IMAGE COMPRESSION USING ONE-DIMENSIONAL VECTOR QUANTIZATION

by

AIMAN HELMI EL-MALEH
B.Sc., King Fahd University of Petroleum & Minerals, 1989

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in the Department of
Electrical and Computer Engineering

We accept this thesis as conforming
to the required standard

[redacted]

Dr. F. El-Guibaly, Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. V. K. Bhargava, Co-Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. G. F. McLean, Outside Member, Dept. of Mechanical Engineering

[redacted]

Dr. R. Vahldieck, Graduate Advisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. S. Dost, External Examiner, Dept. of Mechanical Engineering

Supervisors: Dr. F. El-Guibaly and Dr. V. K. Bhargava

ABSTRACT

Vector Quantization (VQ) is a well-known image coding technique. Previously, vector quantizers operated on two-dimensional (2-D) blocks spanning several adjacent lines in an image. In this work, we investigate the capability of one-dimensional (1-D) VQ as an image coding technique. 1-D VQ operates on blocks spanning a portion of a single scanning line. The advantages of 1-D VQ are considerable reduction in implementation complexity, reduced storage requirement and a faster encoder response. Problems encountered in 1-D VQ are discussed and compared to those encountered in 2-D VQ. 2-D VQ showed a better performance than 1-D VQ. The PSNR was on the average 1.6 dB higher in images coded using 2-D than in those coded using 1-D VQ.

In order to improve the perceptual quality of coded images using 1-D VQ, a special block accessing scheme is used. Furthermore, product structures are used to improve the performance of 1-D VQ and reduce the computational complexity. A novel product structure based on scaling (SVQ) or scaling and rotating (SRVQ) the vectors before quantization is introduced. Furthermore, the mean/shape vector quantizer (M/SVQ) is used and compared to our new structure. By using product structures in 1-D VQ, the signal-to-noise ratio (SNR) of coded images is increased by almost 1 dB using SVQ and by almost 2 dB using M/SVQ for the same coding complexity. From this work we show that, by using the block accessing scheme and product structures, 1-D VQ produces good quality coded images at compression ratios in the range of 8:1 (1 bpp).

Examiners:

[redacted]

Dr. F. El-Guibaly, Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. V. K. Bhargava, Co-Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. G. F. McLean, Outside Member, Dept. of Mechanical Engineering

[redacted]

Dr. R. Vahldieck, Graduate Advisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. S. Dost, External Examiner, Dept. of Mechanical Engineering

Table of Contents

List of Tables

List of Figures

Acknowledgment

I wish to thank my supervisor, Dr. F. El-Guibaly for his encouragement, guidance, and support during the course of this research. Dr. El-Guibaly was an excellent thesis advisor in giving me an unlimited amount of time to discuss and suggest ideas for research. I thank, too, my co-supervisor Dr. V. K. Bhargava for his support, valuable comments and for making everything available for the completion of this work.

I also express my thanks to Mr. W. A. Keddy for providing the image analysis software package (DEDIP) and Dr. Scott Soper of BellCore for providing the standard images.

I also thank the faculty members and staff of the Dept. of Electrical and Computer Engineering for their patronage and co-operation.

Financial assistance received from the University of Victoria (through UVic fellowship), and from Dr. F. El-Guibaly and Dr. V. K. Bhargava (through NSERC), is gratefully acknowledged.

Finally, I would like to thank my parents, relatives and well-wishers for their moral support, motivation, and encouragement.

List of Symbols

1-D	One-dimensional
2-D	Two-dimensional
A-VQ	Address-vector quantization
bpp	Bit per pixel
BSVQ	Black scaled vector quantizer
CVQ	Classified vector quantization
dB	Decibel
DCT	Discrete cosine transform
DM	Distortion measure
DPCM	Differential pulse coded modulation
FSVQ	Finite state vector quantization
HVQ	Hierarchical vector quantization
i.i.d	Independent and identically distributed
KLT	Karhunen-Loeve transform
LBG	Linde, Buzo and Gray algorithm
LFSR	Linear feedback shift register
M/RRVQ	Mean/reflected residual vector quantizer
M/RVQ	Mean/residual vector quantizer
MSE	Mean-squared error
M/SVQ	Mean/shape vector quantizer
MTF	Modulation transfer function
NN	Nearest neighbor
PCM	Pulse coded modulation
pdf	Probability density function
PNN	Pairwise nearest neighbor
PRG	Pseudo random generator
PVQ	Predictive vector quantizer
R-D	Rate-distortion
SNR	Signal-to-noise ratio
SRVQ	Scaled rotated vector quantizer
SVQ	Scaled vector quantizer
TC	Transform coding
t-set	Training set
VQ	Vector quantizer or vector quantization
WMSE	Weighted mean squared error
WSVQ	White scaled vector quantizer

Chapter 1

Introduction

1.1 Background

Image coding aims at reducing the number of bits required to represent and reconstruct a faithful duplicate of the original image. Due to the high spatial correlation in pixel intensities, most image coding techniques try to utilize this correlation as much as possible to reduce the coding rate. Distortion is introduced to the images as a price for the compression. The best image compression algorithms are the ones which reduce the distortion and distribute its effect so that it is least objectionable perceptually.

A fundamental result of Shannon's rate distortion theory [43] is that better performance can always be achieved by coding vectors instead of scalars. Thus, vector quantizers produce better performance than scalar quantizers. Due to this fact, extensive studies of vector quantizers have recently been performed by many researchers. Vector quantizers compress data by matching the input vector to the closest vector in a set of reproduction vectors stored in a look-up table (codebook), where "closeness" is quantified by some convex distortion measure. The index of the selected vector is then sent to the

decoder which has the same copy of the codebook. Decoding is simply done by performing a table look-up operation.

Vector quantization (VQ) has been found to be effective and powerful for image coding due to its inherent ability to exploit the high correlation between the neighboring pixels. However, VQ requires high computational complexity during the encoding process. For every vector to be encoded, a full search of the codebook is performed to find the best matching vector. For a constant rate, VQ codebooks grow exponentially with block size. In order to improve the performance of vector quantizers, large codebooks are needed. Thus, to improve the performance and reduce VQ's complexity many suboptimal techniques have been introduced by enforcing a structural constraint in the codebook. Some of these techniques are tree-structured codebooks [5], product-structured codebooks [5, 18] and classified vector quantization (CVQ) [38, 39].

The performance of vector quantizers using product structures was compared to a conventional adaptive discrete cosine transform coder of roughly comparable complexity [5]. VQ showed a similar and in some ways a better performance. CVQ was also compared to a discrete cosine transform coder employing the same block size and the same rate [38]. CVQ outperformed transform coding by a significant margin.

1.2 Purpose and Approach

The objective of this thesis is to investigate the capability of one-dimensional (1-D) vector quantization (VQ) as an image coding technique. In two-dimensional (2-D) vector quantization, an image is partitioned into two-

dimensional blocks each spanning several adjacent lines and each block is then encoded as a separate entity. 2-D VQ requires lines of buffering in a raster scanned system employing this design. The size of each buffer is equal to the width of the image to be encoded and the number of buffers is dependent on the block size. In 1-D VQ, no lines of buffering are required. This results in considerable reduction in the implementation complexity due to the independent encoding of scan lines. Furthermore, the encoder response is much faster in 1-D than in 2-D VQ. Due to these advantages in 1-D VQ, experiments are made to test the performance of 1-D VQ in coding images.

Since we seek a low coding rate and we are limited by the computational complexity of VQ, we restrict our experiments to a fixed block size of dimension 16. Furthermore, in order to have a good visual quality for coded images we use a codebook size of 1024 codewords. A comparison between 1-D VQ and 2-D VQ is made for a set of six images and the problems encountered in both techniques are compared.

In order to improve the perceptual quality of coded images, a special block accessing scheme is used. This block accessing scheme distributes the distortion in a manner that is least objectionable perceptually. Product structures are used to improve the performance of 1-D VQ without increasing the computational complexity. A novel product structure based on scaling (SVQ) or scaling and rotating (SRVQ) the vectors before quantization is introduced. Furthermore, the mean/shape vector quantizer (M/SVQ) is used and compared to our new structure. From these experiments we show that, by using the block accessing scheme and product-structured codebooks, 1-D VQ produces good quality coded images at a low coding rate (1 bpp).

1.3 Organization

Chapter 2 serves as a brief introduction to digital image coding techniques. First, a brief description of image statistics is given and some of the distortion measures used for image coding are listed. Then, the major image coding techniques are listed and a brief description for each is given.

In Chapter 3, we first define vector quantization and then discuss the optimality criteria for vector quantizers. Next, we present the LBG algorithm and discuss some of the important factors that could affect codebook design. This is followed by presenting some of the suboptimal techniques that aim at reducing the high computational complexity of VQ. Finally, we discuss briefly vector quantizers that exploit the correlation between neighboring blocks.

Chapter 4 is the core of the thesis work. First, a comparison between 1-D and 2-D VQ is made and the problems encountered in 1-D VQ are identified and compared to those in 2-D VQ. Then, a special block accessing scheme is introduced to distribute the distortion and improve the perceptual quality of coded images. Product structures are used to increase the codebook size and improve the performance without increasing the computational complexity. A novel product structure based on scaling (SVQ) or scaling and rotating (SRVQ) the vectors before quantization is introduced. Furthermore, the mean/shape vector quantizer (M/SVQ) is used and compared to our new structure. Both scalar and vector quantizers are used to quantize the scale factors and the mean values. Finally, the WMSE distortion measure is used in an attempt to improve the fidelity of coded edges.

Chapter 5 briefly summarizes the major results of this thesis and lists some suggestions for future investigation.

Chapter 2

Digital Image Coding

2.1 Introduction

Digital representation of an image requires a very large number of bits. A digital image can be thought of as an $N \times N$ array of integer numbers or picture elements (pixels) requiring N^2m bits for its representation where m is the number of bits per pixel.

The goal of image coding is to reduce as much as possible the number of bits necessary to represent and reconstruct a faithful duplicate of the original image. This goal is motivated by the fact that adjacent samples in an image have close grey-scale values, thus exhibiting spatial correlation. By exploiting this correlation appropriately, the number of bits required to represent an image can be reduced significantly.

Applications of data compression are primarily in transmission and storage of information. In transmission applications, compression techniques are constrained by real time and online considerations that limit the size and complexity of the hardware. In storage applications, the requirements are less stringent because most of the processing can be done off line.

Image transmission techniques are useful in broadcast television, video-conferencing services, facsimile transmission of newspapers and printed material, weather photographs and earth-resource pictures sent to earth by satellite, and a number of military applications [35]. Image storage is required most commonly for educational and business documents, medical images, etc. By efficient coding of images, transmission costs and storage requirements for images can be reduced significantly.

We begin this chapter by talking briefly about image statistics. Then, we list some of the distortion measures used for image coding. Furthermore, the major image coding techniques are listed and a brief description for each is given.

We are restricted in our discussion to monochrome images. Images are assumed to be a set of digital samples, typically 512×512, and each sample is quantized uniformly to a discrete set of amplitudes, usually 256 levels (8-bits).

2.2 Image Statistics

Optimum coder design requires knowledge of the source statistics. Long-term first- and second-order statistics of natural images have been measured [35]. Results showed that the probability density of intensity samples was highly nonuniform, depending upon the camera settings and scene illumination, and varies widely from picture to picture. Autocorrelation functions have been approximated by $exp(-k_1 \mid \Delta x \mid -k_2 \mid \Delta y \mid)$ or by $exp[-(k_1(\Delta x^2) + k_2(\Delta y^2))^{1/2}]$ where Δx and Δy are spatial displacements in x- and y-directions respectively, and k_1 and k_2 are positive constants [35].

These autocorrelation functions were found to vary widely from image to image and also vary widely when computed for different blocks (of size, say, 32×32) from one image. The local statistics of an image change abruptly across edges. Thus, it has not been possible to find a simple stationary or nonstationary model for images. A complete model of images is not necessary to analyze the performance of most coders. Usually coders exploit only part of the correlation in an image, and it is sufficient if this part is modeled accurately. Most of the coding techniques that have been developed assume that the source is a stationary random process.

Further, the source statistics must be known if we are to compute the Rate-Distortion (R-D) function. The R-D functions are known only for a few source descriptions, the most well-known being the Gaussian source. A Gaussian model has often been used more for its tractability than for its validity. The main difficulty in modeling images is posed by the presence of edges. Some of this difficulty can be overcome by considering the picture signal as the output of many sources each tuned to a certain type of statistics [38]. If the edge information is subtracted from the picture signal, the rest of the signals appear to be close to a Gaussian process and, therefore, an optimal encoder can be applied [35].

2.3 Distortion Measures for Image Coding

In any coding system, distortion is introduced to coded images as a price for compression. A distortion measure (DM), which quantifies the distortion, is used to evaluate the performance of the coder, as well as to optimize the coder to achieve the minimum distortion at a given bit rate. So, any DM

needs to be perceptually meaningful to asses the quality of coded images and analytically tractable so that it lends itself to optimization.

The mean-squared error (MSE) has been widely used as a DM for its simplicity. The MSE is defined as the average of the squared error between the original pixel and its reconstructed counterpart taken over all the pixels in an image. Limb [26] has shown that the MSE is a reasonable measure for image quality provided that the noise is similar to white noise. However, in images where there is significant correlated noise near the edges, the MSE is not satisfactory.

A DM such as the MSE does not account for perceptual features and does not consider the environment of a pixel in computing the distortion for that pixel. In an attempt to make the DM more meaningful and more correlated with the perceptual features, various Weighted Mean Squared Error (WMSE) DMs have been proposed. These take the form of a linear transformation of the original and coded images, followed by the MSE computation [38]. An example is the Laplacian operator [36]. This linear operator has the effect of placing greater importance on distortions at the edges. While this DM gives a reasonable measure of quality with images that have corrupted edges, it fails to evaluate images with granular noise (similar to white noise) properly. By and large, WMSE distortion measures give about the same correlation with subjective evaluation as the MSE [36].

WMSE can be defined as

$$d(x,y) = \sum_i w_i \mid x_i - y_i \mid^r \tag{2.1}$$

where x_i and y_i are the pixel intensity and its reconstructed counterpart respectively. The weight w_i typically varies for each pixel x_i and is some

(possibly non-linear) function of the relative intensity values of x_i and its immediate neighbors, and r is an even integer.

Yet another variation of distortion measures is the Mean r^{th}-power DM

$$d(x,y) = [\sum_i (x_i - y_i)^r]^{\frac{1}{r}} \tag{2.2}$$

Limb [26] found that the correlation with perception was best for $r = 2$ (MSE) and $r = 4$.

By studying the human visual system, researchers found that our ability to perceive errors depends upon factors such as the spatial frequency content of an image, edge orientation, and relative brightness [5]. For example, we are more sensitive to errors in gray intensities than in white intensities and more sensitive to errors in smooth regions than next to edges. Recognizing these characteristics of the human visual system, some efforts were made attempting to define a perceptually-based DM. DMs based on the single-channel vision model were proposed by Stockham [44], by Mannos and Sakrison [30], by Hall and Hall [21], and by Limb [26]. The general form of their DMs is given by

$$d(x,y) = \sum_i \sum_j \{h(i,j) * [\log x(i,j) - \log y(i,j)]\}^2 \tag{2.3}$$

where $x(i,j)$ and $y(i,j)$ are the pixel intensity and its reconstructed counterpart respectively, and $*$ represents convolution. The various DMs differ only in the specification for the filter $h(i,j)$. The logarithmic nonlinearity in the vision model was incorporated by pre-transforming the original and coded images by a pixel-wise logarithmic operation. This was followed by the linear filter, and then the MSE of the output of the filter was computed. In each

case, the linear filter modeled the modulation transfer function (MTF) of the visual system. Perceptually-based DMs produce good results, however, the complexity of their operations preclude their inclusion in most practical image communication systems.

None of the DMs proposed in the literature correlates well enough with subjective evaluation to be accepted universally as an objective measure that can substitute for subjective evaluation. Most researchers use the signal-to-noise ratio (SNR) as a quantative measurement of reproduction fidelity. The particular definition mostly used is the ratio of peak-to-peak signal-to-mean square error expressed in decibels (dB)

$$SNR = 10\log_{10}[255^2/\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2] \tag{2.4}$$

where N is the number of pixels in the image, and 255 is the maximum intensity value.

By minimizing squared error, the SNR is maximized. So, SNR is useful for comparing various coding schemes which all strive to minimize squared error.

2.4 Image Coding Techniques

Image coding techniques can be classified as either first-generation or second-generation techniques [25]. First-generation techniques are those which have been working within the framework of information and coding theory. These techniques attempt to reduce the number of bits required to express an image by exploiting the spatial correlation between pixels. Second-generation techniques are the ensemble of techniques capable of achieving high compression

beyond the saturation level reached within the framework of information theory (10:1) [25]. This can be achieved because the assumptions of ergodicity and stationarity of image data are not satisfied. Furthermore, the entropy of the image data source is not known and depends heavily on the model used.

Another view of the difference between the first- and second-generation techniques is as follows. Image coding is basically carried out in two steps: first, image data are converted into a sequence of messages and second, code words are assigned to the messages. Methods of the first-generation put emphasis on the second step while methods of the second-generation put it on the first step and use available results for the second step [25].

In this section, we will concentrate on the major techniques of the first-generation namely, predictive coding, transform coding and hybrid coding and give a brief idea about second-generation image coding techniques. Vector quantization can be considered as a second-generation technique since the image data is converted into a set of blocks and then each block is assigned a codeword that represents the address of the best matching template from the codebook. Furthermore, VQ is capable of achieving high compression rate in the range of 32:1 with good quality of coded images [33, 34].

Figure 2.1 [5] attempts to very roughly illustrate typical bit rates for major coders and their corresponding computational complexity. Bit rate is expressed as the number of channel bits per pixel (bpp) and complexity is roughly equated to the number of floating point operations per pixel required to encode and decode an image. As seen from the figure, the complexity tends to increase sharply as the bit rate decreases. Predictive quantizers work best at rates no lower than about 2 bpp, transform, hybrid, and vector quantizers work well down to the tenths of a bpp region.

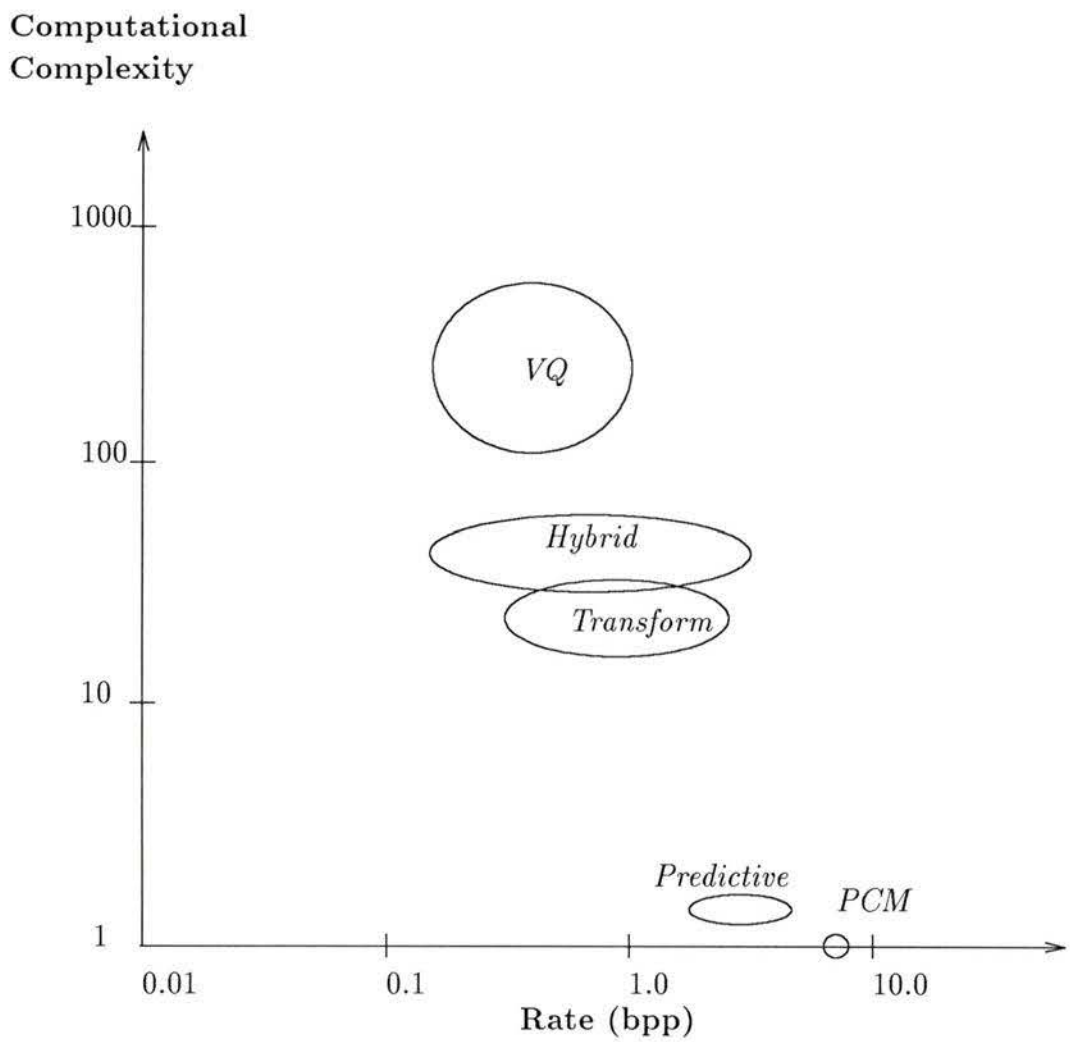


Figure 2.1: Computational complexity and rates of compression techniques

Excellent reviews of image coding methods can be found in [23] and [35]. Detailed expositions on the various coding techniques and bibliographies can be found in a book on digital waveform coding [24] and a book on image processing [36].

2.4.1 Predictive Coding

In basic predictive coding systems, a prediction of the sample to be encoded is made based on the previously encoded sample(s). The difference between the actual sample and the predicted sample is quantized and sent across the channel. The difference signal has a smaller variance than the input signal and hence is quantized with fewer bits.

The number of levels of the quantizer, N, defines the predictive coding scheme. In Delta Modulation N=2 while in DPCM N is greater than two. Although Delta Modulation has been used extensively in encoding other waveforms (e.g. speech), it has not found great use in encoding of pictures. Large amplitude changes occur abruptly at edges in images and the Delta Modulation coder takes too long to catch up, causing slope overload noise [24]. Figure 2.2 shows a typical predictive coder using a scalar quantizer. Design of a good predictor is critical for obtaining maximum compression in DPCM. In its simplest form, DPCM uses the coded value of the previously horizontal pixel as the prediction. However, more sophisticated predictors use the previous line as well as the previous frame of information. Typically, the three or four pixels nearest to the current one, and appearing before it (on the previous and current scan lines), are used for prediction.

Linear prediction theory [24] tells us how best prediction can be done with a linear weighted combination of a fixed number of previous samples. Knowl-

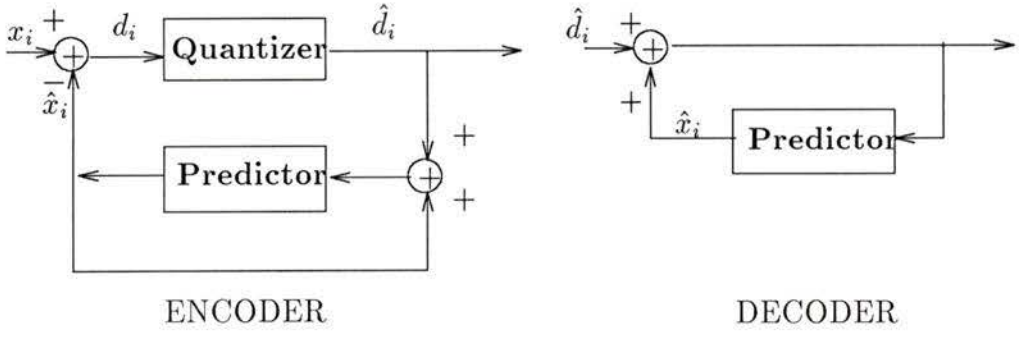
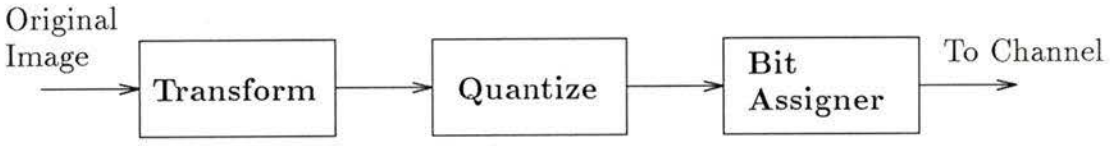


Figure 2.2: Scalar predictive quantizer

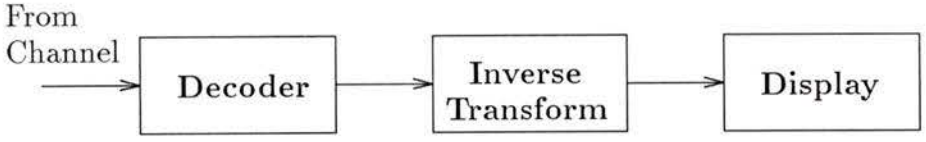
edge of the covariance of the source (assuming it is stationary) is needed to design the best predictor. However, since local image statistics change with time or pixel location in a scene, it is difficult to apply optimal prediction theory to design the predictor. Instead, the predictor is designed by a trial-and-error method.

The overall distortion for DPCM is approximately the same as the quantization distortion [24]. Three types of noise can be seen due to improper design of the quantizer of a DPCM coder. Granular noise is visible in flat areas due to coarse quantization, slope overload occurs if the dynamic range of the quantizer is small, and edge degradation if the amplitude change at an edge is gradual [35].

It has also been observed that about 3-4 bpp are needed for good visual quality with the nonadaptive DPCM systems [24]. This drops to about 2-3 bpp (average rate) with adaptation. Performance deteriorates rapidly below 2 bpp, with slope overload noise becoming highly objectionable.



Transform Encoder



Transform Decoder

Figure 2.3: Block diagram of a Transform Coder

2.4.2 Transform Coding (TC)

In transform coding, a picture is divided into subpictures and then each of these subpictures is transformed independently. The main purpose of transformation is to produce more independent coefficients. These coefficients are then quantized and coded for transmission. At the receiver, the received bits are decoded into transform coefficients and then the inverse transformation is applied to recover the original picture. Figure 2.3 shows a block diagram of a transform coder.

Much of the compression is as a result of dropping coefficients that are small, and coarsely quantizing others as required by the picture quality. The

performance of a transform coder is determined by the following important parameters [35] : size and shape of the subpictures, type of transformation used, selection of the coefficients to be transmitted and their quantization, and the bit assigner which assigns a binary word for each of the quantizer outputs.

Transformation and separate coding of each subpicture neglects the redundancies that exist between the subpictures. So, it is advantageous to have a large subpicture. However, for implementational simplicity as well as to exploit local changes in picture statistics and visual fidelity, a smaller subpicture is desirable [35]. These subpictures are typically 8×8 or 16×16 in size.

An optimum transform is the one which produces statistically independent transform coefficients. This requires the knowledge of image statistics of order higher than 2, which we do not have as yet. So, we seek a transformation that results in uncorrelated coefficients. Furthermore, since most of the compression results from dropping the coefficients with small energy, it is desirable to have a transform which compacts most of the image energy in as few coefficients as possible.

The Karhunen-Loeve transform (KLT) is the best linear transformation in decorrelating coefficients. Furthermore, it packs the energy of the input block most efficiently. That is if the transform coefficients are arranged in decreasing order of energy, the first k coefficients (for any k less than the transform dimensionality) have more energy with KLT than with any other energy-preserving transform [35].

The decorrelation and energy packing permit fewer bits to be used for quantizing the coefficients than are needed for directly quantizing the in-

tensity samples (for a given level of distortion). Bits are allocated to the coefficients according to their variances by scalar-quantizing each coefficient with the appropriate number of bits.

Although KLT is an optimum transform, it is rarely used in practice and a fixed transformation is preferred. This is because it is computationally expensive to compute the KLT for every block, and because the KLT must be transmitted with additional bits to the decoder.

Many other transforms have been invented which provide less correlated coefficients than the image itself and which are easier to implement. The most important ones are Fourier, Hadamard, Haar, Sine, Cosine and Slant transforms. One important difference between these transformations and the KLT is that these transformations do not depend on the statistics of the input image. The Discrete Cosine Transform (DCT) is found to be close to KLT in decorrelating coefficients and energy packing. This is why it seems to be the most widely accepted transform for image coding. For further details on the different types of transforms refer to [9].

The objective performance (MSE) of TC does not improve significantly when the block size is increased beyond 16×16. This indicates that the correlation in images does not extend significantly beyond about 16 pixels in the horizontal and vertical directions. It is found that the subjective quality does not improve with block sizes beyond 4×4 [35]. This is because due to allocating zero bits for the high frequencies, edges appear blurred. By increasing the block size, the percentage of edges in a block increases and hence the blurring effect becomes more visible.

High signal-to-noise ratios are achieved with TC around 2 bpp, while good visual quality is achieved at 1.5 bpp [38]. At lower rates, the block boundaries

become visible, and edges become blurred and disconnected when they cut across blocks. This blurring in edges is due to the fact that zero bits are being allocated for higher frequencies.

Transform coding may also be adaptive by matching the parameters of the coder to the statistics of the subpicture being coded. Adaptation can be made at the transform level, bit assignment level or at the quantizer level. Coding efficiency can be increased by about 20-50 % depending on the adaptation level and the coder complexity [35]. TC is considered a low-bit rate coding system since good visual quality can be achieved at rates below 1 bpp if some adaptation is included.

Many researchers have erroneously assumed that uncorrelated transform coefficients may be optimally quantized using only scalar quantizers [5]. However, Shannon's theory shows that even a memoryless i.i.d. (independent and identically distributed) source can be compressed using block codes [43]. This means that transform coders using a vector quantizer (VQ) would outperform those using scalar quantizers if it were possible to implement a VQ of the required dimension. Vaisey and Gersho [45] have used VQ to code the TC coefficients by grouping them into vectors requiring that all components of each vector have approximately the same perceptual importance.

A transform coder can be thought of as a suboptimal vector quantizer [38]. The scalar quantizers for the individual coefficients taken together constitute a product code VQ for the coefficients vector. This is not the optimal VQ for the given DM since it is structured as a product code. The optimal VQ can be found in the spatial domain itself without the need to transform the input block.

2.4.3 Hybrid Coding

Both DPCM and transform coding techniques have been used with some success in image coding. However, each technique has some attractive features and some limitations.

Transform coding systems achieve superior coding performance at lower bit rates, distribute the coding degradation more equally, show less sensitivity to data statistics and are less vulnerable to channel noise [19]. On the other hand, DPCM systems achieve a better coding performance at a higher bit rate and exhibit simplicity in design and high operating speed. The limitations of this system are the sensitivity of well-designed DPCM to picture statistics and the propagation of channel error on the transmitted picture.

By combining these two systems, we combine the hardware simplicity of DPCM with the higher compression achievable with TC. Habibi [20] has noted that the transform coefficients in transform coders are slightly correlated from block to block and proposed a hybridization of transform and predictive coders. Predictive scalar coders may be used to quantize transform coefficients rather than using scalar quantization. This way, block-to-block correlations may be exploited by further reduction in the bit rate without performance degradation. The best performance is obtained by designing separate predictive coders for each coefficient, leading to hybrid structures with better performance but higher complexity than conventional transform coders. Figure 2.4 shows such a hybrid structure.

2.4.4 Second-Generation Image Coding Techniques

Recent progress in the study of the brain mechanism of vision has led to a new class of coding methods capable of achieving compression ratios as high

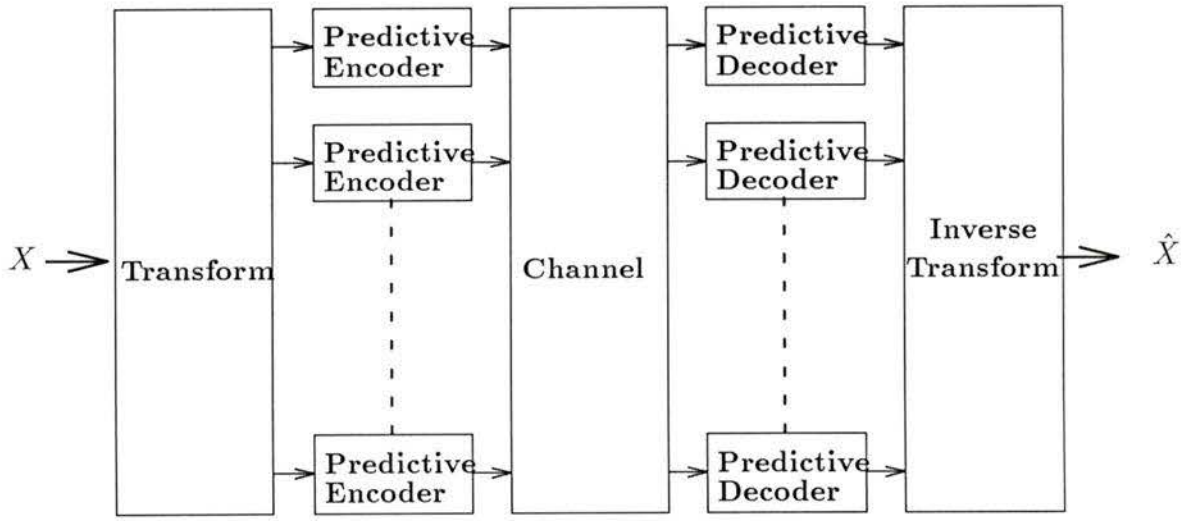


Figure 2.4: Hybrid Predictive/Transform Coder

as 70:1 [25]. These coding schemes try to match the human visual system and try to imitate its functions.

Two groups can be formed in this class : methods using local operators and combining their output in a suitable way and methods using contour-texture descriptions. Of the local operator based techniques are Pyramidal Coding [7] and Anisotropic Nonstationary Predictive Coding [47]. Both of these techniques combine features of predictive and transform coding methods, however, they are considered as second-generation techniques.

Pyramidal Coding is considered as a second-generation technique since its hierarchical structure is similar to that of the nervous system and the functions it uses are close to those of the human visual system. Anisotropic Nonstationary Predictive Coding is considered as a second-generation technique because of the extensive use of the properties of the human visual

system and because of the high compressions achieved.

Contour-texture oriented techniques attempt to segment the image into textured regions surrounded by contours such that the contours correspond, as much as possible, to those of the objects in the image [25]. Contour and texture information are then coded separately. Contours may be extracted in two ways : by region growing or by edge detection techniques. For further details on second-generation image coding techniques refer to [25].

In addition to these techniques which try to imitate the human visual system, some other techniques appeared in the literature which claim very high compression ratios. A new method for image compression using fractals has claimed compression ratios in excess of 1,000 to 1 [6]. First, the main image must be broken into smaller sub-images based on colour separation, edge-detection and a variety of techniques similar to those used in movie colourization. Then, these sub-images are looked up in a library of pre-generated fractals or generated using the collage theorem [6]. A fractal is a class of geometrical figures that provides both a description and a mathematical model for many of the seemingly complex forms found in nature. The statistical similarity is the essential quality of fractals. This new technique of compression is based on many heuristic assumptions and requires extremely numerically intensive functions. The main difficulty in this technique is to separate the original image into segments that can be generated by fractals. It is still not clear in the literature how this can be done efficiently.

Chapter 3

Vector Quantization

3.1 Introduction

According to Shannon's rate-distortion theory [43], one can always obtain a better performance by coding vectors instead of scalars, even though the data source is memoryless. In the past, scalar quantizers were the most common data compression system due to their simplicity and good performance at sufficiently high rate. In addition, few design techniques have existed for vector quantizers. Vector Quantization (VQ*) has begun gaining interest with the advent of the codebook design algorithm introduced by Linde, Buzo, and Gray [27] called the LBG algorithm. VQ has been found to be an efficient coding technique for speech and image waveforms [1, 18, 32]. However, it requires a high computational complexity. In order to reduce VQ's complexity, many suboptimal techniques have been introduced by enforcing a structural constraint in the codebook.

In this chapter, we first define vector quantization as an image coding technique and then discuss the optimality criteria for vector quantizers. Next,

*The abbreviation VQ stands for either *Vector Quantization* or *Vector Quantizer*

we present the LBG algorithm and discuss some of the important factors that could affect codebook design. This is followed by presenting some of the sub-optimal techniques that aim at reducing the high computational complexity of VQ. Finally, we discuss briefly vector quantizers that exploit the correlation between neighboring blocks.

3.2 Definition of VQ

A Vector Quantizer (VQ) can be defined as a mapping Q of K-dimensional Euclidean space R^k into a finite subset Y of R^k. Thus,

$$Q : R^k \rightarrow Y, \tag{3.1}$$

where $Y = \{y_1, y_2, \cdots, y_N\}$ is the set of reproduction vectors, and N is the number of vectors in Y. VQ can also be defined as a combination of two functions : a coder and a decoder. The coder C is defined as a mapping of R^k into the index set J, and the decoder D is defined as the mapping of J into the output set Y where

$$J = \{1, 2, \cdots, N\}.$$

Thus,

$$C : R^k \rightarrow J \quad \text{and} \quad D : J \rightarrow R^k \tag{3.2}$$

An input vector x is quantized by selecting the nearest reproduction vector from the codebook according to a distortion criterion $d(x, y_i)$.

$$Q(x) = \min^{-1\dagger} d(x, y_i) \quad i = 1, \cdots, N \tag{3.3}$$

†The inverse minimum notation means that we select the index i giving the minimum.

This process is done by the encoder which then sends the address of the reproduction vector nearest to x over the channel. The decoder receives the address of the reproduction vector and simply performs a table look-up operation on the same copy of the codebook that the encoder uses. As seen from above, VQ is well suited for applications that require very fast decoding.

If the codebook has N codewords each consisting of k pixels, then the rate of the quantizer in bits per symbol is

$$R = \frac{1}{k} \log_2 N \Rightarrow N = 2^{Rk} \tag{3.4}$$

From the above relationship, we see that the codebook size (N) and hence the complexity of VQ increases exponentially with the rate and block size. In order to reduce the search complexity, the codebook size must be kept small, which in turn puts limits on the block size that can be used (e.g. $k \leq 16$). This puts limits too on the utilization of the source coding theorem which states that the larger the value of k, the closer we approach the rate distortion function.

Since the codebook size must be kept small due to practical reasons, the performance of VQ is intimately tied to the codebook used for quantization. Figure 3.1 shows a memoryless basic vector quantizer.

3.3 Optimal Vector Quantization

A Vector Quantizer is considered to be optimal if it minimizes average distortion $E[d(x, Q(x))]$, where x is the input vector and $Q(x)$ is the best reproduction vector matched to it. There are two necessary conditions for a VQ to be optimal [18] :

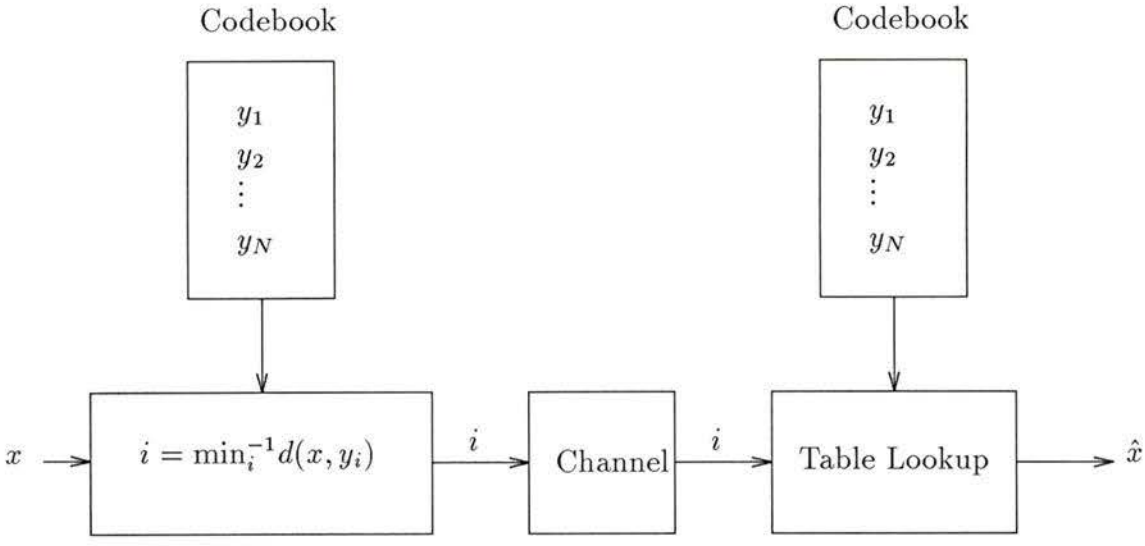


Figure 3.1: Basic vector quantizer

Condition 1 : Given a codebook $Y = \{y_i : i = 1, \cdots, N\}$ and a specific decoder D, the encoder encodes a vector x by selecting the codeword which yields the minimum distortion.

$$C(x) = \min^{-1} d(x, y_i) \quad i = 1, \cdots, N \tag{3.5}$$

By applying this condition, the encoder partitions the input space into several partitions $\{S_i\}$, where the vectors that are encoded to the same codeword are placed in the same partition. Such partitions, according to a minimum distortion rule, are called Voronoi or Dirichlet partitions [15].

Condition 2: Given the Voronoi partitions of the input space, we need to determine the codebook that minimizes the overall distortion. To minimize the overall distortion, the distortion contributed from each cell need to be minimized. This could be done by assigning the generalized centroid of each

partition as a codeword in the codebook.

$$\text{Cent}(S_i) = \min E(d(x,y) \mid x \in S_i) \quad \text{for all } y \in R^k \tag{3.6}$$

The generalized centroid of partition S_i under a distortion measure $d(\cdot,\cdot)$ is unique if the distortion is strictly convex[16]. Since the source is assumed to be ergodic, the conditional expectation may be replaced by a simple sample average,

$$\text{Cent}(S_i) = \frac{1}{\parallel S_i \parallel} \sum_{x_i \in S_i} x_i \tag{3.7}$$

where $\parallel S_i \parallel$ denotes the cardinality of the partition.

These two conditions are the key of the codebook design algorithm. The codebook is iteratively optimized for the old encoder and then a minimum distortion encoder is used for the new codebook. However, the two conditions are not sufficient to ensure optimality. Lloyd [28] constructed two scalar quantizers satisfying the conditions showing that one gives a lower distortion than the other. This means that these two conditions ensure local but not global optimality.

3.4 The LBG Algorithm

The two necessary conditions mentioned above lead naturally to an iterative algorithm for optimal VQ design called the LBG algorithm [27]. The algorithm works either with a probability density function (pdf) or with a training set (t-set) of sample vectors from the true distribution. Usually, the LBG algorithm is used with a t-set since multidimensional pdfs are hard to

determine and describe analytically, and even harder to integrate. When a t-set is used, all expectations are to be interpreted as sample averages.

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

1. Let N = number of codewords in the codebook; distortion threshold $\epsilon \geq 0$. Assume $\hat{A}_0$ as the initial codebook which has N codewords, and a training sequence $(x_j; j = 0, 1, \cdots, \text{n-1})$, and m = number of iterations, set to zero, and $D_{-1} = \infty$ (D_i is the average distortion at iteration i).

2. Given $\hat{A}_m = (y_i; i = 1, \cdots, N)$, find the minimum distortion partition $P(\hat{A}_m) = (S_i; i = 1, \cdots, N)$ of the training sequence : $x_j \in S_i$ iff $d(x_j, y_i) \leq d(x_j, y_l)$, for all l. Compute the average distortion

$$D_m = \frac{1}{n} \sum_{j=0}^{n-1} \min_{y \in \hat{A}_m} d(x_j, y)$$

3. If $(D_{m-1} - D_m)/D_m \leq \epsilon$, stop the iteration with $\hat{A}_m$ as the final codebook otherwise continue.

4. Find the optimal codebook $\hat{x}(P(\hat{A}_m)) = (\hat{x}(S_i); i = 1, \cdots, N)$ for $P(\hat{A}_m)$ where $\hat{x}(S_i)$ is the centroid for the partition cell S_i.

5. Set $\hat{A}_{m+1} = \hat{x}(S_i)$, increment m to $m + 1$, and go to (2).

Since D_0 is finite, the distortion either decreases or remains the same at each iteration of the algorithm. The distortion threshold ϵ determines the number of iterations before the algorithm is terminated. If $\epsilon = 0$, the

algorithm will not terminate unless it reaches a fixed point (i.e. applying the algorithm to a codebook C yields codebook C). For $\epsilon > 0$, the algorithm must stop. For a t-set of finite size n and for convex DMs, this occurs in a finite number of steps [16]. ϵ is usually given the value of 0.001.

The LBG algorithm does not ensure convergence to the globally optimal codebook. However, it provides us with a set of fixed points which is arbitrarily close, depending on the t-set size, to the set of fixed points for the true distribution. Thus, the larger the t-set is, the closer the codebooks are to those obtained with the true distribution. Experience gained in building fully searched codebooks for speech and random sources suggests that on average at least 50 to 100 training vectors are needed for each codevector [8, 27].

The designed codebook depends on the choice of an initial codebook. Different choices of initial codebooks produce locally optimal codebooks. The locally optimal codebook which gives the least average distortion is deemed the globally optimal one.

3.4.1 Distortion Measures and their Centroids

A generalized centroid exists for any convex distortion measure, and is unique if the measure is strictly convex [16]. The centroid is known for the class of weighted r^{th} power distortion measures (§2.4), where r is an even integer. Baker [5] shows that for an even r, the centroid is the sole real-valued root of a monotone polynomial of degree $(r-1)$. The weighted squared error measure is defined as

$$d(x,y) = (y-x)^t W(x)(y-x), \tag{3.8}$$

$$\text{where } W(x) = \begin{bmatrix} w_1 & & & 0 \\ & w_2 & & \\ & & \ddots & \\ 0 & & & w_n \end{bmatrix} \tag{3.9}$$

w_i is the weight assigned to the i'th component of the error vector $(y - x)$. The centroid $\bar{X}$ of the vectors $x_j, j = 1, \cdots, N$ is

$$\bar{X} = \left\{ \sum_{j=1}^{N} W(x_j) \right\}^{-1} \cdot \left\{ \sum_{j=1}^{N} W(x_j) x_j \right\} \tag{3.10}$$

For unweighted squared error, W becomes the identity matrix, and

$$\bar{X} = \frac{1}{N} \sum_{j=1}^{N} x_j \tag{3.11}$$

Generalized centroids are known for only few DMs, most of them belonging to the above class. The centroid for the Mean Absolute Error ($d(x, y) = \sum_{i=1}^{k} |x_i - y_i|$) is also known [38]. It is given by the vector formed by the medians of its components.

In order to fasten and simplify the vector quantization operation, Murakami [31] adopted the following distortion measure defined as :

$$d(x, y) = \max_k |x_k - y_k| \tag{3.12}$$

The centroid used for this measure was the same as the one used for unweighted squared error.

3.4.2 Codebook Initialization

The performance of the final codebook could depend strongly upon the initial codebook chosen, especially for sources with multiple minima [17]. Choosing an initial codebook that leads to the best final codebook is still an open problem. To generate an initial codebook, the following techniques were used in the literature.

1. One possible way to initialize the LBG algorithm is by using a codebook previously designed for some other purpose. Alternatively, one might initialize with evenly spaced codewords in the vector space [13].

2. Random Initialization : In this technique, N codevectors are chosen at random from the t-set. This method was first used in the k-means clustering algorithm [29]. In practice, the *Random Initialization* is implemented by choosing evenly spaced elements in the training sequence (e.g. $x, x_{k+1}, x_{2k+1}, \cdots, x_{(N-1)k+1}$, where x_i is the training vector, m is the number of training vectors, N is the number of codewords desired, and $k = m/N$).

3. Product Codes [1, 18] : Let us assume that we have a collection of codebooks C_i, $i = 0, 1, \cdots, m-1$, each containing N_i vectors of dimension K_i, and having rate $R_i = \log_2 N_i$ bits per vector. Then, the product codebook C is defined as the collection of all $N = \prod_i N_i$ possible concatenations of m codewords taken successively from the m codebooks C_i. The dimension of the product codeword is $k = \sum_{i=0}^{m-1} k_i$.

   Thus, for example, a k-dimensional vector quantizer of rate R bits per vector can be formed using a scalar quantizer k times in succession,

with rate R/k.

4. Splitting Technique [18, 27]: In this technique, the final codebook is constructed recursively from a smaller codebook having the same dimension. First, the optimum 0-rate code is found by finding the centroid of the entire training set. Then, a codebook of size two is formed by splitting the first codeword into two. In order to ensure that the distortion will not increase, the original codeword is kept in the new codebook and the new codeword is formed by scaling the energy of the first. Having formed an initial codebook of size two, the LBG algorithm is then used to design an optimal codebook of size two. Using the same technique, an initial codebook of size four can be formed by adding two additional codewords resulting from perturbations to the two optimal codewords. Then, an optimal codebook having four codewords is designed. For an initial codebook of size N, this process is repeated $\log_2 N$ times.

5. The Pairwise Nearest Neighbor (PNN) Algorithm : The PNN clustering algorithm was introduced by Equitz [13] as an alternative to the LBG algorithm for vector quantizer design. It generates codebooks from a training set in almost 5 percent of the time required by the LBG algorithm. The PNN algorithm starts by assigning a separate cluster for each training vector. Then, it merges two clusters at a time until the desired codebook size is achieved. When clusters are merged, the centroid is used to represent the vectors of the merged cluster. Once clusters have more than one member, things become more complicated. A tradeoff is made between merging close clusters and the number of

vectors affected by the merging process.

Even though the PNN algorithm is to be considered unacceptable because it usually does not generate an optimal codebook, it appears to be an excellent alternative to random initialization for the LBG algorithm [13]. By using it as an initializer for the LBG algorithm, total coding error is reduced and the LBG algorithm converges in fewer iterations.

3.4.3 The Empty Cluster Problem

The empty cluster problem occurs if no training vectors map to one or more of the codewords in the codebook. This problem does not occur in the first iteration of the codebook design but it could occur in later iterations. To eliminate this problem, Cuperman [11] suggested that if the number of training vectors in a cluster is zero, then the codeword of that cluster is discarded. This codeword is replaced by a perturbation of the centroid of the nonempty cluster with the largest partial distortion (i.e. the sum of the distortions for the training vectors in the cluster).

Ramamurthi [38] adopted the modified algorithm of Cuperman but with a new definition of *empty*. A cluster with one training vector is also considered to be empty and is also discarded. If m clusters are discarded, then the largest m clusters in partial distortion are split. This process does not guarantee the convergence of the LBG algorithm. If a divergence is noted by an increase in the distortion in the next iteration, then the old definition of empty cluster is used and the algorithm begins to converge again.

The empty cluster problem could occur too when using the splitting technique for codebook initialization. The problem occurs almost always during

the algorithm's first iteration following a splitting operation, indicating that empty clusters are often spawned by poor splitting decisions [5]. For example, splitting a codevector with only a few training vectors in its partition will often cause an empty cluster, since it is very likely that all of them will be mapped into only one of the two new codevectors. In order to circumvent this problem, Baker [5] suggested a selective splitting method where only a fraction of the total number of codevectors is split, choosing those which contribute the most distortion.

3.5 VQ Complexity and Suboptimal Schemes

In order to improve the performance of block source codes and hence vector quantizers, the block size k needs to be increased. This in turn results in an increase in the codebook size since it grows exponentially with the block size. Furthermore, a large enough codebook is needed to provide a good vector quantizer. This is because the codebook needs to be rich enough of templates that can properly represent edges with different locations and orientations in addition to vectors that do not contain edges.

VQ has inherent complexity in the encoding process. For every vector that is to be encoded, a full search of the codebook need to be performed in order to find the best matching codeword according to a distortion criterion. This full search process is totally inefficient for large codebook sizes. In addition, the limits of present computer technology are quickly reached for small k. For example, for a rate of 1 bpp and a block size of 16 pixels, the codebook size is $2^{16} = 65,536$ codewords. This requires 1 Mbyte for codebook storage and $65,536$ distortion calculations per vector. In order to reduce search complexity, suboptimal techniques are used.

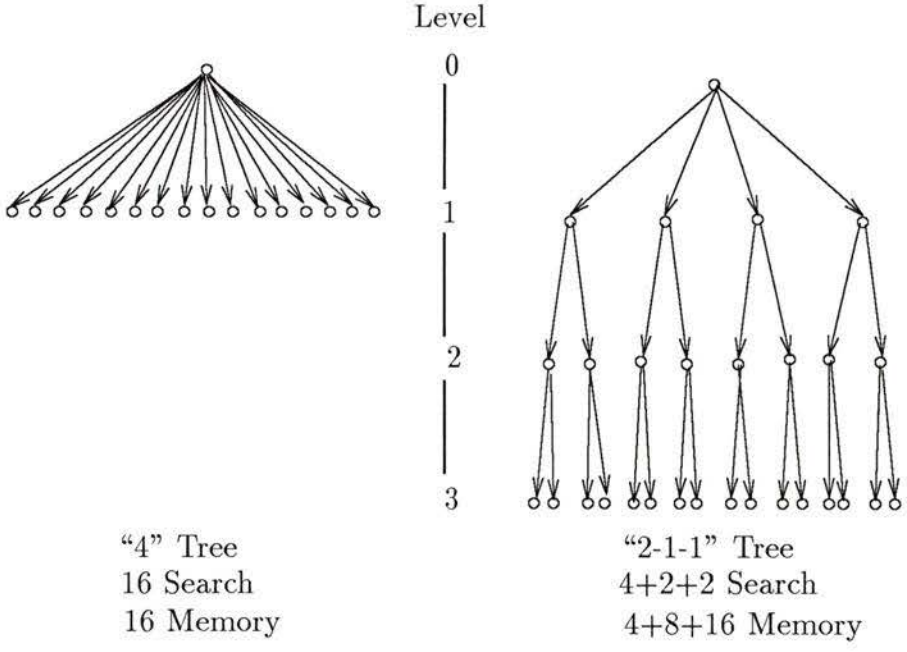


Figure 3.2: Tree-structured codebooks

3.5.1 Tree Structures

One way to reduce the search complexity is by the use of tree structures. A tree structure is imposed upon the codebook and hence only a few distortion calculations are made at each level of the tree. This technique was first suggested by Buzo [8] for speech VQ coding. Figure 3.2 [5] shows the structure of two codebooks of size $N = 16$. A tree is characterized by the number of bits required to select a branch at the current node of the current level. Thus, the full searched tree in Figure 3.2 is a "4" tree, and the other is a "2-1-1" tree.

Search time for tree-structured codebooks falls somewhere between a number which grows linearly with $n = \log_2 N$ (L-level binary trees) to ex-

ponential in n (1-level, or full searched trees). However, the memory needed to store the codebook is minimum for a full searched tree and maximum for the binary tree [5].

Codebooks with tree structures are easily built using the LBG algorithm. Suppose that we want to design an L-level tree codebook, with b_i bits per level ($i = 0, 1, \cdots, L$). B_0 always equals 0 bits and corresponds to the centroid of the training sequence. The codevectors for level 1 are found by using the LBG algorithm and the splitting technique for codebook initialization. Then, the training set is mapped into 2^{b_1} clusters. For each cluster, a codebook of size 2^{b_2} is built. The procedure then repeats till we reach the final level where we have a total number of codewords equivalent to the codebook size N.

Since training vectors are seldom allocated evenly among the codevectors of the tree at a certain level, some codevectors may face severe shortages of training vectors and hence leading to the empty cluster problem. Currently, there is no systematic way to avoid this problem. Some suggestions were made by Baker [5] to avoid this problem.

By drastically cutting search and design time, tree-structured codebooks lack even local optimality. This is due to the structure imposed on the codebook. In speech coding applications, 9- to 10-bit binary tree searched codebooks yield performance comparable to 8-bit full searched codebooks [8]. The performance of tree-structured codebooks can vary widely depending upon their particular structure. Baker [5] compared the performance of several tree structures. He reported the following characteristics for the trees that performed the best :

- Use quaternary rather than successive binary decisions

- Have the fewest number of levels
- Are tapered (i.e. the number of levels per node does not decrease as one progresses down the tree from the root node).

He also showed that for an 8-bit tree-structured codebook, the 3-5 tree has the top performance. It performed 0.25 dB less than the 8-bit full search codebook and 0.83 dB better than the 7-bit full search codebook.

Other techniques can be used to design tree-structured codebooks. One approach is to design a full search codebook and then to use a bottom-up approach to construct a tree search into the codebook [18]. This could be done by grouping the codewords into close disjoint pairs and then forming the centroids of the pairs as the node label of the immediate ancestor of the pair.

3.5.2 Product Structures

Another useful way for reducing search complexity and memory requirements for vector quantizers is by using codebooks with product structures. Product structures enjoy larger and richer codebooks which achieve better performance than the basic VQ for about the same level of complexity [5]. Such structures yield quantizers with vector valued product codebooks: channel symbols are selected independently by two or more encoders, then concatenated to form a larger symbol.

Gain/Shape VQ is one example of product structures which have been successfully used for speech waveform and LPC model compression [8, 40]. Separate coders are used to code the "shape" and "gain" of the waveform, where the shape is defined as the original input vector normalized by re-

moval of a "gain" term such as energy in a waveform coder or LPC residual energy in a vocoder. Baker and Gray [3, 5, 18] introduced new product structures for image waveform compression, namely the Mean/Shape Vector Quantizer (M/SVQ), the Mean/Residual Vector Quantizer (M/RVQ) and the Mean/Reflected Residual Vector Quantizer (M/RRVQ). These product structures are based on the idea of removing the mean from the block and encoding the mean and the residual block separately.

Mean/Shape Vector Quantizer (M/SVQ)

Studies by Baker and Ramamurthi [3, 37] have shown that the basic VQ codevectors can be classified into at least two classes : shade and edge vectors. The majority of these codevectors are from the shade class (almost 75%). Shortages of codevectors from either class causes distortion. Too few shade codevectors causes blocking in smooth regions and too few edge codevectors causes staircased edges and the "washed out" look. In order to avoid the blocking distortion (i.e. the appearance of the block boundaries), around 64 codevectors need to be allocated for the shade class. By removing the mean values from the vectors and then encoding the mean values and the zero-mean vectors separately, very few codewords need to be allocated in the zero-mean or shape codebook. This is because the shade information is principally characterized by a scalar. By doing this, more vectors are allocated to represent non-shade blocks without regard for their absolute intensity values. So, vectors having the same structure with different intensity values will map to the same codeword in the shape codebook.

Gray [5] has observed that the M/SVQ is the optimum mean/shape prod-

uct coder when operated under the squared error criterion. It encodes each source block into the best mean/shape pair (m_i, S_j), where m_i is selected from a samples mean codebook, C_m, and S_j from a shape codebook, C_S. Under the squared error criterion, the optimum encoder must find the pair (m, S) which minimizes the distortion

$$\begin{aligned} d(X, S + m\mathbf{1}) &= [X - S - m\mathbf{1}^{\ddagger}]^T[X - S - m\mathbf{1}] \\ &= [X^TX - 2X^TS + S^TS] + [X^TX - 2mX^T\mathbf{1} + m^2\mathbf{1}^T\mathbf{1}] \\ &\quad -X^TX + 2mS^T\mathbf{1} \\ &= |X - <X>\mathbf{1} - S|^2 + k[<X> - m]^2 \\ &\quad +2k <S> (m - <X>) \qquad (3.13) \\ &= |X - S|^2 + k[<X> - m]^2 - k <X>^2 \\ &\quad +2mk <S> \qquad (3.14) \end{aligned}$$

By imposing the M/SVQ's constraint that all shape codevectors have zero mean (i.e. $<S>= 0$), the optimum encoding rule can be achieved by achieving the following two independent encoding rules :

$$m = \min_{m \in C_m}{}^{-1} [<X> - m]^2 \qquad (3.15)$$

$$S = \min_{S \in C_S}{}^{-1} |X - <X>\mathbf{1} - S|^2 \qquad (3.16)$$

$$= \min_{S \in C_S}{}^{-1} |X - S|^2 \qquad (3.17)$$

As seen from above, it is not necessary to remove the mean $<X>\mathbf{1}$ from a vector X to find its locally optimum shape reproduction, s. Figure 3.3

$^{\ddagger}\mathbf{1}$ is a k-dimensional vector consisting of 1's (i.e. $\mathbf{1} = (11\cdots1)$).

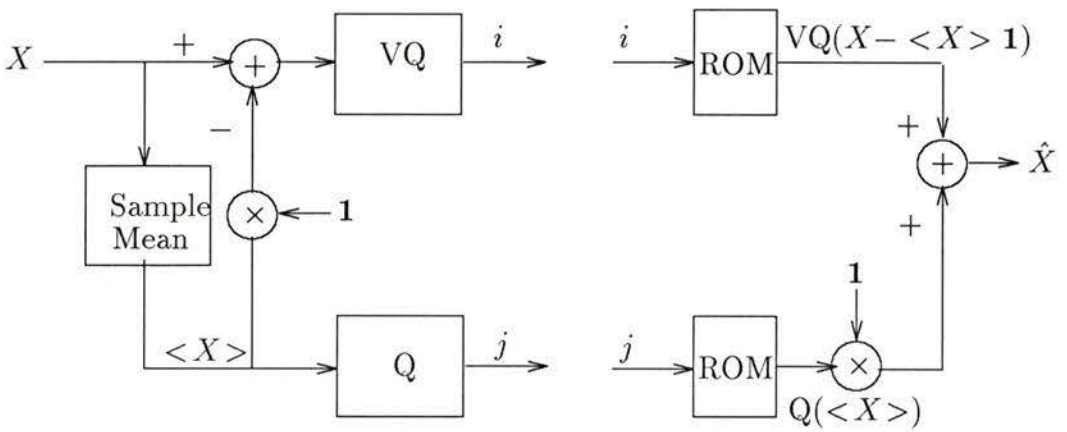


Figure 3.3: Mean/Shape Vector Quantizer (M/SVQ)

[5] shows the block diagram of a Mean/shape Vector Quantizer. Murakami [31] also used a similar structure to the M/SVQ by normalizing each block's sample variance after mean removal. Thus, each vector of the video signal is converted into a vector with zero mean and unit variance. The mean and variance are quantized by a scalar quantizer or two-dimensional quantizer.

Mean/Residual Vector Quantizer (M/RVQ)

The M/SVQ is a locally optimum product code provided that its shape source is constrained with zero mean blocks. For other distortion measures that are perceptually more faithful, it is not yet known how to design a M/SVQ without sacrificing its locally optimum product codebook structure. For this reason, the M/RVQ is introduced. The M/RVQ quantizes the mean values first and then subtracts the quantized means from the vectors. Figure 3.4 [5] shows the block diagram of a Mean/Residual vector quantizer.

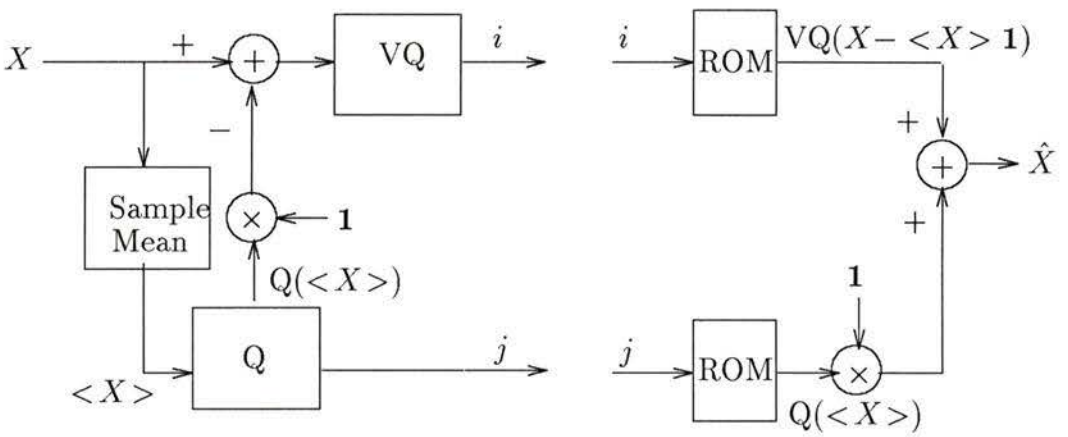


Figure 3.4: Mean/Residual Vector Quantizer (M/RVQ)

The M/RVQ can be designed for any distortion measure possessing a centroid. Furthermore, it does not require its codevectors to have zero mean. Although the M/RVQ is not a locally optimum product vector quantizer, Gray [5] has shown that under the squared error distortion measure, the M/RVQ performs no worse than the M/SVQ for arbitrary sources. The M/RVQ could perform even better than the M/SVQ by encoding the block sample means more accurately than the M/SVQ. This is because an error committed by the sample mean encoder could be compensated for by the residual vector quantizer. However, the M/RVQ is slightly more complex to implement than the M/SVQ. In the M/RVQ, we are forced to remove the quantized mean before quantizing the residual block. However, this step is not necessary in the M/SVQ.

Experimental results on the M/RVQ by Baker [5] have shown that the M/RVQ images have higher SNR than both basic VQ and adaptive DCT

coders for the same bit rate. Furthermore, the M/RVQ reduces distortions present in basic VQ coders of comparable rate and complexity and yields outstanding results for images having fine detail in smooth regions.

Mean/Reflected Residual Vector Quantizer (M/RRVQ)

The mean/reflected residual vector quantizer is an example of a triple product structure (mean, reflection, and residual). By changing the order of rows and/or columns in a block, one M/RRVQ residual codevector can be used to represent four M/RVQ residual codevectors preserving intrablock correlation and pixel adjacency. Figure 3.5 shows an example of how one vector can be used to represent four different vectors by reflecting the rows and/or the columns.

Thus, a residual codebook of this type should perform better than a conventional M/RVQ residual codebook of the same size.

Since r^{th} power error measures are per-pixel measures, they are not affected by rearranging the pixels in a block. The rows and/or columns of a block are reordered in such a way that the gross horizontal and vertical gradients of its residual always point the same direction. Baker [5] has shown experimentally that the M/RRVQ consistently outperforms both the basic VQ and the M/RVQ, riding about 0.5 dB above the M/RVQ and from about 1.0 to 2.0 dB above the basic VQ. It also reduces both blocking and edge distortions present in the M/RVQ encoded image.

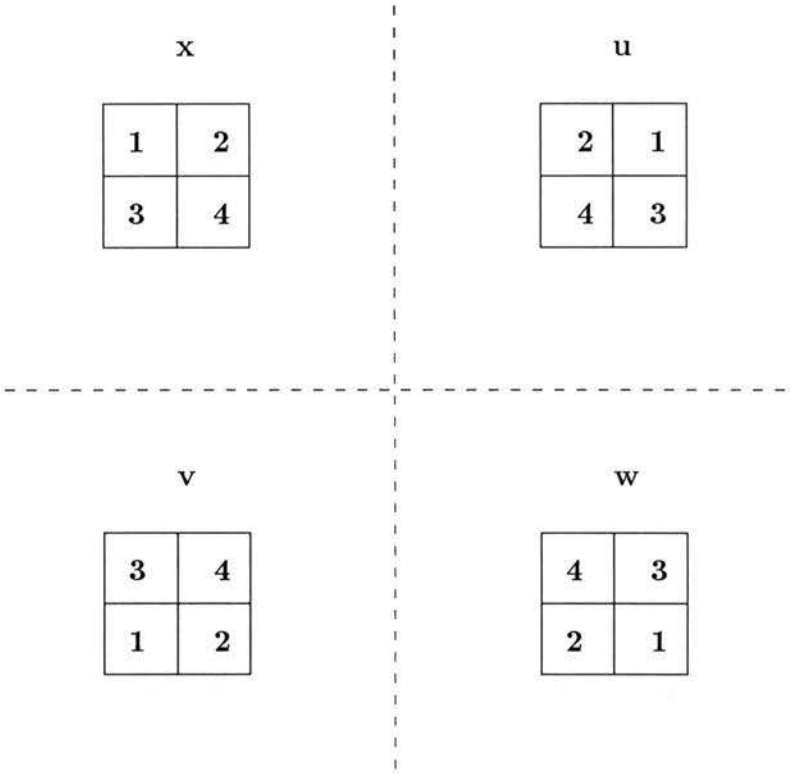


Figure 3.5: Source block requiring four distinct M/RVQ residual codevectors but only one M/RRVQ residual codevector

3.5.3 Classified Vector Quantization (CVQ)

One of the main problems in vector quantization of images is edge degradation. The use of simple distortion measures like the mean squared error which does not give preferential treatment to edges results in edge degradation. Since the codebook is limited in size, the MSE attempts to make the best use of the available codevectors. Most of the codevectors are closer to being classified as shade blocks or blocks with low detail. Hence, some of

the edge blocks are coded with a shade block that gives the least MSE. This results in the visibility of the boundaries of the edge blocks since the adjacent blocks, which are most likely shade blocks, are coded with much lower MSE. This effect is called the stair-case effect.

Ramamurthi [38, 39] suggested a solution to reduce the staircase effect by classifying vectors before quantization. Thus, by classifying vectors according to edge orientation and location, edge integrity is preserved. This is because the set of vectors that will be used to match the input vector have the same edge orientation and location and hence the best one of them is used by a DM. Thus, the DM is relieved of the burden of ensuring edge integrity, and simple distortion measures like the MSE can be used safely and effectively.

Edge-Oriented Classification

Edge-oriented classification has a basis in the psychology of vision. There are special cortical cells in the eye that respond to edges in the image being viewed. The presence of special cells dedicated to the detection of edges and their orientation implies that edges are very important perceptually and gives a justification for the classification of blocks prior to encoding [38].

An input block x is classified into the following main classes :

- Shade : The block has no significant gradient in it.
- Midrange : The block has moderate gradient, but no definite edges.
- Edge : The block has a distinct edge running through it. This class is further classified into different classes depending on the location and orientation of the edge. For small (around 4×4) blocks, an edge is obtained by approximating it as a straight line and specifying its angle

and location. Edges of the same orientation and location are further divided into two classes depending on whether the intensity transition is from left to right or from right to left for vertical edges.

- Mixed Edge : The block has significant gradient, but no single definite edge.

For 16-dimensional vectors, an edge is approximated by four angles : 0° (horizontal edges), 90° (vertical edges), 45°and $-45°$ (diagonal edges). Both horizontal and vertical edges can occur at 3 locations and diagonal edges at 4 locations, thus resulting in 31 different classes.

It is not important perceptually or from the point of view of coding quality to separate the shade and midrange classes. However, it serves in reducing the encoding complexity. Furthermore, the division of edges of the same orientation and location into two complementary classes is also not motivated by any perceptual feature and it is done for the sake of reducing the encoding complexity since it nearly doubles the number of classes [38].

Classified Encoding

The classified vector quantization (CVQ) system is shown in Figure 3.6 [38]. The input vector x is classified by the classifier into one of M classes, $Z_i, i = 1, 2, \cdots, M$. The codebook C is also classified into subcodebooks, $C_i, i = 1, 2, \cdots, M$. Each subcodebook has N_i codevectors, denoted by $Y_{ij}, j = 1, 2, \cdots, N_i$ and,

$$\sum_{i=1}^{M} N_i = N, \tag{3.18}$$

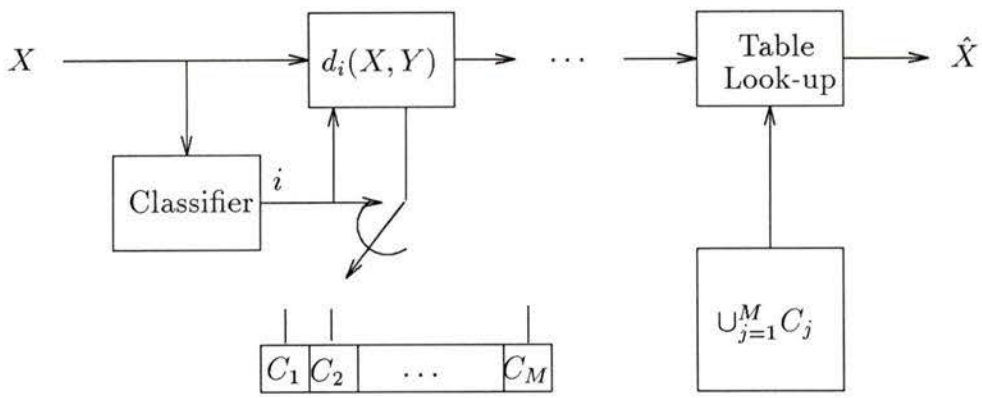


Figure 3.6: Classified Vector Quantizer (CVQ)

where N is the total codebook size. While the classifier attempts to ensure edge integrity by restricting the search to the appropriate subcodebook, the best matching codevector from the subcodebook is found by a distortion measure. In general, the DM may be different for each class. The encoding complexity for the CVQ is considerably reduced compared to that for the ordinary VQ system. If we denote the probability of a vector being in the i^{th} class as P_i, then the average number of nearest neighbor (NN) computations per vector for this system is only

$$\sum_{i=1}^{M} P_i N_i \tag{3.19}$$

compared to N for the ordinary VQ system. The worst case complexity, which is more meaningful for real-time applications, is given by

$$\max_i N_i \quad \text{NN Computations Per Vector} \tag{3.20}$$

Thus, the practicality of vector quantization is improved even as we attempt to improve the perceptual basis of the coding by classification [38].

CVQ Codebook Design

The CVQ coder is equivalent to an ordinary vector quantizer with a special distortion measure. Since a single overall distortion measure is not defined for CVQ but only individual DMs $d_i(X, Y)$ for the classes, the overall average distortion is obtained as a weighted sum of the average distortions in the individual classes. The average distortion per pixel for the ith class is given by

$$D_i = \frac{1}{k}\mathrm{E}[d_i(x, Q(x))|x \in Z_i] \tag{3.21}$$

and the overall average distortion, D, is given by

$$\sum_{i=1}^{M} P_i D_i \tag{3.22}$$

where P_i is the probability of selecting class i. These expressions are sufficient to compute the overall average distortion, but they do not tell us how the subcodebooks ought to be designed so that the overall distortion D is minimized.

The overall distortion measure can be defined as follows :

$$d(X, Y) = \begin{cases} d_i(X, Y) & \text{if } X \text{ and } Y \text{ in same class} \\ \infty & \text{otherwise} \end{cases} \tag{3.23}$$

Searching through the codebook C using this DM is exactly equivalent to classified encoding [38]. The problem of finding the codebook $C(= \cup C_i)$ that

minimizes the distortion D is now a standard optimal VQ problem. The only difficulty is that the optimal set $\{N_i\}$ that minimizes the overall distortion D is not known. If this set is known, then the task will be accomplished. The training set is classified into M classes, and the ith training set is used to design the subcodebook C_i. This is because with the LBG algorithm, only those training vectors that belong to Z_i need to be used to design C_i, since all other training vectors yield infinite distortion with the codevectors Y_{ij} in C_i.

Using asymptotic theory, Ramamurthi [38, 39] has shown that the asymptotically optimal set $\{N_i^*\}$ satisfies the relation

$$\frac{P_i D_i^*}{N_i^*} = \text{constant} \tag{3.24}$$

This implies that for optimality the average partial distortion ($= P_i D_i$) per codevector needs to be the same for every class. If we denote the optimal MSE distortion for class i by B_i^*, then the optimal distortion for the WMSE D_i^*, with weighting matrix $W_i = w_i I$ where w_i is a scalar, is given by $w_i B_i^*$ and the relation becomes

$$\frac{w_i P_i B_i^*}{N_i^*} = \text{constant} \tag{3.25}$$

The larger the weight for a class, the more the importance attached to the average distortion of that class, and the larger the subcodebook size for that class. To determine suitable weights for the different classes, subjective experiments need to be done. Ramamurthi [38] suggested that the average MSE distortion B_i^* should be the same for all edge classes. He also noted that the average MSE distortion achieved in the shade and midrange classes

was much lower than that achieved in the edge classes even though their subcodebook sizes (64 and 256) were a small fraction of the total codebook size N (8192). He also found experimentally that CVQ is successful in overcoming the edge degradation and complexity problems present in image coding with VQ. The visual quality of the coded images was satisfactory in the sense that no annoying degradations were present in the coded images. However, by increasing the block size beyond 5×5, annoying edges began to appear.

3.6 Exploiting Interblock Correlation in VQ

Vector Quantization is an efficient coding technique due to its inherent capability in exploiting the high correlation between the neighboring pixels. However, the interblock correlation is totally ignored in memoryless vector quantizers. The interblock correlation is large between neighboring blocks due to the finite block size. By exploiting the statistical and structural redundancy between the neighboring blocks, the performance of standard VQ could be improved or the bit rate reduced.

In this section, we will present some vector quantizers which exploit the interblock correlation by either incorporating memory or by using several codebooks.

3.6.1 Predictive Vector Quantizers

Predictive Vector Quantizer (PVQ) is a generalization of the scalar predictive quantizer with multiple dimensions. In this scheme, a vector predictor is used to predict the current vector from previously quantized vectors. Then, an error vector $\mathbf{e}$ is formed by subtracting the predicted vector $\hat{X}$ from the actual

present vector X. Thus, the correlation between the vectors is exploited. The error vectors are then coded using a vector quantizer. By quantizing the error vectors instead of the actual vectors, the search space is reduced and hence a better performance is achieved.

Cuperman and Gersho [10] introduced a predictive vector quantizer for speech coding and showed that it can perform better than a memoryless VQ of the same rate. Hang and Woods [22] also used vector predictive quantizer in their coding system.

3.6.2 Finite State Vector Quantization (FSVQ)

Interblock correlation can also be exploited by using different codebooks for each input vector, where the codebooks chosen are based on the past input vectors. This could be represented by a finite state machine where each state S_i represents a separate quantizer with its own codebook C_i. At each state S_i, codebook C_i is used by the quantizer. The next state is selected by a mapping called the state transition function F such that, given a state S_i and a channel symbol V_i then $F(V_i, S_i)$ is the new state of the machine. The decoder can track the encoder once it knows the initial state since the state transition function depends only on the previous encoder outputs and the current state. By using different codebooks based on the previously encoded symbols, FSVQ performs better than a memoryless vector quantizer of the same rate.

FSVQ was introduced by Foster [14] followed by design algorithms by Dunham [12]. Aravind and Gersho [2] applied a FSVQ coding system to encode images by using a state transition function based on a classifier that uses the intensity and geometric correlations between neighboring blocks.

3.6.3 Exploiting Interblock Correlation in Product Structures

Interblock correlation can be exploited in product structures by exploiting the correlation between the scalar products. This approach was developed in [48] for gain/shape VQ of LPC parameters and in [4] for separating mean VQ of images. Correlation between these interblock scalar products could be exploited by using either predictive scalar quantizers, transform coding, or the basic VQ. For example, in the M/SVQ, each of these coders could take advantage of interblock sample mean correlations by either predicting the current sample mean from the previously encoded sample means or by encoding several spatially adjacent sample means simultaneously. Baker [5] reported that both scaler predictive quantizers and memoryless VQ could be used for quantizing the mean values with about 3 to 4 bits per block sample mean to achieve the required accuracy.

3.6.4 Address-Vector Quantization (A-VQ)

Address-Vector Quantization (A-VQ) is based on exploiting the interblock correlation by encoding a group of blocks together using an address codebook. Four neighboring blocks are coded first using the LBG algorithm and then their addresses are grouped together to form a new vector of four addresses. This vector is then represented by one of the codevectors in the address codebook. If we have an LBG codebook of size N, then there are N^4 possible combinations of the address codevectors. However, since the neighboring blocks are highly correlated, the possible address combinations is large but limited. Since the address information need to be represented

properly, a scheme need to be used to reduce the address codebook size. Nasrabadi and Feng [33] introduced a scheme where the address codebook is divided into two regions : active and nonactive. The active region contains the most probable address codevectors. This active region is changed every time a block is encoded. This is done by calculating a score function for each codevector in the address codebook based on block-transition probability matrices. Then, the ordering is done based on the scores bringing the more probable address codevectors into the active region of the address codebook. If the address vector can be found in the active region of the address codebook, a codeword representing the location is transmitted to the receiver. Otherwise, the addresses of the LBG-Codebook are transmitted for each individual block.

Since this technique is lossless, the coded images have the same visual quality as images encoded by the basic VQ technique but will have a significantly lower bit rate. It was found [33] that during coding of a typical image more than 70% of the blocks were coded using the address codebook. The A-VQ can be considered as a vector quantizer with finite memory because the information about the encoded neighboring blocks is used in reordering the address codebook and thus in coding the current block.

One of the disadvantage of A-VQ is that the address codebooks of the encoder and decoder have to be in the same order at any time. Another disadvantage is the computational complexity of reordering the address codebook for each vector to be encoded. This problem could be alleviated by reducing the size of the address codebook. However, the smaller the address codebook size, the less the exploitation of the interblock correlation.

A-VQ can be extended to more than two layers by further grouping the

addresses of the address codewords into a block and encoding them using an address codebook. Nasrabadi and Feng [34] used a three layer coding system where a basic VQ codebook is used in the first layer and address codebooks with codevectors consisting of addresses of the previous layer codebook are used in the other two layers. They reported good performance at bit rates reduced by a factor of two compared to the basic VQ.

3.6.5 Hierarchical Vector Quantization (HVQ)

In HVQ, quadtrees [42] are used to partition the image into several blocks of different sizes such that the content of each block is approximately stationary. Then, blocks of size 4×4 or smaller are quantized using VQ since these blocks contain edge information and need to be reproduced with high fidelity. The larger blocks of size 8×8 or larger are coded using transform coding since it is inherently superior to VQ in regions with low detail.

In this scheme, the interblock correlation is exploited by combining blocks that are correlated to a larger block size. Thus, a variable-rate coding scheme is produced which distributes the distortion uniformly over the entire image by varying the number of bits used per unit area according to the importance of the local area being coded. This scheme was introduced by Vaisey and Gersho [45, 46] and they reported very good quality coding results at rates between 0.35 and 0.6 bpp.

Chapter 4

One-Dimensional Vector Quantization

4.1 Introduction

In the last ten years, vector quantization has been found to be effective and powerful for image coding. An image is partitioned into two-dimensional (2-D) blocks and each block is then encoded as a separate entity. Two-dimensional vector quantization requires lines of buffering in a raster scanned system employing this design. The size of each buffer is equal in size to the width of the image to be encoded and the number of buffers is dependent on the block size. By partitioning the image into one-dimensional blocks each spanning a portion of a single scanning line, no buffers are required. Independent encoding of scan lines means considerable reduction in the implementation complexity. Furthermore, the encoder response is much faster in 1-D than in 2-D VQ. Due to these advantages in 1-D VQ, we would like to investigate the capability of 1-D VQ as an image coding technique. First, a comparison between 1-D and 2-D VQ is made and the problems encountered in 1-D VQ are identified and compared to those in 2-D VQ. Then, a special

block accessing scheme is introduced to distribute the distortion and improve the perceptual quality of coded images. Product structures are used to increase the codebook size and improve the performance without increasing the computational complexity. A novel product structure based on scaling (SVQ) or scaling and rotating (SRVQ) the vectors before quantization is introduced. Furthermore, the mean/shape vector quantizer (M/SVQ) is used and compared to our new structure. Both scalar and vector quantizers are used to quantize the scale factors and the mean values. Finally, the WMSE distortion measure is used in an attempt to improve the fidelity of coded edges.

4.2 One-Dimensional vs. Two-Dimensional Vector Quantization

In this section, the performance of a one-dimensional vector quantizer (1-D VQ) is compared to the performance of a two-dimensional vector quantizer (2-D VQ). Since we seek a low coding rate and we are limited by the computational complexity of VQ, we restrict our comparison to a fixed block size of dimension 16 (4×4 pixels in the 2-D case and 1×16 pixels in the 1-D case). A training set of size $32,767$ blocks taken from two 512×512 images Lena and Peppers was used to design the codebooks. Codebooks were designed using the LBG algorithm [27] using the mean squared error (MSE) as a distortion measure. In order to improve codebook design, training vectors were classified into shade and edge blocks [37]. If M and m are the larger and smaller of two adjacent pixels in a vector, then the vector is classified as an edge when $(M - m)/M$ exceeds the threshold $t = 0.4$. Thus, two code-

books were obtained separately for the shade and edge blocks. Since edges are perceptually more important to encode, the edge codebook was three times larger than the shade codebook. The final codebook was obtained by simply appending the two codebooks together. Codebooks were initialized by choosing evenly spaced vectors from the training set.

Six images were encoded using both 2-D and 1-D VQ for a codebook of size 1024 codewords. The Peak-Signal-to-Noise Ratio (PSNR) was used to compare the objective coding fidelity. The PSNR is defined as follows :

$$\text{PSNR} = 10 \log_{10} \frac{2^{(2m)}}{\frac{1}{k}\sum_{i=1}^{k}(x_i - c_i)^2} \qquad \text{[dB]} \tag{4.1}$$

where m is the number of bits per pixel, x_i and c_i are the ith pixels from the original and coded images, respectively, and k is the number of pixels in the image. For comparing the subjective quality of the coded images, the two images Lena (inside the training set) and Boat (outside the training set) were used. Figure 4.1 shows the two original images Lena and Boat.

The PSNR values for the six images coded using both 2-D and 1-D VQ are shown in Table 4.1. As seen from the table, differences in the PSNR between 2-D and 1-D is ranging from 0.05 dB (in Baboon) to 2.6 dB (in Lena) for the same compression rate (0.625 bpp). This difference in the PSNR values between the 2-D and 1-D cases is due to the fact that pixel intensities in a 2-D block are more correlated than in a 1-D block. This is due to their spatial closeness. Furthermore, correlation is exploited both horizontally and vertically in 2-D blocks. However, it is exploited only horizontally in 1-D blocks. The average block variance was measured for the two images Lena and Peppers using both 1-D and 2-D blocks. Table 4.2 shows the

(a)

(b)

Figure 4.1: (a) Original Lena image (b) Original Boat image

Image	PSNR 2-D (Rate = 0.625 bpp) Codebook Size = 1024	PSNR 1-D (Rate = 0.625 bpp) Codebook Size = 1024
Lena	32.617	30.051
Peppers	31.056	29.407
Barbara	24.545	22.899
Boat	26.745	24.701
Tiffany	27.677	25.864
Baboon	20.205	20.159

Table 4.1: PSNR values for six images coded using 2-D and 1-D schemes

Image	1-D Shade	2-D Shade	1-D Edge	2-D Edge
Lena	274.67	68.4	1633.2	972.4
Peppers	269.36	80.2	1966.9	1059

Table 4.2: Average block variance for 1-D and 2-D blocks for images Lena and Peppers

average block variance for the two images after classification into edge and shade blocks. As seen from the table, 1-D blocks have higher variance (i.e. higher activity) in both edge and shade blocks. This means that 1-D blocks require a higher coding rate than 2-D blocks for producing the same quality. Furthermore, the probability of having an edge in a 1-D block is more than that in a 2-D block . This result is clear from Table 4.3 by showing the percentage from classifying blocks into shade and edge blocks for the six images. As seen from the table, edges in the 1-D case are on the average 4% higher than in the 2-D case.

It is interesting to notice that the PSNR for Baboon image is almost

Image	2-D Shade	1-D Shade	2-D Edge	1-D Edge
Lena	93	88	7	12
Peppers	84	79	16	21
Barbara	68	60	32	40
Boat	77	75	23	25
Tiffany	93	89	7	11
Baboon	45	48	55	52
Average	77	73	23	27

Table 4.3: Percentage of edge and shade blocks in both 2-D and 1-D cases

equivalent for both 2-D and 1-D cases. This could be the result of having a large percentage of edges in both cases. Our results have shown that the PSNR is reduced with the increase in the number of edges in an image. Figures 4.2 and 4.3 show the two images Lena and Boat coded using both 2-D and 1-D VQ at the same rate (0.625 bpp) respectively.

Our results agree with what Baker reported in [3]. He found experimentally that square blocking produced the most perceptually pleasing images. This also agrees with the conclusions of Sakrison et al. [41] that one-dimensional image coding requires higher coding rate than two-dimensional. Although the 1-D technique showed inferior performance to the 2-D technique, we would like to investigate the capability of 1-D VQ as an image coding technique and compare its artifacts to the 2-D technique.

Coding images using 2-D VQ has shown some artifacts which could be summarized into the following :

- Blocked smooth regions
- Washed out textures lacking sharp definition

(a)

(b)

Figure 4.2: (a) Lena image coded using 2-D VQ (Rate = 0.625 bpp) (b) Boat image coded using 2-D VQ (Rate = 0.625 bpp)

(a)

(b)

Figure 4.3: (a) Lena image coded using 1-D VQ (Rate = 0.625 bpp) (b) Boat image coded using 1-D VQ (Rate = 0.625 bpp)

- Poorly encoded sharp edges leading to a staircased appearance

These artifacts can be seen in Figure 4.2. These symptoms arise largely because the codebook lacks sufficient collection of templates rich enough to represent textures and edges at various locations and orientations. Furthermore, the use of conventional distortion measures such as the mean-squared error (MSE) results in edge degradation. This is because MSE does not account for perceptual features when choosing the best matching codeword from the codebook. Thus, a block containing an edge could be encoded with a shade block that destroys the edge information.

To overcome these problems, researchers found it necessary to use suboptimal VQ structures to improve the objective and subjective quality of VQ. Baker and Gray [3, 5] introduced new coder structures which are based on the idea of subtracting the mean from a block and coding mean values and residual blocks separately. Their scheme yielded outstanding results for images having fine detail in smooth regions, however, sharp edges remained a source of trouble.

Although many researchers have tried to find more perceptually meaningful distortion measures, MSE seems to be the most favored distortion measure due to its simplicity and tractability. In order to overcome its weakness, Ramamurthi and Gersho [38, 39] suggested that vectors need to be classified into different classes of shades, textures, and edges of different locations and orientations before encoding. This way, edge integrity is preserved and search complexity is reduced.

Having mentioned the problems encountered in 2-D VQ, we would like to mention the problems that we have encountered using 1-D VQ. First, the weakness of the MSE as a distortion measure is more clear in the 1-D case

than in the 2-D case. This is because a 1-D block spans 16 pixels in the horizontal direction and quantizing a vector from one class with a codeword from a different class may result in wiping some of the image details. This is clearly shown in Figure 4.3 in the upper lip and the nose area of Lena image. Second, blocking effect is very clear in the 1-D case especially in smooth regions. The boundaries of the 1-D blocks are more noticeable to the eye than in the 2-D case. This is because these boundaries repeat regularly every 16 pixels which is more noticeable than in the 2-D case where they repeat every 4 pixels in the horizontal direction. This blocking effect is clear in the two images Lena and Boat shown in Figure 4.3. 1-D VQ has shown some perceptual improvements over 2-D VQ. It does a better job in coding horizontal edges than 2-D VQ (Compare the base of the boat in both Figures 4.2 and 4.3).

4.3 Improving the Perceptual Quality of 1-D VQ Using a Block Accessing Scheme

One of the most annoying problems encountered in 1-D VQ is the block boundaries (i.e. blocking) effect. To improve the perceptual quality of images encoded using 1-D VQ, a solution need to be found to remove the block boundaries effect. One way to solve this problem is to encode blocks from an image by accessing blocks from different locations in each line. By doing this, block boundaries will not be in the same locations and hence will be less noticeable by the eye.

Let us assume a complete scan line as $L(i)$ where $i = 0$ to $W - 1$; W is the width of an image. The first attempt done was by using the following

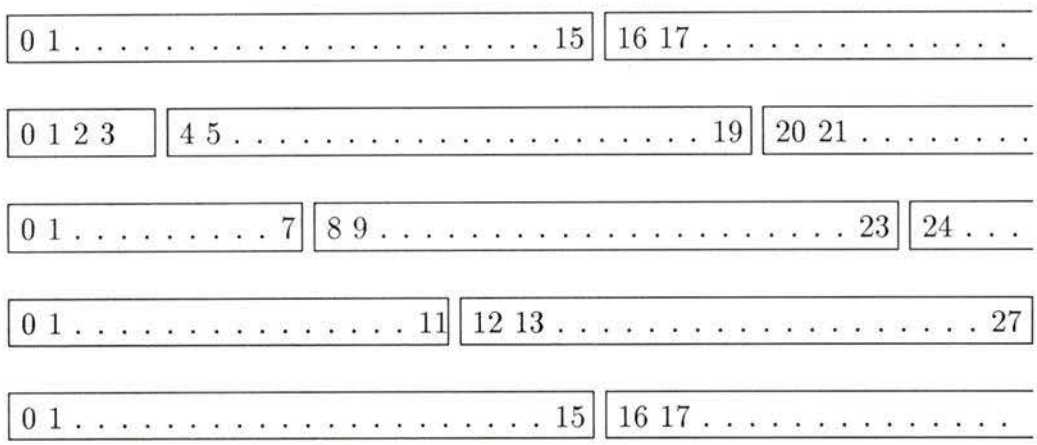


Figure 4.4: The block accessing scheme for the sequence $S1 = \{0, 4, 8, 12\}$

block accessing sequence $S1 = \{0, 4, 8, 12\}$. In encoding the first line, the first block is formed from the pixels $L(0)$ to $L(k-1)$ where k is the block dimension. The second block is accessed from $L(k)$ to $L(2k-1)$, and so on. However, in encoding the second line, the first block is formed from pixels $L(0)$ to $L(3)$ and the remaining $(k-4)$ pixels are made the same as $L(3)$. Then, the second block is accessed from $L(4)$ to $L(k+3)$ and so on. The same thing is done for lines three and four by accessing the first block from $L(0)$ to $L(7)$ in the former case and from $L(0)$ to $L(11)$ in the latter case. In case a scan line ends and the last block does not have 16 pixels, the last pixel in the line is repeated till the block size is complete. Figure 4.4 shows the block accessing scheme for the sequence $S1 = \{0, 4, 8, 12\}$.

By doing this procedure, the perceptual quality of coded images has been improved significantly and the block boundaries effect has been reduced. This is because the blocking effect is distributed more equally through the

image. Figure 4.5 shows the two images Lena and Boat coded using the block accessing sequence $S1 = \{0, 4, 8, 12\}$. As seen from the figure, the details in the nose area in Lena image have become more clear than they were before applying the block accessing scheme. Although using this block accessing sequence resulted in distributing the block boundaries and reducing the blocking effect, the effect of the regularity of the pattern upon which blocks are accessed is noticeable especially in areas where many blocks are matched to the same codeword. This effect is clear in the Boat image shown in Figure 4.5 especially in the cloud area.

To solve this problem, a Pseudo Random Generator (PRG) is used to generate a sequence of random numbers in the range $[0, k-1]$. Two primitive polynomials were used to generate random numbers in the range [0,15] by using a linear feedback shift register (LFSR). The first one, $P_1 = x^4 + x + 1$, was used to generate random numbers of period 16. The second, $P_2 = x^6 + x + 1$, was used to generate random numbers of period 64 by taking the four least significant bits. Figure 4.6 shows the two PRGs using LFSR.

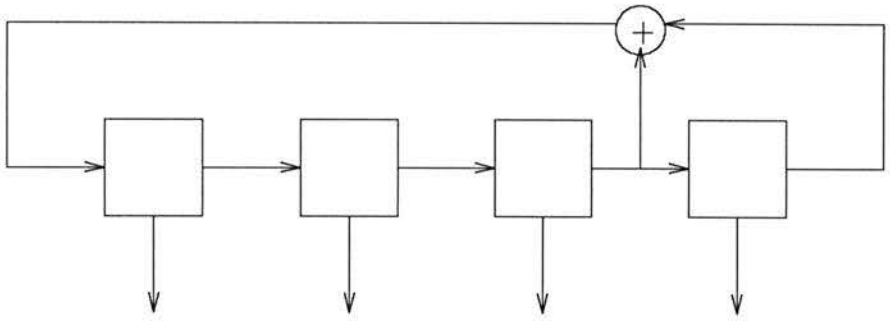
In the former case when the sequence period was 16, the regular pattern was still there but its slope has changed. Even when the sequence period was set to 64, a curved pattern was still noticeable. Figures 4.7 and 4.8 show the two images Lena and Boat coded using the two PRGs for block accessing respectively.

From this experimental work, we deduce that a block accessing sequence need to be found such that the boundaries of the blocks are equally distributed and the effect of the regularity of the pattern is removed. By using the sequence $S2 = \{0, 8, 4, 12\}$, the block boundaries were equally distributed and the effect of the regularity of the pattern was removed. Figure 4.9 shows

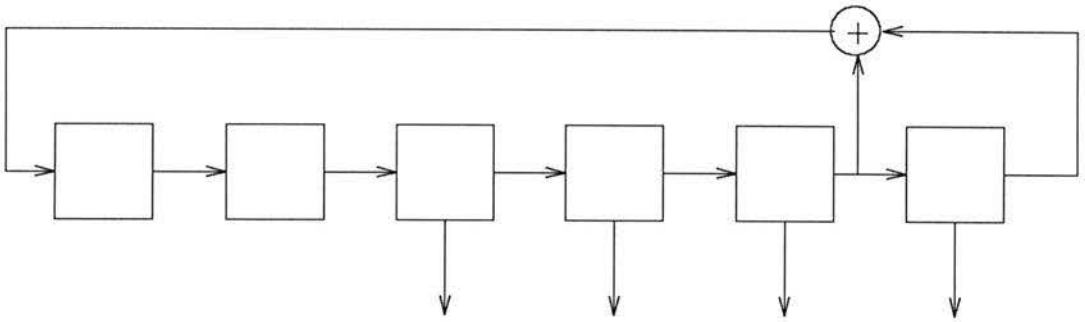
(a)

(b)

Figure 4.5: The two images (a) Lena, and (b) Boat, coded using the block accessing sequence $S1 = \{0, 4, 8, 12\}$



(a)



(b)

Figure 4.6: Pseudo Random Generators using LFSR with the two primitive polynomials (a) $P_1 = x^4 + x + 1$, and (b) $P_2 = x^6 + x + 1$

(a)

(b)

Figure 4.7: The two images (a) Lena, and (b) Boat, coded using the PRG $P_1 = x^4 + x + 1$ as a block accessing sequence

(a)

(b)

Figure 4.8: The two images (a) Lena, and (b) Boat, coded using the PRG $P_2 = x^6 + x + 1$ as a block accessing sequence

(a)

(b)

Figure 4.9: The two images (a) Lena, and (b) Boat, coded using the block accessing sequence $S2 = \{0, 8, 4, 12\}$

Image	Standard 1-D	$S1$	$P1$	$S2$
Lena	30.051	28.614	28.317	28.615
Peppers	29.407	28.230	28.020	28.217
Barbara	22.899	22.937	22.942	22.932
Boat	24.701	24.627	24.597	24.604
Tiffany	25.864	25.876	25.852	25.854
Baboon	20.159	20.191	20.205	20.193

Table 4.4: PSNR values for six images coded using different block accessing schemes

the two images Lena and Boat coded using the block accessing sequence $S2 = \{0, 8, 4, 12\}$.

By using the block accessing sequence S2, the coding rate is increased slightly. Let B be the number of bits per vector, k the vector dimension and N the number of lines in an image, then the coding rate R will be

$$R = \frac{B}{k} + \frac{3}{4}\frac{B}{N} \qquad \text{(bpp)} \tag{4.2}$$

By using the block accessing scheme, we note that the PSNR is almost the same compared to the standard 1-D case. Table 4.4 shows the PSNR values for the six images coded using the various block accessing schemes. The drop in the PSNR for Lena and Peppers is because these two images were used as a training set for codebook design. Due to the block accessing scheme, 75% (S_1 and S_2) or 95% (P_1 and P_2) of their blocks were coded as if they were not part of the training set.

4.4 Improving the Performance of 1-D VQ Using Product Structures

Although the block accessing scheme has improved the perceptual quality of coded images, edges were coded with some distortion. In order to improve edge representation in coded images, the codebook used for VQ needs to have a large number of codewords sufficient enough to represent edges of various locations and orientations. However, increasing the codebook size results in an increase in the encoding complexity since a full search of the codebook is performed for every vector to be quantized. One solution to reduce the search complexity and improve the performance is by using product structures [18].

In this section, we introduce a novel product structure based on scaling (SVQ) or scaling and rotating (SRVQ) the vectors before quantization. Furthermore, the mean/shape vector quantizer (M/SVQ) is used and compared to our new structure.

4.4.1 Scaled Vector Quantizer (SVQ)

This product vector quantizer is based on the idea of scaling vectors to a certain value before quantization. Vectors are scaled by adding a scale factor c to each component of the vector. Thus, after scaling, an input vector $X = (X_1, X_2, \cdots, X_k)$ will become $X_s = (X_1 + c, X_2 + c, \cdots, X_k + c)$. Let h be the highest and l be the lowest pixel intensity in a vector, then the scale factor c can be defined as $c = 255 - h$, called White Scaled VQ (WSVQ) or as $c = -l$, called Black Scaled VQ (BSVQ). By doing the scaling operation, the vector space and hence the search complexity is assumed to be reduced.

Let us assume that a vector consists of k pixels and each pixel is repre-

sented by m bits. Then, the volume of the vector space is given by :

$$V_1 = 2^{mk} \tag{4.3}$$

Define *Coverage* (α) to be the ratio of the codebook size (N) to the volume of the vector space (V) (i.e. $\alpha = N/V$). Then, we have

$$\alpha_1 = \frac{N}{V_1} = \frac{N}{2^{mk}} \ll 1 \tag{4.4}$$

By scaling vectors, the vector space will be confined to the surfaces of a hypercube. So, the search volume becomes

$$\begin{aligned} V_2 &= V_1 - (2^m - 1)^k \\ &= 2^{mk} - (2^m - 1)^k \end{aligned} \tag{4.5}$$

An approximate equation for V_2 that works well for our purpose is defined as follows

$$V_2 = k2^{m(k-1)} = V_1 \frac{k}{2^m} \tag{4.6}$$

Thus, Coverage will be defined as

$$\alpha_2 = \frac{N}{k2^{m(k-1)}} = \alpha_1 \frac{2^m}{k} \tag{4.7}$$

This means that by doing the scaling operation, we reduce the vector space by a factor of $2^m/k$ and increase the coverage by the same factor. For example, for a vector of size $k = 16$ pixels and $m = 8$ bits per pixel (bpp), the vector space is reduced by a factor of 16 and the coverage is increased by a factor of 16. Since m is usually set to 8 bits for grey-scale images, the improvement is dependent on the vector dimension.

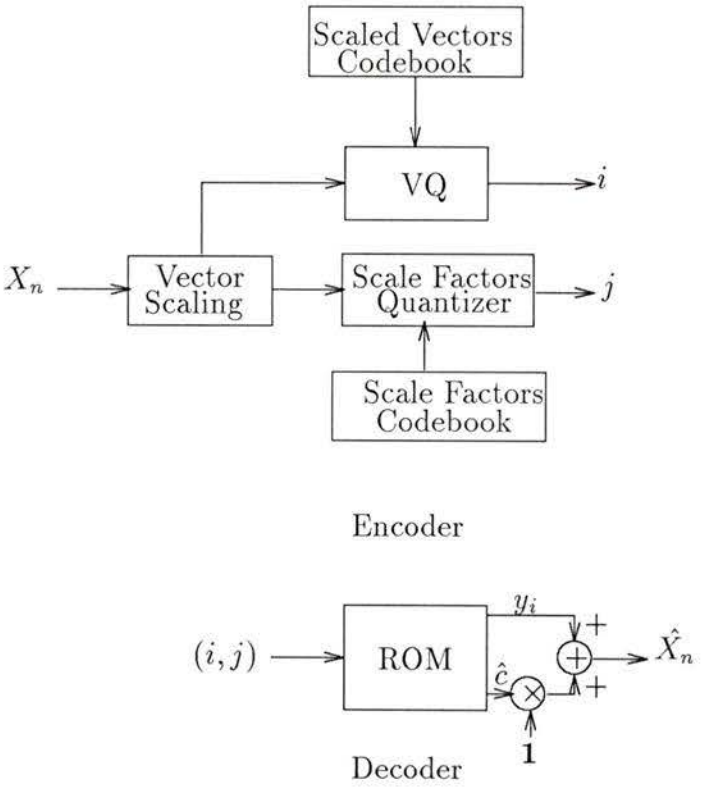


Figure 4.10: Scaled Vector Quantizer (SVQ)

Figure 4.10 shows the structure of a scaled vector quantizer (SVQ). Vectors are scaled using either WSVQ or BSVQ and then the scaled vectors and the scale factors are coded separately.

In order to test the performance of our new product structure, six images were coded using a codebook of size 1024. The codebook was designed using the LBG algorithm with random initialization. In the same way, two codebooks were designed separately for shade and edge vectors and then appended to form the final codebook. Vectors have 16 pixels each and are

Image	BASIC VQ	WSVQ	BSVQ
Lena	28.615	29.883	30.095
Peppers	28.217	29.659	29.775
Barbara	22.932	23.396	23.498
Boat	24.604	25.899	26.268
Tiffany	25.854	27.309	27.067
Baboon	20.193	20.543	20.617

Table 4.5: PSNR for six images coded using Basic VQ, WSVQ, and BSVQ

formed using the block accessing sequence $S2 = \{0, 8, 4, 12\}$. In all our work, we will use this block accessing scheme and refer to it as the basic 1-D VQ. Table 4.5 shows the PSNR for the six images coded using basic VQ, WSVQ, and BSVQ. The scale factors are given 8 bits to compare the performance of the new quantizers with the basic quantizer for the same coding complexity. WSVQ has shown a maximum improvement of almost 1.5 dB in Peppers image and a minimum improvement of almost 0.4 dB in two images. From Table 4.3, we note that the two images which have the least improvement are the two which have the largest percentage of edges (i.e. Baboon and Barbara). BSVQ performed even better than WSVQ by almost 0.1 dB. However, both WSVQ and BSVQ produce perceptually equivalent coded images. Figures 4.11 and 4.12 show the two images Lena and Boat coded using WSVQ and BSVQ respectively. By comparing the coded images using WSVQ and BSVQ to the basic VQ, we find that there is a significant improvement in smooth areas with fine details. For example, note the improvement in the cloud area in the Boat image and in the hat in Lena image. Furthermore, edges are coded with higher fidelity and their distortion is reduced.

(a)

(b)

Figure 4.11: The two images (a) Lena, and (b) Boat, coded using WSVQ

(a)

(b)

Figure 4.12: The two images (a) Lena, and (b) Boat, coded using BSVQ

Image	SRVQ
Lena	30.215
Peppers	29.619
Barbara	23.619
Boat	26.494
Tiffany	27.961
Baboon	21.711

Table 4.6: PSNR for six images coded using SRVQ

4.4.2 Scaled Rotated Vector Quantizer (SRVQ)

After scaling, the vectors in the volume space are mapped into the surfaces of a hypercube. By rotating the scaled vectors such that the pixels that determined the scaling factor lie in the same location, the vector space is reduced to just one surface of the hypercube. Thus, the volume of the vector space and Coverage will be given by :

$$V_3 = 2^{m(k-1)} = V_1 \frac{1}{2^m} \tag{4.8}$$

$$\alpha_3 = \frac{N}{2^{m(k-1)}} = \alpha_1 2^m \tag{4.9}$$

So, the vector space is reduced by a factor of 2^m and the coverage is increased by the same factor.

Experimental results were performed to test the improvement in performance using SRVQ. A codebook of size 1024 codewords was used, and vectors were scaled to the value 255 (i.e. WSVQ) and rotated such that the maximum pixel intensity always lies at the first component of the vector. Table 4.6 shows the PSNR values for the six images coded using SRVQ. As

seen from the table, SRVQ did not perform as well as expected. It showed little improvement over WSVQ for most of the images except Baboon image where it showed an improvement of more than 1 dB. This is not an encouraging improvement since we need to allocate 4 bits for the rotation value of each vector. Furthermore, the distortion is not distributed equally in SRVQ. This is because the pixel that determines the scaling factor will always have a smaller distortion compared to the surrounding pixels. After rotation, the first component of each vector will have the same value (i.e. 255) and hence will be quantized with full accuracy. Figure 4.13 shows the two images Lena and Boat coded using SRVQ.

4.4.3 Quantizing the Scale Factors

In order to reduce the compression rate, the scale factors need to be quantized to the least number of bits possible. This should be done to a level where the perceptual quality of images is not affected. Both a scalar quantizer and a vector quantizer are used to quantize the scale factors.

Scalar Quantization

In this section, a scalar quantizer is used to quantize the scale factors. A scalar quantizer was designed using the LBG algorithm for vectors of dimension 1. A training set was formed from the scale factors of the three coded images : Lena, Peppers, and Boat. The initial codebook was formed by taking evenly spaced samples in the signal space. Codebooks of sizes 32, 64, and 128 were designed. In order to determine the effect of the codebook size on the coded images, six images were coded using the three codebooks. Figure 4.14 shows the PSNR values for the six images coded using differ-

(a)

(b)

Figure 4.13: The two images (a) Lena, and (b) Boat, coded using SRVQ

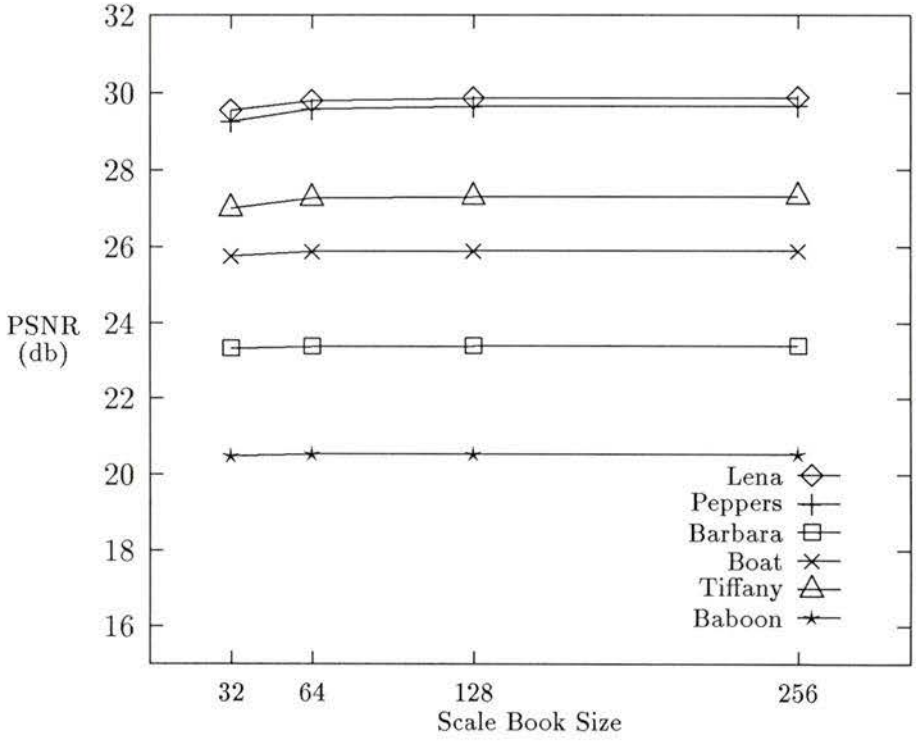


Figure 4.14: WSVQ with different scale codebook sizes

ent codebook sizes for the scale factors. As seen from the figure, 6 bits are needed for quantizing the scale factors in order to achieve the required accuracy. Table 4.7 shows the PSNR values for the six images coded with a scale codebook of size 64. As seen from the table, there is a small drop in the PSNR values (almost 0.05 dB) of the coded images using a scale factor codebook of size 64. Comparing images coded with 6 bits for scale factors to those coded with 8 bits, we find that there is no perceptual difference in the quality of coded images. Figure 4.15 shows the two images Lena and Boat coded using a scale codebook of size 64.

(a)

(b)

Figure 4.15: The two images (a) Lena, and (b) Boat, coded using WSVQ with a scale factor codebook of size 64

Image	WSVQ 6-bit SQ	WSVQ 5-bit VQ
Lena	29.802	29.792
Peppers	29.583	29.637
Barbara	23.381	23.403
Boat	25.870	25.133
Tiffany	27.265	27.326
Baboon	20.534	20.691

Table 4.7: PSNR for six images coded using WSVQ with a 6-bit scalar quantizer (SQ) and a 5-bit vector quantizer (VQ) for quantizing the scale factors

Vector Quantization

By grouping L scale factors from L spatially adjacent source vectors, a vector of scale factors is formed :

$$S = [s_1 s_2 \cdots s_L]$$

Then, by using vector quantization to quantize these scale vectors, the correlation between scale factors could be exploited and thus, their coding rate could be reduced.

In this section, an experiment is made for vector quantizing the scale factors using vectors of dimension 2 and 4. Two codebooks of size 256 were designed for scale vectors of dimension 2 and 4 respectively. Thus, the scale factors are allocated 4 bits each in the former case and 2 bits each in the latter case. The LBG algorithm was used to design the codebooks with an initial codebook formed from evenly spaced vectors from the training set. By coding the six images using the two scale codebooks, it was found that the coded images are affected by the quantization error of the scale factors. This is because a 1-D block spans 16 pixels in the horizontal direction and

an error in quantizing the scale factors is reflected on the whole block. Thus, the quantization error must be kept to a level that produces the required accuracy. A third codebook of size 1024 was designed for vectors of dimension 2. Table 4.7 shows the PSNR for the six images coded by allocating 5 bits per scale factor (i.e. scale codebook = 1024, vector dimension = 2). As seen from the table, the PSNR values for the coded images were better than those coded using a 5-bit and a 6-bit scalar quantizer. Since we have a larger codebook size in the vector quantizer, the encoding complexity of the scale factors is larger. However, quantizing the scale factors could be done in parallel to quantizing the scaled vectors.

4.4.4 Mean/Shape Vector Quantizer (M/SVQ)

The M/SVQ was introduced by Baker and Gray [3] as a product structure vector quantizer for image coding. It was used for quantizing 2-D blocks from images and showed great improvement in performance especially in smooth areas with fine details. In this section, the capability of M/SVQ in improving the performance of 1-D VQ is investigated. Furthermore, its performance is compared to the performance of the basic 1-D VQ and SVQ.

A codebook of size 1024 codewords was used to quantize the shape vectors. Table 4.8 shows the PSNR for the six images coded using M/SVQ. The mean values were coded using 8 bits to compare the performance of 1-D M/SVQ to the basic quantizer for the same coding complexity. As seen from the table, the M/SVQ showed considerable improvement over the basic VQ for the same coding complexity. It showed a maximum improvement of 2.9 dB in Boat image and a minimum improvement of 1.6 dB in baboon image. It also outperformed SVQ by almost 1 dB. By comparing the images

Image	M/SVQ
Lena	31.160
Peppers	31.069
Barbara	24.572
Boat	27.524
Tiffany	28.356
Baboon	21.782

Table 4.8: PSNR for six images coded using M/SVQ

coded using M/SVQ to those coded using the basic VQ, we find that the perceptual quality of coded images is considerably improved. Smooth areas are enhanced and edges are coded with greater fidelity. Figure 4.16 shows the two images Lena and Boat coded using M/SVQ.

An experiment was made by coding images using 1-D M/SVQ without using the block accessing scheme. This was done to test whether the block boundaries effect could be removed by the improvement gained from using M/SVQ. By doing this, the same improvement in terms of PSNR was gained, however, it was found that the effect of the block boundaries is reduced but not removed. From this experiment, we deduce that it is essential to use the block accessing scheme in 1-D VQ. In order to show the difference between images coded with and without the block accessing scheme using M/SVQ, histogram equalization is used for the coded images. Figure 4.17 shows the two images Lena and Boat coded with and without the block accessing scheme after applying the histogram equalization operation.

(a)

(b)

Figure 4.16: The two images (a) Lena, and (b) Boat, coded using M/SVQ

(a) (b)

(c) (d)

Figure 4.17: Images coded with 1-D M/SVQ after histogram equalization (a) Lena with the block accessing scheme, (b) Lena without the block accessing scheme, (c) Boat with the block accessing scheme, and (d) Boat without the block accessing scheme

4.4.5 Quantizing the Mean Values

In this section, both a scalar quantizer and a vector quantizer are used to quantize the mean values. The same experiments that were performed for quantizing the scale factors are made for quantizing the mean values.

Scalar Quantization

A scalar quantizer was designed using the LBG algorithm for vectors of dimension 1. A training set was formed from the mean values of the three coded images : Lena, Peppers, and Boat. The initial codebook was formed by taking evenly spaced samples in the signal space. Three codebooks of sizes 32, 64, and 128 were designed for the mean values. In order to determine the effect of the mean codebook size on the coded images, six images were coded using the three codebooks. Figure 4.18 shows the PSNR values for the six images coded using the three mean codebooks. As seen from the figure, 6 bits are needed for quantizing the mean values in order to achieve the required accuracy. This result is the same as what was found for quantizing the scale factors in SVQ. Table 4.9 shows the PSNR values for the six images coded with a mean codebook of size 64. As seen from the table, there is a small drop in the PSNR (almost 0.05 dB) of the coded images using a mean codebook of size 64. Furthermore, there is no perceptual difference in the quality of images coded with 6 bits for mean quantization compared to those coded with 8 bits. Figure 4.19 shows the two images Lena and Boat coded using a mean codebook of size 64.

By using product structures (i.e. SVQ or M/SVQ), the performance of coded images is improved for the same coding complexity. Figure 4.20 shows the improvement in SNR for the six images using both WSVQ and

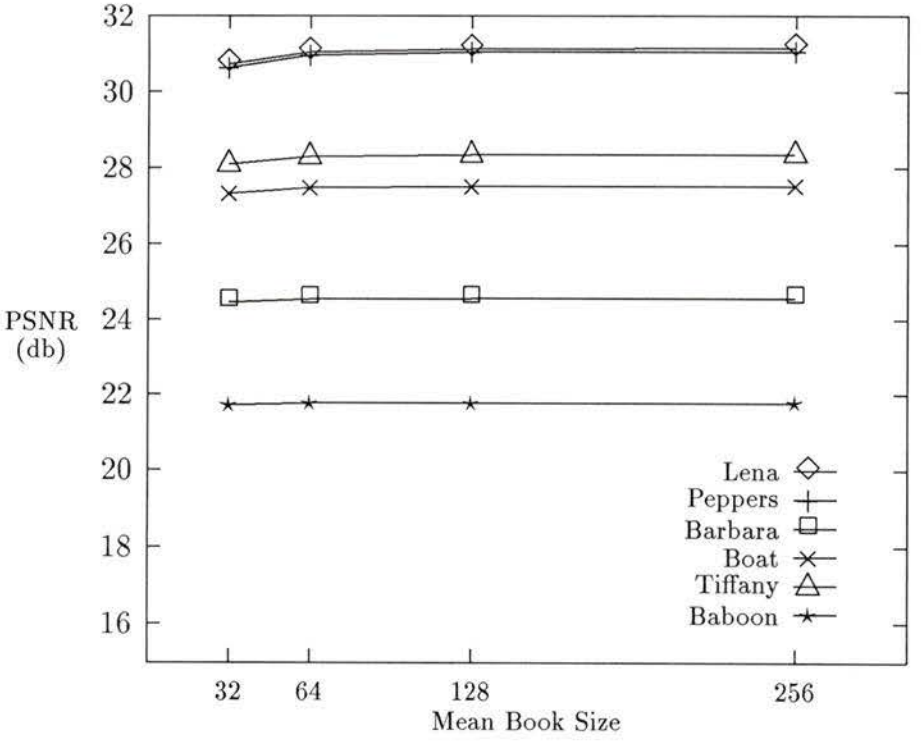


Figure 4.18: M/SVQ with different mean codebook sizes

Image	M/SVQ 6-bit SQ	M/SVQ 5-bit VQ
Lena	31.049	30.968
Peppers	30.959	30.729
Barbara	24.548	24.438
Boat	27.478	27.369
Tiffany	28.293	28.158
Baboon	21.769	21.745

Table 4.9: PSNR for six images coded using M/SVQ with a 6-bit scalar quantizer (SQ) and a 5-bit vector quantizer (VQ) for quantizing the mean values

(a)

(b)

Figure 4.19: The two images (a) Lena, and (b) Boat, coded using M/SVQ with a mean codebook of size 64

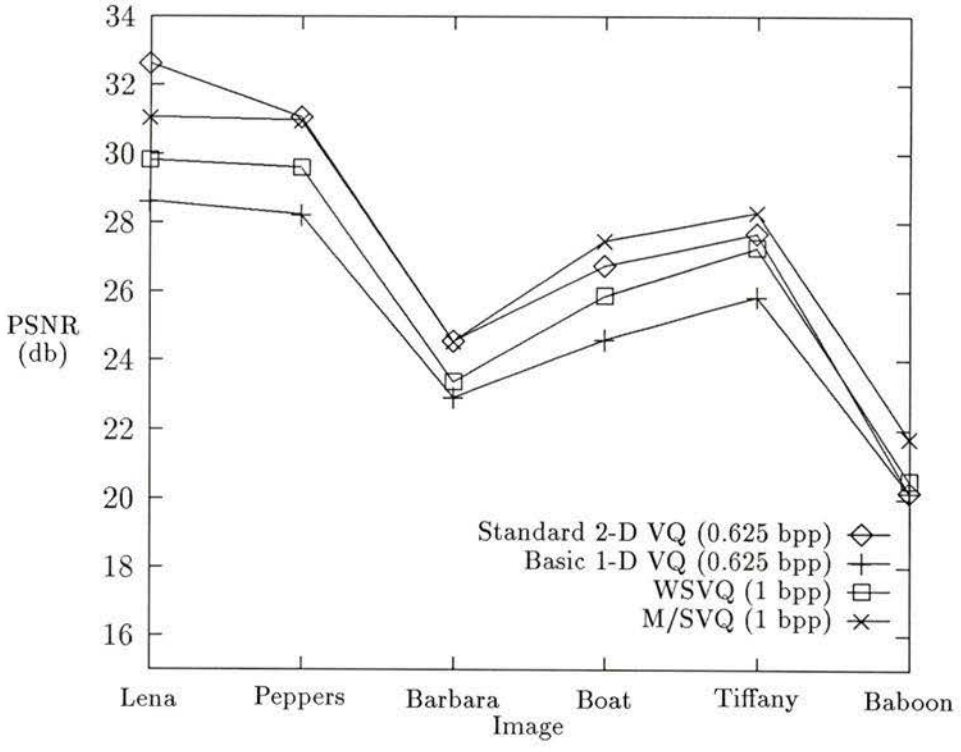


Figure 4.20: Improvement in SNR for the six images coded using WSVQ and M/SVQ compared to the basic 1-D VQ and the standard 2-D VQ

M/SVQ with a 6-bit scalar quantizer. As seen from the figure, by increasing the bit rate from 0.625 bpp to 1 bpp, the performance has been improved considerably.

Vector Quantization

By quantizing the mean values using vector quantization, the correlation between the mean values can be exploited and hence the coding rate could be reduced. In this section, experiments are made for vector quantizing the

mean values using vectors of dimension 2 and 4. Three codebooks of sizes 256, 512, and 1024 were designed for mean vectors of dimension 2 and two codebooks of sizes 1024 and 2048 were designed for mean vectors of dimension 4. By coding the images using these mean codebooks, it was found that the best results are obtained from using a codebook of size 1024 and vectors of dimension 2. Table 4.9 shows the PSNR values for the six images coded using a mean codebook of size 1024 for vectors of dimension 2 (i.e. mean coding rate = 5 bit/sample). Comparing these results to those obtained using a 5-bit and a 6-bit scalar quantizer, we find that a 5-bit vector quantizer performs better than a 5-bit scalar quantizer and worse than a 6-bit scalar quantizer by almost 0.1 dB. From these results, we deduce that either a 6-bit scalar quantizer or a 5-bit vector quantizer could be used for quantizing the mean values without affecting the perceptual quality of coded images.

4.4.6 BSVQ vs. M/SVQ

Since BSVQ involves subtracting the scaling coefficient from the input block, we think that BSVQ is related to M/SVQ which involves subtracting the mean from the input vector. BSVQ and M/SVQ are related by the following relation $c = m - \tau$, where c represents the coefficient of scaling, m represents the mean, and τ represents a positive constant. For example, let an input vector $X = (2, 10, 3)$, then the mean $m = 5$ and the shape vector $S = X - m = (-3, 5, -2)$. If the input vector is scaled using BSVQ, then $c = 2$ and the scaled vector $S = X - c = (0, 8, 1)$. We note that the only difference between the shape vector and the scaled vector is a positive constant $\tau = 3$. Thus, both BSVQ and M/SVQ should obtain a close performance. To

understand why this was not the case, let us analyze the problem. Under the squared error criterion, the optimum encoder must find the pair (c, S) which minimizes the distortion

$$\begin{aligned} d(X, S + c\mathbf{1}) &= [X - S - c\mathbf{1}]^T[X - S - c\mathbf{1}] \\ &= [X^T X - 2X^T S + S^T S] - 2cX^T\mathbf{1} \\ &\quad +c^2\mathbf{1}^T\mathbf{1} + 2cS^T\mathbf{1} \\ &= |X - S|^2 - 2kc < X > +kc^2 + 2kc < S > \\ &= |X - S|^2 + k[(< X > -\tau) - c]^2 \\ &\quad +2kc[< S > -\tau] - k[< X > -\tau]^2 \end{aligned} \tag{4.10}$$

For the case of M/SVQ, $\tau = 0$ and $< S >= 0$. Thus, the optimum encoding rule can be achieved by minimizing the first two terms in equation (4.10) which results into minimizing the two equations (3.15) and (3.17) independently. However, in BSVQ the third term must be considered too to achieve the optimum encoding rule. Since the basic structure of a scaled vector quantizer shown in Figure 4.10 did not account for the third term, this could be the reason why BSVQ achieved a lower performance than M/SVQ. A better performance may be obtained by minimizing the following two equations

$$c = \min_{c \in C_c}{}^{-1} [(< X > -\tau) - c]^2 \tag{4.11}$$

$$S = \min_{S \in C_S}{}^{-1} (|X - S|^2) + [< S > -\tau]) \tag{4.12}$$

4.5 WMSE as a Distortion Measure

As we have seen from the previous section, using product structures reduces the amount of blocking visible in smooth regions and improves the encod-

ing of edges. However, sharp edges are still poorly encoded and need more improvement. In this section, an experiment is made using WMSE as a distortion measure in an attempt to improve the fidelity of coded edges.

The main idea in this experiment is to give more weight to pixels forming an edge to emphasize edge fidelity over shade fidelity. The WMSE distortion measure used is defined as follows :

$$d(x,y) = \sum_{i=1}^{k} w_i(x_i - y_i)^2 \tag{4.13}$$

In order to determine the weight for each pixel in an input vector x, a simple edge detection algorithm (§4.2) is performed to determine the pixels forming an edge. Then, the pixels forming an edge are given the weight w and the rest of the pixels are given a weight of 1.

Three different weightings were used in our experiment $w = 5, 10,$ and 50. By using this distortion measure, the resulting codebook emphasizes edges at the expense of smooth regions. Note that smooth regions will be encoded as in the MSE except those near edges. Three codebooks were designed for the three different weightings using M/SVQ. The two images Lena and Boat were encoded by encoding mean values at full accuracy. From these experiments, it was found that more error is concentrated in pixels near edges. As the weighting factor increases, edges are coded more accurately and smooth regions near edges are coded worse. Furthermore, there was no convincing perceptual improvement in fidelity of coded edges for small weightings $w = 5$ and 10.

Our results agree with what Baker [5] reported from his experiments with various distortion measures. He found that the codebook size has a much greater influence on edge fidelity than the choice of distortion measure. One

way to increase the codebook size and increase the fidelity of coded edges is to use classified vector quantization (CVQ) [38, 39]. By classifying vectors before encoding into different classes according to a certain criterion (i.e. no. of edges and/or location of an edge), the MSE can be used effectively and edges will be encoded more accurately.

Chapter 5

Conclusions

5.1 Summary of results

In this section, we state briefly the main results and conclusions of this work. We begin with the comparison of one-dimensional and two-dimensional vector quantizers. 2-D VQ showed a better performance than 1-D VQ. The PSNR was on the average 1.6 dB higher in images coded using 2-D than in those coded using 1-D VQ. This result is expected since pixels are more correlated in 2-D blocks than in 1-D blocks. This is due to their spatial closeness. Furthermore, correlation is exploited both horizontally and vertically in 2-D blocks. However, it is exploited only horizontally in 1-D blocks.

It was found that the PSNR is reduced with the increase in the number of edges in an image. The weakness of the MSE as a distortion measure was more apparent in the 1-D case than in the 2-D case. This is because 1-D blocks span 16 pixels in the horizontal direction and quantizing a vector from one class with a codeword from a different class may result in wiping some of the image details. Furthermore, the block boundaries (blocking) effect was very clear in the 1-D case especially in smooth regions. 1-D VQ does a better

job in encoding horizontal edges than 2-D VQ.

In order to improve the perceptual quality of coded images using 1-D VQ, a special block accessing scheme was used. After experimenting with several block accessing sequences, it was found that the sequence $S2 = \{0, 8, 4, 12\}$ distributed the distortion in a manner that is least objectionable perceptually.

Although the block accessing scheme has improved the perceptual quality of coded images, edges were coded with some distortion. In order to improve the performance of 1-D VQ, product structures were used. A novel product structure based on scaling (SVQ) or scaling and rotating (SRVQ) the vectors before quantization was introduced. By using SVQ, the PSNR was improved by an average of 1 dB for the same coding complexity. There was a significant improvement in smooth areas with fine details. Furthermore, edges were coded with higher fidelity and their distortion was reduced. SRVQ did not perform as well as expected. It showed little improvement over SVQ. This is not an encouraging improvement since 4 bits need to be allocated for the rotation factor of each vector. Both scalar and vector quantizers were used to quantize the scale factors. It was found that either a 6-bit scalar quantizer or a 5-bit vector quantizer (codebook size = 1024, vector dimension = 2) are needed to quantize the scale factors without affecting the perceptual quality of coded images.

The capability of M/SVQ in improving the performance of 1-D VQ was investigated. Furthermore, its performance was compared to the performance of the basic 1-D VQ and SVQ. M/SVQ showed considerable improvement over the basic VQ for the same coding complexity. It showed an average improvement of 2 dB in PSNR. It also outperformed SVQ by almost 1 dB. For quantizing the mean values, the same results were obtained as those

obtained for quantizing the scale factors.

An experiment was made by coding images using 1-D M/SVQ without using the block accessing scheme. This was done to test whether the block boundaries effect could be removed by the improvement gained from using M/SVQ. By doing this, the same improvement in terms of PSNR was gained, however, it was found that the effect of the block boundaries is reduced but not removed.

Finally, an experiment was made using WMSE as a distortion measure in an attempt to improve the fidelity of coded edges. Three different weightings were used $w = 5$, 10, and 50 by giving pixels that form an edge a weight of w and the rest of pixels a weight of 1. From these experiments, it was found that more error is concentrated in pixels near edges. As the weighting factor increases, edges are coded more accurately and smooth regions near edges are coded worse. Furthermore, there was no convincing perceptual improvement in fidelity of coded edges for small weightings $w = 5$ and 10. Thus, we conclude that the codebook size has a much greater influence on edge fidelity than the choice of distortion measure.

5.2 Future Work

In our work we showed that, by using the block accessing scheme and product structures, 1-D VQ produces good quality coded images at a low coding rate (1 bpp). In this section, we suggest two techniques that could improve the performance of 1-D VQ. These techniques are listed below and are worth investigation :

- 1-D CVQ : One of the main problems in vector quantization of images is edge degradation. By classifying vectors before quantization into several classes according to a certain criterion (i.e. No. of edges in a vector and/or edge location), edge integrity is preserved. Furthermore, the search complexity is reduced since only a subset of the codebook is used to find the best matching vector.

- 1-D PVQ : In our work, we have used a memoryless vector quantizer. We have exploited only the horizontal pixel correlation. Due to the fact that pixels are highly correlated vertically, one can exploit this correlation by using a line-by-line predictor. Codebook is designed for the error vector between the actual vector and the predicted one. Due to the high correlation of pixels vertically, the variance of the error vector will be small and hence, the vector space will be reduced. Thus, a better performance can be achieved for the same coding complexity and for the same rate.

Bibliography

[1] H. Abut, R. M. Gray, and G. Rebolledo, "Vector Quantization of Speech and Speech-Like Waveforms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 423-435, June 1982.

[2] A. Aravind and A. Gersho, "Low-Rate Image Coding With Finite State Vector Quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Mar. 1986, pp. 137-140.

[3] R. L. Baker and R. M. Gray, "Image Compression Using Non-Adaptive Spatial Vector Quantization," in *Proc. 16th Asilomar Conf. on Circuits, Systems and Computers*, vol. I, Oct. 82, pp. 55-61.

[4] R. L. Baker and R. M. Gray, "Differential Vector Quantization of Achronomic Imagery," in *Proc. of the International Picture Coding Symposium*, March 1983, pp. 105-106.

[5] R. L. Baker, *Vector Quantization of Digital Images*, Ph.D. dissertation, Stanford University, June 1984.

[6] M. F. Barnsley and A. D. Sloan, "A Better Way to Compress Images," *Byte*, pp. 215-232, Jan. 1988.

[7] P. J. Burt and E. H. Adelson, "The Laplacian Pyramid As A Compact Image Code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532-540, Apr. 1983.

[8] A. Buzo, A. H. Gray,Jr., R. M. Gray and J. D. Markel, "Speech Coding Based Upon Vector Quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, no. 5, pp. 562-574, Oct. 1980.

[9] R. J. Clarke, *Transform Coding of Images*, Academic Press, 1985.

[10] V. Cuperman and A. Gersho, "Adabtive differential Vector Coding of Speech," in *Proc. Conf. Rec., GLOBECOM'82,* Dec. 1982, pp. 1092-1096.

[11] V. Cuperman, *Efficient Waveform Coding of Speech Using Vector Quantization*, Ph.D. Dissertation, Univ. of California, Santa Barbara, March 1983.

[12] M. O. Dunham and R. M. Gray, "An Algorithm for the Design of Labeled-Transition Finite State Vector Quantization," *IEEE Trans. Commun.*, vol. COM-33, pp. 83-89, Jan. 1985.

[13] W. H. Equitz, "A New Vector Quantization Clustering Algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 10, pp. 1568-1575, Oct. 1989.

[14] J. Foster, R. M. Gray and M. Dunham, "Finite-State Vector Quantization for Waveform Coding," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 348-355, May 1985.

[15] A. Gersho, "On the Structure of Vector Quantizers," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 157-166, March 1982.

[16] R. M. Gray, J. C. Kieffs, and Y. Linde, "Locally Optimal Block Quantizer Design," *Inform. Control*, vol. 45, pp. 178-198, May 1980.

[17] R. M. Gray and E. D. Karnin, " Multiple Local Optima in Vector Quantizers," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 2, pp. 256-261, March 1982.

[18] R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4-29, April 1984.

[19] A. Habibi and G. S. Robinson, "A Survey of Digital Picture Coding," *IEEE Computer*, vol. 7, pp. 22-34, May 1974.

[20] A. Habibi, "Hybrid Coding of Pictorial Data," *IEEE Trans. Commun.*, vol. COM-22, no. 5, pp. 614-623, May 1974.

[21] C. F. Hall and E. L. Hall, "A Nonlinear Model for the Spatial Characteristics of the Human Visual System." *IEEE Trans. System, Man, and Cybernetics*, vol. SMC-7, pp. 161-170, March 1977.

[22] H. M. Hang and J. W. Woods, "Predictive Vector Quantization of Images," *IEEE Trans. Commun.*, vol. COM-33, pp. 1208-1219, Nov. 1985.

[23] A. K. Jain, "Image Data Compression: A Review," in *Proc. of the IEEE*, vol. 69, pp. 349-389, March 1981.

[24] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, 1984.

[25] M. Kunt, A. Ikonomopoulos and M. Kocher, "Second-Generation Image-Coding Techniques," in *Proc. of the IEEE*, vol. 73, No. 4, pp. 549-574, April 1985.

[26] J. O. Limb, "Distortion Criteria of the Human Viewer," *IEEE Trans. on System, Man and Cybernetics*, vol. SMC-9, pp. 778-793, Dec. 1979.

[27] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[28] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129-136, March 1982.

[29] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proc. of the Fifth Berkeley Symp. on Math. Stat. and Prob.*, pp. 281-296, 1967.

[30] J. L. Mannos and D. J. Sakrison, "The Effects of a Visual Fidelity Criterion on the Encoding of Images," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 525-536, July 1974.

[31] T. Murakami, T. Asai, and E. Yamazaki, "Vector Quantization of Video Signals," *Electronic Letters*, vol. 18, pp. 1005-1006, Nov. 11, 1982.

[32] N. M. Nasrabadi and R. A. King, "Image Coding Using Vector Quantization: A Review," *IEEE Trans. Commun.*, vol. 36, pp. 957-971, Aug. 1988.

[33] N. M. Nasrabadi and Y. Feng, "Image Compression Using Address-Vector Quantization," *IEEE Trans. Commun.*, vol. 38, no. 12, pp. 2166-2173, Dec. 1990.

[34] N. M. Nasrabadi and Y. Feng, "A Multilayer Address Vector Quantization Technique," *IEEE Trans. Circuits and Systems*, vol. 37, no. 7, pp. 912-921, July 1990.

[35] A. N. Netravali and J. O. Limb, "Picture Coding : A Review," in *Proc. of the IEEE*, vol. 68, March 1980, pp. 366-405.

[36] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons Inc. , 1978.

[37] B. Ramamurthi and A. Gersho, "Image Coding Using Vector Quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, April 1982, pp. 428-431.

[38] B. Ramamurthi, *Vector Quantization of I images Based on a Composite Source Model,* Ph.D. dissertation, California Santa Barbara ,1985.

[39] B. Ramamurthi and A. Gersho, "Classified Vector Quantization of Images," *IEEE Trans. Commun.,* vol. COM-34, pp. 1105-1115, Nov. 1986.

[40] M. J. Sabin and R. M. Gray, "Product Code Vector Quantizers for Waveform and Voice Coding," *IEEE Trans. Acoust., Speech, Signal Processing*, June 1984.

[41] D. J. Sakrison and V. R. Algazi, "Comparison of Line-by-Line and Two-Dimensional Encoding of Random Images," *IEEE Trans. Commun.*, vol. IT-17, pp. 386-389, July 1971.

[42] H. Samet, "The Quad-Tree and Related Hierarchical Data Structures," *ACM Comput. Surv.*, vol. 16, no. 2, pp. 188-260, June 1984.

[43] C. E. Shannon, "A Mathematical Theory of Communication," *BSTJ*, vol. 27, pp. 379-423, July 1948 and pp. 623-656, Oct. 1948.

[44] T. G. Stockham, "Image Processing in the context of a visual Model," in *Proc. of the IEEE*, vol. 60, July 1972, pp. 828-842.

[45] D. J. Vaisey and A. Gersho, "Variable Block-Size Image Coding," in *Proc. ICASSP'87* , Dallas, April 1987, pp. 1051-1054.

[46] D. J. Vaisey and A. Gersho, "Variable Rate Image Coding Using Quad-Trees and Vector Quantization," *EUSIPCO 1988.*

[47] R. Wilson, H. E. Knutsson, and G. K. Granlund, "Anisotropic Nonstationary Image Estimation and its Applications : Part II- Predictive Image Coding," *IEEE Trans. Commun.*, vol. COM-31, pp. 398-406, Mar. 1983.

[48] D. Wong, B. H. Juang and A. H. Gray, Jr., "An 800 bit/s Vector Quantization LPC Vocoder," *IEEE Trans. Acous., Speech, Signal Processing*, vol. ASSP-30, pp. 770-779, Oct. 1982.

VITA

Surname:	El-Maleh	Given Names:	Aiman
Place of Birth:	Jeddah, Saudi Arabia	Date of Birth:	4 April 1967

Educational Institutions Attended:

University of Victoria	1989 to 1991
King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia	1984 to 1989

Degrees Awarded:

B.Sc. (Hon.) King Fahd University of Petroleum and Minerals 1989

Honours and Awards:

NSERC Scholarship	1991-1993
University of Victoria Fellowship	1990-1991
King Fahd University of Petroleum and Minerals Fellowship	1984-1989

Publications:

1. Aiman H. El-Maleh, Fayez El-Guibaly and V.K. Bhargava, "Image Compression Using One-Dimensional Vector Quantization," (under preparation).

2. Aiman H. El-Maleh, Fayez El-Guibaly and V.K. Bhargava, "Improving the Performance of One-Dimensional Vector Quantization Using Product Structures," *Proceedings of the 1991 First Cyprus International Conference on Computer Applications to Engineering Systems* Nicosia, Cyprus, July 1991, pp. 40-45.

3. Aiman H. El-Maleh, Fayez El-Guibaly and V.K. Bhargava, "A Comparison of One- And Two-Dimensional Vector Quantization of Images,"

Proceedings of the 1991 IEEE Pacific Rim Conference on Communications, Comp uters and Signal Processing, Victoria, Canada, vol. 2, May 1991, pp. 490-493.

4. Aiman H. El-Maleh and Sadiq M. Sait, "A State Machine Synthesizer With Weinberger Arrays," *Proceedings of the 1991 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, Canada, vol. 2, May 1991, 753-756.

5. Sadiq M. Sait and Aiman H. El-Maleh, "State Machine Synthesis with Weinberger Arrays," to appear in *International Journal of Electronics*, 1991.

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis: Image Compression Using One-Dimensional Vector Quantization

Author: [redacted]
(Signature)

Aiman H. El-Maleh
(Name in Block Letters)

July 22, 1991
(Date)