IMAGE COMPRESSION USING SUBJECTIVE VECTOR QUANTIZATION

Ronny Quesnel

Department of Electrical Engineering

McGill University, Montréal

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Masters in Engineering

March 1992

Preface

In early September 1991, a first draft of this thesis was given to me as thesis supervisor for Mr. Ronny Quesnel. On Sunday night, 15 September 1991, Ronny had a fatal automobile accident. The topic of this thesis is one in which Ronny had unbounded interest and enthusiasm. I pledged to see that the thesis be completed to submission and found the process of preparing the thesis for submission one of considerable emotion and one which could not have been successfully completed without the assistance and constant support of Mr. Salvatore Torrente, research assistant and System Administrator of the INSL lab, and a dear friend of Ronny. I can only hope that in the revision process and in the writing of the abstract, I remained faithful to Ronny's style which touched all of us and will never be forgotten. The subject of this thesis is a difficult one, and I can only assure readers that Ronny's conclusions have been substantiated by a rather large population of Information Networks and Systems Laboratory colleagues which viewed his images and provided him with their subjective evaluations.-Dr. Salvatore D. Morgera

Abstract

The goal of this research is to improve the subjective quality of real world imagery encoded with spatial vector quantization (VQ). Improved subjective quality implies that a human perceives less visually objectionable distortion when looking at the coded images. Through study of several basic VQ schemes, the issues fundamental to achieving good subjective quality are uncovered and addressed in this work. Vector quantization is very good at reproducing quasi-uniform textures in an image, but has difficulty in reproducing abrupt changes in textures (edges) and fine detail and can cause a block effect which is subjectively annoying. A second generation coding scheme is developed which takes certain properties of the human visual system into account. A promising method which is developed utilizes omniscient finite state VQ, a new quadratic distortion measure which penalizes the misrepresentation of edges, and brightness compensation based on Steven's power law. The proposed subjective VQ is compared with several classical, first generation VQ methods.

Résumé

Cette recherche a pour but d'améliorer la qualité subjective des images encodées par quantification vectorielle. L'accroissement de la qualité subjective des images signifie que l'être humain percevra moins de distorsions désagréables en regardant les images codées. Cette ouvrage dévoile les points fondamentaux qui régissent l'obtention d'une bonne qualité subjective en présentant plusieurs méthodes de base en quantification vectorielle. La quantification vectorielle est une méthode qui donne de bons résultats dans la reproduction de textures quasi-uniformes composant une image. Mais, le codage par quantification vectorielle peut difficilement reproduire les changements brusquent dans la texture (contours) et le fin détail de l'image, et peut même causer un effet de quadrillage, dans la reproduction de l'image, qui est déplaisant à la vue. Un système de codage, de deuxième génération, est développé qui prend en considération les propriétés du système optique humain. Une méthode de codage prometeuse est développée qui utilise une approche omnisciente en quantification vectorielle par états limités, une mesure de distortion quadratique qui punit les contours érronés d'une image, avec un système compensatoire d'intensité de l'image basé sur la loi des puissances de Steven. Enfin, le système de codage par quantification vectorielle présenté dans cette ouvrage est comparé à plusieurs autres systèmes de codage de première génération, en quantification vectorielle.

Dedication

I will best remember him for the lively and friendly talks that we had together. He was always enthusiastic about discovering new things, not only about academic life, but also about life in general. I am proud to be considered his friend and colleague, and I shall keep fond memories of him forever.

Patrick Lie Chin Cheong

Ronny was a gifted student and a friend to all. We had many wonderful moments working together but one of his most endearing qualities was the pleasure and "joie de vivre" that he brought to every day life.

Michael Mastine

I will remember Ronny as a hardworking and fun-loving, individual. He was dedicated in his work and determined to constantly improve himself as an individual. On the other hand, Ronny never refused to help others who needed his assistance. You will be missed. Au revoir, mon ami.

Albert Pang

Ronny had all the qualities essential to excel in his work and in life. He always believed that teaching others helped him be a better person because he too would learn in the process. He always wanted to be the best he could be; and through this thesis Ronny will continue to share his knowledge and help others to reach further in their work... Ronny, your spirit will always be with us.

Martine Fournelle

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Image compression is essential for applications such as TV transmission, video conferencing and facsimile transmission of printed material as well as where pictures are stored in a database, such as archiving medical images, fingerprints and drawings. Now, with the increasing use of images as a communication medium, image compression techniques which offer a favorable tradeoff vis-à-vis reproduction quality, storage requirements, and computational complexity are needed. The research community is currently investing considerable time in the design of advanced image compression techniques for new and evolving applications.

Image compression techniques can be classed as *lossless* and *lossy*. The former technique permits perfect reconstruction; whereas the latter technique does not and, consequently, offers better compression performance. Lossy techniques produce distorted image signals, and the level of distortion they introduce depends on the characteristics of the signal, the amount of compression that is achieved, and the distortion measure that is used. This work concentrates on a lossy technique and provides insight into the dependence between compression performance and subjective quality of the reproduced images.

Compression techniques can be roughly categorized into waveform, predictive,

statistical, and transform. Each of theses classes can be further subdivided based on whether the parameters of the coder are fixed or whether they change as a function of the data that is being coded (adaptive). Also, schemes using a model of the human visual system as a design starting point are classified as second generation techniques. The compression performance of these techniques is expected to be much higher than those using more standard methods. Reviews of picture coding techniques can be found in [8, 9, 26].

Waveform coding is a discrete-time discrete amplitude representation of the signal. Once the signal is sampled, its amplitude is quantized to one of N levels. Each level is represented by a binary word that is transmitted to the decoder, which in turn converts these binary words to discrete amplitude levels and reconstructs the image. When the sampled signal is one-dimensional, the technique is the well known pulse code modulation (PCM), but when the samples are groups of pixels forming vectors, the technique is called vector quantization. PCM is a good technique to describe images, but is not well suited to image compression. Vector quantization, on the other hand, can yield very large compressions and is a promising method in that respect.

In basic predictive coding systems, a prediction of the sample to be encoded is made from previously coded data that has been transmitted. The error resulting from the subtraction of the prediction from the value of the sample is quantized similarly as in waveform coding. The predictor must use only data that has been transmitted to the decoder, because the decoder must to be able to calculate the prediction on its own to regenerate the encoded signal properly. Differential pulse code modulation (DPCM), which is predictive PCM, has been developed for image coding. Some schemes include human sensitivity curves to quantization noise and perform fairly

well in the range of 1 to 2 bits per pixel.

In transform coding schemes, the original image is divided into subpictures, with each of these subpictures transformed into a set of coefficients. The primary purpose of the transformation is to represent groups of statistically *dependent* picture elements as a set of roughly *independent* transform coefficients. The coefficients contain sufficient information to reconstruct the original image with the use of an inverse transform. The coefficients are quantized and coded for transmission. At the receiver, the received bits are decoded into transform coefficients to which the inverse transform is applied to recover intensities of picture elements. Most of the compression is a result of dropping small coefficients and coarsely quantizing others, as required by the picture quality. The important parameters that determine the performance of transform coders are the size and shape of the subpictures, the type of transformation used, and the selection of the transmitted coefficients and then quantization. The most popular transform method is the discrete cosine transform (DCT) which gives very good results at 0.5 to 1 bits per pixel. Significant effort is focused on this method, and it is evolving as one of the good choices for high performance image compression coding.

Statistical coding is a straightforward application of rate-distortion theory called ***block coding.*** These methods use statistics of the signal such as frequency of occurrence of symbols or patterns of symbols. In Huffman coding, the image signal is divided into small blocks of equal size. A block can take different values and a probability measure is assigned to each of the possible values. A variable length code is then used to give smaller length codes to those blocks which are more likely. The size of the blocks cannot be large, as the size of the codebook required for their storage would be too large. Therefore, many applications use the degenerate case of a single pixel

block. In the Lempel-Ziv algorithm, strings of pixels are matched to strings that were transmitted in the recent past. The position and the length of the encoded string is transmitted to the decoder. The string length can be very large, in which case, good compression is obtained. Statistical coding techniques are, in general, lossless techniques and achieve only small compression performance for most test pictures. Also, because of the rigidity of the techniques, it is very difficult to add other criteria based on human subjectivity. Hence, these techniques are not considered as serious contenders for practical image compression systems, but are useful for the compression of scalar coefficients having a limited range of values.

The goal of this research is to improve the subjective quality of images encoded with spatial vector quantization. By increased subjective quality, we mean that the viewer perceives less objectionable distortion when looking at the coded images. Vector quantization is a general technique in the sense that several variations of the basic scheme exist. Through the study of the performance of the basic scheme, we present the difficulties that need to be addressed in order to design an efficient image compression scheme. We will study some advanced vector quantization schemes, and show how they use the characteristics of the image signal to improve the quality of coded images. We will also include in the design, schemes which consider the psychovisual operations performed by the eye and the human brain on the visual field. The combination of these techniques should improve the quality of the coder, so that the end user can appreciate the subjective difference. In this thesis, we do not consider coding complexity and hardware ease of implementation. These factors are important to consider for any application but extend beyond the scope of this thesis.

Vector quantization and its possible variations are studied in Chapter 2. We first present a brief survey of the vector quantization techniques proposed in the literature.

We then present in greater detail the basic vector quantization scheme and two extensions of interest: predictive vector quantization and finite state vector quantization. In Chapter 3, we study the visual psychophysics of the human and develop schemes based on those observations. We first explain the physiological process behind human vision. We discuss only the early vision process which occurs in the eye and in the first few neuron layers. We then present a model for the brightness perception process and present how such psychovisual information is included in subjective scalar quantizers proposed in the literature. We then develop a new distortion measure based on a more subjective criterion and a scheme utilizing the brightness perception process of the human visual system. In Chapter 4, we present the experimental simulation results obtained with the proposed techniques taken individually, and then combined together. We discuss the subjective quality of the coded images and the implication of the proposed schemes as they are presented. In the conclusion, we summarize the subjective improvements obtained with the proposed techniques and discuss future work.

Chapter 2

VECTOR QUANTIZATION

In this Chapter, we present vector quantization as a source coding technique. Vector quantization is a very general and versatile information encoding method. For this reason, we also discuss several of its possible extensions and variations. We then describe in more detail, memoryless vector quantization as a basic technique and present two other techniques, mean predictive coding and finite-state vector quantization, which also attempt to utilize the inherent memory (or redundancy) of the picture signal. We explain how the encoders and the decoders are designed and describe commonly used techniques to optimize them. For convenience and readability, we use VQ as an acronym for both vector quantization and vector quantizer.

2.1 Survey of Techniques

A fundamental result of Shannon's rate-distortion theory, the branch of information theory devoted to data compression, is that better performance can always be achieved by coding vectors (blocks) instead of scalars. This holds even if the data source is *memoryless*, i.e., consists of a sequence of independent random variables, but greater performance gains can be obtained if the source samples are correlated.

This theory has had a limited impact on system design in the past because it does not provide constructive design techniques for vector coders and traditional scalar coders often yielded satisfactory performance with a minimum of complexity. Before 1980, very few papers were published on the subject of vector quantization, until the vector generalization of an algorithm for optimum design of scalar quantizers [13] was developed by Linde *et al.* [7]. This algorithm turned out to be of major importance in vector quantization and led to several new developments. Since then, vector quantization is increasingly used in the design of a variety of systems.

This section is a succinct survey of the basic design algorithm and many of its variations and applications. We begin with the simplest technique, the memoryless vector quantizer which is the multidimensional extension of pulse code modulation (PCM). Next, variations of the basic VQ which reduce complexity or memory at the expense of a hopefully tolerable loss in performance are described. We then discuss VQ with memory: feedback vector quantizers such as predictive VQ and finite-state VQ. The reader is referred to [10, 15, 20, 21, 31] for more complete surveys of VQ.

A vector quantizer can be defined as a mapping Q of a k-dimensional vector space $\mathbb{R}^k$ into a finite subset $\mathbf{Y}$ of $\mathbb{R}^k$

$$Q : \mathbb{R}^k \rightarrow \mathbf{Y} = \{\mathbf{y}_i : i = 0, 1, \ldots, N-1\}, \tag{2.1}$$

where $\mathbf{Y}$ is the set of reproduction vectors and N is the number of vectors in $\mathbf{Y}$. This set of reproduction vectors is commonly called the codebook and each of the reproduction vectors is a codevector. VQ can also be seen as a combination of two functions: an encoder, which receives the input vector $\mathbf{x}$ and generates the address or index of the reproduction vector specified by $\mathbf{y} = Q(\mathbf{x})$, and a decoder which uses this address to generate the reproduction vector $\mathbf{y}$. If a distortion measure $d(\mathbf{x}, \mathbf{y})$ which represents the cost associated with reproducing vector $\mathbf{x}$ by $\mathbf{y}$ is defined, then

the best mapping $\mathcal{Q}$ is the one which minimizes $d(\mathbf{x},\mathbf{y})$ [15]. The most common distortion measure is the squared-error distortion given by

$$d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2 = \sum_{j=0}^{k-1}(x_j - y_j)^2. \tag{2.2}$$

The VQ, during the encoding process, receives a sequence of source vectors and generates a sequence of channel symbols or indices. There is one index generated for every input vector. The decoder, when receiving the channel symbols, outputs the codevector $\mathbf{y}$ that corresponds to the received index. While the decoder can be easily implemented by a table lookup or a ROM, the encoder is far more complex and must contain the decoder itself as well as a source vector matching procedure.

Source coding using vector quantization requires a large computational effort at the encoder to search the whole codebook in order to identify the nearest matching reproduction vector to the input vector. Several schemes have been proposed to improve on the complexity of full search vector quantization. One such scheme is that of tree-searched VQ, where codevectors are situated at the leaves of the tree and the other nodes contain some information that guides the encoder through the tree. For example, each node of a binary tree could contain a hyperplane that cuts the vector space of that node in two. The encoder tests on which side of the hyperplane the input vector lies and moves down the tree to the corresponding children node, iterating the process until it reaches a leaf, at which point it has selected a codevector. Starting from the top, this method tesselates the whole input vector space into *Voronoi regions*. The main advantage of tree-searched techniques is that they reduce the number of times the distortion function is computed from N to $\log_2(N)$, but achieve this at the expense of increased memory requirements (usually double) to store the codebook. Apart from tree-based methods, some schemes attempt to reduce the amount of processing required to compute $d(\mathbf{x},\mathbf{y})$, while others try to arrange the codebook in a

preordered fashion so that only a subset of the whole codebook needs to be searched once a simple feature is extracted from the input vector, e.g., the norm of $\mathbf{x}$. Since the goal of this thesis is to increase the subjective quality of the encoding-decoding system and *not* to speed up the encoding process, we mention these schemes only for the sake of completeness, but will not describe them further.

Several proposed VQ schemes do some preprocessing on the input vectors before encoding them with vector quantization. Typically, a *gain/shape* technique is used. The sample mean of each input vector is computed and then subtracted from the vector. The error vector obtained is called the shape vector and is encoded with a memoryless VQ. The sample mean, which is called the gain, is transmitted on a side information channel using standard scalar compression techniques. Note that sometimes the variance is also removed from the input vector, transforming it into a unit-variance zero-mean vector, and transmitted the same way the sample mean is. The main goal of this technique is to try to reduce the power of the signal that is vector quantized, therefore obtaining better performance at a fixed rate. The use of side information, however, may be beneficial or not. For very high quality encoding, residual gain/shape VQ will perform very well, but at the expense of higher rates due to the side information. For very small bit rates, or for small vector dimensions, they usually perform poorly because the side information becomes prevalent.

Other techniques try to improve the quality of the codebook to obtain better performance. In classified VQ [11], the input vectors are classified into a few subjective categories which, in turn, separates the training sequence into a few subsequences. The categories could include sharp edge, shading edge, flat area, and mixed gradients. The latter mainly contains vectors which cannot be classified in any of the other classes and most likely consists of vectors representing a high visual activity, but without a

clear edge. The subspaces are usually selected using *objective* criteria like the variance of the block, the edge strength of individual pixels in the block, a combination of these two, or a more complex procedure. The important fact is that the training sequence is split into subjectively similar vectors. The number of subsequences and the number of codevectors allocated to each is a difficult problem that needs to be addressed by the designer. Once the subspaces are well defined, small codebooks are designed for each of the classes and merged together to form the final codebook. The encoder can perform the same classification as during the codebook design process and choose the appropriate codebook to encode the input vectors.

Another scheme, proposed by Goldberg *et al.* [21], introduced the concept of an adaptive codebook. As images are encoded, small portions or the totality of the codebook are replenished. The new codevectors help the decoder to adapt to spatially or temporally varying images. This technique has the disadvantage that not only encoding, but also codebook design processing, have to be performed by the encoder. The decoder also becomes more complex as it needs to update the codebook; a considerable increase in complexity if we compare it to a ROM implementation.

Several other techniques can be classified into the category of *feedback* techniques, or equivalently, coding schemes with memory. The simplest approach is predictive coding, where an estimate of the input vector is computed based on the past information that is available to both the encoder and the decoder. The prediction process could require some side information to be sent to the decoder, but often does not. Once the input vector estimate is obtained, it is subtracted from the input vector, and the encoder processes the residual error. As in the case of gain/shape VQ, the main purpose of this scheme is to reduce the dynamic range of the input vectors, but without the use of side information. Predictive coding schemes usually perform better

than nonpredictive schemes, but the coding performance depends on the accuracy of the predictor.

Another quite different technique to include memory is to use a *finite state* vector quantizer (FSVQ). FSVQ schemes have two attributes: a set of states with an associated encoder to each one, and a next state function or set of transition rules. The codebooks for each of the encoders contain, in most cases, the same number of codevectors N, so that the transmission rate depends only on N, and not on the states themselves. The codebook storage requirement, however, increases linearly with the number of states. These techniques are useful when the source signal displays several different characteristics. They are very well suited to encode Markov field processes. These coding techniques can also be viewed as directed graphs, because of their state-to-state transitions, and could be combined with popular search optimization algorithms such as the Viterbi Algorithm. They are then called *delayed* decision coding techniques because they use several input vectors before making a decision on which reproduction vector to select. This introduces an additional delay into the encoding of several vectors, but ensures better long run average distortion behavior. We will describe FSVQ in greater detail in Section 2.1.2

Up to now, all techniques described use vectors of identical dimensions. There is, however, a technique called *hierarchical* vector quantization which attempts to incorporate vectors of many different dimensions in the encoding process. Of course, at least one codebook is needed for each of the supported dimensions. In image coding, the technique of quadtree decomposition is often used to generate 2×2, 4×4, 8×8 and 16×16 vectors. Flat areas of an image are then encoded using large vectors and low bit rates, whereas highly detailed areas are encoded using small vectors and higher rates. For similar overall rates, this scheme has better spatial definition and,

therefore, encoding quality, than standard coding techniques.

This completes our survey of VQ techniques, wherein we briefly described the techniques and classified them into several categories. Each category tackles one aspect of the design of a VQ system and shows how complex and diversified vector quantization systems can get. We will now present in detail three of the above techniques, memoryless VQ, predictive VQ, and finite state VQ.

2.2 Memoryless Vector Quantization

Memoryless vector quantization is the vector generalization of PCM. In this section, we describe the mathematical representation of the encoding and decoding processes as well as their design procedure.

Let $\{\mathbf{x}_t\}$ be a stationary discrete time sequence of k-dimensional vectors $\mathbf{x}_t \in \mathbb{R}^k$. An N-level vector quantizer consists of a codebook or reproduction alphabet $\hat{\mathcal{A}} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ and a mapping $Q : \mathbb{R}^k \rightarrow \hat{\mathcal{A}}$ or, equivalently, a partition $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ of $\mathbb{R}^k$ such that $Q(\mathbf{x}) = \mathbf{y}_i$ if $\mathbf{x} \in S_i$. In fact, a VQ is usually an encoder mapping of $\mathbb{R}^k$ into binary vectors or channel symbols and a decoder mapping from the channel symbols to $\hat{\mathcal{A}}$, but for the performance analysis, only the mapping Q is important.

The rate of a VQ is given by $R = \frac{1}{k}\log_2 N$ bits per input source symbol, which is the number of binary digits that must be transmitted or stored in order for the receiver to produce $Q(\mathbf{x})$. For images, the input symbols are pixels and a group of k pixels form a source vector. The number of codevectors N in the codebook varies from application to application, and ranges generally between 64 to 1024 for image coders.

Given a distortion measure $d : \mathbb{R}^k \times \hat{\mathcal{A}} \rightarrow [0, \infty)$ assigning a distortion or cost $d(\mathbf{x}, \mathbf{y})$ to the reproduction of $\mathbf{x}$ by $\mathbf{y}$, the performance of the quantizer $\mathcal{Q}$ can be measured by the expected distortion [15]

$$\begin{aligned} D(\mathcal{Q}) &= E[d(\mathbf{x}, \mathcal{Q}(\mathbf{x}))] \\ &= \sum_{i=1}^{N} E[d(\mathbf{x}, \mathbf{y}_i | \mathbf{x} \in S_i)] P_r(\mathbf{x} \in S_i), \end{aligned} \tag{2.3}$$

where E denotes the expectation and $P_r(\mathbf{x} \in S_i)$ is the probability that the input vector $\mathbf{x}$ is in subspace S_i. An N-level quantizer is said to be optimal for a source, if $D(\mathcal{Q})$ is minimized over all N-level quantizers. As in Lloyd's method [13] for $k = 1$ and $d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^2$, two necessary conditions for optimality are:

1. That S be optimal for $\hat{\mathcal{A}}$, which is accomplished by using a minimum distortion or nearest neighbor selection rule

$$\mathcal{Q}(\mathbf{x}) = \mathbf{y}_i, \text{ if } d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j) \; \forall \, j \neq i \tag{2.4}$$

   and which results in the cells S_i being the Voronoi regions of the alphabet $\hat{\mathcal{A}}$.

2. That $\hat{\mathcal{A}}$ should be optimal for S, which is accomplished by choosing $\mathbf{y}_i$ so that

$$E[d(\mathbf{x}, \mathbf{y}_i) | \mathbf{x} \in S_i] = \min_{\mathbf{u}} E[d(\mathbf{x}, \mathbf{u}) | \mathbf{x} \in S_i] \tag{2.5}$$

   where $\mathbf{u}$ is a vector in $\mathbb{R}^k$ and $i = 1, \ldots, N$.

We assume that all subspaces S_i are nonempty, i.e., that $P_r(\mathbf{x} \in S_i) > 0 \; \forall \, i$. The obtained vector $\mathbf{u}$ is called the *generalized centroid* of the set S_i with respect to d and we write $\mathbf{u} = \text{Cent}(S_i)$.

These properties form the basis of Lloyd's iterative method and its generalization for vectors. The algorithm is fairly simple and goes as follows:

a) Start with an initial codebook $\hat{\mathcal{A}}_0$ and set $n = 0$.

b) Find the optimal partition S_n for $\hat{\mathcal{A}}_n$. This is done at the encoder using Equation 2.4. In fact, the obtained partition S_n completely defines the optimal encoder given the decoder $\hat{\mathcal{A}}_n$.

c) Find the optimal decoder $\hat{\mathcal{A}}_{n+1}$ for the new partition S_n using Equation 2.5. This will define completely the new decoder.

d) Alternate between steps b) and c) until some convergence criterion is met. We use the expected distortion as a convergence factor. If its rate of change falls below a certain threshold, e.g., 0.1%, the iterative procedure is stopped.

This technique is commonly referred to as the LBG algorithm, named after the three authors who proposed it. It is also sometimes referred to as the generalized Lloyd algorithm (GLA). The algorithm can be run using either the 'true' expectation corresponding to the known probability distribution functions, or using sample averages based on a long typical training sequence. In the latter case, it is assumed that the long term average is equal to the expected value, i.e., that the source is ergodic, which is *not* a reasonable assumption for images. But, as was suggested in [15], if the training sequence is representative and long enough, the encoder should perform as well for signals within the training sequence as for signals outside of it. We are going to use training sequences in this work, as it is felt that the statistical modeling of images is too complex and prone to innacuracies.

It has been proven that the LBG algorithm is optimal for the scalar case ($k = 1$) [13, 12] when a distortion measure of the form

$$d(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}, \mathbf{x} - \mathbf{y}), \tag{2.6}$$

where f is a convex function of the error $|\mathbf{x} - \mathbf{y}|$, is used. No such conditions have been found for $k > 1$. The LBG algorithm is considered to reach a local minimum

only. The location of the minimum is highly dependent on the initial codebook $\hat{\mathcal{A}}$.

A procedure to properly define the initial codebook $\hat{\mathcal{A}}_0$ is, therefore, needed. As mentioned in [15], it can be selected in a variety of ways, from lower rate codes using the splitting technique [7], from lower dimension codes using the product technique [12], or by selecting N vectors at random from the training sequence [7]. The product technique uses an optimum codebook designed for a smaller dimension (smaller k) and replicates it for the new dimensions. For instance, if a codebook is designed for 1-dimension (scalar), it can be replicated once or twice to yield a 2-dimensional or 3-dimensional codebook, respectively. With this approach, the LBG algorithm converges to very good local minimum and, most of the time, to the global minimum [12], when the source symbols from each dimension are approximately independent of each other so that the input vector space shows some circular symmetry. An image vector space is, however, *not* symmetric, and the product technique should not give good results. We thought that random selection of vectors would not consistently yield good codes and make comparison between techniques difficult. As opposed to the other methods, the splitting technique can be justified qualitatively.

There exists another technique to design codebooks that is most often used to design tree-searched VQ. It is the hyperplane testing method, which starts by defining a hyperplane that cuts the whole input vector space in two. The centroid of each region is used as the best representation vector for that subspace. Each subspace is then again subdivided in two by a hyperplane. The process continues until enough code vectors are generated. In the splitting technique, we start by finding the centroid of the whole input sequence, the optimal 1-codevector codebook. From this centroid, two new vectors are created by adding a small perturbation vector ϵ so that $\mathbf{y}_1 = \text{Cent}(S_0) + \epsilon$ and $\mathbf{y}_2 = \text{Cent}(S_0) - \epsilon$, where S_0 represents the entire training set. The

LBG algorithm uses these two vectors as an initial codebook to generate a locally optimal 2-codevector codebook. Each of these two codevectors represent a subspace of S_0, exactly as in the hyperplane case. Splitting each of the optimal $\mathbf{y_1}$ and $\mathbf{y_2}$ in two will give four vectors that may be used to run the LBG algorithm again. This process is iterated until the desired number of codevectors is obtained. Based on our experience, we are confident that the splitting method yields good results.

In this research, instead of the usual squared-error distortion, we use a weighted quadratic distortion measure of the form

$$d(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^t \mathbf{W_x} (\mathbf{x}-\mathbf{y}), \tag{2.7}$$

where $\mathbf{W_x}$ is a positive definite weighting matrix that depends on the input vector $\mathbf{x}$. A distortion measure of this form - the gain-normalized Itakura-Saito distortion was considered in [7] for speech compression applications. Very few researchers have tried, to our knowledge, to use such a distortion measure to encode images. Note that the distortion measure of Equation 2.7 includes the usual squared-error distortion measure as a special case when $\mathbf{W_x} = I$, the $k \times k$ identity matrix. The development of our distortion measure will be completed in Chapter 3. We now present how the LBG algorithm is applied to a quadratic distortion measure, but need to develop some important relations first.

We assume that the matrix $E(\mathbf{W_x})$, where the expectation of a matrix is the matrix of the component expectations, is positive definite and, hence, invertible. The following vector is, therefore, well defined

$$\mathbf{y} = (E[\mathbf{W_x}])^{-1} E[\mathbf{W_x x}]. \tag{2.8}$$

We also immediately have the following variation of the orthogonality principle:

$$E[\mathbf{W_x}(\mathbf{x}-\mathbf{y})] = E[\mathbf{W_x x}] - E[\mathbf{W_x}]\mathbf{y} \tag{2.9}$$

$$= E[\mathbf{W_x x}] - E[\mathbf{W_x}](E[\mathbf{W_x}])^{-1}E[\mathbf{W_x x}]$$
$$= \mathbf{0}.$$

We now derive several results:

a) Given a distortion measure $d(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^t\mathbf{W_x}(\mathbf{x}-\mathbf{y})$ with $\mathbf{W_x}$ positive definite for all x, then

$$\text{Cent}(S) = E[\mathbf{W_x}|\mathbf{x} \in S]^{-1}E[\mathbf{W_x x}|\mathbf{x} \in S]. \tag{2.10}$$

b) Given a partition $S = \{S_1, S_2, \ldots, S_N\}$ and a reproduction alphabet $\hat{\mathcal{A}} = \{\text{Cent}(S_i) : i = 1, \ldots, N\}$, and letting $\mathcal{Q}$ denote the corresponding quantizer, then

$$E[\mathbf{W_x x}] = E[\mathbf{W_x}\mathcal{Q}(\mathbf{x})] \tag{2.11}$$

Proof of a)

Abbreviate $E[\cdot|\mathbf{x} \in S]$ to $E_S[\cdot]$ and define $\mathbf{y}$ as in (2.8) with E_S replaced with E. Then for an arbitrary $\mathbf{u}$,

$$\begin{aligned} &E_S[(\mathbf{x}-\mathbf{u})^t\mathbf{W_x}(\mathbf{x}-\mathbf{u})] \\ &\quad = E_S[((\mathbf{x}-\mathbf{y})+(\mathbf{y}-\mathbf{u}))^t\mathbf{W_x}((\mathbf{x}-\mathbf{y})+(\mathbf{y}-\mathbf{u}))] \qquad (2.12)\\ &\quad = E_S[(\mathbf{x}-\mathbf{y})^t\mathbf{W_x}(\mathbf{x}-\mathbf{y})] + (\mathbf{y}-\mathbf{u})^tE_S[\mathbf{W_x}](\mathbf{y}-\mathbf{u}) + \\ &\qquad 2(\mathbf{y}-\mathbf{u})E_S[\mathbf{W_x}(\mathbf{x}-\mathbf{y})]. \end{aligned}$$

The right-most term is zero from Equation 2.10 and, therefore,

$$E_S[(\mathbf{x}-\mathbf{u})^t\mathbf{W_x}(\mathbf{x}-\mathbf{u})] \geq E_S[(\mathbf{x}-\mathbf{y})^t\mathbf{W_x}(\mathbf{x}-\mathbf{y})] \tag{2.13}$$

with equality if $\mathbf{u} = \mathbf{y}$. This characterizes $\mathbf{y}$ as the centroid of S.

Proof of b)

Observe that Equation 2.10 implies

$$E[\mathbf{W_x}(\mathbf{x}-\mathcal{Q}(\mathbf{x}))] = \sum_{i=1}^{N} P_r\{\mathbf{x} \in S_i\}E_{S_i}[\mathbf{W_x}(\mathbf{x}-\text{Cent}(S_i))] = \mathbf{0}, \tag{2.14}$$

which, in turn, implies Equation 2.11.

For the case of a sample distribution defined by a training sequence, the centroid of a set S is given by

$$\text{Cent}(S) = (\sum_{\mathbf{x}\in S} \mathbf{W}_\mathbf{x})^{-1}(\sum_{\mathbf{x}\in S} \mathbf{W}_\mathbf{x}\mathbf{x}). \tag{2.15}$$

The numerator and the denominator terms of the centroid can be recursively computed during the encoding process of the algorithm. Computationally, the most difficult part is inversion of the average weighting matrix. In some cases, $\mathbf{W}_\mathbf{x}$ is a diagonal matrix which greatly simplifies the calculations for the inverse operation. For the case of $\mathbf{W}_\mathbf{x} = \mathbf{I}$, we get the more familiar forms of the squared-error case,

$$\text{Cent}(S) = E[\mathbf{x}|\mathbf{x} \in S] \tag{2.16}$$

$$E[\mathbf{x}] = E[Q(\mathbf{x})] \tag{2.17}$$

Furthermore, if $\mathbf{W}_\mathbf{x}$ does not depend on $\mathbf{x}$, then Equations 2.16 and 2.17 remain true.

We have seen that only a decoder $\hat{A}$ or a space partition $S = \{S_1, S_2, \ldots, S_N\}$ and a distortion measure $d(\mathbf{x}, \mathbf{x} - \mathbf{y})$ are needed in order to completely define a memoryless vector quantizer. We presented a procedure to design a memoryless VQ with a quadratic distortion measure, using the LBG algorithm and the splitting technique. Simulation results using this technique will be presented in Section 4.1.

2.3 Predictive Vector Quantization

In the previous Section, we considered memoryless vector quantization. However, since consecutive input vectors of an image are statistically correlated, better performance can be achieved if the intervector dependence is incorporated into the encoder.

There exist several ways to include memory in the VQ. In this section, that of predictive vector quantization is discussed, whereas that of finite-state VQ will be discussed in Section 2.4.2.

In predictive VQ [17], a prediction of an image is formed and the residual image, the error between the prediction and the original, is vector quantized. When applied to images, the prediction changes the distribution of the residuals as compared to the original. If the predictor is good, the standard deviation of the input sequence is smaller than the original, making it easier for the VQ to reproduce the image accurately. As mentioned in [27], significant coding gains can be obtained with this technique.

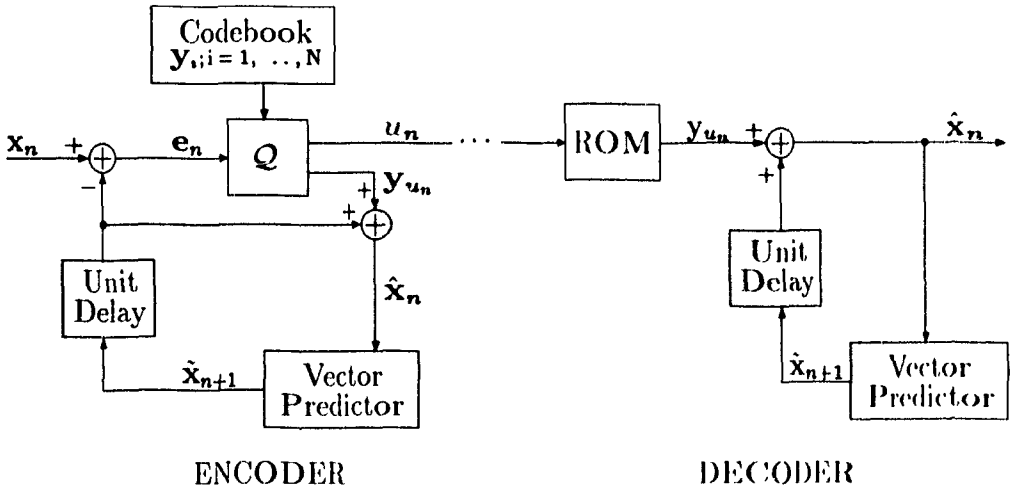

Figure 2.1: Predictive Vector Quantization. Block diagrams showing the structure of the encoder and of the decoder

The basic algorithm for predictive VQ is a vector generalization of scalar predictive quantization. The block diagram of a predictive VQ is shown in Figure 2.1, where $\mathbf{x}_n$, $\hat{\mathbf{x}}_n$, and $\tilde{\mathbf{x}}_n$ are the input vector, the reproduction vector, and the estimated vector, respectively, and where $\mathbf{e}_n$ denotes the error vector which is vector quantized using codevectors $\mathbf{y}_i$ in the codebook $\hat{A}$. U_i denotes the transmitted channel symbols or

indices of the codevectors.

We use very simple prediction in this research, since the study of good prediction schemes is beyond the scope of this thesis. The simplest predictor would use the last transmitted reproduction vector as an estimate. This method, however, introduces some artifacts, in the form of impulsive noise, that are annoying to the human viewer. Hence, we elected to use a predictive mean scheme, where the predicted vector is a constant vector whose intensity is equal to the mean of the last transmitted reproduction vector. This ensures that both the encoder and the decoder can create the predicted vector without the use of side information. This method yields acceptable prediction for slowly varying regions or near horizontal edges, but has the disadvantage of poorly estimating vectors around vertical or diagonal edges. The residual image has a larger standard deviation than it would if a better prediction scheme was used, but since most parts of the test images are uniform, this prediction scheme yields results which are acceptable for our purposes. What happens is that those vectors lying in uniform regions will get clustered around the origin, thereby helping the vector quantizer to represent these vectors with fewer codevectors, while using more codevectors to represent high activity regions. Furthermore, since the predictor does not remove edge information from the original, an encoding scheme with a subjective encoding criterion based on the human vision of edges can be combined with a predictive scheme to yield even better results.

The encoder for the predictive VQ scheme was designed using the LBG algorithm in the same way as in memoryless VQ, but the input vectors were preprocessed by the predictor before being quantized. In this sense, predictive VQ is just an add-on feature that we implemented on a memoryless VQ. We will try to use prediction along with other VQ schemes and present the results in Section 4.2.

2.4 Finite State Vector Quantization

In this Section, we will discuss a more general technique to include memory into a vector quantizer system known as finite-state vector quantization (FSVQ). A general VQ with memory can be completely described by a finite state space $\mathcal{B} = \{0, 1, \ldots, B-1\}$, where each state b in $\mathcal{B}$ is associated with a separate VQ: an encoder α_b, a decoder β_b, and a codebook $\mathcal{C}_b$, and having a set of rules governing the state transitions. The channel symbol space $\mathcal{U} = \{0, 1, \ldots, N\}$ is the same as that of memoryless VQ and contains integers or indices pointing to codevectors in their respective codebook. Consider a data compression system consisting of a sequential machine such that if the machine is in state b, then it uses the quantizer with encoder α_b and decoder β_b. Its next state is selected by a mapping called the next-state function or state-transition function f such that, given a state b and a channel symbol u, then $f(u, b)$ is the new state of the machine. More precisely, given a sequence of input vectors $\{\mathbf{x}_n : n = 0, 1, \ldots\}$ and an initial state b_0, then the subsequent state sequence s_n, channel symbol sequence u_n, and reproduction vector sequence $\mathbf{y}_n$ are recursively defined for $n = 0, 1, \ldots$ by

$$u_n = \alpha_{b_n}(\mathbf{x}_n), \qquad \mathbf{y}_n = \beta_{b_n}(u_n), \qquad b_{n+1} = f(u_n, b_n). \tag{2.18}$$

Since the next-state function depends only on the current state and the channel symbol, the decoder can track the state of the encoder if it knows the initial state and the channel sequence. The possibility to use different quantizers based on the past without increasing the rate (no side information) helps the code to perform better than a memoryless quantizer of the same dimension and rate.

A useful property of a FSVQ is that it can be used in a directed graph encoding system where several transitions are considered before a decision on the minimum

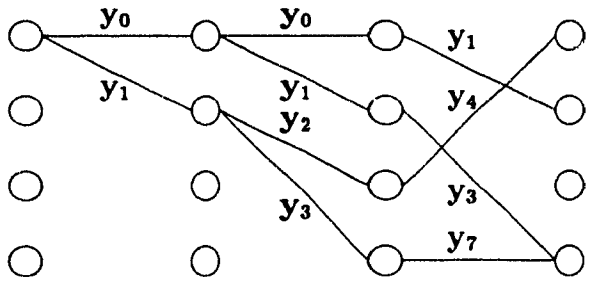

Figure 2.2: Possible paths in a directed graph with four states

distortion reproduction vector is taken. One example of such a directed graph is shown in Figure 2.2 where each column of small circles represents the set of possible states and each line represents a state transition and has a codevector associated to it. This arrangement is referred to as a labeled-transition representation because the codevectors correspond to transitions. Instead of using the ordinary VQ encoder which only looks at the current input vector in order to decide on a channel symbol, a graph search technique such as the Viterbi algorithm can be used to search for a minimum cost path through several levels of the directed graph before making a decision on a channel symbol. This introduces an additional delay into the encoding of several vectors, but it ensures better long run average distortion behavior. This technique is called trellis encoding and is also referred to as lookahead coding, delayed decision coding, and multipath search coding. We point out that the Viterbi algorithm gives optimal results [24] for a directed graph with a finite number of states, but has a complexity that increases with the number of states. Since we are not concerned with computational speed in this work, we use the Viterbi algorithm throughout this thesis.

The general design technique for finite-state vector quantizers is reported in [19]. There are two principal components: the design of a set of initial state-codebooks

C_b and of a next-state function, and the use of a variation of the LBG algorithm to attempt to improve the state-codebooks. The latter is accomplished by a slight extension of the basic algorithm presented in [14] for the design of scalar trellis encoders. The training sequence is first encoded using the FSVQ, and then all the codevectors are replaced by the centroids of the training vectors which map into these codevectors: however, the centroids are conditioned on both the channel symbol and the state. While those conditional averages are likely impossible to compute analytically, they are easily computed by running averages on a training sequence. Using the same notation as for the memoryless case,

$$\mathbf{y}_{i,b} = \mathrm{Cent}(S_{i,b}) = \{ \sum_{\mathbf{x} \in S_{i,b}} \mathbf{W}_{\mathbf{x}} \}^{-1} \{ \sum_{\mathbf{x} \in S_{i,b}} \mathbf{W}_{\mathbf{x}} \mathbf{x} \}, \tag{2.19}$$

where $S_{i,b}$ is the subspace which contains all input vectors that are mapped to $\beta_b(i)$ when the encoder is in state b. As with memoryless VQ, using centroids to adjust the codebook cannot yield a code with a larger distortion and eventually goes to a local minimum.

The design of the first component, the initial state-codebook and the next-state function, is more complicated. We use two different approaches: a simpler, well known method using trellises and a complex, but more promising method referred to in [19] as the omniscient design approach.

2.4.1 Vector trellis encoding

A vector trellis encoder (VTE) is the vector extension to the trellis waveform coder presented in [14]. A good presentation of VTE systems may be found in [19].

The most general case of a trellis decoder consists of a finite-state machine driving a table lookup codebook. Symbols arriving from the channel drive the finite state

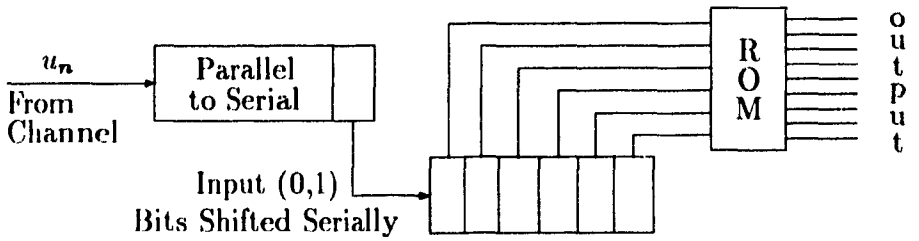

Figure 2.3: Vector Trellis Encoding System: Shift Register Implementation

machine, which in turn selects the reproduction vectors from the codebook. In this section, we consider the special case where the finite state machine is a shift register containing the m most recent channel symbols. The content of the shift register is used to address a ROM or a table index to select the decoder output. This decoder structure is easily implemented in hardware. The encoder is more complex and includes a FSVQ with a next-state function governed by the shift register, and a search encoding algorithm such as the Viterbi algorithm. We call this technique vector trellis quantization (VTQ) to differentiate it from the more general VTE method.

In [14], an algorithm for the design of a scalar VTQ for any number of states and fixed integer bit rates is presented. We use the same technique, but modify it to fit the requirements of VQ and that of *fractional* bit rates. We use a fixed bit rate encoder, therefore the number of transitions out of each state is the same, i.e., each state-codebook has the same size. The next-state function is well defined given the number of states B and the number of transitions from each state $T = 2^{kR}$, where k is the dimension of the vectors and R is the bit rate in bits/input symbol. If j_n is the current content of the shift register and i_n is the incoming channel symbol, then the next-state function is given by

$$b_{n+1} = ((j_n << kR) + i_n) \bmod B, \tag{2.20}$$

where << is the left shifting operation and mod is the modulus operation and simply truncates the right hand side to get the right most $\log_2 B$ bits.

With the next state function well defined, an initial codebook is needed before we can start the iterative procedure of the LBG algorithm. However, the initial state-codebooks cannot be trivially initialized and should be designed from smaller codebooks. Such a procedure has been presented in [14] and is called, simply, the extension. The new, larger decoder, is constructed by adding an additional stage to the shift register of the starting decoder. The extension starts from a decoder with register length l and increases it to $l+1$. Doing so, the total size of the codebook must increase from 2^l to 2^{l+1}. We fill in the new codevectors by duplicating the old codebook. Let the old codebook contain codevectors $\{\mathbf{y}_i : i = 0, 1, \ldots, 2^l - 1\}$, then the extended codebook contains codevectors $\mathbf{y}'_i$

$$\mathbf{y}'_i = \mathbf{y}'_{i+2^l} = \mathbf{y}_i, \qquad i = 0, 1, \ldots, 2^l - 1. \tag{2.21}$$

Because of the regular symmetrical structure of the shift register implementation, the new codebook C' is identical in behavior to the old codebook for the first iteration of the LBG algorithm. When convergence for the register length $l+1$ is obtained after a few iterations, the extension can be run again until the size of the desired codebook is reached. During this procedure, the number of transitions from each state T always remains the same, and the number of states increases (doubles) at every iteration. Hence, the initial codebook should have T codevectors and a single state.

Given that a k-dimensional VTQ with B states and T transitions per state needs to be designed, we follow these steps:

a) Design a codebook with T codevectors using the memoryless VQ approach. Initialize B to 1.

b) Multiply B by 2. Extend the codebook.

c) Find the minimum average distortion encoding of the training sequence. This encoding induces a partition on the training sequence so that the vectors of the partition cell corresponding to a certain codevector can be clustered together to define a new codevector that will replace the old one. Some rule has to be defined for zero size clusters, i.e., codevectors that were not used during the encoding process. We use the codevector of the largest cluster that originates from the same state as the zero size cluster as the new codevector. A small perturbation is added to the new duplicated codevector to ensure that the state-codebook does not contain two identical vectors. If it so happens that all the clusters originating from a state are empty, then the codevectors from the T largest clusters of the whole codebook are used.

d) Compute the average distortion. If the average distortion decreases by more than a small percentage of the previous average distortion, then go to c), else go to e).

e) If the final codebook size is not reached, then go to b).

These steps combining the extension and the LBG algorithm, always assure at least nondecreasing average distortion at every iteration. Simulation results will be shown in Section 4.3.1.

2.4.2 Omniscient finite state vector quantization

This method is presented in [19, 16] and is the most promising method for the design of next-state functions for a labeled-transition FSVQ. We refer to it as the Omniscient FSVQ or OFSVQ. The design procedure for an OFSVQ with B states and rate $R = \frac{1}{k}\log_2 T$, where T is the number of transitions from each states and k is the dimension of the vectors, consists of four steps:

a) Use the training sequence to design an ordinary memoryless VQ with B codevectors, one for each state of the OFSVQ. Denote the resulting codebook by $C = \{\mathbf{c}(b):$

$b \in \mathcal{B}\}$. We refer to this VQ as the state label VQ. The codevectors in this special codebook are called state label vectors. We consider the FSVQ to be in the ideal state b if the last input vector, when quantized with the state label VQ, gets mapped to $\mathbf{c}(b)$. Note that this codebook and this notion of state selection are used only in the quantizer design process; they will not be part of the quantizer that will be implemented. The state label VQ is used for state selection in the initial guess design procedure, and later for nearest neighbor determination of the next-state function. When the final design is completed, the state label VQ is discarded.

b) For each state b, design an initial reproduction codebook $\mathcal{C}_b = \{\beta_b(u), u \in \mathcal{U}\}$, using the memoryless VQ design algorithm for the training subsequence composed of all successors to vectors for which the state label VQ chooses b, that is, the subsequence

$$\{\mathbf{x}_n : b = \min_{\sigma \in \mathcal{B}}{}^{-1} d(\mathbf{x}_{n-1}, \mathbf{c}(\sigma))\}, \tag{2.22}$$

where the inverse minimum notation means that the σ yielding the indicated minimum is the chosen state. Thus, each codebook $\mathcal{C}_b$ is designed to provide good performance in the following finite state machine. Given $\mathbf{x}_n$ and a current state s_n, determine the next state s_{n+1} of the machine by applying the state label VQ to $\mathbf{x}_n$. The VQ corresponding to the next state is then used to encode the next input vector. This machine is not a true FSVQ because the receiver cannot track the ideal state sequence from the initial state and the channel sequence alone, and, hence, it cannot decode the channel sequence properly unless it has some knowledge of the state sequence of the encoder through a side information channel. This finite state machine is called an omniscient FSVQ because the receiver requires what might be called omniscient knowledge to decode the channel sequence.

c) The ideal state selection using the state label VQ is now approximated in a way that the decoder can track, hence obtaining an ordinary FSVQ. Instead of choosing

the next state as the state label vector that best matches the current input vector, we select the state with a label best matching the reproduction vector of the current input vector. In other words, the state label obtained when quantizing the current output codevector with the state label VQ, is used as next state. This operation can be duplicated by the decoder. Thus, given the state labels $\mathbf{c}(b)$ and the decoders β_b designed above, we define the next-state function by

$$f(u,b) = \min_{b \in \mathcal{B}}{}^{-1} d(\mathbf{x}_{n-1}, \mathbf{c}(b)), b \in \mathcal{B}, u \in \mathcal{U}. \tag{2.23}$$

Further, since the reproduction vectors and the state label vectors are known prior to the encoding of the training sequence, we can compute the next state function and store it in a table, hence speeding the encoding process.

d) Attempt to improve the state-codebooks by encoding the training sequence using the given next-state function and updating each codevector by the centroid of the training vectors assigned to it.

The above algorithm is called the nearest-neighbor omniscient design algorithm to emphasize the fact that the next-state function is determined by a nearest neighbor or minimum distortion approximation to the omniscient finite state machine. Further improvements can be obtained by iterating steps c) and d) of this procedure with a state label update procedure included in the codebook relabeling. We form the new state label vectors $\mathbf{c}(b)$ by calculating the centroid of the set of vectors which were encoded with any of the codevectors whose next state is b,

$$\{\mathbf{x}_n : f(b_n, \alpha(\mathbf{x}_n, b_n)) = b\}. \tag{2.24}$$

In other words, we compute the centroid of all source vectors which were mapped to a state transition branch that terminates at state b. Continuing this procedure

usually improves the quantizer, but it can yield a worse quantizer. In our experiments, however, a small increase in distortion is almost always followed by a greater decrease in distortion in subsequent iterations, so we allow the worse quantizer to continue the improvement procedure. When the worse quantizer converges to a better overall quantizer, the iterative procedure has, in effect, escaped one local minimum to another, better local minimum. We will refer to this procedure as the iterative nearest-neighbor improvement algorithm. In the following chapters, however, we will use the terminology OFSVQ to denote the omniscient design techniques in general and refer to the names of the particular variations only when required for clarity.

Chapter 3

VISUAL PSYCHOPHYSICS

One of the most important objectives in the design of visual communication systems is that they only represent, transmit, and display that information which the human eye can see. To transmit and display characteristics of images that a human observer cannot perceive is a waste of channel resources and display media. We must understand, therefore, how we can represent pictures economically and transmit them with the minimum accuracy required by the human eye. In this chapter, we study some of those properties of human vision that are helpful in evaluating the quality of a coded picture when compared with the original and help us in designing the coder to achieve the lowest transmission rate for a given picture quality. This approach attempts to model the human visual system and uses the developed model as a prefilter on the input images. The resulting images contain presumably less information, but, nevertheless, all the information that is required for reconstruction of the original images. Good coding gains can be obtained when the coder, optimized on the prefiltered images, can reproduce them with fidelity.

When further compression or lower rates are required, however, it is necessary to study the features contained in images to which the human observer is most sensitive. In other words, we not only need to use the transfer function of the human eye, but

also the low level processing of the human visual system (HVS) which is the feature detection process. In image processing, features can represent, for example, edge strength and orientation, shading, texture, and color information. It is well known that the statistics of picture signals are nonstationary and that the required fidelity of reproduction demanded by the human eye varies from picture element to picture element (pixel). Consequently, for more efficient digital coding, it is desirable to adapt the coding strategies to those local properties (features) of the picture signal which determine the sensitivity of human observers to quantization noise. We attempt to include theses subjective features in the design of a vector quantizer.

In this chapter, we present as a necessary background a physiological description of important parts of the HVS. We will concentrate on the low level processing of the human optical system (the eye, the photoreceptors, and the first few nerve cells), particularly the brightness perception. We set aside the study of color as it is beyond the scope of this thesis. Then, a method proposed for scalar quantization with a subjective criteria is presented. We also qualitatively discuss the performance of the MSE as a distortion measure and investigate why it fails to produce subjectively acceptable results. Based on these observations, we develop a new distortion measure that can be applied to vector quantizer design.

3.1 Modeling the Human Visual System

In this section, we present models of specific parts of the HVS. We start by giving a functional description of the HVS and then present the theory behind brightness perception. We conclude the section with the presentation of a subjective scalar quantization scheme.

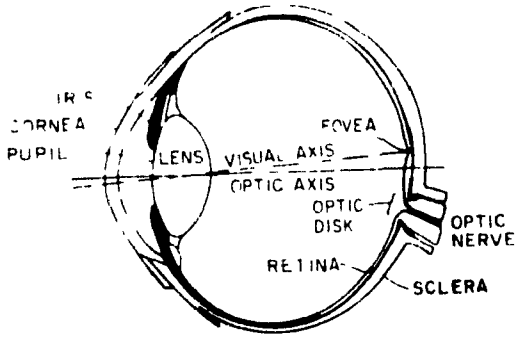

Figure 3.1: Diagram of the cross section of the human eye. (*From A.Netravali and B.Haskell, "Digital Pictures: Representation and Compression", Plenum Press, New York, 1988, p.253.*)

3.1.1 Functional description of the human visual system

We present a functional description of the human eye as a background for constructing a phenomenological model of the visual process consistent with physiology. The treatment is not sufficiently detailed to correlate precisely the biological parts with modules of the model. More detailed descriptions can be found in [2, 18].

Figure 3.1 illustrates the principal components of the human eye. Light from an external object is focussed by the cornea and lenses to form an image of the object on the retina. Refraction occurs at the cornea and is also affected by the varying thickness of the lenses. Since the eye is not a "perfect" optical system, a certain amount of spreading and consequent degradation takes place at the retina. Another source of degradation is eye movement. Of course, voluntary eye movements are necessary and enable us to track objects or to shift our attention from one object to another; however, involuntary eye movements of small magnitude also occur, even during steady fixation, and introduce a certain amount of temporal variation to the image. These involuntary movements consist of slow drifts from the point of fixation, corrective flicks (called saccades) at time intervals of about 0.3 to 0.7 seconds, as well

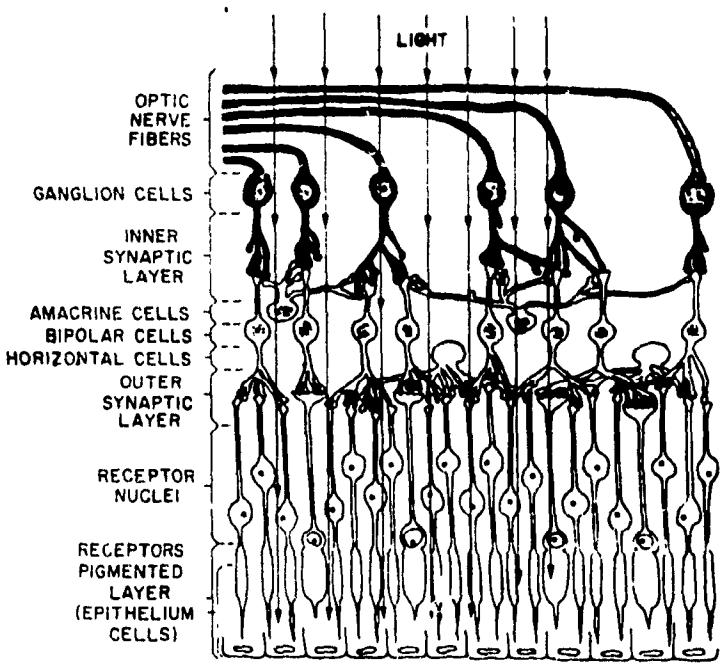

Figure 3.2: Schematic diagram of the retina showing the interconnections between receptors and bipolar, ganglion, horizontal and amacrine cells. (*From A.Netravali and B.Haskell, "Digital Pictures: Representation and Compression", Plenum Press, New York, 1988, p.254.*)

as high frequency tremors. Although involuntary eye movements degrade the image, in general, they are important for maintaining continuous visibility of the image, since a visual stimulus that is stationary on the retina fades and eventually disappears.

A schematic diagram of the retina is shown in Figure 3.2. The retina consists of a layer of photoreceptors and connecting nerve cells. The photoreceptors are curiously at the point of the layer that is farthest from the incoming light, and, therefore, light rays must pass through the layer of nerve cells before reaching the photoreceptors. The receptors contain photosensitive pigments that are capable of absorbing light and initiating the neural response.

The photoreceptors are of two kinds: rods and cones. In the region surrounding the fovea, only cones are present, and they are densely packed. The density decreases

rapidly as we move away from the fovea, whereas the density of rods increases. Cones are responsible for spatial acuity and color vision at normal daylight levels (called the photopic range), while rods are responsible for low light vision (called the scotopic range). Light absorbed by the receptors initiates chemical reactions that bleach the photosensitive pigments, causing a reduction in light sensitivity that is proportional to the fraction of pigment bleached. A change in ambient illumination causes the amount of bleached pigment to rise and fall to a new equilibrium level, thereby providing a mechanism for adapting to different light levels. The HVS can adapt in this fashion through a very large range of illumination levels (10^{13} to 1).

As seen in Figure 3.2, photoreceptors make a synaptic contact with the bipolar cells. A second synapse then connects the bipolar cells to the ganglion cells. Lateral interactions also take place by the means of horizontal and amacrine cells. The axons of the ganglion cells form the fibers of the optic nerve by which the signal is transmitted to the brain. The optic nerves coming from each eye meet at the optic chiasm, where the information is routed such that the left half of the visual field is processed by the right hemisphere and conversely. The first parts of the brain that perform processing of the visual signal are the lateral geniculates, which are two small regions situated near the center of the brain. The bulk of the vision process occurs, however, in the visual cortex which is situated at the back of the brain.

Lateral connections made by the amacrine and horizontal cells are responsible for amplitude companding and spatial frequency preemphasis of the visual signal by mediating the sensitivity of the ganglion cells to light. This effect, called *lateral inhibition*, results in a reduction of the signal from a cell when the neighboring cells are illuminated. The lateral connections result, for each ganglion cell, in a receptive field with an excitatory region in the center surrounded by an inhibitory region, or to

the opposite, in a receptive field with an inhibitory center and an excitatory surround. When excited, ganglion cells produce an electric pulse (firing) that propagates through their axons (optic nerve). It is believed that the information is coded in the firing rate of the cells, with the larger signals corresponding to the most rapid firing rates. A light stimulus exciting a ganglion cell raises its firing rate, but any stimulus in an inhibitory region tends to decrease its firing rate. This implies that a ganglion produces its highest response when there is a light pattern in its excitatory region and none in its inhibitory region. The center-surround pattern is generally circularly symmetric or elliptical. Since the density of the receptors is highest in the fovea, the size of the receptive fields is the smallest near the fovea and increases with distance from the fovea. By combining the response of several receptive fields, the brain is able to extract some simple features from the image, like edge height, orientation, length, thickness, and curvature.

3.1.2 Brightness perception

We present a succinct description of the brightness perception model of the human visual system and discuss the abilities of the HVS to perceive brightness and contrast. More extensive studies of the subject can be found in [2, 5, 18, 30].

As seen in Section 3.1.1, the HVS comprises several different biological parts each playing an important role. Although there is strong evidence that each part interacts with the others, most models that are developed use a simple succession of black boxes as a modeling approach for simplicity. The HVS is roughly divided into three parts, the optical system, the photo-transducers (rods and cones), and the neural connections. The role of the optical system is to project a clearly focussed image with an appropriate luminosity on the retina. The image is, however, not reproduced

perfectly but is blurred or spread. This effect can be modeled by a low-pass filter.

The photo-transducers account for the logarithmic nonlinearities in the intensity adaptation. As indicated by visual acuity experiments, the eye is extremely sensitive to even the smallest amount of light. There are thresholds below which no vision is possible, but such a study is not relevant to visual communication media as images always fall above these thresholds. In this context, a more relevant question arises as to how the photoreceptors are capable of maintaining a more or less consistent response to a very large range of input stimulus intensities. It is known that the range of inputs that can activate the HVS is of the order of 10^{13} to 1, even though the pupil area can only be varied by a factor of 16 to 1. We saw in Section 3.1.1 that the magnitude of the intensity impinging on the photoreceptors is coded into a frequency of pulses or firing rate at the output of the ganglion cells. These frequencies are, however, quite limited in their range, not exceeding a maximum of perhaps 1000 Hz [18], with 40 to 50 pulses per second being considered as the result of spontaneous, undetermined activity. Thus, the coded frequency range is at most 100 to 1, from which we observe there is a significant compression from the input to the output. Since linearity appears to prevail between the receptor potential output (synapse strength) and the resulting coded frequency, it appears that this compression is a result of the photochemical action of the transducers. It has been proposed that the response Ψ is related to the input intensity I by the equation

$$\Psi = k(I - I_0)^n, \tag{3.1}$$

where k is a constant, I_0 is the absolute threshold intensity, and $n < 1$. Experiments have shown that $n \approx 0.33$, approximately the cube root. Equation 3.1 is referred to as Steven's power law. Although the value of n can vary from 0.2 to 0.5 depending on the test patterns and the subject environment, Steven's power law is considered

to be an acceptable way in which to model the compression of the intensity, and due to its simplicity, it is used in many applications.

Experiments to determine brightness perception or the contrast sensitivity curve as a function of spatial frequency reveal that the overall response of the HVS is a bandpass filter with a center frequency in the range of 2 to 5 cycles per degree. The response is asymmetrical in the sense that the high frequency attenuation is steeper than the low frequency attenuation. The system can then be modeled by a cascade of a low-pass filter and a high-pass filter. The role of the high-pass filter is to model some of the lateral inhibitions that occur between the photoreceptors and between the ganglion cells. The low-pass filter models the "imperfections" of the optical system.

It can easily be observed that the brightness of an object remains fairly constant despite very large changes in illumination. For example, we are able to maintain the appropriate brightness ranking for a piece of coal and a sheet of white paper when observed in direct sunlight and under normal indoor illumination. The white paper seems to be equally bright under both viewing conditions, and, in fact, the white paper under indoor illumination is perceived as brighter than the piece of coal under outside illumination, even if the latter reflects more light to the viewer. One way of modeling this brightness constancy phenomenon is to place the logarithmic process between the low-pass filter and the high-pass filter. This is illustrated in Figure 3.3a, which shows the intensity profile of two images, one being six times brighter than the other. Following the observation mentioned earlier, a human observer should not be able to distinguish the two images even if the intensity profile is very different. If, however, the input were first processed nonlinearly by a logarithmic function, Figure 3.3b would result. The size of the rectangular pulse is identical in both case, and this is all that would remain after the ensuing high-pass operation. Hence, both

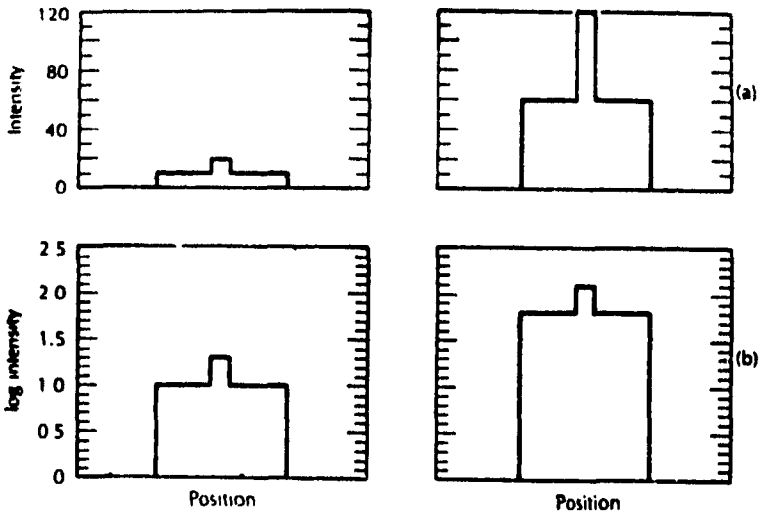

Figure 3.3: A demonstration of the phenomenon of brightness constancy. The intensity profile on the right-hand side of (a) is six times that of the one on the left. If both of these are processed logarithmically, the result is (b), in which it is observed that the two patterns are identical except for the reference level (zero frequency component). (*From T.Cornsweet, "Visual Perception", Academic Press, New York, 1970, p335.*)

objects would be perceived to exhibit brightness constancy.

Brightness perception can be modeled, therefore, by a one channel structure, cascading a low-pass filter, a logarithmic like compression, and a high-pass filter. This model is well known and is called the *multiplicative model.* Another model, called the photoreceptor visual model, use a feedforward two channel structure. The interested reader is referred to [30] for a comparison of these models.

3.2 Subjective scalar quantization

The brightness perception of the HVS is modeled based on a set of psychophysical experimental results, one of which is the contrast sensitivity curve or visibility threshold,

obtained by the just-noticeable-difference experiment. The main procedure of this experiment is to record the size of the just-noticeable-difference steps while varying the background intensity and adjusting the intensity of a small foreground patch to be just visible. The difference in intensity between the just visible test patch and the background intensity is called the visibility threshold. The background intensity is varied through the whole range of visible intensities. This experiment determines the sensitivity of the HVS to a small change in intensity for all possible nominal intensities.

This test is performed with a constant background, but most pictures contain a complex, rather than a uniform, luminance background. It is important, therefore, to know how the visibility threshold of a test stimulus changes when it is viewed in the vicinity of large spatial or temporal changes in the luminance of the background. It is well known that there is a reduction in the just noticeable visibility stimuli, i.e., an increase in the visibility threshold, caused by spatial or temporal nonuniformity of the background. This is referred to as *masking* of the test stimuli by a nonuniform background. The test stimulus is usually a small, near threshold stimulus, whereas the masking pattern is well above the threshold of visibility. *Spatial masking*, i.e., the reduced visibility of a test stimulus on both sides of a large change in the background luminance (e.g., a sharp edge), has been known for quite some time. The visibility threshold of the test stimulus increases rapidly as the test stimulus is brought closer to the sharp edge; however, the spatial masking effect decreases as the height of the edge is decreased. *So the human is much more sensitive to quantization noise in regions of an image having a slowly varying background.* This phenomenon has been measured and results presented in [26]. Subjective curves can be used to design a subjective differential pulse code modulation (DPCM) scalar quantizer, where the quantization

levels are not distributed uniformly, but according to the subjective curve so that each quantization interval is given equal subjective importance.

Another similar procedure was designed in [6]. This procedure attempts to measure the subjective magnitude of the test stimulus (e.g., quantization noise) when it is above visual threshold. At every picture element, a spatial activity function, consisting of a weighted sum of horizontal and vertical gradients at neighboring pixels, is evaluated. Test conditions are first set up by adding a random noise of known power only to picture elements where the magnitude of the spatial activity lies in a certain narrow range. The subjective value of this noise is then determined by comparison with a reference picture in which white noise is added over the whole picture and varied in power until it appears equal in quality to the test picture. The noise visibility function $V(x)$ is then defined as the ratio of the white noise power in the subjectively equivalent reference picture to the power of the noise added to the test picture only at those pixels where the spatial activity lies in an incremental range around x. The obtained subjective curves account for two factors: the decrease in noise visibility near spatial detail, and the fact that, in most pictures, few pixels have high spatial detail, thereby reducing the overall subjective importance of those pixels. These results were also used to design subjective scalar quantizers.

3.3 Distortion Measures

In source coding, the squared-error is by far the most often used distortion measure. Most of the time, it is for historical reasons since everyone has been using it, but why? In this section, we will provide some insight as to why the objective squared-error criterion is so ubiquitous and discuss its weaknesses. We then present an alternative

which offers a degree of subjectiveness in the distortion measure.

3.3.1 Squared-error distortion measure

We saw in Chapter 2 that vector quantization is a technique that evolved from the multidimensional generalization of PCM. Researchers have been using the squared-error criterion (2-norm) for vector quantization principally because of its tractability and ease of computation. Further, the squared-error criterion has an intuitive appeal, as it represents the Euclidean distance between two points. In signalling, where waveforms are often corrupted by white noise, such a distance measure yields optimal results; however, problems arise when source coding and data compression is the objective. Then, distortion is not introduced by an external source anymore, but by the quantization process of the encoder itself. This suggests that, since we have some control on how the distortion is introduced, we should use a distortion measure jointly tailored to the source signal, the encoder, and the end user. Even in light of this, the squared-error continues to be extensively used in most source compression applications. For small distortion levels, it still gives good results, but for larger compression ratios - and larger distortions - the squared-error can be very good or very bad, depending on the source signal, the encoder-decoder characteristics, and the ultimate user or system to which the compressed signal is delivered. Designing an optimal distortion measure that meets all these requirements is quite difficult, but it is desirable to include at least some of them to design a better distortion measure.

When applied to vector quantization, the squared-error criterion has an advantage over other measures; when running the LBG algorithm, the centroid calculations simply become the expectation of the given group of vectors. Thus, and the fact that centroid calculations for any distortion measure which is not quadratic in nature

become extremely complex, have prevented researchers from using other measures. Unfortunately, when applied to the vector quantization of images, the squared-error criterion introduces an adverse effect that is called the *blocking effect.* What happens is that minimum mean squared-error (MMSE) vector quantizers are very good at encoding regions of the image where there is a minimal amount of visual activity. Such regions usually occur in the background or on large flat areas of the foreground and are regions that usually convey little information to the human viewer. The MMSE VQ fails miserably in the encoding of sharp connected edges and very fine details of the image. The blocking effect is easily noticed on diagonal edges, where it gives rise to a staircase instead of a smooth continuous edge.

Considerable insight can be gained as to how this happens, by looking at Equation 2.11, reproduced below:

$$E[\mathbf{W}_\mathbf{x}\mathbf{x}] = E[\mathbf{W}_\mathbf{x}Q(\mathbf{x})], \tag{3.2}$$

where $\mathbf{W}_\mathbf{x}$ is the weighting matrix, $\mathbf{x}$ is an input vector, and $Q(\mathbf{x})$ is the codevector chosen by the quantizer Q to be the best representation of the input vector $\mathbf{x}$. If $\mathbf{W}_\mathbf{x}$ is the identity matrix, as it is for the squared-error, we obtain

$$E[\mathbf{x}] = E[Q(\mathbf{x})] \tag{3.3}$$

which shows that MMSE quantizers are optimized on the first moment (mean) of the training sequence. Furthermore, the squared-error criteria does not take into account the interactions between pixels and the fact that some pixels carry more information to the HVS than others. As mentioned in Section 3.1.1, the HVS uses to great extent the neighboring pixels in its low level processing and shows different sensitivity to distortion for different intensity levels and neighboring background activity. The MMSE criterion is, therefore, not advisable for VQ of images. As seen

in Section 2.2, the quadratic distortion measure allows for both adaptation on the input signal through $\mathbf{W}_{\mathbf{x}}$ and relatively easy centroid calculations. We believe it is possible to use the quadratic distortion measure to help reduce the blocking effect and improve the subjective quality of the coded images

For most test images, the training sequence consists of a large collection of input vectors with a low variance and with varying average intensities, and a smaller collection of input vectors with high visual activity which comprises vectors such as sharp edges, shading edges, and high variance textures. When using a MMSE vector quantizer and the LBG algorithm to design the codebook, we observe that a significant portion of the codevectors are allocated to represent the slowly varying vectors and much fewer to represent vectors with more details. At first, such a distribution of the codevectors may seem reasonable, since flat areas of the image are more sensitive to quantization noise than those areas where more background activity is occurring. Hence, to have more vectors to represent noise sensitive portions of the image seems desirable; however, this does not consider the fact that individual vectors have different subjective meaning. It is true that edges can sustain higher levels of quantization noise before any severe subjective impairments occur, but actual effects depend strongly on the type of distortion that is introduced. In the experiments where distortion was introduced in edges, only the height of the edge was distorted, and the orientation and continuity of the edge was preserved. In the LBG algorithm, the clustering process for a MMSE VQ will preserve all attributes of edges, if the input vectors in the partition of the training sequence which is associated to a codevector all have similar edge characteristics (height, intensity on both sides, position, and orientation). For a reasonably small number of codevectors (e.g., 256), this is very unlikely to happen for most training sequences and, even if a larger number of

codevectors is used, the small variations in edge height and position combined with the averaging (expectation) process will result in a codevector with a slightly higher variance than usual, but missing the correct subjective edge characteristics. In other words, the averaging process enforced by the squared-error attempts to remove edge information from the vectors and to smooth the image, a procedure which is quite contrary to a good subjective encoding process. Apart from contours of the image, almost everything else, and including high variance textures with the exception of very fine details, can be reproduced fairly well with a MMSE VQ.

It should be clear that a crucial step in this research is to design better codebooks than those that are obtained with the MMSE criterion. In fact, with a subjectively good codebook, even a MMSE encoder (i.e., with a squared-error nearest neighbor search) could yield acceptable results. The key to a good vector quantizer is to design a codebook in which the codevectors are representative of the training sequence and carry a lot of information to the viewer. One possible approach is to use classified vector quantization, but this technique only refines the vectors by using more appropriate subsequences and avoids tackling the problems of the squared-error distortion measure.

A fact that further complicates our task to design improved codebooks is the absence of globally optimum methods for the design of the codebook, which force us to use iterative methods like the LBG algorithm. This algorithm does not give control, during the design process, over the number of codevectors that should be used. This choice must be made prior to the first iteration, hence we have to define the number of codevectors for which the encoder will be designed before knowing its performance. During the design of the codebook, we have very little control on the distortion level, and it is difficult to design a procedure that will help reduce the

subjective impairments of the coded image.

A new codebook design procedure is proposed in [25], where a bottom-up approach is taken instead of the top-down approach of the LBG algorithm. This algorithm starts with the whole training sequence and iteratively merges the two closest vectors as computed by the distortion measure. The algorithm continues to merge vectors until the desired number of vector is reached or until further merges would introduce intolerable distortion. The details and implementation of this technique are rather complex and its development falls beyond the scope of this thesis. In this work, we use the LBG algorithm in all tests.

In order to design a good distortion measure, we need to look into the inherent structure of images and try to understand what is important and what is not for our image comprehension and appreciation. Real life scenes contain a lot of information and are complex in nature. A human observer can rapidly detect what is the important information that is conveyed. Furthermore, when looking at a digitized reproduction of the real life scene, it is possible to easily detect the deteriorations based on a few key features. The most important feature is that contours of the computer image be well defined, in the sense that their position, orientation, and height be the same as that of the original. It is also extremely important that edges be connected in the same smooth manner as in the original. Contrast, which is the difference in visibility between textural elements of the image, is also an important factor. If it is possible to easily differentiate adjacent textures, then the contrast of the original has been preserved (provided that the same discrimination is possible in the original scene). Finally, if the accuracy of the very fine details is preserved, then the reproduction is generally considered acceptable. MMSE Memoryless VQ can only reproduce contrast with good fidelity. The contours and the fine details are difficult

to reproduce. Both of these characteristics can be described using edge primitives and image processing techniques (small edges for details and connected edges for contours). In the next section, we propose a distortion measure which attempts to improve the representation of edges.

The role of the distortion measure in the design of a VQ cannot be underestimated. It is with that measure that separation of the training sequence into clusters is done. It is used in the computation of the centroids to form the new codevectors. In fact, the subjective quality of the codebook is intimately tied to the distortion measure and to the training sequence. Unfortunately, it is very hard to design a distortion measure optimized to the low level processing of the HVS, as we do not yet fully understand it. At this point, a combination of what we do know and intuition must be employed in our use of a quadratic distortion measure.

3.3.2 Edge based distortion measure

Whereas humans understand edges as lines separating contours, in the image processing field an edge is defined as a distinct change in magnitude of the intensity level of an image. These two definitions are not similar since the mathematical operator computes abrupt changes in the intensity signal at the pixel level and the concept of an edge is more of a connected boundary between objects. Therefore, we make the distinction between these two concepts by defining an edge to be a transition in an image, and a line (or contour) to be the connection of many edges to form an object boundary. The edge can have a height or a strength depending on how sharp the transition is. Lines are more binary in nature, a pixel is either on a line or it is not. In this sense, edges are what is often referred to as edge primitives and can be used to perform image segmentation as well as texture description. There has been

extensive research in the last twenty years performed on mathematical operators that could produce information about lines and edges. The processing to obtain this information is performed at a fairly low level in the HVS. For instance, the receptive fields in the ganglion cells compute the primitive edge elements that are combined in the brain to obtain more precise information such as strength, orientation, thickness, length, and curvature of contours. Edge information, being computed so early in our visual pathway, is believed to be the most important image primitive on which we base our recognition process. For example, if an artist draws a person he sees on a picture using only a few lines, we will be able to recognize the person, even though the amount of data contained in the drawing is seemingly far less than that in the picture. This would imply that lines carry much more information in the picture than other image primitives.

We noted that vector quantization has difficulty reproducing edges correctly. The VQ, being able to do almost everything else with distinction, should then also be designed with this criteria in mind. To correctly reproduce edges, the distortion measure that we propose severely penalizes the misrepresentation of edges. It puts emphasis on those pixels that lie near an edge and deemphasizes those that are far from it; therefore, the elements of the weighting matrix $\mathbf{W_x}$ must have a large magnitude when pixels are near an edge and small magnitude otherwise. Recalling Equation 2.7,

$$d(\mathbf{x}-\mathbf{y},\mathbf{x}) = (\mathbf{x}-\mathbf{y})^t \mathbf{W_x} (\mathbf{x}-\mathbf{y}). \tag{3.4}$$

We observe that the diagonal elements of the matrix $\mathbf{W_x}$, w_{ii}, represent the weights that multiply $(x_i - y_i)^2$. If we assume that the pixel errors $(x_i - y_i)$ do not depend on the other pixel errors $(x_j - y_j)$, for all $j \neq i$, then we can use a diagonal weighting matrix. We further assume that, for closely matched vectors $\mathbf{x}$ and $\mathbf{y}$, adjacent

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

Figure 3.4: Sobel Operator Template for the Horizontal and Diagonal Directions

pixel errors have independent signs and magnitudes. The distortion measure can be rewritten as

$$d(\mathbf{x}-\mathbf{y},\mathbf{x}) = \sum_{i=0}^{k-1} w_{ii}(x_i - y_i)^2, \tag{3.5}$$

where k is the vector dimension and w_{ii} is the weight associated with pixel x_i. Under these assumptions, we have reduced the problem of defining the matrix $\mathbf{W_x}$ to that of defining the pixel weights w_{ii}. The weights should correspond to the edge strength of individual pixels. Several simple edge operators that produce a pixel-by-pixel edge map can be found in the literature [18]. For our purposes, where only an indication of the edge strength is required, most methods are equivalent. We use the Sobel operator which is a 3×3 template that uses the center pixel and its eight immediate neighbors as shown in Figure 3.4. The Sobel operator computes the gradient of the image at the center pixel in the chosen direction and uses a small amount of averaging to reduce the sensitivity to noise. The gradient is directional and can be rotated in one of eight directions; four of them being similar to the others by symmetry. In our case, where only the magnitude of the edge is required, we compute only the horizontal and vertical components of the gradient and approximate its magnitude by combining the two components as

$$G = \sqrt{S_H^2 + S_V^2}, \tag{3.6}$$

where S_H, S_V, and G are the magnitudes of the horizontal, vertical, and resultant gradient, respectively. The values of G vary from zero to several hundreds, where

larger values represent sharper edges. The edge map of the famous picture Lenna (*cf* Figure 4.1) is shown in Figure 3.5, where bright areas represent flat portions of the image and darker areas represent edges. Note that the dark-to-bright inversion is performed only for display purposes and that in the edge map, the stronger edges have a larger value. Most sharp edges are well detected and we consider this result

Figure 3.5: Edge Map of the Image Lenna.

as satisfying for our purposes, but several smaller edges, like detailing edges of the face, are not well represented. These detailing edges, although very important for the subjective comprehension of the picture, do not produce high edge strengths with the Sobel operator.

To display the edge map in Figure 3.5, we have linearly adjusted the edge strengths so that they fill the complete gray scale of the image (0 to 255). It should be noted that, since the distortion measure is linear with respect to the edge weights w_{ii}, it is invariant to a multiplication of the weights by a constant. In other words, because the distortion measure is used to *compare* the closeness of matching of the codevectors to the input vectors and because the centroid calculations are *normalized* with respect to the weights, the encoding process is invariant to the scalar multiplication of the weights w_{ii} by a constant. Hence, the scaling that is applied to the weights in order to produce the edge map image could be used during the encoding process without affecting the results. This property permits us to transform the value of the edge strengths in a possibly nonlinear fashion and to be able to verify if the obtained weights follow the human subjective understanding of an edge map, by looking at the edge map image.

The range of values of the edge strengths from the Sobel operator can be quite large. The strength of an edge is representative of its energy, and it is commonly assumed that the energy of an edge is proportional to its subjective importance. We propose an alternative which differs from classical edge detection by making the supposition that the importance of an edge has more to do with its informativeness than its energy. This approach has been proposed in [29] and seems to yield edge maps with better information content than the classical method.

It is assumed that the information carried by an edge is related to the frequency of occurrence of edges with a similar strength. In extreme cases where the average edge strength is close to zero, the large edges contain considerable information, whereas in an image where the average edge strength is very large (like in a high textured or very noisy image), then the edges with small magnitude (or, more precisely, the

absence of edges) are those that contain the most useful information. Of course, this argument can still be applied when the relative frequency of edges is similar, but then the method yields the same performance as conventional methods. The concept of relative frequency is intimately related to that of information as defined by information theorists. Therefore, we can use the self-information measure on the relative frequency (probability of occurrence) of each edge, and use that information measure to set the value of the weights. In order to ease the process, we separate the raw data of edge strengths (real numbers) into a few edge classes. Then the probability of occurrence of each class is easily obtained by computing the histogram of the classes. If $N(c_j)$ denotes the number of occurrences of class c_j in the picture, then $H(c_j)$, the information carried by the class c_j, is given by

$$H(c_j) = -\log_2\left(\frac{N(c_j)}{N_p}\right), \tag{3.7}$$

where N_p is the total number of pixels in the picture. Optimally, the edge classes should be determined with a subjective criterion such that each class has a different meaning to the HVS. To our knowledge, such a procedure has not been presented in the literature, therefore we chose to approach the problem heuristically by studying the quality of the edge map image visually. We separate the set of all edge strengths into 32 linear classes and compute the self-information for each class. The informativeness associated with an edge class becomes the new edge weight for all pixels described by that class. The new edge map that is obtained is shown in Figure 3.6. We can observe that the details of the face and the hat are enhanced when compared with the standard edge map. Most other features of the edge map remain practically the same. The edge map follows the intuition of the HVS about important edges, almost as if the edge map was drawn by an artist. The edge map thus obtained is used to define $\mathbf{W_x}$. For a given input vector, the corresponding pixels in the edge

Figure 3.6: Normalized Edge Map of Lenna

map image form a vector which we call a *mask vector.* This mask, or equivalently $\mathbf{W_x}$, is used whenever that input vector is used. Since there is a one-to-one correspondance between pixels of an image and pixels of the mask, the mask information is provided to the encoder in the same way that the original image information is, therefore doubling the storage requirements in the encoder for each image in the training sequence.

The role of this distortion measure is two fold: penalizing the misrepresentation of pixels lying on important edges and biasing the averaging process of the clustering algorithm in the codebook update procedure towards edges. The former is the most

important in the sense that it separates the training sequence into subjectively similar subsequences, which is a crucial step of the codebook design process. The distortion measure also has a characteristic that makes it perform equally well in all areas of an image. If the input vector contains both small and large mask values, the distortion measure favors those pixels with a high mask value, but if the input vector contains only pixels with mask values of equal magnitude, then the distortion measure performs exactly as the squared-error distortion measure. This is true when all weights are small or when all weights are large. So, if the input vector is in a flat area of the original image, the proposed distortion measure performs like the squared-error distortion, which is fine, because we saw that the squared-error attempts to match the average of the input vectors. On the other hand, if the input vector lies in a very high activity region where all mask values are high, this means that the region is not traversed by an edge and represents only texture, in which case the proposed distortion measure reacts again like the squared-error measure, but we saw that textured regions did not suffer too much from the block effect. The weighted quadratic distortion measure offers the same performance as the squared-error criterion in areas where MMSE VQ performs well and attempts to improve on the edge resolution and to remove the block effect on those regions of the input image where it is most needed. Simulation results using this distortion measure will be shown in Section 4.1.

3.4 Previsualized Image Coding

In the previous section, we designed a quadratic distortion measure with a subjective criterion in mind, namely the faithful reproduction of edges. In this section, we discuss the use of brightness perception as a *preprocessor* which would increase the

performance of vector quantizers.

We saw in Section 3.1.2 how the HVS processes the light intensity of images. The multiplicative model that was presented is the result of three modules: a low-pass module to account for the "imperfect" optics of the eye and the finite spatial resolution of the photoreceptors, a companding function like the logarithm or Steven's power law to explain the large range of input intensities that humans can see, and a high-pass filter to account for the lateral inhibitions and to explain the brightness constancy phenomenon. In [28], the multiplicative model is applied to the original image and the previsualized image is vector quantized and transmitted to the receiver. The receiver generates the approximation of the previsualized image and an inverse filter of the multiplicative model is passed through the generated image to reconstruct the compressed image. The results shown in [28] are very encouraging. For simplicity, however, we use only part of the multiplicative model in this research.

The spatial resolution of vector quantized images is always less than that of the original due to the averaging process that occurs during codebook design. In this sense, the VQ is low-pass in nature and we do not require the first module of the brightness perception model to enhance its performance. The other two stages can process the input image to enhance the performance of the VQ, but we feel that the companding of the input intensity is the most helpful one since it reduces the dynamic range of the input signal in a way that is subjectively acceptable. The VQ, when operating on a smaller signal space, yields better matched compressed images; however, because of the expansion that occurs in the receiver to convert the previsualized VQ image back to the intensity domain, the quantization errors introduced by the VQ are expanded as well. Since companding is done so that the HVS is less sensitive to the new errors, the overall performance should be better

that without companding. Such a compression of the input intensity has two other advantages. First, it reduces the number of vectors in the codebook which have the same subjective visual appearance despite the fact that they are mathematically different, and, second, it enables the VQ to process edges in the same manner as the human brain sees them. In other words, special properties of the human visual system like brightness constancy and contrast sensitivity will be preserved.

The companding law that we use is Steven's power law in which we assume that the proportionality constant is unity, in which case, the brightness becomes $B = (I - I_0)^n$. The absolute visibility threshold, I_0, will be determined later. The value of n is taken as 1/3. Now, the intensity that is projected by most visual displays follow this equation

$$I = 0.299R^\gamma + 0.587G^\gamma + 0.114B^\gamma, \tag{3.8}$$

where R, G, and B are the voltages applied to the red, green, and blue ray guns, respectively. The constant γ depends on the characteristics of the display and varies between 1.7 and 3.0 with a typical value of 2.0. For the case of grayscale images, where the RGB components are equal, Equation 3.8 becomes $I = v^\gamma$. The voltage applied to each gun, v, is proportional to the value of the pixel x_i in the digitized image. The brightness associated with a pixel with value x_i is given, therefore, within a multiplicative constant, by

$$B = (x_i^\gamma - v_0^\gamma)^n = (x_i^2 - v_0^2)^{1/3}, \tag{3.9}$$

where v_0 is the smallest pixel value that can be applied to the display guns before the signal becomes invisible under normal viewing conditions. For our display, $v_0 = 10$ and $\gamma = 2$, so that $B = (x_i^2 - 100)^{1/3}$. We use this companding function and its inverse $x_i = (B^3 + 100)^{1/2}$ to transform the original image from the pixel intensity domain into the brightness domain and back to the pixel intensity domain.

Chapter 4

SIMULATION RESULTS

In this chapter, we present the results obtained when encoding images with the standard VQ techniques, and then present results that use the inherent redundancy of the picture signal. We discuss the quality of the coded images both qualitatively and quantitatively. Then, the results obtained with our proposed subjective schemes are presented and compared with those of memoryless vector quantization. Finally, our subjective schemes are combined to the most promising squared-error VQ techniques to illustrate how they interact and work together. We point out that the goal of this thesis is not to derive the most efficient VQ scheme for image coding, but to demonstrate the manner in which subjective criteria can be included in different VQ schemes.

All of the encoding schemes presented in Chapter 2 have their codebooks designed using the LBG algorithm. We now outline the experimental setup that is used.

Since it is not possible, in general, to model real world images using deterministic functions or random processes, we run the LBG algorithm on a typical training sequence of vectors. In other words, since the statistical properties of images are unknown to us, we need to use sampled statistical averages in order to obtain the required expectations. The choice of the training sequence and how it is generated

depends on the application in mind. A large training sequence composed of several images with different characteristics is used to design universal encoders, i.e., encoders that can process equally well almost any image; however, since our intent in this research is to improve the subjective quality of the compressed images, we often use smaller training sequences reduced to one image only. The encoder will then be used to encode the image in the training sequence. This helps to amplify the sometimes subtle subjective improvements. Also, we sometimes use very short training sequences using only a small part of an image, in order to get a better insight on how the proposed schemes perform. Note that in the latter case, the codebook and, consequently, the encoder become extremely specialized to the training sequence. Schemes with good performance over such small sets can yield much worse results on larger training sequences. Intuition and experience tell us, however, that it is rarely the case.

Apart from the size of the training sequence, the arrangement of vectors within the training sequence could be important. For example, schemes with memory require that successive vectors be correlated to work well. Therefore, instead of the usual left-to-right and top-to-bottom raster scan method, which can create large changes in signal characteristics in between rows, we use a slight variation in which rows are scanned alternatively from left-to-right and from right-to-left. In this way, the top-to-bottom structure is kept intact. This ensures that successive vectors in the training sequence are always neighbors in the image.

The test image that we use most often is the famous image of Lenna, shown in Figure 4.1, used internationally as a benchmark. It is a grayscale image of 512×480 pixels with a depth of 8 bits (256 graylevels). Lenna is a very good test image because it has a good dynamic range of intensity and a very good contrast. It contains many

Figure 4.1: Original Picture of Lenna

different textures with different degrees of visual activity ranging from low in the background to high in the feathers. It also contains thin lines and small details on the hat and in the eyes as well as sharp curved contour lines. All these properties put together make it difficult for a VQ scheme to reproduce the image with fidelity. Most of the time, the image of Lenna is used alone to create the training sequence; however, we sometimes use a smaller image that consists of the right eye of Lenna when local effects are desired. When we want to test a coder with more universal capabilities, we use a long training sequence of several images and use Lenna, which is not in the training sequence, as the test image.

The vectors that we use are small blocks of 4×4 pixels, creating 16-dimensional vectors. There are 15360 of those vectors in the training sequence constructed from the image of Lenna, and since we design codebooks for which the number of codevectors N varies from 2 to 512, the average number of vectors of the training sequence per codevector varies from several thousands to 30. In the latter case, the codebook becomes artificially better at encoding the image in the training sequence, an effect that shows that very long training sequences must be used if the number of codevectors in the codebook exceed 512. We often use a codebook size of 16

The results are presented in the form of pictures printed on a laser printer with the Floyd-Steinberg dithering algorithm and with tables giving the bit rates and the signal-to-noise ratios. The bit rates R are given in bits per pixel and are computed by

$$R = N_b/k = \log_2(N_t)/k, \tag{4.1}$$

where k is the vector dimension and N_b is the number of bits required to represent the number of codevectors N_t (or the number of allowed transitions for a finite state VQ). Also, we define a compression ratio as the size in bits of the original image divided by the size in bits of the compressed image. In our case, since the source image is always 8 bits per pixel, the compression ratio for a given transmission rate is given by $8/R$. The signal-to-noise ratio that we use is the peak signal-to-noise ratio (PSNR) defined by

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right), \tag{4.2}$$

where MSE is the mean squared-error.

4.1 Memoryless Vector Quantization

Memoryless MMSE VQ is the basic technique to which the other coders are compared. We present results for a variety of compression ratios with the image Lenna as a test image and as a training sequence. The resultant encoded pictures are discussed to illustrate the strengths and weaknesses of VQ. Then, the results obtained on a long training sequence are presented.

No. codevectors	Rate	Mean	Variance	MSE	PSNR
original	8	124.28	2296.5		
2	0.0625	123.69	1504.3	797.22	19.12
4	0.1250	123.45	1968.7	295.62	23.42
8	0.1875	123.62	2120.4	177.45	25.64
16	0.2500	123.81	2169.2	132.01	26.93
32	0.3125	123.83	2195 0	99.91	28.14
64	0.3750	123.78	2220.2	75.06	29.38
128	0.4375	123.78	2230.0	57.08	30.57
256	0.5000	123.76	2252.2	44.45	31.65
512	0.5625	123.79	2264.9	33.79	32.84

Table 4.1: Results for the MMSE Memoryless VQ of the image Lenna.

We designed different codebooks for the image Lenna, and the results are shown in Table 4.1. The two columns labeled Mean and Variance represent the mean and variance of the decoded image. It is easily observed that the MMSE VQ is able to reproduce the mean of the original image almost perfectly at all the rates. The variance, which is an indication of the visual activity and of the contrast, is a much better indicator of the quality of the reproduced image. The PSNR is also a good indicator. It can be observed that the PSNR increases by about $1.2dB$ each time the number of codevectors in the codebook is doubled, starting from 8 codevectors. After

512 codevectors, the increases are seen to be smaller, since the VQ is reaching the limits of its abilities.

Figure 4.2: Picture Lenna encoded with a MMSE memoryless VQ using a codebook of 256 codevectors.

Figure 4.2 shows the decoded image when the codebook size is 256 codevectors and the rate is 0.5 bits per pixel. Even if the PSNR is fairly good (more than $31dB$), we can see several impairments around the edges. The staircase effect is very noticeable on the contours of the hat, the shoulder, and the cheeks. The sudden intensity changes of the background are also less precise than the original, and a lot of information has been lost in precise details around the eyes. On the other hand, the background texture is very well reproduced. The foreground texture is also well

reproduced where the gradient is smooth, e.g., on the shoulder and in the shadings of the hat, but more distortion is introduced in high energy textures, e.g., in the feathers of the hat. The latter distortion is, however, less distracting when compared to the block effect. It is obvious from this analysis that edge definition is the hardest task for a VQ scheme. We will stress, in the ensuing tests, the improvements that were obtained over Figure 4.2 on those regions where edge definition is lacking, namely the eyes, the lips, and all major contours.

Figure 4.3: Picture Lenna encoded with a MMSE memoryless VQ using a codebook of 16 codevectors.

The picture depicted in Figure 4.3 is encoded at a rate of 0.25 bits per pixel and is shown to better illustrate how the block effect arises. Note that even at such a low

bit rate, shading is already taking place as could be seen in the background, on the hat, and on the shoulder. Also note that the feathers are relatively well reproduced, even if the block effect is also visible there.

4.2 Predictive Vector Quantization

Predictive VQ is a technique in which an estimate of the input image is made with a predictor and the residual image, obtained by subtraction, is vector quantized. Of course, the efficiency of this technique is highly dependent on the quality and the consistency of the predictor. In our case, since we are using a simple predictor, we cannot discuss the performance of predictive VQ schemes, but can discuss the role that predictive VQ might play in the development of better subjective encoders.

No. codevectors	Rate	Mean	Variance	MSE	PSNR
original	8	124.28	2296.5		
2	0.0625	123.24	1991.5	641.77	20.06
4	0.1250	123.11	2161.1	254.71	24.07
8	0.1875	123.37	2156.4	166.77	25.91
16	0.2500	123.44	2183.4	126.85	27.10
32	0.3125	123.42	2213.3	91.80	28.36
64	0.3750	123.39	2235.0	72.51	29.53
128	0.4375	123.53	2248.6	56.60	30.60
256	0.5000	123.47	2260.6	43.24	31.77
512	0.5625	123.46	2269.1	32.45	33.02

Table 4.2: Results for the MMSE predictive VQ of the image Lenna.

The performance of our simple scheme is depicted in Table 4.2. The PSNR is about $0.2dB$ greater than that of MMSE memoryless VQ at similar rates. The fact that the dynamic range of the input signal is reduced by the predictive scheme,

explains why the mean squared-error is smaller. The encoded image of Lenna, at a rate of 0.5 bits per pixels is shown in Figure 4.4. It can be observed that the contours are a little more smoother and the block effect is less visible. A more satisfying improvement, however, is the fact that the overall contrast of the image is increased. The effect might be difficult to observe on the printed images, but is easier to detect on the monitor screen. We believe that this effect is due to the greater ability of predictive VQ to represent small gradients and shading. The vectors in those regions get mapped to residual vectors of small amplitude and are encoded with more finesse and detail; therefore, smooth transitions between regions are better encoded. The subjective quality of the obtained image is greater than that of memoryless VQ, but the problem of correct representation of curves and fine details like the eyelashes remains.

In order to better illustrate the subjective improvements of predictive mean VQ, we present in Figure 4.5 the results obtained when using a smaller image as a training sequence and 4 codevectors. The result obtained with memoryless VQ is very blocky since only 4 different vectors are used to reconstruct the image. The predictive VQ image, despite looking disorganized, reproduces much more accurately the gradients and shades. The boundaries are also better defined even if the blockiness still exists. A predictive scheme, therefore, helps to improve the subjective quality, by allowing the decoder to generate more different vectors than the number of codevectors available in the codebook, with the help of a context dependent parameter which in our case is the mean of the previously transmitted block.

Predictive mean VQ is a technique which enhances the performance of the coder by removing the mean information from the signal and allowing the quantizer to concentrate on the representation of subjectively more important features such as the

Figure 4.4: Picture Lenna encoded with a MMSE predictive VQ using a codebook of 256 codevectors.

edges and the smooth shades. Subjectively, the use of a mean predictive scheme is advisable; a better predictor structure can be found in [27].

4.3 Finite State Vector Quantization

4.3.1 Vector trellis quantization

The vector trellis quantizer (VTQ) that we implemented is a finite state machine for which the next-state function is driven by a shift register. It uses the Viterbi algorithm

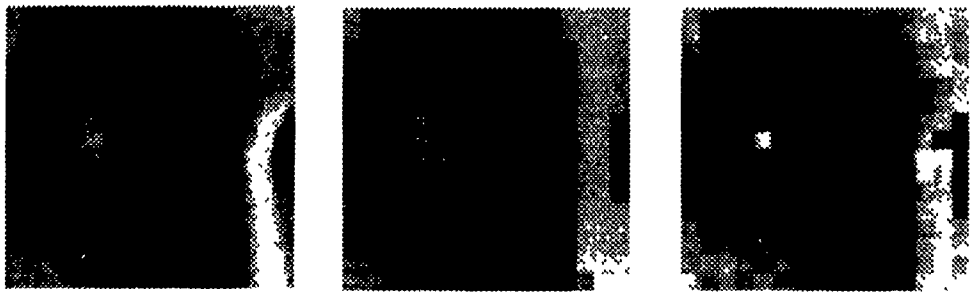

Figure 4.5: Set of pictures illustrating how a predictive scheme improves subjective performance. The image on the left is the original. The center image is the decoded image obtained with a MMSE memoryless VQ with 4 codevectors, and the image on the right is obtained with a MMSE predictive VQ with the same rate.

for optimum coding of vector sequences. The VTQ scheme includes memory in the coder and helps to remove the redundancy that exists between neighboring vectors. Although this scheme does not try to improve the subjective quality of the coded image, the resulting quality is improved because of the existence of more vectors in the codebook than in a memoryless codebook with the same rate. In other words, VTQ is a "cheap" way to increase the number of codevectors in the codebook without increasing the transmission rate; however, the price to pay is a complexity increase that depends on the number of states.

Figure 4.6 shows a typical result obtained with a vector trellis quantizer with 16 states and 16 transitions per state. The PSNR is 29.12dB. Although the block effect is still very visible, we can see a definite improvement over that of Figure 4.3.

4.3.2 Omniscient finite state vector quantization

The OFSVQ coder has memory by virtue of its states. The next-state function performs a task similar to the mean prediction scheme. Because a transition goes to the state which has a label closest to the last transmitted codevector, the transition goes to the state which has the codebook containing the most codevectors of the

Figure 4.6: Image Lenna encoded with a vector trellis quantizer with 16 states and 16 transitions per state.

same average intensity as the next input vector. This kind of specialization of state-codebooks yields very good results when the number of states is high. In fact, for a well behaved image where most successive vectors are in the same intensity range and the number of high contrast edges is small, the scheme performs very well. Figure 4.7 shows the results obtained with 16 states and 16 transitions per state.

We can observe that the picture quality is much better than that of vector trellis quantization and is close to that of memoryless vector quantization with 256 vectors. The PSNR is $29.63dB$, $2dB$ less than the memoryless VQ performance having double

Figure 4.7: Image Lenna encoded with an omniscient finite state vector quantizer with 16 states and 16 transitions per state.

the rate. Notice that the OFSVQ has the same overall number of codevectors in the codebook than the memoryless VQ, but transmits at half the bit rate. To achieve this without reducing the quality is a great accomplishment that demonstrates the usefulness of spatial redundancy removal with finite state vector quantization.

The choice of the number of states and the number of transitions per state is a tradeoff between complexity and transmission rate. Further, if the size of the codebook is restricted to a certain number of codevectors, e.g., 256, as in the previous test, then the tradeoff becomes one of choosing enough states to get sufficient memory

into the coder and enough transitions to provide codevectors with edge information and to allow the next-state function to be complete in the sense that any state can be accessed within a few transitions. The latter condition helps the delayed decision algorithm and the Viterbi algorithm to reduce the long term distortion.

The results obtained with the iterative omniscient finite state vector quantization (IOFSVQ) with 16 states and 16 transitions per states, are only marginally better than those of OFSVQ; however, the performance increases as the ratio of the number of transitions to the number of states decreases. In this case, it becomes more probable to have states which are never reached by any state transitions and are, therefore, unused; the same can be true of some transitions exiting from a not often used state. This fact reduces the efficiency of the coder and some scheme has to be designed in order to correct the situation. The IOFSVQ, because it updates the state label codebook and the state-transition function, is less prone to this kind of problems, although sometimes the very fact that it changes the state-transition table can create a similar effect. The method that we use to correct the situation is to reinitialize an unused codevector to the codevector originating from the same state that was used by the most input vectors. In the case that all the codevectors originating from a state are unused (unused state), we reinitialize the codevectors to those of the most used state. This method often solves the problem, but does it in an intuitive way and does not consider the total distortion accumulated by each state or the state which introduces the most subjective distortion. In fact, much more understanding of the next-state function process is required in order to make good choices, especially when the algorithm is using delayed decoding.

4.4 Subjective Coding

We regroup in this section all of the techniques that were designed with a subjective criterion, namely the luminance companding and the new distortion measure. The discussions of the test cases are, for the most part, qualitative, and use subjective evaluations of the entire image as well as of specific image features. The comparisons are based on what could be seen on screen, but, unfortunately, it will sometimes be difficult to see the differences on the printed copies.

4.4.1 Edge weighted quadratic distortion measure

The edge weighted quadratic distortion measure aims at improving the definition of the edges. It does so fairly well, but sometimes at the expense of another reproduced characteristic. In the discussion of the following images, we concentrate on the reproduction of edges and will discuss the overall effect in a later section.

The reproduction of edges in high quality images is difficult to discuss, although the effects can be observed by the viewer; therefore, we prese it results with a greater visible compression and that show net improvements over edge definition. Figure 4.8 shows the image of Lenna encoded with memoryless vector quantization with 16 vectors and using the weighted distortion measure. This figure can be compared with Figure 1.3. It is easily observed that the edges of the hat, particularly those at the back, are much better defined when using our proposed distortion measure. The definition of the feathers is also increased. Further, the eye regions are more visible using the new method; however, the smooth gradients are not reproduced as well as in the standard method. This is a result of pushing the codevectors towards edges and, since the total number of codevectors is the same, of removing attention

Figure 4.8: Image Lenna encoded with a memoryless VQ using 16 codevectors and the new edge weighted distortion measure

on the flatter areas. This action is consistent with the hypothesis we made that edges are subjectively more important than constant areas. In other words, even if the results of quantization are more visible in the subjective image, it still would be easier to find from the latter if Lenna is crying, smiling, or expressing sarcasm, than from a memoryless mean squared-error image. In this sense, the subjective VQ image contains more information and is, therefore, considered to be of better quality.

In order to more clearly see that the proposed method really increases the edge content of the codebook, we display the respective codebooks in Figures 4.9 and 4.10.

In these figures, each 4×4 codevector is magnified four times in each direction. The smallest visible squares represent one pixel. While there are only *two* codevectors containing distinct edges in the standard codebook, there are as many as *six* in the subjective codebook. This is a dramatic increase, considering that there are only 16 codevectors in total. For larger codebooks, the ratio of edge vectors to the total number of vector is about 1/4 for the MMSE codebook and about 1/3 for the subjective codebook. We can, therefore, affirm that an edge weighted distortion measure can improve the edge content of a memoryless VQ codebook.

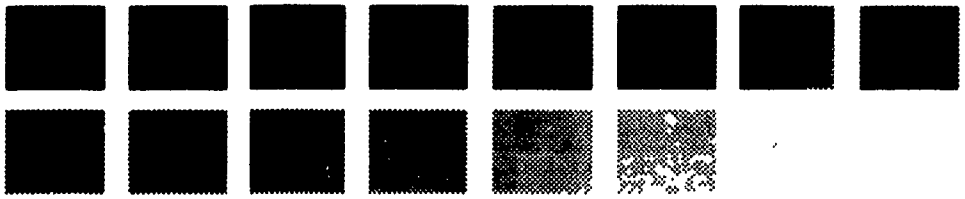

Figure 4.9: Codebook of a memoryless VQ using 16 codevectors optimized on the image of Lenna.

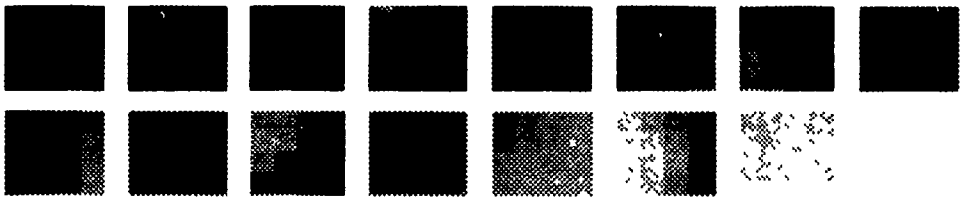

Figure 4.10: Codebook of a memoryless VQ using 16 codevectors and the edge weighted distortion measure, optimized on the image of Lenna.

Since codebooks with higher edge contents permit the reduction of the block effect, and that visibility of this effect is directly related to the dimension of the vectors, we attempted to use a larger block vector size to see how the edge distortion measure would perform in very difficult conditions. In Figure 4.11, the image is encoded with 16 codevectors of 6×6 pixels. We obtain an image in which the block effect is

extremely visible. The codebook of such an encoder is shown in Figure 4.12. The edge content of the codebook is now very low. This is due to the large number of different input vectors, varying in luminance activity, in edge height and direction and in luminance level, that each codevector must represent. This suggests that the proposed distortion measure is good at enhancing edge information from *similar* vectors, but cannot do so when they are more disparate. We expect, then, that this distortion measure would work well within the framework of classified vector quantization.

Figure 4.11: Image Lenna encoded with a memoryless VQ using 16 36-dimensional codevectors and the new edge weighted distortion measure.

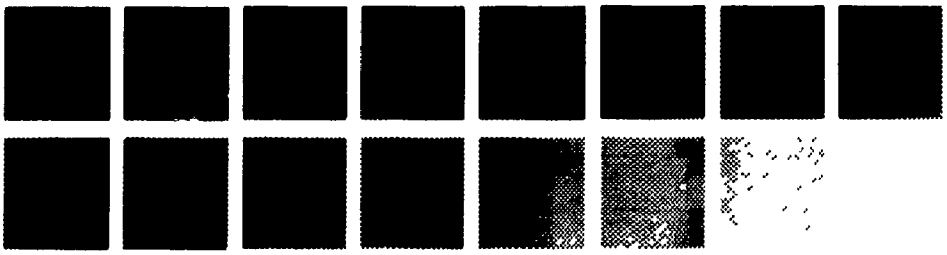

Figure 4.12: Codebook of a memoryless VQ using 16 36-dimensional codevectors and the new edge weighted distortion measure optimized on the image of Lenna.

4.4.2 Luminance to brightness companding

Brightness perception is the process which determines the relative brightness of objects. As was shown in Section 3.1, the luminance coming from objects into our eyes gets companded so that several orders of magnitude of luminance can be perceived and processed by the brain. The effect of the companding is that we become less sensitive to luminance changes at high background luminances, and more sensitive to the same luminance changes at a lower nominal luminance. By applying to images a similar companding as the one occurring in the human visual system, we attempt to force the codebook generation algorithm to concentrate on subjectively important intensity ranges

Figure 4.13 shows the image of Lenna encoded with intensity companding with a 64 codevector memoryless VQ. The difference with the reference image of memoryless MMSE VQ is very small. The newly obtained image has slightly smoother contours and fine gradients are smoother as well. The companding has two effects: it reduces the dynamic range of the input intensities, allowing the VQ to represent the average intensities with fewer vectors and to increase slightly the edge activity of the codevectors, and, since the expanding function magnifies the differences between brightness values, the medium gradient areas of the image look slightly smoother, i.e.,

Figure 4.13: Image of Lenna encoded with a MMSE memoryless VQ with 64 codevectors optimized on brightness companded input intensities

the quantization effect is less visible. This can be observed from Figures 4.14 and 4.15 which represent the codebooks designed on the companded picture and the original picture of Lenna, respectively. While there are approximately the same number of codevectors in both codebooks for the low intensities, the number of subjectively similar codevectors in the high intensity region is less in the brightness companded codebook.

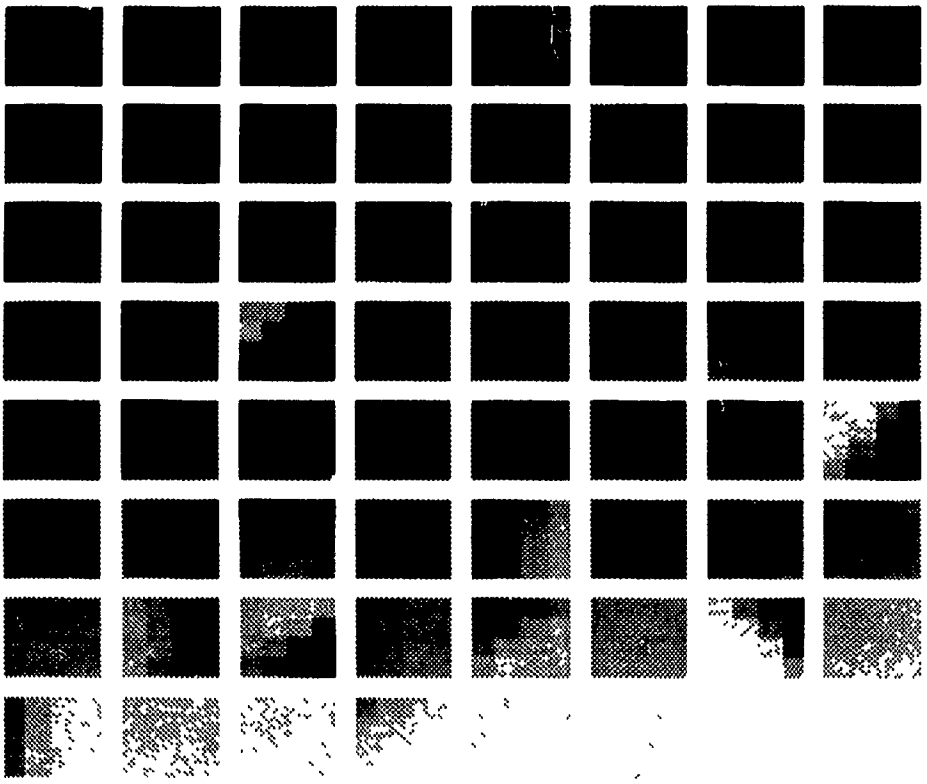

Figure 4.14: Codebook of a MMSE memoryless VQ with 64 codevectors and brightness companding optimized on the image of Lenna.

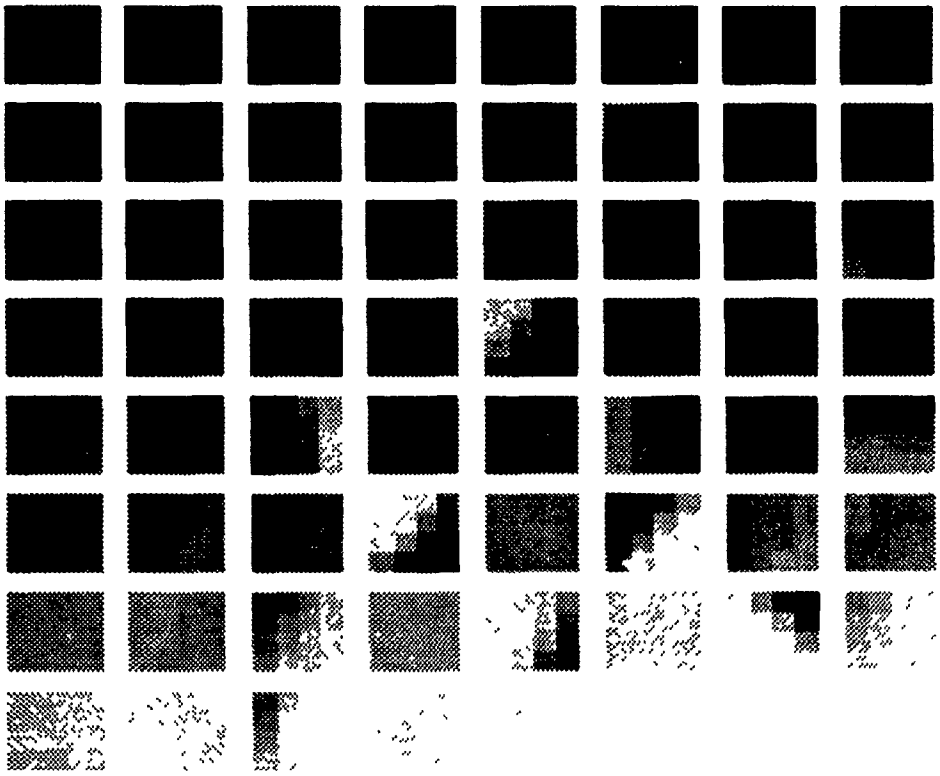

Figure 4.15: Codebook of a MMSE memoryless VQ with 64 codevectors optimized on the image of Lenna.

4.4.3 Edge weighted brightness companded encoding

In this Section, we show the effect obtained by combining two methods for subjective encoding, the edge weighted quadratic distortion measure and brightness companding. The results obtained when encoding the image of Lenna with these methods for several coding rates is shown in Table 4.3. The PSNR attained by this coder are again lower than those of the MMSE coder, but the subjective quality of the resultant images is better.

No. codevectors	Rate	Mean	Variance	MSE	PSNR
original	8	124.28	2296.5		
2	0.0625	117.87	1179.6	879.99	18.69
4	0.1250	120.64	1772.3	376.13	22.38
8	0.1875	122.04	2011.5	218.72	24.73
16	0.2500	122.46	2134.9	155.45	26.21
32	0.3125	122.51	2177.3	116.38	27.47
64	0.3750	122.37	2201.1	90.61	28.56
128	0.4375	122.56	2228.6	70.05	29.68
256	0.5000	122.50	2251.0	55.84	30.66

Table 4.3: Results for the encoding of the image Lenna with the edge weighted distortion measure and brightness companding.

Figure 4.16 shows the coded image with 64 codevectors. It can be observed that the edges are better defined and suffer much less distortion from the blocking effect, however, the subjective methods tend to artificially increase the contrast of the image so that the quantization effect is more visible around smoother areas of the image. Also, the method introduces impulsive noise around sharp edges, but this type of noise is much less annoying to the human viewer than the blocking effect, because of the contrast sensitivity curves of the human visual system. Overall, the subjectively

encoded image is of better quality even if it introduces visible side effects, since these effects create subjectively smaller impairments than MMSE encoding.

Figure 4.16: Image Lenna encoded with the edge weighted distortion measure using 64 codevectors and brightness companding

4.5 Subjective Omniscient FSVQ

In this section, we present results obtained when combining the most successful memory encoding technique with the subjective ones. Doing so, we also experiment with how the proposed subjective schemes can be incorporated in already existing VQ coding techniques.

Figure 4.17: Image Lenna encoded with the edge weighted distortion measure and brightness companding using omniscient finite state vector quantization with 16 states and 16 transitions per state.

The most successful coding technique with memory that we discussed is that of omniscient finite state vector quantization. We designed a coder with 16 states, 16 transitions per state, and using both the edge weighted quadratic distortion measure and luminance companding. The decoded image is shown in Figure 4.17. It can be observed that edges are better defined by the use of the edge weighted distortion measure. The contrast of the decoded image is also increased over that of memoryless MMSE VQ with 256 codevectors; however, in some regions, such as the shoulder and

the top of the hat, the contrast increase is too large and the quantization effect is more visible. There is also some impulsive noise near edges, but this degradation is less annoying than the blocking effect. Overall, the new technique improves the reproduction quality of edges and contours, but represents smooth areas in a coarser way than more standard techniques. Further research in this area should, therefore, concentrate on a scheme which is able to perform as well as MMSE in smooth areas and as well as the subjective quadratic distortion measure on edges. We feel that a predictive subjective classified VQ scheme would possess the required versatility and capability to realize such a goal.

Chapter 5

CONCLUSION

The main goal of this research has been to experiment on low bit rate image coding schemes, using vector quantization and attempting to increase the subjective quality of the decoded image.

5.1 Summary of Work

The problem of low bit rate image coding was studied in two distinct steps: the performance and shortcomings of *standard* vector quantization techniques proposed in the literature was discussed, and subjective methods aimed at eliminating the shortcomings of the former techniques were developed and examined. Discussions of the quality of the coded pictures, for each of the presented coding schemes, were provided along with some insights on the required codebook features and the relative importance of the obtained improvements and impairments.

The design technique of a memoryless vector quantizer was explained in detail, and a new (to image coding applications) quadratic distortion measure with adaptation to the input vector capability was proposed. For codebook design, the variation

of the LBG algorithm pertinent to this distortion measure was presented. A classical distortion measure, the squared-error, was used to encode images which became benchmarks for the ensuing research. The causes of its major source of distortion, the blocking effect, were discussed. It was shown that minimum mean squared-error vector quantizers optimize the reproduction quality of the mean of the input vectors and do not consider subjectively more important features such as edges and contours.

A subjective distortion measure was proposed in the form of an edge weighted quadratic distortion measure. It was shown that such a distortion measure can improve the faithful reproduction of edges, but also introduces other sources of distortions; however, these deteriorations, like edge and smooth gradient noisiness, are subjectively less objectionable than the blocking effect. Also, based on the brightness perception process of the human visual system, a simple prefiltering of the input image to account for the luminance companding done by the photoreceptors was proposed. This method helps to reproduce better smooth gradients in the high intensity range and does not significantly affect the performance of the coder on edge and average luminance reproduction. The combination of these two subjective methods results in a technique that can be applied to more complex vector quantization schemes.

Since typical real world imagery contains considerable spatial redundancy, vector quantizers utilizing this property were presented. Predictive vector quantization schemes were presented as a simple feedback technique to remove the correlation between input vectors. A very simple mean predictive scheme was used to show the possible improvements that can be obtained with this technique. First, because of the use of the predicted mean by the decoder, several different output vectors can be constructed with a single codevector, thus improving the coding quality at very low bit rates. At higher bit rates, this effect is less noticeable. Second, because of

the predicted mean removal, the input vectors, if the prediction values are good, are normalized to quasi-zero mean vectors. These transformed vectors retain the edge content of the original vector, but not the spatial redundancy. The codebook optimized on such training sequences concentrates more on edge features than on average intensities.

Also, two finite state vector quantization techniques were developed as another means in which to add memory to the coder. The first technique was a vector trellis encoding system using a shift register to govern the next-state function. Vector trellis quantization was shown to provide significant coding improvements over memoryless vector quantization at equal transmission rates. This gain in performance is obtained at the expense of increased complexity. The second technique is that of omniscient finite state vector quantization, where the next-state function and the state codebooks are designed to optimize the match between an input vector and the codevectors contained in the state codebook, chosen according to past transmitted reproduction vectors. For our test images, this methods yields the best performance of all coding schemes with memory. The combination of coders with memory with the proposed subjective schemes yielded good quality images at very low bit rates.

5.2 Future Work

The proposed quadratic distortion measure can be tailored to many applications. In this thesis, we chose to emphasize edges to increase the subjective information content of the coded images, but it is possible to design a weighting matrix W_x with another criterion. For example, a more complex visual model could be used to design a distortion measure that possibly could span a larger area and account for

the brightness constancy phenomenon.

Although the distortion measure is crucial to the design of a good quality codebook, classification of vectors into a set of subjectively different training sequences can also play a primordial role. For this reason, the design of a classified vector quantizer with subjective classes based on the low level processing of the human visual system and with a proper distortion measure for each class, could yield interesting results. Also, a hierarchical vector quantizer can be included in the above classified vector quantizer, allowing for greater flexibility in both transmission rate and coding performance.

The generation of a good quality codebook remains the most important factor in any vector quantization technique. Because the nearest-neighbor codebook design algorithm presented in [25] gives much more freedom during the construction of the codebook than the LBG algorithm, it can be used to control the number of codevectors needed to properly encode a training sequence. Furthermore, since the nearest neighbors could be matched with a complex subjective distance measure, this algorithm can be applied for the design of classified vector quantization with promising results.

The quadratic distortion measure can be developed for the previsualized vector quantizer of [28]. Since this vector quantizer operates in a transform domain which is representative of the visual signal input to the brain, it is appropriate to use another distortion measure than the squared-error. Along the same line of thought, the effect of an invertible prefiltering operation which would smear high frequency components of the images (since low and middle frequency components are easily reproduced by vector quantizers) can be studied. Such a transformation would map the luminance domain onto a domain suitable for near distortionless vector quantization, and should

be very robust to quantization noise. Further, a postfilter can be used at very low bit rates, when the blocking effect appears in image areas with medium range visual activity. The post filter would be similar to a sigma-filter, which is an averaging filter over a small window for pixels lying in a similar intensity range. This filter maintains very good edge quality and provides averaging to remove small amplitude noise. In this case, the sigma-filter would fuse the boundaries between two vectors which are part of the same region, so that the transition between the two vectors becomes smoother. Such filtering removes some of the adverse effects of very low bit rate image coding.

Bibliography

[1] Claude E. Shannon, "A Mathematical Theory of Communication", *Bell Syst. Tech. Journal*, vol.27, pp.379–423 and 623–656, 1948.

[2] Tom N. Cornsweet, *Visual Perception*, Academic Press, New York, 1970.

[3] James L. Mannos and David J. Sakrison, "The Effects of a Visual Fidelity Criterion on the Encoding of Images", *IEEE Transaction on Information Theory*, vol.IT-20, no.4, pp.525-536, Jul.1974.

[4] V. Ralph Algazi, "The Psycho-physics of Vision and their Relation to Picture Quality and Coding Limitations", *Acta Electronica*, vol.19, no.3, pp.225-232, 1976.

[5] Charles F. Hall and Ernest L. Hall, "A Nonlinear Model for the Spatial Characteristics of the Human Visual System", *IEEE Transactions on Systems, Man and Cybernetics*, vol.SMC-7, no.3, pp.161-170, Mar.1977.

[6] Arun N. Netravali and Birendra Prasada, "Adaptive Quantization of Picture Signals Using Spatial Masking", *Proceedings of the IEEE*, vol.65, no.4, pp.536-548, Apr. 1977.

[7] Yoseph Linde, Andrés Buzo and Robert M.Gray, "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communication*, vol.COM-28, pp.84-95, Jan. 1980.

[8] Arun N. Netravali and John O. Limb, "Picture Coding: A Review", Proceeding IEEE, vol.68, no.3, pp.366-406, Mar. 1980.

[9] Anil K. Jain, "Image Data Compression: A Review", *Proceedings IEEE*,vol.69, no.3, pp.349-389, Mar. 1981.

[10] Allen Gersho, "On the Structure of Vector Quantizers", *IEEE Transactions on Information Theory*, vol.IT-28, no.2, pp.157-166, Mar. 1982.

[11] Allen Gersho and Bhaskar Ramamurthi, "Image Coding Using Vector Quantization", *Proc. IEEE Conference on Acoust., Speech, and Signal Processing*", vol 1, pp.428-431, May 1982.

[12] Robert M.Gray and E.D.Karnin, "Multiple Local Optima in Vector Quantizers", *IEEE Transactions on Information Theory*, vol.IT 28, no.2, pp.256 261, Mar. 1982.

[13] S.P.Lloyd, "Least Squares Quantization in PCM", *IEEE Transactions on Information Theory*, vol.IT-28, pp.129-137, Mar. 1982 (reprint of 1957 paper).

[14] Lawrence C. Stewart, Robert M. Gray and Yoseph Linde, "The Design of Trellis Waveform Coders", *IEEE Transactions on Communications*, vol.COM 30, no.4, pp.702-710, Apr. 1982.

[15] Robert M.Gray, "Vector Quantization", *IEEE ASSP Magazine*, pp.4 29, Apr. 1984.

[16] Mari O. Dunham and Robert M. Gray, "An Algorithm for the Design of Labeled-transition Finite-state Vector Quantizers", *IEEE Transactions on Communications*, vol COM-33, no.1, pp.83-89, Jan. 1985.

[17] H-M. Hang and J.W. Woods, "Predictive Vector Quantization of Images", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol.33, pp.1208-1219, Nov.1985.

[18] Martin D. Levine, *Vision in Man and Machine*, McGraw-Hill, New York, 1985.

[19] Chang-Da Bei and Robert M. Gray, "Simulation of Vector Trellis Encoding Systems", *IEEE Transactions on Communications*, vol.COM-34, pp.214-218, Mar. 1986.

[20] Morris Goldberg and Huifang Sun, "Image Sequence Coding Using Vector Quantization", *IEEE Transactions on Communications*, vol.COM-34, no.7, pp.703-710, Jul. 1986.

[21] Nasser M. Nasrabadi and Robert A. King, "Image Coding Using Vector Quantization: A Review", *IEEE Transactions on Communications*, vol.COM-36, pp.957-971, Aug. 1988.

[22] Arun N. Netravali and Barry G. Haskell, *Digital Pictures: Representation and Compression*, Plenum Press, New York, 1988.

[23] John G. Proakis and Dimitris G. Manolakis, *Introduction to Digital Signal Processing*, Macmillan Publishing Company, New York, 1988.

[24] Bernard Sklar, *Digital Communications: Fundamentals and Application*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[25] William H Equitz, "A New Vector Quantization Clustering Algorithm", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol.37, no.10, pp.1568-1575, Oct. 1989.

[26] Arun N. Netravali and Birendra Prasada, *Visual Communications Systems*, IEEE Press, New York, 1989.

[27] Eve A. Riskin, E.M. Daly and Robert M. Gray, "Pruned Tree-structured Vector Quantization in Image Coding", *Proc. IEEE Conference on Acoust., Speech, and Signal Processing*", pp.1735-1738, May 1989.

[28] Thomas G. Stockham and Zhenhua Xie, "Optimal Previsualized Image Vector Quantization", *Proc. IEEE Conference on Acoust., Speech, and Signal Processing*", pp.1763–1766, May 1989.

[29] Timothy N. Topper and M.E.Jernigan, "On the Informativeness of Edges", *Proc IEEE Conference on Acoust., Speech, and Signal Processing*", pp.909-911, May 1989.

[30] Zhenhua Xie and Thomas G. Stockham, "Toward the Unification of Three Visual Laws and Two Visual Models in Brightness Perception", *IEEE Transactions on Systems, Man and Cybernetics*, vol.19, no.2, pp.379-387, Mar./Apr. 1989.

[31] Huseyin abut, *Vector Quantization*, IEEE Press, New York, 1990.