1996

Projection based edge recovery in low bit rate vector quantizers

Ajai Narayan
Iowa State University

Projection based edge recovery in low bit rate vector quantizers

by

Ajai Narayan

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department: Electrical and Computer Engineering
Major: Electrical Engineering (Communications and Signal Processing)

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1996

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION 1

1.1 Introduction to Image Data Compression 1

1.2 A Brief Review of Related Work 5

1.3 Objective and Scope of the Dissertation 6

CHAPTER 2. INFORMATION THEORY AND THE RATE DISTORTION FUNCTION 7

2.1 Overview of Information Theory 8

2.1.1 Discrete Memoryless Information Source 8

2.1.2 Information Content of a Message 9

2.1.3 Entropy of an Information Source 10

2.1.4 Noiseless Source Coding Theorem 11

2.1.5 Overview of Rate Distortion Theory 12

CHAPTER 3. THEORY OF CONVEX PROJECTIONS 17

3.1 Definitions and the Mathematical Theory 17

3.2 Application of Convex Projection Theory to Signal Recovery 22

3.2.1 The Gerchberg-Papoulis Algorithm 22

CHAPTER 4. INTRODUCTION TO VECTOR QUANTIZATION 26

4.1 Theoretical Definition of Vector Quantization (VQ) 29

4.2 Codebook of a Vector Quantizer 30
4.3 Implementational Trade-offs in a Vector Quantizer 31
4.4 Classified Vector Quantization (CVQ) 33
4.4.1 The CVQ Classifier 34
CHAPTER 5. IMAGE RESTORATION 39
5.1 Motivation for Image Restoration 39
5.2 Convex Projections as Applied to Image Recovery 42
5.3 Review of Related Work 43
5.4 Vector Quantization and Convex Projections 44
5.4.1 Multiple Codebooks (General Case) 44
5.4.2 Sets and Projections Associated with the Image Data 45
5.4.3 The Algorithm 49
5.5 Implementational Aspects 51
5.5.1 Decoding with Spatial Controlled Smoothing (Particular Case) 55
5.5.2 Sets and Projections Associated with the Image Data 55
5.5.3 The Algorithm 58
5.5.4 Determination of "$p_{\max}$" in the Recovery Algorithm 59
5.5.5 The Lowpass Filter 60
CHAPTER 6. CONVERGENCE ANALYSIS OF THE ALGORITHM 63
CHAPTER 7. EXPERIMENTAL RESULTS 79
7.1 Experimental Results and Performance 79
7.2 Subjective and Objective Distortions 81
7.3 Robustness of the Algorithm 87

7.3.1 Case 1: Robustness to Codebook Variations 87

7.3.2 Case 2: Robustness to the Lowpass Filter Parameters 88

BIBLIOGRAPHY

LIST OF FIGURES

Figure 1.1	A schematic of a typical data compression operation	3
Figure 2.1	The rate-distortion function	15
Figure 3.1	Geometrical interpretation of the signal recovery operation in a Hilbert space	22
Figure 4.1	A diagrammatic representation of a scalar quantizer	28
Figure 4.2	A diagrammatic representation of a vector quantizer	28
Figure 4.3	A diagrammatic representation of a composite source	35
Figure 5.1	Conceptual illustration of the projection approach	43
Figure 5.2	Calculation of the average value of an image block	51
Figure 5.3	Shade space with codevectors and corresponding partitions . . .	52
Figure 5.4	Edge space with a low resolution and high resolution codebooks and corresponding partitions superimposed	53
Figure 5.5	A schematic representation of the projection algorithm	54
Figure 5.6	A 2-D lowpass filter	62
Figure 6.1	A diagrammatic representation of the prediction operation for optimal recovery operation	69

Figure 6.2 Mean square error v/s number of iterations under near optimal conditions 70

Figure 6.3 MOS v/s number of iterations under near optimal conditions . . 71

Figure 6.4 Mean square error v/s number of iterations under suboptimal conditions 72

Figure 6.5 MOS v/s number of iterations under suboptimal conditions . . 73

Figure 6.6 Edge vector space with both the low and the high resolution codebooks superimposed 74

Figure 6.7 A diagrammatic representation of edge block recovery 75

Figure 7.1 A protion of "lenna" image (8 bpp) 81

Figure 7.2 A portion of "peppers" image (8 bpp) 82

Figure 7.3 "Staircase effect" due to vestigial edge encoding of the edge vectors (lenna - 0.254 bpp) 83

Figure 7.4 Multiple codebook decoding using the projection algorithm (lenna - 0.254 bpp) 84

Figure 7.5 "Staircase effect" due to vestigial edge encoding of the edge vectors (peppers - 0.259 bpp) 85

Figure 7.6 Multiple codebook decoding using the projection algorithm (peppers - 0.259 bpp) 86

Figure 7.7 A small portion of the "peppers" image expanded (before the application of the recovery algorithm). 87

Figure 7.8 A small portion of the "peppers" image expanded (after the application of the recovery algorithm). 88

LIST OF TABLES

Table 7.1	Statistics of a few typical photographic images	79
Table 7.2	Number of shade and edge pixels in the test images	80
Table 7.3	Objective performance of the algorithm on the test images . .	84
Table 7.4	Mean opinion score from the ACR test on the test images . .	86

ACKNOWLEDGEMENTS

I would like to sincerely thank Dr. John F. Doherty, for making my study very enjoyable and creatively satisfying. His help throughout my research, from the initial stage of an ad-hoc proposal to its present form is much appreciated. I am grateful for his confidence in me in allowing a free hand in my study and the innumerable discussions we had throughout this study. I am really thankful for his constant insistence that I submit my work to peer review.

I would like to thank Dr. Satish S. Udpa, without whose understanding, timely help and constant support this study would not have been possible. I am very grateful for his initiative in getting me a teaching assistantship which helped with my research tremendously.

I would also like to thank my committee members Dr. Alexandre Megretski, Dr. Peter J. Sherman and Dr. James A. Wilson for their time and advice throughout this study. In particular, the discussions I had with Dr. Sherman helped me enormously in organizing my presentation of the subject.

CHAPTER 1. INTRODUCTION

1.1 Introduction to Image Data Compression

This is an era of information revolution. Information exchange through services like telephone, television, facsimile transmission and computer networks has multiplied enormously over the years. This may be attributed to a large extent to processing information using digital hardware. Digital hardware provides the dual advantages of high speed data processing coupled with extremely compact circuitry, which are primary requirements for most of the information services that have been visualized for the future. Digital representation of analog information has several other advantages as well. Digital signals are less sensitive to transmission noise than analog signals. As a result, they are able to make better use of interference-limited and noise-limited communication media [1]. Digital information can be acquired, manipulated, stored, transmitted and interpreted no different from a sequence of numbers. All the concepts of controlling numerical operations from remote computers through networks can be easily applied to information processing. Digital processing of signals has facilitated implementation of many sophisticated algorithms thereby making them more tractable.

Digital hardware technology has fueled an enormous growth in the information industry. It has resulted in an explosion in the rate of information exchange. The

stupendous rate of information exchange has resulted in generation and transportation of enormous amounts of data. This has necessitated compression as a means to reduce the amount of data required to convey any given information. In fact, data compression has become one of the areas that has received a lot of attention in recent years. From information services point of view, two types of signals are of major concern viz., speech and image signals. Both these signals share common traits like high correlation within the signals, established methods for conversion from analog signals to digital data, tolerance to reasonable levels of subjective distortion in general, and existence of tractable measures for objective distortion. Hence, these signals are excellent subjects for data compression. Image signals are quite pervasive in the field of information processing.

Digital images require enormous amount of data for their representation. For example, a still picture on a video monitor is typically represented by an array of 512×512 samples. So, the number of samples representing the picture is $262,144$ and if each of these samples is represented by an 8-bit number in a computer (for gray scale images), the total number of bits required to represent the picture is more than 2 Megabits. As a second example, typical data rates for reasonable spatial and temporal quality of images in broadcast television is about 100 megabits per second (without compression) [2]. Thus, storage and/or transmission of image data, whether from still or motion pictures, require enormous channel and/or storage capacity. This aspect of image data makes its compression all the more desirable and imperative.

Image data compression is primarily concerned with reducing the amount of data required to represent images. The major advantage of image compression is that the burden on the channel, storage memory and the equipment that transmits/receives

the data is diminished due to reduced amounts of data. Thus, methods designed for image data compression can be viewed from either a transmission or a storage viewpoint. Transmission applications will generally be more sensitive to the processing requirements of an algorithm because information transmission and reception, typically, are real time applications. Archival applications will be more sensitive to the achievable compression as they will, generally, have no time constraints. A block diagram of a typical image data compression operation is shown in Figure 1.1.

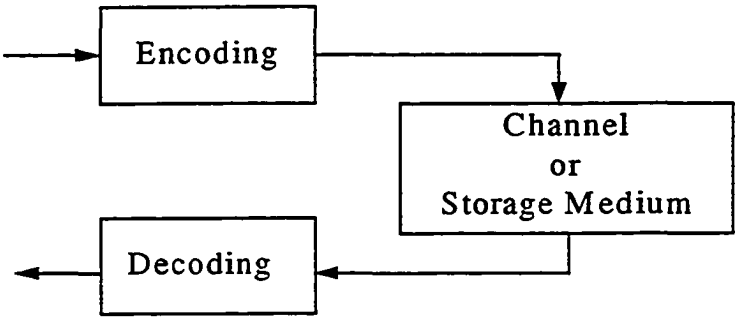


Figure 1.1: A schematic of a typical data compression operation

Image data compression also has disadvantages. Barring very few methods, image compression results in a distorted reconstruction of an image. Image compression introduces additional cost in processing depending on the method used for compression. This cost increases proportionally to the amount of compression required and the effectiveness of a compression method. The thrust of research in the area of photographic image compression has been to generate reconstructions that have better subjective quality with simultaneous reduction in bit rates. This has given rise to several new methods of signal coding, e.g., subband coding, wavelet transforms, fractal signal compression, which are collectively known as the second generation coding

algorithms. All these methods are computationally intensive and require dedicated hardware in order to provide good reconstructions at reasonable speeds.

In the present discussion, it is assumed that the original images are already in digitized form and any distortion the image has suffered due to the process of conversion digitization is considered negligible. Thus, all references to distortion measures is with respect to this digitized image. Image data compression methods can be broadly classified into two classes, viz., reversible and irreversible (lossy) compression. In the former class are compression methods that do not result in any loss of information due to compression. The reconstructions that result from these methods are exact versions of the corresponding original images. But the achievable compression in this class is generally low and critically dependent on the correlation properties of the images in question [2]. The latter class of compression methods are characterized by higher and selectable compression factors irrespective of the correlation properties of the images. The reconstructions from these methods are distorted versions of the corresponding original images. Inherently, there is a trade-off between the compression ratio and the loss of information.

Recently, a significant amount of research effort has been directed towards using postprocessing techniques for restoring decoded images from lossy compression algorithms. Application of image restoration techniques is expected to result in subjectively pleasing reconstructions with tolerable variations in objective distortions. However, the trend has been to consider image restoration as an independent aspect of the decoding process. The restoration technique is considered to have no direct bearing on either the bit rate or the compression algorithm. The effort of this research is to incorporate a viable restoration technique as an integral component of a

compression algorithm. It is expected that such an effort would have a direct bearing on the rate-distortion trade off of the compression algorithm.

1.2 A Brief Review of Related Work

The theoretical idea of restoring a signal by imposing a number of constraints on its coded form has been researched by Youla, Stark and Stevenson[5, 6, 7] etc.. This technique is generally referred to as the theory of Convex Projections (CP), a concept derived from Linear Spaces. Application of the theory to practical situations has again been investigated by Katsaggelos et.al, Zakhor, Mersereau and others [8, 9, 10, 11, 12]. Katsaggelos [8] applied the theory to deblur images using Weiner Filters. Zakhor and Yang et. al studied the application of the theory to block DCT compressed images to eliminate blocking artifacts at the block boundaries in DCT images while Su and Mersereau [12] tried to eliminate both blocking and ringing effects. Sezan and Stark [6] applied the theory to reconstruct tomographic images.

However, there is very little published work on the application of the theory to VQ compressed images. Vector Quantization (VQ) is a very viable alternative to the DCT among block coders with a potential to provide better reconstructions at identical bit rates and execution costs. The main advantages of the VQ approach to compression are:

1. Unlike block DCT, VQ does not suffer from problems of ringing due to truncation/quantization of frequency components.

2. VQ (with prediction) inherently exploits the interblock correlations. Thus the blocking artefacts that are very perceptible in low bit rate DCT compressed

images is not apparent in low bit rate VQ compressed images.

However, vector quantization suffers from one basic problem. At lower bit rates the edges in an image are badly degraded. The degradation of the edges is inversely proportional to the bit rate and hence to the size of the "codebook". This is because bulk of the representations in a codebook encode the edges and very few representations encode the shade regions. A decrease in the transmission bit rate implies a reduction in the number of representations in the codebook which implies a reduction in the edge representations.

1.3 Objective and Scope of the Dissertation

The objective and scope of the dissertation is to investigate the effects of incorporating an image restoration/recovery algorithm on the fidelity of low bit rate VQ compressed images. In particular, the objective is to study if the use of a recovery algorithm in the reduction of bit rates and its effects on the decoded images. The recovery technique is incorporated as an integral part of VQ compression/decompression algorithm. The scope of the study is twofold:

1. To develop numerical implementations of the convex projection (CP) algorithm as applied to Vector Quantization.

2. To study the effects of the approach on subjective fidelity of the decoded images and achievable compression ratios.

CHAPTER 2. INFORMATION THEORY AND THE RATE DISTORTION FUNCTION

Information theory provides the mathematical basis for some of the fundamental problems in communication. A thorough understanding of this mathematical foundation and its application to communication, is a prerequisite before any attempt is made to solve the problems themselves. Data compression is a particular problem in communication that deals with the reduction of data to be transmitted/stored in order to transfer some information. A brief discussion of the mathematical aspects that governs data compression, is therefore, called for. This chapter presents a brief overview of those aspects of information theory that are relevant to the problem of lossy and lossless compression of images.

The object of any data compression algorithm is to reduce the amount of data required to represent some given information. This representation may (or may not) result in a finite amount of information loss. Image compression is no exception to this rule. Since images are also a form of information, all the results in information theory are directly applicable to them. It is therefore logical to express and measure the information contained in an image quantitatively. The topic of information theory was first discussed by Hartley in 1928 [14]. It was later formalized from an engineering perspective by Claude E. Shannon in his now famous papers [15, 16].

2.1 Overview of Information Theory

Historically, the basic concepts of information theory such as entropy, mutual information, and redundancy were first introduced by Shannon [16]. Shannon also stated the fundamental problem in communications as- the reproduction at one point, either exactly or approximately, a message that is selected at another point, from a set of possible messages. Each of these messages is characterized by its probability of occurrence/selection. Shannon defined quantities to measure the information content in individual messages and the average information content of an exhaustive set of messages emanating from any information source.

2.1.1 Discrete Memoryless Information Source

Any information source can be modelled in terms of a set of symbols, their probabilities of occurrences and their interdependencies. A discrete memoryless source (DMS) is the simplest source model that can be constructed to describe any given set of messages. In practice, there are very few real information sources that can be characterized under DMS. However, this class of sources is a very convenient mathematical abstraction that acts as a standard for comparing the performances of different sources of practical interest.

A source is considered to be a DMS when it emits a discrete set of symbols whose probabilities of occurrences are independent of symbols that were emitted previously by the same source. Mathematically, it is said that such sources emit symbols that are independent of each other and identically distributed (i.i.d). In this discussion, an information source is assumed to emit one symbol at every unit of time. A group of such symbols constitutes a message.

A DMS is mathematically defined as follows. Consider a DMS with source symbols $\{x_1, x_2, x_3 ... x_N\}$ with corresponding probabilities $\{P(x_1), P(x_2), P(x_3) ... P(x_N)\}$. Besides the symbols, this source is completely defined by the probabilities of all the symbols. In particular, the knowledge of only $N-1$ independent probabilities is required since the sum of probabilities must equal 1. The complexity of such a DMS is said to be $N-1$. The probability of occurrence of a message can be computed as a product of the probabilities of occurrence of all the symbols that constitute the message.

2.1.2 Information Content of a Message

The information content or the self information of a single message (from a set of possible messages) emanating from a DMS is given by [17]

$$I(x) = \log_2 \frac{1}{P(x)} \tag{2.1}$$

where $\underline{x} = [x_1, x_2, \ldots x_N]$ is the message in question and $P(\underline{x})$ is its probability of occurrence. The use of logarithmic function for information measurement was initially suggested by Hartley. The logarithmic function satisfies all the intuitive properties that are required of a function that defines the information content of a message:

1. The information content of a message is inversely proportional to the probability of its occurrence.

2. The information content in a message with a unity probability of occurrence is zero.

3. The total information contained in two independent messages is the sum of the information content of the individual messages.

The self information expressed by Equation (2.1) can be defined with respect to any logarithmic base. The choice of base dictates the units in which the self information is expressed. The use of 2 as a base is common and the corresponding units are bits.

2.1.3 Entropy of an Information Source

Consider a discrete memoryless information source (DMS) S which can emanate a set of N possible symbols $\{x_1, x_2, ...x_N\}$ with corresponding probabilities $\{P(x_1), P(x_2), ...P(x_N)\}$. The entropy or the average information content of the source is given by [17]

$$H(S) = -\sum_{n=1}^{N} P(x_n) \log_2 P(x_n) \quad \text{bits/symbol} \tag{2.2}$$

where $0\, log_2 0 = 0$. S is the source in question, $H(S)$ is the entropy or the average information content of the source, $P(x_n)$ is the probability of occurrence of the n^{th} symbol of the N possible symbols. The entropy or the average information content of a source is measured in bits per symbol.

The entropy as defined in Equation (2.2) depends only on the shape of the probability density function (pdf) of the source. It can only take on values ranging from 0 to $log_2 N$. The maximum value is achieved when all the symbols from the source are equiprobable. The minimum value is achieved when one of the messages has a unity probability of occurrence while the rest have zero probability. Thus, the flatter the pdf of the source, the larger its value of source entropy.

2.1.4 Noiseless Source Coding Theorem

The object of noiseless source coding is to transform a given sequence of messages into another sequence called a code sequence such that it is shorter than the original data sequence and it is possible to recover the original sequence without any distortion.

Consider an information source S that emits N possible symbols $\{x_1, x_2, ...x_N\}$ with corresponding probabilities of occurrence $\{P(x_1), P(x_2), ...P(x_N)\}$. Let these symbols be converted to some code sequences with respective lengths $L_1, L_2, ...$and L_N. The average length $\overline{L}$ of this code is given by

$$\overline{L} = \sum_{n=1}^{N} P(x_n)\, L_n \tag{2.3}$$

According to Shannon's theory, in order that the code sequences be uniquely decodable, (i.e., for the code to be noiseless), this average code length must be greater than or equal to the entropy of the source i.e., $H(S)$

$$\overline{L} \geq H(S) \tag{2.4}$$

The equality in Equation (3.4) can be achieved if L_n is equal to $-log_2 P(x)$ for all n. This important relation between entropy of a source and the average code length is referred to as the *noiseless source coding theorem.*

Thus, from the noiseless source coding theorem, it can be seen that the smallest possible average code length is achieved when the length of the codeword for each of the possible message sequence equals the self-information content of that message. Based on this justification, the self-information of a message is also called the *ideal code length* of the message.

2.1.5 Overview of Rate Distortion Theory

Rate distortion theory is one of the many astounding contributions of C.E. Shannon to information theory. In the noiseless source coding theorem, Shannon proved that the minimum rate at which a discrete memoryless source can be coded such that the information can accurately be reconstructed (at the destination), is the source entropy $H(S)$. In rate-distortion theory, he considered the problem of source coding with a distortion. In this case, there is no longer a need for accurate reconstruction of source information. Rather, a finite amount of distortion between the reproduced and the original (source) information is expected and quantified [18]. This quantity is also known as the fidelity. The problem is then to minimize the level of distortion for a given data rate, or, equivalently to reduce the data rate for a prespecified distortion level. Speech and images lend themselves very well to this type of source coding. This is because both speech and images are tolerant to reasonable levels of subjective distortion.

It has been shown that, for a broad class of sources and distortion measures, a rate distortion function $R(D)$ can be obtained. This function divides the rate-distortion plane into a set of points for which good codes for compression exist and a set of points for which no code exists at the desired distortion level. The function has the following properties:

(i) For any given level of distortion D_0, it is possible to find a coding scheme with a rate arbitrarily close to $R(D_0)$ and an average distortion arbitrarily close to D_0.

(ii) It is impossible to find a code that achieves reproduction with fidelity D_0 (or better) at a rate below $R(D_0)$.

In the following brief discussion of rate distortion theory, we confine our discussion to discrete memoryless sources (DMS). The same ideas hold even for sources with memory, where conditional probabilities replace the independent probabilities in appropriate expressions.

Consider a DMS with source symbols $\{a_1, a_2, \ldots, a_N\}$ with a corresponding probability distribution $\{p_1, p_2, \ldots, p_N\}$. Sequences of length K of the above N symbols from this source will be encoded into sequences of length K from the reproducing alphabet $\{b_1, b_2, \ldots, b_M\}$. The single-symbol distortion measure is defined as $d(a_n, b_m)$, which assigns a real non-negative value to every (a_n, b_m). Let $W_i = a_{1i}, a_{2i}, \ldots, a_{Ki}$ be a source sequence whose reproduced version in a given encoding scheme is $V_j = b_{1j}, b_{2j}, \ldots, b_{Kj}$. The average distortion per symbol of W_i is then given by

$$d(W_i, V_j) = \frac{1}{K} \sum_{k=1}^{K} d(a_{ik}, b_{jk}) \tag{2.5}$$

The average distortion of the source is then given by the statistical average of $d(W_i, V_j)$ over all the source sequences. In general, if the information is composed of P messages $\{V_1, V_2, \ldots, V_P\}$, and a W_i is assigned to every message V_j such that the distortion $d(W_i, V_j)$ is minimized, then the average distortion over all possible message sequences is given by

$$d = \sum_{all\ W_i} P(W_i)\, d(W_i, V_j) \tag{2.6}$$

where V_j is a function of W_i, $P(W_i)$ is the probability of occurrence of the symbol sequence W_i and $d(W_i, V_j)$ is the distortion value between the source sequence and the corresponding reproduction sequence to which the source sequence is mapped on

to. The rate of the code is given in terms of bits/symbol by

$$R = \frac{1}{K} \log_2 P \tag{2.7}$$

where R is the rate of the code in question, K is the number of symbols within a sequence and P is the number of reproduction sequences in the code.

A code is said to be D-admissible if the average distortion associated with the code is less than or equal to D. The smallest value of R which has a distortion less than D is denoted by $R(D)$ and is known as the rate-distortion function.

The minimum distortion is achieved if each source symbol a_n within the sequence W_i is assigned a reproduction symbol b_{m0} such that $d(a_n, b_{m0}) \leq d(a_n, b_m)$ for all m. This minimum distortion, $\overline{d}_{\min}$, is then given by

$$\begin{aligned} \overline{d}_{\min} &= \sum_{n=1}^{N} P(a_n)\, d(a_n, b_{m0}) \\ n &= 1, \ldots, N \quad \text{all source symbols that are encoded} \end{aligned} \tag{2.8}$$

i.e., for $D = \overline{d}_{\min}$, each source symbol a_n is mapped on to a certain b_{m0}. This is achievable when each of the symbols within the sequence is coded individually as in the case of scalar quantization. When the message sequence is quantized as a block, Equation 2.8 does not hold for every a_n. But the high rate of data compression that can be achieved using block quantization more than makes up for this trade-off.

If the mapping between a_n and b_m is one-to-one, the code will have a rate equal to the source entropy. But, in general this mapping is seldom one-to-one, and always some sort of many-to-one. Thus, more than one a_n is mapped onto every b_m resulting in a data compression with certain non-zero distortion. The function $R(D)$ is a nonincreasing function of D. As D increases, the function $R(D)$ decreases until at some value D, say , it becomes zero and remains at that value for all higher values

of D. To determine the value of , i.e., the point at which $R = 0$, or, from Equation 2.7, $P = 1$. Thus, this expression tells us to find a single codeword that exhibits a minimum average distortion with respect to all the source symbols.

From the above discussion it can be concluded that the rate-distortion function $R(D)$ is a monotonically nonincreasing function of D and the range $\left[\overline{d}_{\min}, \overline{d}_{\max}\right]$ is the range of interest. The typical behavior of the rate-distortion function is depicted in Figure 2.1. Further, $\overline{d}_{\min} = 0$ can be achieved if for each source symbol a_n there exists a symbol b_{m0} such that $d(a_n, b_{m0}) = 0$. In case of discrete sources with a finite

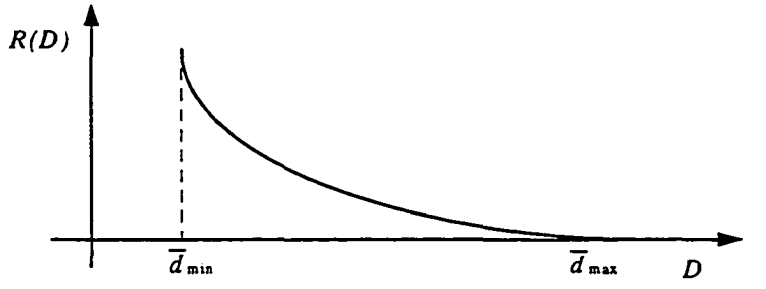


Figure 2.1: The rate-distortion function

alphabet set, zero distortion can be achieved with a finite data rate. However, if the source is continuous, = 0 can only be achieved with an infinite data rate.

From Equation 2.7 it can be deduced that for a constant P, larger values of K results in smaller data rates. However, to construct a D-admissible code having a large value for N and a small value for M, the source that is being coded should have a highly localized probability distribution function (PDF). The probabilities of occurrence of a few symbols have to be extremely high (i.e., approaching 1) while the bulk of the symbols should have vanishingly small probabilities of occurrence.

The idea of Vector Quantization (VQ) is a direct consequence of this deduction from the rate-distortion theory. The rate-distortion theory suggests that a Discrete Memoryless Source (DMS) with a PDF localized over a small range of values, can be coded better using a vector coding approach rather than scalar coding approaches. Vector coding approach results in a bit rate that is very much closer to the rate distortion bound than a scalar coding approach can ever achieve. In vector quantization of images, the image is first partitioned into blocks of dimensions $P \times P$. Each block is then considered as a vector with dimensions P^2. The entire block is then quantized as a single unit rather than quantizing individual pixels. The assumption that is implicitly made here is that a very few realizations (of all the possible ones) have a very high probability of occurrence (approaching 1), while the bulk of the realizations have an extremely low probability of occurrence (approaching 0).

From the above discussion, it is obvious that a white noise with a uniform distribution is a very poor candidate for VQ. Application of VQ is not justifiable if it is previously known that the signal in question is indeed a uniformly distributed white noise. However, if the distribution of the noise is gaussian, laplacian or any other type of distribution that exhibits a high degree of localization in its PDF, significant gains in the bit rates can be achieved for a given SNR.

Several studies have been performed on the statistical properties of both speech and images. It has been observed that the innovations values, which result as the difference between the actual and predicted versions of the sample, predominantly exhibit either Gaussian or a Laplacian distribution. Thus, the use of vector quantization (with codebooks having similar statistical properties) to code the innovations values is justified.

CHAPTER 3. THEORY OF CONVEX PROJECTIONS

This chapter presents the theory of convex projections (CP) as applied to the problem of signal recovery. It also presents an image recovery algorithm due to Gerchberg and Papoulis ([6]) based on CP known commonly as the GP algorithm. The algorithm itself is presented here because the method developed in this dissertation is an extension of the GP algorithm. This chapter is intended to highlight the differences between the two approaches.

3.1 Definitions and the Mathematical Theory

The mathematical theory of convex projections (CP) is developed in the context of a Hilbert space. Hence, a brief introduction to Hilbert spaces and convex sets is presented before discussing CP. The definitions have been extracted primarily from two sources [3, 4]:

Definition 1 *A vector space* $\mathfrak{V}$ *is a set of objects* $x, y, z, \cdots$ *called vectors together with two binary operations* $+, *$ *(denoting summation and product operations respec-*

tively) and scalars $\lambda, \mu, \cdots$, *which satisfy the following properties:*

$$\begin{aligned}
&x + y = y + x && \textit{Commutative law}\\
&x + (y + z) = (x + y) + z && \textit{Associative law}\\
&x + 0 = x && \textit{Existence of an additive identity}\\
&x + (-x) = 0 && \textit{Existence of an additive inverse}\\
&\lambda(x + y) = \lambda(x + y) && \textit{Distributive law}\\
&(\lambda + \mu)x = \lambda x + \mu x && \textit{Distributive law}\\
&(\lambda\mu)x = \lambda(\mu x) = \mu(\lambda x) && \textit{Multipicative associative law}\\
&1x = x && \textit{Existence of a multiplicative identity}
\end{aligned} \tag{3.1}$$

Definition 2 *A pre Hilbert space is a real or complex vector space* $\mathfrak{P}$ *with an inner product* $\langle x, y\rangle$ *satisfying the following properties:*

$$\begin{aligned}
&\langle x, y\rangle = \langle y, x\rangle^*\\
&\langle (x + y), z\rangle = \langle x, z\rangle + \langle y, z\rangle\\
&\langle \lambda x, y\rangle = \lambda \langle x, y\rangle\\
&\langle x, x\rangle > 0 \qquad\qquad \textit{when } x \neq 0
\end{aligned} \tag{3.2}$$

An example of such a space in the complex domain can be defined as follows: Let $\mathfrak{P}$ be a space defined in $\mathfrak{C}^2$ (i.e, the individual vectors in the space have the form $\underline{x} = (\xi_1, \xi_2)$ where ξ_1 and ξ_2 are complex numbers). The inner product of any two vectors $\underline{x} = (\xi_1, \xi_2)$,$\underline{y} = (\eta_1, \eta_2)$, in this space is defined as shown below:

$$\langle x, y\rangle = (\xi_1\overline{\eta}_1 + \xi_2\overline{\eta}_2) \tag{3.3}$$

An inner product space so defined satisfies all conditions of a pre Hilbert space.

Definition 3 *A sequence* x_n *is said to be convergent if there exists a vector* x *such that* $x_n \to x$ *as* $n \to \infty$.

Definition 4 *A sequence x_n is said to be Cauchy in case $d(x_m, x_n) \to 0$ as $m, n \to \infty$, where d is the distance operator.*

Definition 5 *A metric space is said to be complete if every Cauchy sequence in the space is convergent. A pre Hilbert space that is complete is called a Hilbert space.*

Definition 6 *A subset $\mathfrak{C}$ of a Hilbert space H is said to be convex if for every pair of elements f_1, f_2 in $\mathfrak{C}$, the element $\mu f_1 + (1-\mu) f_2$ is also in $\mathfrak{C}$, for all μ, $0 \leq \mu \leq 1$.*

Definition 7 *Let $\mathfrak{V}$ be a vector space over real numbers. A non empty subset W of $\mathfrak{V}$ is said to be a linear manifold in $\mathfrak{V}$ if:*

1. *For any two vectors $\underline{u}$ and $\underline{v}$ in W the vector $u + v$ is also in W;*
2. *For any vector $\underline{w}$ in W and any scalar α, the vector $\alpha\underline{w}$ is also in W.*

Definition 8 *Let $\mathfrak{V}$ be a vector space over real numbers. A non empty subset W of $\mathfrak{V}$ is said to be a closed subset if:*

$$\forall s_n \in W,\ \exists\ \underline{x} \in \mathfrak{V} \quad | \ s_n \to \underline{x} \ \ as \ \ n \to \infty \Longrightarrow \underline{x} \in W \tag{3.4}$$

where s_n denotes a sequence in the subset W

The idea of using CP to solve problems in signal recovery stems from a view that considers such problems to be purely geometric in character. In order to apply CP the following initial formulation must be admitted [5]:

- The original f is a vector known apriori to belong to a linear subspace C_b of a parent Hilbert space H. The available data to the observer is its projection

in another known linear subspace $C_a \in H$. The problem statement is then as follows: Find necessary and sufficient conditions under which f is uniquely determined by $P_a f$ (its projection in C_a).

Further, the conditions for the solution of the aforementioned problem are as follows:

1. f is uniquely determined by $P_a f$ iff C_b and the orthogonal complement of C_a ($C_a^{\perp}$)have only the zero vector in common.

2. The restoration problem is said to be *well posed* iff the angle between C_b and $C_a^{\perp}$ is greater than zero. If not, the problem is said to be *ill posed.*

In general, signal restoration problems, particularly in the discrete domain are always ill posed due to sampling and quantization. Hence, it is impossible to achieve perfect recovery in real world situations but the attempt is to recover as much information as is practical by using better approximations to the convex sets.

Theorem 9 *Let f, g, h be elements of a Hilbert space H, with a zero vector and an inner product. Let C be any linear manifold in H and $C^{\perp}$, its orthogonal complement. Then, according to the projection theorem, every $f \in H$, possesses a unique decomposition*

$$\begin{aligned} f &= g + h \qquad \text{where} \quad g \in C \text{ and } h \in C^{\perp} \\ \text{iff} \quad H &= C \oplus C^{\perp} \end{aligned} \tag{3.5}$$

The vectors g and h are mutually orthogonal i.e., $\langle g, h \rangle = 0$. Two linear operators P and Q defined by the rules $g = Pf$ and $h = Qf$ are the associated orthogonal projection operators projecting onto C and $C^{\perp}$. The projection operators P and Q also satisfy the following conditions $P^2 = P = (1 - Q)$ and $Q^2 = Q = (1 - P)$ [5].

The problem of reconstruction can then be stated as follows: If an element f (in the Hilbert space) belongs to a known subset C_b, but only its projection $g = P_a f$ where $g \in C_a$ is available, can f be reconstructed from the available data i.e., g? This question was addressed by D.C. Youla in his paper on generalized image restoration in 1978. In the remainder of this subsection we provide a brief view into his formulation of the problem.

Let P_a, Q_a, P_b,and Q_b denote the projection operators projecting onto the subsets $C_a, C_a^{\perp}, C_b$ and $C_b^{\perp}$ respectively. Then $f \in C_b$ implies $f = P_b f$ and so

$$\begin{aligned} g = \; & P_a f = P_a(P_b f) = (1 - Q_a)P_b f \\ = \; & (P_b f - Q_a P_b f) = (1 - Q_a P_b) f \end{aligned} \tag{3.6}$$

If further $A = (1 - Q_a P_b)$, then the vector f can uniquely be determined by g iff a bounded inverse for the transformation $T = A^{-1}$ exists. The problem of reconstructing f from g (i.e., $P_a f$) given that $f \in P_b$ is said to be 'completely posed' iff the operator $A = (1 - Q_a P_b)$ has a bounded inverse T, and incompletely posed otherwise. Generally, all practical image recovery problems are incompletely posed. In the case of an incompletely posed image recovery an iterative reconstruction equation exists and is given by

$$f_{k+1} = g + Q_a P_b f_k \qquad k = 1 \rightarrow \infty, \; f_1 = g \tag{3.7}$$

Figure 3.1 shows the geometrical significance of the iterations depicted in the above expression

The three subsets $C_a, C_a^{\perp}$ and C_b are indicated as three straight lines of infinite extent passing through the origin. It is required to restore the vector f = OA from its projection in the CLM C_a. Although it is impossible to synthesize f directly from $P_a f = OB = g$ by direct movement back along BA to C_b it is nevertheless possible

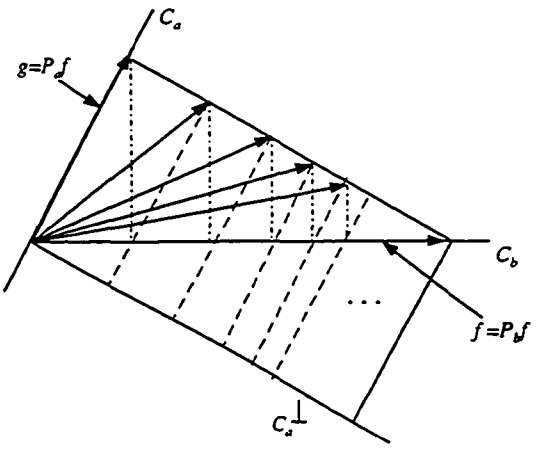


Figure 3.1: Geometrical interpretation of the signal recovery operation in a Hilbert space

to reach intermediate points D, F, H etc., which tend to the limit at A. Projecting OB on to C_b results in OC. The restoration is accomplished by projecting OC on to $C_a^{\perp}$ and adding the result OC′ to OB resulting in the second approximation OD. The process is repeated with successive approximations resulting in vectors OF, OH etc. limiting to the vector OA.

3.2 Application of Convex Projection Theory to Signal Recovery

3.2.1 The Gerchberg-Papoulis Algorithm

The general idea of signal restoration using convex projections has been applied in several signal recovery problems. Spectral extrapolation and tomographic imaging are two oft encountered applications that use convex projections for signal restoration. The restoration techniques and algorithms employed are highly dependent on

individual applications. In this section, we describe a 1-D restoration technique known as the G-P algorithm. This dissertation deals specifically with variations of this algorithm applied to 2-D signals.

The algorithm is defined for time limited functions $f(t)$ over a region ξ, which are assumed to be members of an associated Hilbert space H. These functions can potentially be projected on to several convex sets. However, it is not necessary to use all the defined sets during restoration. Some of the convex sets defined are:

1. C_1: The set of all f's that vanish outside a prescribed region $\delta \subset \xi$. C_1 is a closed linear manifold of H. Given any arbitrary function $f \in H$, its projection onto C_1 is realized by

$$P_1 f = \begin{cases} f(t) & t \in \delta \\ 0 & t \notin \delta \end{cases} \tag{3.8}$$

2. C_2: The set of all f's in H whose Fourier transforms assume a prescribed value G over a closed region $\mathfrak{L}$ in the Fourier plane.

$$P_2 f = \begin{cases} G(\Omega) & \Omega \in \mathfrak{L} \\ F(\Omega) & \Omega \notin \mathfrak{L} \end{cases} \tag{3.9}$$

where F is the Fourier transform of f (i.e., $f \longleftrightarrow F$).

3. C_3: The set of all non negative functions in H that satisfy the energy constraint

$$\int |f(t)|^2 \, dt \leq E \tag{3.10}$$

4. C_4: The subset of all functions in H that are non negative. This set is a closed convex set with '0' vector as the vertex. The projection onto this set is

$$P_4 f = \begin{cases} f_1 & f_1 \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

5. C_5: The set of all f's in H whose amplitudes must lie in a prescribed closed interval $[a, b]$. The projection onto C_5 is realized by the following rule:

$$P_5 f = \begin{cases} a & f(t) < a \\ f(t) & a \leq f(t) \leq b \\ b & f(t) > b \end{cases} \tag{3.12}$$

Any 1-D signal source is described in the time domain as a function $f(t), 0 \leq t \leq \infty$ which represents the amplitude of the signal over all time. Two masks are defined in the algorithm, one in the time domain and one in the Fourier domain. Both masks work to restrict the areas of consideration in their respective domains.

The mask in the time domain is equivalent to P_1 defined above. This mask limits the source image to certain dimensions and is defined as

$$X_s(t) = \begin{cases} 1 & t \in \delta \\ 0 & t \notin \delta \end{cases} \tag{3.13}$$

where δ is a finite slice of the time axis within which the signal $f(t)$ is of interest.

A mask defined over the Fourier domain is equivalent to P_2 defined above. It results in the elimination of certain Fourier components on the Ω axis. The function of the frequency mask is defined by:

$$\chi_c(\Omega) = \begin{cases} 1 & \Omega \in \mathfrak{L} \\ 0 & \Omega \notin \mathfrak{L} \end{cases} \tag{3.14}$$

where $\mathfrak{L}$ is defined as a low pass region seeking to limit the frequency band of the signal under consideration.

The one-dimensional discrete Fourier transform (1-D DFT) of the sequence is given by:

$$F(\Omega) = \int_{\delta} f(t)\, e^{-j\Omega t}\, dt \tag{3.15}$$

For an arbitrary function $\phi(t)$ in H, the G-P algorithm is compactly represented as

$$\phi_{k+1} = P_1 P_2 P_4 P_5 \phi_k \tag{3.16}$$

The transformations P_4 and P_5 are trivial while P_1 and P_2 are defined as follows:

$$P_1\phi = X_s(t)\ \phi(t) = \begin{cases} \phi(t) & t \in \delta \\ 0 & t \notin \delta \end{cases} \tag{3.17}$$

$$P_2\phi = \mathfrak{F}^{-1}\{G + (1 - \chi_c)\Phi\} \qquad \phi \longleftrightarrow \Phi \tag{3.18}$$

and

$$G(\Omega) = \begin{cases} \Phi(\Omega) & \Omega \in \mathfrak{L} \\ 0 & \Omega \notin \mathfrak{L} \end{cases} \tag{3.19}$$

Thus the G-P algorithm can be completely described as:

$$\begin{aligned} \phi_{k+1} &= \mathfrak{F}^{-1}\{G + (1 - \chi_c)\ \mathfrak{F}(X_s\, \phi_k)\} \\ \phi_0 \ &= \mathfrak{F}^{-1}\{G\} \end{aligned} \tag{3.20}$$

In this method an attempt is made to iteratively restore a signal from its lowpass projection subject to certain predefined constraints.

CHAPTER 4. INTRODUCTION TO VECTOR QUANTIZATION

Consider a 1-dimensional discrete time sequence which may have been obtained by sampling a continuous time waveform. From these individual time samples, we can form K-dimensional vectors by grouping K contiguous samples at a time. The amplitudes of the individual samples of a group form the K elements of a vector. In Vector Quantization (VQ), we quantize the entire vector as a single unit instead of quantizing the individual discrete time samples [19]. This is simply a K-dimensional extrapolation of the well known scalar quantization technique.

Consider an digitized image that consists of a grid of 512×512 pixels. Each pixel is a single digital sample with a definite intensity value (magnitude). The entire image (grid) is partitioned into blocks of 4×4. Each block then represents a member of a $16 - D$ vector space. Quantization of vectors in such a space is then defined analogous to quantization of samples in a 1-D space.

In order to explain VQ, we will start with scalar quantization (or 1-dimensional VQ). The 1-dimensional space R^1 is represented by the real line from -∞ to ∞. This entire line will be mapped on to a finite subset of itself consisting of points $r_1, r_2, \ldots r_{n-1}$. Figure 4.1 is a uniform quantizer where $t_1, t_2, \ldots t_n$ represent the transition levels and $r_1, r_2, \ldots r_{n-1}$ the reconstruction levels. Any input value that has an amplitude between t_i and t_{i+1} will be mapped to r_i. This means that for

all subsequent operations r_i effectively substitutes for any value between t_i and t_{i+1}. Obviously, such an operation introduces a certain non zero distortion which is referred to as the quantization noise. This quantizer is termed as a scalar quantizer since it quantizes one discrete value at a given instant.

Figure 4.2 is a vector extrapolation of the scalar quantizer discussed above. It depicts a K-dimensional vector space which is partitioned into n subsets. The boundaries of these subsets are represented by $T_1, T_2, \ldots T_n$ called the transition hyperplanes. Each of these subsets is represented by a single vector known as the representative vector. In Figure 3.2, $R_1, R_2, \ldots R_N$ denote the representative vectors. Thus, the entire vector space is represented by n representative vectors.

A collection of these representative vectors, and so, implicitly, the transition hyperplanes is referred to as the 'Codebook' [19]. The idea of the codebook is discussed in detail in the next section. Any vector in the vector space falls within the boundaries of one of these subsets and so is replaced by the representative vector of that subset in any further operations involving the input vector in question. Here too, there is certain non zero distortion introduced due to the quantization operation. A good VQ design tries to minimize this error in some sense.

VQ will yield good results only when a small number of vectors (among all the possible ones) have high probabilities of occurrence, while the majority of the vectors have extremely low probabilities ($<< 1$) of occurrence. Thus, it is necessary to determine the distribution of the parameters being (vector) quantized before applying the procedure itself. The most important advantage of vector quantization is that it provides a framework for realizing bit rates less than 1 bit/pixel. This is impossible to achieve in a scalar quantization technique.

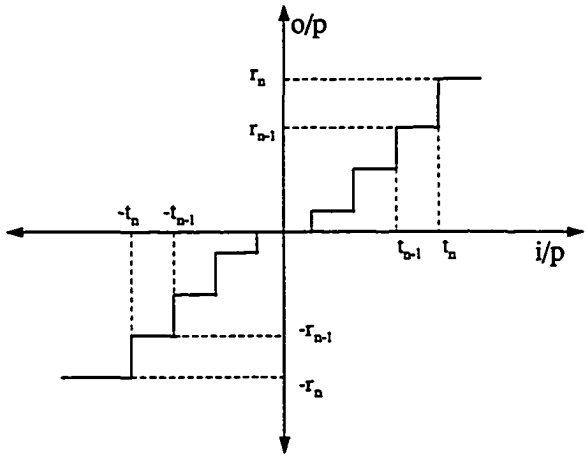


Figure 4.1: A diagrammatic representation of a scalar quantizer

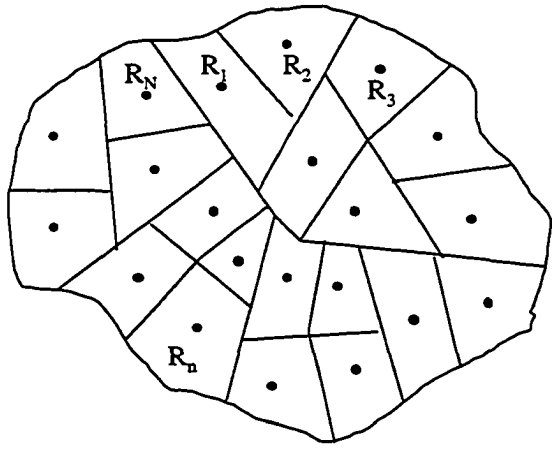


Figure 4.2: A diagrammatic representation of a vector quantizer

4.1 Theoretical Definition of Vector Quantization (VQ)

Vector Quantization is formally defined as the mapping of a K-dimensional real vector space into a finite subset of itself [21], i.e.,

$$Q: \quad R^k \rightarrow Y \tag{4.1}$$

where Q is the defining function of the vector quantizer, R^k is the K-dimensional subspace and Y is a finite subset of R^k.

Mathematically, the operation of VQ is defined by the following expressions

$$\begin{aligned} r_n &= Q(x_n) \\ u_n &= \gamma(r_n) \\ \tilde{x}_n &= r_n = \beta(u_n) \end{aligned} \tag{4.2}$$

where Q is the quantizer function

x_n is the input vector

r_n is the codevector to which the input vector is mapped to

γ is the coding function

u_n is the 'index' or the code symbol for the codevector r_n

β is the decoding function and the inverse operation of γ

A good VQ has to contend with two problems,

1) To obtain a set of representative vectors that are evenly distributed in the vector space (or a part of it) such that any possible input vector can find a representative vector close enough to keep the resulting distortion to a minimum in some sense.

2) To keep the number of representative vectors as small as possible for a given measure of distortion.

4.2 Codebook of a Vector Quantizer

A codebook is the heart of a vector quantizer. In fact, codebook and VQ are synonymously used in the literature. Codebook is a collection of all the representative vectors and, implicitly, the transition hyperplanes in a single file. In the 1-dimensional quantizers, the transition and reconstruction levels form the analog of a codebook. These transition and reconstruction levels are explicitly used in quantizing input values. In VQ, the transition hyperplanes can not explicitly be defined in the codebook, and also do not come into picture during quantization. However, in the operation of determining the nearest representative vector (in some sense), the implicit existence of the transition hyperplanes is acknowledged. This determination of the right representative vector from the entire collection of vectors in the codebook is termed as a "codebook search" and the time expended in doing this is called 'search time'.

The inherent inability to explicitly locate the transition hyperplanes in higher dimensional quantizers results in a large search time to determine the right representative vectors. In 1-dimensional quantization the input value has to be grouped between any two transition levels, and the corresponding reconstruction level is bound to be the best choice. However, in multidimensional cases, the input vector has to be compared with all the representative vectors in the codebook. A measure of distance has to be defined to find a representative vector that is nearest in some sense, and use that vector as the true representation for the original input vector. This measure of distance is generally referred to as a 'distortion measure' in data compression literature. There are a few different expressions for determining the distortion measure. Some of the more widely used ones are:

(1) The absolute error distortion: In this distortion measure, the absolute value

of the difference between the two vectors in question serves as a measure of distortion. The expression for this measure is given by

$$d(x,\hat{x}) = \|x - \hat{x}\| = \sum_i |x_i - \hat{x}_i| \tag{4.3}$$

where $d(x,\hat{x})$ is the distortion measure and $\|x - \hat{x}\|$ is the norm of the difference between the two vectors x and $\hat{x}$.

(2) The squared error distortion: This distortion measure also goes by the name MSE distortion. This distortion measure is proportional to the square of the norm of the difference between two vectors

$$d(x,\hat{x}) = \|x - \hat{x}\|^2 = \sum_i (x_i - \hat{x}_i)^2 \tag{4.4}$$

where the operators are defined and symbols are as defined in (4.3).

The definitions for the different types of distortion measures between two vectors calculated instantaneously leads to another distortion parameter called the average distortion measure. Average distortion measure can be defined as the sum of all the individual distortions divided by the total number of individual distortions under consideration. If an image has N individual block distortions defined during an operation then the average distortion is given by the expression

$$D(x,\hat{x}) = \frac{1}{N} \sum_{n=1}^{N} d(x_n, \hat{x}_n) \tag{4.5}$$

4.3 Implementational Trade-offs in a Vector Quantizer

The three major aspects to be considered with any data compression technique are 1) Bit rate 2) Execution complexity and 3) Subjective/Objective fidelity. Implementation of VQ, like any other technique, involves trade-offs between these three

aspects. Larger compression ratios imply lesser subjective/objective fidelity and vice versa assuming a constant execution complexity. The ongoing research in any compression method is to find ways to provide better fidelity at lower bit rates and a reasonable complexity.

Bit Rate: In case of VQ, the bit rate is dependent upon the size of the codebook (dictionary of representative vectors). For a $16 - D$ vector space, the nominal size of the codebook to obtain visually tolerable reconstructions has experimentally been determined to be about 256 vectors. But, any further reduction in the codebook size results in annoying subjective distortions [7].

Execution complexity: The standard implementation of VQ is very fast, particularly at the decoder but suffers from severe edge degradation. Several improvements over the standard VQ have been suggested in literature to overcome the problems. All these suggestions add significant amounts of complexity to the standard approach and increase the execution time.

Subjective/Objective fidelity: Fidelity of a reconstruction can be viewed from two different perspectives. Objective fidelity is a measure of accuracy between the original and coded waveforms. In case of an image, it is the pixel-by-pixel match between the two. Subjective fidelity is a practical measure of how good the differences between the two waveforms are 'masked'. This is actually a function of discernability of the human eye. A reconstruction with very good subjective fidelity may not have good objective fidelity. Current thrust in compression of real images is to obtain reconstructions that have good subjective fidelity even at the expense of objective fidelity.

4.4 Classified Vector Quantization (CVQ)

As mentioned in the previous section reduction in the codebook size results in severe subjective and objective distortions of the overall reconstructed image. An obvious way to counter this problem is to localize the distortions to certain areas in the image. Classified Vector Quantization (CVQ) is a direct result of this argument. The CVQ approach essentially splits the image into areas of different levels of activity ex., shade regions and edge regions. Thus, distortions can be localized by reducing the number of representations for one region (edge region) with respect to the other. Resorting to this approach, however, several key problems are encountered. The two most serious problems are: (1) Edge degradation and (2) High complexity.

Edge degradation: Image vectors can be classified based on their perceptual effect on the observer. A very broad classification can be shade and edge vectors. Edge and shade vectors do not affect human perception in the same way. Separate distortion measures are required to evaluate shade and edge regions [25]. Edge degradation occurs both in the classical VQ and CVQ approaches. In the classical VQ approach, a codebook constructed using the MSE distortion as a measure, processes all vectors in a space identically, without regard to their perceptual effects. In the case of CVQ, distortions in one region is exacerbated due to the localization property of the approach. Edges constitute a very significant portion of subjective information even though the number of edge vectors are very few in any given image.

High complexity: The complexity of VQ grows exponentially with block size and compression ratio. At high compression ratios (< 0.5 bpp), a simple VQ is no longer sufficient to obtain visually good reconstructions. Improved versions of VQ based on psychophysics of the human vision are to be used. Any such refinements result in

higher complexity and longer execution time for the VQ.

This section deals with one of the more successful approaches to reduce the aforementioned problems. The method was reported by Ramamurti and Gersho in 1986 [?]. The method is known as 'Classified Vector Quantization (CVQ)'. The crucial feature of edge perception is that an edge must appear essentially as sharp, continuous and well defined in the coded image as in the original. One of the important advantages of preserving an edge is that the intensity levels on either side of it need not be reproduced precisely. This is what is known as the *masking effect.* The masking effect permits a very coarse coding of pixels near edges only if the integrity of the edge itself is preserved. The CVQ is proposed as a technique for preserving perceptual features while retaining simple distortion measures like MSE. It is best understood in terms of a composite source model for images where the image source is viewed as a bank of vector subsources. Each subsource generates blocks of a single perceptual class.

4.4.1 The CVQ Classifier

The CVQ is best modelled in terms of a *composite source* model for images, where the source is viewed as a bank of vector subsources, as shown in Figure 3.3. Each subsource generates blocks of a single perceptual class, e.g., blocks with an edge at a particular orientation and location. At each instance, a switch selects on one of the subsources whose output then becomes the output of the composite source. Each subsource is characterized by its own codebook, called a *subcodebook*, which is very small compared to the codebook of the composite source. At each instance, a full search is performed only within the subcodebook hence reducing the search time

dramatically.

The classification algorithm is implemented in two steps, an edge enhancement step followed by a decision tree which extracts edge information from the enhanced version. The enhancement process is described below:

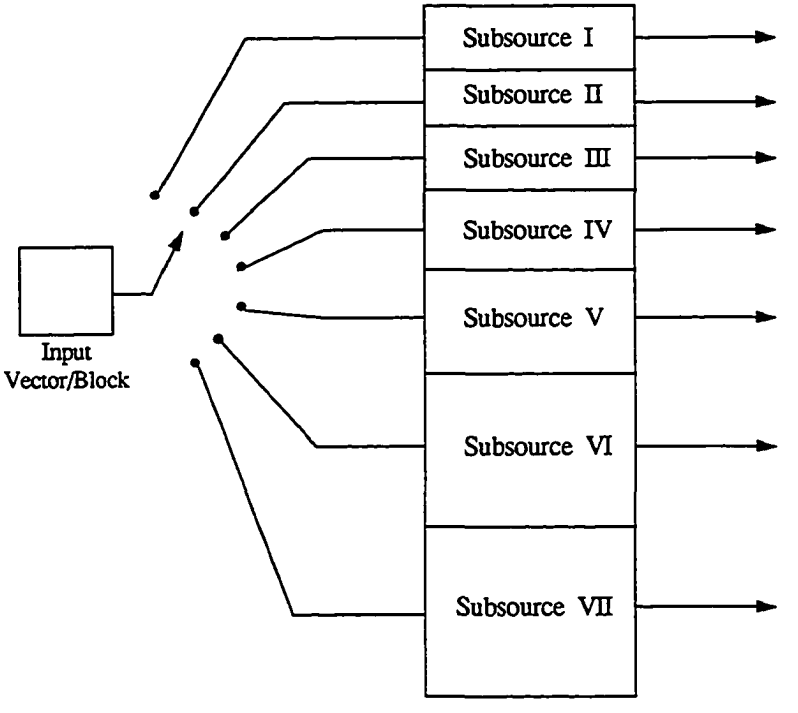


Figure 4.3: A diagrammatic representation of a composite source

The gradients in the horizontal and vertical directions normalized by the average intensity (d_h & d_v) are first calculated. This is because the sensitivity of the human eye is proportional to the normalized gradient and not to the absolute gradient (i.e., the perception of the gradient by the human eye is dependent on the intensity values

of the pixels in question). The expressions for the two gradients are given by

$$d_h = \frac{2(x_{i,j} - x_{i,j+1})}{(x_{i,j} + x_{i,j+1})} \tag{4.6}$$

$$d_v = \frac{2(x_{i,j} - x_{i+1,j})}{(x_{i,j} + x_{i+1,j})} \tag{4.7}$$

The thresholds T_e and T_s are functions of the average intensity, denoted by d_{av}, of the two pixels under consideration when a normal gradient is being compared, and are defined by

$$T_e = \begin{cases} \frac{8.0}{d_{av}} & \text{if } d_{av} < 30.0 \\ 0.2 & \text{otherwise} \end{cases} \tag{4.8}$$

$$T_s = \begin{cases} 0.1 & \text{if } d_{av} < 30.0 \text{ or } d_{av} > 225 \\ 0.2 & \text{otherwise} \end{cases} \tag{4.9}$$

The entries of the tables G_h and G_v are calculated as follows:

$$G_h(i,j) = \begin{cases} 1 & \text{if } d_h > T_e; \quad i = 1, 2 \ldots, p \\ -1 & \text{if } d_h < -T_e; \; j = 1, 2, \ldots, p-1 \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

$$G_v(i,j) = \begin{cases} 1 & \text{if } d_v > T_e; \quad i = 1, 2 \ldots, p-1 \\ -1 & \text{if } d_v < -T_e; \; j = 1, 2, \ldots, p \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

The counters S_h and S_v are incremented according to the following rule:

$$S_h = S_h + 1 \quad \text{if} \quad |d_h| > T_s, \; i = 1, 2, \ldots, p \quad j = 1, 2, \ldots, p-1 \tag{4.12}$$

$$S_v = S_v + 1 \quad \text{if} \quad |d_v| > T_s,\ i = 1, 2, \ldots, p-1 \quad j = 1, 2, \ldots, p \tag{4.13}$$

The Decision Tree:

The decision algorithm works with all the computed tables and counters. H_p is the sum of $+1$'s in G_h and H_n is the sum of -1's. V_p is the sum of $+1$'s in G_v and V_n is the sum of -1's.

A block X is considered to be a shade block if the counters S_h and S_v satisfy the following conditions:

$$S_h < J_s \quad \text{and} \quad S_v < J_s \tag{4.14}$$

It is an edge block with horizontal component of positive polarity and negative polarity if:

$$V_p \geq J_e \quad \text{and} \quad S_v \geq J_e \tag{4.15}$$

respectively.

It is an edge block with vertical component of positive polarity and negative polarity if:

$$H_p \geq J_e \quad \text{and} \quad H_n \geq J_e$$

respectively.

In the case when both the above conditions hold the block X is considered to be of a mixed class. The choice of the thresholds J_s and J_e determine what can be considered as random intensity change and what is not. This paper suggests a composite source modeling approach to VQ encoding. The fidelity of the reconstructions depend on the accuracy of 1) the source model and 2) the representative vectors within a modeled subsource.

Two approximations are inherent in the CVQ approach. They are: 1) For small block sizes, a satisfactory description of edge with any orientation is approximated by a closest straight line across which the intensity changes abruptly. 2) The number of admissible edge orientations are restricted to four: horizontal, vertical and two diagonals. Even with such a minimal number of orientations there are several edge vectors which are distinctly different and need to be represented in the codebook. Even with these restrictions the size of the codebook does not reduce drastically. In fact, the authors of the method themselves suggest a codebook of size 256 with about 217 codevectors catering to the edges. This is because the VQ has to cater to edges that have different variations in the intensity values across the edges. Also, edge transitions in real images are seldom abrupt at the pixel level, but occur over several pixels. Hence it is the opinion of this study that edges of real images cannot be modelled to any reasonable accuracy without exhaustively catering to all types. This, of course, increases the bit rate.

CHAPTER 5. IMAGE RESTORATION

The proposed dissertation work is to study the applicability of the theory of convex projections (CP) for data compression of real images. The motivation for this research is that if image restoration can be used as a part of the decoding process, the result may be a reduction in the amount of information to be transmitted. This is because the decoder will then try to generate the necessary information by extra(inter)polating from the available data. The technique is proposed to be implemented in the spatial domain using the Vector Quantization approach.

5.1 Motivation for Image Restoration

Every image coding method that uses a block coding approach makes a trade-off between two key factors: (1) edge degradation and (2) higher complexity.

Image blocks can be classified based on their perceptual effect on the observer. A very broad classification can be shade and edge blocks. The statistical properties of the two types of blocks are different. The information content of an average edge block is much larger when compared to a shade block. Larger compression ratios (lower bit rates) result in larger distortion of the edges both perceptually and objectively. Thus image restoration can be a viable tool to restore edges based on information provided by the (much less distorted) shade regions, if there is enough mutual information

between them.

Since the statistical properties of shade and edge regions are vastly different, it is necessary that the two regions be coded differently. The classification of regions into different types depends on the effect they have on human vision. In order to achieve good perceptual quality in the decoded images, improved versions of image coding techniques based on psychophysics of the human vision are to be used. Any such refinements in standard coding techniques result in higher complexity and longer execution time. Current thrust in image compression is to obtain reconstructions that have good subjective fidelity even at the expense of objective fidelity. The primary objective is limited to obtaining reconstructions with good subjective fidelity, rather than obtaining a pixel–to-pixel match between the original and the reconstructed images. This dissertation study investigates the convex projection theory as a means to achieve that objective with high compression capability.

The concept of convex projections (CP) does not lend itself to a straightforward implementation in the context of VQ. Even though the codebook of a VQer is indeed a subset of a suitably defined vector/Hilbert space, it is seldom convex and the operation of VQ itself is nonlinear. This is because larger compression ratios dictate smaller codebook sizes resulting in smaller subsets that are highly non-convex. As a matter of fact, a codebook of any size is strictly non convex. But, with increasing size, it approaches the space it actually represents. In the limiting case of infinite size the codebook becomes indistinguishable from the space it represents.

Also, VQ is a highly non linear operation. Encoding of a large image is achieved by partitioning it into constituent smaller image blocks and encoding each of them independently. Such an operation results in distinct blocking effects at the edges of

the partitions. Image restoration in such a case, translates to eliminating the blocking effect with as small a size of the codebook as is practical. The "blocking effect" is a common observation in every known automatic image compression algorithm. In the case of VQ, the effect is very pronounced at the edges than in shade regions. Several approaches have been suggested in the literature to alleviate this problem. One of the more successful attempts in this regard was the Classified Vector Quantization (CVQ) described in the previous chapter.

This dissertation study is an attempt at restoring the edges of an image algorithmically with the help of image recovery techniques, specifically, the theory CP. The approach is to allow an iterative algorithm to progressively reconstruct the edges based on the shade pixels in the 'immediate neighborhood'. The method assumes that there is sufficient mutual information to be exploited between any edge and its surrounding shade regions. Thus, the contour of an edge is dependent on the orientation of the shade pixels on either side of it. The edge reconstruction is left as a problem to be addressed by the theory of CP. The ultimate goal is to avoid an extensive coding of the edges. The technique used in this study is similar to that reported by Sezan and Stark in their paper on image recovery and convex projections [6] and was found effective.

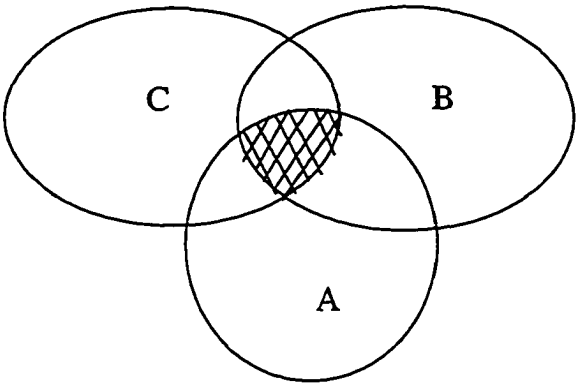
The goal is to have a bit rate lower than 0.35 bpp reported by Ramamurti and Gersho [20]. This is possible because the decoder has a restoration algorithm which interpolates non existent information using some constraints. The recovery algorithm presented in this study is an extension of the idea developed Gerchberg and Papoulis (G-P) for 2-D images. We propose to test the approach primarily on real photographic grayscale images and evaluate the reconstructions for subjective fidelity. In the next

section, we develop 2-D mathematical constructs necessary for the application of the CP theory in the context of a vector quantizer.

5.2 Convex Projections as Applied to Image Recovery

The lossy nature of the encoder assures that encoding is not perfectly invertible at the decoder. This simply means that the decoder can never reconstruct the actual encoded image. It is obvious that more than one image realization satisfies the encoded bit stream. For instance, both the input image (at the encoder) and its lossy reconstruction (at the decoder) are represented by an *identical bit stream* in the channel. Specifically, there exist a set of "images", including the original, that will all yield the same data as the received image when compressed by the same encoding algorithm. In a conventional encoder/decoder tandem the image is decoded as a simple inverse operation with respect to the encoder. However, the judge of the compressed image quality is a human observer. Thus, selecting a particular realization based on constraints posed by human perception without violating the bit stream integrity will yield images with better subjective fidelity.

Conceptually, convex projections (CP) is a method for selecting particular realizations from a larger set that match any given set of constraints. The more the constraints match the original image, the better the decoded image. In Figure 5.1, set A denotes all images that have an identical bit stream when encoded with a particular VQ. Of course, the original image is one of them. Set B denotes a set of all images that have an upper bound on the slope at the interblock boundaries. Set C denotes a set of all images that have identical shade representations but differing edge representations.



A - Images with identical encoding bit stream
B - Images that satisfy a 'constraint'
C - Images that satisfy a second 'constraint'

Figure 5.1: Conceptual illustration of the projection approach

The intersection set of these three constraint set will definitely contain realizations that have better subjective fidelity than those that belong to set A alone.

5.3 Review of Related Work

The practical application of Convex Projections (CP) to image restoration is a relatively new area of research. The very first paper that deals exhaustively with a specific compression method (Block Discrete Cosine Transform, BDCT) was published by Yang and Galatsanos in International Conference on Image Processing (ICIP, Nov. 1994)[10]. This section briefly describes the work reported in that paper. A second paper on the same topic was published by Su and Mersereau in ICASSP, May 1995 [12].

The BDCT compression method partitions an entire image into blocks of 8×8 pixels and transforms each block independently. The transformed coefficients are quantized before transmission. Data compression is achieved by varying the number of quantizer levels (i.e, bits) allocated to each of the coefficients. At low bit rates, the reconstructed image suffers from perceptible blocking effects in the shade regions and ringing effects at edges. The authors in both publications use approaches based on the CP theory in order to eliminate the blocking and ringing effects.

However, there has been no reports or publications on application of the CP theory to compression approaches based on Vector Quantization.

5.4 Vector Quantization and Convex Projections

5.4.1 Multiple Codebooks (General Case)

Since, the theory of CP is developed for Hilbert spaces, images have to be represented as elements from a Hilbert space. An image of dimensions $N \times N$ pixels can be considered to be a member of a N^2-D integer vector space. In the case of 8 bit grayscale images, all vectors are composed of *mod* 255 scalars. For the purposes of VQ, the image can be considered to be a mosaic of constituent image blocks of size $n \times n$ $(n < N)$ pixels. These blocks can in turn be mapped onto a subset of vectors from a n^2-D Hilbert space. The underlying principle in the operation of VQ is that *the projection in a larger vector space can be approximated as a spatial union of projections in a smaller vector space.* In other words, encoding a large image can be accomplished by:

- Partitioning the image into a mosaic of smaller image blocks.

- Encoding each of the smaller blocks individually and retiling them back to reconstruct the larger image.

If b is a image (dimension: $N \times N$) composed of ordered smaller image blocks (dimension $n \times n$) $f^{(a)}, a = 1, 2, 3, ...\infty$, then the image can mathematically be represented as:

$$b = \sum_{i=0}^{N/n} \sum_{j=0}^{N/n} f^{(a)}(x - ni, y - nj) \tag{5.1}$$

Each individual block is characterized as either a shade or an edge block depending on the activity level of each block. A set of criteria to quantify the activity level of an image block used in this study is described in detail in Chapter 4 [20].

5.4.2 Sets and Projections Associated with the Image Data

Let $b \longleftrightarrow B$ $(i.e., B = \mathfrak{F}\{b\})$ denote a 2-D Fourier transform pair. Let V_1 represent the shade partition and V_2, the edge partition of an integer vector space. The sets are then defined as follows.

(i) $\zeta_1 = \{g_m \mid g_m \in V_1, m = 0, 1, ...M_1\}$ corresponds to the shade codebook of the VQ. It denotes a subset of codevectors in a vector quantizer (VQer) that is a representative of the shade partition V_1. The association of a vector into the shade partition is described by Ramamurti and Gersho [20]. A copy of this codebook resides both at the encoder and the decoder.

(ii) $\zeta_2 = \{h_m \mid h_m \in V_2, m = 0, 1, 2, ...M_2\}$ corresponds to a *low resolution* edge codebook of the VQ. It denotes a subset of codevectors in a vector quantizer (VQ) that is a very weak representative of the edge partition V_2. The association

of a vector into the edge partition is described by Ramamurti and Gersho [20]. A copy of this codebook resides both at the encoder and the decoder.

(iii) $\zeta_3 = \{l_m \mid l_m \in V_2, m = 0, 1, 2, ...M_3\}$ corresponds to a *high resolution* edge codebook of the VQ. It denotes another subset of codevectors in a vector quantizer that is a strong representative of the edge partition V_2. The difference between ζ_2 and ζ_3 is in the number and distribution of vectors in each subset ($M_3 >> M_2$). This codebook resides only at the decoder.

(iv) $\zeta_4 = \{b \mid B[u, v] = 0, \forall [u, v] \notin \chi_c\}$ denotes the subset of images whose Fourier transform components equal 0 outside a certain region χ_c in the $u - v$ plane.

(v) $\zeta_5 = \{b \mid \alpha \leq e_n \leq \beta, \forall e_n \in b\}$ denotes a subset of all images whose individual pixels lie between a prescribed closed interval $[\alpha, \beta]$.

(vi) $\zeta_{NL} = \{b \mid \forall f \in V_1,\ f = K_m,\ K_m \in \zeta_1,\ m = 0, 1, 2, ...M_1\}$ denotes a subset of all images that have identical shade vectors but different edge vectors.

The projections on these subsets are defined as given below:

i. *Projection onto* ζ_1: If f is any arbitrary vector in the n^2-D Hilbert space then the VQ operation is defined as:

$$\hat{f} = P_1 f = \begin{cases} \min_m \|f - g_m\|_2 & \text{if } f \in V_1 \\ 0, & \text{if } f \notin V_1 \end{cases} \tag{5.2}$$

where $\hat{f}$ is the vector that has the smallest L_2 norm with f in the subset ζ_1.

ii. *Projection onto* ζ_2: If f is any arbitrary vector in the n^2-D Hilbert space then the VQ operation is defined as:

$$\widehat{f} = P_2 f = \begin{cases} \min_m \|f - h_m\|_2 & \text{if} \quad f \in V_2 \\ 0, & \text{if} \quad f \notin V_2 \end{cases} \tag{5.3}$$

where $\widehat{f}$ is the vector that has the smallest L_2 norm with f in the subset ζ_2.

iii. *Projection onto* ζ_3: If f is any arbitrary vector in the n^2-D Hilbert space then the VQ operation is defined as:

$$\widehat{f} = P_2 f = \begin{cases} \min_m \|f - l_m\|_2 & \text{if} \quad f \in V_2 \\ 0, & \text{if} \quad f \notin V_2 \end{cases} \tag{5.4}$$

where $\widehat{f}$ is the vector that has the smallest L_2 norm with f in the subset ζ_3.

iv. *Projection onto* ζ_4: The projection is defined by a lowpass mask in the Fourier domain. If b is any vector obtained as a spatial union of vectors from the n^2-D Hilbert space then the operation in this subset is defined as:

$$P_4 b = \begin{cases} B[u,v], & \text{if} \quad (u,v) \in \chi_c \\ 0, & \text{if} \quad (u,v) \notin \chi_c \end{cases} \tag{5.5}$$

where $b \longleftrightarrow B$ is the 2-D Fourier transform pair and b itself is a spatial union of vectors (i.e., f's) from the n^2-D space as represented by (1).

v. *Projection onto* ζ_5 : If b is any arbitrary vector (image) in the N^2-D Hilbert space and $e_1, e_2, e_3 \ldots$ are its elements then the projection operation $\overline{f}$ is defined as:

$$\overline{f} = P_5 b = \begin{cases} \alpha & \text{if} \quad e_n < \alpha \\ e_n & \text{if} \quad \alpha \leq e_n \leq \beta, \quad \forall n \\ \beta & \text{if} \quad e_n > \beta \end{cases} \tag{5.6}$$

vi. *Projection onto* ζ_{NL} : If b is any arbitrary image, with its vectors classified to belong to either V_1 or V_2, then the projection operation P_{NL} is defined as

$$P_{NL}\hat{b} = \sum_{i=0}^{N/n}\sum_{j=0}^{N/n}\left[K_m^{(a)}(x-ni, y-nj) + P_3 f^{(a)}(x-ni, y-nj)\right] \qquad (5.7)$$

where $K_m^{(a)}$ is the shade vector that is identical at that position to all the images in the subset ζ_{NL} and all edge vectors are VQ'ed using the subset ζ_3.

During the encoding operation of VQ, subsets ζ_1 and ζ_2 are merged to form a single codebook. The operation of VQ inherently introduces additional high frequency components into the decoded image in the form of discontinuities at every block boundary. ζ_3 and ζ_4 are defined to eliminate these discontinuities. However, all these subsets are not strictly convex, particularly $\zeta_1 \cup \zeta_2$ and ζ_3 due to the inherent non convexity of quantization operation. This can also be shown by the fact that the sets defined above do not satisfy the fundamental definition of convex sets:

Definition 10 *A subset* ζ *of a Hilbert space is said to be convex if together with any* f_1 *and* f_2 *it also contains* $\mu f_1 + (1-\mu) f_2$ *for all* μ , $0 \leq \mu \leq 1$.

This is the reason for imperfect restoration by the method of CP as applied to VQ. However, this does not negate the effectiveness of such methods but only serves to define their limits. Algorithms based on the theory of CP can and do work to restore images, at least partially, and hence improve the reconstruction (from a subjective point of view).

Mathematically, the operation of VQ after classification into shade and edge vectors can be expressed as:

$$\begin{aligned} &\widehat{b} = (P_s + P_e)\, b = \textstyle\sum_{i=0}^{N/n} \sum_{j=0}^{N/n} \left[P_1 f^{(a)}(x - ni, y - nj) + P_2 f^{(a)}(x - ni, y - nj) \right] \\ &\text{where} \qquad P_s\, b = \textstyle\sum_{i=0}^{N/n} \sum_{j=0}^{N/n} P_1 f^{(a)}(x - ni, y - nj) \\ &\text{and} \qquad P_e\, b = \textstyle\sum_{i=0}^{N/n} \sum_{j=0}^{N/n} P_2 f^{(a)}(x - ni, y - nj) \end{aligned} \tag{5.8}$$

where $\widehat{b}$ is the decoded image composed of ordered image blocks which are projections of the vectors in either ζ_1 or ζ_2 at the corresponding positions in the original image. $P_s b$ is the VQ'ed image with all shade blocks reconstructed and edge blocks masked, while $P_e b$ is the VQ'ed image with all edge blocks reconstructed and shade blocks masked.

5.4.3 The Algorithm

Image restoration at the decoder is performed with an iterative algorithm which amounts to a repeated application of the projections defined in the previous section.

Step 1: The VQ'ed image is reconstructed at the decoder with shade and edge codebooks identical to those at the encoder, which is represented as $\widehat{b} = \widehat{b}_0$

Step 2: The reconstructed image is then partially smoothed (using a low pass filter) to reduce the 'blocking effect' inherent in the operation of VQ. This projection operation is symbolically equivalent to $P_4\, \widehat{b}$, a projection of the VQ'ed image onto its low frequency subband.

Step 3: The resulting image is then tested and corrected for any pixel intensity limit violations, i.e., $P_5 P_4\, \widehat{b}$.

Step 4: The smoothed image is remapped back onto the codebook, but with a modification in the mapping operation. All shade blocks are replaced with their corresponding unsmoothed versions from ζ_1, while the edge blocks are VQ'ed using the subset ζ_3. Thus, the sequence of projection operations depends upon the space (V_1 or V_2), to which the vector in question belongs to.

A mathematically compact notation for one iteration of the algorithm can then be given by

$$\begin{aligned} \widehat{b}_{k+1} &= P_{NL} P_5 P_4 \, \widehat{b}_k \\ \widehat{b}_0 \quad &= (P_s + P_e)\, b \end{aligned} \tag{5.9}$$

We also define an indicator function in the spatial domain as follows:

$$\Gamma = \begin{cases} 1, & \text{if } \widehat{f_0}^{(a)} = P_1 f^{(a)} \\ 0, & \text{otherwise} \end{cases} \tag{5.10}$$

where $f^{(a)}$ is the ath vector in the original image and $\widehat{f_0}^{(a)}$ is the corresponding decoded vector. The mathematical expression for image restoration is then given as below:

$$\widehat{f_{k+1}} = \Gamma \, \widehat{f_0} + (1 - \Gamma) \, \min_m \left\| \Im^{-1} \left[\widehat{B}_k \, [u, v] \quad (\chi_c) \right] - l_m \right\|_2 \tag{5.11}$$

where $\widehat{f_0}$ is the vector from the VQ'ed image, $\widehat{f_k}$ is the vector from the k^{th} iteration, $\widehat{B}_k \, [u, v] \, \chi_c$ is the lowpassed Fourier transform of the entire image after k iterations. The following assumptions are inherent in the interpretation of (5.11). All higher frequency components are localized exclusively to the edge blocks. Shade blocks contain negligible amounts of high frequency components, and so, are unaffected by the restoration algorithm.

5.5 Implementational Aspects

At the Encoder:

The input image is partitoned into subblocks of 4×4 pixels. Each subblock is classified as either a shade or an edge block. The mean of each block is predicted causally, both at the encoder and the decoder as a function of the nine adjacent pixels as shown in the Figure 5.2 [22]. If the difference between predicted mean and the actual mean (at the encoder) of a block is greater than a threshold, the actual mean is transmitted as channel information. If not, the predicted mean is used during decoding.

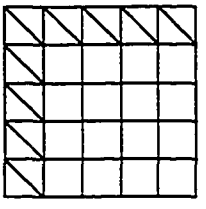


Figure 5.2: Calculation of the average value of an image block

The mean removed blocks are then quantized using a VQ. The codebook for the VQ is composed of two subcodebooks, a shade codebook (20-25 codevectors) and a vestigial edge codebook (8-10 codevectors). This operation is a non-linear approximation of projection of a vector space onto its subset. The indices of the codebook vector nearest to a vector in question is entropy (Huffman/Arithmetic) coded and transmitted through the channel.

At the Decoder:

Indices are received from the channel, and the image is reconstructed using

the corresponding codebook vectors. The resulting decoded image has good shade reproduction but annoyingly degraded edges. The CP algorithm is used to restore the degraded edges without altering the shade regions. This restoration is not exact but experiments on real grayscale images produced subjectively pleasing edges with near complete elimination of the staircase effect.

The result of the projection in Step 2 of the algorithm can be interpreted as follows: The smoothing operation is equivalent to letting the edge subset and the shade subset mutually project onto each other. This operation has very little effect on the shade blocks that are far removed from the edges. However, it drastically affects all the edge blocks and shade blocks in their vicinity. Step 4 effectively nullifies all projections of edge subset onto the shade subset while letting the shade subset project on to the edge subset. Steps 1-4 are performed iteratively to obtain a final image with smallest l_2 norm between the original and the decoded image.

The geometrical representations of the subspaces along with their representative codebook vectors are shown in Figures 5.3 and Figure 5.4.

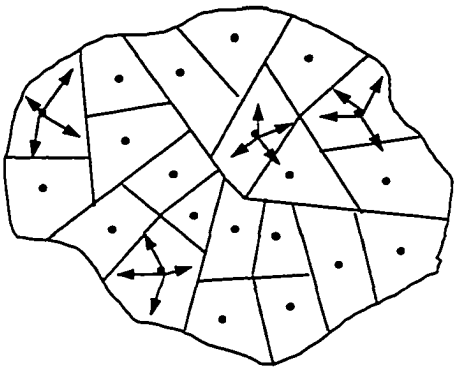
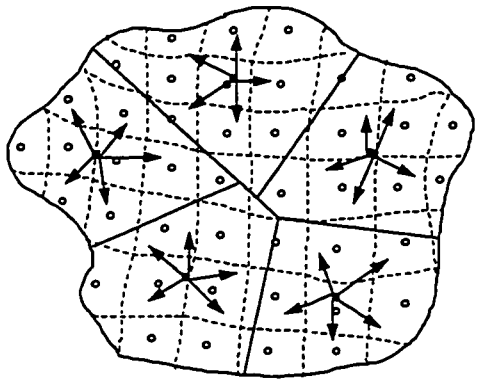


Figure 5.3: Shade space with codevectors and corresponding partitions

Figure 5.3 represents the shade subspace and its representative vectors. The arrows represent the displacements various shade vectors undergo during the smoothing operation. It is assumed that the magnitude of these displacements are do not cross the transition hyperplanes of the respective domains.



■ Representative vectors in the edge subset at the encoder
o Representative vectors in the edge subset at the decoder

Figure 5.4: Edge space with a low resolution and a high resolution codebook and corresponding partitions

Figure 5.4 represents the edge subspace and the corresponding representative vectors. The codebooks of both the encoder and the decoder are superimposed on each other. The solid lines represent the transition hyperplanes of the encoder and the dotted lines those of the decoder. The arrows represent the displacements of various edge vectors undergo during the smoothing operation. Since the edge codebook is very much larger at the decoder than at the encoder, the displacements are large enough to reproject the edge vectors to different domains based on the direction of displacement of each vector. Thus, by projecting the edge vectors onto a more

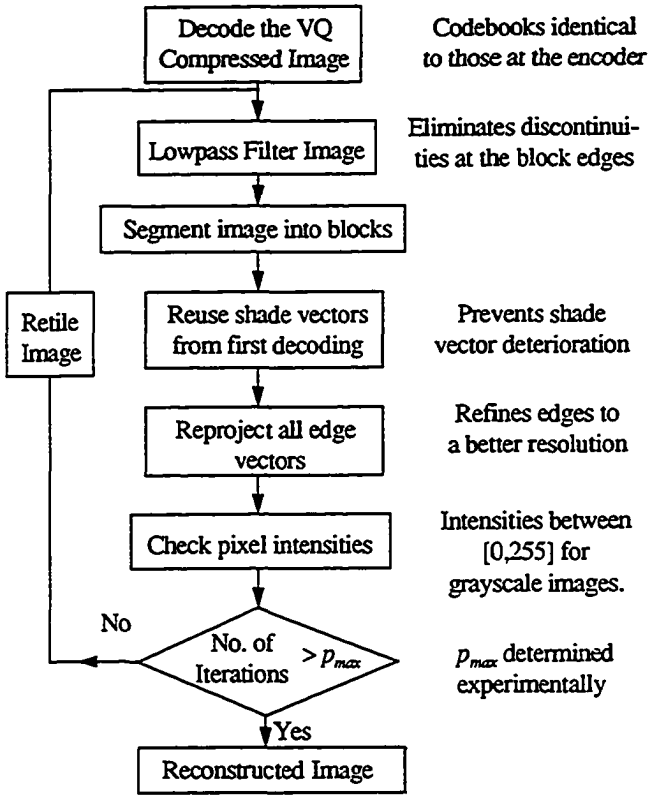


Figure 5.5: A schematic representation of the projection algorithm

elaborate subset compared to that at the encoder, we seek to restore the edges at least partially.

The aforementioned theory of using two different (low and high resolution) codebooks at the decoder can be extended to multiple high resolution codebooks as follows:

Proposition 11 *Decoding can be performed at several resolutions using progressively larger and more detailed edge codebooks at every iteration. The limiting resolution would be that of the encoded original image.*

5.5.1 Decoding with Spatial Controlled Smoothing (Particular Case)

This case is applicable only when the encoding/decoding block sizes are small, typically 4×4. In the theory presented in the general case, if the subset $\zeta_3 \rightarrow \infty$, we have a particular implementation of the restoration algorithm where each of the perturbed edge vector maps onto itself in every iteration. Then, the convergence of the entire image depends solely on the converging ability of the individual edge vectors. The modified version of the theory for this case is presented below.

5.5.2 Sets and Projections Associated with the Image Data

Let $b \longleftrightarrow B$ $(i.e., B = \mathfrak{F}\{b\})$ denote a 2-D Fourier transform pair. Let V_1 represent the shade partition and V_2, the edge partition of an integer vector space. The sets are then defined as follows.

(i) $\zeta_1 = \{g_m \mid g_m \in V_1, m = 0, 1, ...M_1\}$ denotes the subset of codevectors g_m in a vector quantizer (VQ) that are obtained from the shade partition V_1. The association of a vector into the shade partition is described by Ramamurti and Gersho [20].

(ii) $\zeta_2 = \{h_m \mid h_m \in V_2, m = 0, 1, ...M_2\}$ denotes the subset of codevectors h_m in a vector quantizer (VQ) that are obtained from the edge partition V_2. The association of a vector into the partition is described by Ramamurti and Gersho [20].

(iii) $\zeta_3 = \{b \mid B[u, v] = 0, \forall [u, v] \notin \chi_c\}$ denotes the subset of images whose Fourier transform components equal 0 outside a certain region χ_c in the $u - v$ plane.

(iv) $\zeta_4 = \{b \mid \alpha \leq e_n \leq \beta, \forall e_n \in b\}$ denotes a subset of all images whose individual pixels lie between a prescribed closed interval $[\alpha, \beta]$.

(v) $\zeta_{NL} = \{b \mid \forall f \in V_1,\ f = K_m,\ K_m \in \zeta_1,\ m = 0, 1, 2, ...M_1\}$ denotes a subset of all images that have identical shade vectors but different edge vectors.

The projections on these subsets are defined as given below:

i. *Projection onto* ζ_1: If f is any arbitrary vector in the n^2-D Hilbert space then the VQ operation is defined as:

$$\widehat{f} = P_1 f = \begin{cases} \min_m \|f - g_m\|_2 & \text{if} \quad f \in V_1 \\ 0, & \text{if} \quad f \notin V_1 \end{cases} \tag{5.12}$$

where $\widehat{f}$ is the vector that has the smallest L_2 norm with f in the subset ζ_1.

ii. *Projection onto* ζ_2: If f is any arbitrary vector in the n^2-D Hilbert space then the VQ operation is defined as:

$$\widehat{f} = P_2 f = \begin{cases} \min_m \|f - h_m\|_2 & \text{if} \quad f \in V_2 \\ 0, & \text{if} \quad f \notin V_2 \end{cases} \tag{5.13}$$

where h_m is the vector that has the smallest L_2 norm with f in the subset ζ_2.

iii. *Projection onto* ζ_3: The projection is defined by a lowpass mask in the Fourier domain. If b is any vector obtained as a spatial union of vectors from the n^2-D Hilbert space then the operation in this subset is defined as:

$$P_3\, b = \begin{cases} B[u, v], & \text{if} \quad (u, v) \in \chi_c \\ 0, & \text{if} \quad (u, v) \notin \chi_c \end{cases} \tag{5.14}$$

where $b \longleftrightarrow B$ is the 2-D Fourier transform pair and b itself is a spatial union of vectors (i.e., f's) from the n^2-D space as represented by (1).

iv. *Projection onto* ζ_4 : If b is any arbitrary vector (image) in the N^2-D Hilbert space and $e_1, e_2, e_3...$ are its elements then the projection operation $\overline{f}$ is defined as:

$$\overline{f} = P_4 b = \begin{cases} \alpha & \text{if } e_n < \alpha \\ e_n & \text{if } \alpha \leq e_n \leq \beta, \ \forall n \\ \beta & \text{if } e_n > \beta \end{cases} \tag{5.15}$$

v. *Projection onto* ζ_{NL} : If b is any arbitrary image, with its vectors classified to belong to either V_1 or V_2, then the projection operation P_{NL} is defined as

$$P_{NL}\widehat{b} = \sum_{i=0}^{N/n} \sum_{j=0}^{N/n} \left[K_m^{(a)}(x - ni, y - nj) \right] \tag{5.16}$$

where $K_m^{(a)}$ is the codevector that is identical in all the images at that particular position in the subset ζ_{NL}.

During the operation of VQ, subsets ζ_1 and ζ_2 are merged to form a single codebook for encoding. The operation of VQ inherently introduces additional high frequency components into the decoded image in the form of discontinuities at every block boundary. ζ_3 is defined to eliminate these discontinuities.

Mathematically, the operation of VQ after classification into shade and edge vectors can be expressed as:

$$\begin{aligned} &\widehat{b} = (P_s + P_e)\, b = \textstyle\sum_{i=0}^{N/n} \sum_{j=0}^{N/n} \left[P_1 f^{(a)}(x - ni, y - nj) + P_2 f^{(a)}(x - ni, y - nj) \right] \\ &\text{where} \qquad P_s\, b = \textstyle\sum_{i=0}^{N/n} \sum_{j=0}^{N/n} P_1 f^{(a)}(x - ni, y - nj) \\ &\text{and} \qquad P_e\, b = \textstyle\sum_{i=0}^{N/n} \sum_{j=0}^{N/n} P_2 f^{(a)}(x - ni, y - nj) \end{aligned} \tag{5.17}$$

where $\widehat{b}$ is the decoded image composed of ordered image blocks which are projections of the vectors in either ζ_1 or ζ_2 at the corresponding positions in the original image.

$P_s b$ is the VQ'ed image with all shade blocks reconstructed and edge blocks masked, while $P_e b$ is the VQ'ed image with all edge blocks reconstructed and shade blocks masked.

5.5.3 The Algorithm

Image restoration at the decoder is performed with an iterative algorithm which amounts to a repeated application of the projections defined in the previous section.

Step 1: The VQ'ed image is reconstructed at the decoder with shade and edge codebooks identical to those at the encoder, which is represented as $\widehat{b} = \widehat{b}_0$

Step 2: The reconstructed image is then partially smoothed (using a low pass filter) to reduce the "blocking effect" inherent in the operation of VQ. This projection operation is symbolically equivalent to $P_3 \widehat{b}$, a projection of the VQ'ed image onto its low frequency subband.

Step 3: The resulting image is then tested and corrected for any pixel intensity limit violations, i.e., $P_4 P_3 \widehat{b}$.

Step 4: The smoothed image is reprojected back onto the set $\zeta_1 \cup \zeta_2$, but with a modification in the projection operation. All shade blocks are replaced with their corresponding unsmoothed versions, while the edge blocks are left unreplaced, $P_{NL} P_4 P_3 \widehat{b}$. Thus, the sequence of projection operations depends upon the space (V_1 or V_2), to which the vector in question belongs to.

A mathematically compact notation for one iteration of the algorithm can then be given by

$$\begin{aligned} \hat{b}_{k+1} &= P_{NL} P_4 P_3 \, \hat{b}_k \\ \hat{b}_0 &= (P_s + P_e)\, b \end{aligned} \tag{5.18}$$

The numerical implementation of the nonlinear projection operation is realized by defining an indicator function in the spatial domain as follows:

$$\Gamma = \begin{cases} 1 & \text{if } \widehat{f_0}^{(n)} = P_1 f^{(n)} \\ 0 & \text{otherwise} \end{cases} \tag{5.19}$$

where $f^{(n)}$ is the *nth* vector in the original image and $\widehat{f_0}^{(n)}$ is the corresponding decoded vector. The mathematical expression for image restoration is then given as below:

$$\widehat{f_{k+1}} = \Gamma \, \widehat{f_0} + (1 - \Gamma) \, \mathfrak{F}^{-1} \left[\hat{B}_k \left[u, v \right] \ (\chi_c) \right] \tag{5.20}$$

where $\widehat{f_0}$ is the vector from the VQ'ed image, $\widehat{f_k}$ is the vector from the k^{th} iteration, $\mathfrak{F}^{-1}$ is the inverse Fourier transform operator and $\hat{B}_k \left[u, v \right] \chi_c$ is the lowpassed Fourier transform of the entire image after k iterations. The following assumptions are inherent in the interpretation of (5.20). All higher frequency components are localized exclusively to the edge blocks. Shade blocks contain negligible amounts of high frequency components, and so, are unaffected by the restoration algorithm. Figure 5.5 is a schematic representation of the algorithm.

5.5.4 Determination of "$p_{\max}$ " in the Recovery Algorithm

The sole objective of the image recovery approach is to provide subjectively pleasing images. Objective measures (ex. MMSE) cannot be used for $p_{\max}$ determination in this case, because the decoder does not have an *apriori* knowledge of the

original image. Hence, a purely subjective criterion is used in order to determine the number of iterations the decoded image is passed through. This is in accordance with the common practice used in the field of image compression. A quantity known as the mean opinion score (MOS, described in detail in Chapter 7), is obtained for several test images after every iteration. The iteration number at which the MOS achieves the best value is designated as the $p_{\max}$.

Two sets of experiments were performed to determine the effectiveness of the recovery process.

1. The first approach to image recovery was with a single low resolution codebook but no high resolution codebook. The image recovery is achieved with just a lowpass filtering of the decoded image followed by shade vector replacement. The $p_{\max}$ for this approach was 12 iterations.

2. The second approach was with two edge codebooks, a low resolution and a high resolution edge codebook. The image recovery in this case, is achieved both by the lowpass filtering operation and the requantization of the edge vectors to a better codebook. The $p_{\max}$ with this approach was 2 iterations.

5.5.5 The Lowpass Filter

A non-causal lowpass filter is used in the recovery operation. The definition of the filter is such that its impulse response is the prediction/estimation coefficients used in predicting edge block pixels based on its surroundings. Let each pixel in the edge block, $x(i,j)$, be estimated as a linear combination of 48 surrounding pixels:

$$x(i,j) = \sum_{m=-3}^{3} \sum_{n=-3}^{3} a(m,n) \, . \, x(i-m,\, j-n) \qquad m \neq 0,\ n \neq 0 \qquad (5.21)$$

It should be noted that each pixel in an edge block is estimated from a different set of pixels depending on the position of that pixel within the edge block. By the LMMSE approach, the estimation coefficients can be calculated as a set of Weiner-Hopf equations as shown below

$$\underline{a} = R^{-1}\underline{C} \tag{5.22}$$

where A, R and C are represented as follows:

$$\underline{a} = \begin{bmatrix} a(-3,-3) & a(-3,-2) & \cdots & a(3,3) \end{bmatrix}^T \tag{5.23}$$

$$R = \begin{bmatrix} Corr(x_{-3,-3}, x_{-3,-3}) & Corr(x_{-3,-3}, x_{-3,-2}) & \cdots & Corr(x_{-3,-3}, x_{3,3}) \\ Corr(x_{-3,-2}, x_{-3,-3}) & Corr(x_{-3,-2}, x_{-3,-2}) & \cdots & Corr(x_{-3,-2}, x_{3,3}) \\ Corr(x_{-3,-1}, x_{-3,-3}) & Corr(x_{-3,-1}, x_{-3,-2}) & \cdots & Corr(x_{-3,-1}, x_{3,3}) \\ Corr(x_{-3,0}, x_{-3,-3}) & Corr(x_{-3,0}, x_{-3,-2}) & \cdots & Corr(x_{-3,0}, x_{3,3}) \\ \vdots & \vdots & \vdots & \vdots \\ Corr(x_{3,1}, x_{-3,-3}) & Corr(x_{3,1}, x_{-3,-2}) & \cdots & Corr(x_{3,1}, x_{3,3}) \\ Corr(x_{3,2}, x_{-3,-3}) & Corr(x_{3,2}, x_{-3,-2}) & \cdots & Corr(x_{+3,2}, x_{3,3}) \\ Corr(x_{3,3}, x_{-3,-3}) & Corr(x_{3,3}, x_{-3,-2}) & \cdots & Corr(x_{3,3}, x_{3,3}) \end{bmatrix} \tag{5.24}$$

$$\underline{C} = \begin{bmatrix} Corr(x_{0,0}, x_{-3,-3}) & Corr(x_{0,0}, x_{-3,-3}) & \cdots & Corr(x_{0,0}, x_{3,3}) \end{bmatrix}^T \tag{5.25}$$

where $Corr$ is the crosscorrelation between the two samples x_a and x_b, i.e., $Corr(x_a, x_b) \triangleq E[x_a . x_b]$, E being the expectation operator. Figure shows the frequency support of

a generic 2-D lowpass filter whose impulse response would be a sinc function on a spatial 2-D grid.

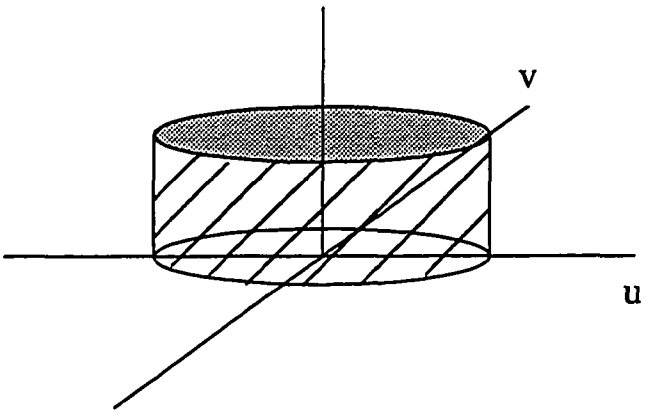


Figure 5.6: A 2-D lowpass filter

CHAPTER 6. CONVERGENCE ANALYSIS OF THE ALGORITHM

This chapter investigates the convergence of the average mean square error (MSE) distortion, as a result of the recovery algorithm. The average MSE distortion, D_l, between the original and reconstructed images, due to VQ'ing with a codebook of size l, is given by the following expression:

$$D_l = \frac{1}{N} \sum_{i=1}^{N} (x_i - x_i')^2 \tag{6.1}$$

where x_i and x_i' are the intensity values of the pixels at identical coordinates in the original and the decoded images and N is the total number of pixels in the encoded image. However, the practical criterion for measuring the fidelity of the reconstruction is a subjective measure. There is a well established empirical measure in the field of image compression. This measure is known in the literature as the Mean Opinion Score (MOS) [27]. It is described in detail in Chapter 7. In this chapter, we present a case for using the MOS as a measure of image fidelity and present the empirical relationship between the MSE (objective fidelity measure) and the MOS (subjective fidelity measure). The reasons for using MOS as a measure of fidelity is the following:

- The primary objective of image recovery is to obtain better subjective renditions. Hence, it is more appropriate to use a quantity that relates directly to human perception than MSE. The MOS has been used to measure speech and image fidelity by several authors [27].

- There is a monotonic inverse relationship between MOS and the average MSE [27]. Hence, investigating MOS is an indirect way of investigating average MSE, subject to human visual system constraints.

By the definition, a reconstruction that is indistinguishable from the original would be a MOS value of 5, while the worst performance would have a MOS value of 0. An ideal recovery process would result in an asymptotic convergence of MSE (i.e, D_l) to the smallest possible value and the MOS to a value as close to 5 as possible. In the remainder of this chapter, we investigate the convergence property of the algorithm based on the mean square error (MSE). However, it has been established that MSE is a very bad criterion to measure subjective fidelity. Even as this dissertation is being written, researchers are investigating several new expressions for measuring perceptual distortion based on a model of human visual processing [25].

The convergence of the algorithm depends on the convergence characteristics of two operations, viz., the VQ and the lowpass filtering. The convergence properties of VQ can be determined by considering the operation to be a reasonable wanderer. The definition of a reasonable wanderer is given below [23].

Definition 12 *A mapping $T : V \rightarrow V$ is said to be a reasonable wanderer if for every $\underline{x} \in V$*

$$\sum_{n=0}^{\infty} \left\| T^n \underline{x} - T^{n+1} \underline{x} \right\|^2 < \infty$$

Definition 13 *A sequence of vectors $x_n \in V$ is said to be weakly convergent to $f \in V$, iff*

$$\langle (x_n - f), y \rangle \rightarrow 0 \quad as \quad n \rightarrow \infty, \qquad for\ every\ fixed\ y \in V$$

and strongly convergent to $f \in V$, iff

$$\|x_n - f\| \to 0 \quad as \;\; n \to \infty$$

Generally, strong convergence always implies weak convergence. However, in a finite-dimensional space the converse implication is also valid [28].

Theorem 14 *Let $T : V \to V$ be an asymptotically regular non-expansive mapping with closed convex domain $V \subset H$ (a Hilbert space), whose set of fixed points $\mathcal{T} \subset V$ is nonempty. Then for any $\underline{x} \in V$, the sequence $T^n\underline{x}$ is weakly convergent to an element of $\mathcal{T}$.*

A full proof for this theorem can be found in [23]. The operation of VQ satisfies both the definition and the theorem completely. It is a non-expansive mapping within the limits of the largest codebook. The 16-D integer vector space, V, is a closed convex set and the finite set of fixed points $\mathcal{T}$ is the set of representative vectors in the codebook. Thus, the operation of VQ is indeed weakly convergent. The three main operations in the recovery algorithm are: (1) Lowpass filtering P_5, (2) Replacement of the shade blocks and (3) Requantization of the edge blocks.

$$\begin{aligned} \widehat{b}_{k+1} &= (P_s + P_e)P_5\, b_k \\ \widehat{b}_{k+1} &= P_sP_5b_k + P_eP_5b_k \end{aligned} \tag{6.2}$$

In the above equation, (P_sP_5) operate only on the shade blocks while (P_eP_5) only operate on the edge blocks. Both P_s and P_e are weakly convergent as far as the operations are performed in the R^{16} space. So, the convergence of the composite operations P_sP_5 and P_eP_5 is dependent on the properties of the filter P_5. In the

remainder of this chapter we look at the conditions required for both convergence and divergence of the algorithm.

Let $\underline{x}$ be an edge vector from the original image (mean removed) which is to be VQ'ed by the edge codebook. Obviously, $\underline{x}$ is a vector in the 16 -D integer vector space R^{16}, with individual elements being $x_0, x_2, \ldots x_{15}$. The vector is estimated based on the intensity values of surrounding pixels. Effectively, each element in the edge vector is estimated with a different set of coefficients and pixels. Mathematically,

$$\hat{\underline{x}} = \widetilde{X}\underline{a} \tag{6.3}$$

where $\hat{\underline{x}}$ is the estimate of the original edge vector, $\widetilde{X}$ is a matrix consisting of the surrounding pixels for each element in the edgevector $\underline{x}$ and $\underline{a}$ is a vector of the prediction coefficients.

At the decoder, any edge vector $\underline{x}$ can be estimated as shown in (6.4), where $\underline{x}'$ is the decoded version of the edge vector $\underline{x}$ in the encoded image, $\widetilde{X}$ is a matrix of surrounding pixels and $\underline{e}$ an error vector,

$$\underline{x}' = \widetilde{X}\underline{a} + \underline{e} \tag{6.4}$$

The coefficient matrix $\underline{a}$ is obtained using a set of original images for pixel data. At the decoder, however, the same set of coefficients are used with quantized pixel data. Thus, there is some amount of mismatch between the data used in the calculation of the predictor coefficients and the data used for predictions. The assumption here is that the surrounding shade vectors are reproduced with sufficient fidelity at the decoder that the coefficient vector $\underline{a}$ is valid with the distorted shade pixels. The prediction of a single edge vector, on a pixel-by-pixel basis, in terms of the surrounding

pixels is represented mathematically as follows,

$$\begin{aligned} \widehat{x}_i &= x^i_{-3,-3}a_{-3,-3} + x^i_{3,-2}a_{-3,-2} + \ldots\ldots + x^i_{3,2}a_{3,2} + x^i_{3,3}a_{3,3} \qquad (6.5) \\ i &= 0, 1, 2, \ldots\ldots, 15 \end{aligned}$$

where x_i is the i^{th} pixel in the edge vector that is being predicted and $x^i_{m,n}$, $(m = -3, -2, ..., 2, 3; n = -3, -2, ..., 2, 3)$ is the pixel that is m rows and n columns removed from the predicted pixel. In matrix representation, the prediction equation would be as follows,

$$\begin{bmatrix} \widehat{x}_0 \\ \widehat{x}_1 \\ \widehat{x}_2 \\ \widehat{x}_3 \\ . \\ . \\ . \\ \widehat{x}_{15} \end{bmatrix} = \begin{bmatrix} x^0_{-3,-3} & x^0_{-3,-2} & \cdots & x^0_{3,3} \\ x^1_{-3,-3} & x^1_{-3,-2} & \cdots & x^1_{3,3} \\ x^2_{-3,-3} & x^2_{-3,-3} & \cdots & x^2_{3,3} \\ x^3_{-3,-3} & x^3_{-3,-3} & \cdots & x^3_{3,3} \\ . & . & \cdots & . \\ . & . & \cdots & . \\ . & . & \cdots & . \\ x^{15}_{-3,-3} & x^{15}_{-3,-3} & \cdots & x^{15}_{3,3} \end{bmatrix} \begin{bmatrix} a_{-3,-3} \\ a_{-3,-2} \\ a_{-3,-1} \\ a_{-3,0} \\ . \\ . \\ . \\ a_{3,3} \end{bmatrix} \qquad (6.6)$$

Let us define two sets of edge codebooks, a low resolution edge codebook with 10 codevectors and a high resolution edge codebook with 256 edge codevectors. The average MSE distortion, between the original and reconstructed images, due to VQ'ing with a codebook of size l, is given by (6.1). So, the average distortions between the original and reconstructed images due to VQ'ing with codebooks of size 10 and 256 will be D_{10} and D_{256} respectively. It has been shown by Gray and Nasrabadi [19, 21] that the average distortion is inversely proportional to the size of the codebook. For example, in the experiments conducted during this study the average distortion D

over all the edge vectors when VQ'ed with an edge codebook of size 10 was found to be 3115.16 and when VQ'ed with an edge codebook of size 256 it was 1356.67. Thus, it is logical to assume that $D_{10} > D_{256}$.

The effectiveness of the edge recovery algorithm depends on two factors:

1. The distribution of the representative vectors throughout the vector space (i.e, the distance between any representative vector and its partition boundary).

2. The amenability of the edge blocks/vectors to estimation based on surrounding blocks.

In the ideal case of uniformly distributed codebook vectors and a constant distance between any representative vector and its partition boundaries, the following assertion can be made. For an asymptotic bounded convergence, it is essential that the norm of the prediction error ($\|\underline{e}\|$) in (6.4), on an average, be smaller than the quantization noise at every iteration. This assertion is explained with reference to Figure 6.1, which shows an enlarged view of a small portion of the edge vector space with the larger (high resolution) codebook superimposed. $\widetilde{X}\underline{a}$, represented by the long dashed arrow, denotes the predicted version of a particular edgeblock (vector), and $\underline{e}$ (represented by the short dashed arrow), the error between the actual vector and its predicted version. The input edge vector to be encoded is represented by the solid arrow and denoted by $\widetilde{X}\underline{a}+\underline{e}$. Due to the VQ operation with a vestigial edge codebook, the error due to encoding will be e_{vq}. Invariably, $\left\|\underline{e}_{vq}\right\| >> \|\underline{e}\|$ because of the extremely small size (and hence a very low resolution) of the edge codebook.

The optimum representation for this input edge vector would be the codevector $\underline{m}$. The codevectors $\underline{b}, \underline{c}, \underline{d}, \underline{f}, \underline{g}, \underline{h}, \underline{i}, \underline{j}$ etc. are the immediate neighbors of $\underline{m}$. Since, at

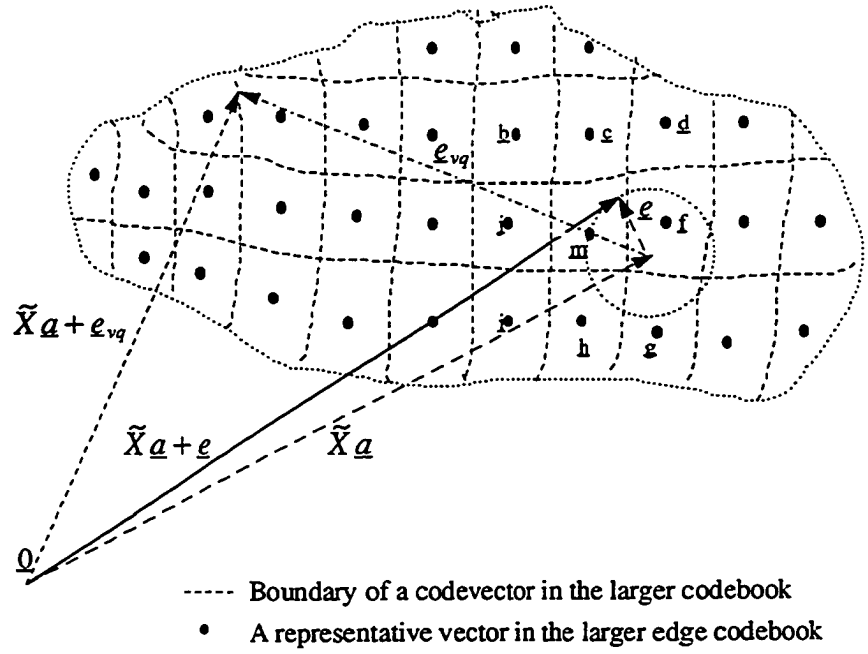


Figure 6.1: A diagrammatic representation of the prediction operation for optimal recovery operation

each iteration the shade regions remain unchanged the prediction vector $\widetilde{X}\underline{a}$, which is based on shade pixels remains constant. The recovery algorithm serves to iteratively reduce the norm of the error vector, $\left\|\underline{e}_{vq}\right\|$, and hence the error itself. It can be seen that by applying the algorithm infinitely we will force e_{vq} to zero and hence the edge vector approaches $\widetilde{X}\underline{a}$ asymptotically. Forcing $\left\|\underline{e}_{vq}\right\| \rightarrow 0$ is counterproductive, because in doing so, we force the edge vector to the cell represented by $\underline{f}$ rather than $\underline{m}$. Thus, the aim of the recovery should be to make $\underline{e}_{vq}$ approach $\underline{e}$, rather than to force $\underline{e}_{vq}$ to zero.

If the norm of $\underline{e}$ is smaller than the quantization error in the partition, the asymptotic worst case error between the original and the recovered images can be

upper bounded by the distance between any two adjacent codebook vectors (eg., distance between $\underline{m}$ and $\underline{j}$). This is because every single edge vector in the image is guaranteed to be asymptotically mapped to the best possible codevector or the one in the immediate neighborhood. Correspondingly, the graph of the MSE can be upper bounded and is depicted as shown in Figure 6.2. The corresponding values of Mean Opinion Score (MOS) is shown in Figure 6.3.

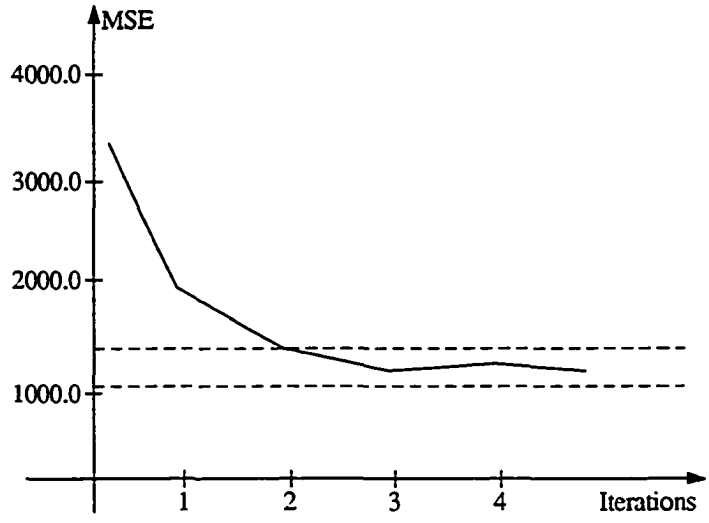


Figure 6.2: Mean square error v/s number of iterations under near optimal conditions

However, it is impossible to obtain a set of prediction coefficients that are optimum to all edge vectors, as described above. This means the norm of $\underline{e}$ cannot be guaranteed to be less than the quantization error for all predicted edge blocks. Under such suboptimal conditions, the graph of MSE over the number of iterations can be depicted as shown in Figure 6.4 and the graph of Mean Opinion Score (MOS) over the number of iterations as in Figure 6.5.

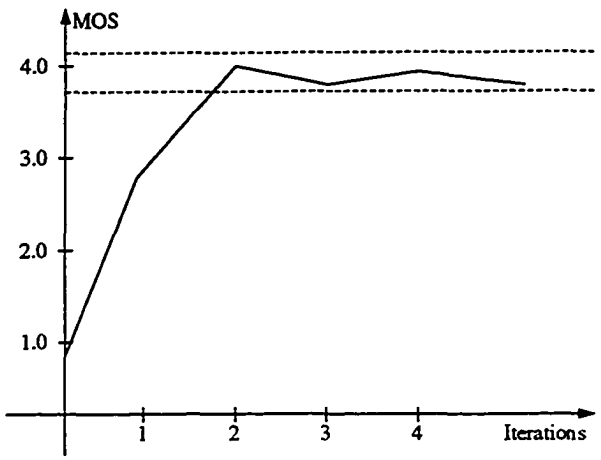


Figure 6.3: The MOS v/s number of iterations under near optimum conditions

Let $\mathfrak{E}_k$ be the average error at k^{th} iteration over all the edge blocks/vectors in an image. Let D_L be the average error when the image in question is VQ'ed with a codebook of size L. Then for an optimum performance,

$$\mathfrak{E}_k = D_L \qquad k \text{ is the iteration parameter} \tag{6.7}$$

$$\textit{where} \quad \mathfrak{E}_k \triangleq \frac{\sum_{i=1}^{N} \|\underline{e}_{ik}\|}{N} \qquad 1, 2, ..., N \quad \text{all edge vectors in the image}$$

Compressing the image using a vector quantizer with a small codebook results in a huge quantization error. If $\underline{e}_{i0}$ is the quantization error of the i^{th} edge vector at 0^{th} iteration (due to the VQ operation with the smaller edge codebook), $\mathfrak{E}$, the average error due to prediction alone, and $\mathfrak{E}_0$ is the average error due to prediction *and* VQ with the smaller codebook (i.e, at 0^{th} iteration),

$$\mathfrak{E}_0 >> \mathfrak{E} \tag{6.8}$$

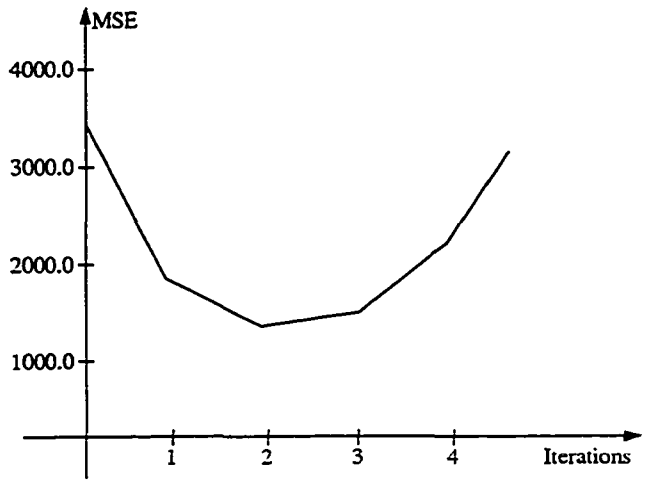


Figure 6.4: Mean-square error v/s number of iterations under suboptimal conditions

where $\mathfrak{E}_0 = \sum_{i=1}^{N} \underline{e}_{i0}/N$. The edge vectors are recovered using the CP based algorithm.

Figure 6.5 depicts the edge vector space with both the smaller and the larger edge codebooks superimposed within the space.

Figure 6.7 is an enlarged view of a very small portion of the edge vector space (encircled portion of Figure 6.6). It is a schematic depiction of the edge vector recovery process. The original vector consists of a set of pixels calculated as an estimation based on neighboring pixels (predominantly shade), given by $\widetilde{X}_{i0}\underline{a}$ and a corresponding error value $\underline{e}_i$ (i.e, $\widetilde{X}_{i0}\underline{a} + \underline{e}_i$). It is represented by the only solid arrow in the figure. When the vector is VQ'ed with a small codebook and hence a large average distortion, the norm of error vector increases to $\|\underline{e}_{i0}\|$.

At the decoder, edge recovery is to be accomplished by using the following *apriori information*,

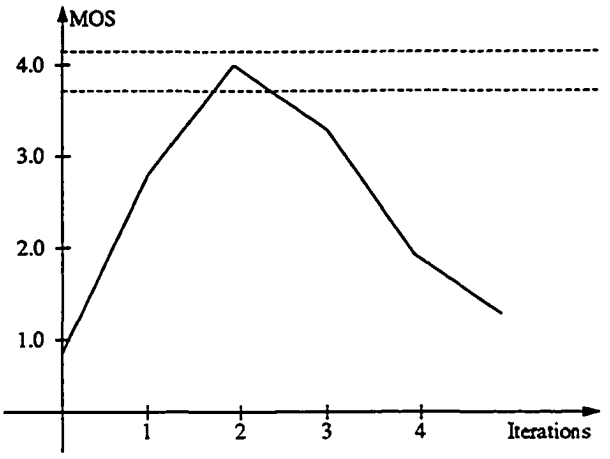
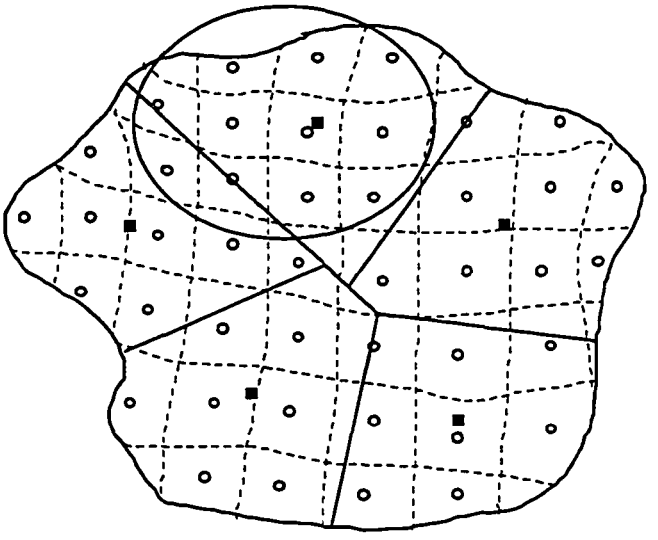


Figure 6.5: TheMOS v/s number of iterations under suboptimal conditions

1. The predicted version of the edge vector in question, based on the surrounding shade pixels.

2. The codevector from the smaller (hence low resolution) edge codebook to which the edge vector was mapped to at the encoder.

It is assumed that the predictors are optimum in the MSE sense and the individual prediction errors are essentially zero mean random values. The norm of these errors is bounded by the distance between the boundary and the representative vector in the smaller edge codebook to which the vectors are mapped to. It is required to requantize these individual edge blocks onto a larger (i.e, a higher resolution) codebook and hence achieve better subjective fidelity. A simple requantization with the larger codebook cannot achieve higher resolution because:



— Partition in the smaller edge codebook
---- Partitons in the larger edge codebook
■ Representative vectors in the smaller edge codebook
o Representative vectors in the larger edge codebook

Figure 6.6: Edge vector space with both the low and the high resolution codebooks superimposed

- Every edge block mapped to a vector in the smaller edge codebook will be remapped, enmasse, onto a vector in the larger edge codebook. This essentially is equivalent to a simple translation of a set of vectors that were all mapped to one representative vector in the smaller edge codebook.

In order to have an increase in resolution, each of the edge block/vector which is mapped onto a single representative vector in the smaller edge codebook, has to be perturbed in the "right" direction. Since, in images there is a huge amount of redundancy between adjacent pixels, this perturbation in the "right" direction can

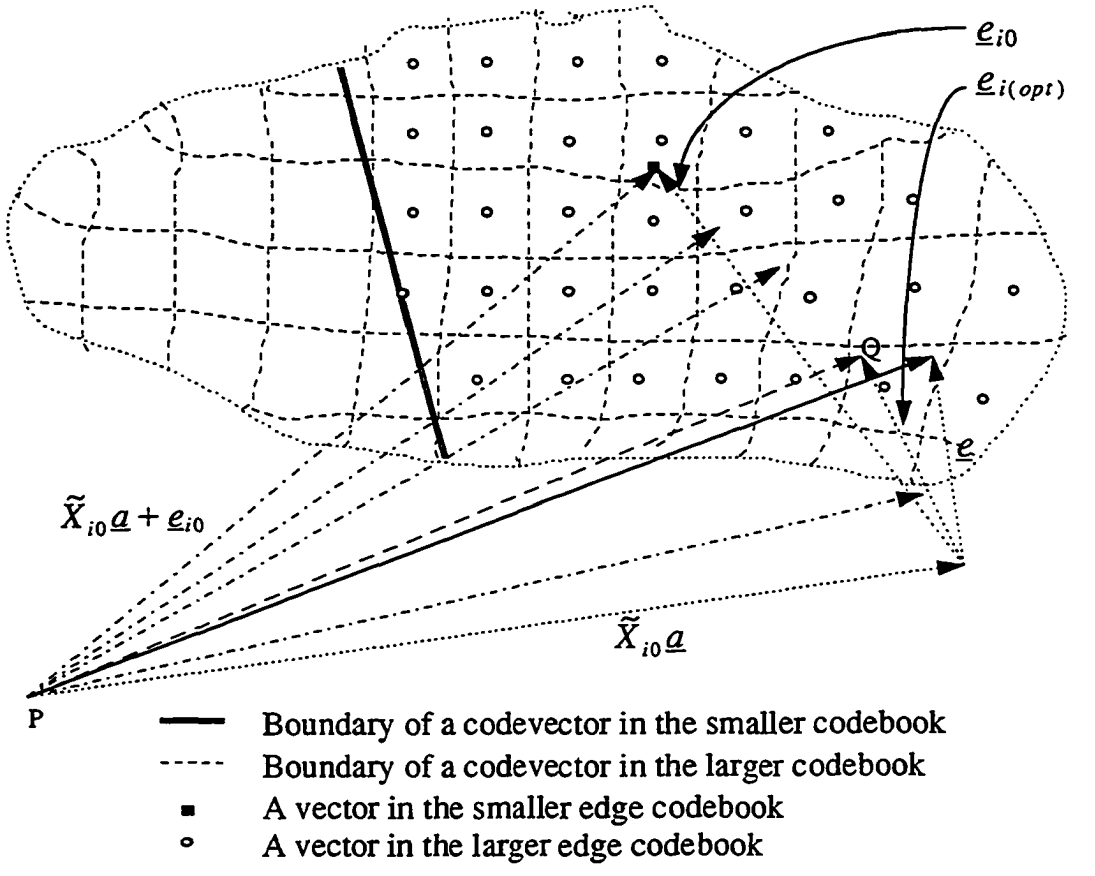


Figure 6.7: A diagrammatic representation of edge block recovery

be achieved by a lowpass filter wherein the individual pixels in each edge vector is a function of the surrounding pixels, predominantly shade. Thus, the lowpass filter in the iterative operation serves to perturb every edge block in the right direction from its mapped value. The filtering operation at each iteration is as follows:

1. Smooth the image as a whole, partition the filtered image back into individual blocks.

2. Replace all shade blocks with their corresponding unfiltered versions, but re-quantize the *filtered edge blocks* with the higher resolution edge codebook.

The aforementioned operations result in a predicted vector ($\widetilde{X}_{i0\underline{a}}$ in Figure 6.6) that is relatively a constant at every iteration. If the operation of smoothing followed by the shade vector replacement is performed infinitely, the pixels in every edge vector will asymptotically approach intensity values equal to their corresponding surrounding shade pixels, thereby losing the actual edge information. This is depicted in Figure 6.6, where the vector ($\widetilde{X}_{i0\underline{a}} + \underline{e}_{i0}$) makes an iterative journey towards the vector $\widetilde{X}_{i0\underline{a}}$. In doing so, the edge vector may shoot beyond the optimal point (from a subjective viewpoint), represented by $\widetilde{X}_{i0\underline{a}} + e_{i(opt)}$, and depicted by the line PQ. The filtering and prediction operation on an arbitrary vector, at the 1^{st} and the n^{th} iteration is given below.

Let $\underline{\hat{x}}_{i0}$, be the VQ'ed version of the i^{th} edge vector at 0^{th} iteration. Let $\widetilde{X}_{i0}$ represent the surrounding pixels used in predicting this vector and $\underline{e}_{i0}$, the error vector. The low-pass filtering LP, (defined by the filter whose impulse response is represented by the template of the predictor coefficients) of the i^{th} edge vector at the 1^{st} iteration decoding can be represented as shown below:

$$\begin{aligned} LP\left[\underline{\hat{x}}_{i0}\right] &= LP\left[\widetilde{X}_{i0\underline{a}} + \underline{e}_{i0}\right] \\ VQ\left(LP\left[\underline{\hat{x}}_{i0}\right]\right) &= \widetilde{X}_{i0\underline{a}} + \underline{\delta}_{i0} + LP\left[\underline{e}_{i0}\right] \\ VQ\left(LP\left[\underline{\hat{x}}_{i0}\right]\right) &= \underline{\hat{x}}_{i1} \\ \underline{\hat{x}}_{i1} &= \widetilde{X}_{i1\underline{a}} + \underline{e}_{i1} \end{aligned} \tag{6.9}$$

where $\underline{\hat{x}}_{i1}$ is the VQ'ed version of the i^{th} edge vector at 1^{st} iteration, $\widetilde{X}_{i1}$ represents the surrounding pixels used in predicting this vector and $\underline{e}_{i1}$, the error vector after

1^{st} iteration. The lowpass filtering has little effect on the estimator term because all the surrounding shade vectors are replaced at each iteration. The small changes in the estimator term is due to two reasons:

- The requantization of the lowpassed edge block/vector will displace it to the nearest vector in the high resolution (larger) codebook. This displacement is dependent on the number and distribution of the vectors in the larger codebook.
- Some of the pixels used in the estimation will be either from within the same edge block or occasionally from other edge blocks. Such pixels will be different at every iteration.

The small differences in the estimator term due to these two factors, is represented by the term $\underline{\delta}_{i0}$. This can be extended, term for term, to the filtering operation at any iteration n as follows,

$$\begin{aligned} LP\left[\underline{\widehat{x}}_{i(n-1)}\right] &= LP\left[\widetilde{X}_{i(n-1)}\underline{a} + \underline{e}_{i(n-1)}\right] \\ VQ\left(LP\left[\underline{\widehat{x}}_{i(n-1)}\right]\right) &= \widetilde{X}_{i(n-1)}\underline{a} + \underline{\delta}_{i(n-1)} + LP\left[\underline{e}_{i(n-1)}\right] \\ VQ\left(LP\left[\underline{\widehat{x}}_{i(n-1)}\right]\right) &= \underline{\widehat{x}}_{in} \\ \underline{\widehat{x}}_{in} &= \widetilde{X}_{in}\underline{a} + \underline{e}_{in} \end{aligned} \tag{6.10}$$

When the size of the edge codebooks are small and hence average quantization errors are very large, $\left\|\underline{e}_{i(n-1)}\right\| >> \left\|\underline{\delta}_{i(n-1)}\right\|$. In this case, the effect of $LP[\underline{e}_{i(n-1)}]$ dominates over $\underline{\delta}_{i(n-1)}$. At larger codebook sizes i.e, with smaller average quantization errors, $\left\|LP\left[\underline{e}_{i(n-1)}\right]\right\|$ and $\left\|\underline{\delta}_{i(n-1)}\right\|$ are comparable and so cannot be ignored.

Case 1: $\left\|LP\left[\underline{e}_{i(n-1)}\right]\right\| >> \left\|\underline{\delta}_{i(n-1)}\right\|$

In this case, where effects of $\underline{\delta}_{i(n-1)}$ is negligible, convergence can be guaranteed by the following two conditions:

$$\|\underline{e}_{in}\| \leq \left\|\underline{e}_{i(n-1)}\right\| \tag{6.11}$$

$$\Delta_n \leq \left\|e_{in} - e_{i(n-1)}\right\| \leq 2\Delta_n \tag{6.12}$$

where $\underline{e}_{i(n-1)}$ and $\underline{e}_{in}$ is the error vector at the $(n-1)^{th}$ and n^{th} iteration respectively. Δ_n is the distance between the codebook vector and its partition to which $\widetilde{X}_{in}\underline{a}+\underline{e}_{in}$ is requantized.

The first condition can be guaranteed by Parseval's theorem since $\underline{e}_n = LP[\underline{e}_{n-1}]$ and hence the error vectors should have smaller energy content at each iteration. This condition dictates the extent of low pass filtering (magnitude of perturbation) of the edge vectors at each iteration.

Case 2: $\left\|LP\left[\underline{e}_{i(n-1)}\right]\right\|$comparable with $\left\|\underline{\delta}_{i(n-1)}\right\|$

In this case, the codebooks are fine enough and $\underline{\delta}_{i(n-1)}$ large enough so that either or both of (6.11) and (6.12) do not hold, resulting in a divergence of the algorithm.

Thus, the average distortion suffered by an image under low resolution encoding and subsequent recovery is lower bounded by the average distortion suffered under high resolution encoding. In practice, this bound is seldom achievable because it implies that every single edge vector reaches its optimum codevector at the same iteration. Decisions as to the number of iterations are made based on subjective improvements in the image. It is not practical to monitor the recovery of individual edge blocks separately.

CHAPTER 7. EXPERIMENTAL RESULTS

7.1 Experimental Results and Performance

The convex projection recovery algorithm was tested on several photographic images from the archives of USC. The dimensions of all images were 512×512 with gray levels ranging from 0 to 255 corresponding to 8-bit amplitude resolution. Table 7.1 gives the statistical properties of all the images used in the experiments.

Table 7.1. Statistics of a few typical photographic images

Statistics	peppers	Lenna	sailboat	baboon	couple
Min. Intensity	16	37	18	16	0
Max. Intensity	211	244	221	214	255
Sample Mean	119.26	140.54	123.55	127.31	121.89
Sample Variance	2139.86	1578.06	3171.39	1320.72	1967.47
Entropy	7.374	7.042	7.266	7.139	7.058

The images "peppers", "sailboat" and "baboon" were used as the training set images while "Lenna" and "couple" were the test images. The training images were partitioned into block sizes of 4×4. Each of the blocks were classified into either a

shade or an edge block. All shade and edge blocks were collected into two separate files. Table 7.2 gives the number of shade and edge blocks in each image. The total number of 4×4 blocks in each image is 16384.

Table 7.2. Number of shade and edge blocks in the experimental images

Statistics	peppers	Lenna	sailboat	baboon	couple
Shade Blocks	13301	13658	13539	13642	13180
Edge Blocks	3083	2726	2845	2742	3204

A shade codebook of size 24 was generated and two edge codebooks of size 8 and 232 respectively were generated using the well known Linde, Buzo, Gray (LBG) algorithm [26]. The encoder codebook was a concatenation of shade codebook and the smaller edge codebook of size 10. The decoder consisted of a copy of the encoder codebook and several edge codebooks at varying resolutions. Edge recovery was based on an iterative CP algorithm which used monotonically higher resolution codebooks at each iteration. Figure 7.1 and Figure 7.2 show small areas of the original images of 'Lenna' and 'peppers' at 8 bpp. Figure 7.3 and Figure 7.4 show the same areas of 'Lenna' image coded at 0.254 bpp before and after the application of the restoration algorithm respectively. Figure 7.5 and Figure 7.6 show the corresponding areas of "peppers" image at 0.259 bpp before and after the application of the restoration algorithm respectively. Figure 7.7 and Figure 7.8 shows a magnified view of a small portion of the peppers image before and after the application of the recovery algorithm.

Figure 7.1: A portion of "lenna" image (8 bpp)

7.2 Subjective and Objective Distortions

Fidelity of signal (image) reconstructions are generally measured using two distortion measures: (1) Objective and (2) Subjective distortions. The objective distortion is an average measure of the error suffered by each individual sample (pixel) due to the compression algorithm. The most widely used measure for this purpose is the peak signal-to-noise ratio (PSNR) which uses the mean square error (MSE) criterion. The expression for the PSNR is as shown below:

$$PSNR = 10 \log_{10} \frac{(Max\ pixel\ value\ in\ the\ image)^2}{E[r(i,j)^2]} \tag{7.1}$$

Figure 7.2: A portion of "peppers" image (8 bpp)

where $r(i,j)$ is the difference between the original and the reconstructed pixel intesity values calculated over the entire image.

It has been well documented in image compression research that distortion measures like the mean squared error (MSE) are inaccurate in predicting perceptual distortion [25]. Image processing literature has practical examples where two decoded images exhibiting very similar MSEs show marked difference subjectively. However, the development of a tangible expression to quantify subjective fidelity is still an ongoing research. In this study, the MMSE criterion is used to quantify the objective fidelity. The improvement in the peak signal-to-noise ratio (PSNR) gained by the recovery algorithm was upper bounded by 1 dB. Table 7.3 gives the PSNR values of

Figure 7.3: 'Staircase effect' due to vestigial edge encoding of the edge vectors (lenna - 0.254 bpp)

some of the images used in the experiments.

For quantifying subjective fidelity, another widely used definition is that of the Mean Opinion Score (MOS) [27]. This score is evaluated by averaging the individual opinion scores for each of the images under consideration. This definition has been adapted into image coding primarily from the CCITT recommendations for speech coding evaluations. CCITT recommends the use of a five point scale {excellent, good, fair, poor, bad} which is numerically mapped to the decimal {5, 4, 3, 2, 1} scale.

Figure 7.4: Multiple codebook decoding using the projection algorithm (lenna - 0.254 bpp)

Table 7.3 Objective performance of the algorithm on experimental images

Images	Comp. Ratio (bits/pixel)	PSNR
Peppers	0.259	34.23
Lenna	0.254	31.82
Sailboat	0.248	33.67
Baboon	0.267	32.03
Couple	0.253	31.01

Figure 7.5: 'Staircase effect' due to vestigial edge vector encoding (peppers - 0.259 bpp)

There are several types of empirical subjective tests defined both in speech and image coding areas. Two of the oft encountered ones are: Absolute Category Rating (ACR) and Degradation Category Rating (DCR). In this study we use the former method, since we are only interested in comparing two realizations of an image, for their cumulative effect on the subject. Table 7.4 gives the ACR-MOS scores for the aforementioned images.

Figure 7.6: Multiple codebook decoding using the projection algorithm (peppers - 0.259 bpp)

Table 7.4 Mean Opinion Scores from the ACR test on the experimental images

Images	No. of Subjects	Before Recovery	After Recovery
peppers	10	3.23	3.76
Lenna	10	3.31	3.92
sailboat	8	3.56	3.86
baboon	8	3.74	3.83
couple	8	3.26	3.81

Figure 7.7: A small portion of the "peppers" image expanded (before the application of the recovery algorithm)

7.3 Robustness of the Algorithm

One of the benchmarks for practical implementation of any algorithm is its robustness. This is a measure of effectiveness of the algorithm when its parameters are mismatched with respect to the input signal. Two such parameters in this algorithm are the VQ edge codebook and the lowpass filter. In order to test for the robustness the following two case studies were performed.

7.3.1 Case 1: Robustness to Codebook Variations

The algorithm was applied with the high resolution edge codebooks from two different sources. One of the edge codebooks was trained on the training set of images.

Figure 7.8: A small portion of the "peppers" image expanded (after the application of the recovery algorithm)

The other was obtained from another researcher who was working in the same area and had his own edge codebook generated using the same classification algorithm. The edge recovery in the images with either codebook, on an average, were as good subjectively. This is because even though the codebooks are different, each will have a vector that is close to the edge block in question (in the MSE sense). The position of those codevectors in the two codebooks may be different but the algorithm still chooses the appropriate vector.

7.3.2 Case 2: Robustness to the Lowpass Filter Parameters

In order to test for this parameter, the algorithm was applied with a simple circular lowpass filter. The region of support for the filter was made approximately

equal to that of the calculated filter. The algorithm seems to be robust as long as the passband and stopband of the two filters were approximately similar. Again very little perceptible differences between the two cases were observed after reconstruction with identical number of iterations. However, any reasonable increase and decrease of the region of support had adverse effects on the final reconstructions.

The possible reasons for this may be the following:

- An increase in the passband reduces the amount of perturbation in the edge blocks. A large increase in the passband will probably result in a perturbation so small that the edge blocks fail to cross over to adjacent partitions. Hence an iterative recovery is no longer possible.

- A decrease in the passband increases the amount of perturbation in the edge blocks. A large reduction in the passband will result in a large perturbation of the edge blocks. In that case the lowpassed edge block will overshoot its optimum codevector in the high resolution codebook and hence cannot be recovered. This is explained in detail in Chapter 6 which deals with the convergence aspects of the recovery process.

BIBLIOGRAPHY

[1] N. S. Jayant and P. M. Noll, "Digital Coding of Waveforms - Principles and Applications to Speech and Video," Prentice-Hall Inc., Englewood Cliffs, New Jersey, NJ, 1984.

[2] R. J. Clarke, "Transform Coding of Images," Academic Press, Orlando, Florida, 1985.

[3] C. L. DeVito, "Functional Analysis and Linear Operator Theory," Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

[4] S. K. Berberian, "Introduction to Hilbert Space," Oxford University Press, New York, NY, 1961.

[5] D. C. Youla, "Generalized Image restoration by the method of Alternating Projections," *IEEE Transactions on Circuits and Systems*, vol. CAS-25, pp. 29–37, October 1978.

[6] M. Sezan and H. Stark, "Image Restoration by the method of Convex Projections Part-2, Applications and Numerical Results," *IEEE Transactions on Medical Imaging*, vol. MI-1, pp. 95–101, October 1982.

[7] R. L. Stevenson, "Reduction of Coding Artifacts in Transform Image Coding," *Proceedings of ICASSP, Minneapolis*, pp. 401–404, May 1993.

[8] A. K. Katsaggelos, "Iterative Image Restoration Algorithms," *Journal of Optical Engineering*, vol. 29, pp. 735–748, July 1989.

[9] A. Zakhor, "Iterative Procedures for Reduction of Blocking Effects in Transform Image Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 91–95, March 1992.

[10] Y. Yang, N. Galatsanos and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block-dct compressed images," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 421–432, December 1993.

[11] Y. Yang and N. Galtsanos, "Edge -preserving reconstruction of images using and a divide-and- conquer strategy," *Proceedings of ICIP-94*, pp. 534–537, November 1994.

[12] K. Jonathan Su and R. M. Mersereau, "Post Processing for Artifact Reduction in JPEG-Compressed Images," *Proceedings of ICASSP, Detroit*, pp. 2363–2366, May 1995.

[13] A. K. Katsaggelos, J. Biemond, R. M. Mersereau and R. W. Schafer, "A Regularized Iterative Image Restoration Algorithm," *IEEE Transactions on Signal Processing*, vol. 39, pp. 914–929, April 1991.

[14] R. V. L. Hartley, "Transmission of Information," *The Bell System Technical Journal*, vol. 7, No. 3, pp. 535–563, July 1928.

[15] C. E. Shannon, "A Mathematical Theory of of Communications (Part I)," *The Bell System Technical Journal*, vol. 27, No. 3, pp. 379–423, July 1948.

[16] C. E. Shannon, "A Mathematical Theory of of Communications (Part II)," *The Bell System Technical Journal*, vol. 27, No. 4, pp. 653–656, October 1948.

[17] R. E. Blahut, "Principles and Practice of Information Theory," Addison -Wesley Publishing Co., Reading, Massachussetts, 1988.

[18] M. Mansuripur, "Introduction to Information Theory," Prentice-Hall. Inc., Englewood Cliffs, New Jersey, NJ, 1987.

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

[20] B. Ramamurti and A. Gersho, "Classified vector quantization of images," *IEEE Transactions on Communications*, vol. COM-34, pp. 1105–1115, November 1986.

[21] N. M. Nasrabadi, "Use of Vector Quantizers in Image Coding," *Proceedings of ICASSP, Tokyo*, pp. 125-127, May 1985.

[22] N. M. Nasrabadi and Y. Feng, "Image compression using address vector quantization," *IEEE Transactions in Communications*, vol. COM-38, December 1990.

[23] E. H. Stark, "Image Recovery: Theory and Applications," Academic Press, San Diego, CA, 1987.

[24] P. M. Clarkson and H. Stark,"Signal Processing methods for Audio, Images and Telecommunications," Academic Press, San Diego, CA, 1995.

[25] P. C. Teo and D. J. Heeger,"Perceptual Image Distortion," *SPIE Symposium on Electronic Imaging: Science and Technology*, Academic Press, San Diego, CA, 1994.

[26] Y. Linde, A. Buzo, R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, January 1980.

[27] B. S. Atal, V. Cuperman and A. Gersho, "Speech and Audio Coding for Wireless and Network Applications," Kluwer Academic Press, Boston, MA, 1993.

[28] P. R. Halmos, "Finite Dimensional Vector Spaces," Springer-Verlag Press, New York, NY, 1993.