

IMAGE SEQUENCE COMPRESSION USING TRANSFORM DOMAIN
QUANTIZATION TECHNIQUES

A Master's Thesis
Presented by
Mustafa Ali TÜRKER

to
the Graduate School of Natural and Applied Sciences
of Middle East Technical University
in Partial Fulfillment for the Degree of

MASTER OF SCIENCE

in

ELECTRICAL AND ELECTRONICS ENGINEERING

35528

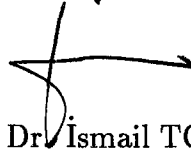
T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

MIDDLE EAST TECHNICAL UNIVERSITY

ANKARA

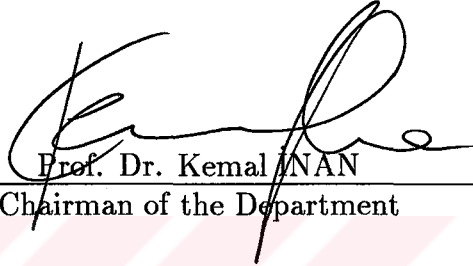
July, 1994

Approval of the Graduate School of Natural and Applied Sciences.



Prof. Dr. İsmail TOSUN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.



Prof. Dr. Kemal İNAN
Chairman of the Department

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Electrical and Electronics Engineering.



Prof. Dr. Mete SEVERCAN
Supervisor

Examining Committee in Charge:

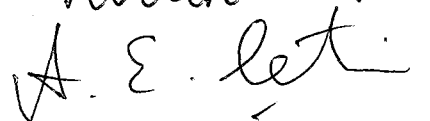
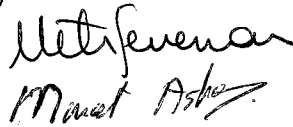
Prof. Dr. Yalçın TANIK (Chairman)

Prof. Dr. Mete SEVERCAN

Prof. Dr. Murat AŞKAR

Assoc. Prof. Dr. Levent ONURAL (Bilkent University)

Assoc. Prof. Dr. Enis ÇETİN (Bilkent University)



ABSTRACT

IMAGE SEQUENCE COMPRESSION USING TRANSFORM DOMAIN QUANTIZATION TECHNIQUES

Mustafa Ali TÜRKER

M. S. in Electrical and Electronics Engineering

Supervisor: Prof. Dr. Mete SEVERCAN

July, 1994, 152 pages.

Optimal scalar and vector quantizers that can be used on the transformed image sequence frames are examined and a new method for vector quantization is proposed. Quantizers are essential components of any compression system and they are the main contributors to the reconstruction error incurred at the decompression. As image transformation, Discrete Cosine Transform (DCT) is chosen, since it is a good approximation to Karhunen-Loève transform which perfectly decorrelates its input data.

For scalar quantization in DCT domain, nonuniform Max quantizers tailored for the distribution of DCT coefficients of transformed image data are used and integer bit allocation method of marginal analysis is chosen. Further, incorporation of human visual system properties to this optimal scalar quantization scheme is extensively discussed and simulations are performed.

For vector quantization in DCT domain, classified vector quantization (CVQ) techniques are examined and a novel scheme named as Classifier Constrained Vector Quantization (CCVQ) is introduced. This technique uses some elements of the input vector as class features and jointly optimizes the VQ codebook and the classifier, using Lagrangian methods. Therefore, it generates subcodebook for each class with optimal sizes and minimum distortion. This new method is tested experimentally and shown to have a very short execution time, although its PSNR performance is very close to full-search VQ techniques.

In order to choose the best quantization policy, optimal scalar quantization and vector quantization are compared both theoretically and practically.

As a result, vector quantization is confirmed to be superior over any scalar quantization scheme.

Finally, the intraframe coded image sequence frames with the above techniques are interframe coded to reveal the potential of compression in temporal direction.

Keywords: Image Compression, DCT, Vector Quantization, Scalar Quantization, Human Visual System.

Science Code : 609.02.08



ÖZ

DÖNÜŞÜM UZAYI NİCEMLEME TEKNİKLERİ İLE GÖRÜNTÜ DİZİSİ SIKIŞTIRILMASI

Mustafa Ali TÜRKER

Yüksek Lisans Tezi, Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Tez Yöneticisi: Prof. Dr. Mete SEVERCAN

Temmuz, 1994, 152 sayfa.

Dönüştürülmüş görüntü dizisi kareleri üzerinde kullanılacak optimal skalar ve vektör nicemleyiciler incelenmiş ve yeni bir vektör niceme yöntemi öne sürülmüştür. . Nicemleyiciler sıkıştırma sistemlerinin zorunlu parçalarından biridir ve açmada beliren yeniden oluşturma hatalarının ana nedenidir. Görüntü dönüştürülmü olarak Ayırık Kosinüs Dönüşümü (DCT), verileri ilintisizleştiren Karhunen-Loève Dönüşümüne en yakın dönüşüm olması sebebiyle seçilmiştir.

DCT uzayında skalar niceme için, dönüştürülmüş görüntü bilgisinin DCT katsayılarının dağılımına göre biçilmiş düzensiz Max nicemleyicileri kullanılmıştır ve tamsayı bit atama metodu olan marjinal analiz seçilmiştir. Bundan başka, insan görü sistemi özelliklerinin optimal skalar niceme işlemine katılmaları derinlemesine tartışılmış ve simülasyonlar gerçekleştirilmiştir.

DCT uzayında vektör niceme amacıyla, sınıflandırılmış vektör niceme (CVQ) teknikleri incelenmiş ve Sınıflandırıcı Kıstaslı Vektör Niceme adı verilen yeni bir yöntem sunulmuştur. Bu yöntem girilen vektörün bazı öğelerini sınıf belirteci olarak kullanır ve VQ kodtablosu ile sınıflandırıcıyı Lagrange yöntemleri kullanarak ortaklaşa optimize eder. Böylece, her sınıf için altkodtablosu optimal büyüklükte ve minimum hata için üretilmiş olur. Bu yeni yöntem deneysel olarak test edilmiştir ve tam-arama VQ tekniklerine çok yakın PSNR başarım olmasına rağmen çok kısa hesaplama süresine sahip olduğu gösterilmiştir.

En iyi niceme yaklaşımını seçmek için optimal skalar niceme ve vektör niceme yöntemleri kuramsal ve deneysel olarak karşılaştırılmıştır. Sonuç olarak, vektör nicemenin herhangi bir skalar niceme işlemine nazaran

üstün olduđu teyid edilmiştir.

Son olarak, yukarıdaki yöntemlerle kareîçi kodlanmış görüntü dizisi kareleri, zamanda sıkıştırma potansiyelini ortaya çıkarmak için karelerarası kodlanmıştır.

Bilim Dalı Sayısal Kodu : 609.02.08



ACKNOWLEDGEMENTS

Until the Spring of 1994, for three years, I had been in a researcher's Utopia. There were cheerful, young people everywhere. Each had different attitudes for life, but all could communicate with the same 'intellectual' language. In our laboratories, we had tremendous computer power and quite an ability to reach information sources. I could adopt any working regimen, but still could have enough tea and sweet music. I had lab-mates of my age, with whom I shared a joyous and intimate friendship.

Now, things are quite different and I get bored time to time. But then, it was Voltaire who said, all the great works of art are due to boredom.

This thesis documents the research efforts on very low bitrate image sequence compression carried out in TUBITAK Ankara Electronics Research and Development Institute. Therefore, the financial and technical support of the Institute is gratefully acknowledged.

I would also like to thank my supervisor Prof. Dr. Mete Severcan for accepting me in his group and for helpful comments throughout this work.

Finally, I am indebted to Yücel Oymak and Arkin Aydın for giving me the opportunity and encouragement to work for this Institute.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ	v
ACKNOWLEDGEMENTS	vii
LIST OF TABLES	xi
LIST OF FIGURES	xii
CHAPTER I :INTRODUCTION	1
1.1 Image Sequence Compression Techniques	3
1.2 The Choice of this Thesis Work	4
1.3 The Aim and Organization of the Thesis	5
CHAPTER II :TRANSFORM CODING	8
2.1 Definitions and Properties	10
2.2 Derivation of DCT from Fourier Cosine Transform	12
2.3 DCT as a Sinusoidal Transform	14
2.4 Properties of DCT	16
2.5 Superiority of DCT	19
CHAPTER III :DISCRETE COSINE TRANSFORM DOMAIN SCALAR QUANTIZATION	25
3.1 Scalar Quantizers for Transform Coefficients	26
3.1.1 Uniform Quantizers	27
3.1.2 Lloyd-Max Quantizers	29
3.2 Distribution of DCT Coefficients for Images	31
3.3 Optimal Bit Allocation	36
3.3.1 Rate-Distortion Theoretic Approach	39

3.3.2	Marginal Analysis Approach	44
3.3.3	Variance Estimation for Bit Allocation	46
3.4	Human Visual System in Transform Coding	47
3.4.1	Neurobiology of the Visual System	48
3.4.2	HVS Modeling	50
3.4.2.1	Simultaneous Contrast	52
3.4.2.2	Neurophysiological Reasons of Visual Illusions	55
3.4.2.3	Monochrome Vision Models	60
3.4.3	Perceptual Domain Transform Coding	62
3.5	Experimental Results	65
3.6	Conclusions	69

CHAPTER IV :DISCRETE COSINE TRANSFORM DOMAIN VECTOR

QUANTIZATION	77
4.1 Preliminaries	79
4.2 Superiority of Vector Quantization	84
4.2.1 Theoretical Advantages of Vector Quantization	85
4.2.1.1 Rate-Distortion Theory	85
4.2.1.2 High-Resolution Quantization Theory	88
4.2.2 Practical Advantages of Vector Quantization	93
4.2.2.1 Space-filling Advantage	94
4.2.2.2 Shape Advantage	94
4.2.2.3 Memory Advantage	95
4.3 Codebook Generation Techniques	95
4.3.1 Generalized Lloyd Algorithm	95
4.3.2 Pairwise Nearest Neighbor Search Algorithm	98
4.3.3 Kohonen Learning Algorithm	99
4.3.4 Stochastic Kohonen Learning Algorithm	102
4.4 DCT Domain Classified Vector Quantization	104
4.4.1 Existing Methods and Their Weaknesses	107
4.4.2 Classifier Constrained Vector Quantization	110
4.5 Experimental Work	113
4.6 Conclusion	115

CHAPTER V :INTERFRAME CODING FOR SCALAR QUANTIZATION AND VECTOR QUANTIZATION IN DCT DOMAIN	126
5.1 Interframe Coding for Scalar Quantized Image Sequence Frames .	126
5.2 Interframe Coding for Vector Quantized Image Sequence Frames .	128
CHAPTER VI :CONCLUSION	132
REFERENCES	136
APPENDICES	149
APPENDIX A :ILLUSTRATION OF VISUAL ANGLE	149
APPENDIX B :PERFORMANCE MEASURES	151



LIST OF TABLES

Table 3.1	Decision and reconstruction levels of optimum 7 bits Laplacian quantizer	33
Table 3.2	Decision and reconstruction levels of optimum Laplacian quantizers	34
Table 3.3	Decision, reconstruction levels of optimum 8 bits Gaussian quantizer	35
Table 3.4	Comparison of overall PSNR performances of the three systems which incorporate HVS with the system that does not include it.	67
Table 3.5	Comparison of the PSNR performances of the frames displayed in the following pages as an output of the block quantization system including or excluding HVS.	69
Table 4.1	Comparison of overall PSNR and required CPU time for the four different algorithms. Iterations for SKLA and KLA indicates the number of times that the complete training set is introduced to the algorithms	114
Table 4.2	PSNR performances of GLA, SKLA, KLA, and CCVQ algorithms with initial codebooks generated using PNN on frames 0,46,116, and 149 of the sequence.	114
Table 5.1	Entropies of differential DCT coefficient, which are scalar quantized into the given number of bits.	130

LIST OF FIGURES

Figure 1.1	Schematic diagram of a general communication system. . .	1
Figure 2.1	Transform coding block diagram	9
Figure 2.2	1D Discrete Cosine Transform basis vectors; N=8	15
Figure 3.1	Illustration of a symmetric, uniform, 3-bit, midrise quantizer	28
Figure 3.2	One-dimensional transform block quantization with scalar quantizers	36
Figure 3.3	Cross-section of the eye	49
Figure 3.4	(A) Multiple Mach bands and (B) double Mach bands . .	54
Figure 3.5	Intensity vs brightness distribution of (A) multiple Mach bands and (B) Double Mach Bands with brightness exag- gerated	55
Figure 3.6	Hermann grid illusion	56
Figure 3.7	An example of illusory contours	56
Figure 3.8	Lateral inhibition	58
Figure 3.9	Experimental sinusoidal gratings varying in frequency and amplitude to determine human visual system MTF	60
Figure 3.10	A complete model for human visual system	63
Figure 3.11	Frame number 0 compressed with block quantization with (a) without HVS (b) with HVS1 (c) with HVS2 (d) with HVS3.	68
Figure 3.12	PSNR performance of block quantization using optimum scalar quantizers on DCT coefficients and HVS weighted DCT coefficients over the sequence.	73
Figure 3.13	Original Frames of number (a) 61 (b) 75 (c) 88 (d) 149. . .	74
Figure 3.14	Frame number 61 block quantized (a) without HVS (b) with HVS, and frame number 75 (c) without HVS (d) with HVS.	75

Figure 3.15	Frame number 88 block quantized (a) without HVS (b) with HVS, and frame number 149 (c) without HVS (d) with HVS.	76
Figure 4.1	A DCT-VQ system for Intra/interframe coding of image sequences	78
Figure 4.2	A basic information transmission system.	86
Figure 4.3	A DCT-CVQ system for Intra/interframe coding of image sequences	105
Figure 4.4	Corresponding spatial edge patterns in (H_1, V_1) plane . . .	109
Figure 4.5	An 8-Node DCT domain direct classifier	110
Figure 4.6	PSNR performance of PNN-GLA Scheme and CCVQ with adaptive initial codebook and classifier over the sequence. .	116
Figure 4.7	PSNR performance of SKLA and CCVQ with adaptive initial codebook and classifier over the sequence.	117
Figure 4.8	PSNR performance of KLA and CCVQ with adaptive initial codebook and classifier over the sequence.	118
Figure 4.9	PSNR performance of CCVQ algorithms. Initial codebook and classifier of CCVQ-1 are generated from the sequence, while those of CCVQ-2 are ready-to-use.	119
Figure 4.10	Original frames of number (a) 0 (b) 46 (c) 116 (d) 149. . .	120
Figure 4.11	First frame of the sequence compressed with (a) VQ with GLA (b) VQ with SKLA (c) VQ with KLA (d) CCVQ. . .	121
Figure 4.12	46 th frame of the sequence compressed with (a) VQ with GLA (b) VQ with SKLA (c) VQ with KLA (d) CCVQ. . .	122
Figure 4.13	116 th frame of the sequence compressed with (a) VQ with GLA (b) VQ with SKLA (c) VQ with KLA (d) CCVQ. . .	123
Figure 4.14	149 th frame of the sequence compressed with (a) VQ with GLA (b) VQ with SKLA (c) VQ with KLA (d) CCVQ. . .	124
Figure 4.15	PSNR performance of block quantization with scalar quantization and optimal bit allocation compared to CCVQ . .	125
Figure 5.1	Probability density function of differences in non-DC DCT coefficient (0,1) which has been scalar quantized into 6 bits.	127
Figure 5.2	Probability density function of differences in non-DC DCT coefficient (1,0) which has been scalar quantized into 5 bits.	128

Figure 5.3	Probability density function of differences in DCT DC coefficient which has been scalar quantized into 8 bits.	129
Figure A.1	Visual Angle	150



CHAPTER I

INTRODUCTION

Probably no single work in this century has more profoundly altered man's understanding of communication than C. E. Shannon's "A Mathematical Theory of Communication" [1]. In the first of a series of articles [2] commemorating the 25th anniversary of the first publication of Shannon's celebrated paper in 1948, J. R. Pierce describes it as a bomb and "something of a delayed action bomb". As the 20th century dawning, the shock wave from that bomb is still spreading and the dust does not seem to get settled when the century ends.

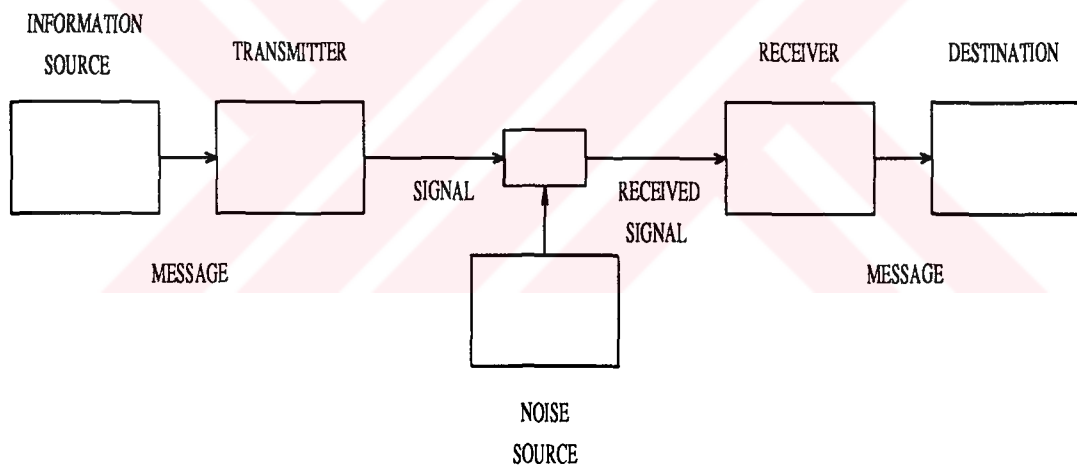


Figure 1.1. Schematic diagram of a general communication system.

Shannon observed that all communication systems can be represented by the block diagram in Figure 1.1. Then, he constructed simple mathematical models that describe the macroscopic functioning of these blocks, and how these models can work together to optimize the overall performance of the system. He also defined the concept of rate distortion function and then sketched the proof of a lossy source coding theorem. With this theorem, he introduced a new avenue of research to information theory, which is called 'source coding with a fidelity criterion' (also called 'rate distortion theory'). However, it was the year

1959, that saw the formulation of rate distortion theory as we know it today. That year was marked in the history of information theory by the appearance of another paper of Shannon [3], in which he laid out basic tenets of rate distortion theory, along with a rigorous proof of a lossy source coding theorem. Shannon's source coding theorem states that the rate distortion function of a discrete-time source evaluated at a distortion level D gives the minimum rate R in bits per source sample at which the source can be represented if the source would be reconstructed at a distortion less than or equal to D .

Rate distortion theory provides a mathematical basis and theoretical limits for the rapidly evolving branch of communication engineering commonly referred as data compression, bandwidth compression, or redundancy reduction. Information, in its many forms, has been a valuable commodity ever since man began thinking. As a result, the ability to store, access, and transmit in an efficient manner has become crucial. This is particularly true in the case of images or video (image sequences). A large number of information units (or binary units for digital pictures, bits) is required to represent even a small image. Moreover, due to the rapid advances in sensor technology and digital electronics, this number grows larger with each new generation of products. The only remedy is application of lossy source coding to images. So, the subject thrived and grew to become a well-rounded and exciting chapter in the annals of science. However practical systems had not been available until the great achievements of solid state physics in integrated circuits appeared and computers flourished. On the other hand, recently developed standard for still image and video compression, like JPEG, CCITT H.261, MPEG-I, enabled inter-workability. Therefore, engineers gained necessary incentive, and development of systems for numerous applications gained impetus.

Some outstanding applications of video compression, which will have heavy influence in daily life when accomplished are :

- High Definition Television (HDTV) : All-digital HDTV systems which have their specifications being established at the moment, both in USA and Japan will certainly use high-fidelity video compression methods to reduce their unacceptable bandwidth requirement due to their high resolution frames. Since it has high-resolution pictures, HDTV can also be expected to

impact the medical industry by enabling doctors to make remote diagnoses in emergency situations, in sparsely populated areas.

- Video telephony and videoconferencing : The visual telephone service is generally a two-way telecommunication service which uses a switched network of broadband or narrowband analogue and/or digital circuits to establish connections among subscriber terminals, primarily for the purpose of transmitting live or static pictures. Since the quality of communication between the subscribers would be increased a lot, these services are expected to cause a significant decrease in traveling expenses.
- Video surveillance systems : This a special application of one-way video transmission system which requires real-time, low-fidelity video compression for saving or transmitting the video data of a place which is needed to be monitored. In the case of compressing the video data into very low bit-rates, open-air surveillance systems can be built which would utilize the existing telephony lines, hence cabling costs would be reduced.
- Interactive video databases (video on demand) : Such data bases which would hold video and images galore using data compression, will impact a wide range of services including the travel industry, marketing of the real-estate, education and entertainment
- Video unification : The ultimate goal of image and video compression. Once standards for all types of images and video sequences are settled which could work in concord with each other, all possible video services including the ones mentioned above can be unified in a large multimedia environment.

1.1 Image Sequence Compression Techniques

Image sequence compression is achieved by excising the correlation existing both spatially, within each frame of the sequence, and temporally between the frames. Possible methods of removing this correlation can be listed as below :

- Subband coding techniques: Dividing the sequence into subbands of varying spatial frequency intervals, a different compression method can be applied

for each band, which would comply with the statistical properties of the corresponding subbands. These techniques are especially useful for progressive transmission.

- **Motion estimation/ motion compensation techniques :** Motion estimation examines the movement of pixels, blocks or objects in an image sequence for obtaining vectors to represent the estimated motion. Motion compensation, then uses this knowledge of motion, to achieve data compression by coding only the difference signal between original and estimated frames.
- **Intra/interframe coding :** It Involves compression of each frame of the sequence individually. Then temporal redundancies are removed using the compressed frame data.
- **2nd generation techniques :** Unlike the above methods, which are statistical in nature, 2nd generation techniques makes use of the knowledge on human visual system, to produce subjectively good-quality pictures at high compression ratios, mainly by preserving components of the image to which human eye is sensitive like edges.
- **Feature based techniques :** Involves the paradigm, which only encodes some crucial features of the image such as facial features for a video telephone application. Encoding may be done by using a codebook for each feature which contains a set of possible shapes for that feature. For an efficient coding such a set should contain orthogonal shapes.

1.2 The Choice of this Thesis Work

Motion estimation/ motion compensation based techniques are used in many existing standards for image sequence compression like CCITT H.261, MPEG-I, MPEG-II. However, they require a long execution time if subpixel accuracy is desired by motion estimation. If motion estimation is carried out on block by block basis with the assumption that all pixels in a block undergo the same displacement, it can account for only translational motion and fails for camera zooms, pans, object rotation, change of object shape etc., causing blocking artifacts. Subband coding is a method which has been quite settled down, since

it is one of the earliest approaches to coding. Finally, feature based techniques seems to be quite appealing since they can produce very high compression ratio. But the results are usually far from being natural.

The choice of this thesis favours intra/interframe compression techniques. If applied in the transform domain, intra/interframe coding is a kind of a divide and conquer technique, which removes the spatial correlation within subblocks of size $N \times N$, and then undertakes the problem of removing redundancy between the subblocks. Finally, temporal compression is done by interframe coding. On the other hand, inspired from 2nd generation techniques, some proper weighting functions can be used to account for human visual system properties sufficiently. When compared to motion estimation/ motion compensation paradigm, although it is more statistical in nature and seemingly more complex, it can adopt any motion estimation technique for interframe coding of the compressed data.

The Discrete Cosine Transform (DCT) is chosen in this thesis to represent the elements of each image subblock in a domain where they are almost uncorrelated.

1.3 The Aim and Organization of the Thesis

The transform coefficients which represents the image subblocks in the DCT domain have real values and must be quantized prior to quantization. Quantization is the main step in encoding which introduces the largest amount of distortion. Therefore, optimal quantizers should be constructed to achieve high performance coding. Generally there exists two types of quantizers :

1. Scalar quantizers which operates on single samples of the source.
2. Vector quantizers which operates on k -dimensional blocks (vectors) of the source.

The main purpose of this thesis is to develop scalar and vector quantizers for the quantization of image data in DCT domain. Efficacious ways for interframe coding has been extensively covered by motion estimation/ motion compensation paradigm. However, the virtue of intra/interframe compression lies in its capability to reduce spatial redundancy within each frame of the sequence. The key for such an effective intraframe coding is a good quantizer.

With this aim, firstly, optimal scalar quantizers are examined. The necessary ingredients of optimal scalar quantization, namely the information regarding the probability distribution function of the source to be quantized and an optimal bit allocation algorithm to distribute the available bits are discussed extensively. Secondly, the case of vector quantizers are inspected emphasizing the proof of the superiority of vector quantization over scalar quantization. Since a very important disadvantage of intra/interframe coding with respect to motion estimation and compensation methods is its computational burden for intraframe coding, the second most important intention of this thesis work is to develop a fast intraframe coding techniques. For that reason, classified vector quantizers are scrutinized. Finally, a classified vector quantization scheme which is called ‘classifier constrained vector quantization’ (CCVQ) is developed. This new technique of vector quantization not only accomplishes the task of optimal quantization, but also has a very short execution time of codebook and classifier generation.

The organization of the thesis is as follows :

In Chapter 2, an objective quality assessment measure PSNR is explained which will be used throughout this thesis to judge the performance of the tested compression systems using the fidelity of the output images with respect to the input.

In Chapter 3, DCT is introduced as a member of the Sinusoidal Family of transforms and the choice of DCT as the unitary transform to represent the images is justified by proving its asymptotic equivalence to the optimal Karhunen-Loève Transform for correlated input data. Also, the derivation of DCT from Fourier Cosine Transform and its relation to Fourier Transform is given.

The subject of Chapter 4 is optimal scalar quantization. The optimal uniform and nonuniform Max quantizers are explained. To be able to construct such optimal quantizers, the distribution of DCT coefficients for images is inspected by an overrunning survey through the literature. The survey reveals the contrasting predictions in the related scientific works, but finally comes up with acceptable distribution functions for the DC and non-DC DCT coefficients. Having determined the optimal scalar quantizers, to construct an optimal system, the choice of the bit allocation method of marginal analysis is justified. Further, human visual system properties are investigated to a large extent and various visual system models are given to be able to carry out a perceptual domain transform coding

scheme. The chapter also includes experimental results comparing the perceptual domain optimal scalar quantization to DCT domain optimal scalar quantization.

The 5th Chapter starts with a preliminary work to present the properties of optimal vector quantizers and norms of distortion to judge optimality. Then, the asymptotic optimality of vector quantization is explained and its superiority to scalar quantization is proved theoretically using high resolution quantization theory findings. Being contented with the supremacy of vector quantization, the method of classified vector quantization is chosen to be the best candidate for adaptive real-time intra/interframe coding systems which uses vector quantization. Later in the chapter, the reader's attention is dragged upon the suboptimalities of existing classified vector quantization methods and the novel technique of CCVQ is presented. Experimental results demonstrating the complexity reduction with the new technique with respect to some other vector quantization techniques which uses full-search codebooks are given. However, the loss of PSNR performance is showed to be only very small. On the other hand, subjective quality of the reconstructed images is demonstrated to be improved, due to edge-oriented classification. The chapter concludes with an experiment revealing the superiority of vector quantization over scalar quantization.

Chapter 6, mainly contains two simple experiments about the interframe coding of the scalar quantized and vector quantized intraframes. The reason is to exhibit the promises of interframe coding in removing the temporal redundancy. No new methods or extensive examinations of possible interframe coding techniques are presented since the aim of this thesis is rather to obtain fast and efficient intraframe coding methods.

Chapter of 7, concludes the thesis with an overall evaluation of the tested methods, proposed CCVQ scheme and directions for feature research.

CHAPTER II

TRANSFORM CODING

Transforms, particularly integral transforms, are used primarily for the reduction of complexity in mathematical problems. Differential equations and integral equations may be changed into algebraic equations whose solutions are more easily obtained, by applying appropriate transforms. In image data source processing, '2-dimensional unitary transforms' (see the following section for description) have found three major applications. Transforms have been utilized to extract features from images. For example, in the Fourier transform, the average value or "d.c." term is proportional to the average image brightness, and the high frequency coefficients indicate the sharpness and orientation of the edges within an image. Dimensionality reduction in computation is another application. Small transform coefficients may be excluded from processing operations, such as filtering, without much loss in processing performance. The third application is 'transform coding'. Transform coding denotes a procedure, in which the image is subjected, prior to coding and transmission, to an invertible transform, with the aim of converting the statistically dependent image elements to independent coefficients. Transforms are employed to represent the pictures in a 'space', in which the attributes of the pictures are not correlated. The basic property of image upon which image compression techniques rely is inter-element correlation. The neighboring picture elements of even fairly active images, i.e. those containing a reasonable amount of spatial detail, have 'on the average' similar amplitudes of intensity and chrominance. This property of images implies that, their power spectral distribution is strongly low pass in nature, thus requiring little coding capacity for transmission. The degree to which images may be compressed whilst still allowing satisfactory reproduction after storage or transmission in compressed form is, therefore, crucially dependent upon their correlation properties.

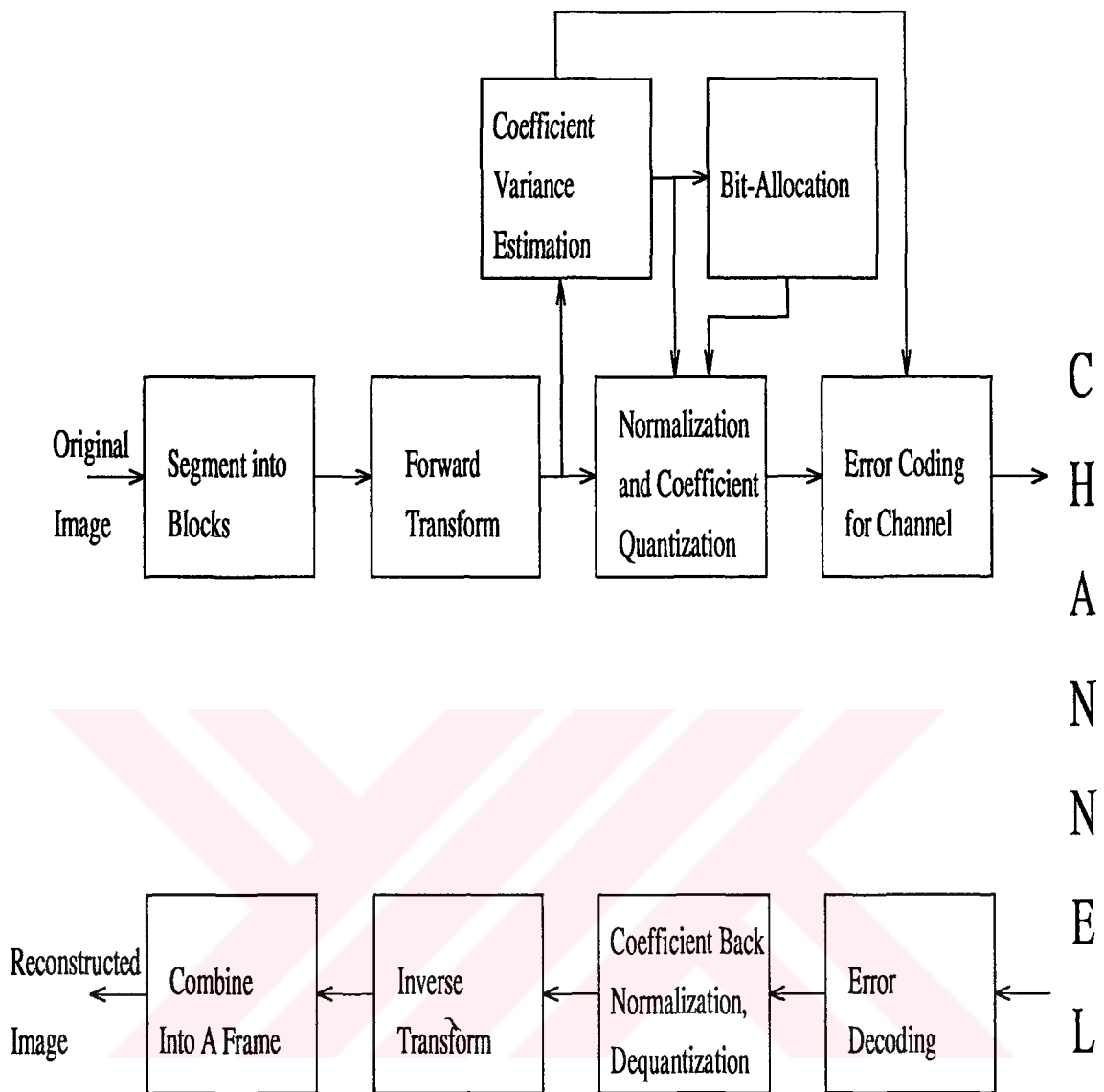


Figure 2.1. Transform coding block diagram

A general transform coding scheme is shown in Figure 2.1. It involves subdividing an $L \times L$ image into smaller, nonoverlapping $N \times N$ blocks and performing a unitary transform on each subblock. Upon redistribution of the signal energy among uncorrelated coefficients, a smaller set of relatively high energy coefficients are quantized and encoded for the channel.

When compared to spatial coding of image data, transform coding is somewhat more versatile, yet more elusive. For an image with high spatial correlation, many high - order transform coefficients will be small and may either be coded with very few bits or deleted completely. The optimal bit allocation problem for the transform coefficients has been studied extensively and optimum solutions

have been found (see Section 3.3). Although transform coding is very immune to variations in input data, it is sensitive to storage and transmission errors than spatially coded images as all coded coefficients have very ‘fragile’ information. Compared to spatial coding, this method is relatively complex and benefits greatly from the recent advances in high-speed digital hardware. Generally, transform coding gives good quality image reproductions at rates between 0.5 and 1.0 bpp range, provided that input image does not have large amounts of intricate spatial detail.

Among various transforms proposed for image data coding, the ‘Discrete Cosine Transform’ (DCT), emerges to be the best candidate to replace the Karhunen-Loève Transform (KLT) for compression applications, which is statistically optimum from the standpoint of energy compaction. DCT is also outstanding for its fast and efficient implementations employing real arithmetic (see Section 2.5).

In the following sections details about DCT will be explained and the choice of DCT as the main domain throughout this thesis will be justified.

2.1 Definitions and Properties

The Discrete Cosine Transform, introduced by Ahmed et al. [5], has been selected to be the transform which will be used throughout this chapter, and also in this thesis work. Reasons for the choice of DCT is stated in Section 2.5. In the below section the position of DCT among all transforms is stated and some of its properties are mentioned.

Wang [6] classified the existing DCTs into four types:

1. DCT-I

Forward :

$$X^{C(1)}(m) = \left(\frac{2}{N}\right)^{1/2} k_m \sum_{n=0}^N k_n x(n) \cos\left(\frac{mn\pi}{N}\right), \quad m = 0, 1, \dots, N;$$

Inverse :

$$x(n) = \left(\frac{2}{N}\right)^{1/2} k_n \sum_{m=0}^N k_m X^{C(1)}(m) \cos\left(\frac{mn\pi}{N}\right), \quad n = 0, 1, \dots, N. \quad (2.1)$$

2. DCT-II

Forward :

$$X^{C(2)}(m) = \left(\frac{2}{N}\right)^{1/2} k_m \sum_{n=0}^{N-1} x(n) \cos \left[\frac{m(2n+1)\pi}{2N} \right], \quad m = 0, 1, \dots, N-1;$$

Inverse :

$$x(n) = \left(\frac{2}{N}\right)^{1/2} \sum_{m=0}^{N-1} k_m X^{C(2)}(m) \cos \left[\frac{m(2n+1)\pi}{2N} \right], \quad n = 0, 1, \dots, N-1. \quad (2.2)$$

3. DCT-III

Forward :

$$X^{C(3)}(m) = \left(\frac{2}{N}\right)^{1/2} \sum_{n=0}^{N-1} k_n x(n) \cos \left[\frac{(2m+1)n\pi}{2N} \right], \quad m = 0, 1, \dots, N-1;$$

Inverse :

$$x(n) = \left(\frac{2}{N}\right)^{1/2} k_n \sum_{m=0}^{N-1} X^{C(3)}(m) \cos \left[\frac{(2m+1)n\pi}{2N} \right], \quad n = 0, 1, \dots, N-1. \quad (2.3)$$

4. DCT-IV

Forward :

$$X^{C(4)}(m) = \left(\frac{2}{N}\right)^{1/2} \sum_{n=0}^{N-1} x(n) \cos \left[\frac{(2m+1)(2n+1)\pi}{4N} \right] \quad m = 0, 1, \dots, N-1;$$

Inverse :

$$x(n) = \left(\frac{2}{N}\right)^{1/2} \sum_{m=0}^{N-1} X^{C(4)}(m) \cos \left[\frac{(2m+1)(2n+1)\pi}{4N} \right] \quad n = 0, 1, \dots, N-1 \quad (2.4)$$

where

$$k_p = \begin{cases} \frac{1}{\sqrt{2}} & \text{when } p = 0 \text{ or } N \\ 1 & \text{when } p \neq 0 \text{ and } N. \end{cases}$$

In the above equations, $x(n)$ is the input (data) sequence and $X^{C(i)}(m)$ is the transform sequence corresponding to DCT- i , where $i = 1, 2, 3, 4$. Note that the normalization factor $\sqrt{(2/N)}$ that appears in both the forward and the inverse transforms can be merged as $2/N$, and moved either to the forward or to the inverse transform. The k_p factor appears in the first order coefficients of the transforms, so as to make the magnitude of the first basis vector of the transform

unitary. Observe from the above equations that un-normalized first basis vector elements have the magnitude $\sqrt{2/N}(\cos(0) = 1)$, and so the sum of their squares is $N \times (\sqrt{2/N})^2 = 2$. Since the sum of the squares of all elements in a basis vector must equal to unity (see Section 2.4), each element in the first order must be normalized by $1/\sqrt{2}$.

DCTs can be derived from Fourier Cosine Transform (FCT) which is nothing but the Fourier Transform (FT) of the even symmetrical version of the signal. Next section is devoted to a short derivation of DCT from FCT. DCT can also be viewed as a member of the vast family of sinusoidal transforms, which have very appealing properties for image coding, hence it is worth to cover it in an extra section. Following this, important properties of DCTs, from the point of view of image coding will be given.

2.2 Derivation of DCT from Fourier Cosine Transform

It is tempting to treat DCTs as discretized approximations of continuous FCT. However, this would be quite wrong, since in digital coding one deals with samples, measurements, and time instants. The continuum is merely an idealization to permit the use of calculus. So, although such a derivation of DCT from FCT gives quite an insight, the various properties of DCT should be dealt separately in the discrete world. Keeping this in mind, the derivation of FCT from FT and DCT from FCT follows:

Given a function $x(t)$ for $-\infty < t < \infty$, its Fourier Transform is

$$\begin{aligned} F[x(t)] \triangleq X(\omega) &= \left(\frac{1}{2\pi}\right)^{1/2} \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \\ F^{-1}[X(\omega)] \triangleq x(t) &= \left(\frac{1}{2\pi}\right)^{1/2} \int_{-\infty}^{\infty} X(\omega)e^{j\omega t} d\omega \quad . \end{aligned} \quad (2.5)$$

where $\omega = 2\pi f$ is the radian frequency and f is the frequency in Hertz.

If $x(t)$ is defined only for $t \geq 0$, we can construct a function $y(t)$ such that

$$y(t) = \begin{cases} x(t) & t \geq 0 \\ x(-t) & t \leq 0 \end{cases} \quad (2.6)$$

then,

$$F[y(t)] = \left(\frac{2}{\pi}\right) \int_0^{\infty} x(t) \cos(\omega t) dt \quad . \quad (2.7)$$

Now, FCT of $x(t)$, $t \geq 0$ can be defined as follows

$$\begin{aligned} F_c[x(t)] &\triangleq X_c(\omega) = \left(\frac{2}{\pi}\right) \int_0^\infty x(t) \cos(\omega t) dt \\ F_c^{-1}[X_c(\omega)] &\triangleq x(t) = y(t) = \left(\frac{2}{\pi}\right) \int_0^\infty X_c(\omega) \cos(\omega t) dt \end{aligned} \quad (2.8)$$

Observe that, from (2.8), it is clear that $F_c[\cdot] \equiv F_c^{-1}[\cdot]$. It can be seen, again from (2.8), that FCT has a kernel given by

$$K_c(\omega, t) = \cos(\omega t) \quad (2.9)$$

up to a normalization constant. Let $\omega_m = 2\pi m \delta f$ and $t_n = n \delta t$ be the sampled angular frequency and time, where δf and δt represent the unit sample intervals for frequency and time respectively; m and n are integers. If we further let $\delta f \delta t = 1/(2N)$, where N is an integer, Equation (2.9) can be rewritten in the form

$$K_c(\omega_m, t_n) = \cos(2\pi m n \delta f \delta t) = \cos \frac{\pi m n}{N} \triangleq K_c(m, n) \quad (2.10)$$

Equation (2.10) represents a discretized FCT kernel. If this kernel is regarded as elements in an $(N + 1)$ by $(N + 1)$ transform matrix, denoted by $[A]$, then mn^{th} element of the matrix is

$$[A]_{m,n} = \cos\left(\frac{\pi m n}{N}\right) \quad m, n = 0, 1, \dots, N. \quad (2.11)$$

When $[A]$ is applied to a column vector $\vec{x} = [x(0), x(1), \dots, x(N)]^T$, the vector $\vec{X} = [X(0), X(1), \dots, X(N)]^T$ is obtained such that,

$$\vec{X} = [A]\vec{x}, \quad (2.12)$$

where,

$$X(m) = \sum_{n=0}^N \cos\left(\frac{\pi m n}{N}\right) x(n) \quad m = 0, 1, \dots, N. \quad (2.13)$$

This type of DCT was first reported by Kitajima in 1980 [7], and was named as Symmetric Cosine Transform (SCT). Other types of DCT can be formed by modifying SCT by scaling, or shifting operations.

With the inspiration from above derivation, DCT can also be obtained from DFT directly in the discrete domain.

Let $\{x(n)\}$ be a given sequence, then an extended sequence $\{y(n)\}$, even symmetric about the point $(2N - 1)/2$, can be constructed so that,

$$y(n) = \begin{cases} x(n) & n = 0, 1, \dots, N - 1, \\ x(2N - n - 1) & n = N, N + 1, \dots, 2N - 1. \end{cases} \quad (2.14)$$

Then, DFT of $y(n)$ is,

$$Y(m) = \sum_{n=0}^{2N-1} y(n)W_{2N}^{nm} \quad m = 0, 1, \dots, 2N - 1. \quad (2.15)$$

where $W_{2N} \triangleq \exp(-j2\pi/2N)$. This can easily be reduced to the form (see [8], page 49, for details of the derivation),

$$Y(m) = \sum_{n=0}^{N-1} x(n)[W_{2N}^{nm} + W_{2N}^{-(n+1)m}], \quad m = 0, 1, \dots, 2N - 1. \quad (2.16)$$

with the use of symmetry property of $y(n)$. If both sides of Equation (2.16) is multiplied by the factor $\frac{1}{2}W_{2N}^{m/2}$, one directly obtains,

$$\frac{1}{2}W_{2N}^{m/2}Y(m) = \sum_{n=0}^{N-1} x(n) \cos \left[(2n + 1) \frac{m\pi}{2N} \right], \quad m = 0, 1, \dots, N - 1. \quad (2.17)$$

Comparing Equation (2.17) with the DCT-II (definition in (2.2)), it can be seen that Equation (2.17) is the DCT-II of the N -point sequence $x(n)$, except for the scaling factors in (2.2). So, as $\{Y(m)\}$ is the $2N$ -point DFT of $\{y(n)\}$ (symmetric version of $\{x(n)\}$), Equation (2.17) shows that, for $m = 1, 2, \dots, N - 1$, the $2N$ -point DFT of the even-symmetric version of an N -point sequence is same as the N -point DCT of the sequence, except for a scaling factor. Note that, the scaling factor in Equation (2.17), corresponds to a simple all-pass filter. This result also implies that N -point DCT can be implemented via $2N$ -point DFT. Moreover, when $\{x(n)\}$ is also real, Haralick [9] showed that $2N$ -point DFT of its even-symmetric version can be calculated via two N -point DFTs, thus reducing the complexity of the implementation of DCT a good deal. In the later years, many other methods to reduce the complexity had been proposed [10].

2.3 DCT as a Sinusoidal Transform

DCT is one of an extensive family of ‘sinusoidal transforms’, documented by Jain [11]. In their discrete form the basis vectors of such transforms consist of sampled sinusoidal and cosinusoidal functions. Observe Figure 2.2, to see that the basis vectors for $N = 8$ are generated by sampling at 8 points, a set of cosine waves of increasing frequency.

Jain [11] showed that the well-known discrete Fourier, cosine, sine, and

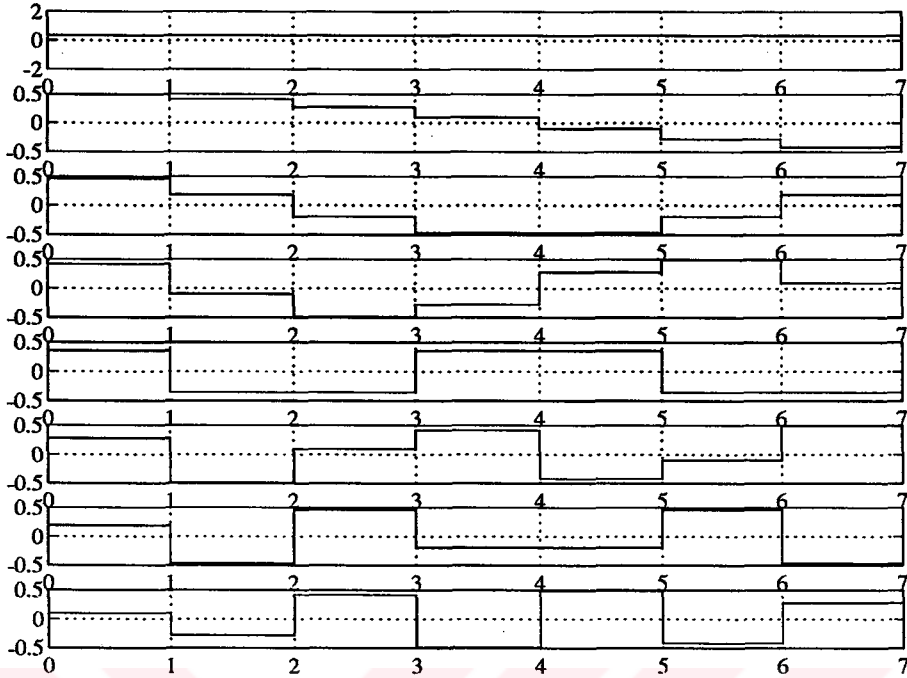


Figure 2.2. 1D Discrete Cosine Transform basis vectors; $N=8$

Karhunen-Loève (for 1st-order stationary Markov processes) transforms are members of this family. To define the sinusoidal transforms family, consider the parametric family of matrices

$$[J] = [J(k_1, k_2, k_3, k_4)] = \begin{bmatrix} 1 - k_1\alpha & -\alpha & 0 & \dots & \dots & 0 & k_3\alpha \\ -\alpha & 1 & -\alpha & 0 & \ddots & \ddots & 0 \\ 0 & -\alpha & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 & -\alpha & 0 \\ 0 & \ddots & \ddots & 0 & -\alpha & 1 & -\alpha \\ k_4\alpha & 0 & \dots & \dots & 0 & -\alpha & 1 - k_2\alpha \end{bmatrix} \quad (2.18)$$

This is a variation of the well-known tridiagonal Jacobi matrix (a matrix $[J]$ is called a Jacobi matrix if $[J]_{m,n} = 0$, for $(m - n) \geq 2$) if $k_3 = k_4 = 0$. For $k_3 = k_4$, J is a symmetric matrix and with proper choice of α it would be admissible as a positive definite covariance matrix. Each covariance matrix guarantees an associated set of complete orthonormal eigenvectors. Such eigenvectors are obtainable

by solving the equation

$$[J]\vec{\phi}_m = \lambda_m \vec{\phi}_m, \quad 1 \leq m \leq N \quad (2.19)$$

for different sets of k_i , $i = 1, 2, 3, 4$. Then $\vec{\phi}_m$'s are the corresponding m^{th} column vectors of the sinusoidal transform matrix $[\Phi]$. So the family of unitary sinusoidal transforms can be defined as the class of complete orthonormal sets of eigenvectors generated by those J matrices which are admissible as covariance matrices, i.e. , those which have such $\alpha, k_1, k_2, k_3 = k_4$ that they become positive definite. For the particular choice of

$$\begin{aligned} \alpha &\triangleq \frac{\rho}{1 + \rho^2} \\ k_1 &= k_2 = 1 \\ k_3 &= k_4 = 0 \end{aligned}$$

the eigenvectors are the basis vectors of N -point DCT.

Note that through derivation of the family of matrices $[J]$ is too tedious. The interested reader may refer to page 117 of [12], and [11] for details.

2.4 Properties of DCT

Two-dimensional DCT is a 'unitary transform' ([13], [14]). A unitary transform is a specific type of linear transformation in which the basic linear operation,

$$F(m_1, m_2) = \sum_{n_1=1}^N \sum_{n_2=1}^N f(n_1, n_2) A(n_1, n_2; m_1, m_2) \quad (2.20)$$

is exactly invertible,

$$f(n_1, n_2) = \sum_{m_1=1}^N \sum_{m_2=1}^N F(m_1, m_2) B(n_1, n_2; m_1, m_2) \quad (2.21)$$

and the forward and inverse transform kernels

$$\{A(n_1, n_2; m_1, m_2)\} \text{ and } \{B(n_1, n_2; m_1, m_2)\},$$

which are sets of basis functions satisfying the following conditions of 'orthonormality' and 'completeness' :

$$\begin{aligned} \sum_{m_1} \sum_{m_2} A(n_1, n_2; m_1, m_2) A^*(n'_1, n'_2; m_1, m_2) &= \delta(n_1 - n'_1, n_2 - n'_2) \\ \sum_{m_1} \sum_{m_2} B(n_1, n_2; m_1, m_2) B^*(n'_1, n'_2; m_1, m_2) &= \delta(n_1 - n'_1, n_2 - n'_2) \end{aligned} \quad (\text{Orthonormality}) \quad (2.22)$$

$$\begin{aligned}
\sum_{n_1} \sum_{n_2} A(n_1, n_2; m_1, m_2) A^*(n_1, n_2; m'_1, m'_2) &= \delta(m_1 - m'_1, m_2 - m'_2) \\
\sum_{n_1} \sum_{n_2} B(n_1, n_2; m_1, m_2) B^*(n_1, n_2; m'_1, m'_2) &= \delta(m_1 - m'_1, m_2 - m'_2) \\
&\text{(Completeness)} \quad (2.23)
\end{aligned}$$

It can be observed that for the above conditions to hold the inverse transform kernel must be the complex conjugate of the forward transform kernel. If it exactly equals to the forward kernel then the transform is an 'orthogonal transform'. DCT is an orthogonal transform.

Unitary transforms can be conveniently expressed in vector-space form. Let \vec{f} and \vec{F} denote the vector forms of an image pixel array and its transform, then two-dimensional unitary transform in matrix form is,

$$\begin{aligned}
\vec{F} &= [A]\vec{f} \quad (\text{forward}) \\
\vec{f} &= [B]\vec{F} \quad (\text{backward}) \quad (2.24)
\end{aligned}$$

Then it is obvious that $[B] = [A]^{-1}$ and for unitary transforms, $[A]^{*T} = [A]^{-1}$. For real, unitary transforms, $[A]^T = [A]^{-1}$.

Testing these conditions on the transform matrices of DCTs given in Equations (2.1),(2.2),(2.3),(2.4), which are

1. DCT-I :

$$[C_{N+1}^I]_{mn} = \left(\frac{2}{N}\right)^{1/2} \left[k_m k_n \cos\left(\frac{mn\pi}{N}\right) \right], \quad m, n = 0, 1, \dots, N; \quad (2.25)$$

2. DCT-II :

$$[C_N^{II}]_{mn} = \left(\frac{2}{N}\right)^{1/2} \left[k_m \cos\left(\frac{m(n+1/2)\pi}{N}\right) \right], \quad m, n = 0, 1, \dots, N-1; \quad (2.26)$$

3. DCT-III :

$$[C_N^{III}]_{mn} = \left(\frac{2}{N}\right)^{1/2} \left[k_n \cos\left(\frac{(m+1/2)n\pi}{N}\right) \right], \quad m, n = 0, 1, \dots, N-1; \quad (2.27)$$

4. DCT-IV :

$$[C_N^{IV}]_{mn} = \left(\frac{2}{N}\right)^{1/2} \left[\cos\left(\frac{(m+1/2)(n+1/2)\pi}{N}\right) \right], \quad m, n = 0, 1, \dots, N-1; \quad (2.28)$$

where,

$$k_p = \begin{cases} \frac{1}{\sqrt{2}} & \text{when } p = 0 \text{ or } N \\ 1 & \text{when } p \neq 0 \text{ and } N. \end{cases}$$

one can see that, they satisfy the below properties [8],

1. For DCT-I $[C_{N+1}^I]^{-1} = [C_{N+1}^I]^T = [C_{N+1}^I]$.
2. For DCT-II $[C_N^{II}]^{-1} = [C_N^{II}]^T = [C_N^{III}]$.
3. For DCT-III $[C_N^{III}]^{-1} = [C_N^{III}]^T = [C_N^{II}]$.
4. For DCT-IV $[C_N^{IV}]^{-1} = [C_N^{IV}]^T = [C_N^{IV}]$.

A unitary transformation is said to be ‘separable’ if its kernel can be written in the form,

$$A(n_1, n_2; m_1, m_2) = A_C(n_1, m_1)A_R(n_2, m_2). \quad (2.29)$$

where the kernel subscripts indicate row and column one-dimensional transform operations. A separable two-dimensional unitary transform can be computed in two steps. First, a one-dimensional transform is taken along each column of the image and next, a second one-dimensional transform is taken along each row of the result. An $(N \times N)$ two-dimensional DCT can also be reduced to two N -point one-dimensional DCTs. Let $[g]$ be an $(N \times N)$ matrix (an image block if g_{mn} is interpreted as the grey level or the intensity of the pixel at the location (m, n) of the image). $[G]$, the two-dimensional orthogonal transform of the matrix $[g]$, and its inverse can be defined as,

$$\begin{aligned} [G] &= [A_N][g][A_N]^T \\ [g] &= [A_N]^T[G][A_N] \quad . \end{aligned} \quad (2.30)$$

Notice that, the notation $[A_N]$, is adopted from the above Equations 1.25-28 to denote an $N \times N$ transform matrix. If $[G]$ is the two-dimensional DCT-II of $[g]$, then the uv^{th} element of $[G]$ is given by,

$$[G]_{uv} = \frac{2c(u)c(v)}{N} \sum_{m=0}^N \sum_{n=0}^N [g]_{mn} \cos \left[\frac{(2m+1)u\pi}{2N} \right] \cos \left[\frac{(2n+1)v\pi}{2N} \right],$$

where $u, v = 0, 1, \dots, N-1$, and

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0 \\ 1 & \text{otherwise.} \end{cases} \quad (2.31)$$

Similarly, the mn^{th} element of $[g]$ can be reconstructed by the two-dimensional IDCT-II of $[G]$ as,

$$[g]_{mn} = \frac{2}{N} \sum_{u=0}^N \sum_{v=0}^N c(u)c(v)[G]_{uv} \cos \left[\frac{(2m+1)u\pi}{2N} \right] \cos \left[\frac{(2n+1)v\pi}{2N} \right],$$

where $m, n = 0, 1, \dots, N-1$. (2.32)

Now the separability property of the two-dimensional DCT can be illustrated by writing Equation (2.31) as follows,

$$[G]_{uv} = \sqrt{\frac{2}{N}} c(u) \sum_{m=0}^N \left\{ \sqrt{\frac{2}{N}} c(v) \sum_{n=0}^N [g]_{mn} \cos \left[\frac{(2n+1)v\pi}{2N} \right] \right\} \times \cos \left[\frac{(2m+1)u\pi}{2N} \right],$$

where $u, v = 0, 1, \dots, N-1$. (2.33)

The inner summation is an N -point one-dimensional DCT-II of the rows of $[g]$, whereas the outer summation represents the one-dimensional DCT-II of the columns of the “semi-transformed” matrix. A fine implication of this property is that, two-dimensional ($M \times N$) DCTs can be implemented via M N -point one dimensional DCTs. Note that the order, in which the row transform and the column transform is done, is theoretically immaterial.

At this point perhaps it is worth mentioning some spatial properties of an $N \times N$ DCT coefficients block. The upper-left coefficient represents the DC or average intensity value of the transformed block. Progression from left to right represents an increasing number of vertical edges while, similarly, top to bottom represents an increasing number of horizontal edges. So bottom-right coefficient yields maximum mix of the vertical and horizontal edges (a chessboard pattern). Hence, discarding high-frequency coefficients in the DCT domain (which is tantamount to low-pass filtering) implies deletion of corresponding basis images from the original image, as DCT is a decomposition process. This is called ‘zonal sampling’ and experimental work on DCTs revealed that coding only 25% of the transform coefficients is enough for the human observers to recognize the inverse transformed image almost fully ([8]).

2.5 Superiority of DCT

In all transform coding systems the central point is, naturally, the decision of the transform that will be applied on the one-, two-, or three-dimensional input data array. While comparing a variety of orthogonal transforms for a certain

input data, parameters like ‘energy packing efficiency’, ‘decorrelation efficiency’, ‘variance distribution’, ‘maximum reducible bits’, implementation (or computational) complexity can be used. Another fact is that, the characteristics of the input source also affects the above parameters, so the source must be carefully modeled and all comparisons must be carried out using the same model of source.

To determine the optimum transformation for a given source in MSE sense the following discussion is valid (see chapter 3 of [8] for details). Consider a an N -dimensional, zero-mean, random vector \vec{X} . If $\{\vec{\phi}_i\}$ is a set of linearly independent basis vectors spanning the N -dimensional space, then \vec{X} can be expanded in terms of $\vec{\phi}_i$'s as,

$$\vec{X} = \sum_{i=0}^{N-1} a_i \vec{\phi}_i \quad (2.34)$$

Then \vec{X} can be accurately represented by N numbers, a_i . But if we were to represent \vec{X} with D numbers ($D < N$), so that bandwidth reduction is possible, we should choose $\{\vec{\phi}_i\}$ such that $N - D$ number of a_i 's are only insignificantly different than zero. Karhunen and Loève in their historical works [15], [16] showed that $\{\vec{\phi}_i\}$ to minimize D can be obtained by solving the diagonalization problem,

$$([A] - \mu_i [I_N]) \vec{\phi}_i = 0 \quad i = 0, 1, \dots, N - 1, \quad (2.35)$$

where $[A] = E[\vec{X} \vec{X}^T]$ is the auto-covariance matrix of the source, μ_i is a Lagrange multiplier and $[I_N]$ is the $N \times N$ identity matrix.

Now, the set of basis vectors $\{\vec{\phi}_i\}$ form the bases for Karhunen-Loève expansion and the matrix $[\Phi] = [\vec{\phi}_0, \vec{\phi}_1, \dots, \vec{\phi}_{N-1}]$ is the Karhunen-Loève Transform (KLT) matrix. The KLT is said to be optimal, because it has the following properties:

1. It completely decorrelates the signal in the transform domain.
2. It minimizes the MSE in bandwidth reduction.
3. It contains the most amount of energy (variance), in the fewest number of transform coefficients.
4. It minimizes the total representation entropy of the sequence.

However, it is computationally unattractive for the following reasons :

1. The basis functions are dependent on the auto-covariance matrix of the source (see Equation (2.35)) and therefore can not be predetermined.
2. Given an auto-covariance matrix, the solution of (2.35) is quite involved. There are only a few cases in which analytical solutions are available.
3. Unlike commonly used transforms, there is no fast KLT in general.

One such case for which the analytical solution to the diagonalization problem of (2.35) is available, is the stationary 1st order Markov signals (also referred as Markov-1). A formal definition of the process can be found, for example in Papoulis [17]. For the present, an m^{th} -order Markov signal can be defined as a signal for which the probability of a given sample having a particular value depends upon a finite number ‘ m ’ of the preceding symbols. That is the source has memory. As far as a 1st order Markov source is concerned it depends only on the previous sample. The degree of dependence is represented with a correlation coefficient, $0 \leq \rho \leq 1$. In digital images, the analogous situation is where the probability of a given pixel having a particular intensity is dependent on the surrounding pixels. In fact, this model is quite efficient for the representation of correlation-displacement relationship of images which are smooth, i.e. , which contain relatively small varying details. Still, usage of this model for the ‘active images’ is often criticized ([18], [19]), saying that it does not provide an accurate, general representation. Indeed, with only three statistical parameters (mean, variance, and one-step correlation coefficient) it does not seem logical to represent such active images, but for sufficiently slowly varying intensity images, it is the most commonly used model for describing the statistics of each line of the image,[20]:

$$\begin{aligned}
 u_k &= \rho u_{k-1} + \epsilon_k & k &= 0, 1, \dots, N-1 \\
 \text{where,} & & E\{\epsilon_k\} &= 0, & E\{\epsilon_k \epsilon_l\} &= (1 - \rho^2) \sigma_u^2 \delta_{k,l} \\
 & & \text{and} & & E\{u^2\} &\triangleq \sigma_u^2 .
 \end{aligned} \tag{2.36}$$

Note that one-step correlation ρ is generally chosen to be $0.9 \leq \rho \leq 0.95$ for image sources.

Auto-covariance matrices of such stationary Markov-1 sources of unity variance, i.e. $\sigma_u^2 = 1$, have the form,

$$[A]_{i,k} = \rho^{|i-k|} \quad i, k = 0, 1, \dots, N-1 . \tag{2.37}$$

It is interesting to observe that, with

$$\alpha \triangleq \rho/(1 + \rho^2) \quad (2.38)$$

and

$$\beta^2 \triangleq (1 - \rho^2)/(1 + \rho^2) \quad (2.39)$$

the following property can be shown [11] for the sinusoidal transforms kernel matrix $[J]$ of Equation (2.18) :

$$[J(\rho, \rho, 0, 0)] = \beta^2[A]^{-1} \quad (2.40)$$

Note that, since the eigenvectors of a matrix is invariant under all commuting transformations, the eigenvectors of $[A]^{-1}$ (and hence of $[J(\rho, \rho, 0, 0)]$) and $[A]$ are identical. Since $[J(\rho, \rho, 0, 0)]$ is also a covariance matrix, the KLT of its underlying random process is the same as the KLT of stationary Markov-1 process. For this case, Ray and Driver, [21], have provided the solution of (2.35).

The attractiveness of DCT stems from its asymptotical equivalence to the optimal KLT. It has been shown that (see [8], chapter 3), DCT-I is asymptotically equivalent to KLT (as $N \rightarrow \infty$ and $\rho = 1$, where N is the size of the transform matrix) and DCT-II is asymptotically equivalent to KLT (as $\rho \rightarrow 1$) for Markov-1 signals. Note that as ρ increases to 1 decorrelation power of DCT-I decreases even when N is very large. However DCT-II performs better for ρ near 1 without any dependency to N .

The proof of asymptotic equivalence above can either be carried out analytically, or just through simple observations on the matrix $[J]$ of Equation (2.18). For $\rho \rightarrow 1$ the matrix $[J(1, 1, 0, 0)]$ (see Equation (2.20)) reduces to KLT for Markov-1 signals, [11]. Infact all sinusoidal transforms are asymptotically equivalent to KLT in some aspect. But as $\rho \rightarrow 1$ (the case in typical images) only DCT-II is equivalent. To show this Jain in [11] utilizes the difference norm

$$\Delta = ||[J(k_1, k_2, k_3, k_4)] - [J(\rho, \rho, 0, 0)]|| \quad (2.41)$$

where $||[X]|| = \sum_{i,j} [X]_{i,j}^2$ is the weak norm (seminorm) of the matrix $[X]$, and tabulates Δ expressions for different sinusoidal transforms. From that it is evident that DCT-II is better than all even-sized sinusoidal transforms for $0.5 \leq \rho \leq 1$.

It must be mentioned, although the choice of DCT can be justified as being the best nearest suboptimum transform to optimum KLT for 1st order stationary

Markov Processes, typical image data is, aside from fitting to 1st order Markov, is not stationary. Still, DCT seems to be more fitting to the covariance matrices of such non-stationary processes [12].

Besides these theoretical near optimalities of DCT-II, Rao and Yip, in chapter 6 of [8], testify DCT-II as being the second best transform after KLT for Markov-1 signals with $\rho = 0.9$, using the following performance measures

- Rapidness in the decrease of variance distribution of coefficients.
- Energy packing efficiency, i.e. portion of energy contained in the first M of N transform coefficients.
- Fractional correlation left undone by a transform, a measure defined in [22]. KLT completely decorrelates a signal. Whereas auto-covariance matrix in KLT domain is diagonal, the covariances matrices in the domain of other discrete transforms have non-zero off-diagonal elements which are indicative of the correlation between the various sequences involved.
- Minimum information rate in bits per transform coefficient as yielded by rate-distortion function assuming coefficients have Gaussian probability distribution.

Similar comparisons had been carried out by Clarke (see [12] chapter 3) for Markov-1, $\rho = 0.91$, in measures of energy packing efficiency and decorrelation efficiency. He also compared various transforms in terms of energy contained in the DC and four lowest-order coefficients in the case of 8×8 transform for three test images. Results showed that DCT-II is the best substitute for KLT.

The complexity of KLT has made any attempt in deriving fast algorithms quite futile, while DCT-II and others that are asymptotically equivalent to it are amenable to fast algorithms, making real-time applications feasible. For the two-dimensional case of images fast implementations can be classified as follows

- By reducing 2D-DCT to a lexicographically ordered 1D-DCT, [23], using the separability property of 2D-DCT, (Equation (2.33)). Note that most VLSI implementations of 2D-DCT make use of this property! The method is only partially recursive, but allows simple index mapping. In this case, fast 1D-DCT algorithms must also be examined and many examples of them are reviewed in chapter 4 of [8].

- Block matrix decomposition of the transform matrices, thus aiming for a recursive structure [24]. The author of this thesis have implemented this algorithm on a PC-based computer as it is the only fully recursive algorithm with relatively less number of real multiplications and additions among others.
- Implementation via 2D-FFT (see Equation (2.17)), [25]. Then any FFT algorithm can easily be employed.
- Implementation via other discrete transforms such as Walsh-Hadamard Transform (WHT) [26].

When compared to DFT, DCT alleviates some of the problems which arise in the application of DFT to a data series. Naturally, DFT is applied to sampled data, and so the transform domain has a “repeat” spectra. Sampling rate should be such that aliasing does not occur. Conversely, the fact that transform coefficients are also sampled, i.e. calculated at a finite number of frequencies means that a line spectrum is generated instead of a continuous spectrum, for which the input must have been “periodic”. Therefore the DFT representation is not that of an isolated segment of the input, but is of that sample periodically repeated. Such a waveform contains severe discontinuities due to the level difference between the start and end of the repeated segment. These result in spurious spectral components. However, in the case of DCT the segment is made even symmetric before transforming (see Equation (2.14)). Then the start and end of the new, even symmetric segment is at the same level! This fact is, also, the reason why even sized DCTs are preferred.

In addition to all the appealing properties of DCT mentioned above, DCT is outstanding as being a ‘real transform’, thus allowing real arithmetic.

CHAPTER III

DISCRETE COSINE TRANSFORM DOMAIN SCALAR QUANTIZATION

In the previous chapter the choice of DCT as a good substitute for KLT for image data sources has been justified. If image data is to be scalar quantized in the DCT domain, the outline of a practical transform coding algorithm is as follows:

1. Divide the $N \times N$ image into small rectangular blocks of size $n \times n$.
2. Transform each block.
3. Estimate the variances of transform coefficients using sufficient number of transformed blocks.
4. Using the variances of coefficients employ a bit allocation algorithm to determine the optimal number of bits that each DCT coefficient should be quantized into.
5. Quantize the transform coefficients to the predetermined number of bits, using an optimal non-uniform quantizer tailored for the probability distribution of the coefficients. Such quantizers are designed for unity variance data source, so normalize the coefficients to have unity variance prior to quantization.
6. Entropy encode the quantizer output (upon demand) to reduce the bit-rate without causing distortion at the expense of dealing with the buffering problem. Buffering problem arises when a variable rate code is connected to a channel or decoder demanding fixed rate input.
7. Encode the quantizer or entropy coder output. Also encode the variances and bit allocation table for the reproduction of the coefficients. Encoding is done for error control.

8. Transmit or store the data which is in the form of codewords now.

The scheme outlined above is also called ‘2D Block Quantization’. Block quantization is the method of quantizing a block of independently distributed continuous random variables on an element-by-element basis (therefore should not be confused with VQ). Each element of the block is quantized by a scalar quantizer into a predetermined integer number of bits. It has been widely employed in transform coding and subband coding especially for images.

Transform coefficient quantization and bit allocation for the quantizers are two crucial steps in block quantization which worth discussing in detail. Bit allocation is the first problem to be solved prior to quantization. However, for a better understanding, the discussion about bit allocation problem is deferred until other subjects about quantizers are covered.

Apart from the essential steps of bit allocation and quantization, in the later sections the role of ‘human visual system’ in DCT domain transform coding will also be discussed. The chapter will be concluded with experimental work involving various DCT domain SQ schemes.

3.1 Scalar Quantizers for Transform Coefficients

Major part of error introduced in the compression process using transform coding is caused by the quantization of the DCT coefficients. The transform operation is normally carried out digitally, and therefore the input image data is already being (linearly) quantized by the frame grabber. That’s why, some authors refer to the coefficient quantization step as ‘requantization’ [12].

Quantization is the process of subdividing the range of a signal into nonoverlapping regions. An output level is then assigned to represent each region. A formal definition can be made as follows,

Let X be a real scalar random variable at the quantizer input, with variance σ_x^2 and pdf $p_x(\cdot)$. An L -level quantizer $Q(\cdot)$ maps the continuous variable X into a discrete variable Y such that

$$Q(X) = Y = y_k \text{ if } x_k \leq X < x_{k+1}$$

where

$$y_k \in \{y_1, y_2, \dots, y_{L+1}\} \quad x_k \in \{x_1, x_2, \dots, x_L\} \quad (3.1)$$

are the reconstruction levels and decision threshold values respectively. Then quantization error $Q = X - Y$ is also a random variable with pdf $p_q(\cdot)$ and variance

$$\sigma_q^2 = E[Q^2] = \int_{-\infty}^{\infty} [x - Q(x)]^2 p_x(x) dx. \quad (3.2)$$

If the region of integration is divided into L intervals, we obtain

$$\sigma_q^2 = \sum_{k=1}^L \int_{x_k}^{x_{k+1}} [x - y_k]^2 p_x(x) dx. \quad (3.3)$$

Quantizers can be categorized as uniform or nonuniform, symmetric or non-symmetric, midrise or midtread and memoryless or with memory. Our discussion will be limited to memoryless, midrise quantizers, those for which the mapping of a sample at a time is not affected by earlier or later input samples and zero is not one of the output levels. Symmetric quantizers have their reconstruction and decision threshold values even symmetric for both positive and negative ranges of the input. Then, positive (negative) levels completely describe the quantizer.

3.1.1 Uniform Quantizers

For uniform quantizers decision intervals are of same length Δ , and the reconstruction levels are the mid-points of the decision intervals,

$$x_1 = -\infty; \quad x_{L+1} = \infty; \quad x_{k+1} - x_k = \Delta; \quad k = 2, \dots, L-1. \quad (3.4)$$

The error introduced can be expressed in two parts. Errors which are due to the finite number of levels of the quantizer and are in the finite ranges of $(-\frac{\Delta}{2}, \frac{\Delta}{2}]$ are called ‘granular’ errors, while errors due to the bursts in the input samples, i.e. if $x > (x_L + \Delta)$ or $x < -(x_L + \Delta)$ are called ‘overload’ errors (see Figure 3.1). Total error variance is

$$\sigma_q^2 = \sigma_{q(\text{granular})}^2 + \sigma_{q(\text{overload})}^2. \quad (3.5)$$

If the source to be quantized is uniformly distributed and bounded with $x \in (x_{min}, x_{max})$, then there is no overload distortion, and simply $\Delta = \frac{x_{max} - x_{min}}{2^R}$, where R is the number of bits/sample. If the source has a nonuniform pdf, the optimum uniform quantizer is found to have stepsize Δ_{opt} which minimizes σ_q^2 of Equation (3.5) or Equation (3.3) with respect to an error measure such as mean

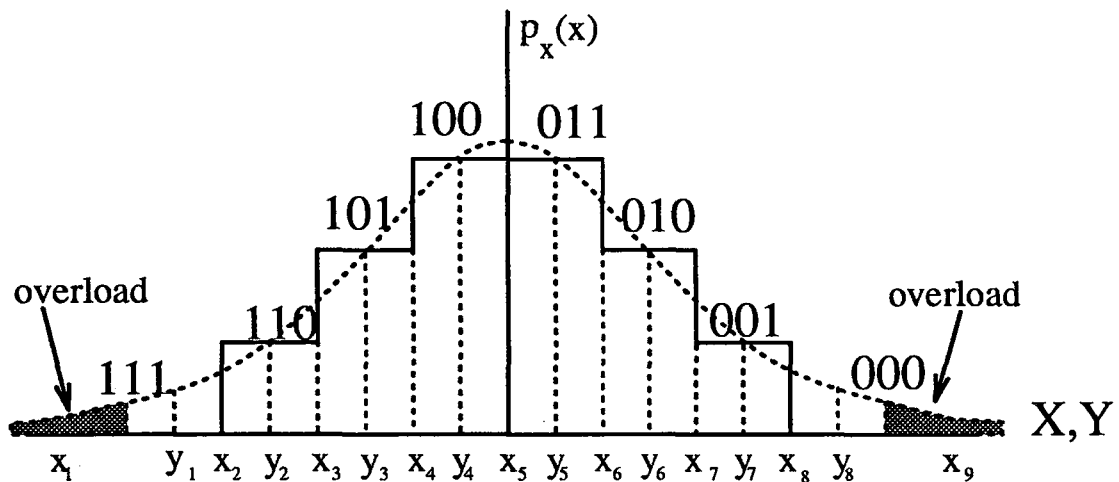


Figure 3.1. Illustration of a symmetric, uniform, 3-bit, midrise quantizer

square error (MSE). Such quantizers are called minimum MSE (MMSE) uniform quantizers or uniform Max quantizers. Then, equation to be solved is

$$\left. \frac{d\sigma_q^2}{d\Delta} \right|_{\Delta_{opt}} = 0. \quad (3.6)$$

where σ_q^2 of Equation (3.3) can be used where

$$\begin{aligned} x_k &= \left(k - \frac{L+2}{2} \right) \Delta \quad ; \quad k = 2, 3, \dots, L \\ y_k &= \left(k - \frac{L+1}{2} \right) \Delta \quad ; \quad k = 1, 2, \dots, L \end{aligned} \quad (3.7)$$

for the case of midrise quantizers. Explicit solutions for this minimization problem is not available for $L > 3$, but numerical ones have been proposed by [39],[36].

Equidistant decision and representation levels are simple conceptually, as well as for implementation. However, a random variable can be quantized with smaller error variance if nonuniform spacing between the levels is employed. First workers to show this were Panter and Dite (1951), [40]. Later, an iterative solution for the determination of optimum decision and reconstruction levels, when pdf of the input and MMSE as the optimization criterion are given, is proposed by Lloyd and Max, [41],[39]. Such a MMSE quantizer is named as Lloyd-Max quantizer or nonuniform Max quantizer. Lloyd-Max quantizers are uniform only for uniform pdf (in which case Lloyd-Max quantizer equations become linear, such a quantizer is also called linear quantizer) and for $L \leq 2$. In the following section Lloyd-Max quantizers are discussed in detail.

3.1.2 Lloyd-Max Quantizers

Given the distribution of the source, a quantizer can be optimized to yield MMSE if the quantization error variance, σ_q^2 of Equation (3.3), is minimized. Therefore, necessary conditions for a minimum are

$$\frac{\partial \sigma_q^2}{\partial x_k} = 0 \quad ; \quad k = 2, 3, \dots, L \quad ; \quad \frac{\partial \sigma_q^2}{\partial y_k} = 0 \quad ; \quad k = 1, 2, \dots, L$$

Max showed that, [39], these differentiations leads to

$$x_{1,opt} = -\infty \quad ; \quad x_{L+1,opt} = \infty,$$

$$x_{k,opt} = \frac{y_{k,opt} + y_{k+1,opt}}{2} \quad ; \quad k = 2, 3, \dots, L. \quad (3.8)$$

$$y_{k,opt} = \frac{\int_{x_{k,opt}}^{x_{k+1,opt}} x p_x(x) dx}{\int_{x_{k,opt}}^{x_{k+1,opt}} p_x(x) dx} \quad ; \quad k = 2, 3, \dots, L. \quad (3.9)$$

These results state that the optimum decision levels lie halfway between the optimum reconstruction levels, which, in turn, lie at the center of mass of the probability density in the interval between the decision levels. Then iterative methods can be used to calculate optimal levels (see [42], [43], [44]).

Note that, the conditions above, although necessary, are not sufficient for designing optimum quantizers as they are also satisfied for locally minimum solutions. Fleisher, [45], and more recently Trushkin, [46], have shown that for the MSE criterion, a sufficient condition for having a unique solution of Lloyd-Max equations is log-concavity of the probability density function of the source :

$$\frac{\partial^2 \log p_x(x)}{\partial x^2} < 0. \quad (3.10)$$

It is also proven by Fleisher, [45], that if the log-concavity property holds for a symmetric distribution, then associated unique quantizer is symmetric too. But as pointed out by Sharma, [47], the optimal quantizer need not be symmetrical for all symmetrical distributions. Due to above facts, for various distributions the following results apply:

- The Gaussian (or normal) density function, $f(x) = Ae^{-\alpha x^2}$, where generally, $A = \frac{1}{\sigma\sqrt{2\pi}}$ and $\alpha = \frac{1}{2\sigma^2}$, is strictly log-concave, and hence, given the number of levels, it has a unique nonuniform symmetrical quantizer associated with it.

- The Laplacian (or two-sided exponential) density function, $f(x) = \frac{\alpha}{2}e^{-\alpha|x|}$, where $\alpha = \frac{\sqrt{2}}{\sigma}$, is only semi-normal, i.e. the inequality in Equation (3.10) should be replaced by equality. Still, Trushkin, [46] has shown that it does have a unique solution of Lloyd-Max equations. There is also a special set of equations presented in [38], requiring much less number of iterations for optimal quantizers with even number of levels tailored for Laplacian pdf. Note that the nonuniform quantizer for Laplacian pdf is also symmetric.
- The gamma density function, $f(x) = Ax^b e^{-cx}$, does not satisfy the log-concavity property. Kabal, [42], has shown that the optimal uniform and nonuniform quantizers of even number of levels for gamma pdf are not symmetrical, and presented the optimal levels.
- For Cauchy density function, $f(x) = \frac{\alpha/\pi}{\alpha^2+x^2}$, there is no evidence in the literature which examines the log-concavity. If the test is applied it can be seen that,

$$\frac{\partial^2 \log p_x(x)}{\partial x^2} = \frac{2(x^2 - a^2)}{(x^2 + a^2)^2} \quad (3.11)$$

Then inequality does not hold for $|x| \geq |a|$, hence existence of a unique MSE optimized quantizer is definitely suspicious.

- The Rayleigh density function, $f(x) = \frac{x}{\alpha^2} e^{-x^2/2\alpha^2} U(x)$, where $U(x)$ is the unit step function (1 for $x \geq 0$), is not symmetrical but log-concave as,

$$\frac{\partial^2 \log p_x(x)}{\partial x^2} = -\left(\frac{1}{x^2} + \frac{\log e}{\alpha^2}\right) < 0 \quad (3.12)$$

So there exists a unique quantizer (not symmetric) associated with it, but in the literature there is no trace of such a quantizer.

Some interesting properties of Lloyd-Max Quantizers are as follows (for the proofs see [4]):

- The quantizer output is an unbiased estimate of the input, i.e. , $E\{Y\} = E\{X\}$.
- The quantization error is orthogonal to the quantizer output, i.e. , $E\{(X - Y)Y\} = 0$.
- It is sufficient to design the quantizer for zero mean, unity variance distributions.

It is worth noting that theoretical optimality of the Lloyd-Max quantizer has also been proved experimentally by Jain, [20] for Gaussian and Laplacian densities. If the quantizer output is to be coded at a fixed number of levels, for a given rate Lloyd-Max quantizer has the closest SNR value to the hypothetical Shannon Quantizer (see Section 3.3.1), outperforming optimal uniform quantizers about 2 dB for Gaussian and 4.3 dB for Laplacian sources.

3.2 Distribution of DCT Coefficients for Images

It has been shown in the previous section that probability distribution of the DCT coefficients is an essential information for designing optimal quantizers tailored for them. However, there is a great confusion in the literature about the actual probability density function (pdf) of the coefficients. It is worth mentioning at this point that, the disorderly usage of terms probability distribution and density should not confuse the reader, since all the functions of concern are exponentials and hence, their derivatives or integrals are scaled versions of each other.

The first prediction about the distributions of DCT coefficients was made by Chen and Smith, [27], in 1977. Depending on the fact that each cosine transform domain sample is formed from cosine weighted sum of all the pixels in the original image, based on the central limit theorem (CLT), [17], they approximately modeled non-DC and DC coefficients by Gaussian densities. In 1978, Pratt, [13], conjectured that the DC coefficient should have a Rayleigh distribution since it was the sum of positive values, and that, based on CLT, non-DC coefficients should be Gaussian. In 1980, Netravali and Limb, [28], agreed with Pratt and also stated that the histogram of DC coefficients were roughly bell-shaped. Modestino et al., [29], also accepted the same assumption in 1981. Then, Murakami et al., [30], were the first to deny the general reliance on CLT. In 1982, they assumed Gaussian distribution for DC, and Laplacian for non-DC coefficients of Hadamard Transform. In the same year, Ngan, [31], pointed out that CLT was not applicable to short-term statistics considered in image coding, and examining the histograms of DCT coefficients of a head-and-shoulders image, he proposed the usage of Gaussian function for DC, and Laplacian function for non-DC DCT coefficients. In 1983, Reininger and Gibson, [32], realized the confusion

and applied Kolmogorov-Smirnov (KS) test to see the goodness of fit of Gaussian, Laplacian, Rayleigh, and Gamma distributions to DCT coefficients of five test images for block sizes of 8, 16, and 32. KS test statistic is a distance measure between the sample distribution $F_X(x)$ and the tested distribution $F_T(x)$,

$$D \triangleq \max_{i=1,2,\dots,N} |F_X(x_i) - F_T(x_i)|. \quad (3.13)$$

where N samples of data is used for testing. The distribution that yields the smallest KS statistic was the best fit for the data. They concluded that in all cases Gaussian distribution was the best fit for the DC coefficient, and that Laplacian distribution yielded the smallest KS statistics for non-DC coefficients in almost all images, except for a highly detailed image, for which Gaussian distribution was the best fit at all block sizes. This result was also excepted by Chen and Pratt, [33], in 1984, and they built their scene adaptive coder based on it. In 1985, Modestino et al. [34], once more proved empirically that non-DC coefficients depart very much from Gaussian statistics by forming so called quantile-quantile (Q-Q) plots, and percentile-percentile (P-P) plots between DCT coefficient distributions of ten test images and Gaussian distribution. They also proposed an adaptive quantizer to match the input statistics, but it was not fast enough to be used in real-time applications. This was almost the final work on the distribution of DCT coefficients. Gaussian assumption for DC and Laplacian assumption for non-DC coefficients became widely accepted. Only in 1986, Eggerton and Smith, [35], claimed that Cauchy distribution fits better to non-DC coefficients, but due to the facts stated in the Section 3.1.2, building quantizers for Cauchy distribution was not practical and did not draw much interest.

For implementation of nonuniform quantizers tailored for Laplacian pdf, the first attempt was made by Paez and Glisson, [36], but Adams and Giesler, [37], points out that some of their innermost decision and reconstruction values were in error by as much as 12%. Reasons for error were approximations and single precision arithmetic. Computer technology was not improved enough to calculate the values accurately, at that time. Although Noll and Zelinski, [38], reports that 8% error in threshold values would decrease the SNR 0.05 dB only, for the sake of perfection we have gone for the most accurate results. Here we tabulate the results presented by Ngan, [31]. They are accurate upto five decimal digits and confirms with the results of Adams and Giesler. The threshold

Table 3.1. Decision and reconstruction levels of optimum 7 bits Laplacian quantizer

x_i	y_i	x_i	y_i	x_i	y_i
0.00000	0.01639	0.83357	0.85792	2.22938	2.27641
0.03304	0.04969	0.88282	0.90772	2.32563	2.37485
0.06660	0.08351	0.93323	0.95873	2.42646	2.47808
0.10070	0.11788	0.98486	1.01099	2.53234	2.58659
0.13535	0.15282	1.03778	1.06458	2.64376	2.70094
0.17059	0.18836	1.09208	1.11957	2.76138	2.82182
0.20643	0.22451	1.14780	1.17603	2.88592	2.95002
0.24290	0.26128	1.20502	1.23402	3.01824	3.08647
0.28000	0.29872	1.26384	1.29366	3.15938	3.23230
0.31778	0.33684	1.32434	1.35502	3.31059	3.38888
0.35625	0.37567	1.38661	1.41821	3.47341	3.55794
0.39545	0.41524	1.45077	1.48334	3.64979	3.74164
0.43540	0.45556	1.51694	1.55054	3.84220	3.94276
0.47611	0.49666	1.58524	1.61995	4.05388	4.16499
0.51762	0.53858	1.65582	1.69170	4.28913	4.41326
0.55996	0.58133	1.72884	1.76598	4.55388	4.69449
0.60315	0.62498	1.80447	1.84296	4.85665	5.01881
0.64726	0.66954	1.88290	1.92284	5.21034	5.40187
0.69231	0.71508	1.96435	2.00587	5.63580	5.86973
0.73836	0.76164	2.04907	2.09227	6.17040	6.47107
0.78543	0.80923	2.13731	2.18235	6.89260	7.31412
				8.02641	8.73870

values given for 256 level Gaussian are calculated using Max algorithm. It is very important to note that, these tables are for zero mean, unity variance sources. The mean of DC coefficient must be subtracted from itself and all coefficients must be variance normalized prior to quantization. Calculation of variances of coefficients is described in Section 3.3.3.

The necessary quantization tables for Lloyd-Max quantization of DCT coefficients, are given in Tables 3.1, 3.2, and 3.3.

Table 3.2. Decision and reconstruction levels of optimum Laplacian quantizers

Bits	x_i	y_i	Bits	x_i	y_i
1	0.00000	0.70711		0.00000	0.03258
	0.00000	0.41976		0.06619	0.09979
2	1.12686	1.83397		0.13449	0.16919
	0.00000	0.23340		0.20507	0.24094
	0.53318	0.83296		0.27806	0.31519
3	1.25274	1.67251		0.35366	0.39213
	2.37965	3.08680		0.43206	0.47199
	0.00000	0.12399		0.51348	0.55498
4	0.26442	0.40484	6	0.59816	0.64135
	0.56675	0.72866		0.68637	0.73139
	0.91984	1.11102		0.77841	0.82542
	1.34443	1.57784		0.87462	0.92381
	1.87764	2.17743		0.97541	1.02701
	2.59722	3.01701		1.08125	0.13550
	3.72421	4.43142		1.19267	1.24984
	0.00000	0.06404		1.31026	1.37069
5	0.13220	0.20035	1.43478	1.49887	
	0.27320	0.34605	1.56709	1.63530	
	0.42428	0.50251	1.70820	1.78110	
	0.58697	0.67144	1.85938	1.93766	
	0.76322	0.85500	2.02216	2.10667	
	0.95548	1.05596	2.19851	2.29034	
	1.16697	1.27798	2.39090	2.49146	
	1.40199	1.52601	2.60256	2.71366	
	1.66648	1.80695	2.83777	2.96188	
	1.96891	2.13088	3.10245	3.24303	
	2.32214	2.51340	3.75870	3.95015	
	2.74694	2.98048	4.18397	4.41780	
	3.28050	3.58051	4.71828	5.01876	
4.00074	4.42098	5.43991	5.86106		
5.12949	5.83800	6.57225	7.28344		

Table 3.3. Decision, reconstruction levels of optimum 8 bits Gaussian quantizer

x_i	y_i	x_i	y_i	x_i	y_i	x_i	y_i
0.0000	0.0084	0.5497	0.5586	1.1630	1.1735	1.9802	1.9964
0.0169	0.0253	0.5675	0.5764	1.1842	1.1949	2.0130	2.0296
0.0338	0.0422	0.5853	0.5943	1.2056	1.2164	2.0466	2.0636
0.0507	0.0591	0.6033	0.6122	1.2273	1.2381	2.0810	2.0983
0.0676	0.0760	0.6212	0.6302	1.2491	1.2600	2.1161	2.1340
0.0845	0.0930	0.6393	0.6483	1.2711	1.2821	2.1522	2.1705
0.1014	0.1099	0.6574	0.6665	1.2933	1.3045	2.1893	2.2081
0.1183	0.1268	0.6756	0.6847	1.3157	1.3270	2.2274	2.2467
0.1353	0.1438	0.6939	0.7030	1.3384	1.3498	2.2666	2.2864
0.1522	0.1607	0.7122	0.7214	1.3613	1.3728	2.3069	2.3275
0.1692	0.1777	0.7306	0.7399	1.3844	1.3960	2.3486	2.3698
0.1862	0.1947	0.7491	0.7584	1.4078	1.4195	2.3917	2.4136
0.2032	0.2117	0.7677	0.7771	1.4314	1.4433	2.4363	2.4590
0.2202	0.2287	0.7864	0.7958	1.4553	1.4673	2.4826	2.5062
0.2373	0.2458	0.8052	0.8146	1.4795	1.4917	2.5308	2.5553
0.2543	0.2629	0.8241	0.8335	1.5040	1.5163	2.5810	2.6066
0.2714	0.2800	0.8430	0.8525	1.5288	1.5412	2.6334	2.6603
0.2885	0.2971	0.8621	0.8717	1.5539	1.5665	2.6884	2.7166
0.3057	0.3142	0.8813	0.8909	1.5793	1.5921	2.7463	2.7760
0.3228	0.3314	0.9006	0.9102	1.6051	1.6180	2.8074	2.8388
0.3400	0.3487	0.9199	0.9297	1.6312	1.6444	2.8722	2.9056
0.3573	0.3659	0.9395	0.9492	1.6577	1.6711	2.9413	2.9770
0.3746	0.3832	0.9591	0.9689	1.6846	1.6982	3.0154	3.0538
0.3919	0.4005	0.9788	0.9888	1.7119	1.7257	3.0955	3.1372
0.4092	0.4179	0.9987	1.0087	1.7397	1.7537	3.1829	3.2286
0.4266	0.4353	1.0187	1.0288	1.7679	1.7821	3.2793	3.3299
0.4441	0.4528	1.0389	1.0490	1.7966	1.8110	3.3870	3.4441
0.4615	0.4703	1.0592	1.0694	1.8258	1.8405	3.5099	3.5757
0.4791	0.4878	1.0796	1.0899	1.8555	1.8705	3.6537	3.7317
0.4966	0.5054	1.1002	1.1106	1.8857	1.9010	3.8287	3.9257
0.5143	0.5231	1.1210	1.1314	1.9166	1.9322	4.0562	4.1867
0.5320	0.5408	1.1419	1.1524	1.9481	1.9640	4.3951	4.6036

3.3 Optimal Bit Allocation

Below is a block diagram of the 2D block quantization scheme which has been outlined at the beginning of this chapter.

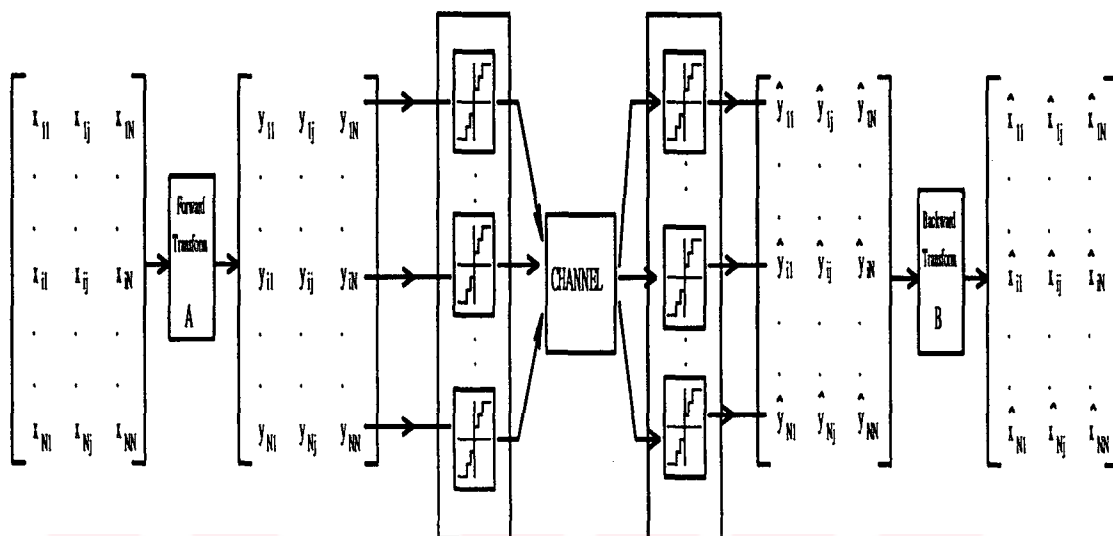


Figure 3.2. One-dimensional transform block quantization with scalar quantizers

As it can also be guessed from Figure 3.2 optimal bit allocation is the problem of distributing a given number B of bits among M quantizers, such that the total average distortion of all M quantizers would be minimized. Assuming that all coefficients have been quantized (even to 0 bits), M is generally equal to the number of coefficients (N^2 in case of $N \times N$ block quantization).

Consider a two-dimensional $N \times N$ block of image data $[X]$ with elements (pixels) x_{ij} , $i, j = 1, 2, \dots, N$. If $[X]$ is transformed using a real, unitary transform (see Section 2.4) into another $N \times N$ block $[Y]$ with elements (transform coefficients) y_{ij} , $i, j = 1, 2, \dots, N$, then $[Y] = [A][X]$, and $[X] = [A]^{-1}[Y] = [A]^T[Y]$. Assigning a different scalar quantizer to each transform coefficient, one obtains $\hat{y}_{ij} = q_{ij}(y_{ij})$ or $[\hat{Y}] = Q([Y])$. Inverse transformation gives the reconstruction block, $[\hat{X}] = [A]^T[\hat{Y}]$. The aim of bit allocation is to minimize the overall mean-square distortion of the system, or in other words, the reconstruction error variance :

$$\begin{aligned} D &= \frac{1}{N \times N} \mathbb{E} \left\{ \sum_{i,j=1}^N (x_{ij} - \hat{x}_{ij})^2 \right\} \\ &= \frac{1}{N \times N} \sum_{i,j=1}^N \mathbb{E} \{ (x_{ij} - \hat{x}_{ij})^2 \} \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{N \times N} \sum_{i,j=1}^N \sigma_{e_{ij}}^2 \\
&= \sigma_e^2
\end{aligned} \tag{3.14}$$

where $e_{ij} = x_{ij} - \hat{x}_{ij}$ is the reconstruction error for ij^{th} input sample.

Since unitary transforms are variance preserving, the only source of distortion in the system are (assuming error-free transmission) the quantizers. Therefore, the reconstruction error variance for the ij^{th} element equals to the quantization error variance of the ij^{th} quantizer. Defining the reconstruction error matrix $[E] = [X] - [\hat{X}]$, the quantization error for ij^{th} element $q_{ij} = y_{ij} - \hat{y}_{ij}$, and the quantization error matrix $[Q] = [Y] - [\hat{Y}]$, the proof is as follows :

$$\begin{aligned}
\sigma_e^2 &= \text{E}\{[E]^T[E]\} \\
&= \text{E}\{[[X] - [\hat{X}]]^T[[X] - [\hat{X}]]\} \\
&= \text{E}\{[[A]^T[[Y] - [\hat{Y}]]]^T[[A]^T[[Y] - [\hat{Y}]]]\} \\
&= \text{E}\{[[Y] - [\hat{Y}]]^T[A][A]^T[[Y] - [\hat{Y}]]\} \\
&= \text{E}\{[[Y] - [\hat{Y}]]^T[[Y] - [\hat{Y}]]\} \\
&= \text{E}\{[Q]^T[Q]\} \\
&= \sigma_q^2
\end{aligned} \tag{3.15}$$

Assuming b_{ij} number of bits are allocated for the ij^{th} quantizer, let's define a quantizer distortion function $f(b_{ij})$, such that

$$\begin{aligned}
\sigma_q^2 &= \frac{1}{N \times N} \sum_{i,j=1}^N \sigma_{q_{ij}}^2 \\
&= \frac{1}{N \times N} \sum_{i,j=1}^N \sigma_{y_{ij}}^2 f(b_{ij})
\end{aligned} \tag{3.16}$$

where $\sigma_{y_{ij}}^2$ is the variance of the ij^{th} transform coefficient. Interpretation of $f(b_{ij})$ is that , depending on the pdf of the quantizer input (in our case, transform coefficients), type of quantizer and number of levels of the quantizer, it gives the mean-square distortion that the quantizer will produce if the input has unity variance. In general it is not possible to obtain closed form expressions for $f(b_{ij})$. Seeking for approximate solutions,if the quantization is fine enough to assume that the source pdf is constant within a quantizer interval, then

$$f(b_{ij}) = k2^{-pb_{ij}} \tag{3.17}$$

where k and p take on constant values depending on the type of the quantizer and pdf of the source [12], [44]. For fine quantization i.e. $b_{ij} \geq 5$, $p \approx 2$ (for large b_{ij} , p can be approximated regardless of whether the quantizer is uniform or non-uniform). For instance, Max [39] determined

$$\begin{aligned} f(L) &= 1.32L^{-1.74} & L = 4 \\ f(L) &= 2.21L^{-1.96} & L = 36 \end{aligned} \quad (3.18)$$

where L is the number of quantizer levels. Later, Wood [48] proposed an approximation for the MSE of the optimum quantizer (see Section 3.1.2) which, for the Gaussian distribution, tends to $2.7L^{-2}$ as $L \rightarrow \infty$. Moreover, for large L , Jayant and Noll (see [44], sect. 12.4.1) set $p = 2$ and choose k as 1.0, 2.7, 4.5, and 5.7 for Max quantizers with, uniform, Gaussian, Laplacian and gamma distributions respectively at the input. For the uniform quantizer Clarke (see [12] sect. 4.6.1) suggests

$$\begin{aligned} f(L) &= L^{-1.6} & L \leq 4 \\ f(L) &= 1.66L^{-2} & L > 4 \end{aligned} \quad (3.19)$$

So, although the choice of unitary transform has no effect on the development of theory, the probability distribution and variance distribution of its coefficients have crucial importance. At this point the aim of bit allocation can again be stated as minimization of the average distortion

$$D = \frac{1}{N \times N} \sum_{i,j=1}^N \sigma_{y_{ij}}^2 f(b_{ij}) \quad (3.20)$$

subject to the constraint

$$\sum_{i,j=1}^N b_{ij} = B \quad (3.21)$$

Properties of the bit allocation scheme that will solve this problem should be as follows :

- It should be adaptable to systems which consist of different types of quantizers, having different expressions of quantizer distortion function $f(b_{ij})$.
- It should consider arbitrary coefficient pdf's, as long as the quantizer distortion function for that kind of source exists.

- It should output integer number of bits.
- It should have $b_{ij} \geq 0$ since practically it must be so, and $b_{ij} \leq b_{\max}$ due to perceptual reasons.
- It should be applicable for all scalar quantizers as long as they have convex and non-increasing distortion functions (non-increasing in the sense that as the number of quantizer levels increase the amount of distortion that the quantizer will introduce should increase). The reason for the requirement of convexity will be given in Section 3.3.1.

In the following sections, the methods for achieving minimum distortion with the total allocatable bits constraint will be described and their usefulness in terms of above properties will be discussed.

3.3.1 Rate-Distortion Theoretic Approach

This section will cover a common approach to bit allocation problem which has been in use for more than two decades and still seems appealing to some researchers. Historical evolution of the algorithm will be given and reasons why its optimality in theory does not ensure its optimality in practice will be discussed.

The first work dealing with bit assignment in block quantization was reported by Huang and Schultheiss [49], who considered blocks of correlated Gaussian random variables transformed by KLT (see Section 2.5). Since KLT coefficients of a block were Gaussian and uncorrelated, they were also independent. So they developed their bit allocation scheme assuming the block quantization of independent, Gaussian random variables and chose the quantizer function of $f(b_{ij}) = k2^{-2b_{ij}}$ (see 3.3). Then, Equation (3.16) becomes

$$D = \sigma_q^2 = \frac{1}{N \times N} \sum_{i,j=1}^N \sigma_{y_{ij}}^2 k2^{-2b_{ij}}. \quad (3.22)$$

If quantization of each coefficient were to introduce equal and constant amount of error,

$$d = \frac{\sigma_{q_{ij}}^2}{k} = \sigma_{y_{ij}}^2 2^{-2b_{ij}}. \quad (3.23)$$

Then,

$$b_{ij} = \frac{1}{2}(\log_2 \sigma_{y_{ij}}^2 - \log_2 d). \quad (3.24)$$

So using Equation (3.21) one obtains

$$\sum_{i,j=1}^N (\log_2 \sigma_{y_{ij}}^2 - \log_2 d) = 2B, \quad (3.25)$$

or equivalently

$$\log_2 d = -\frac{2B}{N \times N} + \frac{1}{N \times N} \sum_{i,j=1}^N \log_2 \sigma_{y_{ij}}^2. \quad (3.26)$$

Substituting Equation (3.26) into Equation (3.24) optimum bit assignments are obtained :

$$\begin{aligned} b_{ij} &= \frac{1}{2} \left[\log_2 \sigma_{y_{ij}}^2 - \frac{1}{N \times N} (-2B + \sum_{k,l=1}^N \log_2 \sigma_{y_{kl}}^2) \right] \\ &= \frac{B}{N \times N} + \frac{1}{2} \left(\log_2 \sigma_{y_{ij}}^2 - \frac{1}{N \times N} \sum_{k,l=1}^N \log_2 \sigma_{y_{kl}}^2 \right) \\ &= \frac{B}{N \times N} + \frac{1}{2} \log_2 \frac{\sigma_{y_{ij}}^2}{[\prod_{k,l=1}^N \sigma_{y_{kl}}^2]^{1/N \times N}}. \end{aligned} \quad (3.27)$$

Note that $\frac{B}{N \times N}$ is the average rate of the system and since Huang and Schultheiss used KLT, the variance $\sigma_{y_{ij}}^2$ are in fact the eigenvalues of the source covariance matrix (see Section 2.5).

Notice that Equation (3.24) (where Equation (3.27) stems from) has the form $b_{ij} = \frac{1}{2} \log_2(\sigma_{y_{ij}}^2/d)$ and so it is very similar to Shannon's famous rate-distortion function for coding zero-mean, σ^2 -variance, discrete-time, continuous amplitude, memoryless, Gaussian sources under mean-square error distortion criterion [3]. This similarity (noticed first by Davisson [50]) needs examining since it will reveal the underlying approximation in the Huang and Schultheiss bit allocation scheme.

Assume a discrete-time source of Gaussian distributed random variables, $N(0, \sigma^2)$, which are statistically independent (i.e. memoryless) and can take any value on the real-line (continuous amplitude). Distortionless transmission of such a source is impossible as the channel capacity must be infinite. Since a channel word should be assigned for each point on the real-line, there will be infinite number of channel words, implying that each channel word will have infinite dimension. So the channel capacity, C , determines the quality of the reproduction, because assuming the channel is not wide enough to transmit all the information, the amount it can handle will determine the amount of mutual information between the source and the reproduction. Defining a measure for distortion, the

rate-distortion function $R(D)$ is the minimum value of mutual information for a given distortion level D . Therefore, $R(D) < C$ insures the possibility of obtaining distortion as low as D . If the source is memoryless, $R(D)$ depends on source the pdf and the chosen reproduction fidelity criterion (also called distortion measure, cost function, loss function etc.). In the popular case of difference distortion measures where the measure $\rho(\cdot)$ is of a single argument such that $\rho(x, y) = \rho(x - y)$ is used, there exists an easy to calculate lower bound for $R(D)$ if $\int e^{s\rho(z)} dz$ is finite $\forall s < 0$ (see [51] section 4.3.1). This bound is called ‘Shannon lower bound’ after its inventor [3], and the associated (hypothetical) block quantizer is called ‘Shannon quantizer’ [20]. Shannon lower bound is different for a given pdf and difference distortion measure and the hypothetical Shannon quantizer needs an infinite size block of source samples to encode to achieve the bound. Note that if the source is correlated (i.e. with memory) $R(D)$ can not be calculated analytically, rather $R_N(D)$ for $N \rightarrow \infty$ is calculated where $R_N(D)$ is the N^{th} order rate-distortion function for coding blocks of size N of the source. In transform coding, since one deals with finite number of coefficients for, say, an interval of speech or a block of image, transform coefficients are a discrete-time source. For KLT transform coefficients are independent and continuous in amplitude (i.e. can achieve any value on the real-line). Given that such a source has $N(0, \sigma^2)$ as a pdf and fidelity criterion is the squared-error, it was first Shannon [3] (see [51], theorem 4.3.2 for a complete derivation) who showed that

$$\begin{aligned}
 R(D) &= \frac{1}{2} \max(0, \log_2 \frac{\sigma^2}{D}) \\
 &= \begin{cases} \frac{1}{2} \log_2 \frac{\sigma^2}{D}, & 0 \leq D \leq \sigma^2, \\ 0, & D \geq \sigma^2. \end{cases} \quad (3.28)
 \end{aligned}$$

Note that, if $D \geq \sigma^2$; i.e., the allowable distortion is greater than the signal variance (energy), then there is no need to code and transmit it at all [50]. Also note that for this case of Gaussian pdf and MSE difference measure, Shannon lower bound is also equal to $R(D)$. Conceptually, the performance of Equation (3.28) is achievable by encoding in such a way that the reconstruction error is Gaussian with variance D for each possible sample (coefficient) value and is sample-to-sample independent. Shannon Quantizer has such output since it uses infinite size sample blocks, which will cause infinite end-to-end delay! Another case achieving Gaussian error distribution is transmission over a Gaussian channel

with noiseless feedback (see [50] and references there, and [51] section 4.3.3). So there exists no practical way of achieving the bound. As a result, the underlying assumption of Huang and Schultheiss was that they assumed that all quantizers in their block quantizer were Shannon quantizer i.e. their quantizer distortion function $f(n) = k2^{-2n}$ belongs to a Shannon Quantizer. The only reason for choosing a Shannon quantizer may be that Shannon lower bound coincides with $R(D)$'s of many practical quantizers at a low distortion values i.e. if the number of levels of the quantizers are large (see the start of Section 3.3 and [51] section 4.3.4).

The only reasonable property of Huang and Schultheiss scheme may be their Gaussian assumption of source samples and use of MSE, which at first sight seems contradictory since the following two objections may immediately be raised :

1. Sources, in particular video sources, are not Gaussian.
2. MSE is not appropriate to account for the perceived distortion by human observers.

These objections are true, but it is also true that the value of Gaussian $R(D)$ for MSE criterion is an upper bound for the $R(D)$'s of all other non-Gaussian pdf sources , since it has the maximum differential entropy (see [52] page 1563, and [51] theorem 4.6.3). Although, a system would achieve optimum performance (can encode into fewer bits) if it has been tailored for the second order statistics (i.e. covariance matrix, or spectral density if source is stationary, or exact pdf with known variance), such information may not be available or variable, like video data source for instance. Such was the case for the the early days of Huang and Schultheiss and so Gaussian assumption was favorable, but today such information is more or less available (see Section 3.2). The second objection is still upto debate. After all, MSE can be weighted to account for perceptual criteria [35].

Hence disadvantages of Huang and Schultheiss scheme are as follows :

1. They assumed that all quantizers used are of the same type namely Shannon quantizers for Gaussian distributed sources and quadratic difference distortion measure.
2. They used KLT for it is the optimum transform, but it is not practical.

Besides, a different choice of unitary transform would not necessarily yield Gaussian coefficients for Gaussian input (see Section 3.2). Moreover it may not even have uncorrelated coefficients. So the scheme lacks flexibility

3. Allocated bits are not necessarily integer or may even be negative. Therefore rounding to nearest integer or zero is required which destroys the schemes presumed optimality.

Despite all these facts, the above scheme has been employed by many care-free researchers such as [54], [55], [27], and many others even contemporary ones [56], [57].

Wintz and Kurtenbach [58] noticed the unrealizable quantizer function above and used uniform quantizers. Their quantizer function was found experimentally and approximately as

$$f(b_{ij}) = \exp\left(-\frac{1}{2}b_{ij} \ln 10\right) \quad (3.29)$$

Then using Lagrange multipliers and treating b_{ij} as a continuous variable they minimized Equation (3.20) imposing the constraint in Equation (3.21). That is, solution of

$$\frac{\partial}{\partial b_{ij}} \left[\frac{1}{N \times N} \sum_{i,j=1}^N \sigma_{y_{ij}}^2 e^{-\frac{1}{2}b_{ij} \ln 10} - \lambda \left(\frac{B}{N \times N} - \frac{1}{N \times N} \sum_{i,j=1}^N b_{ij} \right) \right] \quad (3.30)$$

is sought for b_{ij} , $i, j = 1, 2, \dots, N$. The result was

$$b_{ij} = \frac{B}{N \times N} + \frac{2}{\ln 10} \ln \frac{\sigma_{y_{ij}}^2}{\left[\prod_{k,l=1}^N \sigma_{y_{kl}}^2 \right]^{1/N \times N}}. \quad (3.31)$$

Then the real valued bits assigned were rounded to the nearest integer and in case Equation (3.21) was not satisfied they applied two heuristic rules to adjust some of the b_{ij} .

This unequal bit assignment procedure was also used, though suboptimum, by some researchers in mostly wrong ways! (see the references in section 4.7 of [12]).

Finally Segall [59] modified the same rate-distortion theoretic approach and solved the problem of negative bit assignments. His algorithm was as follows :

1. Find the inverse function of the derivative of the quantizer function as $h(x) \triangleq f'^{-1}(x)$ and find the root of the nonlinear equation

$$\frac{1}{N \times N} \sum_{i,j=1|\sigma_{y_{ij}}^2 > \theta}^N h\left(\frac{\theta f'(0)}{\sigma_{y_{ij}}^2}\right) = B. \quad (3.32)$$

The solution may be obtained by an iterative technique such as the Newton-Rhapson method.

2. The bit assignment is

$$b_{ij} = \begin{cases} \left(\frac{\theta^* f'(0)}{\sigma_{y_{ij}}^2} \right), & 0 \leq \theta^* \leq \sigma_{y_{ij}}^2, \\ 0, & \theta^* > \sigma_{y_{ij}}^2. \end{cases} \quad (3.33)$$

Notice that the coefficients whose variance falls below θ^* are not coded at all. For the inverse function to exist $f(\cdot)$ must be convex and nonincreasing. This assumption for the quantizer functions has been mentioned before. As a matter of fact, all scalar quantizers have such quantizer functions. Only vector quantizer may have non-convex and/or increasing functions [60], but they are beyond the scope of this chapter. The drawbacks of the method are

- Allocated bit values are still not integer. There is no upper limit to the allocatable number of bits for a coefficient.
- all quantizer are assumed to be of the same type and so, have the same quantizer function.

For these remaining drawbacks it was again Segall who suggested a completely different scheme based on marginal analysis.

3.3.2 Marginal Analysis Approach

Realizing the requirement of integer bit allocation algorithms Segall, suggested that marginal returns could be used [59]. The concept of marginal returns was first introduced by Fox [61] for a completely different task. In the economics literature there have been several treatments of a resource allocation problem. This problem is not only mathematically equivalent to the bit allocation problem, but also it also requires non-negative integer results. One solution was due to Fox. Pratt ([13] section 6.2) made use of the Segall's suggestion and developed an integer bit allocation algorithm, which was also favoured by Jain, later [20]. Outline of the algorithm is as follows :

1. Initially set the allocations as $b_{ij}^0 = 0$ and set iteration index $n=1$.
2. $b_{ij}^n = b_{ij}^{n-1} + \delta(i-l, j-m)$, where for the lm^{th} coefficient the marginal return $\sigma_{y_{ij}}^2 [f(b_{ij}^{n-1}) - f(b_{ij}^{n-1} + 1)]$ is maximum.

3. If $\sum_{i,j=1}^N b_{ij} = B$, stop; otherwise $j \rightarrow j + 1$ and go to step 2.

$\delta(\cdot, \cdot)$ above is the Kronecker delta function. If ties occur in the marginal returns a bit is given to all the coefficients which are tied and algorithm is initiated again with those allocations.

For a total number of B bits, $N \times N$ marginal returns must be searched B times, which requires $(N^2 - 1)B$ comparisons. For image coding since $N \times N$ is typically 8×8 , 16×16 , or 64×64 , the computational burden becomes quite severe. Jain proposed a slight change in the algorithm to speed it up, if the quantizers used have the MSE distortion function of the form $f(n) = k2^{-pn}$ [4]. Actual improvement in the algorithm was made by Tzou [62], who noticed that at any iteration if two coefficients already have same number of bits assigned then the one with larger variance takes the next bit. So the key to his method for computational reduction is sorting the variances in descending order, before the bit allocation procedure. The algorithm he proposed is runs about $\frac{N \times N}{8}$ times faster than the above procedure. A simple computer program, that implements Tzou's method exists in appendix A.5 of [8].

Marginal analysis approach results in the global minimization of the quantization error in block quantization if the MSE distortion functions of the quantizers used are convex. So is the case for optimal nonuniform Lloyd-Max quantizers tailored for various pdfs. The algorithm uses the quantizer function directly and so does not have to make the assumption of Gaussian input nor usage of a Shannon quantizer. Therefore, different type of quantizers and/or same type of quantizer tailored for different pdfs can be used. For instance non-DC DCT coefficients are shown to exhibit Laplacian distribution while DC coefficients have Gaussian distribution (see Section 3.2). Then this algorithm can be used for block quantization of DCT blocks with Lloyd-Max quantizers tailored for Gaussian or Laplacian whenever needed. It outputs optimal positive integer values and can even be modified to limit the maximum allocatable number of bits, by deleting those coefficients which has already been allocated maximum number of bits from the search list.

3.3.3 Variance Estimation for Bit Allocation

Variations of transform coefficients are essential ingredients of any bit allocation algorithm.

Assume that image sequences consisting of frames of size $M \times M$ are fed to the system of Figure 2.1 and each frame is divided into sub-blocks of size $N \times N$. In practice, a large number of test images can be used to estimate the coefficient variances. If the $N \times N$ image sub-blocks are assumed to be governed by a zero mean separable Markov process one can construct the row and column correlation matrices in the spatial domain

$$[Cor_r] = \sigma_r^2 \begin{bmatrix} 1 & \rho_r & \rho_r^2 & \dots & \dots & \rho_r^{N-2} & \rho_r^{N-1} \\ \rho_r & 1 & \rho_r & \rho_r^2 & \ddots & \ddots & \rho_r^{N-2} \\ \rho_r^2 & \rho_r & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \rho_r^2 & \ddots & \ddots & \ddots & \rho_r^2 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 & \rho_r & \rho_r^2 \\ \rho_r^{N-2} & \ddots & \ddots & \rho_r^2 & \rho_r & 1 & \rho_r \\ \rho_r^{N-1} & \rho_r^{N-2} & \dots & \dots & \rho_r^2 & \rho_r & 1 \end{bmatrix} \quad (3.34)$$

$$[Cor_c] = \sigma_c^2 \begin{bmatrix} 1 & \rho_c & \rho_c^2 & \dots & \dots & \rho_c^{N-2} & \rho_c^{N-1} \\ \rho_c & 1 & \rho_c & \rho_c^2 & \ddots & \ddots & \rho_c^{N-2} \\ \rho_c^2 & \rho_c & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \rho_c^2 & \ddots & \ddots & \ddots & \rho_c^2 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 & \rho_c & \rho_c^2 \\ \rho_c^{N-2} & \ddots & \ddots & \rho_c^2 & \rho_c & 1 & \rho_c \\ \rho_c^{N-1} & \rho_c^{N-2} & \dots & \dots & \rho_c^2 & \rho_c & 1 \end{bmatrix} \quad (3.35)$$

where ρ_r and ρ_c are adjacent row and column correlation coefficients respectively and usually they are assumed to be about 0.9. Transforming these two matrices, say with DCT-II one obtains $[Cor_r^{DCT-II}]$ and $[Cor_c^{DCT-II}]$. Then variance of the ij^{th} coefficient is

$$\sigma_{DCT-II}^2(i, j) = \sigma_{r, DCT-II}^2(i, i) \sigma_{c, DCT-II}^2(j, j) \quad (3.36)$$

where $\sigma_{r, DCT-II}^2(i, i)$ and $\sigma_{c, DCT-II}^2(j, j)$ are the diagonal elements of the matrices $[Cor_r^{DCT-II}]$ and $[Cor_c^{DCT-II}]$ respectively [63].

Another way of calculating coefficient variances is Mandela re-ordering [34]. It indicates the sorting of those coefficients which are at the same spatial frequency

at each sub-block. That is the $(i, j)^{th}$ Mandela block consists of $(M/N)^2$ coefficients, gathered by selecting the coefficient at the spatial coordinate (i, j) of each transformed block. Computationally the variance calculation from the Mandela re-ordered sub-blocks can be expressed as follows, (with a slight change in the coefficient notation of Section 3.3)

$$\sigma_{DCT-II}^2(i, j) = \frac{N \times N}{M \times M} \sum_{k=0}^{(M/N)-1} \sum_{l=0}^{(M/N)-1} [y(i + kN, j + lN) - \bar{y}(i, j)]^2 \quad (3.37)$$

where $\bar{y}(i, j)$ is the mean value of the coefficient at the $(i, j)^{th}$ position in the block. For DCT, only the DC coefficient has a mean value and it can be calculated by averaging all the DC coefficients. Non-DC coefficients have their mean values 0.

3.4 Human Visual System in Transform Coding

For any image processing system, the final observer of the result is almost always human. Then, it would be logical to incorporate the human visual system (HVS) characteristics into the image coding process, so that subjectively more pleasing results could be obtained.

To set the terminology, in this section the term ‘apparent brightness’ or shortly ‘brightness’ will be used to refer a perceptual entity meaning the perceived brightness of a point, while ‘intensity’ or equivalently ‘luminance’ will refer to a physical quantity meaning the amount of light radiated from the point.

The ‘luminance’ of an object is independent of the luminances of the surrounding objects, while ‘brightness’ of an object which is the perceived luminance of it, depends highly on the luminance of the surround. Two objects with identical luminances could have different brightnesses under different surroundings.

Basically the eye behaves as a function of four independent variables :

- Intensity
- Colour
- Variations in spatial details
- Variations in temporal details.

Dependence on light intensity and variations in spatial detail will be covered in the following sections. But color dependence is excluded as it is not directly

related with the statistics of the image and so can not be adapted to the statistical compression techniques that will be presented in this thesis.

On the other hand, in terms of dependence on temporal details the most important phenomena occurs when observing a repeatedly flashing source of light at a constant rate. If this rate is above a, so called 'critical fusion frequency', the flashes of the light source are indistinguishable from steady light of the same average intensity. Critical fusion frequency, generally does not exceed 50-60 Hz [64]. Therefore temporal response should be considered in the design of image displays, frame grabbers etc.

Properties of HVS can be incorporated to the following areas of image coding :

- Image Quality Assessment
- Image Compression
- Image Enhancement.

HVS characteristics in response to light intensity and spatial factors have not, yet, been specified completely, and so have not been applied to image coding effectively. Still, to demonstrate the usage of a more multidisciplinary approach, they have been incorporated to some of the data compression schemes tested in this thesis. For that reason, it has been attempted to use an inefficient but correct model for HVS, so that the image coding scheme can be modified to preserve information to which the human eye is sensitive.

In the following sections neurobiological and psychophysical properties of the human visual system will be discussed with sufficient extend with an aim to derive a practical model.

3.4.1 Neurobiology of the Visual System

It is essential that, before constructing models for the HVS, a brief, functional examination of the visual perception should be made. The visual system can operate at levels of illumination varying over ten orders of magnitude (i.e., ten log units or a range of 10^{11} !). Cornea and lens focus the light on the retina (see Figure 3.3). The lens has an aperture, controlled by the iris, which is variable over the range 2-8 mm. in response to ambient illumination. The retina contains two

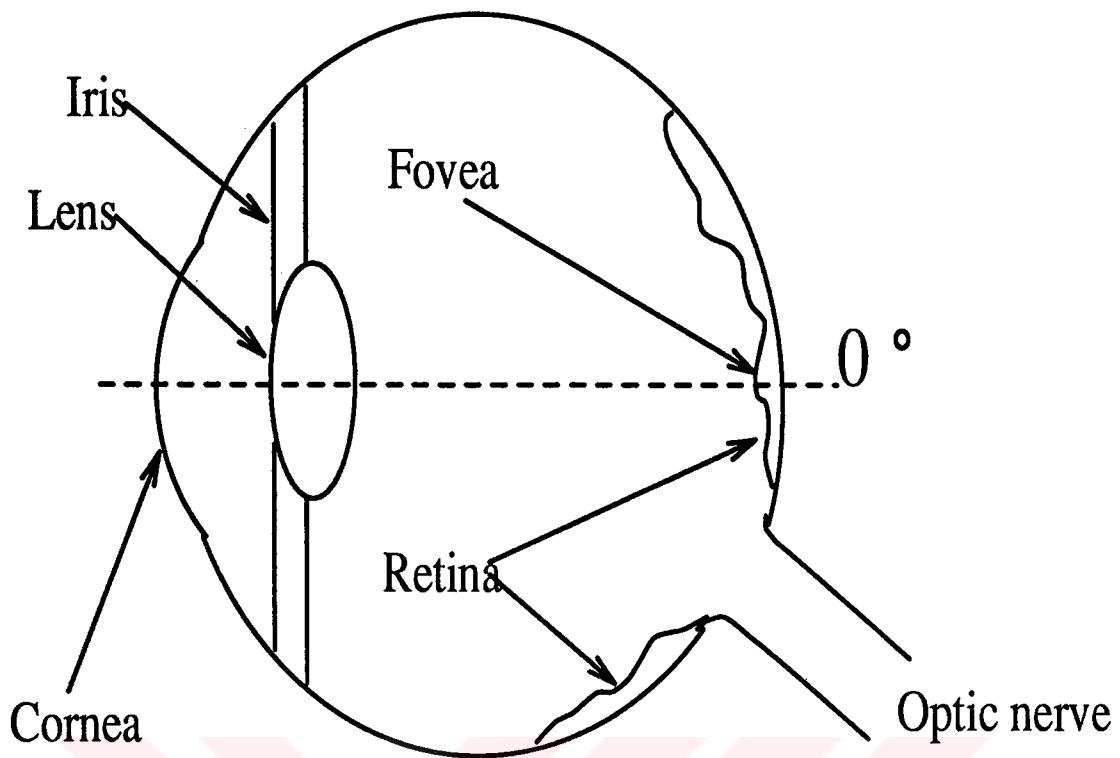


Figure 3.3. Cross-section of the eye

types of photoreceptors, called 'rods' and 'cones', which detects the incident light intensity by means of a photochemical reaction. The rods, about hundred million in number, are relatively long and thin. Although they lack colour response, they are extremely sensitive (responding to one out of ten photons). They take part in reception for the lower several orders of magnitude termed as 'scotopic' region, which goes from darkness to a dimly lighted room (night vision is achromatic). The cones are shorter and thicker. Some 6.5 million cones are responsible from colour vision and the higher 5-6 orders of magnitude, called as the 'photopic' region (goes from well lighted room to bright sunlight). Both cones and rods are active in the region in between, which is termed as the 'mesopic' region. As a result the response to additional light depends highly on background illumination. Our interest is primarily the photopic region, since electronic image displays are well-lighted.

The distribution of both types of detector across the retina is highly nonuniform. The cones are packed in the center of the retina (termed as the 'fovea'), near the optical axis of the lens, at a density of 120 cones per degree of arc subtended in the field of vision. It is interesting that, in bright light the pupil of the

eye is limited to 2 mm in diameter by the iris, and acts like a low-pass spatial filter (for green light) with a passband of 60 cycles/degree. Then, the human eye filters out spatial frequencies above 60 cycles/degree, eliminating aliasing errors in the sampled image formed by the cones spaced at a density of 120 cones/degree (as if the designer was aware of the sampling theorem!). Maximum visual acuity is on the fovea, having an area of 1.5° where the viewer fixes his attention with highest resolution. Cones end about 7° off axis, and rods concentrate at 20° off axis. Rods fall half the density at visual periphery (75°) and vanish at the fovea.

Considering the encoding and transmission part of the system, the cones which are 'laterally connected' to 'horizontal cells', are also connected to 'bipolar cells' extending forward through the nerve layer of the retina towards the lens. Bipolar cells connect 'ganglion cells', which join to form the 'optical nerve'. The optical nerve, consisting of a bundle of 1 million fibers, leaves the eye through the retina at some 15° off axis, causing a blind spot, and transmits the neural image to the brain. Notice that due to fewer number of optical nerve fibers compared to that of cones and rods, the retina itself performs image compression by using the interconnected neural cells mentioned above. Compression ratio is 7:1 in daylight and 100:1 at dark.

Above discussion is related to the neurobiological properties of the visual system. Neurobiology deals with the structure the nervous system, or neuroanatomy and its functions, or neurophysiology. Since 1950's, many discoveries on the subject have been made. From image coding point of view, psychophysiological and psychophysical properties of the visual system is important, as a model for the spatial frequency of it is sought. So much is sufficient for the neurobiology of HVS, to establish a basis for better understanding of psychophysiology of it. But neurobiology of the vision is a very interesting subject, so interested readers are strongly encouraged to refer to the excellent book of Nobel prize winner author D. H. Hubel, [65], for further contemporary information.

3.4.2 HVS Modeling

Before attempting to model the HVS, it is necessary to identify some basic perceptual phenomena about it. Such phenomena mainly depends on the perception of brightness and darkness by human beings which is mainly effected by 'contrast' existing in the view.

The most important perceptual phenomena about the sense of brightness is the one called 'brightness constancy'. It stands for the fact that the sense one has about the brightness of a real object in a natural environment is largely independent of the changes in the overall level of illumination. For example, a white page of paper is sensed as "white", no matter how it is illuminated, whether by bright sunshine or by the much weaker light of a desk lamp. In other words, human perception tends to keep constant the brightness "attributes" of objects despite variations in the overall level of illumination. Without this constancy a white page would assume all possible gray levels that lie between white and black, in the course of a day if the only source of light is the sun.

However, this constant sense of brightness is altered if there is a 'contrast' between the object and its background or between the object and the surrounding objects. In case of a contrasting background the altered sense of brightness can be explained by contrast laws (refer [66], section 4.5.1).

In case of multiple objects, a large number of different effects may occur. A single illuminated object, in a completely dark surround, always appears bright. Even a piece of coal appears "bright" if illuminated under such isolation conditions. However, when two objects of different reflectances are placed next to each other, the previous sensation is modified. For instance, if a piece of chalk is placed against the coal in the latter example, chalk turns the coal into "black". Consequently, the chalk appears "white". Or if the background illumination of darkness around the coal is increased the coal begins to appear "darker". Thus, 'brightness' (or 'darkness') sensation depends on spatial and temporal changes in the physical distribution of light flux reaching our eyes from various portions of the visual field.

Due to the above facts, the types of contrast can be generalized into two :

1. Successive Contrast : The dependency of the brightness of a surface (region) to the surface preceding it in time. As an example of this case, where temporal changes in luminance influence the brightness, consider prolonged viewing of a constant luminance. Then local adaptation to the region would occur, and aftereffects would occur. Examples of such aftereffects are those that occur when eyes are closed after viewing a bright object or the sustained stationary image one has, after removing his/her sight from a very fast

moving (or turning) object. Such after images are called ‘dark light’. On the other hand, if the stimulus contrast is not too large, a prolonged exposure results in a reduction of apparent contrast and even stimulus fading. Then, eye movements or blinks will revive the faded image (but not for too long).

2. Simultaneous Contrast : The dependency of the brightness of one region on the relative brightnesses of adjacent regions. A single object (assuming no background exists!) under constant illumination appears brighter or darker, depending only on whether it reflects a higher or lower percentage of the incident light. However, its brightness would be altered if at the background of the objects there exist brighter or darker regions adjacent to its borders.

Effects of simultaneous contrast and successive contrast can be best observed through ‘visual illusions’. Visual illusions, which are due to some contrast distribution, help to extract salient features of HVS since they are the direct consequences of perceptual phenomena in humans under natural views. Therefore, discovering neurobiological and psychophysical properties of HVS to account for visual illusions would lead us to correct HVS models. Therefore, the more visual illusions a model accounts for, the more efficient it is. Unfortunately, there exists no way of incorporating phenomena due to successive contrast into a practical HVS model. Therefore, the attention is focused to simultaneous contrast in this thesis.

3.4.2.1 Simultaneous Contrast

The alteration of brightness constancy of an object, in front of a contrasting background has been explained above, by contrast laws. The viewing condition where there exist a uniform (homogeneous) background or a nonuniform background (the case of multiple objects), is said to have simultaneous contrast.

In the case of uniform background, the psychophysical experiments to obtain the contrast laws are called “just noticeable difference” experiments [67]. It consists of presenting to the subject two neighboring illuminated areas, one with luminance L and the other with $L + \Delta L$. The incremental difference ΔL is increased until the brightness difference between the two adjacent regions can just be noticed by the subject. The procedure is carried out for a whole range of L values. Traditionally ([66], page 140), human behaviour in this context has been

described by the linear equation, known as Weber's law :

$$\Delta C_W = \frac{\Delta L}{L}, \quad (3.38)$$

where ΔC_W is the change in contrast and it is constant. An alternative relationship was suggested by Fechner over 100 years ago:

$$\Delta C_F = \frac{\Delta L}{\log_{10} L}. \quad (3.39)$$

Accordingly, from Equation (3.39) one obtains $C_F = a_1 + a_2 \log_{10} L$.

Obviously, these laws can not be applied for the whole range of L . Brightness constancy is limited by the saturation effects that occur in extreme cases of illumination (see [68] section II.B.2). If the light level is very low the range of apparent brightness is very restricted. In complete darkness one perceives no brightness differences. Moreover, the above laws account for different portions of the perceptible L range. In fact, they coincide for low L region where $\log_{10} L \propto L$. Due to their limited applicability, in image coding Stevens' power law

$$C_S = L^{1/n} \quad (3.40)$$

is more widely preferred and n is often chosen to be 3 [69]. One important connotation of the laws above in image coding is that a disturbance of a given magnitude (noise for example) will be more noticeable in the darker areas of an image.

Until now it is assumed that the viewed object has a uniform brightness over its surface. Therefore this situation can be called 'area contrast'. A more careful examination of brightness of objects surface would reveal another consequence of simultaneous contrast, called 'border contrast' effects.

Since simultaneous contrast is due to spatially adjacent regions to an object, its effects are most striking at the borders of the object. These are called 'border contrast' effects. Only if the borders enclose a large homogeneous region, a change of brightness, spreads uniformly over the entire region and 'area contrast' effects occur.

The most frequently encountered visual illusions, which demonstrate the border contrast effects are the Cornsweet illusion and Mach bands [67]. Examine double Mach bands from Figure 3.4. The intensity distribution is actually plotted in Figure 3.5 in part (B). But the brightness distribution exhibits a brighter

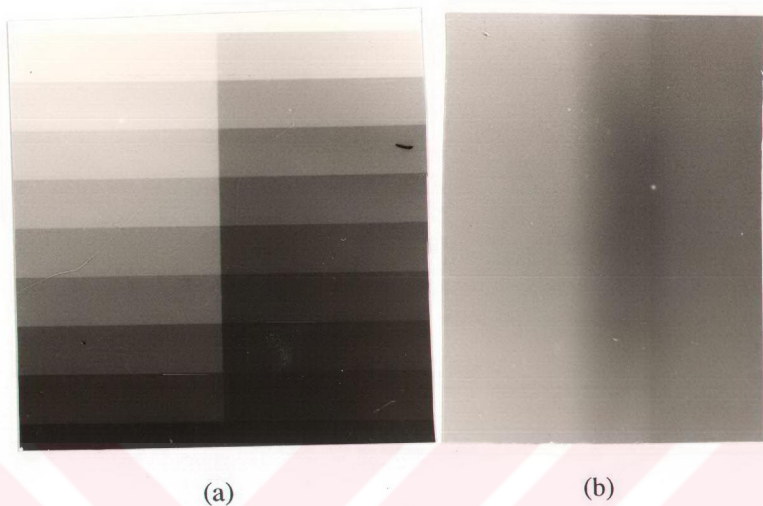


Figure 3.4. (A) Multiple Mach bands and (B) double Mach bands

stripe in the region before intensity degradation starts and a darker one before it starts increasing. The example of multiple Mach bands demonstrates both border contrast and area contrast effects. The Mach bands on the left has an decreasing light intensity, from top to bottom, with identical steps. However, the brightness does not “seem” to decrease uniformly. Rather it decreases fast at the top, then a sudden decrease is observed towards the bottom. This confirms the logarithmic law of Fechner’s. The multiple Mach bands at the right hand side has a decreasing light intensity in a logarithmic fashion and hence seems to have more uniformly decreasing brightness.

The Mach band effect shows that the threshold of noticeable difference ΔL in the contrast laws is sensitive to the presence of a nearby luminance transition. In image coding, making use of this ‘spatial masking’ property of human vision, there has been effort to “hide” the coding artifacts around the edges where it is least noticeable [70]. Therefore, once the edge is reconstructed well after coding, it would mask the distortion around it.

Aside from double Mach bands another, striking visual illusion due to border effects of simultaneous contrast is the Hermann grid illusion, illustrated in

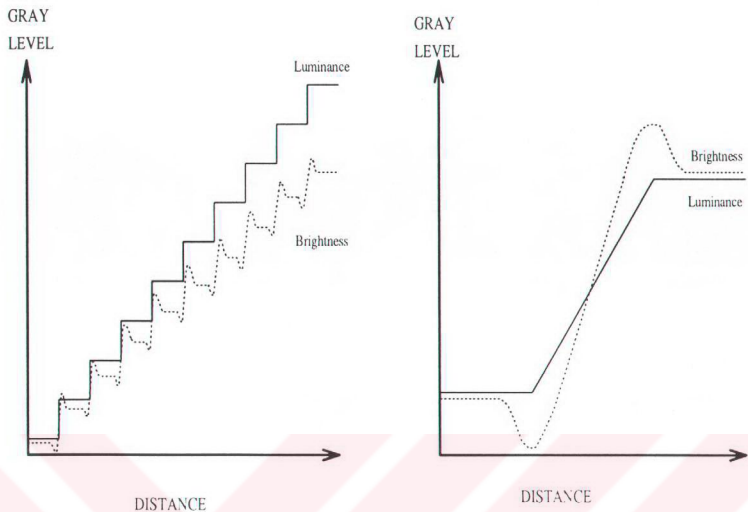


Figure 3.5. Intensity vs brightness distribution of (A) multiple Mach bands and (B) Double Mach Bands with brightness exaggerated

Figure 3.6. Dark blobs appear at the intersections. However, if one focuses his/her eye at a particular intersection, that is if an intersection is viewed 'foveally' (recall Section 3.4.1), then the blob vanishes.

Certain combinations of incomplete figures generate edges, which are clearly visible although there exists no corresponding luminance steps for such edges. An example is given in Figure Figure 3.6.7. Also, observe in the figure that the surface outlined by the illusory contour appears lighter in brightness than the white page surrounding it. From this point of view, it is interesting to observe that the perception of an edge, even an illusory edge is accompanied by the area contrast effect.

In the next section, the neurophysiological reasons underlying the above psychophysical phenomenon will be given.

3.4.2.2 Neurophysiological Reasons of Visual Illusions

In Section 3.4.1 it has been mentioned that the eye is able to respond to an enormous range of intensities (of the order of 10^{11}). This suggests that it

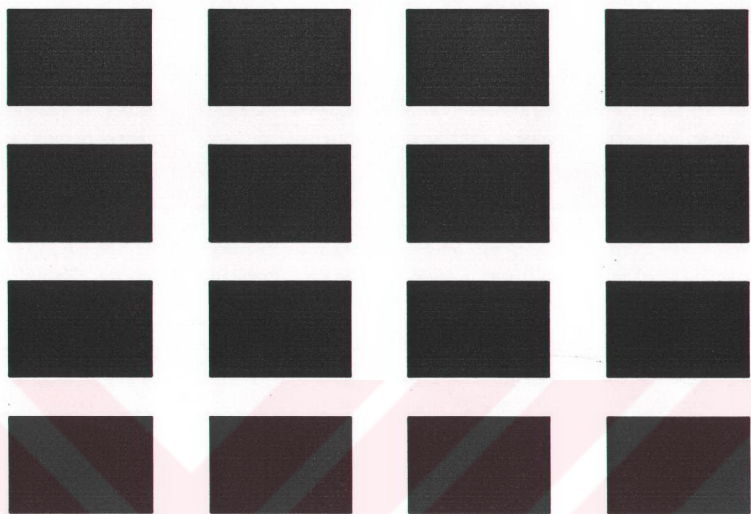


Figure 3.6. Hermann grid illusion

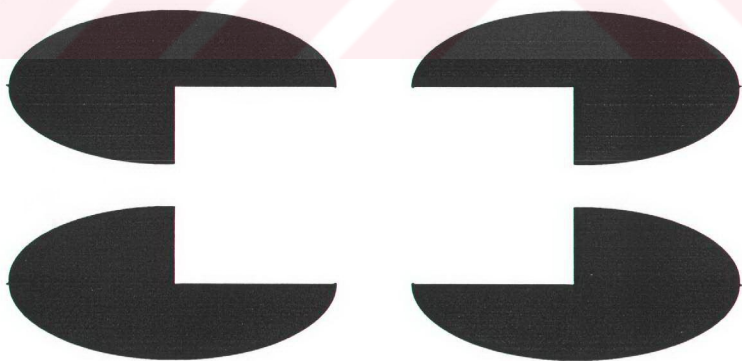


Figure 3.7. An example of illusory contours

should have an adaptation mechanism to varying range of intensities. One way of accomplishing this is to undo the increase in illumination, i.e. reduce the effective retinal illumination. This is called 'dark-glasses' principle and it is provided by changing pupil aperture. But the pupil diameter can vary only about 4:1. So it is apparent that a very large degree of adaptation exists at the receptor level. A signal with a frequency span of $10^{11}/4^2$ can not (at least physically!) travel along the optical nerve. Actually, the frequency does not exceed a maximum of perhaps 1 kHz ([66], section 4.5.1). Since linearity prevails between impulse frequency at the output of the ganglion cell and the photoreceptors' potential output at its receptive field, it is clear that this data compression is a result of a photochemical reaction at the transducing elements (rods and cones). The neurophysiology of these reactions are explained in [71] section IV.A. Neurophysiological experiments carried out by many scientists reveal that (see [71] section III.A.4), the membrane potential of photoreceptors are actually proportional to the contrast which is related to the intensity through the contrast laws given in the previous section. As a result, the photochemical data compression at the photoreceptor level accounts for the visual phenomena of brightness constancy.

The area of retinal photoreceptors which can influence the behaviour of a ganglion cell, whose axons make up an optical nerve, is called the 'receptive field' of that cell. In 1953 Kuffler [72] has shown with psychophysiological experiments that these fields have an excitatory region at the center and a surrounding antagonistic region. Light shining on a photoreceptor at the excitatory region changes its membrane potential which ultimately raises the firing rate of the impulse train at its ganglion cell output. On the other hand, light falling on a photoreceptor at the surrounding region inhibits and decreases this firing rate. In general, the effects of all inputs, excitatory and inhibitory undergoes an algebraic summation at the synapse connecting the ganglion cell to the optical fiber, forming a point spread function of shape like a Mexican sombrero (the neurophysiological reasons are explained in [67], chapter 6 and [71] section IV.C). Figure 3.8 illustrates this very important phenomena due to spatial interactions in the retinal receptive fields, which is called 'lateral inhibition'. Kuffler also, discerned two kinds of ganglion cells: 'ON-centered' cells which increase their firing rate when light "increments" are presented at their excitatory receptive fields, and 'OFF-centered' cells which are excited by light decrements.

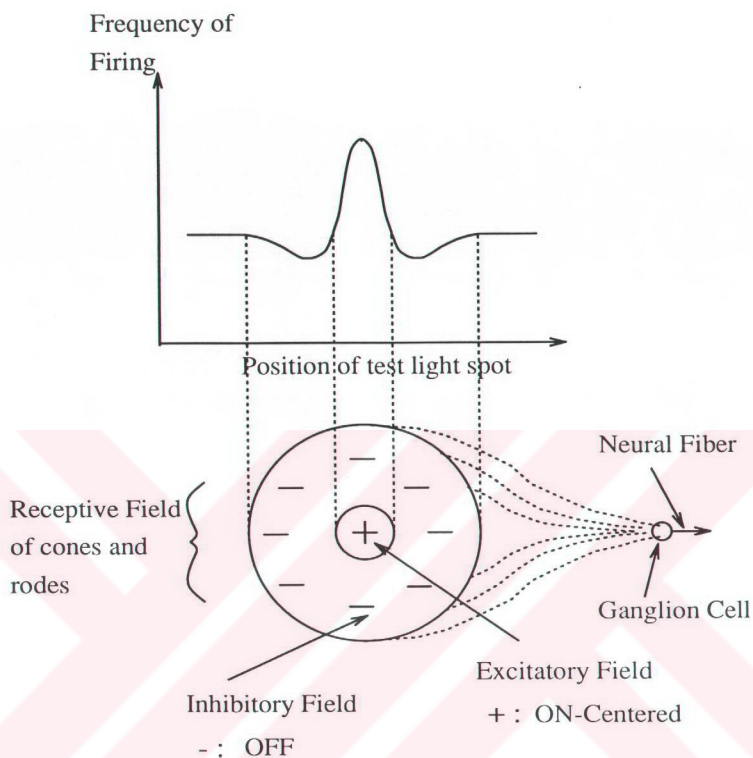


Figure 3.8. Lateral inhibition

To find the point spread function in Figure 3.8, a flashing spot of light is focused on the middle of the excitatory region of a particular receptive field and the frequency of impulses at the corresponding optical fiber is recorded. Then the diameter of the spot is increased until at a threshold value at which the firing rate suddenly drops. If the diameter is further increased, a saturation level at the firing rate can be reached meaning that the spot covers the whole inhibitory surround. Since the spot approximates the impulse excitation in systems theory, the so obtained point spread function is the spatial impulse response of the HVS. Calculating its Fourier Transform one can get the 'Transfer Function' of the HVS. Recall from linear systems theory that there exist two ways of obtaining

the transfer function experimentally. The first one is to apply sinusoidals at every frequency, then the curve tracing the amplitudes of output sinusoidals would be the transfer function (output is also a sinusoid since sinusoidals are eigen functions of linear systems). Another way, is to apply a square wave, then the transfer function can be calculated using the response and the Fourier transform of the square wave excitation. Either way is applicable to the spatial case to obtain the so called ‘Modulation Transfer Function’ (MTF). The word “modulation” indicates that, since the eye is sensitive to only the ratios of illumination (contrast, contrast laws), the square-wave or sine-wave gratings that will be used should have a constant DC component. In ordinary linear systems the input (stimulus) square or sine waves should have amplitude one, but since there is no notion of a unit for HVS, one can vary that amplitude of the stimulus wave until it is just noticeable for the subject. Hence, the MTF obtained in this way is also called the ‘Contrast Sensitivity Function’ (CSF) or the ‘Threshold Sensitivity Function’ of the HVS. One can observe his/her own CSF in Figure 3.9, where sine waves of increasing frequency in horizontal direction and decreasing amplitude in vertical direction is drawn on a constant DC level. Notice that, it has the form of a high pass filter, with a peak at a certain spatial frequency.

The lateral inhibition phenomena accounts for the bordering effects in simultaneous contrast, and hence explains the related illusions. Mach in 1866 first predicted that due to the Mach bands illusion, each functional field in the retina (now called receptive fields) should have an excitatory component next to an inhibitory component. Today we know that if the intensity distribution function of double Mach bands is passed through a linear system with MTF as in Figure 3.9, the corresponding brightness distribution can be obtained, which is directly related to the phenomena of lateral inhibition.

Lateral inhibition also explains the Hermann grid illusion. Relationship between the area of intersection in the Hermann grid illusion and the strength of illusion provides estimates for the spatial ranges of the excitatory and inhibitory interactions. In particular, increasing the area of intersection until the illusion vanishes one can determine the diameter of the center and antagonistic surround of the receptive field which is the neurophysiological correlate of that area. Detailed explanations of the illusions with lateral inhibition can be found in [68].

Before examining models for the HVS, there is one more factor which is

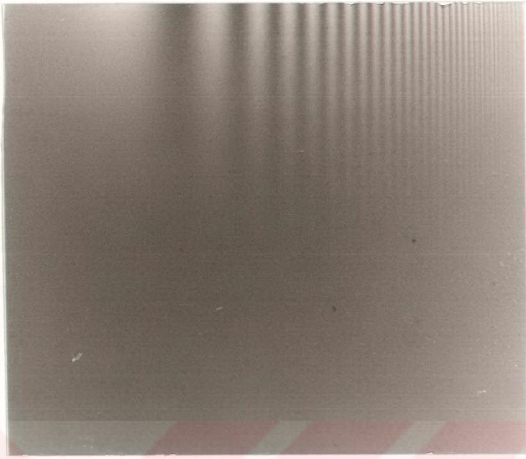


Figure 3.9. Experimental sinusoidal gratings varying in frequency and amplitude to determine human visual system MTF

independent of the retinal and neural functions, but is the effect of purely the optical (pupil/lens) system. Westheimer and Campbell [73] have measured the distribution of light on the retina formed by the human eye optics and obtained an isotropic, 2D lowpass filter with line spread function of the form

$$h(x) = e^{-\lambda|x|}, \quad (3.41)$$

where x is in minutes of arc and $\lambda = 0.7$ for a pupil diameter of 3 mm. Then, Fourier Transform of $h(x)$ gives the spatial response

$$H(\omega) = \frac{2\lambda}{\lambda^2 + \omega^2}. \quad (3.42)$$

3.4.2.3 Monochrome Vision Models

For developing a model the psychophysical phenomenon that should be taken into account are :

1. Spherical aberration of the lens and the corresponding low pass filter characteristics given in Equation (3.42).

2. Amplitude (point) nonlinearity at photoreceptors due to the light adaptation requirement.
3. MTF with high pass characteristics, that occur in the ganglion cells due to lateral inhibition.

These properties form the so called ‘early vision’ part of the HVS. Note that, the phenomena of lateral inhibition is represented by the MTF described in Section 3.4.2.2. But to do so the system should be considered linear, homogeneous, and isotropic. But HVS has none of these properties. Still it can be assumed to operate at the “linear region of the nonlinearity” if measurements are confined to small increments of luminance [74]. The photoreceptors do not have homogeneous distribution over the retina (recall Section 3.4.1) but the amount of uncertainty this fact adds to a models output which is developed for homogeneous systems is unknown, yet experimental data predicts that it is not much. The MTF of Figure 3.9 is also called line spread function, since it is found by inputting 1D stripes. If the system was isotropic (circularly symmetric), it can readily be extended to 2D MTF, otherwise 2D MTF must be calculated by using oriented lines. Such experiments show that the response is minimum at 45° , where a frequency of 10 cycles/degree is reduced only by 15% from the value it has for the vertical grating case, meaning that the change can be ignored [74].

After this early vision part of the system information is carried to the striate cortex for further processing. At the cortex level Hubel and Wiesel [65] has found that there exists two kinds of cells : simple and complex. Simple cells behave like ganglion cells and have receptive fields formed from excitatory and inhibitory optical nerves. However, complex cells do not have such regions in their receptive fields, rather they react to the intensity changes of stimulus at certain orientation. This phenomenon that about the existence of spatially selective filtering at the cortex was also reported later by Campbell and Robson, with psychophysical experiments [75].

A model for the HVS covering all the above phenomenon would look like Figure 3.10. Many researchers constructed multiple channel models (refer [69] and [68] section V.C) depending on the findings of Campbell and Robson. But from the image coding point of view they are not useful, because of their complexity. On the other hand Davisson [76] supplied experimental data for the output

of early vision part by using exponential sine wave gratings. Since HVS has logarithmic point non-linearity at its input block (Fechner law) the exponential would be cancelled and direct data on the MTF could be obtained. Hall and Hall [74] developed a pioneering single channel model using this data and later Mannos and Sakrison described the MTF with a, now famous, expression

$$H(f) = a(b + cf)exp[(-cf)^d]. \quad (3.43)$$

3.4.3 Perceptual Domain Transform Coding

Application of the model of Figure 3.10 is quite simple to transform domain coding schemes. Since, transform coding is, in fact, a spatial frequency decomposition, each transform coefficient can be considered to correspond to a frequency channel. Then, the frequency channel selectivity of the cortical processing would be satisfied. As a further simplification the first two blocks of low pass filtering and nonlinearity due to physical properties of eye and adaptation mechanism of photoreceptors respectively can be thought to be neutralized. Using the properties of the display devices that are employed to view the images.

Since all monitors have finite resolution, the spatial details almost always remains within the pass band of the low pass filter for normal viewing distances. Moreover, monitors using cathode-ray tubes converts the input signal to light display with exponential nonlinearity. Therefore, the logarithmic nonlinearity due to Fechner's law would also be cancelled [78]. That leaves us with the MTF described by the Mannos and Sakrison's expression.

Passing transformed image blocks from the MTF of HVS, one would be imitating the HVS and consequently would obtain the mapping of the image to the 'perceptual domain'. Stockham has proved experimentally that, coding of images in the perceptual would give better subjective results at the reconstruction [79]. The reason is that, the distortion introduced to the frequency components at which the HVS is sensitive would not be amplified by the HVS. Rather, the distortion would be distributed to all frequency in a perceptually equal fashion.

Mannos and Sakrison after extensive experiments have arrived at the following quantitative HVS model:

$$H(f) = 2.6(0.192 + 0.114f)exp[-(0.114f)^{1.1}] \quad (3.44)$$

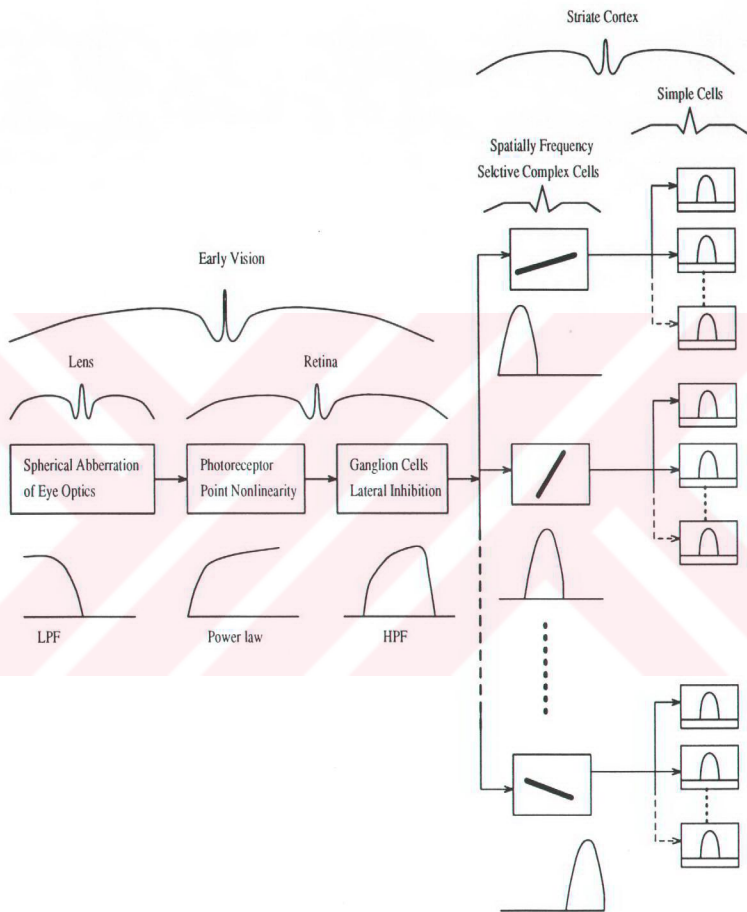


Figure 3.10. A complete model for human visual system

This MTF has a peak at $f=8$ cycles/degree.

Incorporation of HVS in the transform domain implies that the DFT coefficients should be multiplied with $H(f)$ at the corresponding frequencies. A DFT coefficient at the $(i, j)^{th}$ spatial location corresponds to $f = \sqrt{i^2 + j^2}$. With the impact of DCT in transform coding, Nill has noticed that, since DCT is actually the DFT of the even extended image (recall Section 2.2) and the viewer is not observing this extended image, the physical significance of DCT coefficients for the HVS would be lost. Therefore Nill [80] proposed that the MTF should be multiplied by the weighting function $A(f)$ defined as follows :

$$|A(f)| = \left(\frac{1}{4} + \frac{1}{\pi^2} \left[\ln \left(\frac{2\pi f}{\alpha} + \sqrt{\frac{4\pi^2 f^2}{\alpha^2} + 1} \right) \right]^2 \right)^{1/2} \quad (3.45)$$

where α has the typical value of 11.636 1/degree. He also proposed the following as a good representation of HVS :

$$H(f) = (0.2 + 0.45f)exp[-(0.18f)] \quad (3.46)$$

which has a peak at around $f=5$ cycles/degree. However Nill, has not supported his work with experimental results on images. Later Ngan et al. [81] have used Nill's multiplicative function $A(f)$ with their MTF in the DCT domain adaptive coding scheme that they developed. The MTF they used had a peak at $f=3$ cycles/degree and had the form

$$H(f) = (0.2 + 0.45f)exp[-(0.18f)] \quad (3.47)$$

After multiplying this MTF by $A(f)$, the peak frequency shifts to 4 cycles/degree. They achieved acceptable quality reproductions at rates of 0.2 or 0.3 bpp.

Very recently, Chitprasert and Rao [82] developed the following MTF for DCT weighting based on extensive experiments :

$$H(f) = 2.46(0.1 + 0.25f)exp(-0.25f). \quad (3.48)$$

It had a peak at 3.75 cycles/degree. For 8×8 DCT-II this MTF results in the

following weighting matrix :

$$\begin{bmatrix} 0.4942 & 1.0000 & 0.7023 & 0.3814 & 0.1856 & 0.0849 & 0.0374 & 0.0160 \\ 1.0000 & 0.4549 & 0.3085 & 0.1706 & 0.0845 & 0.0392 & 0.0174 & 0.0075 \\ 0.7023 & 0.3085 & 0.2139 & 0.1244 & 0.0645 & 0.0311 & 0.0142 & 0.0063 \\ 0.3814 & 0.1706 & 0.1244 & 0.0771 & 0.0425 & 0.0215 & 0.0103 & 0.0047 \\ 0.1856 & 0.0845 & 0.0645 & 0.0425 & 0.0246 & 0.0133 & 0.0067 & 0.0032 \\ 0.0849 & 0.0392 & 0.0311 & 0.0215 & 0.0133 & 0.0075 & 0.0040 & 0.0020 \\ 0.0374 & 0.0174 & 0.0142 & 0.0103 & 0.0067 & 0.0040 & 0.0022 & 0.0011 \\ 0.0160 & 0.0075 & 0.0063 & 0.0047 & 0.0032 & 0.0020 & 0.0011 & 0.0006 \end{bmatrix}$$

3.5 Experimental Results

Consider the transform coding system of Figure 2.1. In Sections 3.1.2 and 3.3.1 it has been justified that optimal scalar quantizers for the DCT coefficients are nonuniform Lloyd-Max quantizers tailored for Gaussian pdf for DC coefficients and Laplacian pdf for non-DC coefficients. In Section 3.3, bit allocation problem has been extensively discussed and marginal analysis approach has been accepted as the most appropriate and optimal method for the bit assignment to above quantizers.

In the experiments performed in this thesis work simulations have been carried out on the popular 8 bpp gray level test sequence of Miss America, which is a 256×256 , 150 frame image sequence. Mandela re-ordering over the first three frames of the sequence is used for estimating the variances of the transform coefficients for the marginal analysis bit allocation scheme with Tzou's method (Section 3.3.2).

Resulting DC mean value was 69.447273, and variance matrix was

$$\begin{bmatrix} 1001.888 & 12.31474 & 2.092180 & 0.666371 & 0.263281 & 0.159800 & 0.099777 & 0.107693 \\ 9.338483 & 1.134139 & 0.454674 & 0.163167 & 0.093111 & 0.055508 & 0.037780 & 0.051546 \\ 2.638924 & 0.381950 & 0.228907 & 0.098627 & 0.052894 & 0.040769 & 0.028633 & 0.036653 \\ 0.947318 & 0.177208 & 0.084764 & 0.048139 & 0.035792 & 0.024271 & 0.018360 & 0.022453 \\ 0.300800 & 0.068600 & 0.035387 & 0.023661 & 0.019765 & 0.018055 & 0.010678 & 0.032635 \\ 0.150318 & 0.033848 & 0.014368 & 0.010657 & 0.010369 & 0.010679 & 0.006217 & 0.012386 \\ 0.074249 & 0.024604 & 0.010855 & 0.011842 & 0.007866 & 0.017729 & 0.006572 & 0.078064 \\ 0.050986 & 0.013281 & 0.007293 & 0.009072 & 0.005673 & 0.013931 & 0.005353 & 0.067355 \end{bmatrix}$$

Each coefficient is normalized using this matrix prior to quantization, so that scalar quantizers prepared for unity variance sources could be used. Note that, for the DC coefficients, mean value should be subtracted before normalization. Therefore the DC mean value and the variance matrix must be sent to the decoder. Then, DC coefficients are optimally scalar quantized into a pre-determined level of 8 bits, using nonuniform Lloyd-Max quantizers tailored for Gaussian pdf. It has been observed that if optimum uniform quantizers were used for the DC coefficients of the first frame the PSNR performance would be 38.59 dB, while for optimal nonuniform quantizers it is 38.61 dB. This choice of 8 bits for DC coefficients is common to most of the transform domain compression schemes, as it is better to quantize the DC coefficients finer, since, otherwise, blocking artifacts may occur at the reconstruction. In the literature there exists post-filtering techniques to overcome this type of distortion which is visually quite disturbing [83]. Comparisons of performance of many techniques with the rate-distortion bound reveal that block quantization performs the best within 1-2 bpp range ([51] section 5.1). Therefore decision is made on compressing each frame of the sequence into 1 bpp. Remaining 56 bits per each sub-block is then distributed to the Lloyd-Max quantizers tailored for Laplacian pdf to quantize the non-DC coefficients. The below map indicates the bits allocated to the corresponding transform coefficients. This map must also be sent to the decoder. Observe that most of the bits are allocated to the low frequency zone, reflecting the energy compaction property of DCT. Observe also that, 24 out of 64 coefficients are coded, which is more than 25%, the presumably enough percentage of coefficients for the satisfactory reconstruction (see Section 3.3.1).

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

To account for the human visual system properties, the three of the HVS

Table 3.4. Comparison of overall PSNR performances of the three systems which incorporate HVS with the system that does not include it.

	No HVS	HVS-1	HVS-2	HVS-3
PSNR	38.61	38.68	38.49	35.34

models described in Section 3.4.3 are compared to the result of the above system subjectively. The three MTF expressions for the HVS are

1. MTF of Chitprasert and Rao in Equation (3.48). Denoted as HVS-1.
2. MTF of Ngan et al. in Equation (3.47) together with the weighting function $A(f)$ in Equation (3.45). Denoted as HVS-2.
3. MTF of Nill in Equation (3.46) together with the weighting function $A(f)$ in Equation (3.45). Denoted as HVS-3.

It has been observed that the HVS model of Chitprasert and Rao was perceptually the most good looking one. See Figure 3.11 for a personal subjective evaluation of your own and check Table 3.4 for the corresponding PSNR performances. A typical viewing distance can be ten times the height of the picture. So the images printed in this thesis book should be viewed at a distance of one meter.

For the sake of comparing a system including a HVS model and a system excluding it, all 150 frames are encoded using the system described above with and without HVS weighting.

DC mean of the HVS weighted sub-blocks was 34.320869 and variance distribution was

244.6946	12.31474	1.031916	0.096935	0.009069	0.001152	0.000140	0.000028
9.338483	0.234692	0.043272	0.004749	0.000665	0.000085	0.000011	0.000003
1.301585	0.036351	0.010473	0.001526	0.000220	0.000039	0.000006	0.000001
0.137803	0.005158	0.001312	0.000286	0.000065	0.000011	0.000002	0.000000
0.010362	0.000490	0.000147	0.000043	0.000012	0.000003	0.000000	0.000000
0.001083	0.000052	0.000014	0.000005	0.000002	0.000001	0.000000	0.000000
0.000104	0.000007	0.000002	0.000001	0.000000	0.000000	0.000000	0.000000
0.000013	0.000001	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000



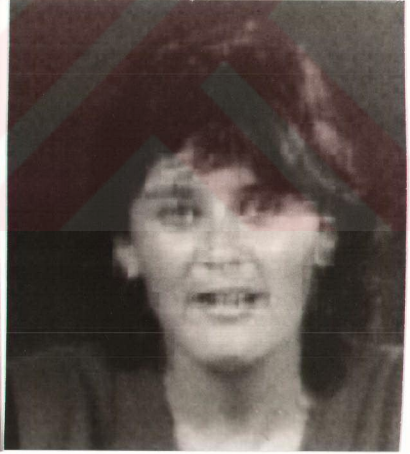
(a)



(b)



(c)



(d)

Figure 3.11. Frame number 0 compressed with block quantization with (a) without HVS (b) with HVS1 (c) with HVS2 (d) with HVS3.

Table 3.5. Comparison of the PSNR performances of the frames displayed in the following pages as an output of the block quantization system including or excluding HVS.

	SQ	HVS1-SQ
Frame #61	38.01	38.55
Frame #75	39.47	39.69
Frame #88	38.14	38.08
Frame #149	38.13	38.28

Bit allocation after HVS weighting yields the following map.

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

Observe in Figure 3.12 that the HVS weighted system outperforms the system lacking HVS in many frames.

On the other hand, actual improvement was in the subjective measures. All 10 viewers who evaluated the two sequences favoured the HVS weighted one, under normal viewing conditions.

See Figures 3.13, 3.14 and 3.15 for a subjective evaluation of your own and Table 3.5 for corresponding PSNR performances.

3.6 Conclusions

It has already been mentioned in Section 2.5 that, particular image frames can quite efficiently be represented by Markov-1 models. Consider a Gaussian Markov-1 source with correlation coefficient ρ . The rate-distortion function for such a source (see [51] example 4.5.2.2) under MSE criterion assuming unity

variance is

$$R(D) = \frac{1}{2} \log_2 \frac{1 - \rho^2}{D}. \quad (3.49)$$

For 1bpp rate, this bound for $\rho = 0.8$ is about 7 dB lower than that of a memoryless Gaussian source whose rate-distortion function appears in Equation (3.28) which can be written as

$$R(D) = \frac{1}{2} \log_2 \frac{1}{D}, \quad (3.50)$$

for a unity variance source.

Ordinary PCM treats any source as memoryless and employs uniform quantization. If the source is passed through a ‘compressor’ prior to quantization, which in effect makes its pdf uniform and optimizes the quantization (assuming an ‘expander’ is employed at the receiver), then the PCM system is said to have optimum companding [44]. The performance of such a system equals that of nonuniform Lloyd-Max quantization of the source and still falls short of the $R(D)$ bound for memoryless sources, due to the facts mentioned in Section 3.3.1. If the source has memory, $R(D)$ bound is even lower. Therefore, block quantization bridges the gap between PCM and rate-distortion bound for correlated sources with known pdf (see [51] figure 5.1.2).

Recall from Equation (3.17) and the discussion following it, that assuming low-distortion (i.e. high-resolution quantization), variance of the reconstruction error incurred when a source of variance σ_x^2 is scalar quantized into b bits is $\sigma_e^2 = k2^{-2b}\sigma_x^2$. When an $N \times N$ block of the source is PCM encoded using identical quantizers for each source sample (i.e. k is same), the average reconstruction error variance is

$$\sigma_{e,PCM}^2 = \frac{1}{N \times N} \sum_{i,j=1}^N k2^{-2R}\sigma_x^2, \quad (3.51)$$

where R , which is the number of bits allocated for each coefficient, is constant and therefore, also equals to the average rate of the system. Since an orthogonal transform (such as DCT) is variance preserving

$$\sigma_x^2 = \frac{1}{N \times N} \sum_{i,j=1}^N \sigma_{y_{ij}}^2, \quad (3.52)$$

if the block of source was first transformed and then PCM encoded, the average reconstruction error variance would be

$$\sigma_{e,PCM}^2 = \frac{1}{N \times N} \sum_{i,j=1}^N k2^{-2R}\sigma_{y_{ij}}^2, \quad (3.53)$$

For block quantization, through optimum bit allocation, total acceptable reconstruction error for the $N \times N$ block is distributed equally to all coefficient quantizers. Then, again assuming identical quantizers, average reconstruction error variance for block quantized transform block is

$$\sigma_e^2 = \frac{1}{N \times N} \sum_{i,j=1}^N k2^{-2b_{ij}} \sigma_{y_{ij}}^2 \quad (3.54)$$

Substituting b_{ij} from Equation (3.27) to this equation one obtains the minimum possible average reconstruction error variance

$$\min\{\sigma_{r,BQ}^2\} = \frac{1}{N \times N} k2^{-2R} \prod_{i,j=1}^N \sigma_{y_{ij}}^2 \quad (3.55)$$

where R is the average rate of the system and $R = B/(N \times N)$

Using Equations 3.53 and 3.55 the maximum gain that block quantization can theoretically attain over PCM can be calculated as

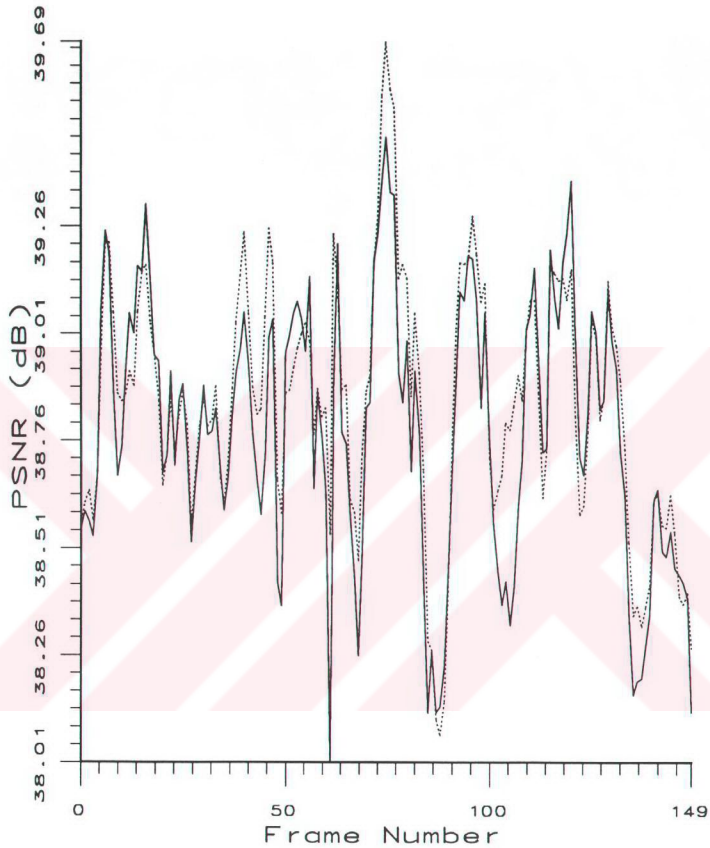
$$\begin{aligned} G_{BQ,max}^{N \times N} &= \frac{\sigma_{e,PCM}^2}{\min\{\sigma_{r,BQ}^2\}} \\ &= \frac{1}{\frac{N \times N}{\prod_{j,l=1}^N \sigma_{y_{jl}}^2} \sum_{i,j=1}^N \sigma_{y_{ij}}^2} \end{aligned} \quad (3.56)$$

The practical gain that block quantization attains can be obtained by multiplying the denominator of Equation (3.56) with a correction factor that accounts for the practical quantizer function. The correction factor is nothing else but the parameter k mentioned in Equation (3.17), at the start of Section 3.3. As an example, $G_{BQ,max}^{N \times N}$ must be multiplied with 1/2.7 for Lloyd-Max quantizer designed for Gaussian pdf which corresponds to a 4.3 dB loss over the maximum attainable gain.

This section also reveals that subjective improvement can be obtained after the objective improvement by block quantization by HVS weighting. However, since HVS weighting requires additional computation there should be a compromise between the added complexity and achieved subjective improvement. Results given in this chapter, reveals that, the HVS incorporating methods tested do not supply enough subjective improvement to be traded with the added complexity. Therefore simpler ways of incorporating HVS should be investigated, which may be a topic of further research.

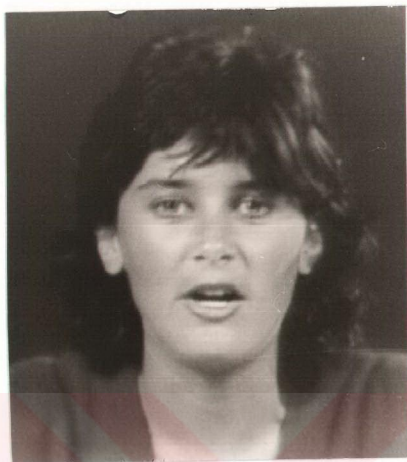
System proposed here has all of its blocks implemented optimally. In the literature there is no such system since probably due to the confusing choices of many references, the authors are misled.

Consider the system proposed in Figure 3.1. This chapter consists of the optimal ways of implementing the blocks of the system assuming that scalar quantizers are used. Therefore, one can conclude that the results obtained are the best possible ones for systems using scalar quantizers. The only improvement that can be done on this system is employing entropy coding or, going even further, optimizing the scalar quantizers used, with respect to a preceding entropy code, which is still a very hot topic of research.

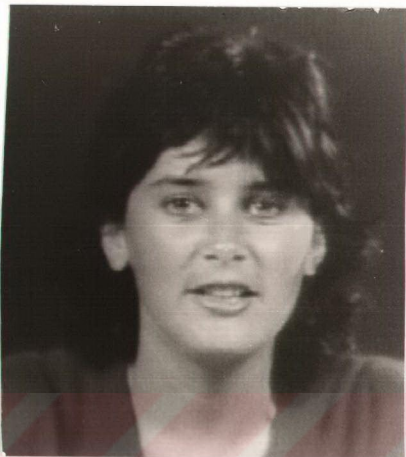


— DCT-SQ DCT-HVS-SQ

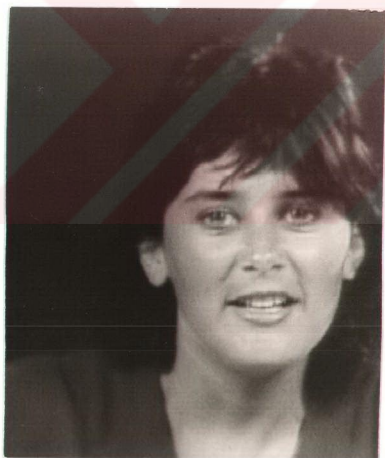
Figure 3.12. PSNR performance of block quantization using optimum scalar quantizers on DCT coefficients and HVS weighted DCT coefficients over the sequence.



(a)



(b)

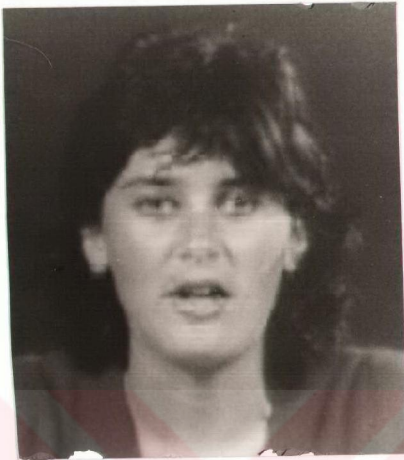


(c)



(d)

Figure 3.13. Original Frames of number (a) 61 (b) 75 (c) 88 (d) 149.



(a)



(b)

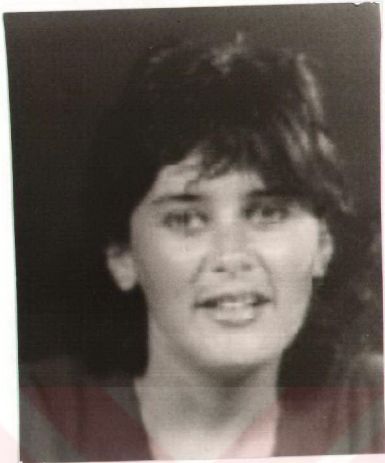


(c)

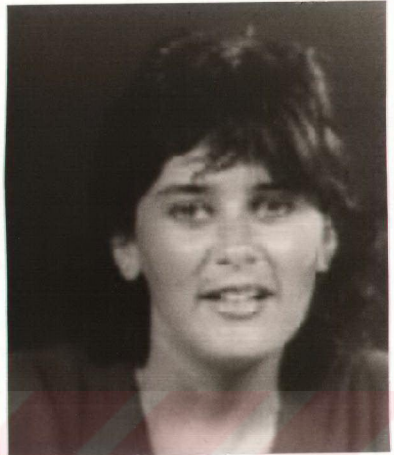


(d)

Figure 3.14. Frame number 61 block quantized (a) without HVS (b) with HVS, and frame number 75 (c) without HVS (d) with HVS.



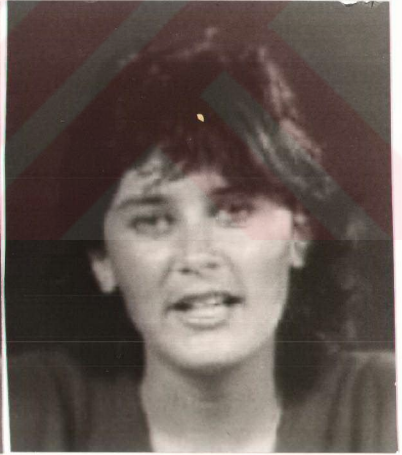
(a)



(b)



(c)



(d)

Figure 3.15. Frame number 88 block quantized (a) without HVS (b) with HVS, and frame number 149 (c) without HVS (d) with HVS.

CHAPTER IV

DISCRETE COSINE TRANSFORM DOMAIN VECTOR QUANTIZATION

A scalar quantizer maps an input value into one of a 'finite' number, say N , of reproduction values. Due to the finite number of reproduction levels, only the index of the closest reproduction level in the collection is enough to be transmitted over the channel. The transmission rate of the system would be $\log_2 N$ bits/sample. With the expense of some distortion, this rate may be significantly smaller than the rate required without quantization. For example, an input sample which is a real number would require an infinite number of bits to be transmitted if it is not quantized (or it would require a quantizer with infinite number of levels for distortionless reconstruction as it was mentioned in Section 3.3.1). Vector Quantization (VQ) generalizes scalar quantization to the simultaneous encoding of a vector of, say k , input samples to one of a finite number, say N , of reproduction vectors. This collection of reproduction vectors is called the 'codebook'. The codebook is usually generated off-line by training it with a number of vectors that represents the statistical properties of the actual vectors to be encoded. Then, transmission rate of the system would be at most $\log_2(N/k)$ bits/sample.

The most frequently cited theoretical rationale for VQ is due to Shannon, who showed that the limit on data compression performance (i.e. the theoretical rate-distortion bound for the source) in the sense of minimizing the average distortion for a particular communication rate, can be approached arbitrarily closely by coding blocks or vectors of source samples rather than coding them individually [3]. However, VQ has both a higher search complexity and also a larger memory requirement than scalar quantization that grow exponentially with the vector dimension. On the other hand, if VQ is applied in the transform domain by taking a subset of transform coefficients for each sub-block as vector inputs, its complexity decreases since vector dimension decreases. Besides, it overcomes

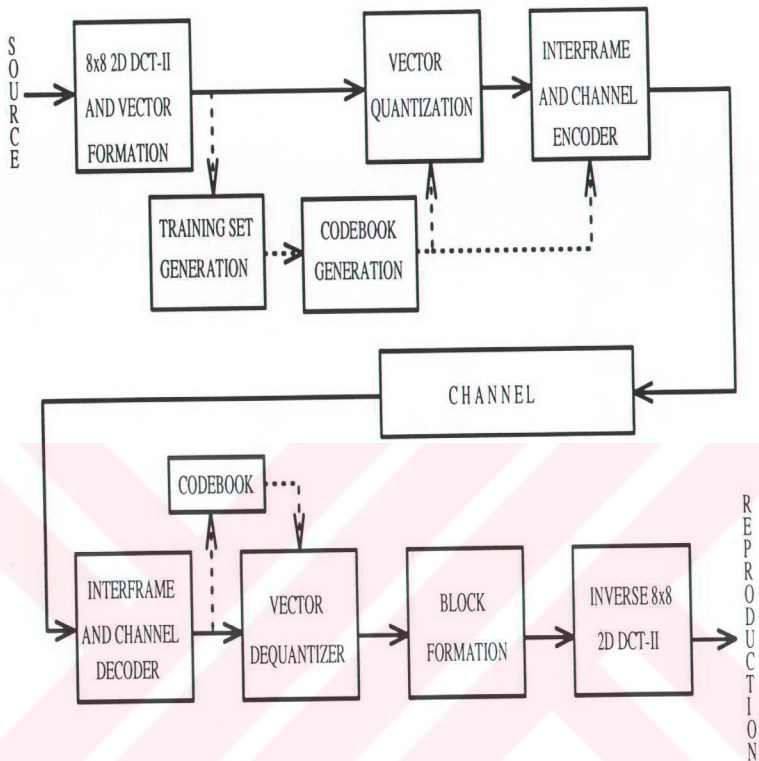


Figure 4.1. A DCT-VQ system for Intra/interframe coding of image sequences

the suboptimality of TC by removing the local redundancies. Figure 4.1 shows a DCT-VQ scheme for image sequence compression.

To construct a vector quantizer, the first thing to do is to decide on the design structure of the codebook. The codebook may be standard (or full-search), tree-structured, product-code, fixed-rate, variable-rate, finite-state, classified etc. Excellent reviews on various VQ design structures can be found in [10], [84], and [85]. Secondly, a satisfactory size of training set of vectors should be chosen and the codebook should be generated such that overall distortion due to the encoding of the training vectors would be minimized. In the statistical literature, training sets are also called learning sets. There are also a large number of codebook

generation techniques.

It is impossible and unreasonable to review and compare experimentally all the existing VQ techniques. So considering the specifications of a desired system one should limit the choice of possible methods of concern. Image sequence coding is usually done for real-time applications, therefore requires fast encoding, decoding. Besides the chosen method should perform well on a variety of images, hence better be adaptive, which also requires fast execution. Among VQ techniques tree-structured VQ (TSVQ) and classified VQ (CVQ) has the most reduced complexity. TSVQ sacrifices much reconstruction quality to construct a tree structured codebook. On the other hand CVQ has relatively higher performance when compared to TSVQ and also it exhibits high edge fidelity if classification is edge oriented. For these reasons, in this thesis work, CVQ has been chosen to be the best candidate for real-time, adaptive, intraframe coding of image sequences and a new classified VQ technique named as 'classifier constrained vector quantizer' is proposed. For experimental comparison, standard, fixed-rate vector quantizers are used with codebooks generated using various techniques. All vector quantizers of concern in this work operate in DCT domain.

For a fixed-rate vector quantizer all the codeword label lengths are same. The choice of fixed-rate vector quantizers for this thesis work, is important, since there is quite an effort to design variable-rate codes at the moment. Most channels demand fixed-rate. If the code to be transmitted is a variable length code and the channel is a synchronous channel demanding fixed rate, then a buffer interconnecting the output of the encoder to the channel input must be instrumented. This brings the issues of buffer constrained optimization of the encoder and buffer control policies to omit buffer 'overflow', and 'underflow'.

Before going into details about the experimented VQ techniques, basic properties of VQ will be explained and superiority of VQ with respect to scalar quantization will be justified.

4.1 Preliminaries

Vector quantization is a multidimensional optimization problem where a codebook of vectors are optimized to represent a source of vectors. To set the terminology, a vector quantizer maps each input vector $\vec{X} \in \mathbb{R}^k$ onto a set of

codewords $Q(\vec{X}) = \vec{Y}_i$, where $\vec{Y}_i \in \mathcal{C} \subset \mathbb{R}^k$ and $i = 1, 2, \dots, N$, so that \mathcal{C} denotes the codebook and N its size. The vectors of the codebook are chosen from a collection of k -dimensional vectors called the ‘output alphabet’. Here output alphabet is assumed to be the set \mathbb{R}^k . Optimality conditions of a vector quantizer for minimum distortion are :

1. Nearest Neighbor Condition : The Euclidean space \mathbb{R}^k must be partitioned into disjoint regions such that

$$\mathcal{R}_i = \{\vec{X} \in \mathbb{R}^k : \|\vec{X} - \vec{Y}_i\| < \|\vec{X} - \vec{Y}_j\|, \forall j \neq i\}, \quad (4.1)$$

2. Centroid Condition : The codevectors must be the centroids of the above regions (note that, regions of such an optimal partitioning is called Voronoi regions or Dirichlet partitions),

$$\vec{Y}_i = E[\vec{X} \mid \vec{X} \in \mathcal{R}_i]. \quad (4.2)$$

If pdf of the input vectors happens to have regions of zero probability, then one or more of the Voronoi region may contain no vectors and so the centroid can not be calculated. Such a degenerate case is called the ‘empty cell problem’.

3. Zero Probability Boundary Condition : The probability of a collection of points being equidistant from at least two of the codevectors is zero :

$$Pr(\vec{X} : \|\vec{X} - \vec{Y}_i\| = \|\vec{X} - \vec{Y}_j\| \text{ for some } i \neq j) = 0 \quad (4.3)$$

Otherwise, tie-breaking would be required, which means that reaching a unique quantizer is not possible.

All these conditions are direct generalization of their scalar case counterpart, derived by Lloyd, so they are also called Lloyd’s conditions. Proofs of these three conditions can be found in [10].

There are two types of optimality. A quantizer is ‘locally’ optimum if every small perturbation of the codevectors does not lead to decrease in the distortion, D . A quantizer is ‘globally’ optimal, if there exists no other codebook that gives a lower level of distortion. If a codebook satisfies the first two conditions it is widely believed that it is only locally optimal, but there is no general theoretical

derivation of this result, in the literature. For the particular case of discrete input distribution, such as a sample distribution produced by a training sequence, a codebook that meets all the three conditions is proved to be locally optimum ([10], section 11.2).

The codebook \mathcal{C} and the partition $\{\mathcal{R}_1, \dots, \mathcal{R}_N\}$ completely describes the quantizer Q . The performance of a vector quantizer Q with a given codebook \mathcal{C} can be measured by the expected (average) distortion

$$\begin{aligned} D(Q) &= E[d_k(\vec{X}, Q(\vec{X}))] \\ &= \sum_{i=1}^N Pr(\vec{X} \in \mathcal{R}_i) E[d_k(\vec{X}, \vec{Y}_i) | \vec{X} \in \mathcal{R}_i] \\ &= \sum_{i=1}^N Pr(\vec{X} \in \mathcal{R}_i) \int_{\vec{X} \in \mathcal{R}_i} d_k(\vec{X}, \vec{Y}_i) p(\vec{X}) d\vec{X}, \end{aligned} \quad (4.4)$$

where $\vec{X} = \{x_1, x_2, \dots, x_k\}$ is a real, k -dimensional random vector on \mathbb{R}^k , described by a joint (multidimensional) probability density $p(\vec{X})$ and $d_k(\vec{A}, \vec{B})$ is a distortion measure in \mathbb{R}^k space. Here the expectation is with respect to the density function $p(\vec{X})$. Since this distribution is usually unknown or not analytically expressible, the above performance measure is practically meaningful if the quantizer is to be used on a stationary and ergodic sequence of vectors so that the time-averaged distortion

$$\frac{1}{M} \sum_{i=1}^M d_k(\vec{X}_i, Q(\vec{X}_i))$$

converges with probability one to $D(Q)$ as $M \rightarrow \infty$. If the sequence is M-ergodic the above time-averaged distortion equals to $D(Q)$. That is ergodicity allows one to substitute sample (or time) averages for ensemble averages. Notice that, this fact gives clues about the size of the training set of vectors [86]! Furthermore, under certain condition on the distortion measures the above discussion can be extended to asymptotically mean stationary sources (from strictly mean stationary sources) and moreover if a source is not ergodic, again under certain conditions it can be treated using ergodic decomposition. These subjects are beyond the scope of this thesis but interested reader can refer to [87].

If $d_k(\vec{A}, \vec{B})$ operates on the elements of its argument vectors, it is called to be a ‘single letter fidelity criterion’ [88]. That is,

$$d_k(\vec{A}, \vec{B}) = \frac{1}{k} \sum_{i=1}^k d(a_i, b_i) \quad (4.5)$$

For a better understanding of the forthcoming discussion, single letter fidelity criteria should be examined more carefully.

Let $\|\vec{X}\|$ denote a seminorm (seminorm, since it has a single argument) on \mathbb{R}^k , that is it satisfies the following conditions :

1. $\|\vec{X}\| \geq 0$,
2. $\|a\vec{X}\| = |a| \cdot \|\vec{X}\|$ for $a \in \mathbb{R}^1$,
3. $\|\vec{X} + \vec{Y}\| \leq \|\vec{X}\| + \|\vec{Y}\|$ (the triangular inequality).

Common examples are the l_v or Hölder norms (sometimes also denoted as L_p norms) defined by

$$\|\vec{X}\|_v \triangleq \left\{ \sum_{i=1}^k |x_i|^v \right\}^{1/v}. \quad (4.6)$$

For $v = 1$ one obtains the mean absolute error metric l_1 which is

$$\|\vec{X}\|_1 = \sum_{i=1}^k |x_i|, \quad (4.7)$$

and for $v = 2$ one obtains the mean-square error metric (the Euclidean norm), l_2 which is

$$\|\vec{X}\|_2 = \left\{ \sum_{i=1}^k |x_i|^2 \right\}^{1/2}. \quad (4.8)$$

All Hölder norms have variants called as the r^{th} power or r^{th} -law of them and can be expressed as follows

$$\|\vec{X}\|_v^r \triangleq \left\{ \sum_{i=1}^k |x_i|^v \right\}^{r/v}. \quad (4.9)$$

Note that, a chosen distortion measure is more useful if it is a distance or metric, and hence satisfies the triangular inequality. Some r^{th} power of Hölder norms do not have this property, but for instance, 2^{nd} power or 2^{nd} -law of l_2 metric, which is the well-known squared-error measure (the Euclidean distance)

$$\|\vec{X}\|_2^2 = \sum_{i=1}^k |x_i|^2, \quad (4.10)$$

is a metric and it is very commonly used in practice due to its mathematical convenience. Notice that, for a single argument (seminorm), it actually gives the

vector variance of its argument vector. Another common form of measure is the weighted squared-error measure

$$\|\vec{X}\|_{2,weighted}^2 = \sum_{i=1}^k w_i |x_i|^2, \quad (4.11)$$

Since l_2 norms can also be written in inner product form, $\|\vec{X}\|_2 = \{\vec{X}^T \vec{X}\}^{1/2}$ (similarly $\|\vec{X}\|_2^2 = \{\vec{X}^T \vec{X}\}$), weighted measures can further be generalized as

$$\begin{aligned} \|\vec{X}\|_{2,Bweighted}^2 &= \{\vec{X}^T [B] \vec{X}\} \\ &= \sum_{i=1}^k \sum_{j=1}^k [B]_{ij} x_i x_j. \end{aligned} \quad (4.12)$$

where $[B]$ is a $k \times k$, positive definite, symmetric matrix. Observe that if $[B]$ is also a diagonal matrix, then Equation (4.12) reduces to Equation (4.11) and if it is the identity matrix, Equation (4.12) further reduces to Equation (4.10).

Turning our attention back to $d_k(\vec{X}, \vec{Y})$, if it has the form

$$d_k(\vec{X}, \vec{Y}) = L(\vec{X} - \vec{Y}) \quad (4.13)$$

it is called as a ‘difference distortion measure’ and then, any of the above norms are adoptable since they all have single arguments, i.e. they turn to be full norms for difference distortion measures. As an example if the norm of Equation (4.9) is chosen

$$\begin{aligned} d_k(\vec{X}, \vec{Y}) &= \|\vec{X} - \vec{Y}\|_v^r \\ &= \left\{ \sum_{i=1}^k |x_i - y_i|^v \right\}^{r/v}, \end{aligned} \quad (4.14)$$

or if the norm of Equation (4.12) is chosen, then

$$\begin{aligned} d_k(\vec{X}, \vec{Y}) &= \|\vec{X} - \vec{Y}\|_{2,Bweighted}^2 \\ &= [\vec{X} - \vec{Y}]^T [B] [\vec{X} - \vec{Y}] \\ &= \sum_{i=1}^k \sum_{j=1}^k [B]_{ij} (x_i - y_i)(x_j - y_j). \end{aligned} \quad (4.15)$$

There also exists distortion measures which does not have the form of Equation (4.13) but depends on \vec{X} and \vec{Y} in a more complex manner. An important example is Itakura-Saito distortion measure [89], which is very useful in LPC speech coding systems but usually unacceptably complex for image coding. It has the form

$$d_k(\vec{X}, \vec{Y}) = [\vec{X} - \vec{Y}]^T [R(\vec{X})][\vec{X} - \vec{Y}] \quad (4.16)$$

where for each \vec{X} , $[R(\vec{X})]$ is a $k \times k$, positive definite, symmetric matrix. For some of the other distortion measures, not mentioned here, the interested reader may refer to [90].

In VQ codebook generation and performance evaluation for image coding the most commonly used measure of distortion is that of Equation (4.10) i.e., the Euclidean distance. However, others can also be easily used. For instance, some researchers strongly favour l_1 metric since it requires no multiplication [91] and furthermore there is experimental evidence that for some coding methods, if the design of the codec is based on l_1 metric rather than l_2 , subjectively better results are obtained (see [92] for the case of wavelets). But the possibility that usage of l_1 metric produces subjectively better results for any coding scheme is still a subject of debate. As a result, while not subjectively meaningful in many cases, generalization of l_2 metric, permitting input-dependent weightings have proved useful although slightly complicated. Hence, throughout this thesis work l_2 metric will be used in designing the VQ systems of concern. Other metrics given above are useful for calculating performance bounds for VQ.

Once codebooks fitting above optimality conditions are generated using one of the above metrics, the source can be quantized to nearest codevector, nearest in the measure of one the above metrics. If all codevectors are tested to find the nearest, this type of quantization is called 'full search' and it is the method used throughout this thesis work. Assuming the metric used for testing is that of Equation (4.10) i.e., the Euclidean distance, the number of multiply-adds required to quantize a single k -tuple with a codebook of size N is Nk . Defining $R \triangleq (\log_2 N)/k$ as the bits per dimension (sample), this computational cost becomes $k2^{Rk}$, meaning that the 'computational cost' grows exponentially with the dimension and the bits per dimension. Another important part of the quantization cost is the 'memory' or 'storage' cost, i.e. how much memory is needed to store the codevectors? Assuming one storage location per vector dimension, the cost is again $k2^{Rk}$.

4.2 Superiority of Vector Quantization

The very appealing properties of VQ had been set forth much before first practical vector quantizers began to appear and initiated a great deal of research.

The theoretical advantages of VQ are mainly two-fold, those depicted by rate-distortion theory and those by high-resolution quantization theory. Especially, high-resolution theory results can be used to reveal the practical advantages of VQ over SQ.

4.2.1 Theoretical Advantages of Vector Quantization

Assuming no constraint has been imposed on a finite dimensional vector quantizer except for the resolution (i.e. finite codebook size), it has mainly two attributes :

1. Vector dimension k .
2. Resolution or codebook size N .

Most of the results in this section are derived under two asymptotic conditions over these attributes :

1. Vector size k is large and N is arbitrary. Rate-distortion theory provides results on vector quantizer performance under this condition.
2. Codebook size N is large (i.e. small distortion) and k is arbitrary. This is the subject of high-resolution or asymptotic quantization theory. The essence of high-resolution quantization theory is to assume that there are so many output points that the probability density of the input is approximately constant across any particular input vector.

4.2.1.1 Rate-Distortion Theory

Since the unique purpose of data compression is to minimize the transmission bit-rate for a desired level of distortion or vice versa, it is important in a particular situation to know the theoretical lower bound on the bit-rate for any quantizer. Rate-distortion theory, a branch of information theory, deals with obtaining such lower bounds of bit-rate or distortion, without requiring the design of actual quantizers. That is, given the source sample it does not bother to calculate explicitly what the reproduction must be. It just considers the basic transmission system of Figure 4.2, and any kind of source. The source may be in continuous-time or in discrete-time, and its amplitude may be, again, continuous

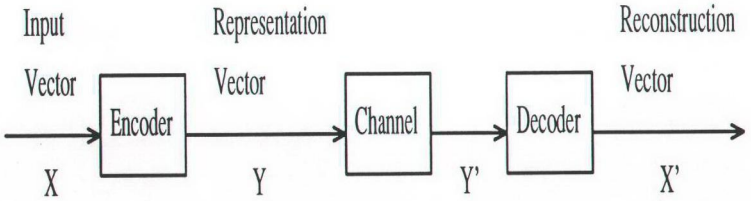


Figure 4.2. A basic information transmission system.

(e.g. transform coefficients) or discrete (e.g. pixels of an image). For a given distortion metric, one can compute either $R(D)$, the ‘rate-distortion function’, defined as the minimum achievable rate (per dimension) for a given distortion D , or its inverse $D(R)$, the ‘distortion-rate function’, defined as the minimum achievable distortion for a given rate R .

For our specific purpose, consider discrete-time continuous amplitude sources. Since the given input sample may have any real value, for lossless reconstruction, the output should also have continuous amplitude, meaning the representation should have infinite number of bits/sample and that the channel should handle infinite rate (recall Section 3.3.1). Therefore, for such sources $R(0) = \infty$. For finite ‘capacity’ channels, it is inevitable that $D > 0$ and $R(D)$ gives the minimum capacity that the channel should have to carry enough information for a reconstruction having distortion D . Note that, by doing this the output becomes no more continuous. That is, output space would be partitioned into finite number of regions, since finite channel capacity implies finite number of representations. Actually, the minimum number of such partitions is $2^{R(D)}$.

For an encoder to achieve that $R(D)$, $D > 0$, bound, it must completely decorrelate the input, i.e. excise all the redundancy in the source. This can only be done by encoding the whole source at one time, which is practically impossible. The encoder would have to wait until all the source arrives and so, the encoding delay would be infinite. Therefore considering the practical case of coding k -tuples of source, the k^{th} order rate-distortion function, $R_k(D)$, is sought. $R_k(D)$ approaches $R(D)$ as $k \rightarrow \infty$. At this point, to understand the concept of rate-distortion theory, one should immediately contemplate the case of the case of having zero distortion. A source of k -tuples of continuous

amplitude samples would require an infinite size codebook of k -tuples, each having continuous amplitude random variables as elements i.e. codevectors are from an infinite output alphabet, namely \mathbb{R}^k . All that rate-distortion theory says is, to achieve a distortion $D > 0$, the minimum number of partitions in the output space (alphabet), should be $2^{R_k(D)}$, if k -tuples are coded, which is greater than the minimum number of partitions $2^{R(D)}$, if ∞ -tuples are coded. In other words, we are just demanding more channel capacity to limit the encoding delay, but still keep the distortion at the level D .

The optimality of VQ in the asymptotic sense of large vector dimension, is proved by the fact

$$\lim_{k \rightarrow \infty} R_k(D) = R(D) \quad (4.17)$$

which was first shown by Shannon [3]. But still, the ‘operational’ (practical) rate-distortion function of any vector quantizer available falls short of the bound $R_n(D)$. The reason is given below.

Assume $D > 0$, for obtaining $R_k(D)$, rate-distortion theory assumes that one can choose $2^{R_k(D)}$ number of reconstruction vectors from the infinite set of \mathbb{R}^k . But then there exists infinite number of choices. Limiting the number of choices to M , one can obtain the ‘constrained-alphabet’ rate-distortion function $R_{M,k}(D)$ [93], [94], [95] (the reader should discriminate the difference between constrained-resolution and constrained-alphabet!). If one lets the output alphabet (collection of possible choices of codevectors) be the set \mathbb{R}^k , meaning $M \rightarrow \infty$, $R_{M,k}(D) \rightarrow R_k(D)$. Going even further, also impose a partitioning over the source space consisting of L regions, where $L \geq M$. Then the relation between resulting rate-distortion functions is $R_{L,M,k}(D) \geq R_{M,k}(D) \geq R_k(D) \geq R(D)$. Because, when a partitioning of the source space is imposed, one sacrifices from the freedom of treating each input vector separately and has to map all vectors in a region to the same codevector, thus further increasing the distortion. Equality with $R_{M,k}(D)$ is met when $L \rightarrow \infty$. Although $R_{M,k}(D)$ can be described theoretically, it is impossible to calculate. But in the case of $R_{L,M,k}(D)$, where source and reproduction alphabets are finite, the elegant Blahut algorithm [96] can be used for computing. In the development of $R_{L,M,k}(D)$ it is permitted that all the source vectors in a partition cell has the same and constant probability of being mapped to any of the output vectors. On the other hand, for all present practical

VQ schemes, $L = M$ and all input vectors in a partition cell is mapped, with probability 1, to the same codevector which is also ‘within’ that cell. This preclusion of inter-cellular mappings restricts the ‘operational’ rate-distortion function of all existing VQ schemes to be bounded by $R_{L,M,k}(D)$. Hence, all upto date VQ designs falls short of the $R_k(D)$ bound. Various choice of L and M bridges the gap between operational rate-distortion functions and $R_k(D)$, and from the ongoing discussion it is, now, clearer that the VQ paradigm, can only ‘asymptotically’ achieve the $R(D)$ bound for a given source, while scalar quantizers can, in any way, not. Still, even the best performing quantizer ever, can not carry us to the promised land by Shannon.

4.2.1.2 High-Resolution Quantization Theory

High resolution quantization theory provides results (upper and primarily lower bounds) on the minimum distortion attainable by a k -dimensional quantizer for the case of large but finite (or fixed, or constrained) number of reconstruction vectors, and for the case of constrained output entropy, again, for large but unconstrained resolution.

Adopting the terminology used in Section 4.1, the output entropy is

$$H = - \sum_{i=1}^N p_i(\vec{X}) \log_2 p_i(\vec{X}) \quad p_i(\vec{X}) = Pr\{\vec{X} \in \mathcal{R}_i\} \quad (4.18)$$

Such performance bounds are often tighter [97] than the general lower bounds provided by the rate-distortion theory (Section 4.2.1.1). Furthermore, it is often the case experimentally that these bounds are fairly accurate when the number of allowed reproduction vectors is only moderate.

The initial fundamental assumption in almost all studies of asymptotic quantization is that the probability density $p(\vec{X})$ is sufficiently ‘smooth’ to ensure that $p(\vec{X})$ is effectively constant over small bounded sets (partition regions for high resolution). In particular, for N large enough, $p_i(\vec{X}) = p(\vec{X} | \vec{X} \in \mathcal{R}_i) \cong b_i$. Then,

$$\begin{aligned} P_i &\triangleq Pr(\vec{X} \in \mathcal{R}_i) \\ &= \int_{\mathcal{R}_i} p(\vec{X}) d\vec{X} \\ &\cong b_i \int_{\mathcal{R}_i} d\vec{X} \\ &= b_i V(\mathcal{R}_i), \end{aligned} \quad (4.19)$$

or equivalently,

$$p_i(\vec{X}) \cong \frac{P_i}{V(\mathcal{R}_i)}, \quad (4.20)$$

where given a set $G \subset \mathbb{R}^k$, its volume $V(G)$ is defined by

$$V(G) = \int_G d\vec{X}. \quad (4.21)$$

$V(G)$ is the Lebesgue measure of G in \mathbb{R}^k , and it can be defined only if G is a Borel set. In particular the volume of the unit sphere (a Borel set) in \mathbb{R}^k is defined by

$$V_k = V(\{\vec{U} : \|\vec{U}\| \leq 1\}). \quad (4.22)$$

For the 2^{nd} power of Hölder norms, I_v^2 it can be expressed as [98]

$$V_k(v) = \frac{2^k [\Gamma(1/v)]^k}{k \Gamma(k/v) v^{k-1}}, \quad (4.23)$$

where $\Gamma(\cdot)$ is the gamma function.

In 1948 Bennett [99] modelled the one-dimensional quantizer (scalar quantizer) as a zero memory monotonic increasing nonlinearity $C(\cdot)$ (called the compressor, recall Section 3.6) followed by a uniform N -level scalar quantizer in turn followed by the inverse nonlinearity $C^{-1}(\cdot)$ (called the expander). Based on some implicit mild regulatory conditions on $p(\vec{X})$ in the case of large N , he derived the integral below as an approximation to the minimum squared-error distortion attainable by such a scalar quantizer :

$$D \cong \frac{(L_2 - L_1)^2}{12N^2} \int_{L_1}^{L_2} \frac{p(\vec{X})}{\lambda^2(\vec{X})} d\vec{X}. \quad (4.24)$$

where $\lambda(\vec{X}) = dC(\vec{X})/d\vec{X}$, (L_1, L_2) is the interval that consists of the $N-2$ finite regions $\mathcal{R}_i, i = 2, \dots, N-1$ of the scalar quantizer. He also assumed that L_1 and L_2 are appropriately chosen so that the contribution to the distortion due to the tail or ‘overload’ regions can be neglected (recall Equation (3.5) and Figure 3.1).

The necessary and sufficient conditions for Bennett’s integral to give the correct distortion was derived many years after Bennett by Bucklew [100]. Bucklew also gave an example for which Bennett’s integral does not give correct asymptotic distortion of a quantizer. However, Bennett’s integral was a convenient analytical tool for optimization studies of k -dimensional quantizers and having realized the virtues of high resolution quantization theory many authors began to obtain approximate solutions for the distortion in cases of large N and constrained entropy.

Let's denote the minimum distortion for quantizing k -dimensional vectors into a large number of quantization levels N (high-resolution) under the metric r^{th} power of Hölder norms l_v (Equation (4.9)) as $D(N; k, r, l_v)$. Bennett's integral was for $D(N; 1, 2, l_2)$. Panter and Dite [40] simplified Bennett's integral and obtained $D(N; k, 2, l_2) = C_1 N^{-2/k}$, where C_1 was a constant depending on $p(\vec{X})$ and k (recall that for $k = 1$ this result was also used in Section 3.3). For the scalar case, Algazi showed that [101]

$$D(N; 1, r, l_2) = \frac{1}{r+1} (2N)^{-r} \|p_1(\tilde{x})\|_{1/(1+r)} \quad (4.25)$$

where $p_1(\tilde{x})$ is the one-dimensional pdf of the random variable \tilde{x} . For the case of constraint entropy, Gish and Pierce [102] solved the one dimensional problem as

$$D(H; 1, r, l_2) = \frac{1}{r+1} 2^{-r} 2^{-r[H-H(p_1(\tilde{x}))]} \quad (4.26)$$

where H is the fixed output entropy constraint and $H(p_1(\tilde{x}))$ is the differential entropy, $H(p_1(\tilde{x})) = -\int p_1(\tilde{x}) \log_2 p_1(\tilde{x}) d\tilde{x}$. Later, Gray and Gray [103] obtained a simple proof of the Gish and Pierce's result via Hölder's and Jensen's inequalities instead of using variational techniques as Gish and Pierce did. Gray and Gray also provided a simple proof for Algazi's result.

Then generalizations of the above results to vector quantization ($k > 1$) followed. Zador, in his historical paper [104] obtained

$$D(N; k, r, l_2) = A(k, r) N^{-r/k} \|p(\vec{X})\|_{k/(k+r)} \quad (4.27)$$

and

$$D(H; k, r, l_2) = B(k, r) e^{-(r/k)[H-H(p(\vec{X}))]} \quad (4.28)$$

where $H(p(\vec{X}))$ is the differential entropy of the random vector \vec{X} and $\|p(\vec{X})\|_v$ is as below with a k -dimensional integration

$$\|p(\vec{X})\|_v = \left[\int [p(\vec{X})]^v \right]^{1/v} \quad (4.29)$$

Zador did not derived $A(k, r)$ and $B(k, r)$ explicitly, rather gave lower and upper bounds for them. Later, it was noticed by Bucklew [105], who also developed upper bounds for the distortion with l_v^r metric, that Zador's upper bound actually converges his lower bound for $k \rightarrow \infty$. The most striking feature of Zador's expression in Equation (4.27) was that $A(k, r)$ had no dependency on the pdf of

the random vector. Gersho, unlike Zador, came up with a valid explanation of the situation for multidimensional case with l_2^r , and obtained coinciding result with Zador's [106]. Since his conjectures will be used explicitly in the next chapter, details of the Gersho's work should be given.

Recall from Equation (4.1) that an optimal vector quantizer must have Dirichlet partition. In general, each bounded Dirichlet region is a 'polytope' and is convex. A convex polytope is a region of k -dimensional space bounded by $(k - 1)$ -dimensional hyperplanes so that any point lying on a line segment joining points in the bounding hyperplanes is within the region. On the other hand the unbounded regions or 'overload' regions makes sufficiently small contribution to the distortion (N is large). If a uniform probability density is placed on the space contained by the polytopes (as it is, due to high resolution assumption), then the optimum reconstruction k -tuples are the centroids of the polytopes (recall Equation (4.2)).

A convex polytope T is said to generate a 'tessellation', if there exists a partition of \mathbb{R}^k , whose regions are all congruent to T . For instance, on the two dimensional plane \mathbb{R}^2 , all triangles, quadrilaterals, hexagons generate tessellations. Keeping above facts in mind, here are the Gersho's conjectures :

1. For large N optimal partitioning of \mathbb{R}^k is essentially a tessellation.
2. Let T_k be the class of admissible polytopes in \mathbb{R}^k such that a convex polytope $T \in \mathbb{R}^k$ is an element of T_k if it generates a Dirichlet partition. Now define the 'coefficient of quantization' as

$$C(k, r) \triangleq \frac{1}{k} \inf_{T \in T_k} \frac{\int_T \|\vec{X} - \vec{Y}\|_2^r d\vec{X}}{[V(T)](1 + r/k)}, \quad (4.30)$$

where the argument of infimum is nothing but the normalized 'moment of inertia' of the k -dimensional volume T . This coefficient of quantization is determined by the optimal cell shape (a polytope such that $T \in T_k$), and it is independent of $p(\vec{X})$.

After these conjectures, Gersho proves that $C(k, r)$ is lower bounded as

$$C(k, r) \geq \frac{1}{k+r} V_k^{-r/k}, \quad (4.31)$$

using the fact that the sphere in any dimension k has the lowest moment of inertia.

He, also obtains a general form for the Bennett's integral :

$$D(N; k, r, l_2) = N^{-r/k} C(k, r) \int p(\vec{X}) [\lambda(\vec{X})]^{-r/k} d\vec{X}, \quad (4.32)$$

where $\lambda(\vec{X})$ is the asymptotic output point density function (check the counterpart of it in Bennett's integral in Equation (4.24)). The region of integration in Equation (4.32), is the union of all bounded regions (i.e. unbounded Drichlet regions, polytopes is excluded) due to the negligible overload distortion. Notice that, each region \mathcal{R}_i makes an equal contribution to the distortion.

Using Hölder's inequality on Equation (4.32) one can obtain [106] :

$$\int p(\vec{X}) [\lambda(\vec{X})]^{-r/k} d\vec{X} \geq \| p(\vec{X}) \|_{k/(k+r)}, \quad (4.33)$$

with the equality attained when λ is proportional to $p(\vec{x})^{k/(k+r)}$. Since this is already the property of a k -dimensional compressor, equality holds, and an identical expression to Zador's (Equation (4.27)) is obtained

$$D(N; k, r, l_2) = C(k, r) N^{-r/k} \| p(\vec{X}) \|_{k/(k+r)}. \quad (4.34)$$

Using the lower bound for $C(k, r)$ given at Equation (4.31) in Equation (4.34), a lower bound for $D(N; k, r, l_2)$ can be obtained, which is known as the Gersho lower bound or the sphere bound.

Yamada et al. [97] extended Gersho's results for l_2^r to l_v^r . In their derivations they used spherical regions with 'effective radius' $R(\mathcal{R}_i)$

$$R(\mathcal{R}_i) = [V(\mathcal{R}_i)/V_k]^{1/k}, \quad (4.35)$$

and obtained a lower bound for the distortion (not an approximate solution like Equation (4.32)). They then proceed to show how these lower bounds for various choices of k and r , are infact tighter than the Shannon lower bound (Equation (3.28)).

As a last note, consider $\| p(\vec{X}) \|_v$ as in Equation (4.29), where $p(\vec{X})$ is the k^{th} order joint probability density function of the k -dimensional vectors from a stationary, zero mean, random source. Since, the source is zero mean its covariance function will be same as its correlation function which is the inverse Fourier transform of spectral density function. Then, from the definition of spectral density function one can obtain

$$\| p(\vec{X}) \|_{k/(k+r)} = (2\pi)^{r/2} \frac{(k+r)^2}{2k} |C|^{r/2k} \quad (4.36)$$

where $|C|$ is the determinant of covariance matrix $[C]$ of the source (see [104] for the proof). If the source samples are also independent and identically distributed, then

$$|C|^{1/k} = \bar{\sigma}_g^2 = \left[\prod_{i=1}^k \sigma_i^2 \right]^{1/k} \quad (4.37)$$

where $\bar{\sigma}_g^2$ is the geometric mean of the variances of elements of the vector.

4.2.2 Practical Advantages of Vector Quantization

For VQ there exists three properties of vector elements which, when utilized appropriately, result in optimal performance [52]:

1. 'Dependency' : It can be classified into two : 'Linear dependency' and 'nonlinear dependency'. Linear dependency is the correlation between vector elements. When elements of a vector are decorrelated (say by a unitary transform) they become no more linearly dependent, but still they can have a statistical dependency. Nonlinear dependency is whatever dependency remains after linear dependency is removed.
2. 'Dimensionality' : As vector dimension increases VQ can adopt any suitable cell type to fill the k -dimensional space effectively.
3. 'Probability density function shape' : In the case of convex polytopes in Section 4.2.1.2 the cells do not only have the same shape but also the cell sizes were also same. Therefore, the cell spacing was uniform. Uniform cell spacing (identical cell sizes) is reasonable for uniform pdf assumption of high resolution quantization theory. In case of, nonuniform pdf, the cell sizes can vary to increase the performance of VQ.

To reveal why and by how much a vector quantizer outperforms a scalar quantizers, the above three properties can be represented in three analytical factors which are called the vector quantizer advantages.

Using the results of high-resolution quantization theory examined in Section 4.2.1.2, Equation (4.34) to be specific, a vector quantizer advantage factor over scalar quantizer can be defined as follows

$$\begin{aligned} \Delta(k, r) &\triangleq \frac{D(N; 1, r)}{D(N; k, r)} \\ &= \frac{C(1, r) \|p(\hat{x})\|_{1/(1+r)}}{C(k, r) \|p(\vec{X})\|_{k/(k+r)}}, \end{aligned} \quad (4.38)$$

where \tilde{x} is a random variable and \vec{X} is a random vector belonging to the same source and governed by the corresponding marginal and joint density function $p(\tilde{x})$ and $p(\vec{X})$, respectively. Now define a dummy density function which would result if vector elements were independent :

$$p^*(\vec{X}) \triangleq \prod_{i=1}^k p(\tilde{x}). \quad (4.39)$$

With this definition, Equation (4.38) can be re-written as follows

$$\Delta(k, r) = F(k, r)S(k, r)M(k, r). \quad (4.40)$$

where $F(k, r)$ is the ‘space filling advantage’, $S(k, r)$ is the ‘shape advantage’, and $M(k, r)$ is the ‘memory advantage’ factors corresponding to dimensionality, probability density function shape and dependency properties listed above, respectively.

4.2.2.1 Space-filling Advantage

Space-filling advantage is defined by as follows

$$F(k, r) \triangleq \frac{C(1, r)}{C(k, r)}. \quad (4.41)$$

Observe that, $F(k, r)$ depends only on the coefficient of quantization (Equation (4.30)) and hence due to Gersho’s second conjecture in Section 4.2.1.2, it depends only on the efficiency with which the polytopes fill the space. Since $C(k, r)$ is a strictly decreasing function of k , it is obvious that $F(k, r) \geq 1$ for $k \geq 1$. This factor guarantees that VQ will always achieve superior performance over scalar quantization and so it has long been recognized as the fundamental advantage of VQ over scalar quantization by many researchers. For $k = 1, 2$ and all r , and for $k = 3$ and $r = 2$, $C(k, r)$ can be calculated since admissible polytopes for such cases are known [106], [98]. For $k > 3$, fixing r , lower bounds on $C(k, r)$ such as Gersho’s lower bound can be used to find upper bounds for $F(k, r)$. Conway and Sloane have also conjectured a tighter lower bound [107]. For a table of upper bounds on $F(k, r)$ refer [98]

4.2.2.2 Shape Advantage

Shape advantage can be written as follows

$$S(k, r) \triangleq \frac{\|p(\tilde{x})\|_{1/(1+r)}}{\|p^*(\vec{X})\|_{k/(k+r)}}. \quad (4.42)$$

Adapting Equation (4.6) to this continuous argument case of Hölder norms Equation (4.42) can be re-written as follows

$$S(k, r) = \frac{\int p(\tilde{x})^{1/(1+r)} r+1 d\tilde{x}}{\int p(\tilde{x})^{k/(k+r)} d\tilde{x}}. \quad (4.43)$$

Hence $S(k, r)$ depends only on the shape of $p(\tilde{x})$ using Hölder's inequality one can show that $S(k, r) \geq 1$ for all $k \geq 1$ and $r \geq 0$ [98].

4.2.2.3 Memory Advantage

Memory advantage is clearly

$$M(k, r) \triangleq \frac{\|p^*(\vec{X})\|_{k/(k+r)}}{\|p(\vec{X})\|_{k/(k+r)}}. \quad (4.44)$$

For input k -tuples of independent identically distributed (i.i.d.) elements, by definition $p^*(\vec{X}) = p(\vec{X})$ and so $M(k, r) = 1$, meaning all the advantage of VQ is due to space-filling and shape factor. For other cases one should have the expression of k -dimensional joint probability function of the source and has to compute it over k -dimensional integrals, which is a tedious task (refer [98] and [52] example 2 for such cases).

4.3 Codebook Generation Techniques

No matter what type of a VQ structure is chosen, a codebook is required. The key to efficient VQ data compression is a good codebook. In this section most recent codebook generation techniques as well as the classical method of generalized Lloyd algorithm will be reviewed.

All methods included in this section are for minimizing the distortion introduced due to VQ quantization with finite dimension codebooks generated using finite number of training vectors. The only constraint imposed on the methods below is the resolution of the codebook to be generated, in other words its size. Other type of constraints are beyond the scope of this section.

4.3.1 Generalized Lloyd Algorithm

Lloyd method I for scalar quantizer design (proposed in 1957 as an unpublished technical note, but later edited in [41]) was in use at statistical literature

since 1965. It was called the '*k*-means algorithm' (perhaps that's why the script for vector dimension is always chosen as "*k*" in VQ) and used for clustering and pattern recognition problems [108]. It was first in [90], that it appeared in compression literature and provided a practical codebook design algorithm for VQ that would yield codebooks meeting all three of the Lloyd's conditions. Since then, some authors called the algorithm as LBG algorithm after the initials of Linde, Buzo and Gray, others used the name generalized Lloyd algorithm (GLA). GLA has pioneered the overwhelming research work on VQ and taken part in most of the work on VQ either for comparison purposes or as a tool for other algorithms. It requires an initial codebook to start with and iteratively optimizes the encoder for the decoder and decoder for the encoder using the codebook and a given distortion measure.

The outline of the algorithm is as follows :

Step 0 Start with :

- A set of training vectors $\{\vec{X}_1, \dots, \vec{X}_M\} = \mathcal{X} \subset \mathbb{R}^k$,
- An initial codebook $\{\vec{Y}_1^{(0)}, \dots, \vec{Y}_N^{(0)}\} = \mathcal{C}^{(0)} \subset \mathbb{R}^k$,
- A termination constant ϵ (typically 0.001),
- A distortion measure $d_k(\vec{A}, \vec{B})$,

Initialize :

- The iteration counter $t=1$,
- The average distortion $D^{(0)}$ of the quantizer to ∞ .

Step 1 Encode all the training set vectors by mapping them to their nearest codevector i.e. map a given training set vector \vec{X} to $\vec{Y}_i^{(t)}$ if $d_k(\vec{X}, \vec{Y}_i^{(t)}) \leq d_k(\vec{X}, \vec{Y}_j^{(t)}) \forall j \neq i$ and $j = 1, \dots, N$.

This mapping will result in a minimum distortion partition and impose decision region around each codevector. Also check if any non-zero probability boundary condition occurred and if it did, apply a suitable tie-breaking rule.

Step 2 Calculate the resulting average distortion

$$D^{(t)} = E\{\min_{\vec{Y} \in \mathcal{C}^{(t)}} d_k(\vec{X}, \vec{Y})\}$$

if $(D^{(t-1)} - D^{(t)})/D^{(t-1)} \leq \epsilon$ QUIT, else CONTINUE with Step3.

Step 3 Check if there exists any empty cells. If so, make an alternate bit assignment. Update the codebook by replacing each codevector $\vec{Y}_i^{(t)}$

with a new codevector $\vec{Y}_i^{(t+1)}$ that minimizes the average distortion in its decision region. If the distortion measure is the Euclidean distance this new codevector would be the centroid of the training set vectors enclosed by the region. SET $t = t + 1$ and GOTO Step 1.

If at any iteration the optimum partitioning generated in Step 1 has a cell with zero probability of having a training set vector, empty cell problem occurs, since the new centroid can not be calculated. A variety of heuristic solutions exist for handling the problem. Aside from assigning an arbitrary vector as centroid, one method may be to split the centroid of the cell with highest number of training set vectors into two and delete the empty cell. Another method is to split the centroid of the cell with highest partial average distortion.

One can also check, before terminating, if the zero-probability boundary condition is satisfied. That is no training set vector is left to be equidistant from two (or more) codevectors. The situation can be detected in Step 1 of the algorithm and can be overcome using again heuristic methods, such as assigning the vector to the tied cell with smallest partial distortion or to the tied cell with least number of elements. If this condition is not satisfied and the algorithm terminates, then the resulting codebook can, in principle, be improved by breaking the tie in any different manner and running the algorithm for one more iteration. Although the improvement due to this additional step seems practically negligible, this step becomes very important, if the size of training set is small. Notice that, GLA does not converge to a unique codebook since ties can be broken in any fashion.

The general convergence analysis of GLA was carried out only in [109], with extensive mathematics. Their result was that, given a distortion function with the property that $d : \mathbb{R}^k \times \mathbb{R}^k \rightarrow [0, \infty)$ and three more additional properties which are important only technically for the proofs and has no practical meaning, if GLA is to be trained with M samples, having an empirical distribution function of F_M , from an asymptotically mean stationary ergodic source described by distribution function F with properties ;

1. F contains no singular-continuous part,
2. $\int d(\vec{X}, \vec{Y}) dF(\vec{X}) < \infty$ for each $\vec{Y} \in \mathbb{R}^k$,

then GLA converges with probability one. The empirical distribution function of

a M length sequence of random vectors, just places the probability $1/M$ on each of the elements of the sequence.

GLA is a descent algorithm i.e through its iterations the distortion function decreases monotonically. However, considering codebooks of more than one codevector for which the distortion function is not convex and contains multiple local minima, GLA may well be trapped in a poor local minimum instead of converging to the global minimum [110]. For that reason, starting with a good initial codebook becomes important. Initial codebooks may be obtained by randomly choosing some of the training set vectors to be initial codevectors, or splitting can be used. These techniques are very well reviewed in [85] and [111] section 12.1.2

4.3.2 Pairwise Nearest Neighbor Search Algorithm

Pairwise Nearest Neighbor Search Algorithm (PNN) has been proposed as an alternative to GLA by Equitz [112]. But later Bottemiller revealed that [113], it was identical to Ward's hierarchical clustering method [114].

At the start of the algorithm each of M training set vectors are treated as individual clusters and merges the nearest two, reducing the number of clusters to $M - 1$. This merging is meant to continue until a desired number of clusters are obtained or the average distortion exceeds a predetermined threshold. But unfortunately when clusters got more than one member, one has to take into account not only the closeness of the centroids of the clusters, but also the number of members in the candidate pair. For instance, merging two clusters one having hundred vectors and the other only one vector, may cause a higher overall increase in the distortion than the case where clusters having four vectors and one vector are merged, even though the former pair had their centroids closer, since for the former merge a much larger number of vectors must be moved. So a new closeness measure is developed rather than just using Euclidean distance :

$$n_{ij}S_{ij}^2 = n_iS_i^2 + n_jS_j^2 + \frac{n_in_j}{n_i + n_j} |\bar{x}_i - \bar{x}_j|^2, \quad (4.45)$$

where n_i is the number of training vectors in i^{th} cluster, $n_{ij} = n_i + n_j$, \bar{x}_i is the centroid of the i^{th} cluster, S_i^2 is the average squared error between the centroid and vectors of the i^{th} cluster, and S_{ij}^2 is that of the cluster formed by merging the i^{th} and j^{th} clusters.

The key to efficient execution of this algorithm is to quickly find the two closest pairs. In [112] usage of $k-d$ trees is employed so that the complexity of the entire PNN algorithm (including the generation of the tree and searches) would be $O(M \log_2 M)$ which is independent from the desired size of the codebook. On the other hand, a single merge of full-search has a computational complexity of $T \log_{10} T$, if there exists T clusters at the moment.

Full-search PNN has a performance comparable to that of GLA, but fast PNN, lacks optimality and has worse performance. Still it is favourable, since it runs about twice as fast as GLA. If output of PNN algorithm is fed to GLA as initial codebook, GLA converges in about half as many iterations without the danger of being trapped in poor local minimum since it has been started with a good initial condition. In that case the resulting codebook performs better than both GLA and also PNN generated codebooks.

4.3.3 Kohonen Learning Algorithm

Kohonen learning Algorithm (KLA) was originally proposed by Kohonen [115], as a supervised learning scheme for self-organization of artificial neural networks as well as for Voronoi tessellation (partitioning) of vector quantizers. Considering an initial codebook, each time a training vector arrives KLA calculates the closest codevector and tilts it towards the training vector.

This direct application can be modified to be a supervised learning scheme (also known as Learning Vector Quantization which is available via anonymous ftp in a nice program package) where both classification and compression can be done. Let us assume that each codevector has a class label accompanying its codebook address label, then KLA can supervise the training of the codebook, if it knows the true classification of the training set vectors, by rewarding correct classifications and punishing incorrect ones. With the presentation of a training vector, if the class of closest codevector matches that of the training vector, then the codevector is moved towards the training vector. Otherwise, it is punished by being moved away from the training vector.

When applied to VQ, KLA can be generalized as follows. With the presentation of a new training-set vector $\vec{X}(n)$ at the n^{th} iteration, each codevector $\vec{Y}_i(n)$

is updated using the formula

$$\begin{aligned}\vec{Y}_i(n) &= \vec{Y}_i(n-1) + a_i(n)h_i(j^*, n)[\vec{X}(n) - \vec{Y}_i(n-1)] \\ j^* &= \operatorname{argmin}_j \|\vec{X}(n) - \vec{Y}_j(n-1)\|.\end{aligned}\quad (4.46)$$

where $a_i(n) \in [0, 1]$ is the step-size for i^{th} codevector at n^{th} iteration, which must asymptotically decay to zero as n increases. $h_i(j^*, n)$ is a neighbourhood function that scales the step-size of i^{th} codevector at n^{th} iteration, depending on its physical closeness to the $j^{*\text{th}}$ codevector which has the shortest Euclidean distance to $\vec{X}(n)$, the Euclidean winner. Therefore, KLA belongs to the class of competitive learning algorithms.

Then, vectors in a certain neighbourhood of the Euclidean winner will be updated. In the original proposal neighbourhood was reduced to update only the winner, i.e.

$$h_i(j^*, n) = \begin{cases} 1 & \text{if } i = j^* \\ 0 & \text{if } i \neq j^*. \end{cases}\quad (4.47)$$

Actually, for the distortion to converge to a local minimum i.e., for the nearest neighbor and centroid conditions to hold, it suffices to assume that only the Euclidean winner is updated [116].

Initial codebook $\mathcal{C}(0) = \{\vec{Y}_1(0), \vec{Y}_2(0), \dots, \vec{Y}_K(0)\}$ can be chosen arbitrarily, satisfying the only condition that no two or more codevectors can be equal.

Assuming independent and identically distributed random vectors are fed to the algorithm at each iteration n , convergence of KLA, in the case of a single vector codebook was proved asymptotically, as $n \rightarrow \infty$ [116]. A sufficient condition for the convergence is found to be

$$a(n) \geq \frac{a(n-1)}{1 + a(n-1)} \text{ where } a(0) = 1. \quad (4.48)$$

Then

$$\sum_{n=0}^{\infty} a(n) = \infty \quad (4.49)$$

meaning, although $a(n)$ monotonically decays to zero, its sequece sum should diverge. The reason is, if there exists a single codevector, through the iterations of Equation 4.46, the value of that codevector should finally converge to $E\{\vec{X}\}$. Then, Equation 4.46 would become a mean estimator, and if Equation 4.48 holds its estimation error variance asymptotically becomes zero.

However there is no rigorous proof of convergence for the multiple codevector case, for which the distortion function is not necessarily convex and contains multiple local minima. But if it is assumed that the algorithm converges to some extent, the step-size would decrease so much that the behaviour of each codevector in its own cell will obey the asymptotic property discussed above. Therefore, convergence of each codevector to the centroid of its own cell would be guaranteed, and centroid condition would be met to yield a locally optimum codebook.

The lowest convergence time was shown to be achieved with the unique schedule of $a(n) = 1/n$. However in practice, hyperbolic, or exponential or linear step-size schedules are preferred [116]. In the above learning formula of Equation (4.46), notice that each codevector has a step-size updating formula of its own. This was not the case in the original proposal by Kohonen. But, treating each codevector individually and hence updating its assigned step-size only when it is the Euclidean winner fits better to the convergence analysis of KLA. Also it gives quite a flexibility in step-size scheduling so that schedules for different input types can be derived.

The neighborhood function of Equation (4.46), is used to ensure that enough number of codevectors would be affected with the presentation of a new training vector. If only the Euclidean winner is updated throughout the iterations (i.e. there exists no neighborhood), it might happen that some of the codevectors never win due to poor initial conditions. Solutions for better neighborhood mechanism can be derived (see Section 4.3.4), both to solve this problem, and also for better partitioning the input space for obtaining better local minima of the distortion function.

Since KLA updates the codebook with each presentation of a training set vector it acts on instantaneous gradient. Thus, unlike true gradient algorithms like GLA the distortion function is not necessarily decreases monotonically and so it may escape from poor local minima. Another important property of KLA is that since it is an on-line algorithm, KLA is amenable to parallel implementations.

Finally as a practical note, although the original algorithm requires a new training set vector to be introduced at each iteration, in our experiments we have used a limited training set in our application, but introduced it to the algorithm repeatedly, until the algorithm converges.

4.3.4 Stochastic Kohonen Learning Algorithm

Recall from Section 4.3.1 that GLA generates only locally optimum VQ codebooks. To achieve globally optimal performance, algorithms must be devised that do not share the deterministic nature of GLA. A family of optimization techniques called ‘stochastic relaxation’ (SR) is characterized by the common feature that each iteration for the search for the minimum of a ‘cost function’ consists of perturbing the ‘state’, i.e. the set of independent variables of the cost function, in a random fashion. Applying SR to VQ cost function is the distortion function and the state is the codebook. The magnitude of the perturbations decreases in time so that the convergence is achieved with probability one. Since the perturbation can be such that it increases the cost function, the non-zero probability of accepting the higher cost enables the SR techniques to escape from poor local minima and find a solution closer to the global minimum if the perturbations are ‘frozen’ sufficiently slowly. An important family of SR algorithms is known as ‘simulating annealing’ (SA) [117]. It was Kirkpatrick et al [118], who introduced a general way of finding the global minimum of a multi-dimensional non-convex cost function defined over a finite state space under the name of simulated annealing. They chose such a name for their method, as they were greatly influenced from a scheme of simulating the dynamics of a melted substance in a heat bath. Since the cost function in their method is decreased in a non monotonic way, it was a stochastic procedure. Later SA is successfully used to solve the problem of globally optimum codebook design in VQ [119]. A detailed discussion on the design of globally optimum vector quantizers using SA and SR can be found in [120]. SA has also been applied to still image compression. In [121] comparison of SA with other codebook generation techniques for VQ image compression is given.

Although SA achieves the goal of globally optimum codebook generation the execution time of the algorithm is very long, hence not suitable for practical applications. As a practical example, if codebook size is 256 and vector dimension is 4, it is about 1050 times slower than GLA for a training set of zero-mean, unity variance uncorrelated Gaussian samples and about 1650 times slower than GLA for a training set of zero-mean, unity variance Gauss-Markov1 source of $\rho = 0.8$ [121]. Inspired from SA algorithms in [116] an algorithm called ‘soft competition

scheme' is developed, that modifies the neighborhood and step-size function of KLA in a stochastic manner, therefore can be called as 'Stochastic Kohonen Learning Algorithm' (SKLA). SKLA is only about 20-25 times slower than GLA. SKLA has been adapted to image sequence coding with slight changes in [122].

Consider a training-set $\mathcal{X} = \{\vec{X}_1, \dots, \vec{X}_M\}$ of M k -dimensional vectors and a state vector \vec{S} of size M which contains corresponding states (e.g. codevector labels) of each vector in the set. \mathcal{X} can be partitioned into N distinct regions \mathcal{R}_i (e.g. Voronoi partitioning of a vector quantizer), by assigning each vector \vec{X}_i in \mathcal{X} , one of N states $s_i \in \{1, 2, \dots, N\}$ such that if $\vec{X}_i \in \mathcal{R}_n$, then $s_i = n$ (a typical VQ encoding of the training set). Finally, consider a distortion value $D(\vec{S})$, associated with any \vec{S} i.e. any partition of \mathcal{X} such that

$$D(\vec{S}) = \sum_{i=1}^M d_k(\vec{X}_i, \vec{Y}_{s_i}) \quad (4.50)$$

where \vec{Y}_n is the centroid of the region \mathcal{R}_n

Theory of simulated annealing, states that, the distortion value $D(\vec{S})$, associated with the state vector \vec{S} , can be globally minimized by iteratively updating \vec{S} using any stochastic transition rule (e.g. each iteration of a codebook design algorithm) between the states which yields an inhomogenous, finite-state Markov chain with Gibbs stationary distribution of $Pr(\vec{S}) = Ae^{-D(\vec{S})/T}$, if the parameter T is reduced sufficiently slowly. Here $Pr(\vec{S})$ is the probability to attain a specific state \vec{S} . Now let's make an analogy between the VQ codebook and the state vector \vec{S} . Assume that, training set of vectors are encoded using the updated codebook yielded at the end of each iteration of a codebook generation algorithm, and also assume that the labels of the encoded training set vectors are saved in a so called state vector. Then the codebook generation algorithm turns out to be a vector source. If this has the above property of distribution, the source will eventually converge and start to output identical vectors. The theory of simulated annealing promises that, the partition imposed by this final state vector on the training set, is the globally optimum partition.

Reconsider Equation (4.46), the codebook update rule of KLA. Due to the analogy stated above the neighbourhood function is the stochastic transition rule between the successive updates of the codebook such that overall iteration process would represent a Markov chain with Gibbs distribution. Hence, direct dependence of neighbourhood function on the Euclidean winner is dropped and

we change $h_i(j^*, n)$ in Equation (4.46) into $h(i, n)$, where

$$h(i, n) = \begin{cases} 1 & \text{with probability } p = Ae^{\frac{-d_k[\bar{X}(n) - \bar{X}_i(n-1)]}{T}} \\ 0 & \text{with probability } 1 - p. \end{cases} \quad (4.51)$$

Step-size function, obeying the conditions in Equation (4.48), can be selected as,

$$a_i(n) = \frac{1}{\frac{1}{a_i(n-1)} + h(i, n)p} \text{ where } a(0) = 1. \quad (4.52)$$

T in Equation (4.51) is the temperature and it is decreased sufficiently slowly at each pass of the training-set. In the limit $T \rightarrow 0$ (freezing), probability of $h(i, n)$ becomes 1 for the Euclidean winner and 0 for others, meaning that the algorithm should have been converged by then. The cooling schedule chosen in our experiments is $T(m) = T_0\lambda^{-m}$, where m is a counter which is increased by one at each sweep through the training-set. Choice of T_0 and λ depends on the statistics of the training-set vectors. In our experiments with slowly varying image sequences we have chosen T_0 as 15° and λ as 1.07.

4.4 DCT Domain Classified Vector Quantization

Generally, classified vector quantization (CVQ) paradigm, involves the subdivision of the VQ codebook so that each subcodebook could be used for vector quantizing a particular class of input data. A classifier capable of differentiating between different types of features determines the class of an incoming vector and switches the appropriate codebook for that class to quantize the vector (See Figure 4.3). Subcodebooks may have different sizes and each of them can be generated by a different distortion measure or may use a different measure while choosing a codevector to encode a given vector. The rationale for this approach is that numerous small codebooks, each tuned to a particular class of vectors, can provide comparable or perceptually higher image quality with lower search complexity as compared to a single large codebook.

All the subcodebooks can be concatenated to form a union codebook and so all codevectors can be labeled from 1 to N . Then no overhead information would be required since class information would be included in the label of the codevector already. N is the size of the overall codebook and

$$N = \sum_{i=0}^M N_i. \quad (4.53)$$

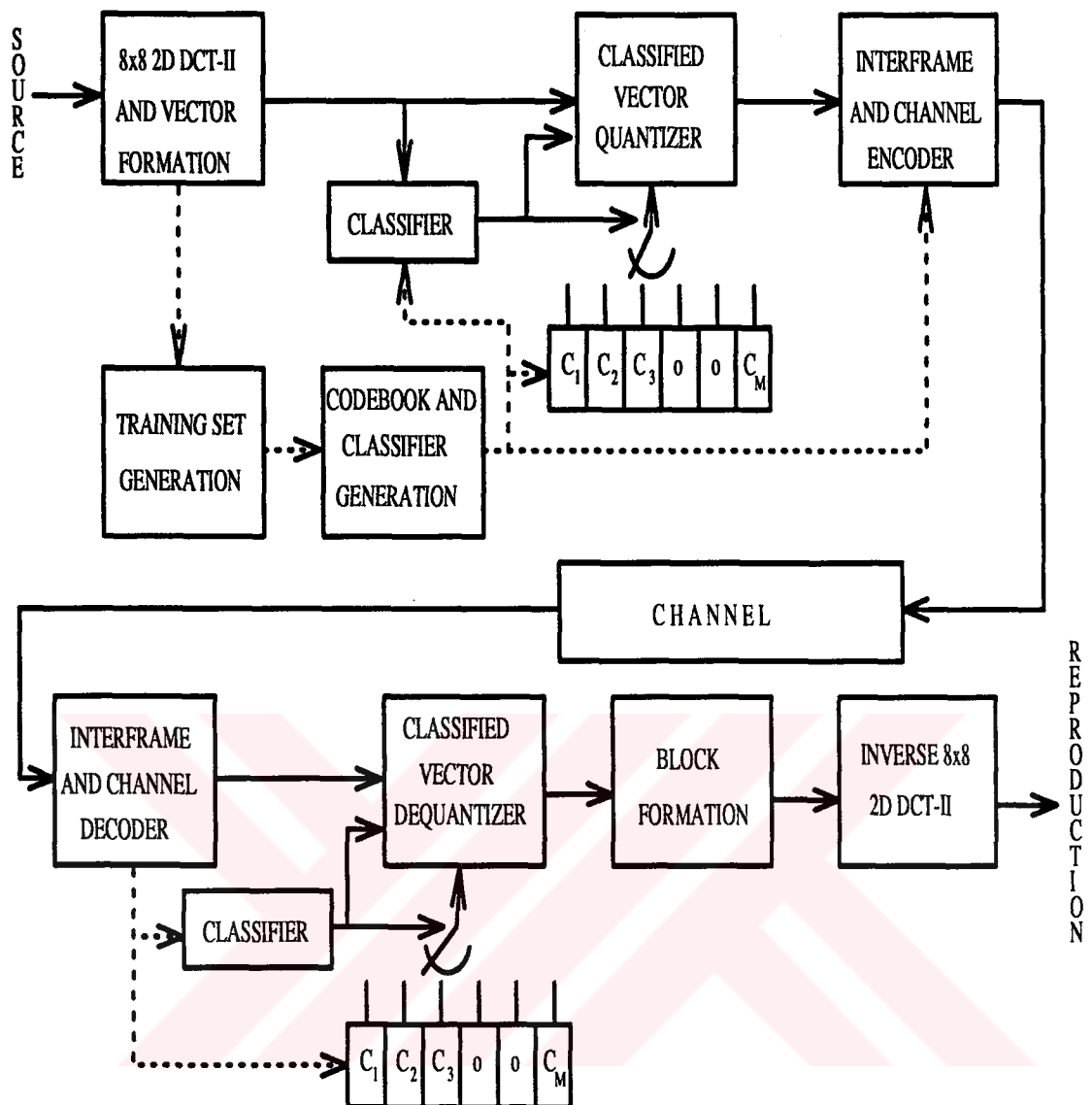


Figure 4.3. A DCT-CVQ system for Intra/interframe coding of image sequences where M is the number of classes and N_i s are the subcodebook sizes. It is also possible to separately encode the class label (from 1 to M) and the codevector index within that class (1 to N_i). But then, M and N_i s should be a power of 2 or else, the result would be a variable rate code. That is, codevector labels would have variable lengths.

In DCT domain some of the classification criteria may be

- Block activity : A measure of activity can be used to classify the transformed blocks. The most popular measure of activity is the AC-energy

content, which was first proposed by Gimlett [123]. Such an activity measure can be expressed as

$$E^{AC} = \left[\sum_{i=1}^N \sum_{j=1}^N a(i, j) f^2(i, j) \right] - a(0, 0) f^2(0, 0). \quad (4.54)$$

where $a(i, j)$ is a weighting factor which is suggested to be inversely proportional with the corresponding coefficient variance by Gimlett. Later, McLaren and Nguyen, proposed that it should be proportional to the frequency of the corresponding coefficient to account for the human visual system properties [124].

- **Spatial edge type :** Although edges constitute only a small spatial portion of the images, they carry most of the perceptual information content. The reason is, there is neurobiological evidence that there are special cells in the brain, which are sensitive to edges and psychophysiological evidence that these cells form groups in the visual cortex, each group being sensitive to a certain orientation of the edge, within a range of angles (see Section 3.4). Therefore human visual perception is very susceptible to the degradations along edges. On the other hand, none of the metrics that can be used in VQ accounts well enough for the degree of distortion in edges. In particular Euclidean distance is a poor measure for the edge distortion. Since there exists relatively less number of edge blocks in the training set and there is no preferential treatment to them by the distortion metric, edges are not well represented in the codebook. That's why, all VQ schemes suffer from edge jaggedness, in other words the staircase effect along the edges. Even when there exists enough number of edge codevectors, still a given suitable metric for VQ, like Euclidean distance, does not ensure that edge vectors would be coded by edge codevectors. These facts demonstrate enough reason for the attractiveness of CVQ using edge-oriented classifiers. In DCT domain edge-oriented classifier can be designed, since it is known that, certain DCT coefficients reflect orientation and distinction of various edge patterns [125].

Upon making the decision on the classification criteria, a suitable method for designing the codebook must be devised, and the problem of obtaining optimal subcodebook sizes must be solved. In the following section existing methods for this task and a novel classified VQ, codebook generation algorithm will be given.

4.4.1 Existing Methods and Their Weaknesses

Notice from Equations 4.32, 4.36 and 4.37 distortion incurred by vector quantization increases with vector variance and decreases with codebook size. Therefore, for each subcodebook to contribute equal amount of distortion to the overall distortion subcodebook sizes should be inversely proportional to the vector variances of the vectors to be quantized with that subcodebook. This problem of optimum subcodebook size determination in CVQ is analogous to the optimum bit allocation problem in block quantization and should certainly be solved for optimum performance.

The first systems with classification in the literature were using AC-energies as classification criteria having the form of Equation (4.54). Chen and Smith used it with no weighting factors to classify the transformed image frame subblocks into equally populated levels [27]. The number of classification levels was generally dependent upon the image size, coding rate, and dynamic range of AC energies of the subblocks. Then they employed scalar quantization with optimum bit allocation. Adaptation of such a classification scheme to CVQ is simple. In that case, average AC energies of the subblocks (vectors) in a block would be comparable to the average vector variance of vectors in that level. But then, the range of vector variance would be large for reasonable number of levels and the problem of optimum subcodebook size determination would become untractable.

Another classification criteria is edge-oriented classification which was first introduced by Ramamurthi and Gersho [126]. Their system involved no transforms and they used a spatial edge-oriented classifier which discriminates between primarily shade (no gradient), midrange (moderate gradient, no edge), horizontal edge, diagonal edge, and mixed (no definite edge but significant gradient) image subblocks. They obtained subcodebooks for each class using full-search GLA and employed a brute-force method to obtain optimal subcodebook sizes by generating locally optimum subcodebooks for every permissible size until an asymptotically true relation of distortion (which is not valid for all classes!) is met partially.

For edge-oriented classification in DCT domain, one can also use the DCT coefficients which are on the first row ($V_i = c_{0i}, i = 1, \dots, N$) and first column ($H_i = c_{i0}, i = 1, \dots, N$) of the the DCT block of size $N \times N$ [127]. The usage of

V_i and H_i for edge detection is as follows.

- Their magnitude reflects the distinction of the edge i.e. how fast the intensity changes within the edge.
- Their polarity reflects the orientation of the edge i.e. whether the intensity changes from darker to lighter or vice versa.
- Their relative magnitude reflects the direction of the edge i.e. the angle of the edge with respect to horizontal or vertical axis.

Observe in Figure 4.4 how values of the edge feature vector formed by using just V_1 and H_1 reflects the above listed properties of the edge. In the Figure 4.4, each (H_1, V_1) pair corresponds to a spatial edge pattern and a simple model of that pattern is drawn on the plane. Note that, the models on the circular path corresponds to the same $\sqrt{H_1^2 + V_1^2}$. As this value gets larger the edges become more distinct, as it gets smaller the edges become more pale.

By clustering the edge feature vectors $\vec{X}_f = \{H_1, V_1\}$, one can obtain a DCT domain edge oriented classifier, such that an incoming block is classified to the nearest (H_1, V_1) class node (see Figure 4.5).

To solve the problem of optimum subcodebooks sizes D. S. Kim and Lee [127] assumed that all subcodebooks were of the same size, but they have employed an edge feature vector clustering scheme with weighted MSE measure. Then, the classifier clustering scheme would give more weight to less distinct edge classes which have smaller vector variance and less weight to distinct edge classes which have larger vector variance. Then the partial distortion contribution from the classes would be balanced. But then, this weighted clustering scheme would not form the optimum partitioning on the class feature vector space. J. W. Kim, and Lee [125] proposed using bit allocation techniques of Section 3.3, but this is not possible since the exact quantizer functions of the vector quantizers for the subcodebooks can not be determined. Moreover, approximate distortion functions such as those in Section 4.2.1.2 would not hold as subcodebook sizes are relatively small, in other words not large enough for the high resolution quantization theory results to hold.

Riskin has proposed an optimal bit allocation method for CVQ subcodebooks [128] which uses generalized BFOS algorithm (named after the inventors

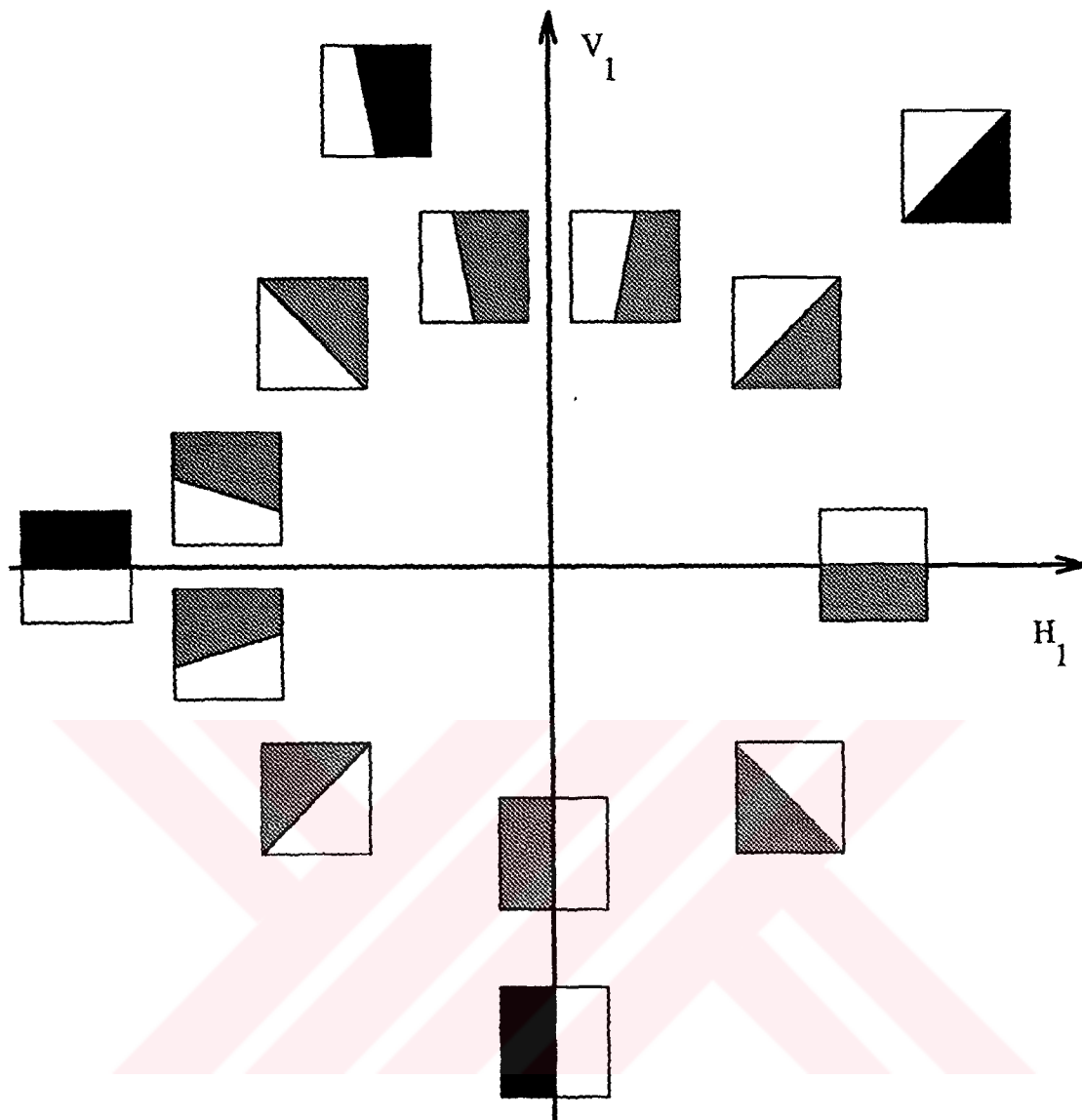


Figure 4.4. Corresponding spatial edge patterns in (H_1, V_1) plane

of the original algorithm : Breiman, Friedman, Olshen, and Stone [129]) to determine the lower convex hull of the distortion functions of the quantizer. But this method, also, requires unacceptable computation time for the practical applications demanding real-time encoding.

In fact, none of the above methods are appropriate and efficient for adaptive real-time applications. In the next section, we will present a new method for generating both the codebook and the classifier simultaneously in DCT domain.

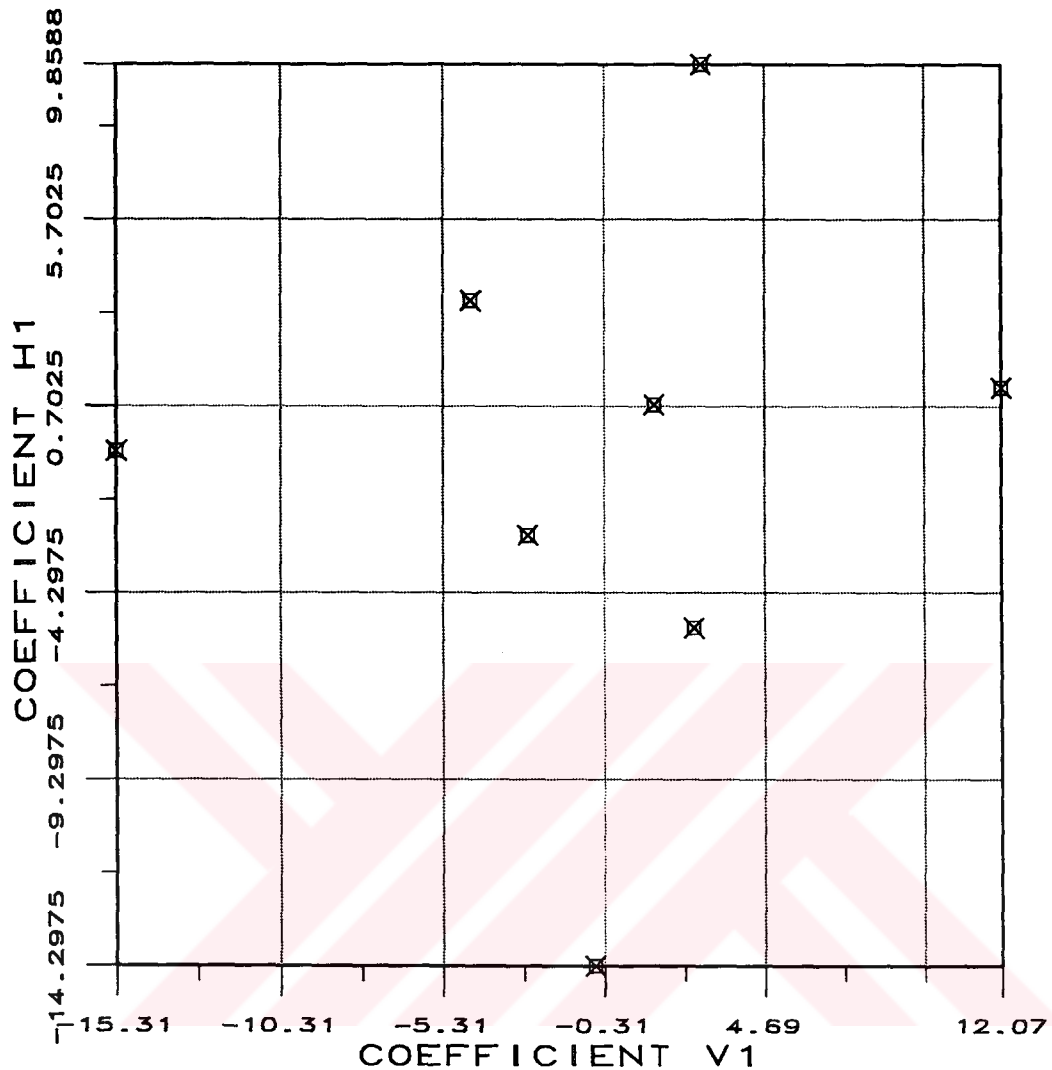


Figure 4.5. An 8-Node DCT domain direct classifier

4.4.2 Classifier Constrained Vector Quantization

A ‘classifier constrained vector quantizer’ [130] maps each input vector $\vec{X} \in \mathbb{R}^k$ onto a set of codewords, $Q(\vec{X}) = \vec{Y}_{i,j}$ where $\vec{Y}_{i,j} \in C_i$ and $C_i \subset C \subset \mathbb{R}^k$, subject to the constraint that all codewords in C_i are of the same class as the source vector. Given a vector $\vec{Z} \in \mathbb{R}^k$, let $\vec{Z}_f \in \mathbb{R}^m$ be the class feature vector of dimension $m \leq k$, extracted from \vec{Z} by choosing certain elements of \vec{Z} . Assume that the classifier partitions feature vector space into M regions, $\mathcal{P} = \{P_1, \dots, P_M\}$, and the quantizer partitions input vector space into N regions, $\mathcal{R} = \{R_{1,1}, \dots, R_{1,N_1}, \dots, R_{M,1}, \dots, R_{M,N_M}\}$, so that \vec{V}_i is the representative feature vector of the region P_i , and $\vec{Y}_{i,j}$ is the representative vector of the region $R_{i,j}$.

Then $\gamma(\vec{X}) = i$ if $\vec{X}_f \in P_i$ is the classifier, $\alpha(\vec{X}, \gamma(\vec{X})) = j$ if $\vec{X} \in R_{\gamma(\vec{X}),j}$ is the encoder and $\beta(j, \gamma(\vec{X})) = \vec{Y}_{\gamma(\vec{X}),j}$ is the decoder. For generating optimum quantizer and classifier partitions, the average quantizer and classifier distortions,

$$D_Q(\alpha, \beta, \gamma) = E[d_n(\vec{X}, \beta(\alpha(\vec{X}, \gamma(\vec{X})), \gamma(\vec{X})))]$$

and

$$D_{Cl}(\alpha, \beta, \gamma) = E[d_m(\beta(\alpha(\vec{X}, \gamma(\vec{X})), \gamma(\vec{X}))_f, \vec{V}_{\gamma(\vec{X})})]$$

where

$$d_r(\vec{A}, \vec{B}) = \frac{1}{r} \sum_{i=1}^r (a_i - b_i)^2$$

$$\vec{A} = [a_1, a_2, \dots, a_r]^T, \quad \vec{B} = [b_1, b_2, \dots, b_r]^T$$

must be jointly minimized. Analogous to entropy constrained VQ [131], methods, CCVQ algorithm [132], minimizes a Lagrangian modified distortion expression $J_\lambda = D_Q(\alpha, \beta, \gamma) + \lambda D_{Cl}(\alpha, \beta, \gamma)$, by alternately improving α , β and γ . In other words, starting with an initial coder $(\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)})$, a transformation

$$(\alpha^{(t+1)}, \beta^{(t+1)}, \gamma^{(t+1)}) = T(\alpha^{(t)}, \beta^{(t)}, \gamma^{(t)}) \quad (4.55)$$

is employed as follows:

1. Fixing $\gamma^{(t)}$ and $\beta^{(t)}$, $\alpha^{(t+1)}$ is calculated to minimize $J_\lambda(\alpha^{(t+1)}, \beta^{(t)}, \gamma^{(t)})$. This can be done by searching those codevectors which are in the same class as the incoming training vector and choosing the label of the nearest one to encode the training vector.
2. Fixing $\gamma^{(t)}$ and $\alpha^{(t+1)}$, $\beta^{(t+1)}$ is calculated to minimize $J_\lambda(\alpha^{(t+1)}, \beta^{(t+1)}, \gamma^{(t)})$. This can be done by choosing the centroid of the training set vectors which are mapped to the same label in the previous step, as the new codevector for that label.
3. Fixing $\alpha^{(t+1)}$ and $\beta^{(t+1)}$, $\gamma^{(t+1)}$ is calculated to minimize

$$J_\lambda(\alpha^{(t+1)}, \beta^{(t+1)}, \gamma^{(t+1)}).$$

This can be done by the choosing the centroid of the class features of the training set vectors which are found to be in the same class, at the previous steps, as the new node for that class.

Steps (2) & (3) can be realized by the generalized Lloyd algorithm (GLA), while Step (1) can be implemented by a full search algorithm since the subcodebook sizes are small enough.

Here is the outline of the algorithm with Steps 1,2, and 3 corresponding to the same steps above :

Step 0 Start with :

- A Lagrangian multiplier λ ,
- A termination constant ϵ ,
- A codeword index set \mathcal{K} ,
- A classifier-word index set \mathcal{I} ,
- A training set of vectors $\mathcal{X} \subset \mathbb{R}^k$,
- An initial codebook $\{\beta^{(0)}(j, i)\}_{j \in \mathcal{J}, i \in \mathcal{I}}$,
- an initial classifier $\{\gamma^{(0)}(\vec{X})\}_{\vec{X} \in \mathcal{X}}$,

Initialize :

- The iteration counter $t=1$,
- The Lagrangian distortion $J^{(0)} = \infty$.

$$\text{Step 1 } \alpha^{(t+1)}(\vec{X}, \gamma^{(t)}(\vec{X})) = \underset{j \in \mathcal{J}}{\operatorname{argmin}} d_n(\vec{X}, \beta^{(t)}(j, \gamma^{(t)}(\vec{X}))).$$

$$\text{Step 2 } \beta^{(t+1)}(j, i) = \underset{Y \in \mathbb{R}^k}{\operatorname{argmin}} E[d_n(\vec{X}, \vec{Y}) | \gamma^{(t)}(\vec{X}) = i, \alpha^{(t+1)}(\vec{X}, \gamma^{(t)}(\vec{X})) = j].$$

$$\text{Step 3 } \vec{V}_i = \underset{V \in \mathbb{R}^m}{\operatorname{argmin}} E[d_m(\vec{V}, \beta^{(t+1)}(j, i)_f) | \alpha^{(t+1)}(\vec{X}, i) = j].$$

$$\text{Step 4 } J^{(t+1)} = D_Q(\alpha^{(t+1)}, \beta^{(t+1)}, \gamma^{(t+1)}) + \lambda D_{Cl}(\alpha^{(t+1)}, \beta^{(t+1)}, \gamma^{(t+1)}).$$

$$\text{Step 5 If } |J^{(t)} - J^{(t+1)}| / J^{(t+1)} \leq \epsilon,$$

SET $t=t+1$ and GOTO (1)

Otherwise, QUIT.

The above CCVQ algorithm generates jointly optimal subcodebooks and a classifier. Optimality is in the sense that the distortion is minimized. That is the input vector space and classifier vector spaces are jointly optimally partitioned. However, optimality is local, as basically GLA is used to minimize the distortion which is proved to give only locally optimal results (see Section 4.3.1). GLA is chosen as the iterative technique to minimize distortion, because it is the fastest one among others presented in the sections above.

One very important virtue of this joint optimization of codebook and classifier simultaneously is that optimal subcodebook sizes are obtained as a side

product. Therefore, CCVQ should theoretically outperform all other CVQ methods discussed in Section 4.4.1, which suffer from the problem of non-optimal subcodebook sizes.

4.5 Experimental Work

Simulations of the GLA, KLA, SKLA, and CCVQ algorithm has been done on 256x256, 8 bpp, 150 frame image sequence Miss America. Each frame of the sequence is divided into 8×8 subblocks and DCT-II of each subblock is taken. DC coefficients are scalar quantized separately into 8 bits using a 256 level, non-uniform, Lloyd-Max quantizer. Since DCT non-DC coefficients are highly correlated along the zig-zag scan path and 25% of them is presumed to be sufficient for reconstruction with subjectively unnoticeable distortion [8], 14 non-DC coefficients are zig-zag scanned to form vectors. Codebook size is chosen to be 256, so that resulting bit-rate would be 0.25 bpp for each frame. A fine initial codebook extracted from the first frame of the sequence using PNN is fed to GLA and CCVQ algorithms. For SKLA and KLA algorithms initial codebook is obtained by sampling the training set formed from the first three frames of the sequence. Then first frame vectors are used as the training-set for all algorithms. Full search VQ with the generated codebooks are used to encode and decode all 150 frames at 0.25 bpp without codebook refreshment.

Overall PSNR performance of the algorithms, together with their codebook generation CPU-time on SUN Sparctation 10/30 with non-optimized C-language source codes and number of iterations they required for the particular experiment performed can be found in Table 4.1. Note that, at the given execution time and number of iterations in Table 4.1, CCVQ algorithm generates not only the codebook but also an 8-node classifier. Besides it can be observed from the execution times in the table that CCVQ is the fastest one among others due to the decreased search time by classification. The clustering map of the classifier nodes for this experiment can be found in Figure 4.5.

PSNR performance of the algorithms can be observed in Figures 4.6, 4.7, and 4.8.

For the sake of comparison CCVQ algorithm is also tested with a ready to use initial codebook and classifier formed from many head and shoulder shots of

Table 4.1. Comparison of overall PSNR and required CPU time for the four different algorithms. Iterations for SKLA and KLA indicates the number of times that the complete training set is introduced to the algorithms

	PNN-GLA	SKLA	KLA	PNN-CCVQ
PSNR	34.19 dB	34.07 dB	33.85 dB	34.00 dB
TIME	54.1 s	2697.1 s	1596.9 s	29.6 s
ITERATIONS	12	187	103	20

Table 4.2. PSNR performances of GLA, SKLA, KLA, and CCVQ algorithms with initial codebooks generated using PNN on frames 0,46,116, and 149 of the sequence.

	PNN-GLA	PNN-SKLA	PNN-KLA	PNN-CCVQ
Frame #0	38.99	38.23	36.94	38.97
Frame #46	34.72	34.68	34.35	34.43
Frame #116	34.59	34.51	34.33	34.38
Frame #149	33.41	33.61	33.41	33.53

human speakers. Starting with such an initial codebook and classifier, CCVQ algorithm is executed using the first frame vectors of the sequence for training. PSNR results of this newly generated codebook compared to those of CCVQ that accepted PNN generated initial codebook and classifier mentioned above, can be found in the Figure 4.9. Observe that CCVQ still performs well with overall PSNR 34.00 dB.

Also see Figures 4.11, 4.12, 4.13, and 4.14, for a subjective evaluation of your own on some of the encoded frames of the sequence, and compare them with the original frames in Figure 4.10. Table 4.2 can be referred for corresponding PSNR performances. Notice that, since the codebooks are generated using the vectors of the first frame of the sequence as training set vectors, PSNR of the first frame is very high.

4.6 Conclusion

Due to the substantial storage and computational costs typically associated with VQ (see Section 4.1) one should always check the performance of VQ, in any specific application, with respect to SQ to help assess the benefits accrued by VQ. For that purpose, below is the results of block quantization with optimum bit allocation applied to the sequence Miss America at 0.25 bpp rate.

The overall PSNR performance was 32.72 and optimum bit allocation map was found to be

$$\begin{bmatrix} 8 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

PSNR performance for individual frames is compared to that of CCVQ in Figure 4.15.

This result also confirms experimentally that VQ is superior to scalar quantization and should therefore be favourable especially for rates lower than 0.5 bpp.

A very important issue, also covered in this chapter is the novel CCVQ algorithm. CCVQ generates locally and jointly optimal subcodebooks and a classifier faster than the other methods tested. Due to joint optimization optimal subcodebook sizes are obtained. Having optimal subcodebook sizes equalizes the partial contribution of each subcodebook to overall distortion, and hence minimizes the distortion in the CVQ system. From this point of view CCVQ outperforms any other existing classified vector quantization system.

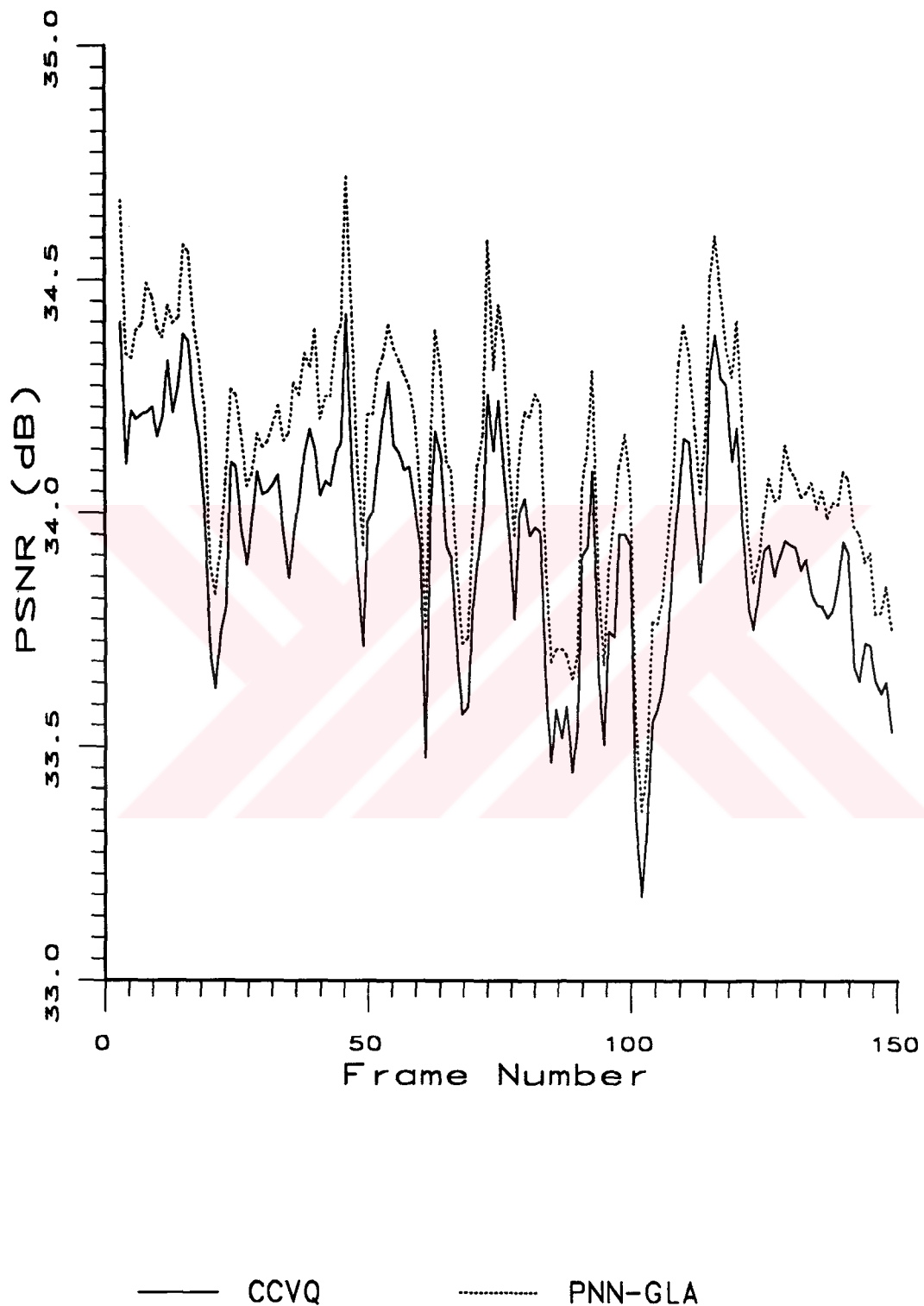


Figure 4.6. PSNR performance of PNN-GLA Scheme and CCVQ with adaptive initial codebook and classifier over the sequence.

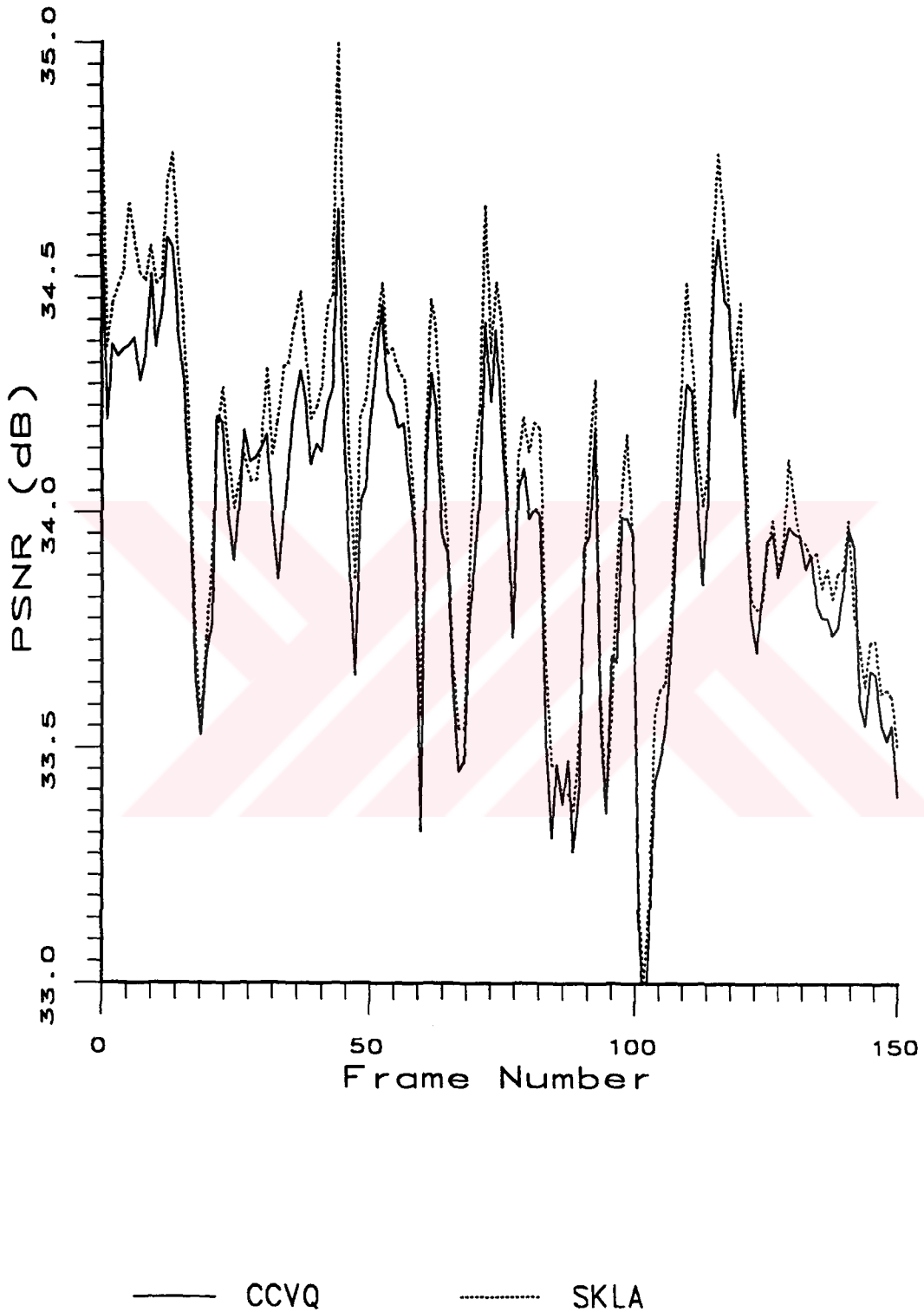


Figure 4.7. PSNR performance of SKLA and CCVQ with adaptive initial code-book and classifier over the sequence.

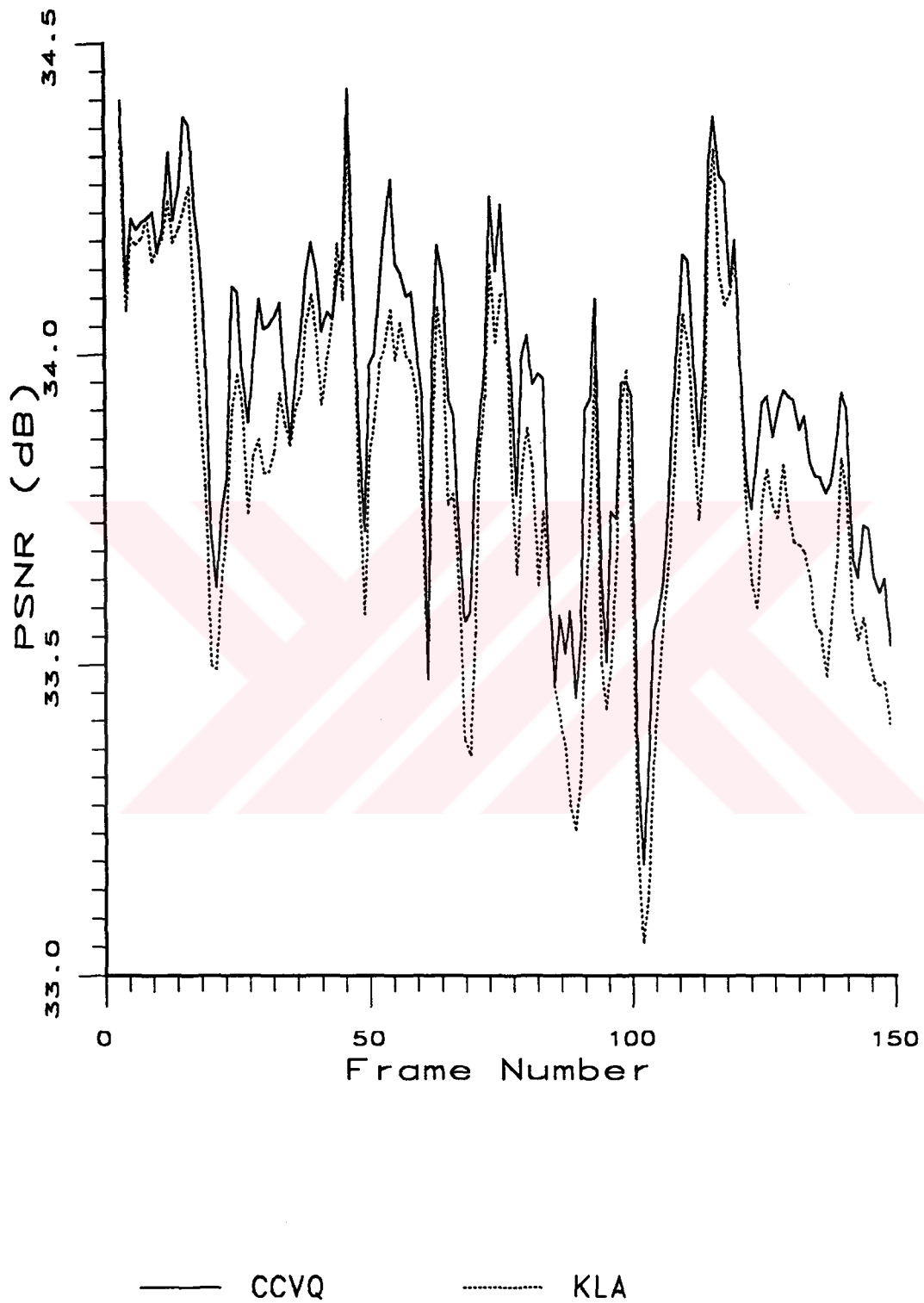
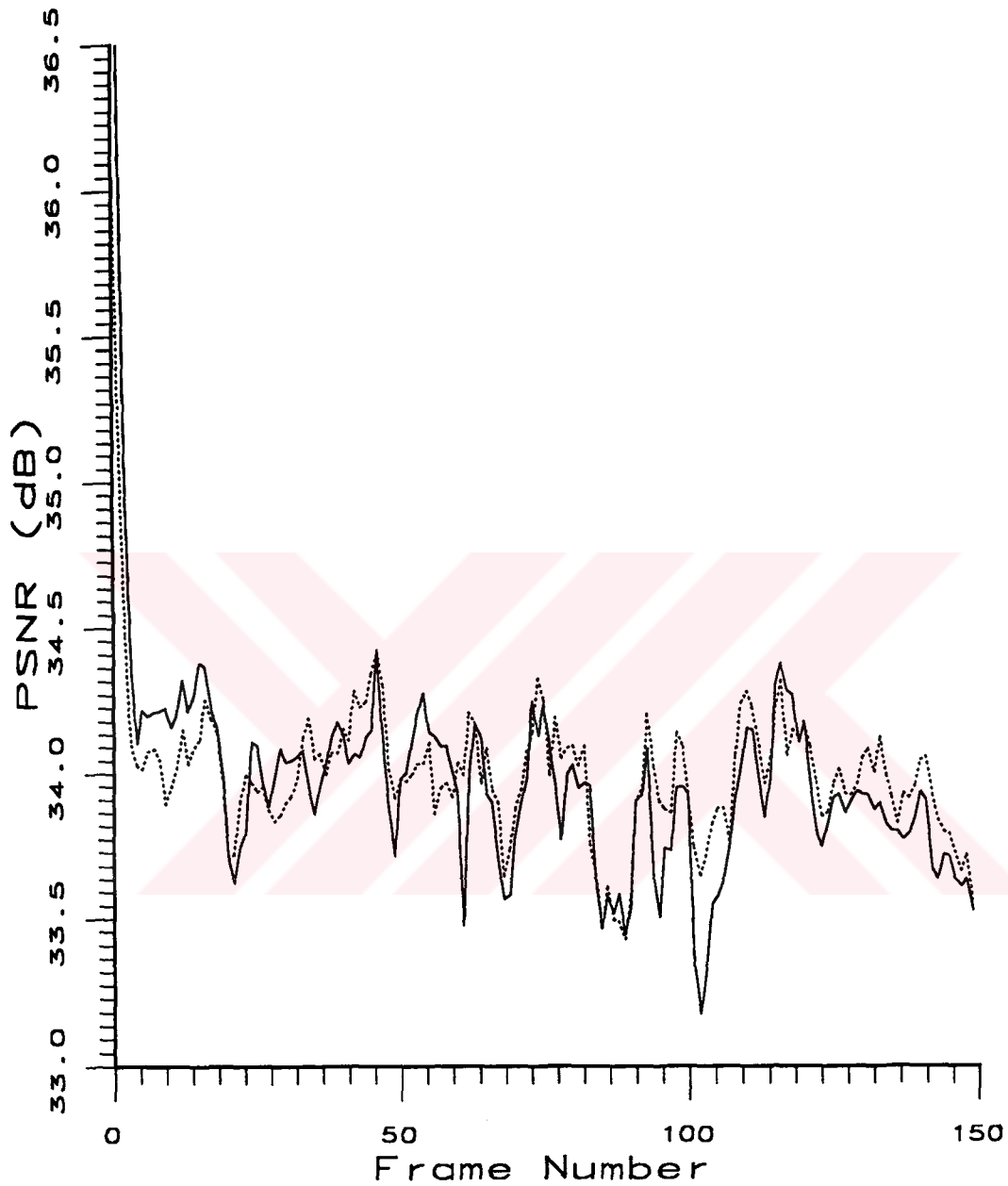


Figure 4.8. PSNR performance of KLA and CCVQ with adaptive initial codebook and classifier over the sequence.



— CCVQ-1 ····· CCVQ-2

Figure 4.9. PSNR performance of CCVQ algorithms. Initial codebook and classifier of CCVQ-1 are generated from the sequence, while those of CCVQ-2 are ready-to-use.



(a)



(b)



(c)

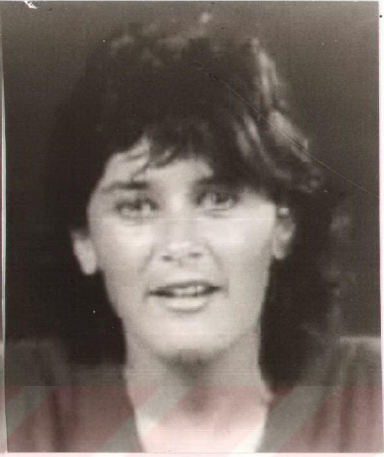


(d)

Figure 4.10. Original frames of number (a) 0 (b) 46 (c) 116 (d) 149.



(a)



(b)



(c)



(d)

Figure 4.11. First frame of the sequence compressed with (a) VQ with GLA (b) VQ with SKLA (c) VQ with KLA (d) CCVQ.



(a)

(b)



(c)



(d)

Figure 4.12. 46th frame of the sequence compressed with (a) VQ with GLA (b) VQ with SKLA (c) VQ with KLA (d) CCVQ.



(a)



(b)



(c)



(d)

Figure 4.13. 116th frame of the sequence compressed with (a) VQ with GLA (b) VQ with SKLA (c) VQ with KLA (d) CCVQ.



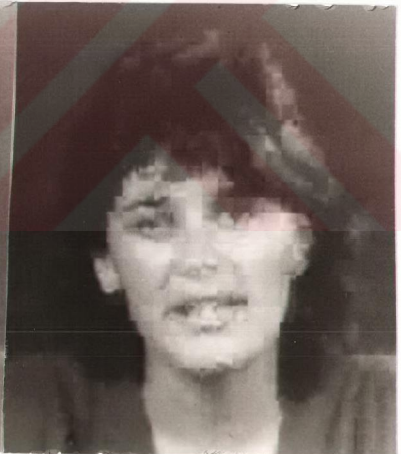
(a)



(b)

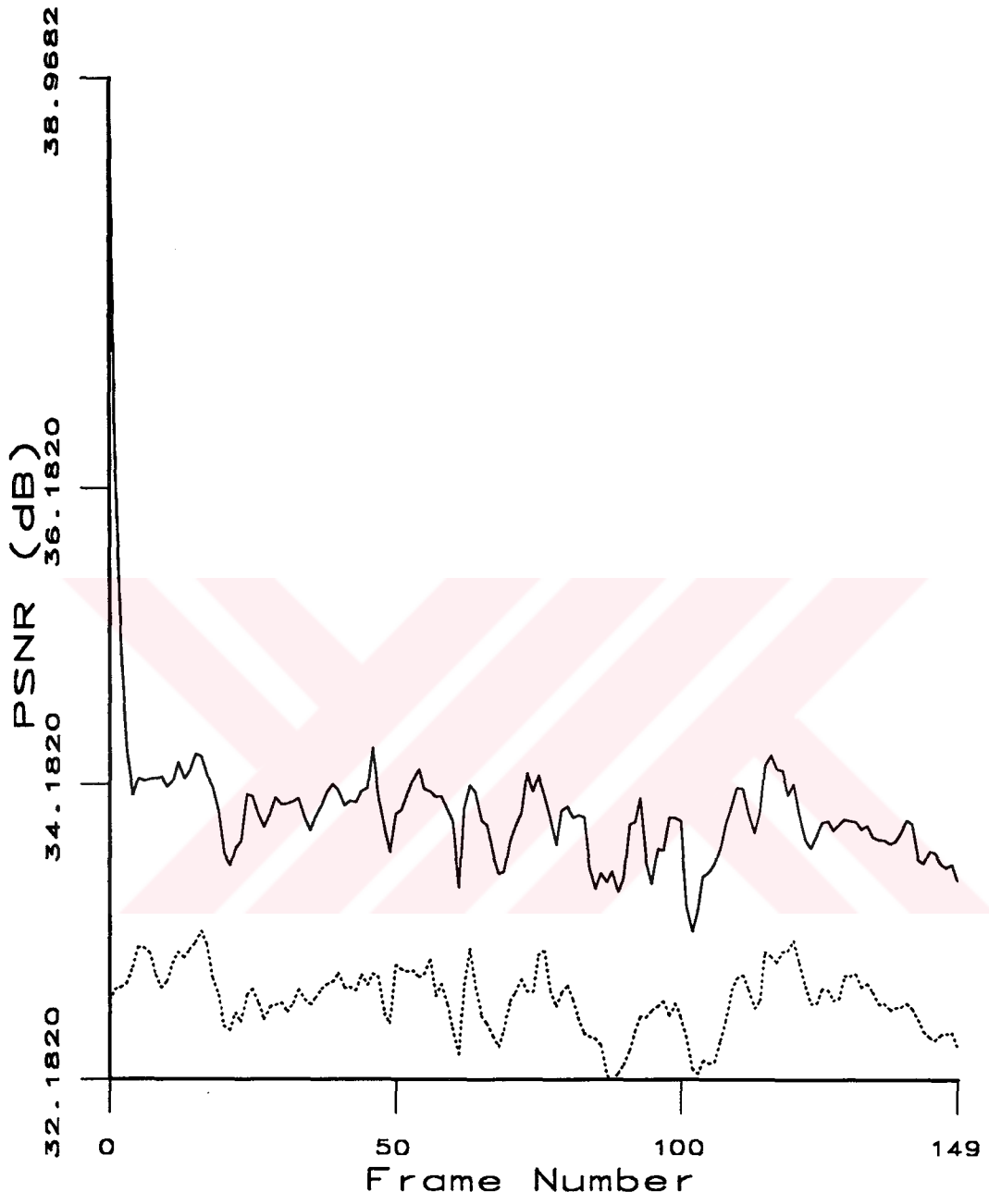


(c)



(d)

Figure 4.14. 149th frame of the sequence compressed with (a) VQ with GLA (b) VQ with SKLA (c) VQ with KLA (d) CCVQ.



— CCVQ SQ

Figure 4.15. PSNR performance of block quantization with scalar quantization and optimal bit allocation compared to CCVQ

CHAPTER V

INTERFRAME CODING FOR SCALAR QUANTIZATION AND VECTOR QUANTIZATION IN DCT DOMAIN

The main purpose of data compression is to excise the redundant information in a source of data. In the case of image sequences as the source of data the redundant information is both spatial and temporal. In the previous chapters optimal ways of encoding each frame of an image sequence using scalar quantization and vector quantization have been explained. These methods are optimal in the sense that they almost completely decorrelate spatial information in the image sequence. The subject of this chapter is the ways of removing redundancy in the temporal direction. As a matter of fact, although it depends on how fast the objects are moving in the view image sequence captured, typically the correlation in temporal direction is much larger than that in spatial detail.

Once spatial compression on the image sequence frames is carried out any interframe coding technique can be employed on the compressed data. The emphasis in this thesis has been on intraframe compression. However, for interframe compression, DPCM followed by entropy coding is tested for scalar quantization case and for vector quantization, a method for generating fixed rate code is examined, which is called label replenishment.

5.1 Interframe Coding for Scalar Quantized Image Sequence Frames

Among many possible techniques a simple variation of DPCM is chosen, just to demonstrate the potential of compression in the temporal direction. As the test sequence, again 150 frame Miss America sequence is chosen. Note that, Miss America is a 50 fr/sec sequence.

In terms of quantized DCT subblocks, the difference of each coefficients label in two successive frames of the sequence is found. Collecting label difference

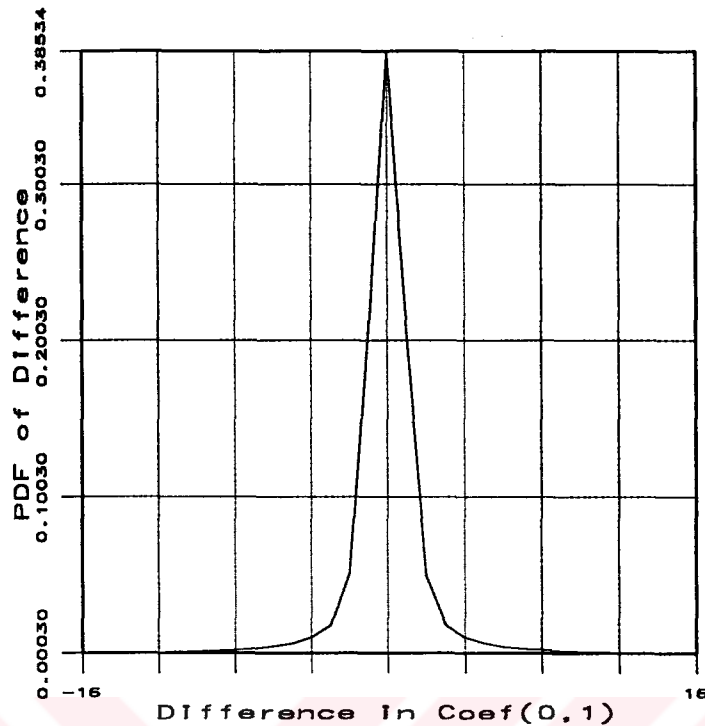


Figure 5.1. Probability density function of differences in non-DC DCT coefficient (0,1) which has been scalar quantized into 6 bits.

data for each coefficient, probability density functions for all coefficients which are quantized into more than one bit is found. The optimal bit allocation map for scalar quantization, found in Section 3.5 is used which is rewritten below :

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

In Figures 5.1, 5.2, and 5.3 probability density functions for some of the coefficients can be observed.

Now, assume an entropy coder which would encode the label differences into variable number of bits equal to their entropy. Existence of such prefix-free

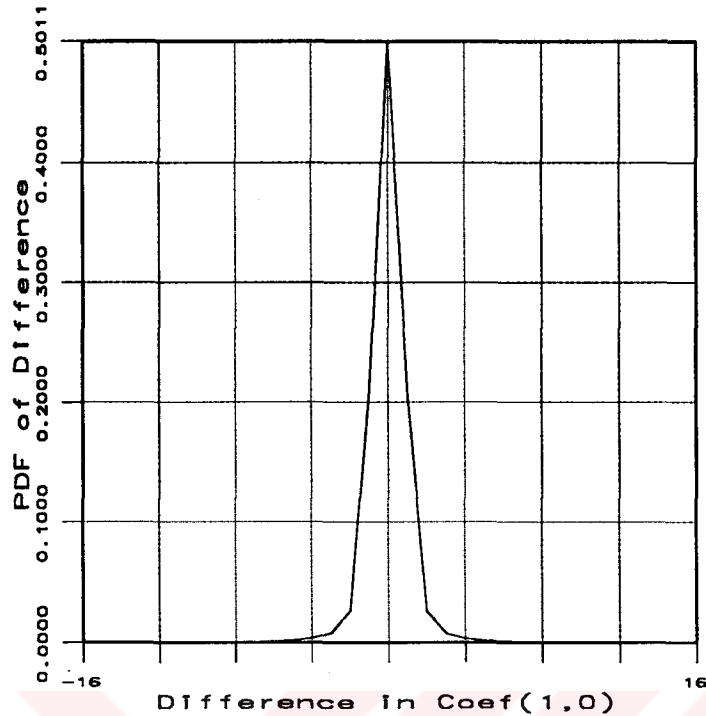


Figure 5.2. Probability density function of differences in non-DC DCT coefficient (1,0) which has been scalar quantized into 5 bits.

codes have been proved [1] and further Huffman proposed such a code [133]. In Table 5.1 corresponding entropies of label differences are given for all coefficients which have been allocated more than one bit. For those coefficients, encoded into one or zero bits, there is no need for further encoding. Summing up the given entropy values in the table and adding the one bit encoded ones, one can conclude that $30.489063 + 7 = 37.489063$ bits is actually enough for encoding each 8×8 DCT block and still obtain the same PSNR performances given in Figure 4.674823798. The rate will be $37.489063/64 = 0.59$ bits per pixel and if 50 fr/sec transmission of frames of size 256×256 is required, would demand a channel rate of 1933312 bits per second.

5.2 Interframe Coding for Vector Quantized Image Sequence Frames

It is interesting that, in the literature only a few methods have been proposed for interframe coding of VQ compressed image sequence frames [134], [135], [136], [137]. Again, just for the sake of demonstrating the potential of attainable

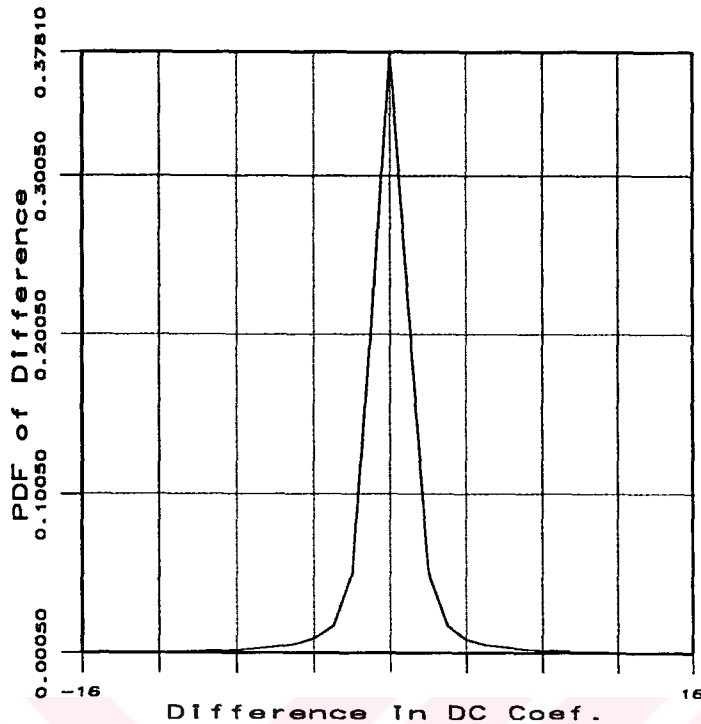


Figure 5.3. Probability density function of differences in DCT DC coefficient which has been scalar quantized into 8 bits.

compression, the simple method of label replenishment is chosen.

A label memory is devised, which keeps the VQ codevector label of each 8×8 DCT subblock of the previously encoded frame. Then, a comparison with the labels of the current frame is made and in case of coincidence an escape bit of 0 is sent. For differing labels escape bit is turned to 1. As mentioned before in Section 4.5, DC coefficients, even in VQ employing systems, are better scalar quantized. Although the differential scalar quantization labels for these DC coefficients can be entropy coded, like in the previous section, it is, now, preferred to design a fixed rate code. Therefore, examining Figure 5.3, for the probability density function of the differential DC coefficient labels, it is observed that 3 bits would be enough to represent differences within ± 8 , which covers the range of highest probability. Larger differences would be truncated, causing an ignorable degradation.

Experiments are carried on temporally sub-sampled Miss America sequence into 8.33 fr/sec, since this frame rate is sufficient for such a slowly varying sequence like Miss America. Afterall, if necessary frame interpolation can be made

Table 5.1. Entropies of differential DCT coefficient, which are scalar quantized into the given number of bits.

Coefficient	# Bits Allocated	Entropy (in bits/sample)
(0,0)	8	2.700021
(0,1)	6	2.663982
(0,2)	4	1.938385
(0,3)	3	1.766086
(0,4)	2	1.418827
(0,5)	2	1.787335
(1,0)	5	1.985687
(1,1)	4	1.884603
(1,2)	3	1.700194
(1,3)	2	1.600633
(2,0)	4	1.794678
(2,1)	3	1.671292
(2,2)	2	1.498267
(3,0)	3	1.587847
(3,1)	2	1.489350
(4,0)	2	1.488968
(5,0)	2	1.512908

at the receiver side of the system. With CCVQ as the VQ technique, simulation shows that, including the escape bit, the required transmission rate would be 80853 bits per second, meaning that compression to $80853/(8.33 \times 65536) = 0.148$ bpp is achieved. Since actually 50 fr/sec were considered, the effective compression is $80853/(50 \times 65536) = 0.0247$ bpp and compression rate achieved is $8\text{bpp}/0.0247 = 323$ times.

Notice that frame size of the sequence is actually quite large compared to those used in many very low bit-rate systems. A typical frame size would be 112×96 , in which case the demand for transmission rate may be reduced even down to 14400 baud, permitting a possible use on the existing telephony lines. Moreover, entropy coding would also reduce the bit-rate with no additional

distortion, but with a requirement for buffering.



CHAPTER VI

CONCLUSION

Transform Coding schemes are effective approaches for image data compression. Among various transforms investigated for image coding, DCT is preferred from the standpoints of ease of implementation and closeness to optimum Karhunen-Loève transform in energy compaction. Due to such virtues, DCT is primarily used in many image compression systems including all the existing standards for image compression.

Since DCT coefficients have real, continuous values, their lossless transmission over any finite bandwidth channel is impossible, and hence quantization is necessary. In fact, this is the case for all unitary transforms. Practical quantizers which map their input into finite number of levels (reproductions) always introduce error. However, this finite mapping also accomplishes compression. This opposing effects of quantizers with respect to communication engineers' wishes can only be dealt by optimizing the trade-off between the distortion and compression (rate). Given the output rate the optimum quantizer in this aspect is the one which introduces minimum distortion. In this thesis, optimal quantizers are sought for compressing image sequence data down to low bit-rates and very low bit-rates. For a quantizer to be optimized, information about the statistics of the source is essential. From this point of view transform domain quantization is appealing, since transform coefficients are actually weighted summations of the input and therefore contain statistical information. Similarly, intra/interframe coding is also appealing in this sense as quantizers used would act on spatial data rather than temporal (unlike motion estimation/ compensation paradigm) which have more stationary statistical properties.

In order to determine optimal quantizers and optimal systems which utilizes such quantizers, an extensive survey through the literature is necessary. An example from Section 3.2 can be given to clarify, the need for a careful literature

survey. Recently, Eggerton and Smith, [35], claimed that Cauchy distribution better represents the actual distribution of non-DC DCT coefficients for images. However, as shown in Equation (3.11), Cauchy distribution is not log-concave. Consequently, since it has been proven by Trushkin [46] that Lloyd-Max equations has a solution (meaning an optimal quantizer exists), for only log-concave distributions, there exists no quantizer, which is optimal for it. Therefore even if Cauchy distribution fits better, it has no use in practice. In case of scalar quantizers, this detailed survey has ended up with an optimal block quantization system which uses pdf optimized scalar quantization and marginal analysis for bit allocation, which has not been tried in the literature before. Moreover, to adopt a multidisciplinary approach, incorporation of HVS properties has also been tested. As a result, the best performing scalar quantization system, possible, without entropy coding is obtained. It is observed that incorporation of HVS in the way Section 3.4.3 describes does not exhibit sufficient improvement to compensate for the additional complexity. However, it “exhibits” an improvement. As conclusion, the approach is promising and may well be a future direction of research.

Shifting the attention from scalar quantizers to vector quantizers, a choice has been made in favour of classified vector quantization systems. The reason is that, the computational complexity of the resulting intraframe coding system should be decreased for the proposed systems’ being useful at practical real-time applications. Under scrutiny, it has been observed that all existing classified VQ schemes were lacking optimality due to arbitrary choices of subcodebook sizes. The major contribution of this thesis work for data compression science, emerges at this point. A new classified VQ scheme which jointly optimizes the quantizer and the classifier is introduced (the CCVQ algorithm). This new algorithm generates optimal VQ subcodebooks for each class with optimal sizes. Hence, it solves the problem of having optimal subcodebook sizes which all the other CVQ schemes suffer from. The resulting classified VQ system which had quite a reduced complexity and considerable edge-fidelity, is then compared to a number of full-search VQ techniques which aim to produce optimal codebooks. It is a fact that no VQ codebook, generated by remaining obedient to some kind of a rule (such as classification), can introduce same or less distortion than an independent

optimal codebook. Hence, the classified VQ codebook obtained via CCVQ algorithm would certainly introduce more distortion than the full-search codebooks used for testing. However, experimental results revealed that the performance of the new scheme is quite comparable. Therefore, this result has the promise that the new scheme can be very useful for image sequence compression since it is fast, efficient, and has high subjective fidelity. Again for further research, ways of incorporating HVS to VQ can be sought, which has only very rare examples in literature.

Another, less covered area in the literature is the interframe coding techniques using the compressed frames. Although, the potential of such techniques has been demonstrated in Section 5.2, in the literature there has been very little effort to develop methods for temporal codebook adaptation, efficient label and/or codeword replenishment. These problems which have not been dealt in this thesis, due to the focus on quantization and intraframe coding remains to be a wide-open research frontier.

Comparing best performing scalar quantization techniques to vector quantization, this thesis work also contains evidence that VQ is better than scalar quantization both in theory and experimentally. However, because of the substantial computational and memory costs associated with VQ, scalar quantization can be preferred at high rates. In addition scalar quantization is more robust, especially when used in an operational environment of channel errors. This is the reason why scalar quantizers are, until now, have been preferred by the existing standards like JPEG, and MPEG-I. As a result it can be concluded that VQ can be of great use for rates below 1 bpp.

In this thesis work, an attempt has been made to extensively survey existing solutions for the problems in quantization that the thesis addresses. Optimal methods are clearly identified, in spite of the fact that the underlying background in the literature was quite confusing. Some of these optimal methods have been brought together to obtain optimal systems, which have not been tested in the literature before. From this point of view, this work may be very valuable for those interested in the subject and those looking for new research areas under the subject of quantization, which has already been studied for about half a century, now.

On the other hand, a new, optimal method, CCVQ, has also been proposed.

Optimality is based on the statistics of the source, therefore, this new method can readily be used for coding of any source type, where classification and compression is needed. Image sequence compression is just a mere application.



REFERENCES

- [1] C. E. Shannon, "A Mathematical Theory of Communications", Bell Systems Technical Journal, 27, 379 (1948).
- [2] J. R. Pierce, "The Early Days of Information Theory", IEEE Trans. on Information Theory, 19, 3 (1973).
- [3] C. E. Shannon, "Coding Theorems for a Discrete Source with A Fidelity Criterion", IRE Nat. Conv. Rec., 4, 142 (1959).
- [4] A. K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall (1989).
- [5] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform", IEEE Trans. on Computers, 23, 90 (1974).
- [6] Z. Wang, "Fast Algorithms for the Discrete W Transform and for the Discrete Fourier Transform", IEEE Trans. on Acoustic, Speech, and Signal Processing, 32, 803 (1984).
- [7] H. Kitajima, "A Symmetrical Cosine Transform", IEEE Trans. on Computers, 29, 317 (1980).
- [8] K. R. Rao, and P. Yip, Discrete Cosine Transform, Academic Press (1990).
- [9] R. M. Haralick, "A Storage Efficient Way to Implement the Discrete Cosine Transform", IEEE Trans. on Computers, 25, 764 (1976).
- [10] A. Gersho, and R. M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers (1992).
- [11] A. K. Jain, "A Sinusoidal Family of Unitary Transforms", IEEE Trans. on Pattern Analysis and Machine Intelligence, 1, 356 (1979).
- [12] R. J. Clarke, Transform Coding of Images, Academic Press (1985).
- [13] W. K. Pratt, Digital Image Processing, John Wiley & Sons (1978).

- [14] A. K. Jain, “Advances in Mathematical Models for Image Processing”, Proceedings of the IEEE, 69, 502 (1981).
- [15] K. Karhunen, “Ueber lineare methoden in der Wahrscheinlichkeitsrechnung”, Ann. Acad. Sci. Fenn. Ser A.I. Math. Phys., 37, (1947)
- [16] M. Loève, Probability Theory, 2nd Ed., Princeton, Van Nostrand (1960).
- [17] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill (1965).
- [18] W. Mauersberger, “Generalised Correlation Model for Designing Two-dimensional Image Coders”, Electronic Letters, 15, 664 (1979).
- [19] R. Booton, and P. Ready, “Inadequacies of the Markov Model in Linear Predictive Coding of Images”, SPIE Semin. Proc., 204 (1987).
- [20] A. K. Jain, “Image Data Compression: A Review”, Proceedings of the IEEE, 69, 349 (1981).
- [21] W. D. Ray, and R. M. Driver, “Further Decomposition of the Karhunen-Loève Series Representation of Stationary Random Process”, IEEE Trans. on Information Theory, 16, 663 (1970).
- [22] M. Hamidi, and J. Pearl, “Comparison of Cosine and Fourier Transforms of Markov-I Signals”, IEEE Trans. on Acoustic, Speech, and Signal Processing, 24, 428 (1976).
- [23] F. A. Kamangar, and K. R. Rao, “Fast Algorithms for the 2D-Discrete Cosine Transform”, IEEE Trans. on Communications, 31, 899 (1982).
- [24] M. L. Haque, “A Two-Dimensional Fast Cosine Transform”, IEEE Trans. on Acoustic, Speech, and Signal Processing, 33, 1532 (1985).
- [25] N. Nasrabadi, and R. King, “Computationally Efficient Discrete Cosine Transform Algorithm”, Electronic Letters, 19, 24 (1983).
- [26] S. Venkataraman, V. R. Kanchan, K. R. Rao, and M. Mohnaty, “Discrete Transforms via the Walsh-Hadamard Transform”, Signal Processing, 14, 371 (1988).

- [27] W. Chen, and H. Smith, "Adaptive Coding of Monochrome and Color Images", IEEE Trans. on Communications, 25, 1285 (1977).
- [28] A. N. Netravali, and J. O. Limb, "Picture Coding: A review", Proceedings of the IEEE, 69, 366 (1980).
- [29] J. W. Modestino, D. G. Daut, and A. L. Vickers, "Combined Source-Channel Coding of Images Using the Block Cosine Transform", IEEE Trans. on Communications, 29, 1261 (1981).
- [30] H. Murakami, Y. Hatori, and H. Yamamoto, "Comparison Between DPCM and Hadamard Transform Coding in the Composite Coding of the NTSC Color TV Signal", IEEE Trans. on Communications, 30, 469 (1982).
- [31] K. N. Ngan, "Adaptive Transform Coding of Video Signals", IEE Proceedings, 129, 28 (1982).
- [32] R. C. Reininger, and J. D. Gibson, "Distribution of the Two-Dimensional DCT Coefficients of Images", IEEE Trans. on Communications, 31, 835 (1983).
- [33] W.-H Chen, and W. K. Pratt, "Scene Adaptive Coder", IEEE Trans. on Communications, 32, 225 (1984).
- [34] J. W. Modestino, N. Farvardin, and M. A. Ogrinc, "Performance of Block Cosine Image Coding with Adaptive Quantization", IEEE Trans. on Communications, 33, 210 (1985).
- [35] J. D. Eggerton, and M. D. Srinath, "A Visually Weighted Quantization Scheme for Image Bandwidth Compression at Low Data Rates", IEEE Trans. on Communications, 34, 840 (1986).
- [36] M. D. Paez, and T. H. Glisson, "Minimum Mean-Squared Error Quantization in Speech PCM and DPCM Systems", IEEE Trans. on Communications, 20, 225 (1972).
- [37] W. C. Adams, Jr., and C. E. Giesler, "Quantizing Characteristics for Signals Having Laplacian Amplitude Probability Distribution Function", IEEE Trans. on Communications, 26, 1295 (1978).

- [38] P. Noll, and R. Zelinski, "Comments on 'Quantizing Characteristics for Signals Having Laplacian Amplitude Probability Density Function'", IEEE Trans. on Communications, 27, 1259 (1979).
- [39] J. Max, "Quantizing for Minimum Distortion", IRE Trans. on Information Theory, 6, 7 (1960).
- [40] P. F. Panter, and W. Dite, "Quantisation Distortion in Pulse-Count Modulation with Non-uniform Spacing of Levels", Proceedings of the IRE, 39, 44 (1951).
- [41] S. P. Lloyd, "Least Squares Quantization in PCM", IEEE Trans. on Information Theory, 28, 129 (1982).
- [42] P. Kabal, "Quantizers for the Gamma Distribution and Other Symmetrical Distributions", IEEE Trans. on Acoustics, Speech, and Signal Processing, 32, 836 (1984).
- [43] K. N. Ngan, and K. S. Leong, "Fast Convergence Method for Lloyd-Max Quantiser Design", Electronics Letters, 22, 844 (1986).
- [44] N. S. Jayant, and P. Noll, Digital Coding of Waveforms, Prentice-Hall (1984).
- [45] P. E. Fleischer, "Sufficient Conditions for Achieving Minimum Distortion in a Quantizer", IEEE Int. Conv. Rec., 1, 104 (1964).
- [46] A. V. Trushkin, "Sufficient Conditions for uniqueness of a Locally Optimum Quantizer for a Class of Convex Error Weighting Functions", IEEE Trans. on Information Theory, 28, 187 (1982).
- [47] D. K. Sharma, "Design of Absolutely Optimal Quantizers for a Wide Class of Distortion Measures", IEEE Trans. on Information Theory, 28, 187 (1982).
- [48] R. C. Wood, "On Optimum Quantization", IEEE Trans. on Information Theory, 15, 248 (1969).
- [49] J. J. Y. Huang, and P. M. Schultheiss, "Block Quantisation of Correlated Gaussian Random Variables", IEEE Trans. on Communications, 11, 289 (1963).

- [50] L. D. Davisson, "Rate-Distortion Theory and Applications", Proceedings of the IEEE, **60**, 800 (1972).
- [51] T. Berger, Rate Distortion Theory, Prentice-Hall (1971).
- [52] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding", Proceedings of the IEEE, **73**, 1551 (1985).
- [53] J. A. Roese, W. K. Pratt, and G. S. Robinson, "Interframe Cosine Transform Image Coding", IEEE Trans. on Communications, **25**, 1329 (1977).
- [54] M. Tasto, and P. A. Wintz, "Image Coding by Adaptive Block Quantisation", IEEE Trans. on Communications, **19**, 957 (1971).
- [55] A. Habibi, "Hybrid Coding of Pictorial Data", IEEE Trans. on Communications, **22**, 614 (1974).
- [56] L. Wang, and M. Goldberg, "Progressive Image Transmission by Transform Coefficient Residual Error Quantization", IEEE Trans. on Communications, **36**, 75 (1988).
- [57] L. Wang, and M. Goldberg, "Block Transform Image Coding by Multistage Vector Quantization with Optimal Bit Allocation", IEEE Trans. on Communications, **39**, 1360 (1991).
- [58] P. A. Wintz, and A. J. Kurtenbach, "Waveform Error Control in PCM Telemetry", IEEE Trans. on Information Theory, **14**, 650 (1968).
- [59] A. Segall, "Bit Allocation and Encoding for Vector Sources", IEEE Trans. on Information Theory, **22**, 162 (1976).
- [60] Y. Shoam, and A. Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers", IEEE Trans. on Acoustics, Speech, and Signal Processing, **36**, 1445 (1988).
- [61] B. Fox, "Discrete Optimization via Marginal Analysis", Management Science, **13**, 210 (1966).
- [62] K.-H Tzou, "A Fast Computational Approach to the Design of Block Quantization", IEEE Trans. on Acoustics, Speech, and Signal Processing, **35**, 235 (1987).

- [63] J. A. Roese, W. K. Pratt, and G. S. Robinson, "Interframe Cosine Transform Image Coding", IEEE Trans. on Communications, 25, 1329 (1977).
- [64] D. J. Granrath, "The Role of Human Visual Models in Image Processing", Proceedings of the IEEE, 69, 552 (1981).
- [65] D. H. Hubel, Eye, Brain, and Vision, Scientific American Library (1988).
- [66] M. D. Levine, Vision in Man and Machine, McGraw-Hill, (1985).
- [67] T. N. Cornsweet, Visual Perception, Academic Press, (1970).
- [68] A. Fiorentini, G. Baumgartner, S. Magnussen, P. H. Schiller, and J. P. Thomas, "The Perception of Brightness and Darkness, Relations to Neuronal Receptive Fields", eds. L. Spillmann, and J. S. Werner, Visual Perception, The Neurophysiological Foundations, Academic Press (1990).
- [69] D. J. Sakrison, "On the Role of the Observer and a Distortion Measure in Image Transmission", IEEE Trans. on Communications, 25, 1251 (1977).
- [70] S. A. Karunasekera, N. G. Kingsbury "A Distortion Measure for Image Artifacts Based on Human Visual Sensitivity", Proc. Intl. Conf. on Acoust., Speech, and Signal Process., April 18-22, Adelaide, Australia, V-117 (1994).
- [71] J. Walraven, C. Enroth-Cugell, D. C. Hood, D. I. A. MacLeod, and J. L. Schnapf, "The Control of Visual Sensitivity, Receptor and Postreceptor Processes", eds. L. Spillmann, and J. S. Werner, Visual Perception, The Neurophysiological Foundations, Academic Press (1990).
- [72] S. W. Kuffler, "Discharge patterns and functional organization of mammalian retina", Journal of Neurophysiology, 16, 37 (1953).
- [73] G. Westheimer, and F. W. Campbell, "Light Distribution Formed by the Living Human Eye", Journal of the Optical Society of America, 52, 1040 (1962).
- [74] C. F. Hall, and E. L. Hall, "A Nonlinear Model for the Spatial Characteristics of the Human Visual System", IEEE Trans. on Systems, Man, and Cybernetics, 7, 161 (1977).

- [75] F. W. Campbell, and J. G. Robson “Application of Fourier Analysis to the Visibility of Gratings”, Journal of Physiology, 197, 551 (1968).
- [76] M. Davidson, “Perturbation Approach to Spatial Brightness Interaction in Human Vision”, Journal of the Optical Society of America, 58, 1300 (1968).
- [77] J. L. Mannos, and D. J. Sakrison, “The Effects of a Visual Fidelity Criterion on the Encoding of Images”, IEEE Trans. on Information Theory, 20, 525 (1974).
- [78] J. O. Limb, “Distortion Criteria of the Human Viewer”, IEEE Trans. on Systems, Man and Cybernetics, 9, 778 (1979).
- [79] T. G. Stockham, Jr., “Image Processing in the Context of a Visual Model”, Proceedings of the IEEE, 60, 828 (1972).
- [80] N. B. Nill, “A Visual Model Weighted Cosine Transform for Image Compression and Quality Assessment”, IEEE Trans. on Communications, 33, 551 (1985).
- [81] K. N. Ngan, K. S. Leong, and H. Singh, “Adaptive Cosine Transform Coding of Images in Perceptual Domain”, IEEE Trans. on Acoustics, Speech, and Signal Processing, 37, 1743 (1989).
- [82] B. Chitprasert, and K. R. Rao, “Human Visual Weighted Progressive Image Transmisson”, IEEE Trans. on Communications, 38, 1040 (1990).
- [83] J. Luo, C. W. Chen, K. J. Parker, and T. S. Huang “A New Method for Block Effect Removal in Low Bit-Rate Image Compression”, Proc. Intl. Conf. on Acoust., Speech, and Signal Process., April 18-22, Adelaide, Australia, V-341 (1994).
- [84] N. M. Nasrabadi, and R. A. King, “Image Coding Using Vector Quantization : A Review”, IEEE Trans. on Communications, 36, 957 (1988).
- [85] R. M. Gray, “Vector Quantization”, IEEE ASSP Mag., 1, 4 (1984).
- [86] D. Cohn, E. A. Riskin, and R. Ladner, “Theory and Practice of Vector Quantizers Trained on Small Training Sets”, unpublished technical report, recently submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.

- [87] R. M. Gray, and F. Saadat, "Block Source Coding Theory for Asymptotically Mean Stationary Sources", IEEE Trans. on Information Theory, **30**, 54 (1983).
- [88] J. C. Kieffer, "A Survey of the Theory of Source Coding with a Fidelity Criterion", IEEE Trans. on Information Theory, **39**, 1473 (1993).
- [89] F. Itakura, "Maximum Prediction Residual Principle Applied to Speech Recognition", IEEE Trans. on Acoustics, Speech, and Signal Processing, **23**, 67 (1975).
- [90] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design", IEEE Trans. on Communications, **28**, 84 (1980).
- [91] V. J. Mathews, "Multiplication Free Vector Quantization Using L_1 Distortion Measure and its Variants", IEEE Trans. on Image Processing, **1**, 11 (1992).
- [92] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image Compression Through Wavelet Transform Coding", IEEE Trans. on Information Theory, **38**, 719 (1992).
- [93] W. A. Finamore, and W. A. Pearlman, "Optimal Encoding of Discrete-Time Continuous-Amplitude Memoryless Sources with Finite Output Alphabets", IEEE Trans. on Information Theory, **26**, 144 (1980).
- [94] W. A. Pearlman, and A. Chekima, "Source Coding Bounds Using Quantizer Reproduction Levels", IEEE Trans. on Information Theory, **30**, 559 (1984).
- [95] R. P. Rao, and W. A. Pearlman, "Alphabet-Constrained Vector Quantization", IEEE Trans. on Information Theory, **39**, 1167 (1993).
- [96] R. Blahut, "Computation of channel capacity and rate-distortion functions", IEEE Trans. on Information Theory, **18**, 460 (1972).
- [97] Y. Yamada, S. Tazaki, and R. M. Gray, "Asymptotic Performance of Block Quantizers with Difference Distortion Measures", IEEE Trans. on Information Theory, **26**, 6 (1980).

- [98] T. D. Lookabaugh, and R. M. Gray, “High-Resolution Quantization Theory and the Vector Quantizer Advantage”, IEEE Trans. on Information Theory, 35, 1020 (1989).
- [99] W. R. Bennett, “Spectra of Quantized Signals”, Bell Systems Technical Journal, 27, 446 (1948).
- [100] J. A. Bucklew, “Two Results on the Asymptotic Performance of Quantizers”, IEEE Trans. on Information Theory, 30, 341 (1984).
- [101] V. R. Algazi, “Useful Approximations to Optimum Quantization”, IEEE Trans. on Communications, 14, 297 (1966).
- [102] H. Gish, and J. N. Pierce, “Asymptotically Efficient Quantizing”, IEEE Trans. on Information Theory, 14, 676 (1968).
- [103] R. M. Gray, and A. H. Gray, Jr., “Asymptotically Optimal Quantizers”, IEEE Trans. on Information Theory, 23, 143 (1977).
- [104] P. L. Zador, “Asymptotic Quantization Error of Continuous Signals and the Quantization Dimension”, IEEE Trans. on Information Theory, 28, 139 (1982).
- [105] J. A. Bucklew, “Upper Bounds to the Asymptotic Performance of Block Quantizers”, IEEE Trans. on Information Theory, 27, 577 (1981).
- [106] A. Gersho, “Asymptotically Optimal Block Quantization”, IEEE Trans. on Information Theory, 25, 373 (1979).
- [107] J. H. Conway, and N. J. A. Sloane, “A Lower Bound on the Average Error of Vector Quantizers”, IEEE Trans. on Information Theory, 31, 106 (1985).
- [108] D. Pollard, “Quantization and the Method of k-Means”, IEEE Trans. on Information Theory, 28, 199 (1982).
- [109] M. J. Sabin, and R. M. Gray, “Global Convergence and Empirical Consistency of the Generalized Lloyd Algorithm”, IEEE Trans. on Information Theory, 32, 148 (1986).
- [110] R. M. Gray, and E. D. Karnin, “Multiple Local Optima in Vector Quantizers”, IEEE Trans. on Information Theory, 28, 256 (1982).

- [111] M. Rabbani, and P. W. Jones, Digital Image Compression Techniques, SPIE Optical Engineering Press (1991).
- [112] W. H. Equitz, "A New Vector Quantization Clustering Algorithm", IEEE Trans. on Acoustic, Speech, and Signal Processing, 37, 1568 (1989).
- [113] R. L. Bottemiller, "Comments on "A New Vector Quantization Clustering Algorithm"", IEEE Trans. on Signal Processing, 40, 455 (1992).
- [114] J. Ward, "Hierarchical Grouping to Optimize an Objective Function", J. Amer. Stat. Assoc., 58, 236 (1963).
- [115] T. Kohonen, Self-Organization and Associative Memory, Springer-Verlag (1989).
- [116] E. Yair, K. Zeger, and A. Gersho, "Competitive Learning and Soft Competition for Vector Quantizer Design", IEEE Trans. on Signal Processing, 40, 294 (1992).
- [117] P. J. M. Laarhoven and E. H. L. Aarts, Simulated Annealing: Theory and Applications, D. Reidel (1987).
- [118] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, 220, 671 (1983).
- [119] A. E. Cetin, and V. Weerackody, "Design of Vector Quantizers Using Simulated Annealing", IEEE Trans. on Circuits and Systems, 35, 1550 (1988).
- [120] K. Zeger, J. Vaisey, and A. Gersho, "Globally Optimal Vector Quantizer Design by Stochastic Relaxation", IEEE Trans. on Signal Processing, 40, 310 (1992).
- [121] C.-M. Huang, and R. W. Harris, "A Comparison of Several Vector Quantization Codebook Generation Approaches", IEEE Trans. on Image Processing, 2, 108 (1993).
- [122] M. A. Turker, and M. Severcan, "Intraframe Coding with DCT-VQ for Image Sequence Compression", Proc. Mediterranean Electronics Conf., April 12-14, Antalya, Turkey, I-238 (1994).

- [123] J. I. Gimlett, "Use of Activity Classes in Adaptive Transform Image Coding", IEEE Trans. on Communications, **23**, 785 (1975).
- [124] D. L. McLaren, and D. T. Nguyen, "Activity Function for DCT Coded Images", Electronics Letters, **25**, 1704 (1989).
- [125] J. W. Kim, and S. U. Lee, "Discrete Cosine Transform - Classified VQ Technique for Image Coding", Proc. Intl. Conf. on Acoust., Speech, and Signal Process., May 23-26, Glasgow, Scotland, 1831 (1989).
- [126] B. Ramamurthi, and A. Gersho, "Classified Vector Quantization of Images", IEEE Trans. on Communications, **34**, 1105 (1986).
- [127] D. S. Kim, and S. U. Lee, "Image Vector Quantizer Based on a Classification in the DCT Domain", IEEE Trans on Communications, **39**, 549 (1991).
- [128] E. A. Riskin, "Optimal Bit Allocation via Generalized BFOS Algorithm", IEEE Trans. on Information Theory, **37**, 400 (1991).
- [129] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees, The Wadsworth Statistics/Probability Series (1984).
- [130] M. A. Turker, and M. Severcan, "Adaptive Intraframe Coding for Very Low Bit-Rate Image Sequence Compression Using Classifier Constrained DCT-VQ", Proc. Data Compression Conf., March 29-31, Utah, USA, 452 (1994).
- [131] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Entropy-Constrained Vector Quantization", IEEE Trans. on Acoustics, Speech, and Signal Processing, **37**, 31 (1989).
- [132] M. A. Turker, and M. Severcan, "A Classifier Constrained VQ Algorithm for the Compression of Image Sequence Frames in DCT Domain", Proc. European Signal Processing Conf., September 13-16, Edinburgh, Scotland, to appear (1994).
- [133] D. A. Huffman, "A Method for Construction of Minimum Redundancy Codes", Proceedings of the IRE, **40**, 1098 (1952).
- [134] M. Goldberg, H. Sun, "Image Sequence Coding Using Vector Quantization", IEEE Trans. on Communications, **34**, 703 (1986).

- [135] M. Goldberg, H. Sun, "Frame Adaptive Vector Quantization for Image Sequence Coding", IEEE Trans. on Communications, 36, 629 (1988).
- [136] H. Sun, and C. N. Manikopoulos "Vector Quantization with Replenishment Technique for Video Signal Coding", SPIE Visual Communications and Image Processing IV, 595 (1989).
- [137] S. Panchanathan, and M. Goldberg, "Adaptive Algorithms for Image Coding Using Vector Quantization", Signal Processing: Image Communication, 4, 81 (1991).





APPENDICES

APPENDIX A

ILLUSTRATION OF VISUAL ANGLE

Quantitatively, the visual angle is calculated as follows :

$$\tan A = \frac{S_o}{d_o} \quad (\text{A.1})$$

where S_o is the height of the object, d_o is the viewing distance, and A is the visual angle subtended at the viewing distance (see Fig. A.1). Generally, normal viewing distance in image coding is assumed to be four times the picture height as the pictures are displayed in monitors.

The visual angle is measured in terms of degrees. Conversions to other measures are as follows :

$$1^\circ \text{ (degree)} = 60' \text{ (minutes of arc)} \quad (\text{A.2})$$

$$1' = 60'' \text{ (seconds of arc)} \quad (\text{A.3})$$

As an example, visual angle of the Sun which is about 93 million miles away is $30'$, while visual angle of the Moon is same as that although it is 240 thousand miles away.

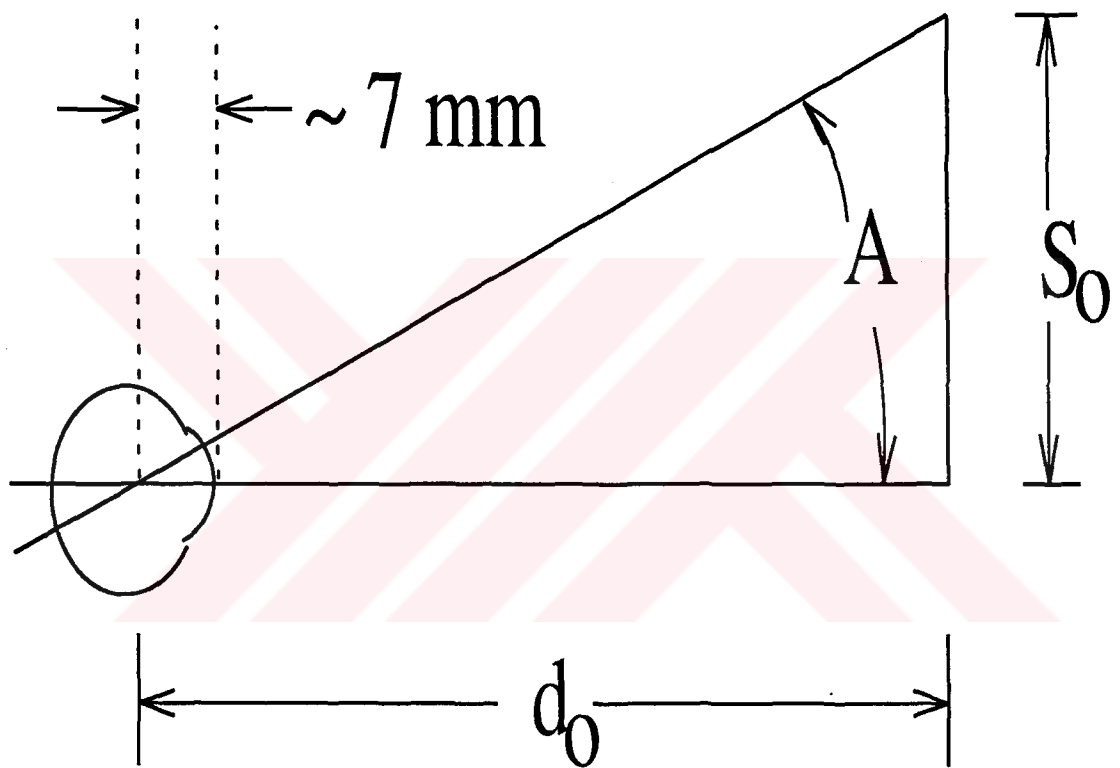


Figure A.1. Visual Angle

APPENDIX B

PERFORMANCE MEASURES

In image coding the best way to assess the quality of a reproduction image or image sequence is of course the inspection of it by a human observer. On the other hand, some quantitative measures of performance of a compression system is also essential, mainly to be able to compare with the results in the literature. Such quantitative measures can be grouped into the following two classes:

1. Objective Measures : These are the statistical evaluations of the error signal, incurred due to compression.
2. Subjective Measures : Measures which, in some way accounts for the human visual system, so that they give numerical representations of the assessment that a human observer would make with bare eyes.

Although there exist many powerful subjective measures and many other objective ones, the choice of this thesis work favours an objective performance measure, namely peak signal-to-noise ratio (PSNR).

Assume that an image frame, which is to be compressed contains $N \times N$ pictorial elements (pixels) which have already been quantized into a certain number of levels and therefore has the value $X(i, j)$ where $i, j = 1, \dots, N$. Average signal power of this input can then be calculated as follows

$$E\{X^2\} = \sigma_x^2 \approx \frac{1}{N \times N} \sum_{i=1}^N \sum_{j=1}^N X^2(i, j). \quad (\text{B.1})$$

When this source is encoded and reconstructed by a compression system each pixel $X(i, j)$ will be reproduced as another discrete value $Y(i, j)$. Then, average least squares error (LSE) power (or average mean square error (MSE) power) of the reconstruction will be

$$E\{|X - Y|^2\} = \sigma_{e, LSE}^2 \approx \frac{1}{N \times N} \sum_{i=1}^N \sum_{j=1}^N |X(i, j) - Y(i, j)|^2. \quad (\text{B.2})$$

Normalized LSE (NLSE) distortion can now be calculated as follows

$$\bar{\sigma}_{e,LSE}^2 = \frac{\sigma_{e,LSE}^2}{\sigma_x^2}. \quad (\text{B.3})$$

SNR of this system is defined as follows [4]

$$SNR \triangleq 10 \log_{10} \frac{1}{\bar{\sigma}_{e,LSE}^2}, \quad (\text{B.4})$$

and Peak SNR (PSNR) of it is defined as follows

$$PSNR \triangleq 10 \log_{10} \frac{|X_{min} - X_{max}|^2}{\sigma_{e,LSE}^2}. \quad (\text{B.5})$$

For typical images represented with 8 bits per pixel, $X_{min} = 0$ and $X_{max} = 255$. PSNR performance of a system is usually 12-15 dB higher than SNR of it.

Since the quality assessment for most of the compression systems that one can find in the literature is made using PSNR, PSNR will be the performance measure throughout this thesis work.