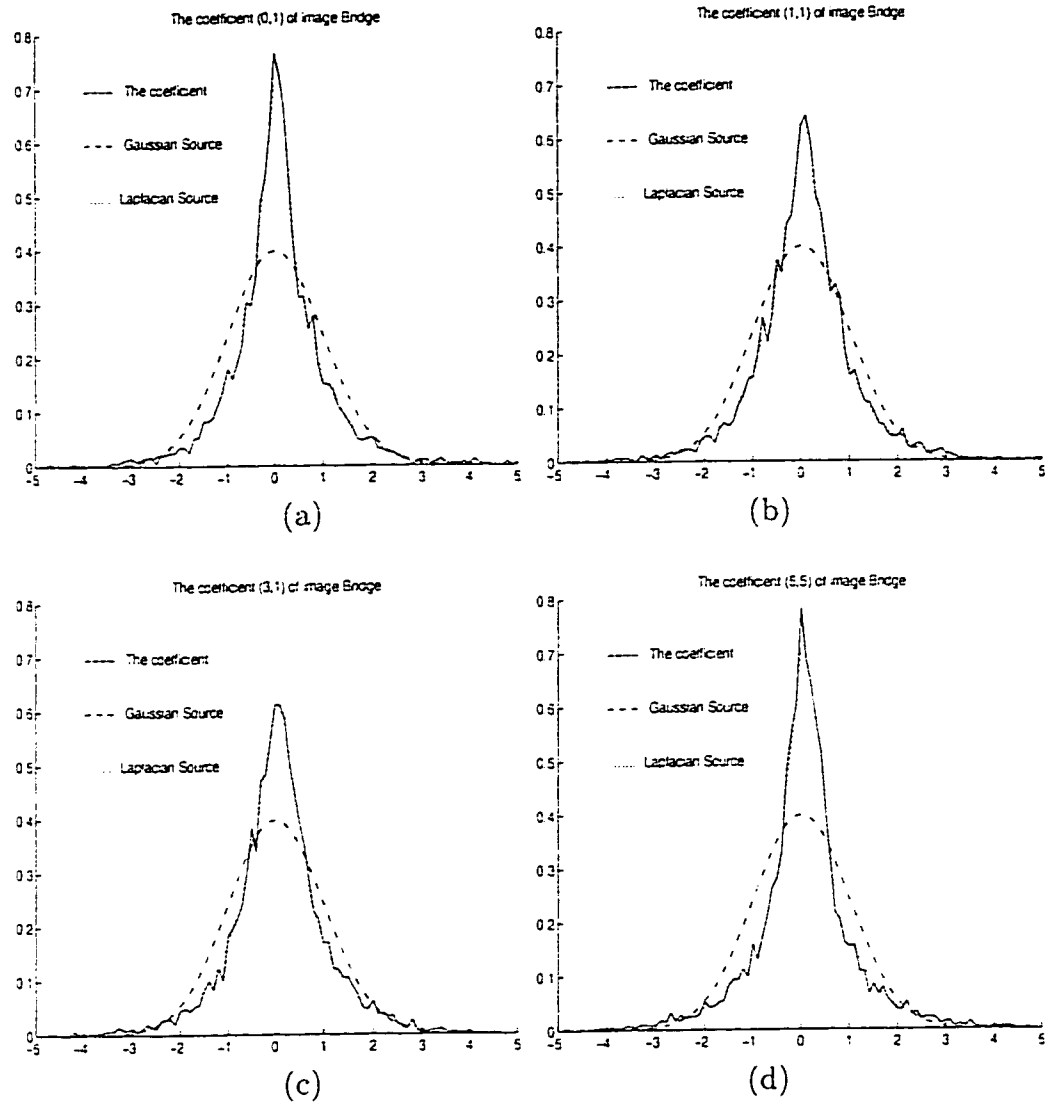

Figure 4.2: Distribution of some DCT coefficients of the image *Bridge*. (a) Coef.(0,1). (b) Coef.(1,1) (c) Coef.(3,1). (d) Coef.(5,5).

Table 4.2: The energy of the coefficients for the image *Bridge*

0.0	5523.9	1988.1	920.5	513.3	298.9	200.7	146.1
9524.3	2001.3	1012.7	555.2	348.1	211.6	149.6	113.1
3817.7	1115.2	668.1	408.4	269.5	173.0	127.4	97.8
1685.9	646.0	443.3	299.5	215.4	141.0	105.3	80.2
807.2	373.1	272.4	220.8	161.5	111.5	88.1	71.6
475.1	237.5	184.3	143.8	114.2	88.1	72.3	56.5
323.0	158.7	138.8	103.3	86.3	65.9	53.7	46.4
195.3	121.4	99.0	78.8	70.4	58.6	46.2	38.4

Table 4.3: The energy of the coefficients for the image *Light*

-0.0	4867.3	2315.0	1566.4	1283.7	1160.3	875.8	796.7
3606.4	2089.4	1465.9	1262.3	1130.3	1047.1	930.8	820.3
1687.4	1300.0	1193.5	1095.7	1147.3	981.9	912.9	796.3
1167.9	1031.8	1007.5	990.9	1022.6	979.4	838.4	829.9
923.5	877.1	928.8	927.0	955.7	941.6	891.4	805.6
855.9	836.8	869.3	948.1	981.0	918.3	839.2	830.5
807.9	796.6	830.1	913.3	953.2	926.3	900.5	808.7
714.3	727.6	831.0	984.9	972.4	939.2	874.3	764.4

For retaining the high-frequency information, many techniques have been developed some of which were explained in Section 2.2. In order to retain more information that is contained in an image, we now propose the use of a two-stage quantizer. The function of the first stage is to encode the more important low-pass components of the image. The second stage encodes the high-frequency components ignored in the first stage. Since the correlation between the image pixels at the input of the first stage is high, a transform coding scheme is appropriate for this stage.

Tables 4.4 - 4.6 show the energy of the various DCT coefficients of the residual images after quantizing the low-frequency coefficients. It can be observed that even though the low-frequency coefficients are quantized in the first stage, in some images, the low-frequency coefficients of the error signal have a considerable amount of energy. These coefficients are also quantized one more time in the second stage.

We present two versions of the proposed algorithm. In the first version a DCT transform coding scheme along with an LBVQ for the first stage (Figure 4.3) is used, while in the other version as shown in Figure 4.4 a standard JPEG encoder is used for the first stage.

In either case the second stage works on an "error" or residual image formed

Table 4.4: The energy of the coefficients for the residual image *Lenna*

-0.0	327.4	108.0	71.9	47.8	96.0	44.3	25.8
167.2	89.1	74.1	54.2	114.6	61.1	34.1	19.3
62.3	63.4	54.6	173.8	84.1	48.7	22.8	15.2
41.1	42.5	123.1	86.1	53.2	29.4	18.7	11.6
28.7	46.0	44.9	43.3	28.9	18.6	11.8	9.8
21.4	20.9	21.2	19.5	15.6	11.4	9.6	8.3
12.2	11.8	10.7	11.1	10.3	8.5	7.3	6.6
9.2	8.7	7.7	7.7	7.5	7.2	6.3	5.7

Table 4.5: The energy of the coefficients for the residual image *Bridge*

0.0	231.9	127.1	110.9	96.9	298.9	200.7	146.1
440.4	133.3	115.2	100.7	348.1	211.6	149.6	113.1
194.0	120.8	106.7	408.4	269.5	173.0	127.4	97.8
135.7	115.0	443.3	299.5	215.4	141.0	105.3	80.2
116.3	373.1	272.4	220.8	161.5	111.5	88.1	71.6
475.1	237.5	184.3	143.8	114.2	88.1	72.3	56.5
323.0	158.7	138.8	103.3	86.3	65.9	53.7	46.4
195.3	121.4	99.0	78.8	70.4	58.6	46.2	38.4

Table 4.6: The energy of the coefficients for the residual image *Light*

0.0	225.9	130.6	110.4	113.1	1160.3	875.8	796.7
200.2	133.9	114.9	111.9	1130.3	1047.1	930.8	820.3
123.6	114.0	108.2	1095.7	1147.3	981.9	912.9	796.3
116.0	107.5	1007.5	990.9	1022.6	979.4	838.4	829.9
109.4	877.1	928.8	927.0	955.7	941.6	891.4	805.6
855.9	836.8	869.3	948.1	981.0	918.3	839.2	830.5
807.9	796.6	830.1	913.3	953.2	926.3	900.5	808.7
714.3	727.6	831.0	984.9	972.4	939.2	874.3	764.4

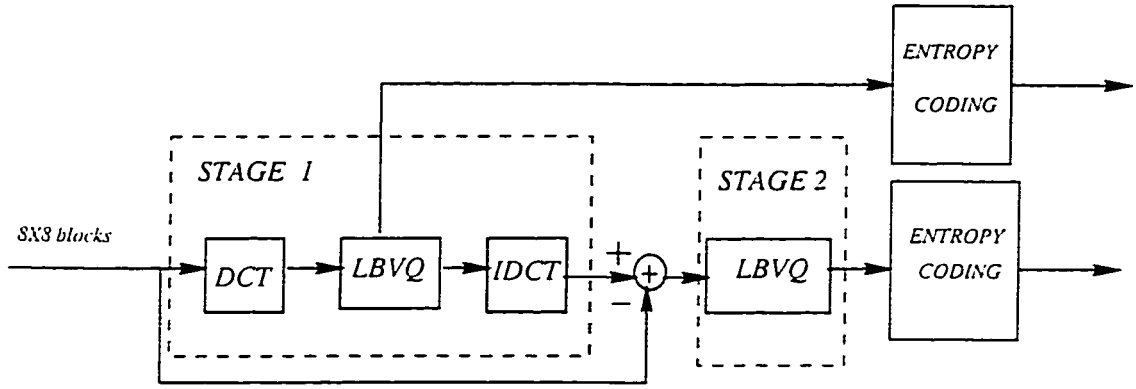

Figure 4.3: The block diagram of the two-stage residual VQ using transform coding for the first stage.

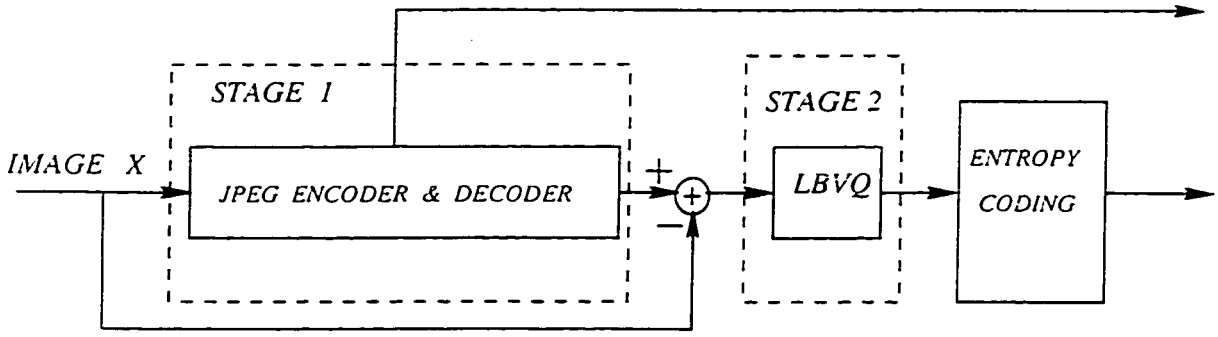

Figure 4.4: The block diagram of the two-stage residual VQ using JPEG for the first stage.

by subtracting the output of the first stage from the original image. Since the correlation between pixels of the "error" image is low, a compression scheme suitable for a memoryless source would suffice. To observe this, the distribution of the error signals are found. Since the bit rate in the first stage is low, the distribution of the error signal is far from a Gaussian distribution. To see the effect of the quantizer on the distribution of the error samples, the first 14 coefficients are kept, and the rest set to zero. Then, the distribution of error samples are plotted. As shown in Figures 4.5 - 4.7 even if a fine quantizer is used in the first stage, the distribution does not fit to that of a Gaussian source. The differences are more, in the case of having a lossy quantizer in the first stage. Thus, using a Gaussian codebook in this case is not very effective. An optimum VQ, a Gaussian codebook and an LBVQ is used for the second stage. Using a lattice-based vector quantizer for the second stage gives a better performance. This is due to the simplicity of the search in lattice, that allows us to have a high-dimensional VQ. The results are presented in the next section.

4.3 SIMULATION AND RESULTS

The proposed two-stage LBVQ is applied to the images *Light*, *Bridge*, *Baboon*, *Lenna*, each of size 512 × 512, and the bit rate is compared with the standard JPEG. The objective measure for the coder performance used in this section is the mean square criterion. It refers to the average of the squares of the error between the original image and the reconstructed one. The mean square error is expressed in terms of the Peak Signal-to-Noise Ratio (PSNR).

To simulate the first version of the proposed method (Scheme 1), each image is partitioned into 8 × 8 blocks and DCT is computed over each block. The DC coefficients are quantized separately. Because of the strong correlation between the DC components of the adjacent blocks, differential pulse code modulation is used

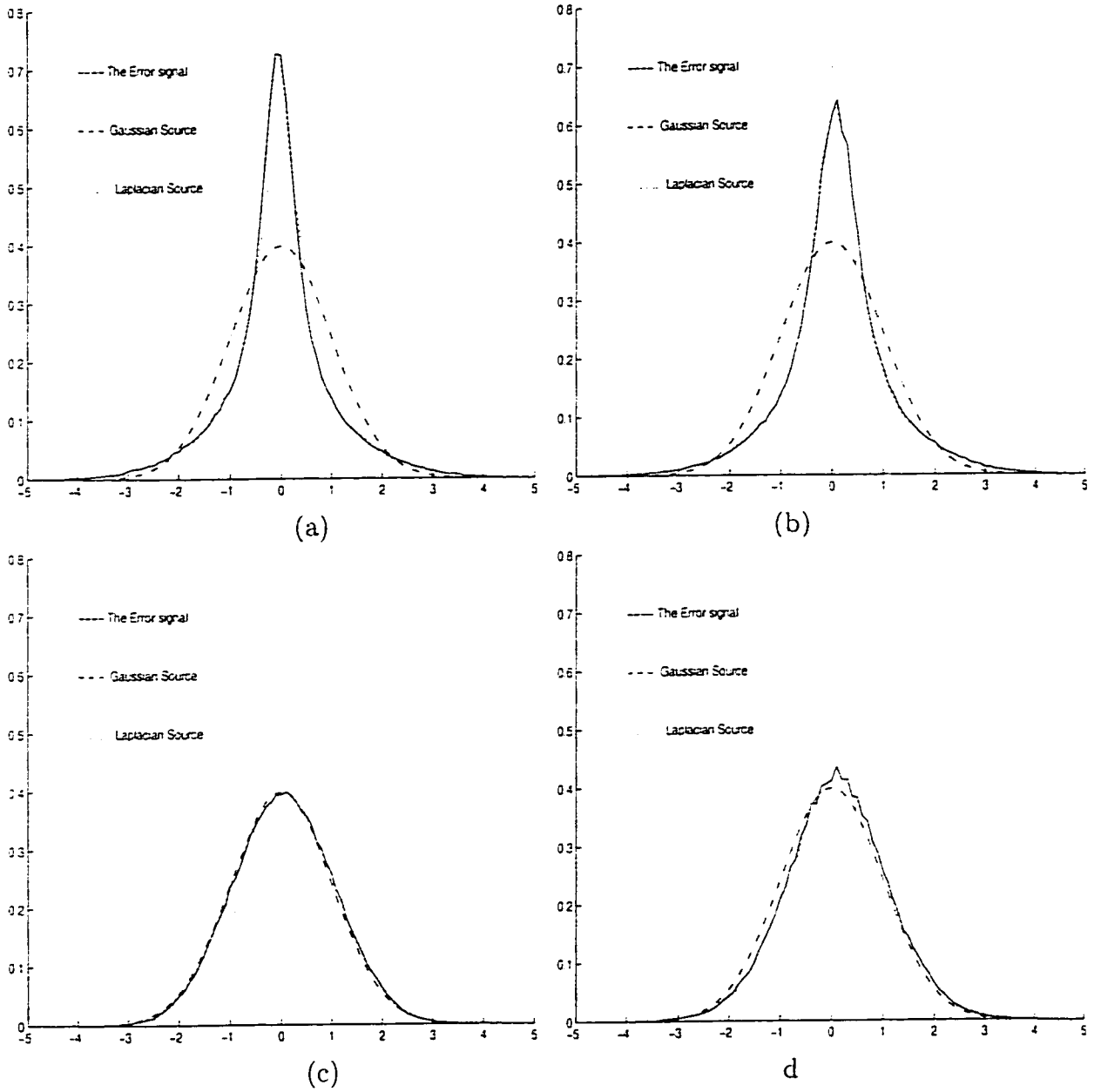

Figure 4.5: Distribution of error signal for the image *Baboon* with the first 15 DCT coefficients quantized. (a) Lossless quantizer. (b) LBVQ. (c) Locally normalized error signal from lossless quantizer. (d) Locally normalized error signal from LBVQ.

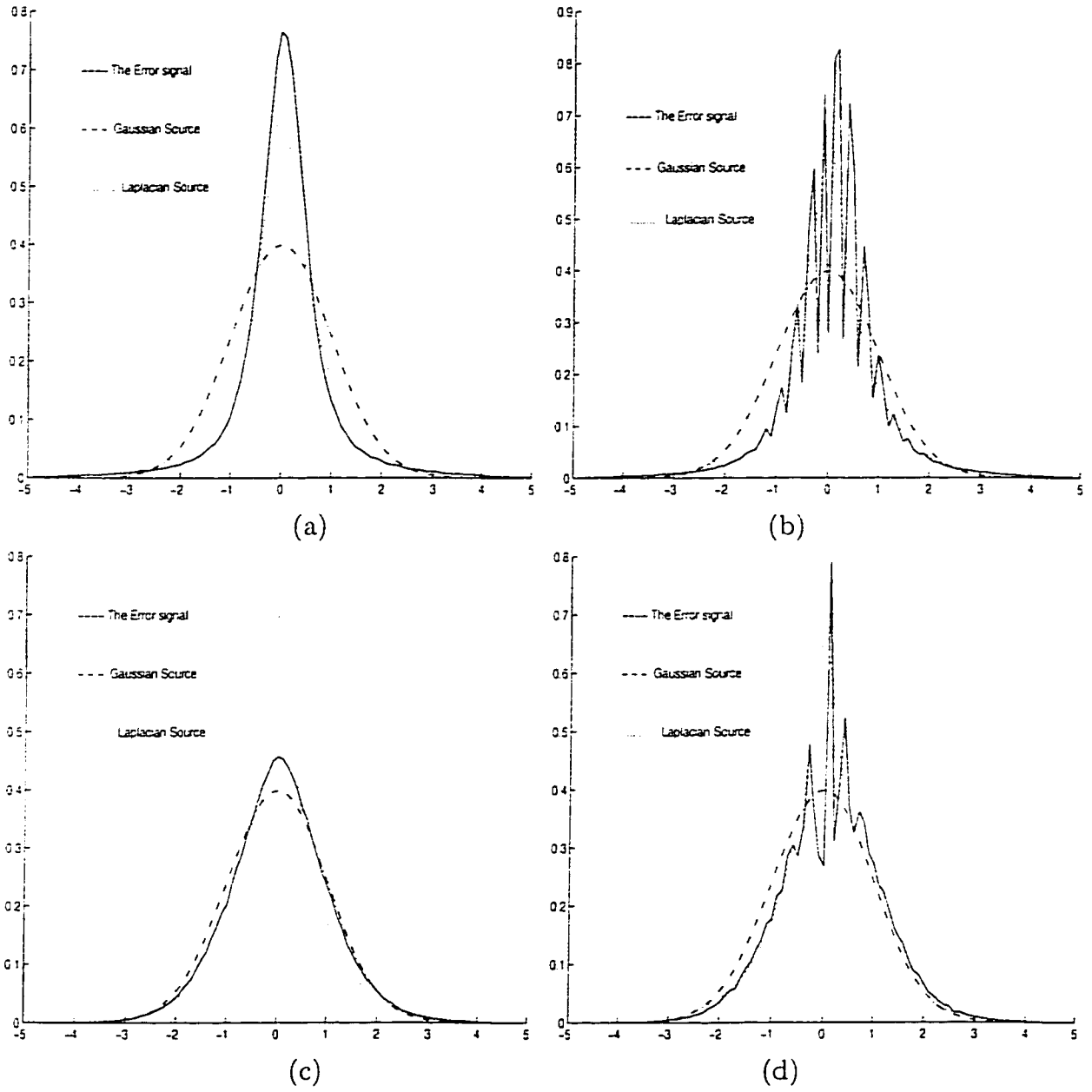

Figure 4.6: Distribution of error signal for the image *Lenna* with the first 15 DCT coefficients quantized. (a) Lossless quantizer. (b) LBVQ. (c) Locally normalized error signal from lossless quantizer. (d) Locally normalized error signal from LBVQ.

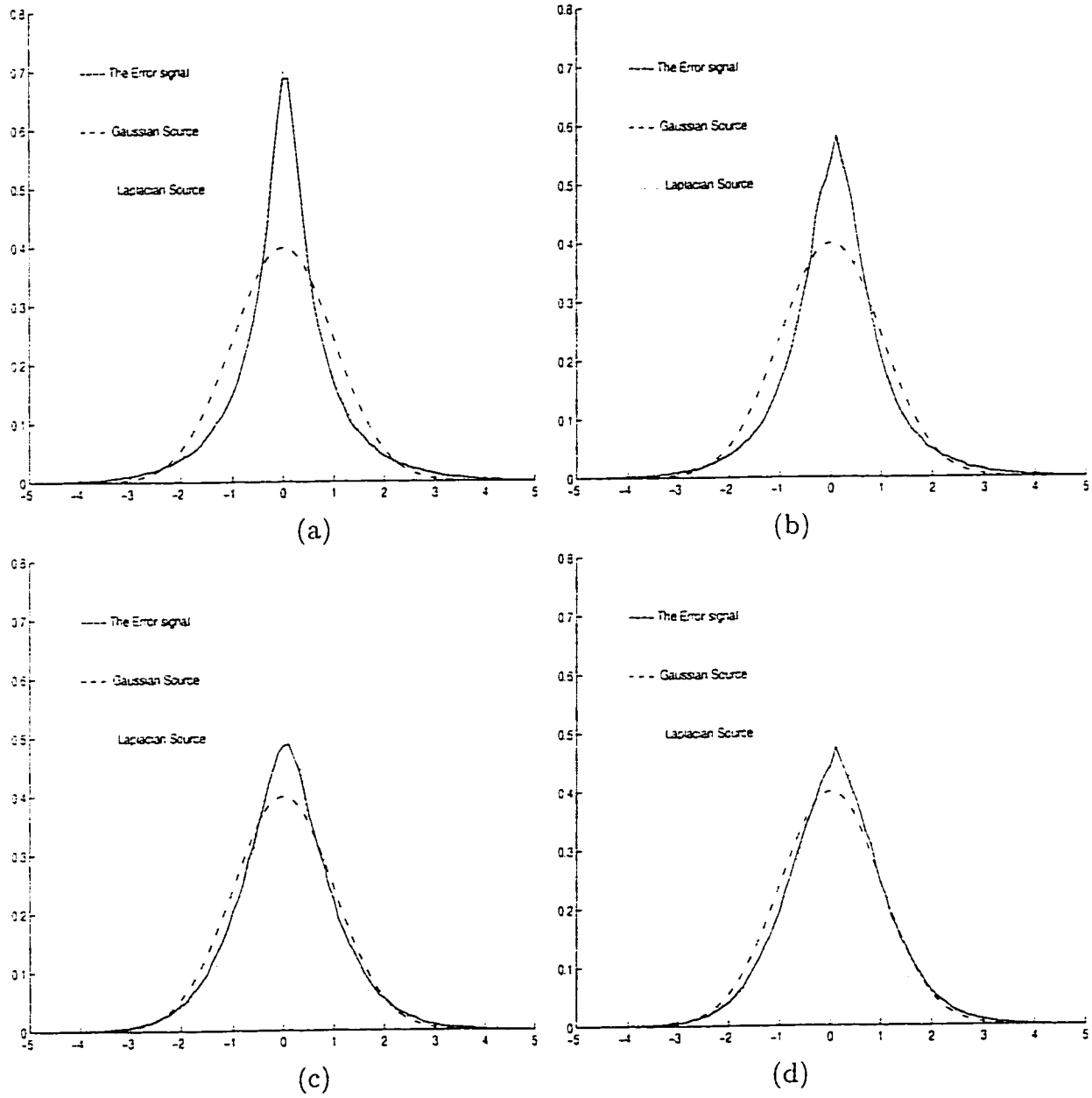

Figure 4.7: Distribution of error signal for the image *Bridge* with the first 15 DCT coefficients quantized. (a) Lossless quantizer. (b) LBVQ. (c) Locally normalized error signal from lossless quantizer. (d) Locally normalized error signal from LBVQ.

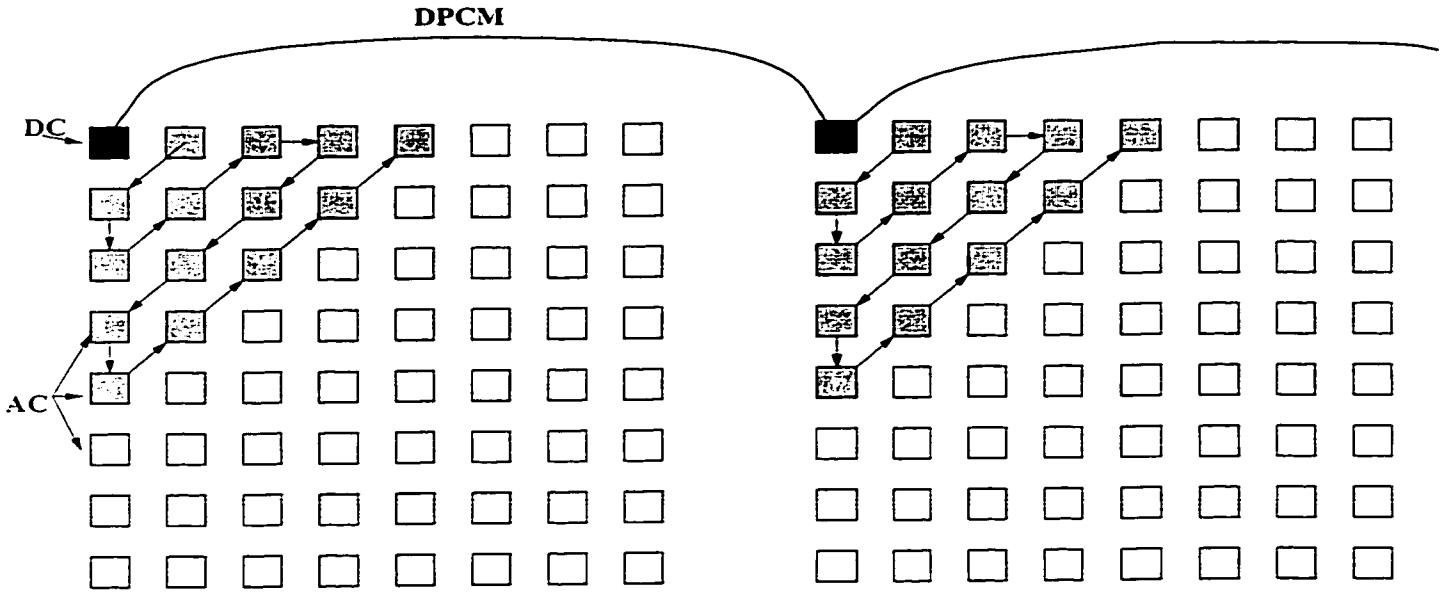

Figure 4.8: The quantizing order of DCT coefficients in the first stage.

for quantizing them. In the first stage a fraction of AC coefficients are quantized. As shown in Fig (Figure 4.8), the first 14 AC coefficients are chosen in a zig-zag order. As a result, the coefficients corresponding to the 14 lowest frequencies (i.e. the information contained in the static region of the image) are quantized. The rest of the coefficients are set to zero.

For quantizing the DCT coefficients, a lattice-based VQ is used. We use the cubic lattice z^{16} with a spherical contour for truncating. The scale factor is obtained using an iterative algorithm. The encoder uses a fast quantization technique due to Conway and Sloane [33] for finding the nearest lattice point for each vector. To calculate the bit rate in this stage, two methods are used. In the first method, a theoretical entropy coding is assumed and in the second method, the bit rate is estimated by the number of lattice points on each hyper-sphere and the number of points falling on that sphere after quantization.

In the second stage, the difference between the original image and the one

Table 4.7: Performance comparison of optimum VQ and Gaussian codebook in the second stage

Image	First stage LBVQ		Second stage					
			Optimum VQ			Gaussian-codebook		
	PSNR	bps	DIM	bps	PSNR	DIM	bps	PSNR
Lenna	31.76	0.106	4×4	0.25	33.44	4×4	0.5	31.93
			4×4	0.5	35.81			
			2×2	1.0	36.14			
Bridge	26.04	0.181	4×4	0.5	28.69	4×4	0.5	26.15
			2×2	1.0	30.078			
Baboon	25.02	0.181	4×4	0.5	28.69	4×4	0.5	25.11
			4×4	0.25	26.67			
			2×2	1.0	29.4			
Light	19.53	0.163	4×4	0.25	19.9	4×4	0.5	19.53
			4×4	0.5	20.89			
			2×2	1.0	23.84			

Table 4.8: Performance comparison of LBVQ in the second stage and JPEG

Image	Proposed Scheme 1							JPEG	
	First stage		Second stage		Total				
	Ent. bps	Enum. bps	Ent. bps	Enum. bps	Ent. bps	Enum. bps	PSNR dB	Rate bps	PSNR dB
Lenna	0.106		0.98		1.2		40.84	1.86	40.68
	0.080	0.16	0.80	1.18	1.0	1.3	35.70	0.66	35.76
Bridge	0.181		0.99		1.3		38.05		
	0.169	0.37	0.94	1.50	1.2	1.9	31.98	1.93	32.05
Baboon	0.181		0.99		1.4		37.26		
	0.168	0.40	0.90	1.61	1.2	2.1	33.00	1.94	31.77
Light	0.160		0.93		1.2		32.28	3.06	31.06
	0.150	0.36	0.87	1.57	1.1	2.0	27.29	2.50	27.89

Ent. denotes the result of entropy coding and Enum. the result of enumeration method.

Table 4.9: Performance comparison of two-stage VQ (Scheme 1) using entropy coding and JPEG

PSNR	Radius		bps	
dB	First stage	Second stage	Two-stage VQ	JPEG
40.56	9	9	1.2	1.8
39.4	9	7	1.13	1.41
38.25	4	9	1.05	1.15
36.7	9	4	0.98	0.83

quantized from the first-stage is quantized. In this stage, an optimum VQ and a Gaussian codebook are used to quantize the error image. Table 4.7 compares the results of the two methods of using the optimum VQ and Gaussian codebook for the second stage. As it was expected from the distribution of the error image, using the Gaussian codebook is not effective and does not result in a good performance. In the second stage, a lattice-based VQ is also used, and the lattice points are truncated as those bounded by a sphere. The bit rate is estimated by theoretically as well as by applying an enumeration method. The simulation results are shown in Table 4.8 in which the bit rates as obtained by using the proposed two-stage RVQ and the standard JPEG are depicted for various PSNRs. As it is seen from this table, the result of the proposed method is better than that of the JPEG except for image *Lenna* for which the performance using JPEG is better for PSNR = 35.7dB. Table 4.9 compares the performance of the proposed method for the image*Lenna* and that of JPEG for more extensive value of PSNR. It can be seen that two-stage VQ shows better results for PSNRs more than 38 dB. As seen in Table 4.1, for the image *Lenna*, most of the energy is distributed among the low-frequency coefficients. By neglecting the high-frequency coefficients, the JPEG still has a good performance. However, for other images the high-frequency coefficients are not negligible.

In the other version of the algorithm (Scheme 2), all the methods used in scheme 1 are again applied to the residual image obtained by using the standard

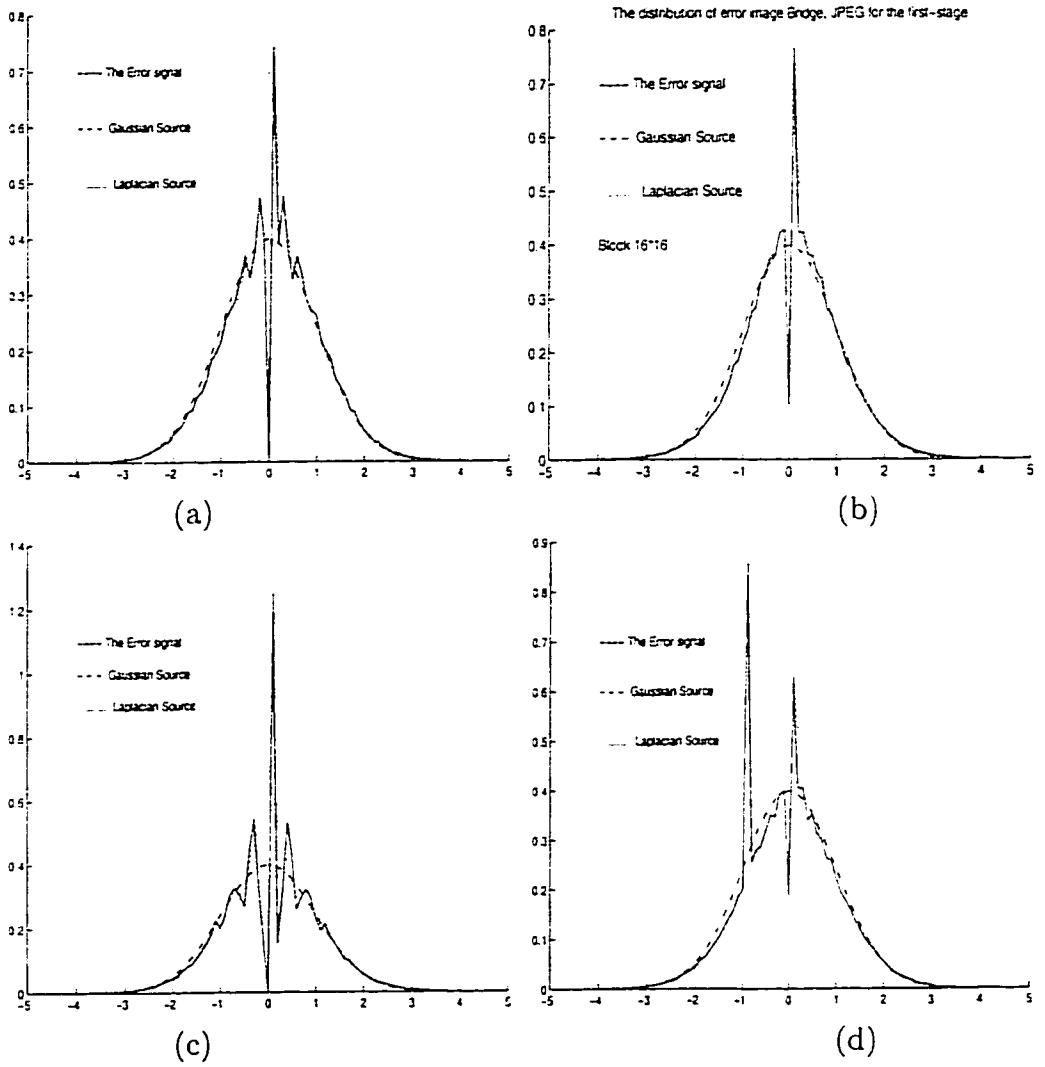

Figure 4.9: Distribution of the error signal for some images when the first stage is a 50% JPEG. (a) Image *Baboon*. (b) Image *Bridge*. (c) Image *Lenna*. (d) Image *Light*.

JPEG with a low rate as the quantizer of the first stage. Figure 4.9 shows the distribution of the error image which is the difference of quantized image obtained by using a JPEG compression technique and the original image. As it can be observed, the distribution of error samples is far from a Gaussian distribution. For this version, only an LBVQ is used as a quantizer in the second stage. The results for different images are shown in Table 4.10. Using the LBVQ for two stages shows a superior result compared to the JPEG. Even the use of the enumeration technique to calculate the bit rate results in a better performance. Table 4.11 compares the result of this scheme with that of the JPEG for only the image *Lenna*. The results in this table are calculated by an approximation of the the entropy coding. From this table, it can be seen that for moderate to high bit rate the rate achieved by applying the proposed scheme is considerably better than that obtained by using the JPEG for the same quality coded image.

For lower rates, the performance of JPEG is better than that of the proposed schemes. For example, a PSNR of 36.0 dB can be achieved at a rate of 0.7 bps using the JPEG. The same PSNR is achieved with the proposed Scheme 2 at a rate of 1.05. This is due to the fact that for lower values of PSNR, almost all the savings in the bit rate comes from the first stage. The bit rate of the second stage remains almost a constant, since the error signal on which the second stage operates is uncorrelated, and the entropy coding in this case is not very effective. This can be seen from the third column of Table 4.11.

In the first row of Table 4.11, JPEG compression with the quality value (defined by XVIEW which determines the compression rate) of 50% is used. The output of the second stage is equivalent to the output of a standard JPEG when the quality value is more than 97%. Such a quality value can be achieved by the JPEG with a rate of 3.9 bps. In our algorithm the second stage needs only one additional bit per

Table 4.10: Performance comparison of Two-stage VQ (JPEG in the first stage and LBVQ in the second stage) and JPEG

Image	First stage		Second stage		Total		JPEG	
	JPEG	PSNR	Ent.	Enum.	PSNR		PSNR	Rate
	bps	dB	bps	bps	dB	bps	dB	bps
Lenna	0.66	35.77	0.187	0.57	40.11	1.23	40.35	1.73
Bridge	1.3	29.54	0.186	0.57	33.80	1.87	33.29	2.22
Baboon	1.28	28.96	0.187	0.58	33.00	1.86	32.95	2.2
Light	1.84	24.36	0.181	0.50	29.73	2.34	29.26	2.74

Ent. denotes the result of entropy coding and Enum. the result of enumeration method.

Table 4.11: Performance comparison of JPEG and two-stage RVQ (Scheme 2)

PSNR dB	Two-stage RVQ			JPEG
	First stage(JPEG)	Second stage	Total	
	bps	bps	bps	bps
47.73	0.66	0.94	1.6	3.9
43.66	0.37	1.05	1.42	3.0
41.62	0.25	0.98	1.23	2.15
36.0	0.14	0.9	1.05	0.7

sample with total rate of 1.6 bps to achieve this result.

4.4 SUMMARY

In this chapter, we have proposed a two-stage residual quantization method for image compression. Two schemes have been presented. In the first scheme, an LBVQ is used to quantize the low-frequency transform coefficients. In the second one the standard JPEG is used to quantize the input image. It has been shown that the error signal does not have a distribution close to Gaussian. In both schemes, a high-rate LBVQ has been applied to quantize the residual signals comprising the

difference between the original image and the reconstructed image in the first stage. The results have been compared with the standard JPEG, showing an improvement of upto 2 bits for high-bit rate compression.

Chapter 5

INDEXING OF LBVQ USED IN TRANSFORM CODING

5.1 INTRODUCTION

Encoding the Discrete Cosine Transform (DCT) coefficients of an image involves two steps: quantizing and indexing the quantized points (Figure 2.18). For the first step, many lossy scalar and vector quantization techniques have been designed. For the second step, depending on the quantizer, different noiseless coding schemes are used to index the output points of the quantizer.

Due to the regular structure of lattices, many researchers have used lattice-based vector quantizer for quantizing DCT coefficients of images, but only a few methods have been suggested for indexing the output points [15]. Fischer [29] has combined an Lattice-Based Vector Quantization (LBVQ) with a noiseless code to encode the DCT coefficients of images. In [15], the output lattice points are labeled using an enumeration method for Laplacian sources, and it has been shown that the combination of LBVQ and noiseless code outperforms the uniform scalar quantizer combined with a noiseless coding for each coefficient. Fischer has also shown that the result can be further improved by using several quantizers [29]. The problem with the enumeration method is that full enumeration requires too many recursive calculations. This can be avoided if the mapped points are localized.

Due the asymptotic equipartition property of random variables, any sequence of blocks gets divided into two sets, typical set and non-typical set. For a sufficiently large dimension, the typical set has a probability close to 1, and all of its elements are nearly equiprobable. According to this property, the DCT coefficients of an image can be localized and the high probability area can be found.

In this chapter, an LBVQ is used to quantize the DCT coefficients of images. For reducing the effective bit rate, first, the output points are grouped according to the different parameters of blocks, which correspond to the probability density

function of the blocks. Then, shorter representations are assigned to more frequently used lattice points. Grouping is done in two ways. In the first method, the output points are grouped depending on the number of their non-zero components and their values. These output points are indexed with respect to their groups and the positions of non-zero elements in their respective blocks. In the second method, the output points are grouped according to a radial parameter defined by

$$r = \sum_{i=0}^{N-1} |x_i|^{\nu} \tag{5.1}$$

where $\nu = 2$ for spherical boundary and $\nu = 1$ for a pyramid boundary. In this work, spherical boundary is used, i.e. $\nu = 2$. Only the output points on the most probable spheres are indexed using the enumeration method. Since these spheres have small radii, the number of points on them is not too large, thus making their enumeration not too difficult. For the indexing of the points on the spheres with large radii, the positions of the non-zero elements and their values are used. We use a prefix variable length code to index these values.

5.2 LATTICE-BASED VQ

For a vector quantizer, the image samples are segmented into M blocks and the pixels in each block are considered as a vector. In an optimum vector quantizer, most of the output vectors belong to the typical set. In fact, with the iteration method such as Generalized Lloyd Algorithm (GLA), the codevectors are mostly concentrated in the typical set, $\mathcal{A}_{\epsilon}^{(N)}$. As it was mentioned in Section 2.5, the asymptotic equipartition property is valid when the dimension is large. The problem with an unstructured VQ is that the complexity of the quantizer increases exponentially as the the block dimension increases. Using lattice points as a codebook can solve this problem.

Usually in a lattice-based vector quantizer, the lattice is truncated such that the desired number of lattice points fall inside the truncated boundary. For example, for a given dimension N and a bit rate R, 2^{NR} is the number of lattice points that are used. As a result, for a large value of N the radius of truncation is small, so different values that each pixel can assume is limited to two or three levels. For instance, for a cubic lattice when the dimension is 16 and the bit rate is 0.5 bps the codebook size is 2^8. If the lattice points are truncated with a spherical boundary, the radius of truncation has to be chosen such that 2^8 points (code words) fall inside the boundary. In this case, the 16-dimensional lattice has to be truncated with a sphere of radius 2.

$$\sum_{i=1}^{16} x_i^2 = 4$$

It means that there are only 5 different levels given by (-2,-1,0,1,2).

For a source with a given probability density function, only a few of these lattice points are used. For example, in an image, where the correlation between the adjacent pixels is high, most of the output points are near the hyper-plane bisectors. Another illustration of this fact is that the DCT coefficients of an image are concentrated near the origin or axes. To take advantage of these regularities, a geometric vector quantizer has been suggested [15], [22]. It is known [15] that almost all codewords lie in the high probability region specified by the entropy of the source. The geometrical shape of the region of high probability depends on the source statistics. For example, these shapes are spheres for the memoryless Gaussian source, pyramids for the Laplacian source, and hypercubes for a uniform source. The probability density function is constant and, therefore, the codewords are uniformly distributed in this region. This is the idea behind geometric source coding. The intersection of the lattice points and the region of high probability for the source is chosen as the codebook. As a result, with simple encoding and decoding algorithms, this approach yields a good VQ for memoryless Gaussian, Laplacian and uniform sources.

Using Kolmogorov-Smirnov test, it has been shown that the DCT of an image, computed block wise, has Gaussian dc components and Laplacian ac components [51]. This makes the combination of the transform coding and the geometric coding an efficient source coding scheme.

Since the quantization step for an LBVQ is simple, using lattices for the high-dimensional VQ is possible, and according to the asymptotic equipartition property, in high dimension the output points are localized in the typical set. Hence, using an LBVQ in high dimension is a promising scheme for data compression. In order to have a good quality image, we propose a high-dimensional LBVQ with a large radius of truncation. However, the indexing of the lattice points, even for the low dimension is still a problem. Efficient algorithms exist for implementing a lattice quantizer with an N-dimensional hypercube boundary. In this case, indexing can be done by using one-dimensional code components over a bounded interval. However, for other desirable boundaries, such as spherical or pyramid, indexing requires an excessive storage or complex enumeration algorithms. In this work, we present a method for indexing the lattice points used as codewords of an LBVQ.

5.3 PRINCIPLE OF THE PROPOSED METHOD

In order to achieve a high-quality and low-complexity source coding, the lattice points are truncated with a large enough radius, making a large number of lattice points to fall inside the boundary. The problem of high-bit rate due to this large number of points is resolved by assigning a shorter representation to more frequently used lattice points. Grouping is done in two ways. In one method, all the output points are grouped depending on the number of their non-zero components. The output points are indexed with respect to their groups and the position of non-zero

elements in each group. In the other method, the output points are grouped based on the radial parameter, and a prefixed coding is used to index the output points.

5.3.1 Method based on grouping according to non-zero values

The correlation between the adjacent pixels of a typical image is high. As a result, if we divide the whole image into small blocks, usually there will not be significant changes in the pixel values of one block. This explains the concentration of the DCT coefficients of a typical image near the origin or axes. As a result, after scaling, there are only a few non-zero components in each block. These non-zero elements are the basis for indexing each block. In an LBVQ, the infinite lattice is truncated with a defined boundary. Here, we use a spherical boundary for truncation. Using an iterative procedure, the scale factor is selected such that the average distortion is minimized.

Depending on the radius of truncation, the components of each output vector can take only a few values. For example, if the radius of truncation is 9 in lattice z^{16}, symbols can only take values 0 to 9. We group the output points according to the number of their non-zero elements and the absolute value of these elements. For instance, the group with only two 1's and fourteen zeros includes vectors such as [1000100....0], [00.. − 10.. − 10..0] and [010..0..0.. − 1]. Our simulation results indicate that, if the DCT coefficients of an image, e.g., *Lenna* , are quantized with a cubic lattice z^{16} truncated with radius 9, there will be around 500 groups. Forty four per cent of the output vectors are mapped into the origin, sixteen per cent of the points are encoded into the vectors having a single 1 and fifteen 0's, and 6 per cent have two 1's and fourteen 0's. In more than fifty percent of the 500 groups, only one block is encoded. Table 5.1 shows some groups of the DCT coefficients of the image *Lenna* , and the number of blocks in each group. The total number of

Table 5.1: Selected groups for the image *Lenna*, block size 4×4

Group's number	Number of block elements in the group with absolute values of 1	2	3	4	5	6	7	8	Number of blocks in the groups
1	0	0	0	0	0	0	0	0	6881
2	1	0	0	0	0	0	0	0	2598
3	2	0	0	0	0	0	0	0	1025
4	3	0	0	0	0	0	0	0	543
58	3	0	1	0	0	0	0	1	7
112	6	2	1	0	0	0	0	0	3
465	8	0	0	1	1	0	0	0	1
494	10	0	1	1	0	0	0	0	1

Table 5.2: Distribution of codevectors and number of bits used for blocks in each category for the image *Lenna*

Category	Output distribution	Number of bits
1	45%	1
2	34%	9 - 19
3	21%	17 - 40

blocks is 15,360. The numbers in the first row of the table show the absolute values, and the numbers in the other rows show the number of non-zero symbols in each group.

Depending on the distribution of the output blocks in each group, these groups can be classified into different categories. For example, in the image *Lenna* , we divide these groups into three different categories: the origin or all-zero vectors, the next seven most probable groups, and the rest. As a result, 44 per cent of the points are represented by the first category, 34 per cent belong to the second category, and only 22 per cent are in the last category. Table 5.3.1 shows the distribution of the output points for each category and the number of bits used to index the output points in each category for the image *Lenna* .

The first category, i.e., the one containing all-zero vectors can be indexed with only one bit. In this way about one-half of the 16-dimensional blocks can be encoded only with one bit per block. The second category consists of seven most probable groups. These usually consist of code-vectors with one or two non-zero symbols. To index the code-vectors in the second category, we need 9 to 19 bits. The first four bits specify the category and the group and the remaining five to fifteen bits specify the positions of the non-zero symbols in the vector and its sign. Finally, in the last category, 12 bits are used to specify the category of the group and 5 to 30 bits are used for defining the positions of the non-zero components. Although in category 3, sometimes more than 30 bits are used to index a block, the effect on the overall bit rate is negligible, since only one or two blocks belong to these groups. Figure 5.1 shows the code-length in different categories using this method. These observations show that using this grouping method, a considerable bit reduction can be achieved. In addition, most of these non-zero elements are low-frequency components. Considering this fact results in a lower bit rate for high dimensions.

5.3.2 Method based on grouping according to the radial parameter

In this method, the lattice points are grouped according to the radial parameter, r (see Eqn. 5.1). If the DCT components of an image are quantized with a z^{16} lattice truncated with a sphere of radius 9, there are eighty one different groups. For the image *Lenna*, forty four per cent of the output vectors are mapped into the origin, sixteen per cent of the points on the sphere with a radius 1 and six per cent on the sphere with a radial parameter of 2. In most of the spheres (groups), only a few output points are mapped on the sphere. For example, in the simulation of the DCT components of the image *Lenna*, only 22 output points fall on the sphere with $r = 24$, and only 7 output points are mapped on the sphere with the radial parameter 51.

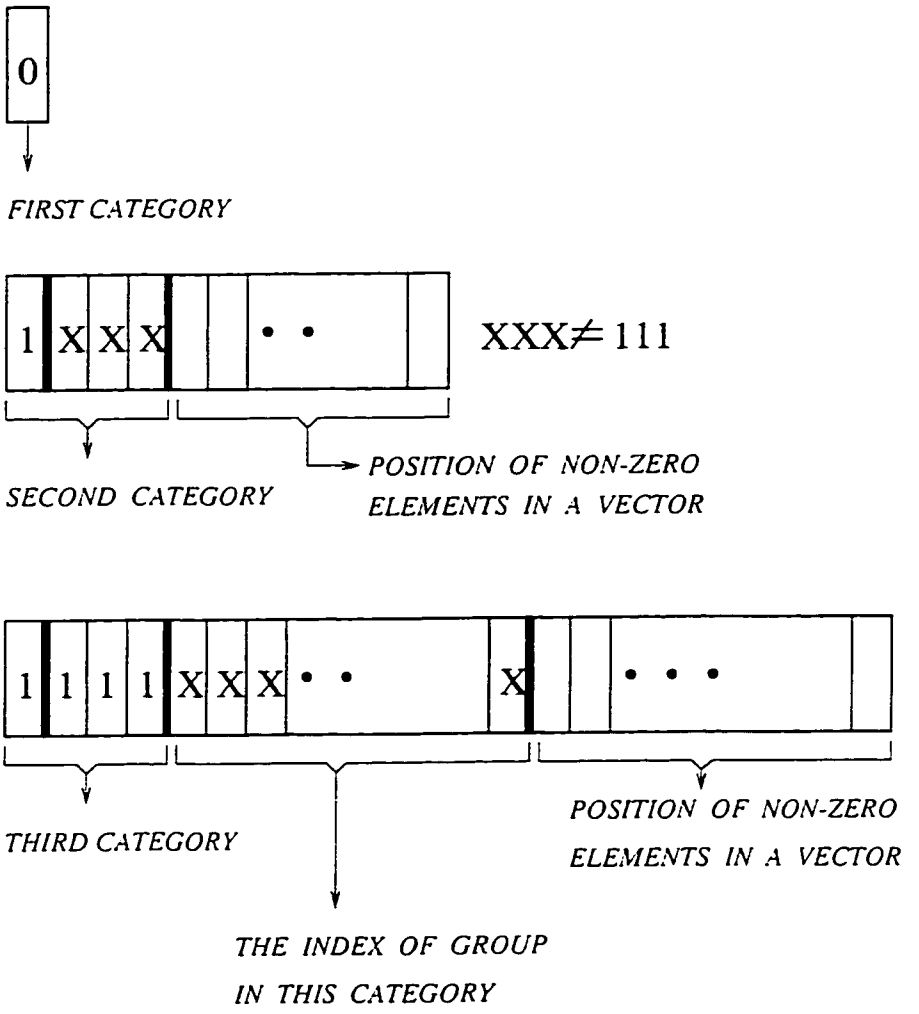

Figure 5.1: The code length in different categories for the method based on grouping according to non-zero values.

For indexing the output points on each sphere, the enumeration method explained in [15] can be used. The number $N_e(N, r)$ of integer points on the N-dimensional sphere with the radial parameter r can be calculated using the recurrence relation given by

$$N_e(N, r) = N_e(N-1, r) + 2\sum_{j=1}^{m} N_e(N-1, r-j^2), \tag{5.2}$$

where m is the largest integer such that $m^2 \leq r$. Using this equation, the index of each point can be calculated recursively. The codeword assigned to each point consists of two parts. The b most significant bits specify the sphere on which the point lies. The rest of the bits identify the location of the point on the sphere. Since the number of the points on the spheres with large radii is huge, and a full enumeration requires too many recursive calculations, the enumeration of these points is quite difficult. Using a partial enumeration, i.e., enumerating only the points on small spheres can reduce the search complexity considerably.

In this work, only the output points on the most probable spheres are indexed using the enumeration method. Since these spheres have small radius, the enumeration is not very difficult. For the indexing of a points on a sphere with large radius, the values of its vector components are used. Most of these values are less than 3, (quite often 0); thus we use a prefix variable length code to index these values. Although for indexing the points on a sphere with a large radial parameter, as many as 40 bits may be used, the effect on the overall bit rate is negligible, since only a few points are mapped onto such a sphere.

For different groups (i.e., spheres with different radii), Table 5.3 shows per cents of points falling on them, number of lattice points on each sphere and the number of bits used to index the output points mapped onto these spheres in a 16-dimensional space. The table also shows the number of bits obtained by using a full

Table 5.3: Distribution of codevectors and number of bits used for blocks in selected groups

radius	Per cent of output points on the sphere	Number of Lattice points on the sphere	Number of bits		
			Group with radius	Enumeration	Group with values
0	44	1	1	1	1
1	16	32	3+(5)	2+(5)	4+(5)
2	6	480	3+(9)	3+(9)	4+(10)
3	3	4480	3+(13)	5+(13)	4+(14)
4	3	29152	3+(8-20)	5+(15)	4+(5-20)
5	2	140736	3+(12-24)	7+(18)	4+(10-20)
21	0.001	3.9e+9	3+(24-48)	22+(33)	13+(24-40)

enumeration method and by employing the method based on grouping according to the number of non-zero elements (Section 5.3.1). In full enumeration, the prefix bits which indicate the sphere, are an estimation for Huffman coding suggested in [29]. The code presentation is shown in Figure 5.2.

5.4 SIMULATION AND RESULTS

Images which are quantized and encoded using the proposed methods are shown in Figure 5.4. Each image is partitioned into 8×8 blocks and the DCT is computed over each block. The DC coefficients are quantized separately using differential pulse code modulation. A scalar quantizer is designed to quantize the difference component of the DC coefficients. The quantized coefficients are then entropy coded. The ac coefficients are quantized with an LBVQ using z^N cubic lattice. The infinite lattice is truncated with spherical contours with different radii . In each case, the Conway and Sloane's fast quantization technique [34] [33] is used for finding the nearest lattice points.

In the first method, grouping according to the values of the non-zero elements,

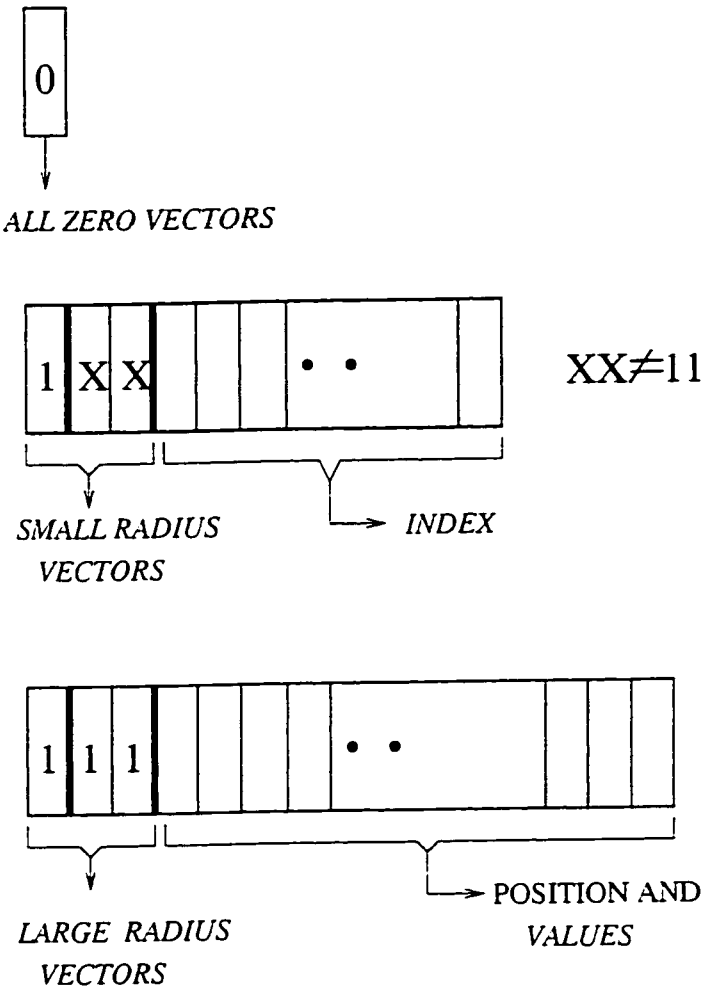

Figure 5.2: The code length for different groups in the proposed method based on grouping according to the radial parameters.

Figure 5.3: The test images *Light, Lenna, Baboon, Bridge, Girl and Tree.*

Table 5.4: PSNR and bit rate using the method based on grouping according to the radial parameter for the image *Lenna*

Radius	Scheme 1		Scheme 2	PSNR
	# of groups	bpb	bpb	dB
3	18	11.69	7.3	28.55
6	222	21.36	17.6	31.21
9	653	32.44	27.3	32.49

according to the density of the non-zero elements, the categories are defined. These categories should be specified in the header. In some cases, when the radius is large, some groups are common for different images. These groups can be predefined for the decoder in order to make the header shorter. In all our simulations, there are only three categories as mentioned in Section 5.3.1.

Table 5.4 shows the number of groups, the bit rate for one block, (bpb) and the PSNR for different radii of truncation for the ac coefficients of the image *Lenna*. In this case, the two methods, Scheme 1 and Scheme 2, have been used. In the first scheme, the location of the non-zero elements are represented by 6 bits. After

specifying the group of non-zero elements, their signs and locations are transmitted. In this method, there is no restriction on the location of the coefficients within the block. Since most images have a low-pass spectrum, the non-zero elements are usually concentrated in the left-upper corner of each block (the low-frequency coefficients). Hence, in the second scheme, the blocks are divided into four quadrants. In this scheme, each coefficient is specified by the quadrant number and the position of the coefficient within the quadrant. However, with this scheme, savings in bit rate is achieved by determining the quadrant number of the coefficients belonging to the first quadrant by default. Table 5.4 also shows the result of using this scheme for the image *Lenna* (Scheme 2).

Tables 5.5 and Table 5.6 compare the performance of the proposed method with JPEG. It is seen that the proposed methods, for the image *Light* , yields superior performance compared to JPEG. For the bit rate around 1.8, the PSNR with the new method is 26.4 dB, while JPEG results in a PSNR of 24.3 dB. For the image *Lenna*, however, JPEG performs better than the proposed method. Using quantization table, JPEG has different scale factors for different coefficients. In this way, the high frequency coefficients almost vanish. In the image *Lenna* where pixels are highly correlated, by doing entropy coding twice, JPEG achieves higher compression. However, in images with lower correlation, JPEG cannot deliver similar results. In such cases, our method yields better performance, since high-frequency coefficients are also taken into consideration.

In the second method, based on grouping according to the radial parameter values, the output points which are mapped into the origin are only quantized with one bit. For the next two spheres (with radii 1 and 2), enumeration method is used. The rest of the output points are indexed with the values of non-zero elements. The advantage of this method compared to the first one is that the header is very small

Table 5.5: The performance comparison of Method 1 and JPEG for the image *Lenna*

PSNR	JPEG	Method 1	
		Scheme1	Scheme 2
dB	bps	bps	bps
32	0.35	0.58	0.5
31	0.29	0.4	0.37
28.5	0.2	0.25	0.21

Table 5.6: The performance comparison of Method 1 and JPEG for the image *Light*

PSNR dB	Method 1 bps	JPEG bps
26.4	1.8	2.25
24.0	1.4	1.78
20.01	0.39	0.62

and the groups need not be defined in the header. Besides, the code is not dependent on the image. Since the number of lattice points chosen inside the boundary is very large, the performance of this method is better than other LBVQ's, where the number of the lattice points are limited by the bit rate. Furthermore, indexing of the lattice points in this method is not based on enumeration which requires too many recursive operations. In the piecewise uniform VQ [32], the lattice points are divided into several zones and each zone has its own scaling factor. Although the most probable sections are quantized by a fine quantizer, the number of codewords are limited by the rate and its indexing method is still employs enumeration. Table 5.7 compares the result of a uniform LBVQ, full enumeration which is suggested in [15] (PVQ) and piecewise uniform VQ with the proposed method (these results are taken from [32]). For some images our method outperforms these methods. For example, for the image *Lenna*, the proposed method yield an improvement of about 5dB over the other methods. For some other images the result is comparable to other methods. Figure 5.4 compares the result of this method with the PVQ for

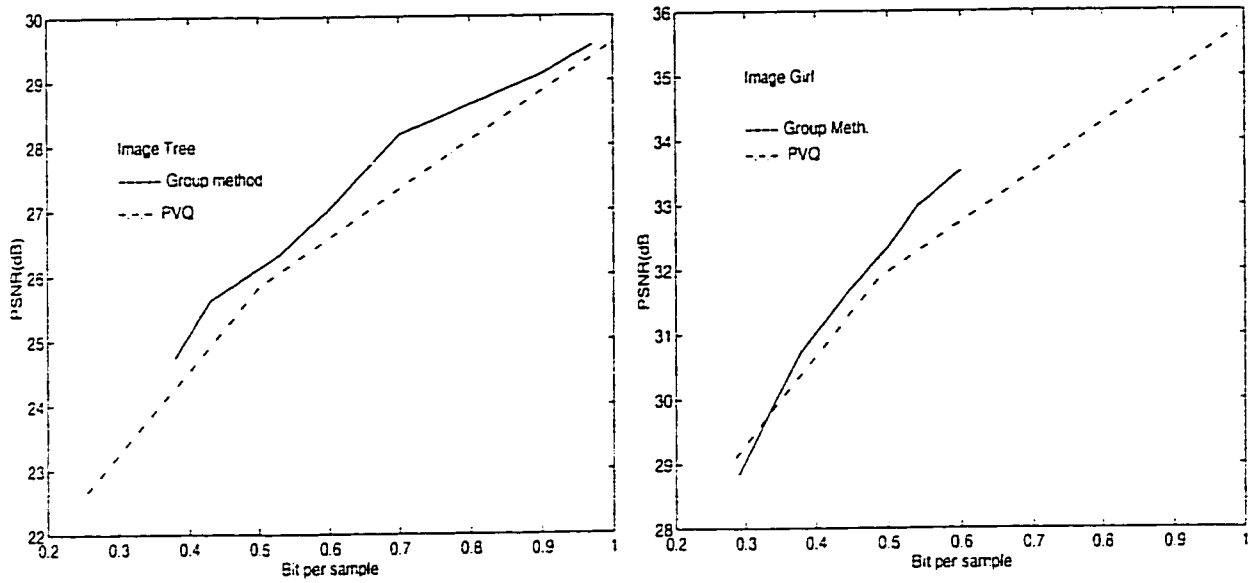

Figure 5.4: The comparison of the method based on grouping according to the radial parameter and PVQ.

image Girl and Tree.

This method is also compared with JPEG in Table 5.8. For most images, this method outperforms the JPEG. For example, in the image *Light*, this method uses 1.3 bits per sample for a PSNR of 24 dB while the JPEG needs 1.8 bits per sample to get the same result. Among the images tested, only for the image *Bridge*, the JPEG shows better performance than the proposed method. Figure 5.5 compares the performance of this method with that of the JPEG. For the image *Lenna*, the PSNR is only slightly better than that of the JPEG. However, for the image *Light* an improvement of up to 3 dB is achieved. Figure 5.6 compares the quantized image *Lenna* obtained using this method for the bit rate 0.27 with the output of JPEG for the same bit rate. It is seen that the blocking effect is reduced with the proposed method.

To show the efficiency of the method in regard to its complexity, the proposed

Table 5.7: The performance comparison of the method based on grouping according to the radial parameter and other indexing method for some images

Image	Uniform z^{16} VQ PSNR(bps)	PVQ PSNR(bps)	Piecewise Uniform Z^{16} PSNR(bps)	Grouping method PSNR(bps)
Lenna	28.13(0.5)	27.62(0.5)	28.23(0.5)	33.89(0.44)
Girl	32.78(0.5)	31.9(0.5)	32.95(0.5)	32.35(0.5)
Tree	26.31(0.5)	26.05(0.5)	26.43(0.5)	26.33(0.53)

Table 5.8: The performance comparison of the method based on grouping according to the radial parameter and that of JPEG for some images

Image	Grouping with radial parameter PSNR(bps)	JPEG PSNR(bps)
Lenna	32.8(0.33)	32.90(0.374)
	30.98(0.25)	31.37(0.29)
Light	26.65(1.6)	26.70(2.3)
	23.97(1.4)	24.36(1.838)
Bridge	28.88(1.3)	28.70(1.095)
	27.03(1.0)	26.90(0.692)
Baboon	30.32(1.5)	30.95(1.75)
	28.0(1.17)	28.84(1.28)

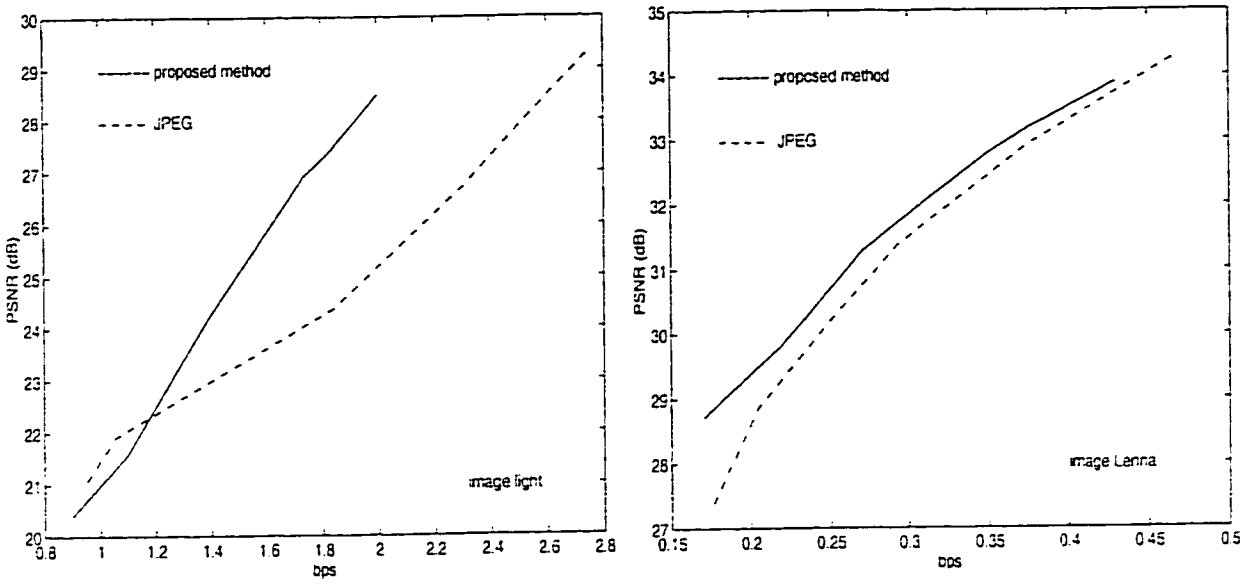

Figure 5.5: The comparison of the method based on the grouping according to the radial parameter and JPEG for the images *Light* and *Lenna*.

method is compared with an optimal VQ for some images. Table 5.9 shows the complexity of the two methods. The results of VQ are for low to medium dimensions. Although the VQ shows better performance with higher dimensions, but the complexity of calculation is higher. In some cases the codevector is far from the input vector such that it is not necessary to calculate the distortion between all the elements of the input vector and the vector in the codebook. Thus, in the calculation of complexity for the VQ, the number of multiplications and additions are considered to be one-half of a full search. In the calculation of the complexity, each comparison is counted as equilvalent to an addition.

We also tested the error samples of an LBVQ to investigate the advantages of using the universal Gaussian codebook to quantize the error sample. It has been observed that, the error samples are far from having a Gaussian distribution function, a result that was also obtained when quantizing the low frequency coefficients of image. Figure 5.7 shows the distribution of error samples when the first stage is

a

b

Figure 5.6: The comparison of the method based on grouping according to the radial parameter and JPEG for the image *Lenna* for bit-rate 0.27 bps. (a) Proposed method. (b) JPEG.

Table 5.9: The comparison of complexity of VQ and proposed method for some images

Image	Optimal VQ				Proposed method			
	Dim bps	Rate dB	PSNR	Complexity Multiplication/ Addition	Dim	Rate bps	PSNR dB	Complexity Multiplication/ Addition
Lenna	4 × 4	0.5	31.37	128 / 256	4 × 4	0.29	31.37	2 / 5
Baboon	2 × 2	1.5	28.95	32 / 64	4 × 4	1.17	28.0	2 / 5
Bridge	2 × 2	1.5	28.5	32 / 64	4 × 4	1.3	28.88	2 / 5
Light	2 × 2	1.5	24.11	32 / 64	4 × 4	1.4	23.97	2 / 5
	2 × 2	2.0	27.04	128 / 256	4 × 4	2.3	26.7	2 / 5

LBVQ for the images *Lenna* and *Bridge*. Next, the JPEG data compression scheme is used in the first-stage. The distribution of error samples for low to high compression for image Lenna and Bridge. is shown in Figure 5.8 Our investigation shows that, if a transform coding is used in the first stage of a two-stage vector quantizer, the distribution of error samples are far from having a Gaussian distribution. and using a universal Gaussian code book is not efficient.

5.5 SUMMARY

In this chapter, two methods using a combination of LBVQ and noiseless coding for the encoding the DCT coefficients of an image have been presented. The first method is based on the grouping of the quantized coefficients according to the number of their non-zero elements. The second one classifies the output points according to their radii. Simulation results for different images have been presented and compared

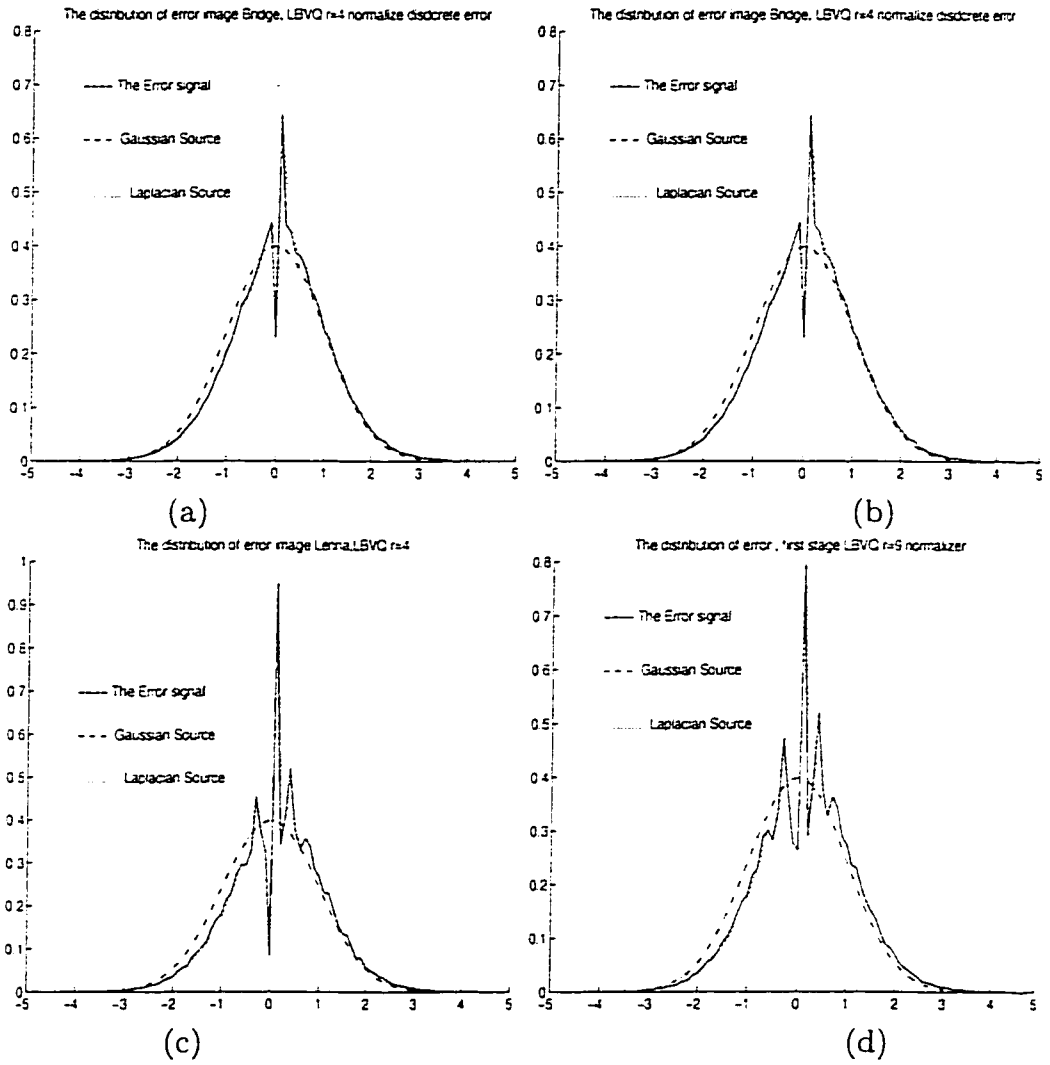

Figure 5.7: The distribution of error image, first stage LBVQ. (a) r=4, image *Bridge*. (b) r=9, image *Bridge*. (c) r=4, image *Lenna*. (d) r=9, image *Lenna*.

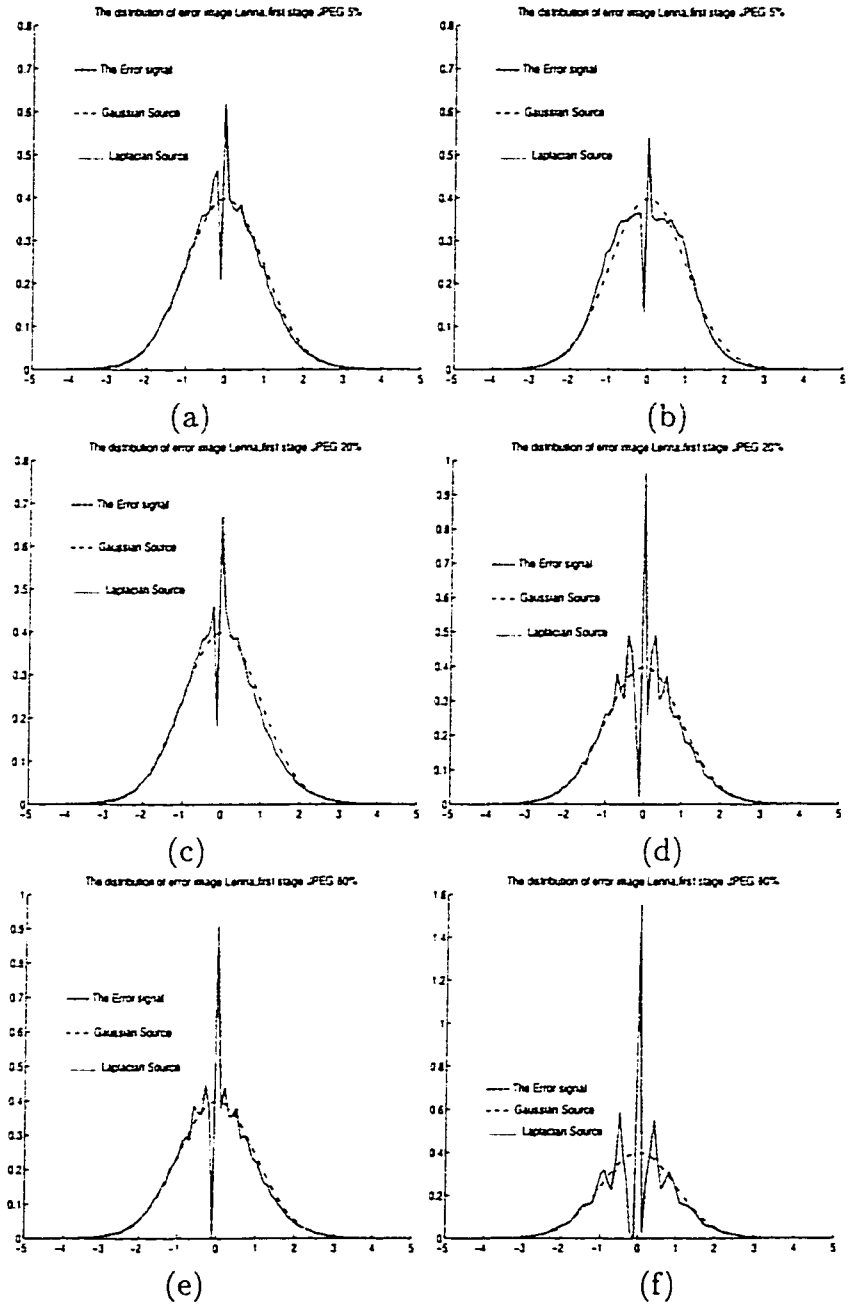

Figure 5.8: The distribution of error image, first stage JPEG. (a)JPEG 5% image *Bridge*. (b) JPEG 5% image *Lenna*. (c) JPEG 20% image *Bridge*. (d) JPEG 20% image *Lenna*. (e) JPEG 80% image *Bridge*. (f) JPEG 80% image *Lenna*.

with the JPEG and other LBVQs. The first method gives better results for some images, but the second method outperforms the JPEG for most of images used to test the method.

Chapter 6

CONCLUSION

6.1 CONCLUDING REMARKS

Vector quantization is an efficient method of data compression, especially for speech and images. This thesis has been concerned with the problems of search and codebook memory requirements of vector quantizers for image compression.

In order to reduce the search complexity of vector quantizers, a multi-stage vector quantizer with a unique codebook has been introduced. A low-rate optimum vector quantizer has been used in the first stage and a universal Gaussian codebook, designed for a memoryless Gaussian source, for the other stages. It has been shown that the locally normalized error samples of images have a distribution close to a normal distribution. Since a Gaussian memoryless signal is successively refinable, the error samples are also successively refinable. As a consequence, the codebook designed for a memoryless Gaussian source can be used in different stages of a multi-stage vector quantizer to encode the image error samples. An optimum codebook designed for a normally distributed source has been used to quantize error sample of different images, and the results have been compared with the reconstructed images quantized by an optimum vector quantizer. The results from the proposed technique is very close to that from the optimum vector quantizer. Since with the proposed method only one codebook is needed in different stages of the residual VQ, different structures and mapping techniques can be used to reduce the search complexity.

Since the compression search complexity can also be reduced by quantizing only the more important parts of of an image, the low-frequency coefficients have been quantized in the first stage of a residual multi-stage quantizer. In this way the smaller size of the source results in a reduced search complexity of search. The second stage is then used to restore the information neglected in the first stage. The function of the second stage is to work on the residual image obtained by subtracting the output of the first stage from the original image. This task has been implemented

in two ways. In the first scheme, a lattice-based vector quantizer has been used as the quantizer, while in the other one, a standard JPEG with a low rate has been used as the quantizer of the first stage, and a lattice-based vector quantizer for the second stage. The resulting bit rate of the two-stage lattice-based vector quantizer in either scheme has been found to be considerably lower than that of the JPEG in the same quality of the encoded images in moderate to high rates applications. With the proposed two-stage lattice-based vector quantizer, an improvement of up to 2 bits has been achieved.

Although the proposed two-stage vector quantizer provides considerably better performance than the JPEG for high bit-rate compression, it is not effective for lower rates. This is due to the fact that a major fraction of the bit rate comes from the second stage and the bit rate associated with this stage remains almost constant. Thus, the third part of this thesis has been concerned with the low bit-rate compression. In this part, the DCT coefficients have been quantized with a lattice-based vector quantizer in which the lattice points are truncated with a large radius. As a result, a large number of points fall inside the boundary of the hyper sphere or the codebook, and thus, images are encoded with high quality and low complexity. In order to reduce the bit rate, a shorter representation is assigned to the more frequently used lattice points. To index the large number of lattice points falling inside the boundary, two methods have been proposed. Both these methods are based on the grouping of the lattice points according to their frequencies of occurrence. In the first method, these points are grouped according to the non-zero elements of the quantized DCT coefficients. In the second scheme, the grouping is carried out according to the radial parameter of the lattice points. After grouping, a lattice point is indexed according to its group and position of its non-zero elements. For most of the images tested, the proposed methods have been found to outperform the JPEG in terms peak signal to noise ratio and visual quality of reconstructed image

at the same computational complexity. However, for the other lattice-based vector quantizer schemes, the proposed method yields better performance with lower computational complexity.

6.2 SCOPE FOR FURTHER INVESTIGATION

Availability of a universal codebook for coding any source with no loss in quality would be very attractive. In one of the proposed methods in this thesis, an optimum codebook has been used in the first stage, and a universal Gaussian codebook in the other stages. It would be of interest to investigate the use of an universal Gaussian code book in all stages of a quantizer for applications in which has a Gaussian distribution such as row SAR data.

The idea of having a universal codebook could be extended to the frequency domain. It is well known that the ac coefficients of an image has a Laplacian distribution. On the other hand, random variables drawn from a Laplacian distribution are successively refinable when the distortion is measured using the absolute distortion criterion. It may be desirable to design a Laplacian universal codebook, under the absolute distortion criterion, to encode the ac coefficients of an image. It is obvious that in this case, the different stages should be able to work on the difference of quantized and the DCT original coefficients, rather than on the error samples.

Developing some structures and mapping techniques for the universal codebook would also be of interest to investigate. One approach could be the one in which the codevectors are grouped according to their norm squares.

In Chapter 5, the indexing of the lattice points in a lattice-based vector quantizer has been carried out by choosing cubical lattice and a spherical boundary. Further investigation is needed with the use of different boundaries and lattices in order to improve the performance of the proposed lattice-based vector quantizer.

REFERENCES

[1] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[2] T. M. Cover and J. A. Thomas, *Information Theory.* New York: John Wiley & Sons, 1991.

[3] J. Rissanen and G. Langdon, "Compression of black and white images with arithmetic coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 858–868, June 1981.

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

[5] R. M. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4–29, April 1984.

[6] C. E. Shannon, "Coding theorem for a discrete source with fidelity criterion," *IRE National Convention Record*, pp. 142–163, 1959.

[7] P. Zador, "Asymptotic quantization continuos random vector," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 139–141, March 1982.

[8] A. Buzo, A. Gray, R. Gray, and J. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech and Signal*, vol. 28, pp. 562–575, Oct. 1980.

[9] C. F. Barnes, *Residual Quantizers.* PhD thesis, Brigham Young University, Provo, Utah, Dec. 1989.

[10] M. J. Sabin and R. M. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Trans. Acoust., Speech and Signal*, vol. ASSP-32, pp. 478–488, June 1984.

[11] K. Sayood, J. D. Gibson, and M. C. Rost, "An algorithm for uniform vector quantizer design," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 805–814, Nov. 1984.

[12] N. M. Nasrabadi and R. A. King, "Computationally efficient adaptive block-transform coding," in *Proc. EUSIPCO-83*, pp. 729–733, Sept 1983.

[13] T. Saito, H. Takeo, K. Aizawa, and H.Harashima, "Adaptive discrete cosine transform image coding using gain/shape vector quantization," in *IEEE Trans. Acoust., Speech and Signal*, pp. 129–132, IEEE, Apr 1986.

[14] G. K. Wallace, "The jpeg still-picture compression standard," *Communications of the ACM*, vol. 34, pp. 30–44, April 1991.

[15] T. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 468–483, July 1986.

[16] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 373–380, July 1979.

[17] A. Gersho, "On the structure of vector quantizers," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 157–166, March 1982.

[18] K. Sayood and J. Gibson, "Lattice quantization," *Adv. Electron, Electron Phys.*, vol. 72, 1988.

[19] D. Jeong and J. Gibson, "Uniform and piecewise uniform lattice vector quantization for memoryless Gaussian and Laplacian sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 786–804, May 1993.

[20] J. Conway and N. Sloane, *Sphere packing, Lattice and Groups.* Newyork: Newyork Springer-Verlag, 1988.

[21] M. V. Eyuboglu and G. D. Forney, "Lattice and trellice quantization with lattice and trellice-bound codebooks-high rate theory for memoryless sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 46–59, Jan 1993.

[22] T. Fischer, "Geometric source coding and vector quantization," *IEEE Trans. Inform. Theory*, vol. 35, pp. 137–145, January 1989.

[23] R. Laroia and N. Farvardin, "A structured fixed-rate vector quantizer derived from a variable-length scalar quantizer part I:memoryless sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 851–867, May 1993.

[24] A. Habibi, "Comparison of nth order DPCM encoder with linear transformations and block quantization techniques," *IEEE Trans. Commun.*, vol. Com-19, pp. 948–957, Dec. 1971.

[25] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, October 1991.

[26] A. Segall, "Bit allocation and encoding for vector sources," *IEEE Trans. on Audio and Electroacoustics*, June 1973.

[27] Y. Huang and P. M. Schultheiss, "Block quantization of correlated Gaussian random variables," *IEEE Trans. Commun.*, pp. 289–296, Sept. 1963.

[28] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[29] T. R. Fischer, "Entropy-constrained geometric vector quantization for trasform image coding," in *Proc. Int. Conf. Acoustics, Speech, and Singnal Processing*, pp. 2269 –2272, IEEE, April 1991.

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

[31] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–137, March 1982.

[32] D. Jeong and J. Gibson, "Image coding with uniform and piecewise uniform vector quantizer," in *Proc. IEEE Global Telecomm. Conf.*, Dec. 1991.

[33] J. Conway and N. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 820–824, November 1983.

[34] J. Conway and N. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 227–231, March 1982.

[35] A. Papoulis, *Probability, Random Variables, and Stochastic processes.* New York: McGraw-Hill, 1991.

[36] T. Berger, *Rate Distortion Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1969.

[37] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 269–275, March 1991.

[38] A. Gamal and T. M. Cover, "Achievable rate for mulytiple description," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 851–857, Nov. 1982.

[39] J. Pan and T. Fischer, "Two-stage quantization-lattice vector quantization," *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 155–163, Jan. 1995.

[40] D. Lee, D. Neuhoff, and K. Paliwal, "Cell-conditional multistage vector quantization," in *Proc. Int. Conf. Acoustics, Speech, and Singnal Processing*, May 1991.

[41] D. Llebedeff, P. Mathieu, M. Barlaud, C. Lambert-Nebout, and P.Bellemain, "Adaptive vector quantization for raw SAR data," in *Proc. Int. Conf. Acoustics, Speech, and Singnal Processing*, pp. 2511–2514, 1995.

[42] S. D.Silvey, *Statistical inference*, ch. 9. London, England: Chapman Hall, 1975.

[43] H. Witsenhausen, "On source network with minimal breakdown degradation," *Bell Syst. Tech. J.*, vol. 59, July-Aug. 1980.

[44] J. K. Wolf, A. D. Wyner, and J. Ziv, "Source coding for multiple description," *Bell Syst. Tech. J.*, vol. 59, Oct. 1980.

[45] L. Ozarow, "On a source coding problem with two channels and three receivers," *Bell Syst. Tech. J.*, vol. 59, Dec. 1980.

[46] R. Ahlswede, "The rate distortion region for multiple descriptions without excess rate," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 721–726, Nov. 1985.

[47] D. J. Sakrison, "The rate distortion function for a class of sources," *Inform. Contr.*, vol. 15, pp. 165–195, 1969.

[48] A. Lapidoth, "On the role of mismatch in rate distortion theory," *IEEE Trans. Inform. Theory*, vol. IT-43, pp. 38–47, Jan. 1997.

[49] B. Juang and A. Gray, "An algorithm for vector quantizer design," in *Proc. Int. Conf. Acoustics, Speech, and Singnal Processing*, vol. 1, pp. 597–600, IEEE, 1982.

[50] R. Laroia and N. Farvardin, "Trellis-based scalar-vector quantizer for memoryless s ources," *IEEE Trans. Inform. Theory*, vol. 40, pp. 860–870, May 1994.

[51] R. Reininger and J. Gibson, "Distribution of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. COM-31, pp. 835–839, June 1983.