IMAGE CODING EMPLOYING VECTOR QUANTISATION

by

Aharon H. Kubrick

A Thesis submitted for the degree of
Doctor of Philosophy

City University
School of Engineering
Information Engineering Centre
London
March 1993

Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor, Dr. Tim Ellis for the continual guidance and encouragement that he provided throughout the course of this research, and most of all for his friendship. During my two years stay in London, Janet and Tim Ellis have been most kind and generous to me and I owe them an unrepayable debt.

In addition, I am indebted to Professor A. C. Davies of King's College London who encouraged me to start this research, and who took interest in my work throughout the course of this research. Rochelle and Tony Davies have been most kind friends.

I wish to thank Professor L. Finkelstein, the Dean of the School of Electrical Engineering and Applied Physics, for his friendly hospitality and his kind attention.

Many thanks to the staff at the Information Engineering Centre, especially Professor M. Cripps, Dr. P. Samwell, Dr. R. Comlley, Dr. S. Khan, and Mrs. S. Gilling for their friendly hospitality.

Many thanks to my colleagues, especially J. Ngwa-Ndifor, S. Omarouayache, R. Hung, L. Tun, S. Brock-Gunn, S. Mylonas, and A. Purbenyamin for the nice atmosphere that existed during my stay with them. Many thanks also to all in the Computer unit, especially D. Tongue, and A. Lack for their help.

Last but not least, I thank my wife Katy and my daughter Liat for having always encouraged me and for trying to understand my strange way of thinking.

DEDICATION

TO

MY MOTHER

Who sacrificed so much.

KATY

Who surely knows why.

LIAT

Who hopefully will find the answers.

DECLARATION

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

ABSTRACT

The work described in this thesis is concerned with the coding of digitised images employing vector quantisation (VQ). A new VQ-based coding system, named *Directional Classified Gain-Shape Vector Quantisation* (DCGSVQ), has been developed. It combines vector quantisation with transform coding techniques and exploits various properties of the human visual system (HVS) like frequency sensitivity, the masking effect, and orientation sensitivity, to produce reconstructed images with good subjective quality at low bit rates (0.48 bit per pixel).

A *content classifier*, operating in the spatial domain, is employed to classify each image block of 8x8 pixels into one of several classes which represent various image patterns (edges in various directions, monotone areas, complex texture, etc.). Then a classified gain-shape vector quantiser is employed in the cosine domain to encode vectors of AC transform coefficients, while using either a scalar quantiser or a gain-shape vector quantiser to encode the DC coefficients. A new vector configuration strategy for defining AC vectors in the cosine domain has been proposed to better adapt the system to the local statistics of the image blocks. Accordingly, the AC coefficients are first weighted by an equivalent modulation transfer function (MTF) that represents the filtering characteristics of the HVS, and then they are grouped into *directional vectors* according to their direction in the cosine domain. An optional simple method for feature enhancement, based on inherent properties of the proposed strategy, has also been proposed enabling further image processing at the receiver.

A new algorithm for designing the various DCGSVQ codebooks has been developed in two steps. First, a general-purpose new algorithm for classified VQ (CVQ) codebook design has been developed as an alternative to empirical methods proposed in the literature. The new algorithm provides a simple and systematic method for codebook design and reduces considerably the total number of mathematical operations during codebook design. We have named this new algorithm *Classified Nearest Neighbour Clustering* (CNNC). A fast search algorithm has also been developed to reduce further computational efforts during codebook design.

Secondly, a new optimisation criterion which is more suitable for shape codebook design has been developed and employed within the CNNC algorithm to design classified shape codebooks for the DCGSVQ. We have named this algorithm *modified CNNC*. The new algorithm designs the various shape codebooks simultaneously giving the designer full freedom to assign more importance to certain classes of vectors or to certain training vectors. The DCGSVQ system has been shown to outperform the full search VQ, the CVQ, and the transform coding CVQ (TC-CVQ) producing nicer coded images with better signal to noise ratio (SNR) figures at various bit rates.

To improve further the perceived quality of coded images, a new postprocessing algorithm that can be applied at the decoder without increasing the bit rate has been developed. The proposed algorithm is based on various characteristics of the signal spectrum and the noise spectrum, and exploits various properties of the HVS. The proposed algorithm is a general-purpose algorithm that can be applied to block-coded images produced by various systems like VQ, transform coding (TC), and Block Truncation Coding (BTC). The algorithm is modular and can be applied in an adaptive way depending on the quality of the block-coded image.

The last theme of this work has been the identification of useful fidelity criteria for image quality assessment. Quality predictors in the form of some subjectively weighted error measures were sought such that a smooth functional relationship exists between them and quality ratings made by human viewers. Quality predictors that incorporate simplified models of the HVS have been proposed and tested on a large set of VQ-coded images. Two such predictors have been shown to be better suited for image quality assessment than the commonly used mean square error (MSE) measure.

NOTATIONS

AC	Alternating current.
A_i	The average partial distortion per codevector in class i.
$\bar{a}_i$	The ith shape vector.
BPF	Band-pass filter.
bpp	Bits per pixel.
B_t	The number of bits used to code a vector at time t.
BTC	Block truncation coding.
cm	Centimetre.
cpd	Cycles per degree.
CVQ	Classified vector quantiser (quantisation).
CNNC	Classified Nearest Neighbour Clustering.
C	A codebook.
C^*	The optimal codebook.
C_i	The ith codebook.
C_s	A shape codebook.
C_g	A gain codebook.
dB	Decibles.
DC	Direct current.
DCT	Discrete cosine transform.
deg	Degrees.
$d(\bar{x}, \bar{y})$	A distortion measure between two vectors.
D	The overall average distortion.
D_i	The average distortion in class i.
D_i^*	The optimal distortion in class i (minimum distortion).
DBS	Direct broadcast satellite.
DCGSVQ	Directional classified gain-shape vector quantiser (quantisation).
$E[\cdot]$	Mathematical expectation.
F_c	The number of codewords transmitted per second.
$f(i, j)$	The image signal in the spatial domain.
$F(u, v)$	The image signal in the cosine domain.
f_x, f_y	The spatial frequency coordinates in Fourier domain.
f_u, f_v	The spatial frequency coordinates in cosine domain.
GSVQ	Gain-shape vector quantiser (quantisation).
GLA	Generalised Lloyd's algorithm.
HVS	Human visual system.
$H(\omega)$	The equivalent modulation transfer function of the HVS.
$I(x, y)$	The intensity of an image at the position (x, y).
k	The dimension of a vector.
KL	Karhunen-Loeve transform.
LBG	Linde, Buzo and Gray algorithm.
LPC	Linear predictive coding.
LPF	Low-pass filter.

M	The number of classes or codebooks.
Mb/Sec	Megabits per second.
ME	Merging error.
m_i	The number of vectors contained in region S_i or in class i.
mL	A measure of the quantity of light, millilumens.
mm	Millimetres.
MSE	Mean square error.
MTF	Modulation transfer function.
N	The number of codevectors contained in a VQ codebbok.
N_i	The number of codevectors assigned to class i.
N_i^*	The optimal number of codevectors assigned to class i.
$\{N_i^1\}$	Initial allocation of codevectors to class i.
n_t	The size of the training set.
$p(\cdot)$	Probability density function.
pixels	picture elements.
P_i	The probability of a vector being in the ith class.
$\Pr(\cdot)$	Discrete probability.
$q(\cdot)$	The quantisation operator.
R	The number of bits needed to encode one codevector.
r	rate (R/k).
ρ	The training ration (n_t/N).
R_{ave}	The average codeword length.
R^k	The k-dimensional Euclidean space.
SA	Simulated annealing.
Sec	Seconds.
S_i	A region defined in the k-dimensional space.
SNR	Signal to noise ratio.
T	Transmission rate.
TC	Transform Coding.
TV	Television.
$\bar{v}_1, \bar{v}_2$	Directional vectors.
VLSI	Very large scale integrated circuits.
V_{ptp}	The maximum intensity value (255 for 8 bpp).
VQ	Vector quantisation or vector quantiser.
$W(x, y)$	The value of a weighting function at pixel (x, y).
$\bar{x}$	A vector.
$\bar{x}_t$	The vector $\bar{x}$ as observed at time t.
$\hat{x}$	The reproduction vector (decoded vector).
$\bar{X}_i$	The centroid of class i.
$\bar{y}_i$	The ith codevector.
σ_j	The jth gain value.
$(\cdot)^T$	Vector or matrix transpose.
$\emptyset$	The empty set.
$*$	Convolution.
$\parallel \bar{x} \parallel$	The *norm* of a vector (its magnitude).

Chapter 1

Introduction

Teaching beyond teaching;[1]
No leaning on words and letters.

1.1 Background

Digital image processing has a broad spectrum of applications, such as remote sensing via satellites, image transmission and storage for business and military applications, medical processing, radar, sonar and acoustic image processing, robotics and automated inspection of industrial parts. Images acquired by satellites are useful in tracking of earth resources, geographical mapping, prediction of agricultural crops, urban growth, weather prediction, and other environmental applications.

Image transmission and storage applications include broadcast television, teleconferencing, videophone, transmission of facsimile images for office automation, communication over computer networks, closed-circuit television based security monitoring systems and military communications. Current forecasts for world-wide communications in the '90s and beyond, point to a proliferation of digital transmission as a dominant means of communication for voice and imagery data. Digital transmission is expected to provide flexibility, reliability, and cost effectiveness with the added potential for communication privacy and security through encryption.

[1]Each chapter starts with a quotation from the *Zenrin Kushu* - a Zen saying anthology

A new development which is attracting much interest is the provision of color television pictures of significantly better quality, allowing the use of appreciably bigger screens. The availability of new transmission channels, such as via direct broadcast satellite (DBS) and wide-band fiber optic cable systems, has led to an upsurge of interest in new TV transmission methods such as high definition TV (HDTV). Unfortunately, the enormous data generated by such systems can defeat any transmission media, including fiber optic cables.

Medical applications of digital image processing include the processing of X-rays, ultrasonic, computer tomography (CT), and nuclear magnetic resonance (NMR) images. New archival systems are needed to store these digital images with as little as possible loss of fine details. Similar demands exist for radar and sonar images which are used for detection and recognition of various types of targets, or in guidance and maneuvering of aircraft or missile systems.

Many new applications in video technology will be digital. In the computing arena, interest is already growing in the integration of video with graphics and audio. Often referred to as *multimedia* applications, these will be used in interactive education, next-generation graphics systems, network videoconferencing, and other user-friendly systems.

Unfortunately, all these applications share in common the increasing demand for digital storage and transmission media. Unlike the digital audio technology of the '80s, sampled video source signals require very high bit rates ranging from about 97 megabits per second (Mb/Sec) for broadcast-quality video to 216 Mb/Sec for studio digital signal processing. Even when still pictures (images) are involved, as in image archival systems, a mountain of data is needed to represent them. For example, a color image with resolution of 1000 by 1000 picture elements (*pixels*) at 24 bits each will occupy 3 megabytes of storage.

The costs of digital storage and transmission media are generally proportional to the amount of digital data that can be stored or transmitted. While the cost of such media decreases every year, the demand for their use increases at an even higher rate. Therefore, there is a continuing need to minimise the number of bits necessary to transmit or store signals while maintaining acceptable signal fidelity or quality. The branch of electrical engineering that deals with this problem is termed *data compression* or *coding.*

The theoretical foundations of data compression lie in a branch of information theory known as *rate-distortion theory*, originally set forth by Shannon [Shannon 1948, Berger 1971]. An important aspect of data compression is the *quantisation* process defined as the process of approximating continuous-amplitude signals by digital (discrete-amplitude) signals. The independent quantisation of each signal value or parameter is named *scalar quantisation*, while the joint quantisation of a block of parameters is named *block quantisation.* A fundamental result in information theory indicates that for any source of data more efficient coding can be achieved by processing carried out in block (vector), rather than scalar, form. This fundamental result and the increasing availability of high speed integrated circuitry for implementation have stimulated the growing interest over the past ten years in a new method for speech and image coding - *Vector Quantisation* (VQ).

In VQ [2], a *vector* always refers to a k-dimensional ordered set of real numbers. The vector components may represent signal samples or numerical values of certain parameters or features that have been extracted from a signal. In the most direct application of VQ to waveform or image compression, a group of contiguous signal samples is grouped into a vector so that each vector simply describes a small segment of the original signal. This leads to efficient exploitation of the correlation between samples within an individual vector.

VQ is actually a *pattern matching* technique. In essence, the vector of input samples is a *pattern* that must be approximated by one of a finite set of prototype patterns that is stored in a *codebook.* To describe this pattern, we simply identify the *index* (address) of that pattern in the codebook that "best" approximates the input pattern. Efficient coding is achieved by replacing the original vector with the address of the selected pattern.

Image coding employing VQ consists of the following basic steps. First, image vectors are usually formed by dividing a digitised image into contiguous, non-overlapping, square blocks of pixels which are arranged as k-dimensional vectors.

[2] This abbreviation is used to denote either *vector quantisation* or *vector quantiser* as determined by context.

Then, for each input image vector, the *encoder* finds a "nearest" codevector from a predefined codebook according to some meaningful fidelity criterion, or distortion measure, which assigns a "cost" to any such mapping. A binary codeword is then used to identify the index of the chosen codevector and this codeword is transmitted to the *decoder*. The decoder simply looks up the appropriate codevector from a copy of the codebook and outputs it as the reproduction vector. Finally, the entire reconstructed image is built-up by rearranging the reconstructed image blocks in appropriate order. The system described above is a *memoryless full search* VQ since each vector is encoded individually, and all the codevectors are tested for each input vector.

VQ falls in the generic group of block coding techniques, such as transform coding (TC) and block truncation coding (BTC), all of which code a block of source samples as one entity. In theory, VQ approaches the optimal compression limit (the rate-distortion limit) as the dimension of the vector increases. Furthermore, for any coding method based on vectors of the same dimensionality, a VQ that performs as well or better at the same rate can be designed [Ramamurthi 1985]. Thus, VQ is the optimal coding method for a given dimensionality.

One unique feature of VQ is that high compression ratios are possible with relatively small block sizes, unlike other compression methods such as transform coding. Use of smaller block sizes (4x4 or 5x5 pixels) in block coding has been known to lead to better subjective quality [Limb 1979]. A second feature of VQ is that, unlike TC or BTC, the decoding process is simply a table look-up, making VQ ideally suited for single-encoder, multiple-decoder applications such as videotext and archiving where the decoder should be as simple and cheap as possible.

Coders based on memoryless VQ have achieved reasonable quality at rates above 0.6 bits per pixel (bpp). To encode images below 0.6 bpp without any significant degradation in quality, it is necessary to increase the block size so that a single block can exploit spatial correlation over a larger region. It has been shown, however, that the block size is typically limited to 5x5 pixels or even smaller by the encoding and storage complexity of VQ. It has also been noted that edge quality is harder to maintain with larger block sizes

[Ramamurthi 1986]. One technique that has achieved lower bit rates is the combination of VQ with transform coding, as in [Saito 1988] and in [Kim 1989]. A linear transformation is performed on a relatively large block size (8x8 or 16x16 pixels), and VQ is then employed on groups of transform coefficients. Good visual quality of coded images has been obtained at bit rates in the range of 0.45 to 0.65 bpp. Further compression requires that a coder exploits the two-dimensional spatial correlation between neighbouring blocks in the image. Such coders are, in general, more complex, and their design is somewhat heuristic.

1.2 Purpose

Various VQ systems have been developed during the last ten years. While the effectiveness and importance of VQ in data compression and in speech recognition is undisputed, a major obstacle to its use is the computational complexity of real-time implementation. In a VQ system, the *rate* is defined as the number of bits needed to identify each codevector. The rate r in bits per vector component is given by $r = (1/k)\log_2 N$ where k is the vector's dimension and N is the number of codevectors. Thus, for a fixed k value and an increasing rate or for a fixed rate and an increasing k value, the codebook size, and therefore encoding complexity, grows exponentially. This growth requires a corresponding increase in storage requirements and computational effort.

Studies of image coding with vector quantisation have revealed that VQ systems, working at rates of 0.5 bpp or less, produce coded images with reduced resolution as well as reduced *edge integrity.* Reduced edge integrity refers to the inaccurate reproduction of an edge in terms of orientation and location. We have noted that, in general, the non-edge areas of the coded images are coded without visually annoying degradation; however, edges are coded very poorly. The cause of this problem is the small number of codevectors, within the codebook, that contain parts of edges in them thus being unable to represent the variety of edges that must be coded. Even if the codebook contains edge codevectors, the distortion measure usually used for encoding does not ensure that an edge block will be coded with an edge codevector.

The very end of many image processing system is the human eye. The sensitivity of the human visual system (HVS) to stimuli of varying levels of contrast, luminance and different spatial and temporal frequencies varies greatly. These inconsistencies can be exploited to determine the *subjective redundancies* within the image, i.e., that information which can be discarded without subjectively degrading the perceived quality of the final image [McLaren 1991]. That is why many researchers have studied the HVS in order to evaluate the effects of different types of distortion in coded images on the human observer.

In the light of this discussion, the main purpose of this research is to develop a new VQ-based coding system that integrates valuable knowledge about the human visual system with emerging VQ techniques. It is postulated that if the coding scheme is matched to the HVS and attempts to imitate its functions, at least for the known part of it. high compression ratios along with good quality of reconstructed images can be expected. The new system should be computationally less demanding, compared to a full search VQ, and should perform better than, or as well as, a full search VQ. In addition, the new system should be capable of preserving edge integrity while operating at low bit rates.

In order to limit the scope of this study, we will consider only monochrome, still images. We exclude the possibility of exploiting temporal correlation present in a sequence of image frames. We also do not exploit the correlation present between the color components in color images. These correlations are significant, and must certainly be included in a future coding system once the basic coding technique is fully understood. Nevertheless, it should be noted that the coder developed herein for still images can easily be enhanced to code a sequence of color images. Therefore, the results of this study will be highly relevant to such future work.

A *memoryless VQ* encodes each image block independently of its neighbours. Such block coders may preserve continuity of tone within each block, but they can not ensure continuity from block to block. As a result, there is a tendency for the decoded intensity to change rather abruptly from one block to another making the block boundaries visible. This problem, which is a common problem of memoryless block coding schemes, and the reduction in edge integrity, as described above, can be dealt with by postprocessing of reconstructed images.

Typically, with postprocessing we attempt to remove the out-of-band noise introduced by the quantisation process. The processing algorithm does not require side information to be transmitted from the encoder and hence the bit rate is not affected. Thus, another goal of this research is to develop an efficient post-filtering algorithm intended to reduce noise in reconstructed block-coded images without significantly modifying the image signal.

Identification of useful fidelity criteria for image compression system design and for image quality assessment has been a persistent difficulty for researchers. Ideally, the distortion measure that is used for codebook design and quantisation should be *tractable* so that it can be analyzed, *computable* so that it can be evaluated in real time and used in minimum distortion systems, and *subjectively relevant* so that large or small distortion values correlate with bad and good subjective quality, respectively. Most distortion measures in use today are certainly tractable and, to some extent, subjectively relevant (as, for example, the Itakura-Saito distortion used in voice coding [Itakura 1968]). However, as bit rate decreases and distortion increases, simple distortion measures have not always correlated well with subjective judgements. Since VQ is expected to be especially useful at low bit rates, we intend to test a number of perceptually-based distortion measures and find one that is in good accord with subjective assessments of block-coded images.

1.3 Outline of Thesis

The thesis is organised as follows. In chapter 2, we introduce the basic concepts of vector quantisation. Basic vector quantisers that are relevant to our work, i.e., those systems that have been used as basic building blocks in our work toward the development of a new perceptually-based VQ system, are described. In particular we focus on the full-search VQ along with gain-shape VQ (GSVQ), and classified VQ (CVQ). A new algorithm for designing CVQ codebooks is also described and shown to provide a simple and systematic method for codebook design. We have named this algorithm *Classified Nearest Neighbour Clustering* (CNNC).

A review of the human visual system (HVS) is presented in chapter 3. Various properties of the HVS which can be used in image processing and coding are described. Armed with this knowledge about the HVS and the basic VQ systems, we propose a new image coding system in chapter 4. We have named this system *Directional Classified Gain-Shape Vector Quantisation* (DCGSVQ). Various properties of the HVS are incorporated into the proposed system to improve further the subjective quality of reconstructed images. A new algorithm for designing the various codebooks, needed for the DCGSVQ, is also proposed based on the mentioned CNNC algorithm. We have named this new algorithm the *modified CNNC algorithm.*

A new postprocessing algorithm aimed at reducing noise introduced in block-coded images while retaining edge integrity and edge sharpness is proposed in chapter 5. The proposed algorithm is based on various characteristics of the signal spectrum and the noise spectrum. In addition, various properties of the HVS have been incorporated in the filtering algorithm in order to improve the perceived quality of the processed block-coded images.

Chapter 6 is concerned with the evaluation of several subjectively relevant distortion measures. Quality predictors, in the form of some subjectively weighted error measure, are sought such that a smooth functional relationship exists between them and quality ratings made by a number of human viewers. Based on simplified models of the HVS, such quality predictors are proposed and tested on a large set of VQ-coded images.

Finally, in chapter 7, we review the thesis and draw conclusions from our findings. In addition, a short discussion of directions for future research is presented. We have tried to keep the thesis as readable as possible, therefore most of the mathematical developments and the detailed description of the various algorithms have been gathered in the eight appendices that conclude this thesis.

Chapter 2

Basic Vector Quantisers

The jewel remains clean in the mud;
The pine keeps its color after the snow.

Over the last ten years various VQ-based coding systems have been designed and applied to digital image coding at low bit rates (in the range of 0.3 to 1.5 bits per pixel (bpp)). Excellent reviews of VQ systems, their applications, and codebook design algorithms can be found in [Gray 1984] and [Nasrabadi 1988]. In addition, a new book devoted to VQ and signal compression [Gersho 1992] has been published recently. Therefore, a full review of the various available VQ systems is omitted in this thesis. Instead, we will discuss those basic systems that are relevant to our work, i.e., those systems that have been used as basic building blocks in our work toward the development of a new perceptually-based VQ system. In particular, the basic full-search VQ along with gain-shape VQ (GSVQ) and classified VQ (CVQ) are described in this chapter. A new algorithm for designing CVQ codebooks is also described and shown to provide a simple and systematic method for codebook design.

2.1 Full Search VQ

Consider a k-dimensional vector $\bar{x} = (x_1, x_2, \ldots, x_k)$ whose components $\{x_l,\ 1 \leq l \leq k\}$ are real-valued, continuous-amplitude random variables. In vector quantisation, the vector $\bar{x}$ is mapped onto another real-valued, discrete-amplitude k-dimensional vector $\bar{y}$. We say that $\bar{x}$ is quantised as $\bar{y}$, and $\bar{y}$ is the quantised value of $\bar{x}$. The following notation is used

$$\bar{y} = q(\bar{x})$$

where $q(\cdot)$ is the quantisation operator. The vector $\bar{y}$ is called the *reproduction vector* or the *output vector* corresponding to $\bar{x}$. Typically, $\bar{y}$ takes on one of a finite set of values $C = \{\bar{y}_i,\ 1 \leq i \leq N\}$ where $\bar{y}_i = (y_{i1}, y_{i2}, \ldots, y_{ik})$. The set C is referred to as the *reconstruction codebook*, or simply the *codebook*. N is the *size* of the codebook, and $\{\bar{y}_i\}$ is the set of *codevectors*. The size N of the codebook is also called the *number of levels*, a term borrowed from scalar quantisation terminology. Thus, a quantiser that employs N codevectors is referred to as an N-level quantiser.

To design a VQ codebook, the k-dimensional space, R^k, of the random vector $\bar{x}$ is partitioned into N regions or *cells* $\{S_i,\ 1 \leq i \leq N\}$ and each cell S_i is associated with a vector $\bar{y}_i$. The quantiser then assigns the codevector $\bar{y}_i$ if $\bar{x}$ is in S_i, i.e.,

$$q(\bar{x}) = \bar{y}_i, \quad \text{if } \bar{x} \in S_i \ . \tag{2.1}$$

This process of codebook design is known as *training* or *populating* the codebook.

When $\bar{x}$ is quantised as $\bar{y}$, a quantisation error results and a *distortion measure* $d(\bar{x}, \bar{y})$ can be defined between $\bar{x}$ and $\bar{y}$. As the vectors $\bar{y}_t$ at different times t are transmitted, one can define an overall average distortion

$$D = \lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} d(\bar{x}_t, \bar{y}_t) \ . \tag{2.2}$$

If the vector process $\bar{x}_t$ is stationary and ergodic then the sample average in equation (2.2) tends in the limit to the mathematical expectation

$$\begin{aligned} D &= E[d(\bar{x}_t, \bar{y}_t)] = \\ &= \sum_{i=1}^{N} \int_{S_i} d(\bar{x}_t, \bar{y}_i) p(\bar{x}_t) d\bar{x}_t = \\ &= \sum_{i=1}^{N} \Pr(\bar{x}_t \in S_i) E[d(\bar{x}_t, \bar{y}_i) \mid \bar{x}_t \in S_i] \end{aligned} \tag{2.3}$$

where $\Pr(\bar{x}_t \in S_i)$ is the discrete probability that $\bar{x}_t$ is in S_i, $p(\bar{x}_t)$ is the multi-dimensional probability density function (pdf) of $\bar{x}_t$, and the integral is taken over all the components of the vector $\bar{x}_t$.

The quantiser is said to be an optimal (minimum-distortion) quantiser if the distortion in equation (2.3) is minimised over all N-level quantisers. Two necessary conditions for optimality have been defined in [Gray 1980] :

1. Given a codebook $C = \{\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_N\}$, the optimal quantiser is realized by using a minimum-distortion or *nearest neighbour* selection rule

   $$q(\bar{x}) = \bar{y}_i , \quad \text{iff } d(\bar{x}, \bar{y}_i) \leq d(\bar{x}, \bar{y}_l), \quad l \neq i \ ; \ l = 1, 2, \ldots, N \ . \tag{2.4}$$

   That is, the quantiser chooses the codevector that results in the minimum distortion with respect to $\bar{x}$. Equivalently, for a given codebook the decision regions (cells) must be such that

   $$S_i = \{\bar{x} \ : \ d(\bar{x}, \bar{y}_i) \leq d(\bar{x}, \bar{y}_l) \ \text{ for all } \ l \neq i \ ; \ l = 1, 2, \ldots, N\} \ .$$

   These regions are called *Voronoi cells* or *Dirichlet regions* [Gersho 1982a]. The regions S_i partition the k-dimensional Euclidean space, i.e.,

   $$\cup_{i=1}^{N} S_i = R^k \quad \text{and} \quad S_i \cap S_l = \emptyset \quad \text{for} \quad i \neq l$$

   for an arbitrary tie-breaking rule.

2. Given a partition of the k-dimensional space, $\{S_i\}_{i=1}^{N}$, each codevector $\bar{y}_i$ is chosen to minimise the average distortion in cell S_i. That is, $\bar{y}_i$ is that vector $\bar{u} \in S_i$ which minimises

   $$D_i = E[d(\bar{x}, \bar{u}) \mid \bar{x} \in S_i] \ . \tag{2.5}$$

   We call such a vector the *centroid* of the cell S_i, and we write

   $$\bar{y}_i = \text{cent}(S_i) \ .$$

In practice, we are given a set of *training vectors* $\{\bar{x}_t \ ; \ t = 1, 2, \ldots\}$ that define an empirical (discrete) probability density function instead of the actual density function of the data source which is usually unknown. A subset of those vectors will be in cell S_i thus, the average distortion in that cell can be computed by

$$D_i = \frac{1}{m_i} \sum_{\bar{x} \in S_i} d(\bar{x}, \bar{y}_i) \ , \tag{2.6}$$

and the centroid can be computed by

$$\bar{y}_i = \min_{\bar{u} \in S_i}^{-1} \left[\frac{1}{m_i} \sum_{\bar{x} \in S_i} d(\bar{x}, \bar{u}) \right] \tag{2.7}$$

where m_i is the number of vectors $\bar{x}$ in S_i, and the inverse minimum means that $\bar{y}_i$ is that vector $\bar{u} \in S_i$ that minimises the average distortion in the cell S_i.

The centroid depends on the distortion measure and is computable only for a few distortion measures one of which is the weighted mean square error (WMSE) defined as

$$\text{WMSE} \;=\; \frac{1}{n}\sum_{t=1}^{n}(\bar{x}_t - \bar{y}_t)^T W(\bar{x}_t - \bar{y}_t) \tag{2.8}$$

where $(\cdot)^T$ denotes transpose, W is a $(k \times k)$ positive-definite weighting matrix, and n is the number of vectors $\bar{x}_t$. The mean square error (MSE) is a special case of the WMSE and is defined by

$$\text{MSE} \;=\; \frac{1}{n}\sum_{t=1}^{n}(\bar{x}_t - \bar{y}_t)^T(\bar{x}_t - \bar{y}_t) \;=\; \frac{1}{n}\sum_{t=1}^{n} \| \bar{x}_t - \bar{y}_t \|^2 \;. \tag{2.9}$$

Both distortion measures have been widely used in image coding although they have been shown to correlate poorly with subjective judgements, particularly for low bit rates (see, for example, [Mannos 1974]). One can easily show that for either the WMSE or the MSE the average distortion D_i in cell S_i is minimised by the centroid

$$\bar{y}_i \;=\; \frac{1}{m_i}\sum_{\bar{x}\in S_i} \bar{x} \;. \tag{2.10}$$

That is, $\bar{y}_i$ is simply the sample mean of all the training vectors contained in S_i.

A basic vector quantiser is depicted in Fig. 2.1. The input $\bar{x}$ to the encoder is a k-dimensional vector. In our application, the image is partitioned into contiguous, non-overlapping, square blocks (for example, blocks of 4x4 pixels) and these blocks, rearranged as 16-dimensional vectors, form the input data vectors. The encoder computes the distortion $d(\bar{x}, \bar{y}_i)$ between the input vector $\bar{x}$ and each codevector, $\bar{y}_i$; $i = 1, 2, \ldots, N$ from a codebook C. The optimum encoding rule is the nearest neighbour rule, described earlier, in which the index i is transmitted to the decoder if codevector $\bar{y}_i$ is closest to the input vector $\bar{x}$. The decoder simply looks up the i-th codevector, $\bar{y}_i$, from a copy of the codebook C, and outputs it as the reproduction vector $\hat{x}$. This type of quantisation is known as *full search* vector quantisation since all codevectors are tested for each input vector.

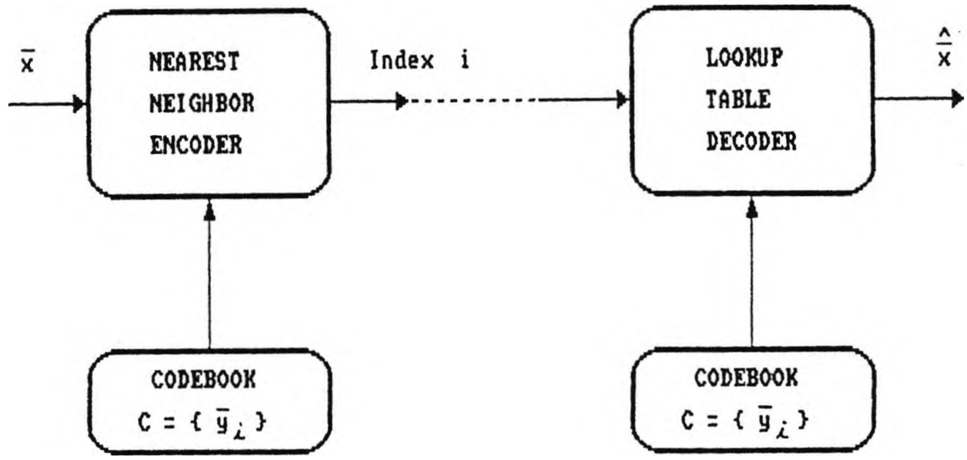


Figure 2.1: *A full search VQ.*

For purposes of transmission, each codevector $\bar{y}_i$ is encoded into a codeword of B_i binary digits (bits). In general, the different codewords may have different lengths (*variable-length coding*) and the *transmission rate* T is then given by

$$T = R_{ave} F_c \quad \text{bits/Sec} \tag{2.11}$$

where

$$R_{ave} = \lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} B_t \quad \text{bits/codeword} \tag{2.12}$$

is the average codeword length, B_t is the number of bits used to code the vector $\bar{y}_t$ at time t, and F_c is the number of codewords transmitted per second. For a codebook of size N and using equal-length codewords, the number of bits needed to encode each vector is

$$R = \log_2 N \ . \tag{2.13}$$

For such a code, the average number of bits per dimension (per vector component) is given by

$$r = \frac{1}{k} R = \frac{1}{k} \log_2 N \quad \text{bits/dimension} \ . \tag{2.14}$$

In designing a data compression system, one attempts to design the quantiser so that the distortion in the output is minimised for a given transmission rate.

2.1.1 Codebook Design

Given an adequate statistical specification of the source and a desired codebook size, the most challenging task in VQ design is to design a codebook that contains the best, or nearly the best, collection of codevectors which efficiently represent the variety of source vectors to be coded. As explained earlier, an optimal VQ is one which employs a codebook C^* that yields the least average distortion among all possible codebooks. The design algorithm for an optimal codebook is not known in general; however, a clustering algorithm, the LBG algorithm also known as the generalised Lloyd's algorithm (GLA) [Linde 1980], has been widely used to design optimal codebooks in a local sense. That is, the codebook designed with the LBG algorithm is optimal for small perturbations of the codevectors, but is not necessarily globally optimal over all possible codebooks [Sabin 1986].

The LBG algorithm uses a long training set of vectors (in our case image vectors) generated by a particular source, and allocates them into a predefined number of clusters (cells). The training image vectors are generated from different training images by dividing each image into small image blocks and arranging each block as a k-dimensional vector. The algorithm starts from a given *initial partition*, or an *initial codebook*, and improves this in a stepwise manner. In this process the training vectors are first assigned to the clusters S_i (represented by the initial codevectors) according to the nearest neighbour rule. Then the codebook is updated by computing the centroid of the training vectors in each cluster. Computing the centroids and reassigning vectors is repeated until the decrease in the overall distortion (between successive iterations) is below a certain threshold. At the conclusion of the algorithm, the centroids are grouped into a codebook, which is best at representing the training set of vectors and hence the particular source.

The main parameters in such iterative codebook design are : the composition of the training set, the size of the training set, n_t, the size of the desired codebook, N, the *training ratio* $\rho = n_t/N$, and the choice of the initial codebook. The composition of the training set is important in that we desire to form a universal codebook that will be able to perform well on all of the data that the coder will ever see. A poorly designed codebook will work satisfactory when the encoding data has the same properties as the training set, but will give poor results when presented with other data.

The size of the training set is important as this roughly determines the number of vectors that will lie in each cluster (cell). For a codebook design to be healthy it is required to have a minimum training ratio ρ of about sixteen [Vaisey 1988]. If the ratio drops below this level then it is likely that the source will not be well represented and that a significant number of clusters will have centroids related to some anomalous characteristic of the training set. This means that these codevectors will be underutilized when coding general images (out of the training set).

The final design parameter is the choice of the initial codebook. Various approaches for initial codebook selection have been developed and several of them are described in [Gray 1984] and [Yuan 1988]. Generally this choice is made based on a random or uniform sampling of the training set. Other methods impose different constraints on the initial codebook, for example, the requirement that the initial codevectors be greater than a minimum distance from each other or that the initial codebook have a minimum number of codevectors with some desired property (such as high detail). One commonly used technique, the *splitting* technique [Linde 1980], has been used in this work and will be briefly described next.

According to the splitting technique, one first finds the optimum 0-rate codebook (one codevector) which is the centroid of the entire training sequence. This single codevector is then slightly perturbed and split to form two codevectors. The LBG Algorithm is then run to get a 1-rate locally optimal codebook that consists of two codevectors. The design continues in this way in stages so that the final codebook of one stage is split to form an initial codebook for the next stage. In that way, new codebooks are designed and improved until the desired number of codevectors is reached. The final codevectors constitute the codebook that is then used for image coding.

The various approaches for initial codebook selection exist because of the high dependence of the final codebook quality on this initial choice. It should be noted that there may be a significant perceptual difference between codebooks designed from different initial conditions even if they appear to perform almost the same in terms of the MSE. In other words, the LBG algorithm easily gets trapped in local minima of the distortion, resulting in a suboptimal codebook [Gray 1982]. Thus, improving the design algorithm by allowing it to "jump out" of local minima will reduce this dependence of the design on the choice of the initial codebook and will enable it to find a solution closer to the global optimum. Simulated Annealing (SA), for example, gives a promising tool in a search for a method to obtain this goal [Vaisey 1988].

The essential idea behind SA is to add randomness to the search for the global minimum of the distortion function, allowing the algorithm to "pull" itself out of local minima. In applying SA to the codebook design problem, a way is sought to perturb a given partition in a simple manner. This task is accomplished by picking a training vector from one cluster and then moving it into another cluster, changing appropriately the centroids of the two clusters involved. Each pick and switch is made with all possible choices being equally likely. If the total distortion decreases then the switch is accepted; however, if the total distortion increases, a probabilistic decision is made whether or not to accept this switch. To reduce the amount of computation needed, the choice of possible cluster assignments for each training vector can be reduced to one of a small group. In addition, the number of switches needed before system equilibrium is declared may be reduced too. Thus, by altering the LBG algorithm with techniques motivated by SA a better codebook may be designed resulting in improved reconstructed images (see [Zeger 1992] for a detailed explanation).

Employing a training set of vectors and the LBG algorithm for codebook design may also result in the following problem. If one examines the partitions associated with the codevectors, obtained at the conclusion of the LBG algorithm, one can notice that a considerable number of codevectors are not fully utilized, that is, they are associated with partitions of negligible size. A more uniform distribution of the training set in the codebook space would result in an improvement in the quality of the coded images. That is, the set of available codevectors would suit better the vectors derived from natural images, rather than to suit the peculiarities of the vectors derived from the training images. In order to obtain this uniform distribution, a merge-and-split algorithm for the codevectors has been developed [Giusto 1990]. It can be called at each LBG step for the twofold purpose of eliminating the codevectors associated with partitions of negligible size (merge), and of replacing them by codevectors associated with new smaller partitions found by subdividing (split) those ones of large size.

There are other techniques for designing VQ codebooks. One such technique is the Kohonen self-organisation feature map [Lippmann 1987], [Ryan 1988]. Kohonen self-organisation feature map is a neural network paradigm which learns a "point estimate" of the distribution of the input data samples. This means that the point density function of the resulting codevectors tends to approximate the probability density function of the input vectors. Kohonen's algorithm creates a VQ codebook by adjusting variable *weight vectors*, also known as *connection weights*, which connect each input node to each of the N output nodes arranged in a two-dimensional grid.

Weights between input and output nodes are initially set to small random values and an input vector is presented. The distance (distortion measure) between the input and each output node is computed and the output node with minimum distance is selected. Once this node is selected, connection weights to it and to other nodes in its neighbourhood are modified to make these nodes more responsive to the current input. This process is repeated for each input vector until eventually the weight vectors converge and become fixed after a *learning gain* (see [Lippmann 1987]), that decreases in time, is reduced to zero. These weight vectors constitute the desired codebook and may be used for image coding.

The adjustment of all weight vectors in a given neighbourhood reduces the problem of unutilized codevectors that often occurs with the LBG algorithm and also forces the weight vectors to conform to the input data distribution so that neighbouring weight vectors (codevectors) are, in general, closer to each other than non-neighbouring codevectors. "Self-organisation" refers to the process of ordering the weight vectors in this way. Kohonen has noted that organisation is expedited by starting with large neighbourhoods then decreasing the size of the neighbourhoods as the process proceeds. The network-based algorithm described above adjusts codevectors with each input training vector, thereby making it possible to train the coding system on-line, continually adjusting it to the incoming data stream.

2.1.2 Drawbacks of Full Search VQ

The critical encoding task of VQ is to find the best matching codevector from the codebook for a given input vector. The search process requires the computation of a distortion measure between the input vector and each of the N codevectors in order to select the most similar codevector for each input vector. Typically this search must be performed for each of a sequence of input vectors presented to the VQ processor at a very high rate. For an N-level quantiser and the MSE, $k2^{kr}$ multiplications, $k2^{kr}$ subtractions, $(k-1)2^{kr}$ additions, and $2^{kr}-1$ comparisons need to be performed for each input vector. The complexity of this computation which grows exponentially with the rate, r, and the vector's dimension, k, and the very demanding throughput requirement are the key obstacles to real-time implementation of VQ for many applications of practical interest.

The *search problem* described above is also encountered during codebook design with the LBG algorithm. Each time, a new set of centroids are calculated, the training vectors are reassigned to the new N clusters defined by these centroids. The assignment process is basically the quantisation process defined in equation (2.4) which is a nearest neighbour search problem. Similar nearest neighbour search problems have also been encountered in pattern-recognition algorithms. Therefore, a number of fast-search algorithms have been proposed in the pattern-recognition literature [Friedman 1975], [Sethi 1981] and more recently in the VQ literature [Cheng 1984], [Chen 1991], which are designed to reduce the computations in a full search system. Most of these algorithms are based on geometrical notions in Euclidean spaces and tend to trade off multiplications with comparisons and with increased storage requirements.

One such simple, but efficient, nearest neighbour search algorithm has been proposed in [Soleymani 1987]. According to this algorithm, the reduction in computational efforts and complexity is achieved by performing a test *prior* to calculating the distortion for a given codevector, thereby avoiding the distortion calculation for those codevectors which fail this test. The mentioned algorithm is described in Appendix A and will be used later on as a basis for the design of a new fast search algorithm developed by us to tackle the search problem encountered during codebook design for the new VQ system, proposed in this research.

The search problem have been tackled from a different angle too. It has been recognised that dedicated hardware processor architectures for VQ have the potential of vastly improving the computational power achievable with current and emerging technologies. Rapid growth in very large scale integrated circuit (VLSI) technology is beginning to close the gap between simulated and real-time algorithms. We can now implement very sophisticated algorithms in a seemingly ever-diminishing area on VLSI chips. Furthermore, the design and VLSI implementation of dedicated, high performance processor architectures is becoming more attractive as chip layout and fabrication procedures are modernized. Therefore, since 1984 significant strides have been taken in the design of real-time VQ-based speech and image coders implemented with VLSI [Davidson 1986]. Systolic architecture concepts have also been applied to hardware VQ realisations, enabling even higher vector throughput rates [Ni 1985].

Another way of reducing computational costs in a very significant manner is by introducing variations on the basic VQ scheme which result in some reduction in performance. Such VQ schemes are, for example, *Tree-searched VQ, Multistep VQ*, and *Product VQ* collectively called *constrained VQ* [Gersho 1992]. Since Product VQs are of special interest to us we pause to define them for later use.

Let $\{C_i\}_{i=1}^{M}$ denote a collection of codebooks each consisting of N_i codevectors of dimension k_i. Then the *product codebook* C is defined as the collection of all $N = \Pi_i N_i$ possible concatenations of M codevectors drawn successively from the M codebooks C_i. The dimension of the product codebook is $k = \Sigma_{i=1}^{M} k_i$, the sum of the dimensions of the component codebooks. The product codebook is denoted mathematically as a Cartesian product :

$$C = \times_{i=1}^{M} C_i = \{(\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_M) \; ; \; \bar{x}_i \in C_i \; ; \; i = 1, 2, \ldots, M\}$$

The search problem encountered during codebook design and encoding, is reduced since the various codebooks are smaller in size and consist of codevectors with reduced dimensionality compared with full search VQ. A product VQ is particularly useful when there are different aspects of the input vector that one might wish to encode separately.

The second problem and most notable is edge degradation. Image blocks that contain parts of edges in them constitute a small fraction of the image. Hence, the training set of image vectors, used for codebook design, is populated with a small fraction of edge vectors. Since in addition, no preferential treatment is given to edges by the distortion measure employed for codebook design (usually the MSE), the edge vectors are poorly represented in the final codebook. As a result, edges are in general poorly coded, appearing jagged in the reconstructed image, a phenomenon referred to in the literature as the *staircase effect* [Gersho 1982]. Since edges are a very important portion of the perceptual information content in an image, the perceived quality of the entire image suffers dramatically. This type of degradation is called *staircase noise* and is known to be highly correlated with the coded signal.

A magnified reconstructed image of 'Lena', coded at 0.5 bpp using a full search VQ, is shown in Fig. 2.2. The staircase noise is visible along diagonal edges such as the boundary line of the girl's shoulder and the black arch in the background. It can be noticed that the intensity change that occurred across an edge in the original image occurs instead at the block boundaries in the coded image, making the block boundaries visible. These block boundaries form the "steps" of the staircase causing the jagged appearance of edges in the coded image. Furthermore, since each block is coded independently of its neighbours (a *memoryless* VQ), continuity of edges across the block boundaries can not be ensured.

The third problem, which is a common problem of block coding schemes, is the noticeable degradation caused near the block boundaries in the coded image. This effect is called the *blocking effect* in the Transform Coding (TC) literature [Reeve 1983] and is particularly objectionable in areas where the intensity changed gradually in the original image. In such areas there is a tendency for the decoded intensity to change rather abruptly from one block to another. In other words, memoryless block coders may preserve continuity of tone within each block, but they can not ensure continuity from block to block.

This type of degradation is called *grid noise* [Ramamurthi 1986a]. It occurs in a correlated fashion along the boundaries of image blocks, and is particularly noticeable with VQ and *Block Truncation Coding* (BTC) [Delp 1979] since the block size is typically small (4x4 pixels). In TC, the block size is larger than the

Figure 2.2: *'Lena' coded by a VQ at 0.5* bpp

typical size in VQ or BTC (typically 8x8 pixels), thus resulting in fewer block boundaries in the coded image making the blocking effect less apparent. The grid noise is visible in Fig. 2.2 especially in monotone areas such as the girl's shoulder, her face, and several areas in the background.

We have described the basic drawbacks of a full search VQ which can roughly be categorised into: (*i*) computational complexity and (*ii*) reduced quality of the reconstructed images. Several techniques have been developed which apply various constraints to the structure of the VQ codebook in order to reduce computational complexity. These methods generally compromise the performance achievable with unconstrained VQ, but often provide very useful and favorable trade-offs between performance and complexity. Constrained VQs can often be designed for larger dimensions and rates, hence quality that is simply not possible for unconstrained VQ becomes practicable. One such system, termed *gain-shape VQ* (GSVQ), is described next as an example of a product VQ.

2.2 Gain-Shape VQ

Gain-shape vector quantisation is a technique where separate, but interdependent, codebooks are used to code the *shape* and *gain* of a given waveform. The *shape* is defined as the original input vector normalised by removal of a *gain* term that is defined appropriately. For example, the gain term may be defined as the energy in a waveform coder or as residual energy in linear predictive voice coding (LPC) [Gray 1984]. The basic notion in GSVQ is that the same pattern of variation in a vector may return with a wide variety of gain values. Thus, it makes sense to handle the dynamic range of the vector separately from the shape of the vector. GSVQ was introduced in [Buzo 1980] where it was termed "gain separation" and used for the case of LPC vocoders. The GSVQ notion was subsequently extended and optimised in [Sabin 1984] where it was applied to waveform coding.

Consider a VQ whose codebook is formed as the Cartesian product of a finite set of vectors and a finite set of scalars. That is,

$$C = C_s \times C_g = \{(\bar{a}_i, \sigma_j),\ i = 1, 2, \ldots, N_1\ ;\ j = 1, 2, \ldots, N_2\}$$

where $C_s = \{\bar{a}_i,\ i = 1, 2, \ldots, N_1\}$ is a set of vectors drawn from R^k, and $C_g = \{\sigma_j,\ j = 1, 2, \ldots, N_2\}$ is a set of scalars drawn from the non-negative real numbers. The codebook C_s is called the *shape codebook* and C_g is called the *gain codebook*.

Like the VQ, the GSVQ can be decomposed into two component operations, the vector *encoder* and the vector *decoder*. The encoder views the input vector $\bar{x}$ and generates the index pair (i, j) of the product codevector specified by $q(\bar{x})$. The decoder uses the index pair to look up the product codevector $(\bar{a}_i, \sigma_j)$. If we define a distortion measure $d(\bar{x}; \bar{a}, \sigma)$ which measures the "cost" associated with reproducing $\bar{x}$ by the pair $(\bar{a}, \sigma)$, then the best mapping $q(\bar{x})$ is one which selects the pair $(\bar{a}_i, \sigma_j)$ that minimises $d(\bar{x}; \bar{a}_i, \sigma_j)$.

In applying GSVQ for waveform coding, we associate a reproduction vector $\hat{x}_{ij}$ with each product codevector $(\bar{a}_i, \sigma_j)$, with $\hat{x}_{ij} = \sigma_j \bar{a}_i$. The distortion between an input vector $\bar{x}$ and $(\bar{a}_i, \sigma_j)$ is then defined as the distortion between $\bar{x}$

and $\hat{x}_{ij}$. Thus, for a distortion measure of the form

$$d(\bar{x}, \hat{x}_{ij}) = (\bar{x} - \hat{x}_{ij})^T(\bar{x} - \hat{x}_{ij}) \tag{2.15}$$

we get

$$d(\bar{x}, \bar{a}_i\sigma_j) = \bar{x}^T\bar{x} - 2\sigma_j\bar{x}^T\bar{a}_i + \sigma_j^2\bar{a}_i^T\bar{a}_i \ . \tag{2.16}$$

If each *shape vector* is normalised in the sense that $\bar{a}_i^T\bar{a}_i = 1$ for $i = 1, 2, \ldots, N_1$, then

$$d(\bar{x}, \bar{a}_i\sigma_j) = \bar{x}^T\bar{x} + \sigma_j^2 - 2\sigma_j\bar{x}^T\bar{a}_i \ . \tag{2.17}$$

The normalisation of the shape codevectors is the key to the shape-gain encoding procedure, since it makes the choice of the optimum shape vector independent of the choice of the gain value. Consequently, the pair $(\bar{a}_i, \sigma_j)$ that minimises Eq. (2.17) can be found in two steps.

1) Select the shape vector $\bar{a}_i$ to maximise $\bar{x}^T\bar{a}_i$.

2) Given $\alpha = \max_i[\bar{x}^T\bar{a}_i]$, select σ_j to minimise $[\sigma_j^2 - 2\sigma_j\alpha]$.

The first step chooses the shape vector that has the highest correlation to the input vector. The second step scalar quantises the value of this correlation in the least squares sense, since minimising $[\sigma_j^2 - 2\sigma_j\alpha]$ is equivalent to minimising $(\sigma_j - \alpha)^2$. It should be noted that the input vector need not be itself gain normalised in order to choose the best shape vector during the first encoding step. Such normalisation would significantly increase the encoder complexity. In addition, the second step can be accomplished by comparing α to a set of ordered thresholds which represent the decision boundaries of a scalar quantiser.

A GSVQ is depicted in Fig. 2.3. The shape encoder stores the vectors $\{\bar{a}_i,\ i = 1, 2, \ldots, N_1\}$ and finds the index i of the most highly correlated vector. The correlation value α is passed to the gain encoder, which stores the decision thresholds between the gain values and outputs the index j of the nearest gain value. The shape decoder and the gain decoder, respectively, look up the indexed shape vector and gain value, and the two outputs are multiplied to produce the reproduction vector $\hat{x}$.

Several algorithms were developed for the design of GSVQ codebooks based on a training sequence of data (see [Sabin 1984] for a detailed description). It

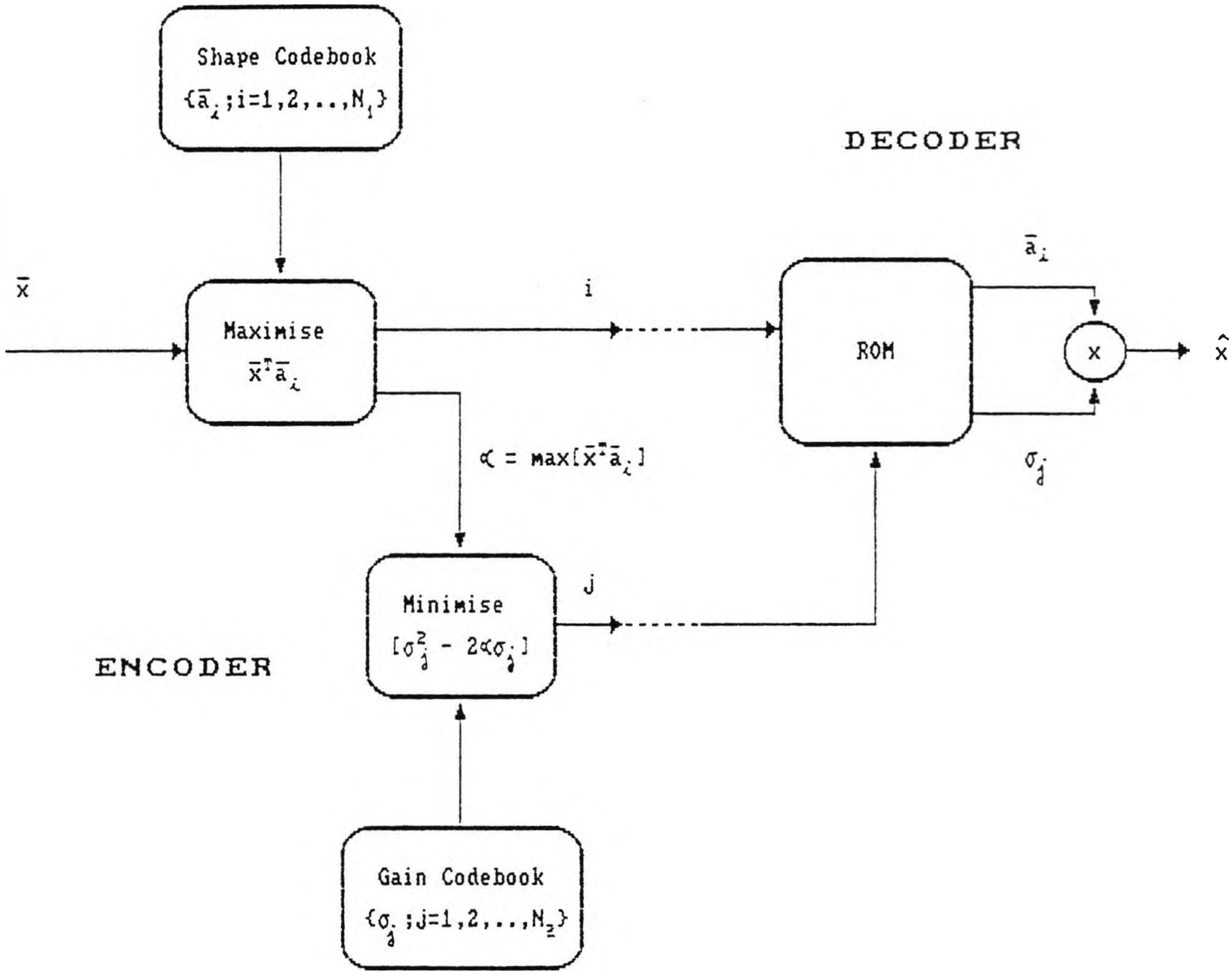


Figure 2.3: *A GSVQ*

should be noted that the encoder of a GSVQ is optimal for a given product codebook, but the codebook itself is in general suboptimal because of the constrained product form [Gray 1984]. Nevertheless, GSVQ has widely been employed for voice, waveform, and image coding. Because of its reduced complexity, both computational and storage, GSVQ is capable of being used at higher dimensions than full search VQ (having fewer codevectors to be stored and searched) providing an efficient method for data compression.

Several ways to tackle the computational problem encountered in VQ have been mentioned thus far without paying much attention to the degradation in perceived quality of coded images. Full search VQ, for example, is based on classical information theory without considering valuable properties of the intended receiver, i.e., the human observer. No attempt has been made to incorporate well-known properties of the HVS into the above coding scheme. Thus, one could intuitively expect that if such properties could be successfully combined with the compression process, an improvement in compression performance would result.

A new VQ-based coding method, *classified vector quantisation* (CVQ), which takes into account basic properties of the HVS and reduces both edge degradation and coding complexity, has been proposed in [Ramamurthi 1986]. The basic concepts of this new system along with a novel CVQ codebook design algorithm, developed in this research, are described in detail in the following section. They will be used as a basis for the development of a new coding scheme described in a later chapter.

2.3 Classified VQ

Classified vector quantisation (CVQ) is a VQ-based coding method for preserving perceptual features while retaining simple VQ distortion measures during the codebook design and encoding processes. CVQ is based on a composite source model so that small image blocks (4×4 pixels) with distinct perceptual features, such as edges, are generated from different subsources, i.e. belong to different classes. A finite set of idealized edge classes and a few nonedge classes are defined and tested.

In CVQ (Fig. 2.4), a *classifier* is employed to analyze each block (vector) $\bar{x}$, prior to encoding it, to determine which class the block belongs to. A distortion measure $d_i(\bar{x}, \bar{y})$ is then used, to pick the best codevector from the sub-codebook C_i which was specifically designed for that class. In other words, a separate VQ is employed for each class. M different classes are defined by the classifier and for each class a special codebook of size N_i is designed. The total number of codevectors is $N = \sum_{i=1}^{M} N_i$, with indexing from 1 to N. The index of the codevector, closest to the input vector, is sent to the decoder. The decoder simply retrieves the corresponding codevector from the codebook $C = \bigcup_{i=1}^{M} C_i$, and outputs it as the reproduction vector $\hat{x}$.

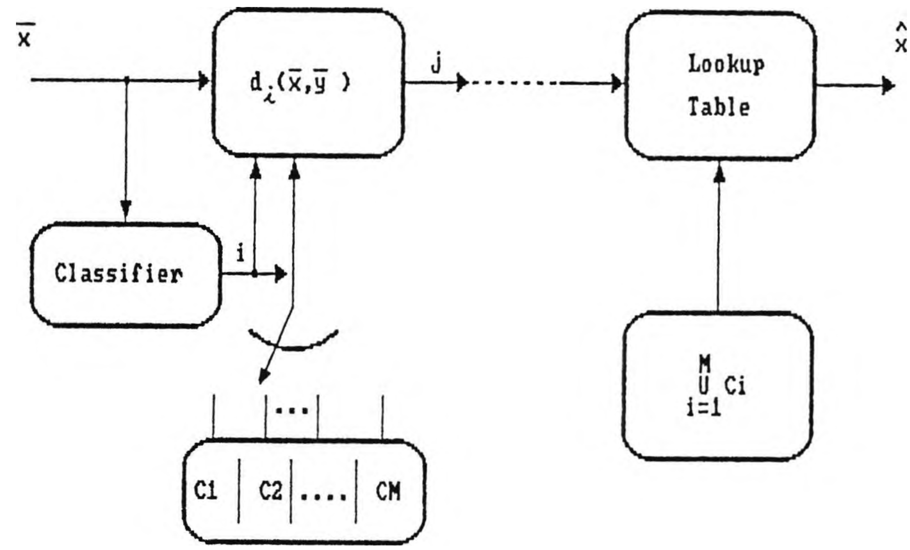


Figure 2.4: *A CVQ system*

For a CVQ system the overall average distortion D is given by

$$D = \sum_{i=1}^{M} P_i D_i \tag{2.18}$$

where P_i is the probability of a vector being in the ith class, D_i is the average distortion in the ith class and M is the number of classes. The average distortion D_i in each class is a function of the corresponding class codebook and its size N_i. The relation between the optimal distortion D_i^* and N_i is not known in general for finite N_i. Thus, it is not possible to find analytically the optimal set $\{N_i^*\}$ that yields the optimal D_i^*. In [Ramamurthi 1986], the authors solved the problem for asymptotically large N_i values for the case when each distortion measure $d_i(\cdot,\cdot)$ is the WMSE with a constant weighting matrix W_i. They have shown that, for optimality, the *average partial distortion per codevector* A_i in each class should be the same for every class, i.e.,

$$A_i = \frac{P_i D_i^*}{N_i^*} = \text{constant} \quad \text{for } i = 1, 2, \ldots, M\ . \tag{2.19}$$

Based on this relation, an empirical method for CVQ codebook design was proposed in [Ramamurthi 1986].

In [Ramamurthi 1986], the training set of image vectors, used for codebook design, is first classified into subsets, each subset containing vectors which belong to one class. A brute-force approach for obtaining the optimal codebook

C^* would be to design individual sub-codebooks using the LBG algorithm with every permissible allocation for the set $\{N_i\}$ such that $N_i \neq 0$ for all i, and $\sum_{i=1}^{M} N_i = N$. The number of permissible allocations, although very large, is finite, so that eventually an optimal allocation could be found along with a corresponding codebook that yields the least average distortion D^*. This approach, where all possible allocations for the set $\{N_i\}$ are tried, is computationally demanding and time consuming. Thus, the authors in [Ramamurthi 1986] have suggested that equation (2.19) should be used as a guide in experimentally determining a satisfactory set $\{N_i\}$.

Starting with an initial allocation $\{N_i^1\}$ of codevectors, M different sub-codebooks are designed using the LBG algorithm (M times). For each class the partial distortion D_i could then be calculated and relation (2.19) could be checked. If this relation is not satisfied, a different allocation $\{N_i^2\}$ is tested and the same procedure is carried out. By trial and error, an allocation $\{N_i\}$ is eventually found with corresponding distortions D_i such that relation (2.19) is approximately satisfied.

No systematic way has been proposed in [Ramamurthi 1986] for choosing the initial allocation of codevectors and no guidelines have been set for changing allocations during codebook design. Moreover, since in practice the sizes of the codebooks C_i will not be large enough to make relation (2.19) valid, this algorithm will yield suboptimal results in the sense of minimum squared error.

As an alternative to this empirical method, we propose a new algorithm which we name *Classified Nearest Neighbour Clustering* (CNNC). This new algorithm offers a systematic method for CVQ codebook design. Moreover, it designs the M sub-codebooks simultaneously in an optimal way, within the limit that CVQ is by definition a suboptimal system. Our new algorithm is based on two algorithms: the *Nearest Neighbour Clustering* algorithm proposed in [Helmuth 1980], and the *Pairwise Nearest Neighbour* algorithm proposed in [Equitz 1987] and [Equitz 1989] for VQ codebook design.

2.3.1 The CNNC Algorithm

The training image vectors $\{\bar{x}_t \in R^k \ ; \ t = 1, 2, \ldots, n\}$ are first grouped in several classes employing a classifier. The description of the classifier employed for the classification process could be found, for example, in [Ramamurthi 1986] and [Aravind 1986]. Then, within each class, clusters are defined each containing one training vector (see Fig. 2.5, at s=0). At this stage each training vector is also the centroid of the cluster. A search, within each class, is then conducted to find a pair of clusters which incur minimum *merging error* when merged. The merging error is defined as the additional distortion introduced by representing the two clusters worth of data with a single centroid. It is defined in Appendix B by

$$ME_{jl} = \frac{m_j m_l}{m_j + m_l} \parallel \tilde{X}_j - \tilde{X}_l \parallel^2 \qquad (2.20)$$

where m_j and m_l denote the number of vectors in clusters S_j and S_l respectively, and $\tilde{X}_j$ and $\tilde{X}_l$ are the centroids of these clusters.

The pairs of clusters, found in the different classes, are kept on a merging list and the pair of clusters which incurs the minimum merging error among all the pairs on the merging list, is selected in each stage and merged (see Fig. 2.5, at s=1 and s=2). After merging a pair of clusters in one class, their centroids are replaced with a weighted average centroid $\tilde{X}_p$ [Helmuth 1980] defined by

$$\tilde{X}_p = \frac{m_j \tilde{X}_j + m_l \tilde{X}_l}{m_j + m_l} . \qquad (2.21)$$

A search is then conducted in that class to find a new pair of clusters as candidates for merging in the next stage. The process is halted when the number of clusters left in the different classes accumulates to the desired number of codevectors N. The centroids of these clusters are the codevectors that constitute the desired CVQ codebook (see Fig. 2.5, at $s = T$). To conduct the geometric search for a closest pair of clusters within each class more efficiently, we have developed a fast search algorithm which is described in Appendix C.

An alternative criterion, for selecting the pair of clusters to be merged in each stage of the clustering process, is proposed. This criterion is based on the statement, mentioned earlier with respect to equation (2.19), that for optimality the average partial distortion per codevector in each class should be the same

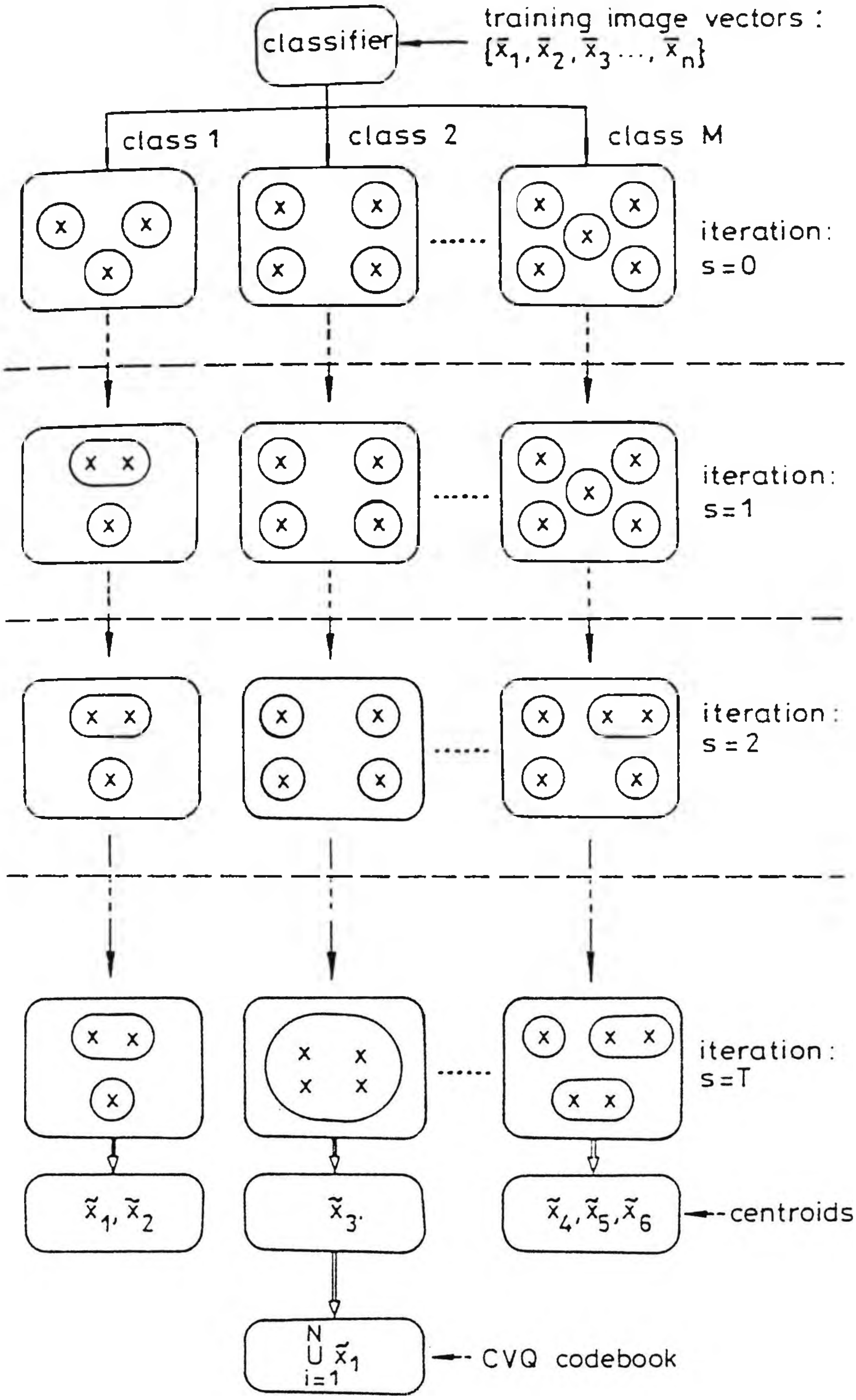


Figure 2.5: *CNNC algorithm - diagrammatic representation*

for every class. Thus, instead of merging the pair of clusters which incur the minimum merging error among all the pairs on the merging list, the pair of clusters belonging to the class in which the average partial distortion per codevector A_i is minimal, could be merged. For a long training set of vectors, employing this criterion within the CNNC algorithm could yield the approximation of the relation of equation (2.19) without the trial and error process, proposed in [Ramamurthi 1986]. It should be noted though, that for asymptotically large N_i values both criteria yield the same optimal results. Nevertheless, both criteria have been tested and experimental results are summarised in the next section.

2.3.2 Experimental Results

We present coding results with CVQ at different rates employing codebooks designed with the CNNC algorithm. The training set of vectors used for CVQ codebook design was obtained from two 256×256 images, 'Lena' and 'Baboon' (Fig. 2.6). The image block size was 4×4 pixels and thus the dimension of the vector was k=16. A classifier was implemented, based on basic ideas described in [Ramamurthi 1986], with several changes that enabled image blocks containing complex edges to be classified. Fifteen different classes were defined by the classifier, consisting of twelve edge classes and three nonedge classes. Only vertical, horizontal and two diagonal edge orientations were recognised. Each orientation was further subdivided in two classes, depending on whether the intensity change across the edge was from high to low level or vice versa.

CVQ codebooks at different rates, 0.5 - 0.625 bpp, were designed employing the CNNC algorithm. Two different criteria for selecting the class within which a pair of clusters should be merged during each stage of the CNNC algorithm were tested: the minimum merging error, $\min(ME_{jl})$, and the minimum average partial distortion per codevector, $\min(A_i)$.

The *signal to noise ratio* (SNR) was used as a rough indicator of the quality of the coded images. The SNR is defined by

$$\mathrm{SNR} = 10\log_{10}\left[\frac{V_{ptp}^2}{\mathrm{MSE}}\right] \quad \text{in dB} \tag{2.22}$$

where V_{ptp} is the maximum intensity (255 for 8 bpp), and MSE is the mean

Figure 2.6: *Original images of 'Lena' and 'Baboon'*

square error between the original image and the coded image. In addition, as it is well known that signal to noise ratio does not correlate well with subjective (human) quality assessments [Lukas 1982], numerous subjective tests have been carried out. SNR results for coding 'Lena' and 'Baboon' with these codebooks applying both merging criteria are presented in Table 2.1.

	Lena		Baboon	
Rate bpp	$\min(ME_{jl})$	$\min(A_i)$	$\min(ME_{jl})$	$\min(A_i)$
0.625	29.70	29.36	25.28	24.65
0.562	27.77	27.67	23.41	23.24
0.500	26.34	26.10	22.25	22.17

Table 2.1: CVQ system - SNR results for 'Lena' and 'Baboon'

It could be noted from Table 2.1 that CVQ codebooks, designed with the CNNC algorithm and the $\min(ME_{jl})$ criterion, yield better SNR results than codebooks that were designed with the $\min(A_i)$ criterion. The reason for this observation is that the relation of equation (2.19), on which the $\min(A_i)$ criterion is based, is true for asymptotic assumptions (N_i large for any $i = 1, 2, \ldots, M$).

However, for finite values of N_i, this criterion yields suboptimal results in the sense of minimum squared error. On the other hand, the $\min(ME_{jl})$ criterion yields optimal results within the suboptimal framework, which characterises CVQ systems.

In Fig. 2.7 two reconstructed images of 'Lena' at 0.562 bpp are shown. The image of Fig. 2.7*a* was produced by a CVQ system employing a codebook designed with the CNNC algorithm and the $\min(A_i)$ criterion. The image of Fig. 2.7*b* was produced by a CVQ system employing a codebook designed with the CNNC algorithm and the $\min(ME_{jl})$ criterion. Both images are magnified by a factor of two in order to show the differences between them.

a *b*

Figure 2.7: *'Lena' coded by CVQ systems at 0.562 bpp*
a. $\min(A_i)$ criterion *b.* $\min(ME_{jl})$ criterion

In general, the image of Fig. 2.7*a* looks better than the image of Fig. 2.7*b* - the edges are well defined and homogeneous areas are less "blocky". However, it should be emphasised that this observation is not always true. Subjective tests of reconstructed images at different rates provided no consistent results and thus, no clear conclusion can yet be drawn as to which merging criterion yields codebooks that produce images of better subjective quality.

In Fig. 2.8, two reconstructed images of 'Baboon' at rate 0.562 bpp are shown. These images were produced by the same CVQ systems as the images in Fig. 2.7 and are also magnified by a factor of two. Owing to the noisy nature of these images, it is difficult to decide which is better reproduced and thus we present them merely to show the good performance of CVQ systems.

a *b*

Figure 2.8: *'Baboon' coded by CVQ systems at 0.562 bpp*
a. $\min(A_i)$ criterion *b.* $\min(ME_{jl})$ criterion

Because the CVQ coder must ultimately operate on other input vectors that are not part of the training set, it is important to test the performance of the CVQ system outside the training set, i.e., with images which were not used for codebook design. Two images from the USC[1] database were used as test images: 'House' (USC15.IMG) and 'Tree' (USC416.IMG). These images were coded by CVQ systems employing the same codebooks that were used above for coding

[1] University of Southern California

'Lena' and 'Baboon'.

Coding results (SNR) for the test images are presented in Table 2.2. It could be noted from Table 2.2 that, for the 'Tree' image, better coding results were obtained when codebooks, designed with the CNNC algorithm and the $\min(ME_{jl})$ criterion have been employed. These results are in accordance with the coding results presented in Table 2.1. On the other hand, for the 'House' image, better coding results were obtained when codebooks designed with the CNNC algorithm and the $\min(A_i)$ criterion have been employed.

	Tree		House	
Rate bpp	$\min(ME_{jl})$	$\min(A_i)$	$\min(ME_{jl})$	$\min(A_i)$
0.625	23.82	23.54	27.71	27.84
0.562	22.97	22.92	26.76	26.84
0.500	22.33	22.16	26.22	26.26

Table 2.2: CVQ system - SNR results for 'Tree' and 'House'

Although the $\min(A_i)$ holds an imperceptible edge in SNR over the $\min(ME_{jl})$ criterion for the 'House' image, a thorough investigation was carried out in order to explain this observation. The codebooks, designed above with 'Lena' and 'Baboon' as training images, have been studied carefully as well as the coding results obtained for the training and the test images. Our findings are presented next.

The sum of errors, produced by the CVQ coder during the encoding process of image vectors which belong to class i, is defined as the *partial error* in that class, and is denoted by D_i. The partial error found in a certain class and the ratio between it and the total encoding error D (between the original image and the coded one) are of great importance for the evaluation of the employed codebooks. Such data indicate how well vectors which belong to a certain class are being represented by the employed CVQ codebook. If for a certain image the partial error in class i represents a large percentage in the total encoding error found for that image, it means that class i is poorly represented by that CVQ codebook. Moreover, a comparison of partial errors in different classes, obtained by encoding different images, can indicate how "close" are images to each other, i.e., whether they belong to the same ensemble of images, having common features or similar structure. Coded images which differ substantially in partial errors, obtained in several classes, are said to be of different nature, i.e., belong to different ensembles of images. A comparison of partial errors, caused during the encoding of the training images and the test images, is presented in Table 2.3. The CVQ codebook (at 0.625 bpp) was designed with the CNNC algorithm employing the $\min(ME_{jl})$ criterion.

			Lena	Baboon	Tree	House
Class	N_i	N_i/N (%)	D_i/D (%)	D_i/D (%)	D_i/D (%)	D_i/D (%)
9	231	22.6	36.6	56.5	23.8	**5.9**
14	158	15.4	5.7	5.7	2.5	**0.6**
15	146	14.4	3.2	5.6	1.6	**0.0**
4	27	2.6	1.7	2.8	6.2	**24.9**

Table 2.3: Coding evaluation for different images coded at 0.625 bpp employing a codebook designed with the $\min(ME_{jl})$ criterion

In Table 2.3, N_i is the number of codevectors representing class i, N is the total number of codevectors which constitute the CVQ codebook (1024 for 0.625 bpp), D_i is the partial error in class i, and D is the total encoding error for a certain image. The N_i figures for the first three classes in Table 2.3 are the largest for that codebook. Class 9 is the "mixed" class which consists of all the vectors that have very complex structure. Class 14 is the "two45" class which consists of all the vectors that have two significant edges in the 45 deg direction. Class 15 is the "two135" class which consists of all the vectors that have two significant edges in the 135 deg direction. Class 4 is named "neghoriz" and it consists of all the vectors that have one significant edge in the horizontal direction with intensity that changes from low levels to high levels when the image block (vector) is being scanned from top to bottom.

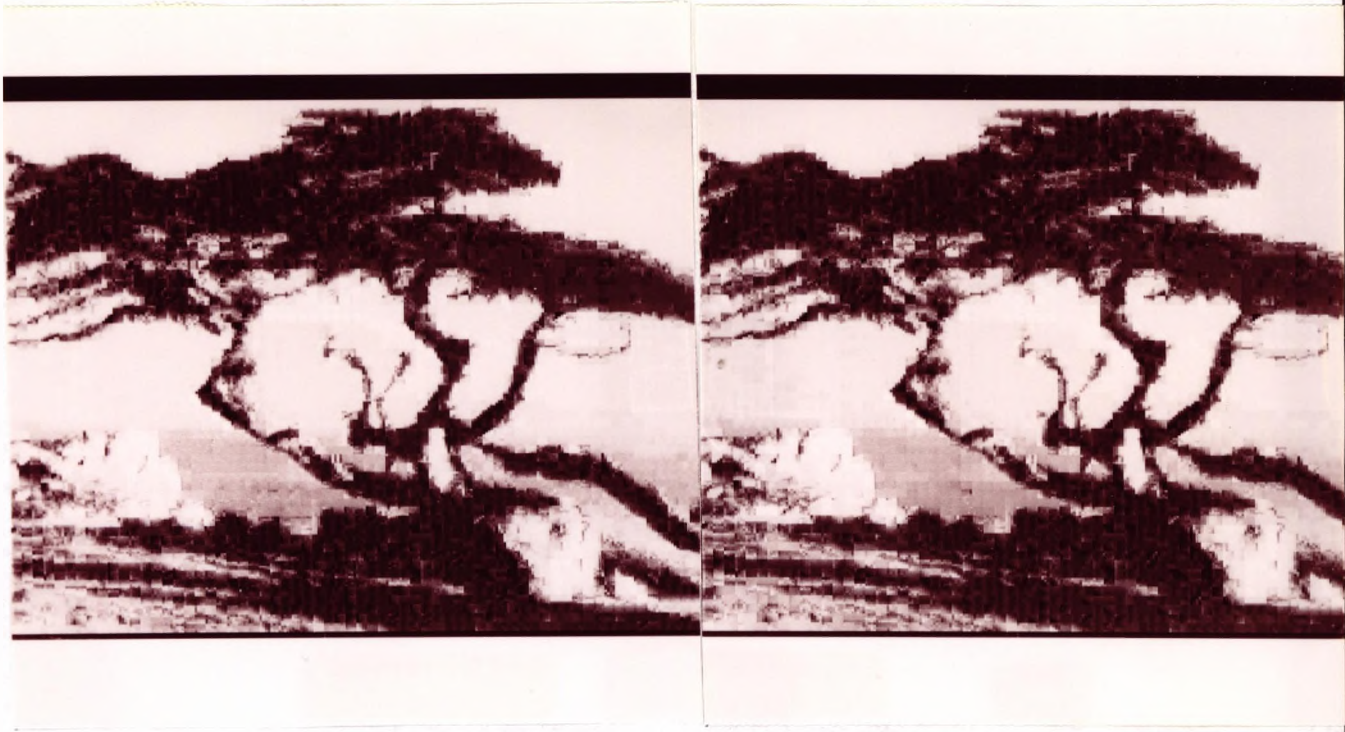
The results, presented in Table 2.3 are representative of coding results obtained for the different codebooks employed throughout our study. Based on these results, we argue that the 'Tree' image belongs to the same ensemble of images as 'Lena' and 'Baboon'. Therefore, coding results for the 'Tree' image are in accordance with coding results obtained for the training images. On the other hand, the 'House' image does not belong to that ensemble (see particularly the differences in D_i/D figures for classes 9 and 4). This observation explains the results presented in Table 2.2. Nevertheless, the differences in SNR values shown in Table 2.2 are insignificant thus, it is believed that each of the merging criteria may be used for codebook design employing the CNNC algorithm.

Subjective tests of reconstructed test images have revealed that conclusions drawn earlier for the training images hold for the test images as well. No perceptual meaning has been found to either one of the merging criteria and no clear conclusion can be made as to which criterion yields codebooks that produce images of better subjective quality.

In Fig. 2.9 two reconstructed images of 'Tree' at rate 0.562 bpp are shown. Both images look very similar with a slight advantage to the image of Fig. 2.9*b* ($\min(ME_{jl})$ criterion) which looks less "blocky" in several homogeneous areas. In Fig 2.10 two reconstructed images of 'House' at rate 0.562 bpp are shown. Both images look very similar with a slight advantage to the image of Fig 2.10*a* ($\min(A_i)$ criterion) which looks less "blocky" in several homogeneous areas.

A direct comparison between codebooks generated by the CNNC algorithm and those generated by the empirical algorithm proposed in [Ramamurthi 1986] seems impractical. As mentioned before, no clear guidelines have been set in [Ramamurthi 1986] for choosing the initial allocation of codevectors in the different classes or for changing allocations during codebook design. Thus, no attempt has been made to design codebooks employing the empirical method. Instead, a theoretical study of the complexity of the CNNC algorithm in comparison to the brute-force method described in section 2.3 and [Ramamurthi 1986] has been carried out in Appendix D.

Accordingly, it could be shown that, for the CNNC algorithm, the upper bound on the number of required multiplications during the design process is $O([n-N+M]^3[k+2]/6M^2)$ where $O(\cdot)$ denotes proportional to. On the other hand, for the brute-force algorithm, it could be shown that the lower bound is $\{(2nk/M)N(N-1)!\}/\{(M-1)!(N-M)!\}$ where ! denotes factorial. Consequently, for n=8196, M=15, and N=512, approximately 6.1×10^9 multiplica-



a *b*

Figure 2.9: *'Tree' coded by CVQ systems at 0.562 bpp*
a. $\min(A_i)$ criterion *b.* $\min(ME_{jl})$ criterion

a *b*

Figure 2.10: *'House' coded by CVQ systems at 0.562 bpp*
a. $\min(A_i)$ criterion *b.* $\min(ME_{jl})$ criterion

tions are required for the CNNC algorithm while requiring 7.1×10^{33} multiplications for the brute-force algorithm. The advantage of our new algorithm is apparent.

Finally, it should be noted that the weighted average centroid, produced when two clusters in a class are merged, does not necessarily belong to the same class. Thus, at the conclusion of the CNNC algorithm the codevectors that are the centroids of the remaining clusters do not necessarily belong to the classes within which they were developed. The authors in [Ramamurthi 1986] encountered this problem while designing different codebooks for the different classes employing the LBG algorithm. They proposed a replacement procedure, arguing that it makes little perceptual difference whether the codevector or the training vector nearest to it, which does belong to the appropriate class, is employed to code a particular block. We have adopted their approach and at the conclusion of the CNNC algorithm each codevector was tested by the classifier. If it did not belong to the particular class within which it was developed, it was replaced in the way proposed in [Ramamurthi 1986].

2.4 Conclusions

The basic notion of vector quantisation along with codebook design methods have been described. Various drawbacks of a full search VQ have been discussed, particularly the computational complexity caused by the search problem and the degradation in the perceived quality of coded images. Classified VQ has been proposed to overcome several of the mentioned drawbacks and its basic principles were described based on a composite source model.

A new algorithm for CVQ codebook design, the classified nearest neighbour clustering (CNNC) algorithm, was presented in this chapter as an alternative to the empirical method proposed in [Ramamurthi 1986]. The new algorithm reduces considerably the total number of required operations (multiplications, additions etc.) during codebook design, in comparison with the brute-force method described in section 2.3 and in [Ramamurthi 1986]. Two different criteria for selecting the class within which a pair of clusters should be merged during each stage of the CNNC algorithm were presented: The minimum merging error, $\min(ME_{jl})$, and the minimum average partial distortion per codevector, $\min(A_i)$. In addition, a fast search algorithm was proposed (see Appendix C) to reduce computational efforts during codebook design.

Codebooks for CVQ systems at different rates were designed, and coding results (SNR values) were compared to decide which criterion is most appropriate for codebook design. In general, the $\min(ME_{jl})$ criterion yielded better SNR results than the $\min(A_i)$ criterion provided that the encoded images belong to the same ensemble of images as the training images that were used during codebook design. Subjective tests of reconstructed images at different rates, on the other hand, provided no consistent results, and thus no clear conclusion can be drawn as to which merging criterion yields codebooks that produce images of better subjective quality. Nevertheless, we have shown that the CNNC algorithm in conjunction with the fast search algorithm, described in Appendix C, provides a systematic and effective method for CVQ codebook design making CVQ systems more feasible and easy to implement.

Close inspection of CVQ-coded images at rates lower than 0.562 bpp has revealed the fact that the staircase effect and the blocking effect are still noticeable. Some improvement may be achieved if the perceptual importance of the various classes could be taken into consideration. For example, by assigning different weights w_i to the different classes, as proposed in [Ramamurthi 1986], the number of codevectors found eventually for each class can be influenced without changing the clustering process or the encoding process. The larger the weight assigned to a class, the more the importance attached to that class, and the larger the number of codevectors found for that class.

Employing a weight w_i for the ith class, when the $\min(A_i)$ criterion is used within the CNNC algorithm, is equivalent to increasing the probability of that class by the factor w_i with appropriate normalisation so that the probabilities add up to 1. Employing a weight w_i for the ith class, when the $\min(ME_{jl})$ criterion is used, is equivalent to increasing the merging error incurred when two clusters are merged within the ith class. This approach has an intuitive appeal when applied to the problem of edge degradation, which was the major reason for the developing of CVQ systems, since it permits us to control the relative sub-codebook sizes and thus ensures edge integrity at the expense of SNR results.

The various concepts described above along with the CNNC algorithm and the fast search algorithm, proposed by us, will be used in the development of a new coding system. Various properties of the HVS should be exploited to improve the perceived quality of coded images and therefore are summarised in the following chapter.

Chapter 3

The Human Visual System - A Review

Easy to see the tiger's stripes;
Hard to see the mind's shadows.

The very end of almost every image processing system is the human eye. Therefore, it is useful to know how and what the eye sees. The answers to these questions lie far outside from our usual engineering context and that is why, a summary of the mechanism of vision is given here based on [Hubel 1979], [Cornsweet 1970], and on other sources. Particularly, properties of the *human visual system* (HVS) which can be used in image processing and coding are described. It is postulated that if the coding scheme is matched to the HVS and attempts to imitate its functions, at least for the known part of it, high compression ratios along with good quality of the reconstructed images can be expected.

The major source of information about the human visual system is through physiological studies. The visual systems of all vertebrates are similar and measurements of the visual systems of the cat and monkey in particular provide useful information concerning the behavior of the HVS. A second source of information is psychophysical experiments. In psychophysics the visual system is treated as a "black box" with visual stimuli as the input and perceived sensations as the output. Functional relationships are sought between some physical properties of the stimulus and the corresponding psychological, or perceptual, response. Based on these sources of information, the visual path in a primate from the eye to the cortex is described preceded by an introduction to the nervous system.

3.1 The Nervous System

The human visual system is part of the nervous system which can be compared to a complicated communication network supervised by the most powerful computer: the brain. The communication in this network is carried out through nerve cells called *neurons*. A neuron has a body of size varying between 5 and 100 μm. A main fiber called the *axon* and a number of fiber branches called *dendrites* are attached to this body as can be noticed in Fig. 3.1. The information transfer from one neuron to another is made electrochemically. The places in which neurons come into contact, i.e., where chemical mediator substances are released by one neuron and act upon the receiving neuron, are called *synapses*. The transmitting and the receiving neurons are called *presynaptic* and *postsynaptic* neurons, respectively. A neuron can receive signals from thousands of presynaptic neurons and can transmit to thousands of postsynaptic neurons.

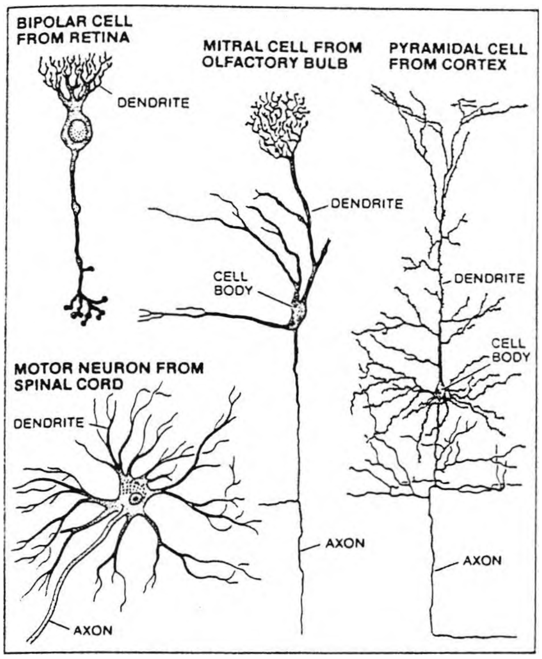


Figure 3.1: *Typical neurons* (after [Kuffler 1976]).

The membrane that forms the walls of any neuron is semipermeable. This means that it has holes through it that permit the passages of some molecules and ions into and out of the cell, but prevent the passage of others. The membrane also manifests a phenomenon called the *sodium pump*, i.e., it actively pumps sodium ions from the inside to the outside. As a result of the sodium pump and the differences in permeability of the membrane to different ions, the concentration of positive ions is greater on the outside of the cell than on the inside. This concentration difference appears as a voltage across the membrane, called the *resting potential.* The resting potential is between about 0.06 and 0.09 volts for any living resting neuron [Cornsweet 1970] with the outside being more positive than the inside. The difference in ion concentration across the membrane is also called a *polarization* since it is manifested in positive and negative poles across the membrane.

Most of the mammalian neurons are normally stimulated into action in the same way. If the endings of one neuron are in close proximity to a second neuron and those endings are stimulated, they release a chemical, called a *neural transmitter*, that changes the permeability of the receiving cell membrane. Since there is an excess of positive ions on the outside of the membrane, the rate of flow of positive ions into the cell will become greater if the permeability increases. The action of the excitatory chemical transmitter substance thus results in the depolarization of the membrane i.e., the voltage across the membrane is reduced. If a region of a neuron undergoes enough depolarization, a further series of events are triggered in the membrane that result in a wave of strong depolarization that self-propagates, sweeping over the entire neural membrane. There is a minimum number of molecules of the chemical transmitter that should act on the membrane in order to produce enough depolarization to trigger the processes of propagation. Once a given region of the axon has become depolarized, it quickly restores itself to its initial, polarized state. Thus, only a short region of the neuron is depolarized at any given instant, and this region will sweep progressively over the surface. The wave of depolarization is called a *spike* or *nerve impulse.*

It should be noted that the signals in the nervous system are almost identical in all neurons regardless of the information they carry: visual, audible, etc. Moreover, their shape does not vary from species to species, i.e., a signal recorded from a cat is similar to that recorded from a human being. The signal recorded at a given neuron is a pulse train. Each pulse has a magnitude of about 100 mV and a duration of about 1 mSec. The repetition rate (frequency) of these pulses is proportional to the intensity of the stimulus imposed on the neuron under test as shown in Fig. 3.2. Note, in this figure, that as the strength of the input stimulus increases from zero, it must exceed a threshold before any pulses are generated. Above that threshold, the frequency of firing of the neuron first increases almost linearly with input strength, and finally, at high input strength, the frequency asymptotically approaches its maximum level.

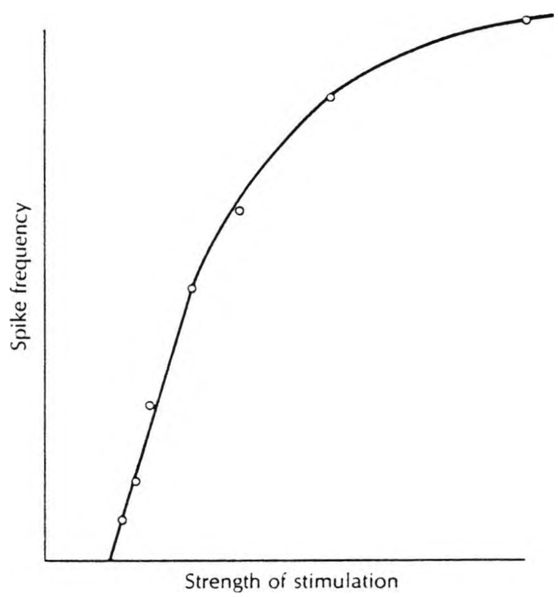


Figure 3.2: *The relationship between the intensity of the input stimulus to a neuron and the resulting frequency of its pulses* (after [Cornsweet 1970]).

Accordingly, it can be summarised that the nervous system communicates through *frequency modulation.* The signals received and processed by the brain are symbols representing external or internal events and what allows the brain to distinguish between two identical signals is the pathway used by each of the signals. There is a specific ensemble of neurons corresponding to each type of excitation forming a one-to-one mapping between different parts of the body and the brain.

In the following sections we will describe the visual path in a human being from the eye to the brain emphasising various details which can be useful to us in our efforts to devise better image coding systems.

3.2 The Eye

Fig. 3.3 illustrates the principal components of the human eye.

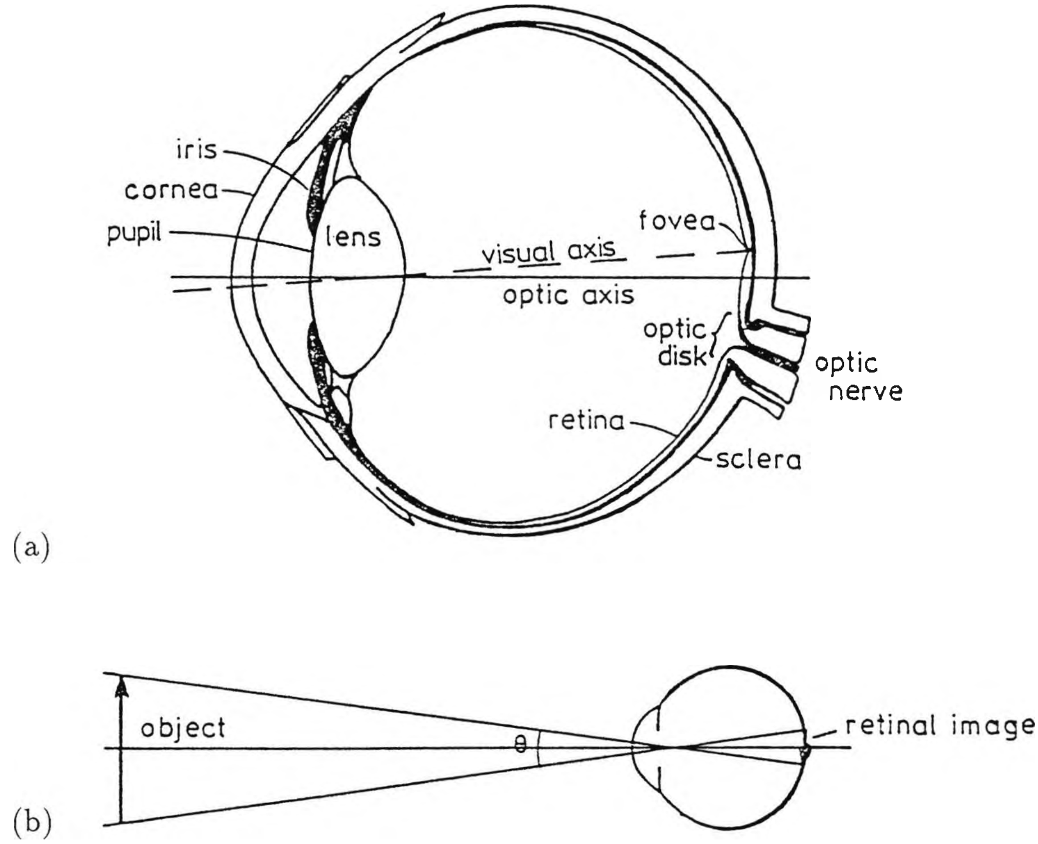


Figure 3.3: (a) *Diagram of a cross section of the human eye.*
(b) *Retinal image formation* (after [Cornsweet 1970]).

Light from an external object is focussed by the cornea and the lens to form an image of the object on the retina at the back of the eyeball. Although most of the refraction takes place at the cornea additional focussing is achieved by varying the thickness, and hence the refractive power, of the lens. This enables the eye to accommodate to varying object distances. The light passes through the cornea and enters the inner eye through the pupil. The amount of light allowed to enter is regulated by the pupil which acts like the diaphragm in a photo camera. The pupil's diameter varies between 2 and 9 mm. This aperture can be modeled as a low-pass filter having a cutoff frequency which depends on the pupil's diameter. The highest cutoff frequency corresponds to 2 mm while continuous enlargement of the pupil's diameter decreases the cutoff frequency. The lens of the eye is not perfect even for persons with no weakness of vision. This imperfection is the source of the spherical aberration which appears as a blur in the focal plane. Such a blur can also be modeled as a two-dimensional low-pass filter.

The retina is the neurosensorial layer of the eye occupying an area of about 12.5 cm^2. It transforms the incoming light into electrical signals that are transmitted to the visual cortex through the optic nerve. The retina consists of a layer of photoreceptors and connecting nerve cells. The photoreceptors are at the part of the layer furthest from the centre of the eyeball so that light must pass through the layer of nerve cells to reach the photoreceptors (see Fig. 3.4).

The photoreceptors contain photosensitive pigments which absorb light and initiate the neural response. There are two types of photoreceptors: *rods* and *cones*. The rods and the cones have different operating characteristics as illustrated in Fig. 3.5. The rods are the more sensitive, coming into operation at lower levels of background luminance than the cones. However, the rod curves are compressed with increasing luminance and saturate completely after about 4 log units. By contrast, the cones do not saturate: their operating curves shift along the intensity axis so as to span a constant 3 log unit range of intensities about each luminance level.

The visual system can operate at levels of illumination varying over 10 orders of magnitude, i.e., 10 log units or a range of 10^{10}. The rods are responsible for the low-resolution, night vision that exists below 0.1 mL. This is termed the *scotopic region*. The cones, on the other hand, are responsible for the high-resolution, daylight vision that exists above 1 mL. This is termed the *photopic region*. Since the range of luminances encountered in television falls within this region we are primarily concerned with cone vision.

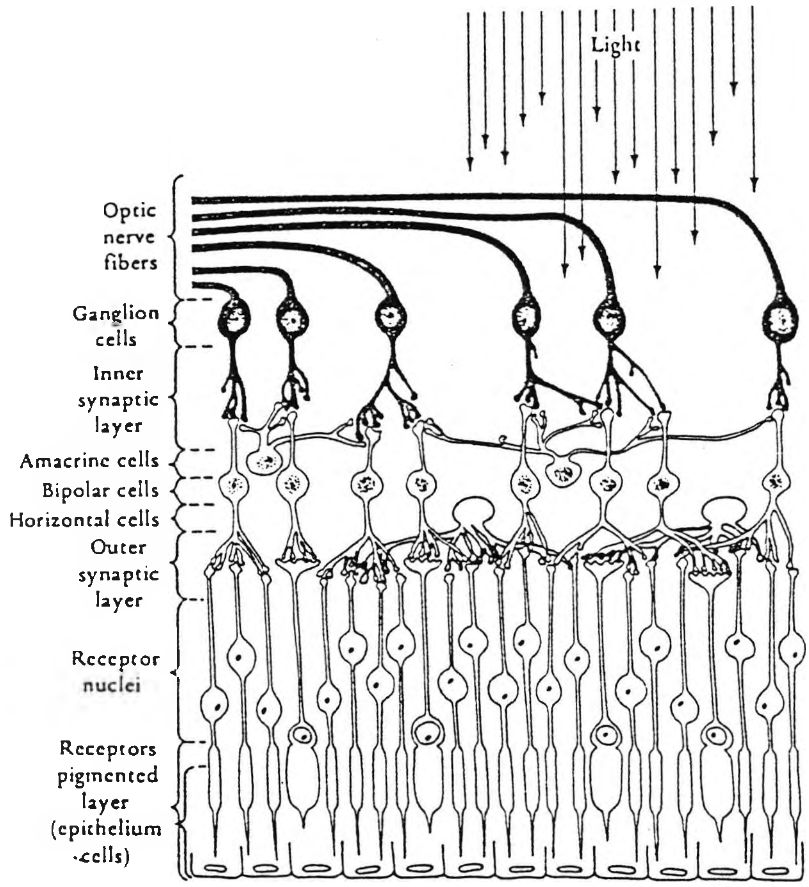


Figure 3.4: *Schematic diagram of the retina* (after [Dowling 1966]).

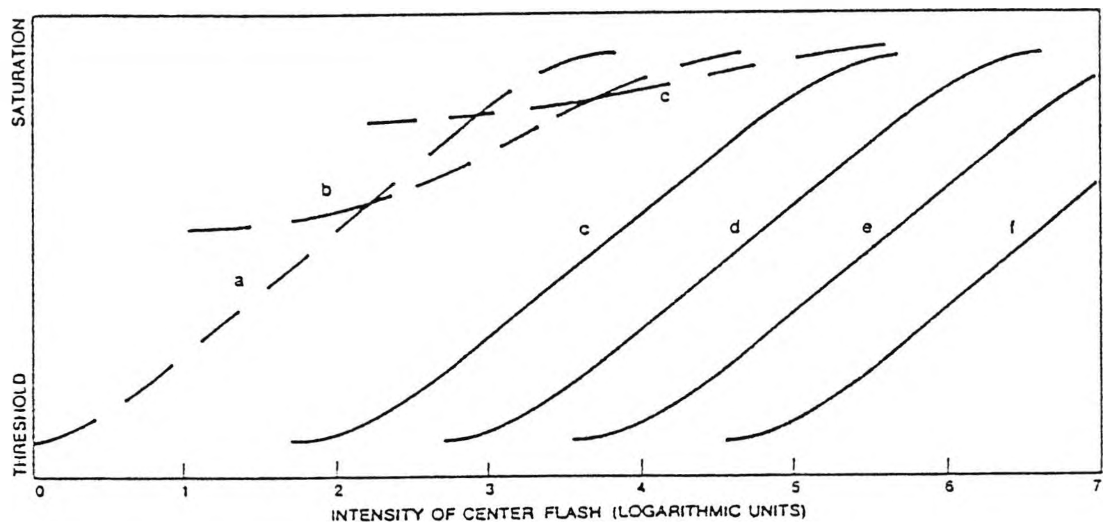


Figure 3.5: *Operating characteristics of rods (dashed lines) and cones (solid lines) at six background luminances (a-f)* after [Werblin 1973].

Fig. 3.6 is a plot of the distribution of rods and cones over the whole of the retina, along with a drawing of the eye for reference. The cones are densely packed in the centre of the retina, a region called the *fovea*. being spaced about 0.5 minutes of arc apart in the human retina. This density starts dropping off outside a circle of roughly 0.5 deg radius but is still appreciable within a circle of 1 deg radius. The rods begin to appear about 1 deg from the centre, and their density increases to a maximum at about 20 deg, thereafter falling to a low level. There are about 120 cones per degree at the fovea thus fixing the visual resolution to 1 minute of arc (i.e., 60 cycles per degree of arc subtended in the field of vision).

There are three different classes of cones in the normal human retina, each with a different absorption spectrum and each spectrum being shifted with respect to the others along the wavelength dimension (see Fig. 3.7). Each of these curves is also different from the spectral sensitivity curve of *rhodopsin*, the rod pigment. Therefore, the normal retina contains four classes of receptors with different absorption spectra. However, based on various experiments, it is quite evident that the rods do not contribute to wavelength discrimination. i.e., the cones alone are responsible for colour vision. Cones and rods respond differently to test flashes of increasing intensity especially at switch-off. Fig. 3.8 shows typical receptor responses to test flashes in the retina of the mudpuppy. It can be noticed that at this level of the visual system, the receptors respond by means of slow graded (analog) electrical potentials.

The anatomy of the retina shows five types of cells organised in layers (see Fig. 3.4). The photoreceptors make synaptic contacts with *bipolar cells* which extend forward through the nerve layer of the retina (toward the lens and source of light) and connect again via synaptic contacts to *ganglion cells*. The axons of these ganglion cells then continue the communication path to the central nervous system by joining to form the *optic nerve* which leaves the eye just to the inside of the centre of the retina. The region where the axons from all the ganglion cells converge and exit from the eyeball, and through which the retinal arteries and veins pass is called the *optic disk*. It contains no receptors and therefore it is also called the *blind spot*. Except for this region, the distribution of receptors is radially symmetric about the fovea.

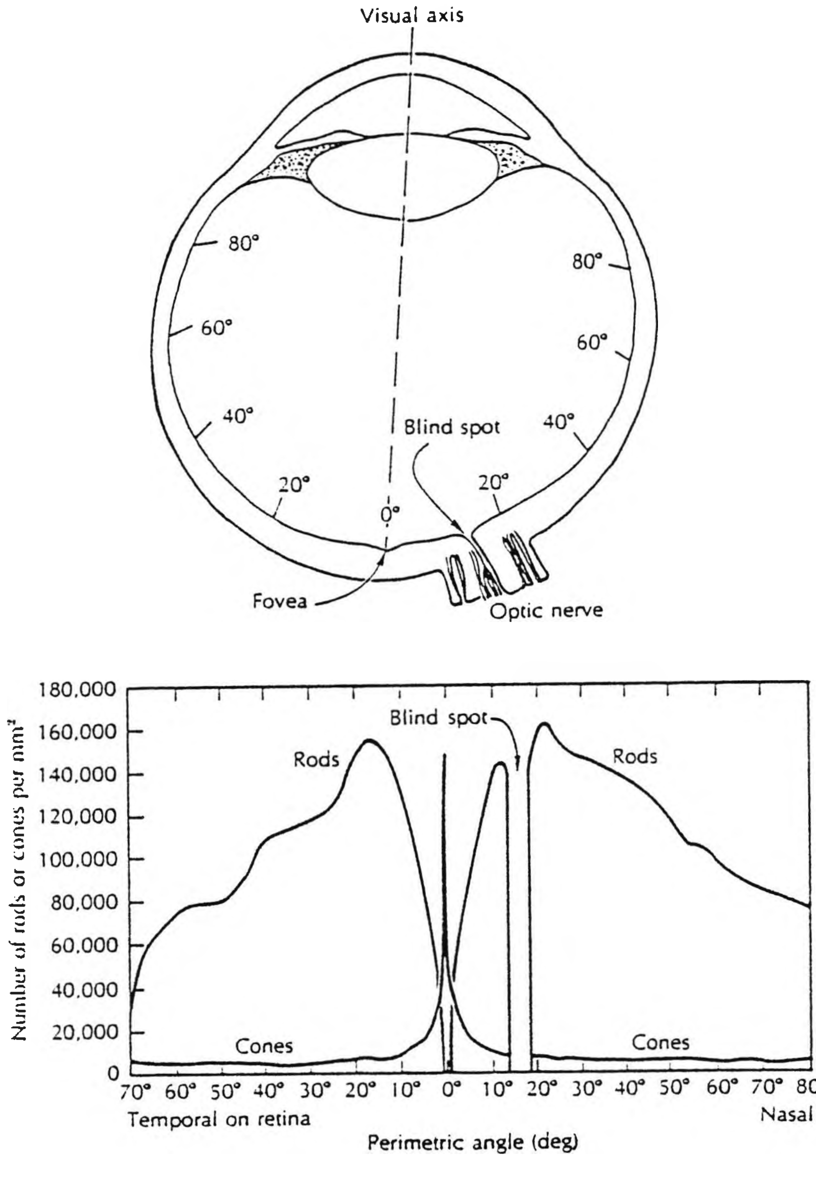


Figure 3.6: *The distribution of cones and rods across the retina* (after [Cornsweet 1970]).

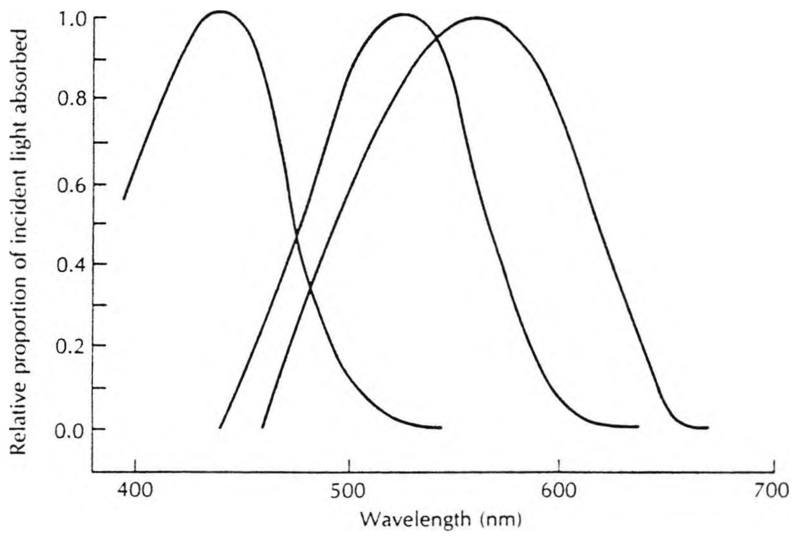
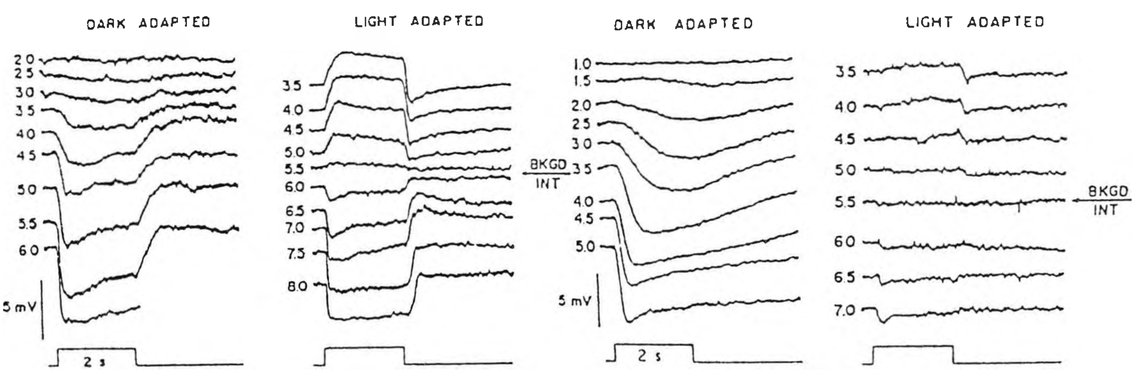


Figure 3.7: *Absorption spectra for three classes of cones* (after [Wald 1965]).



Rod response

Cone response

Figure 3.8: *Receptors responses to* 2 Sec *test flashes of increasing intensity (in* log *units).* (after [Norman 1974]).

Laterally connecting the photoreceptors are *horizontal cells* which receive inputs from some photoreceptors and may act via synaptic connections on other photoreceptors and bipolar cells. *Amacrine cells* make similar lateral connections between the bipolar cells and may act on ganglion cells. Some of these actions indicate feedback loops in lateral connections. These lateral and feedback connections are responsible for the so-called *lateral inhibition* phenomenon [Cornsweet 1970, pp. 284-310], and give rise to the concept of the *receptive field* of a cell. Lateral inhibition has a great effect on the ability of the eye to perform edge detection, thus we will describe it in some detail based on physiological data which have been collected through experiments on various animals (especially the horseshoe crab, *Limulus*). In addition, the concept of receptive fields will be introduced due to its importance to the study of ganglion cells and other cells in the visual cortex.

Various preliminary studies of the lateral eye of the Limulus have indicated that many of the properties of this eye are similar to those of higher animals, including humans. Therefore, it has been studied very extensively by different researchers. Most of this work has been carried out by Hartline and his colleagues [Hartline 1932], [Hartline 1947] and [Hartline 1956], whose studies have had an enormous influence upon current theories of the processes of human brightness perception.

The eye of the Limulus is faceted (as the eye of the fly). In the faceted eye of the Limulus, there is a separate optical system (a lens and cornea) for each facet, and the lens in each facet forms an image in the plane of the receptors contained in the facet. The characteristics of the Limulus eye are such that light from a roughly circular area in the visual field falls on the visual pigment in each facet, and adjacent facets receive light from overlapping but somewhat displaced fields.

In the human eye, there is a single optical system that forms an image on a mosaic of receptors. Each receptor receives an amount of light proportional to the amount radiated or reflected from some particular region of the visual field. The fields of different human receptors overlap because of diffraction, aberration and scattered light. Thus, comparing the optics of the Limulus eye to the optics of the human eye, we can consider each facet in the Limulus to be the rough analog of each receptor in the human eye [Cornsweet 1970].

Various experiments have revealed that each fiber in the optic nerve of the Limulus appears to be connected to a single facet and it would only fire when a spot of light falls on that facet, named the *excitatory facet*. If the facets neighbouring that particular one are illuminated, the nerve fiber will not respond. However, inhibitory interactions among facets have been found to be a noteworthy feature of the action of the Limulus eye. Suppose that a dim light always falls on the excitatory facet and a brighter spot of light is then moved across the Limulus eye. The results (the firing rate of the excitatory facet) would be as plotted in Fig. 3.9.

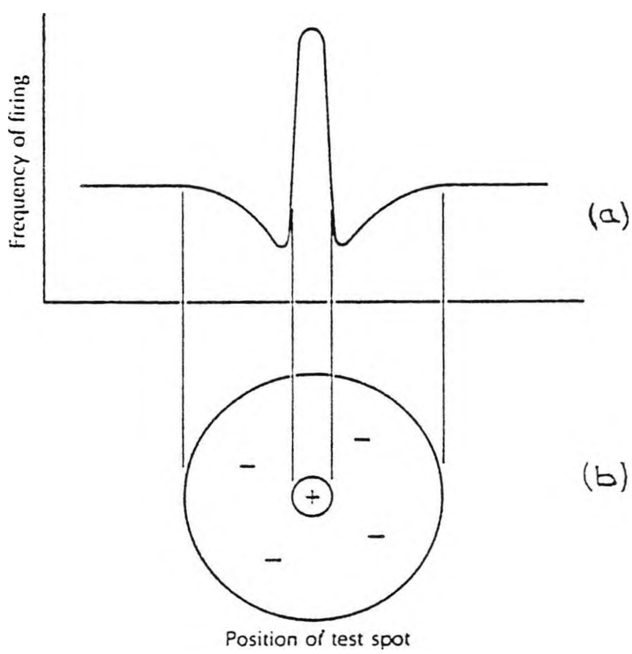


Figure 3.9: *Spatial impulse response of the excitatory facet - experimental results* (after [Cornsweet 1970]).

When the spot of light falls on facets that are far away from the excitatory facet it would be firing constantly, since it is constantly illuminated. However, as the moving spot approaches the excitatory facet it falls on facets that are closer and closer to the excitatory one and inhibitory effects would take place, i.e., neighbouring facets would inhibit the excitatory one more and more strongly, reducing its firing rate. When the moving spot falls directly on the excitatory facet, however, the firing rate would increase to its maximum. This phenomenon is called *lateral inhibition*.

Some of the results of the interactions between facets in the Limulus have obvious implications for the understanding of human perception. For example, suppose that the intensity distribution depicted as a dashed line in Fig. 3.10 is presented to the Limulus eye. Then the activity levels of the facets would be as indicated by the solid line in that figure due to lateral inhibition (the open circles represent the facets activity if there were no inhibition).

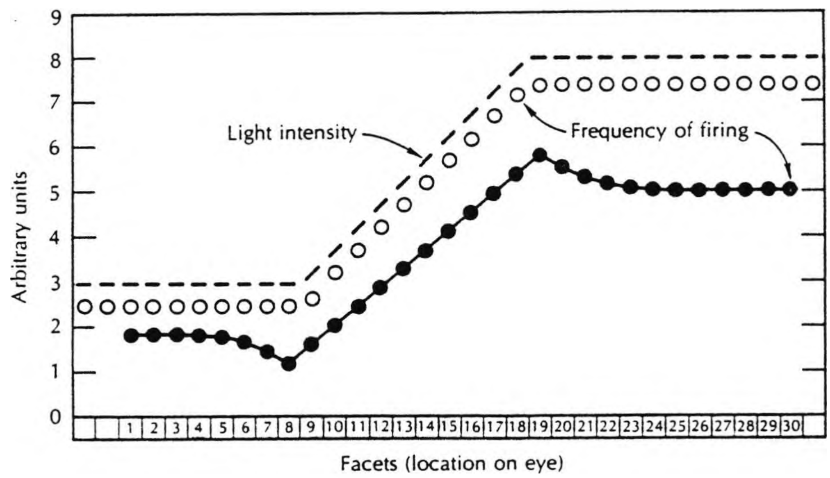


Figure 3.10: *Qualitative prediction of the facets activity in the Limulus eye for a given intensity distribution* (after [Cornsweet 1970]).

When the intensity distribution in Fig. 3.10 is presented to a human subject and he is asked to plot the brightness distribution, he will draw something that looks just like the solid line in Fig. 3.10. Similar electrophysiological experiments on the retina of mammals, mostly cats and monkeys, also show clear evidence of lateral inhibitory interactions.

Fig. 3.9*b* is a description of a two-dimensional "top view" of the Limulus eye region over which light affects the activity of the excitatory fiber. This region is called the *receptive field* of the fiber under test. The small region, centered on the excitatory facet, is labeled "+" because light falling on it causes an increase in the fiber firing rate. On the other hand, when the spot of light falls anywhere in a surrounding region of the excitatory facet, its net effect on the facet being recorded from is inhibitory, as indicated by the ring labeled "-". There is summation within the receptive field. Since inhibitory influences add to each other, two spots of light simultaneously falling within the inhibitory area will generally produce stronger inhibition than a single spot. Similarly, two spots of light falling within the excitatory area will produce stronger excitation than a single spot. Finally, it should be noted that the size of the receptive field depends upon the intensity of the stimulating light spot and must also depend upon the size of the light spot being used to measure it.

In an analogous way, similar receptive fields exist in the human eye due to lateral connections between adjacent receptors and nerve cells in the retina. A receptive field of a cell in the human eye is defined as the retinal region over which light affects the activity of that cell. Two types of receptive fields have been found : an "on-centre" field in which light falling in the centre has an excitatory effect while having an inhibitory effect when falling on the surrounding ring, and an "off-centre" field that is just the opposite. A systematic analysis shows that even a very small spot of light (0.1 mm in diameter) can cover several overlapping receptive fields causing excitation of some ganglion cells and inhibition of others. It has also been found that these cells are spatially grouped, i.e., cells processing information coming from a given area of the retina are grouped.

The receptive fields of retinal cells are always very nearly isotropic (circularly symmetric). Thus, at this level in the visual system, the information is processed independently from spatial orientation. Another important observation is that the receptive fields are very small in the vicinity of the fovea and progressively increase in diameter as we go away from the fovea.

The receptive fields of bipolar cells display a concentric organization with an excitatory centre and an inhibitory surround similar to the one depicted in Fig. 3.9*b*. The excitatory centre is mediated by the direct receptor-bipolar connections while the inhibitory surround is derived from the lateral interactions of the horizontal cells. Bipolar cells respond by means of graded potentials (similar to the receptors but with steeper responses).

Two types of bipolar cells have been identified. One responds to central stimulation with a hyperpolarizing potential and the other with a depolarizing potential. With either cell type annular illumination antagonizes the sustained potential produced by central illumination. The bipolar cells drive ganglion cells which respond in the form of nerve impulses. The firing rate of the ganglion cells carries the information to higher centres.

Referring back to Fig. 3.9*a*, the spatial impulse response clearly corresponds to a high-pass behavior that is at the origin of the effect of *contrast enhancement.* Since illumination of the surround is fed forward to inhibit the signal elicited by illumination at the centre of the receptive field, the overall response is lowered when the field is uniformly illuminated and raised in the vicinity of a luminance edge. Consequently, the *neural image* (the perceived image) of an edge is enhanced by means of undershoots and overshoots that correspond to the Mach band phenomenon in psychophysics [Ratliff 1965] (see also Fig. 3.10). Edge detection is mainly due to ganglion cells, and their effect can be modeled as a two-dimensional high-pass filter. However, it should be noticed that this filter is shift-variant because of the nonconstant spatial resolution of their receptive fields which change as a function of the distance from the fovea.

The response of ganglion cells to a centre spot of light adapts to widely changing levels of background illumination. This behavior is shown in Fig. 3.11. The firing rate of the cell goes from threshold level (resting firing rate, zero or tens pulses per second) to saturation (200-300 pulses per second) with a change of intensity of centre excitation of just under one log unit (a factor of 10). Adaptation mechanisms shift this active graded response region to span the ambient background level. In this active region this relation is nearly linear in firing rate versus the logarithm of intensity, and the slope in this linear region is virtually independent of the background illumination. Thus, the absolute light intensity is ignored by ganglion cells which measure only differences in their receptive fields.

There is sufficient direct physiological evidence to support the hypothesis that the relationship between the light intensity input to the visual receptors and the neural output level is approximately logarithmic. In 1932, Hartline and Graham reported the recording of neurological signals from single receptors in the eye of the Limulus [Hartline 1932]. Subsequently, Fuortes measured the electrical properties of the nerve cells of the eye of the Limulus [Fuotes 1958], [Fuotes 1959]. His findings were consistent with the hypothesis that light stimulating the receptor causes it to release a chemical mediator that decreases the

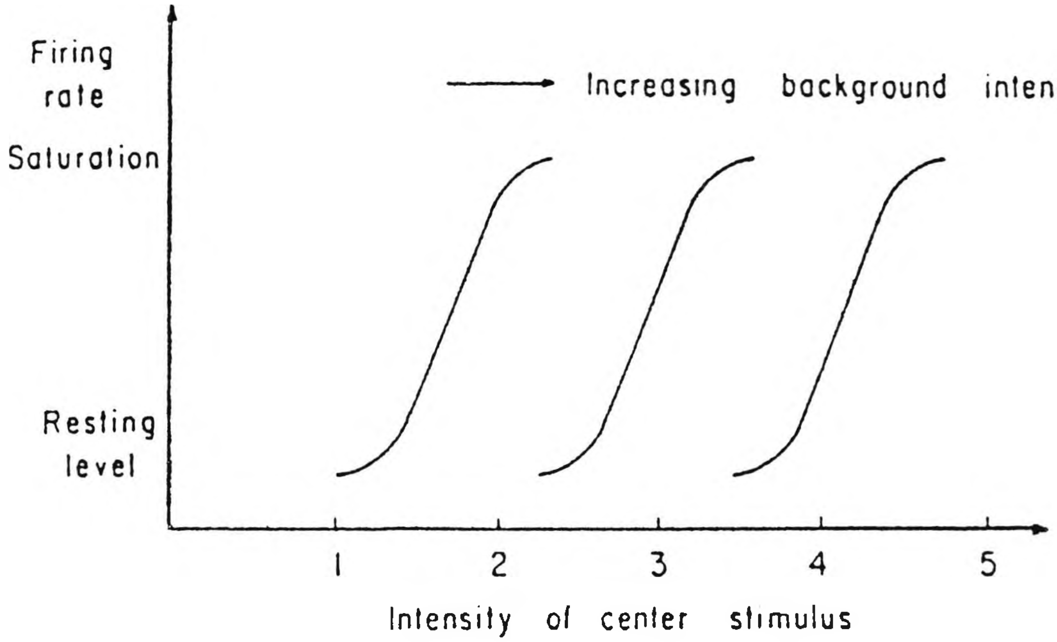


Figure 3.11: *Firing rate as a function of stimulus intensity for several background intensity levels* (after [Sakrison 1977]).

resistance (increases the permeability) of the membrane of a cell lying next to the receptor. This decrease in resistance in turn causes a change in the voltage between the inside and the outside of the cell. This change in the voltage is related to the frequency with which the cell fires nerve impulses.

Rushtone [Rushtone 1961] has pointed out that the resistance of the cell membrane is proportional to the logarithm of the total light incident on the receptor. Furthermore, he concluded that the relationship between the membrane potential and the frequency of nerve impulses is a linear one thus, the frequency is a logarithmic function of light intensity. These findings are summarised in Fig. 3.12 which is a plot of the change in the resistance of the cell membrane or the impulse frequency against the logarithm of the light intensity. In Fig. 3.12, R_0 is the resistance of the cell membrane in the dark, R_1 is the resistance under illumination, and $\log I$ is the logarithm of the light intensity.

Stevens [Stevens 1970] has replotted Hartline's data on a log-log scale and pointed out that the functional relationship follows a power law with an exponent of 0.29. Stevens has also reported that the exponent for perceptual brightness of 5 deg targets in the dark is 0.33 [Stevens 1975]. This number represents all of the nonlinearities in the system since the experimental paradigm now includes

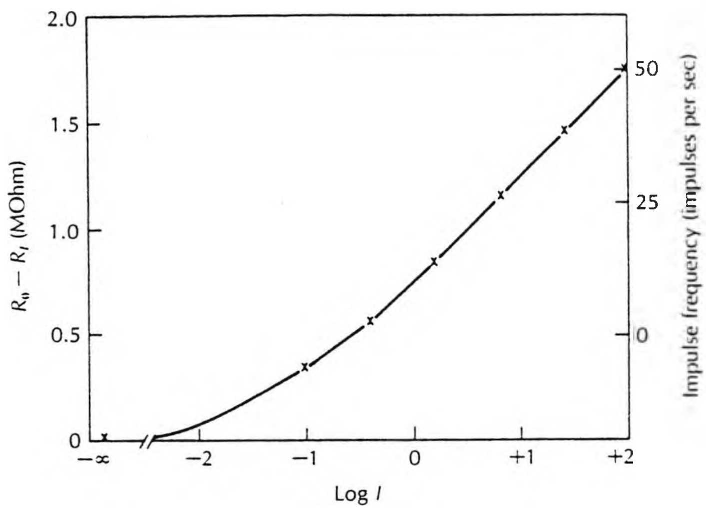


Figure 3.12: *The effects of different light intensities upon a neuron in the eye of the Limulus* (after [Rushtone 1961]).

the perception of the stimulus and the response is not just the monitoring of a neurological signal. In other words, the 0.29 exponent is based on physiological data while the 0.33 exponent is based on psychophysical data. Utall, in discussing the coding of sensory magnitude, has concluded that the main site of response compression occurs at the receptors [Utall 1973], and the above data indicate this is true. Hence, it seems that the nonlinearity in the HVS can be approximated by a logarithmic function or a power law function.

From the discussion in this section, it is quite evident that significant processing takes place in the retina. This may be thought of as a preprocessing of the visual information prior to encoding and transmission to higher centres which will be described in the following section. It is also evident that the information that we have about the visual world, and our perception of objects and visual events in the world, depend only indirectly upon the state of that world. They depend directly upon the nature of the images formed on our retina. Since the eye is not a perfect sensor of visual signals, the retinal images are different in many important ways from the real objects. Finally, it should be noticed that if information about some aspect of an object is not contained in the retinal image of it, the information can not be recovered by later parts of the visual system [Cornsweet 1970].

3.3 The Visual Path from the Eye to the Cortex

The output from each eye is conveyed to the brain by about a million nerve fibers bundled together in the optic nerve. These fibers are the axons of the ganglion cells of the retina. Fig. 3.13 is a schematic representation of the visual pathways beyond the retina.

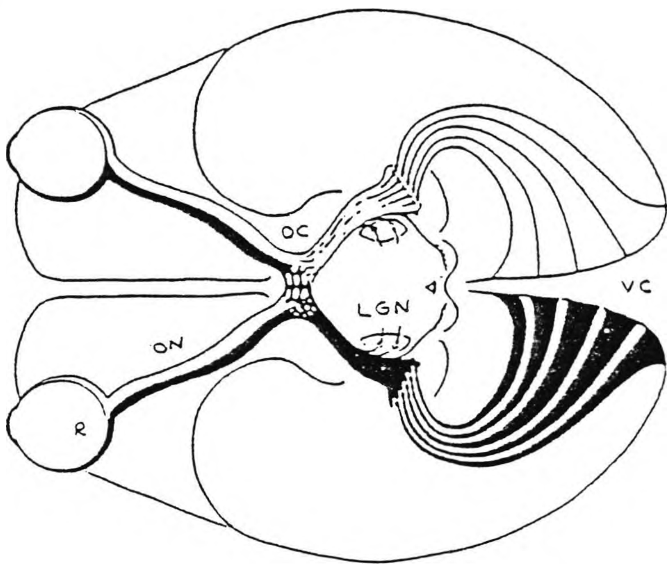


Figure 3.13: *Pathways in the human visual system.*

Shown are the retina (R), optic nerve (ON), the optic chiasm (OC), the lateral geniculate nucleus (LGN) and the visual cortex (VC). The black and white paths indicate mapping of the right and left visual fields, respectively. It should be noticed that the fibers of each optic nerve split up at the optic chiasm according to the half-retina from which they originate. Fibers from the right half-retinas go off to the lateral geniculate nucleus on the right side of the brain, and fibers from the left half-retinas go to the lateral geniculate nucleus on the left side of the brain. Consequently, the information content of the left half of the visual field is processed by the right side of the brain and the information content of the right half of the visual field is processed by the left side of the brain.

At the LGN the optic nerve fibers map out the appropriate half retina from the two eyes in a regular manner. They terminate on geniculate neurons which in turn send axons to the *primary visual cortex* (also known as the *striate cortex* or area 17). From there, after several synapses, the visual messages are sent to a number of further destinations: neighbouring cortical areas (areas 18 and 19) and also several targets deep in the brain. Feedback paths from the visual cortex to the lateral geniculate nuclei have also been observed [Hubel 1979]. Thus, it should be understood that the primary visual cortex is in no sense the end of the visual path. It is just one stage, probably an early one in terms of the degree of abstraction of the information it handles.

The cellular analysis of the LGN shows a layered organisation of cells. Each geniculate body is divided into six layers, three left-eye layers interdigitated with three right-eye ones. Cells on each layer receive information only from one eye, a characteristic called *ocular dominance*. Moreover, neurons receiving information from a given area of the retina are grouped, independently from the specific layer. In other words, the receptive fields of all the cells encountered along a radial pathway traversing the six layers have virtually identical position in the visual field.

The cells in the LGN function in a way very similar to that of ganglion cells maintaining independence of orientation, i.e., the information is processed independently from spatial orientation. In addition, the receptive fields of single geniculate cells are similar to those of retinal ganglion cells but with a stronger inhibitory surround, which improve further contrast enhancement [Hubel 1961].

Located at the back of the brain, the visual cortex is a folded layer of neurons of about 2 mm in thickness. Cellular analysis of a cortical tissue under a microscope shows that the cortex as a whole contains some 10^{10} (10 billion) neurons. The neurons are hierachically organised in half a dozen layers that are alternatively cell-sparse and cell-rich. In contrast to these marked changes in cell density in successive layers at different depths in the cortex, there is marked uniformity from place to place in the plane of any given layer and in any direction within that plane.

The neurons are classified as *simple cells*, *complex cells*, and *hypercomplex cells*. The receptive field of a simple cell is essentially identical to the field of a ganglion cell, except that it is elongated (see Fig. 3.14). Any given simple cortical cell will respond to spots of light anywhere within an elongated area whose long axis may have any particular angular orientation, and it may be of the "on-centre" or the "off-centre" type. It is at this level of the visual system that specific processing for a given spatial orientation is introduced.

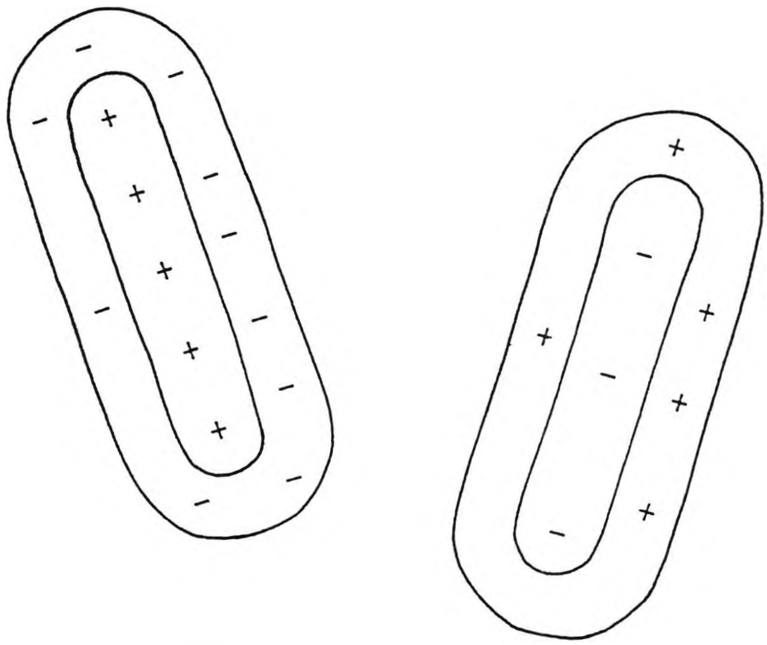


Figure 3.14: *Receptive field of a simple cell* (after [Kuffler 1976]).

Complex cells are also sensitive to the orientation of the excitation and so are hypercomplex cells. However, hypercomplex cells respond most effectively when corners or discontinuities are present in the visual field.

A systematic, cell-by-cell, analysis of the visual cortex indicates a columnar organisation in both ocular dominance and orientation dominance. In one typical vertical penetration of the cortex (labeled "1" in Fig. 3.15) a microelectrode encounters only cells that respond preferentially to the left eye (L_r) and, in layer IV, cells that respond only to the left eye (L). In another vertical penetration (labeled "2") the cells all have right-eye dominance (R_l) or, in layer IV, are driven exclusively by the right eye (R). In an oblique penetration (labeled "3") there is a regular alternation of dominance by one eye or the other eye. Repeated penetrations suggest that the cortex is subdivided into regions with a cross-sectional width of about 0.4 mm and with walls perpendicular to the cortical surface and layers. These regions have been called the *ocular dominance columns.*

It is interesting to note that if the same experiment is repeated by looking, not for the ocular dominance, but the response to a given orientation, the same

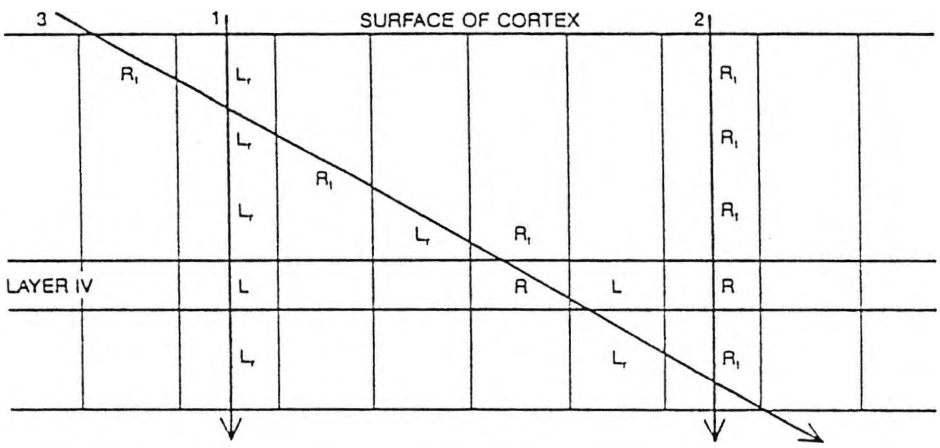


Figure 3.15: *Grouping of cortical cells according to ocular dominance* (after [Hubel 1979]).

columnar organisation can be observed (see Fig. 3.16). A microelectrode penetrating the cortex perpendicularly encounters only cells that prefer the same orientation (apart from the circularly symmetrical cells in layer IV, which have no preferred orientation). In two perpendicular penetrations a millimeter apart, however, the two orientations observed are usually different. Huble and Wiesel have found that each time the microelectrode moved forward as little as 25 to 50 micrometers in parallel direction to the surface of the cortex the optimal orientation changed by a small step, about 10 degrees on the average. The steps continued in the same direction, clockwise or counterclockwise, through a total angle of anywhere from 90 to 270 degrees. Occasionally such a sequence would reverse direction suddenly (as can be noticed in Fig. 3.16), from a clockwise progression to a counterclockwise one or vice versa.

The visual cortex must therefore be subdivided into roughly parallel slabs of tissue, with each slab, called an *orientation column*, containing neurons with like orientation preference. Combining the results of these two experiments, a columnar model of the cortex is obtained, as shown in Fig. 3.17, where the bars in the columns indicate the preferred orientation.

Given what has been learned about the primary visual cortex, it is clear that a block of cortex about a millimeter square and two millimeters deep can be considered an elementary unit of the primary visual cortex. It contains one set of orientation slabs subserving all orientations and one set of ocular-

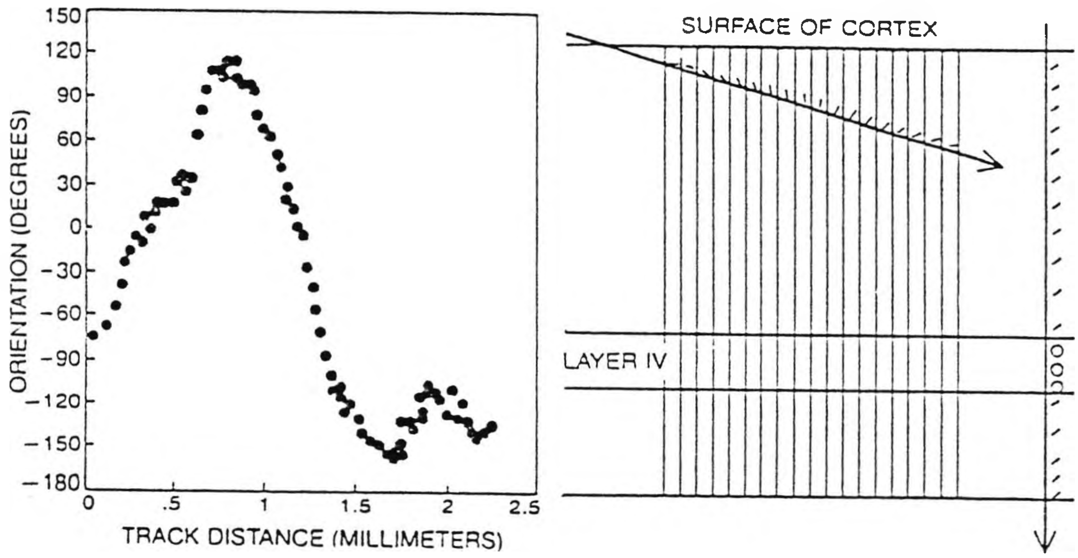


Figure 3.16: *Grouping of cortical cells according to orientation preference* (after [Huble 1979]).

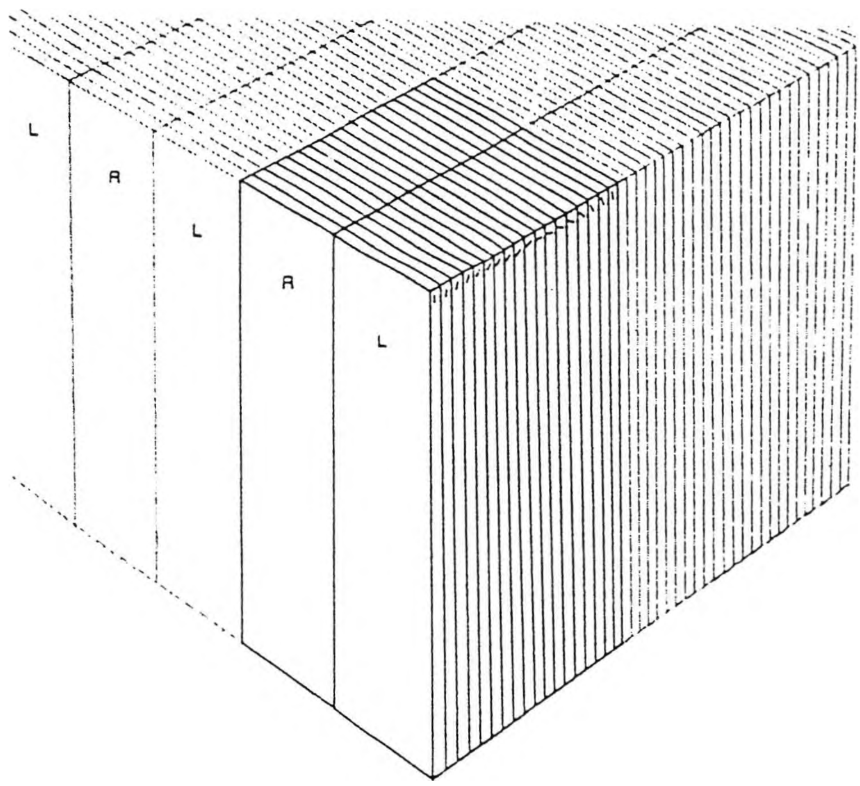


Figure 3.17: *Columnar model of the cortex* (after [Huble 1979]).

slabs subserving both eyes. Hubel and Wiesel have summarised the subject properly stating that "To know the organisation of this chunk of tissue is to know the organisation for all of area 17; the whole must be mainly an iterated version of this elementary unit" [Hubel 1979].

All the properties described thus far can be summarised in a block diagram (see Fig. 3.18) where parts related to the lens, the retina, and the cortex are indicated.

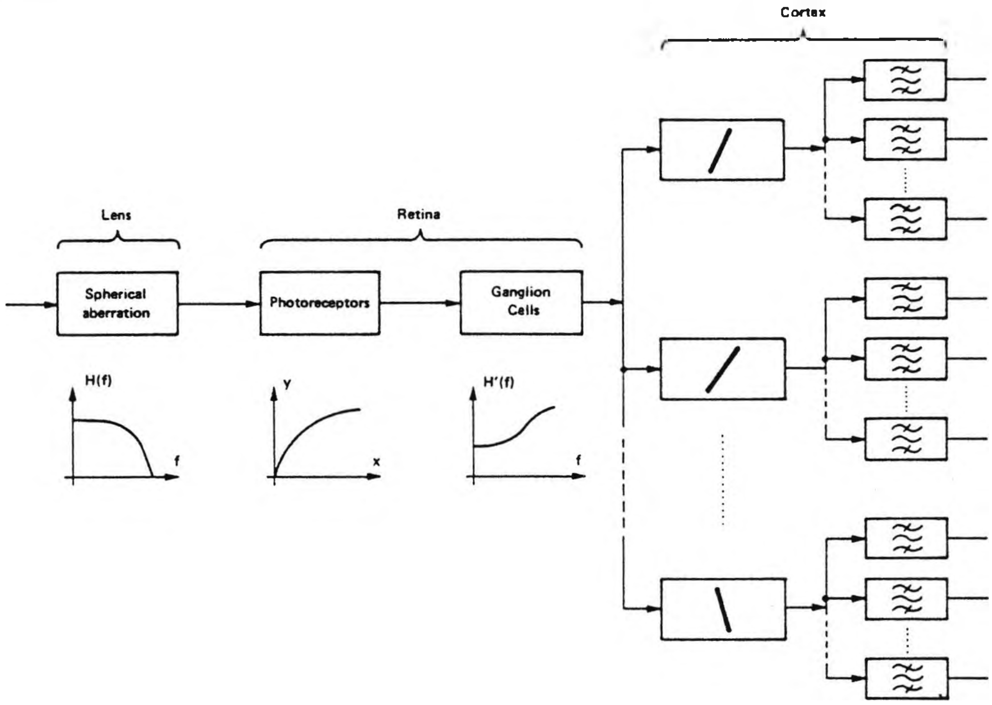


Figure 3.18: *Block diagram of the HVS* (after [Kunt 1985]).

The first block is a spatial, isotropic, low-pass filter. It represents the spherical aberration of the lens, the effect of the pupil, and the frequency limitation by the finite number of photoreceptors. It is followed by the nonlinear characteristic of the photoreceptors. Here, a logarithmic curve for simplicity or a law of the type L^{τ} for more accuracy, can be used. At the retina, this nonlinear and very likely memoryless transformation is followed by an isotropic high-pass filter corresponding to the lateral inhibition phenomenon of the ganglion cells and LGN cells. Finally, there is a directional filter bank that represents the processing performed by the cortical cells followed by another filter bank for detecting the intensity of the stimulus.

3.4 Psychophysics of Vision

In this section we consider the second principal source of information about the visual system - psychophysics. Various perceptual effects, which have been verified through psychophysical experiments, will be described emphasising those which are relevant to our efforts to devise better image coding systems.

In psychophysics an observer is shown a stimulus and then asked to make some response indicating his sensation of some attribute of the stimulus. The visual system is thus treated as a "black box" with visual stimuli as the input and perceived sensations as the output. Functional relationships are sought between physical properties of the stimulus and the subjective response of the observer. Since we are concerned with black and white images our interest is restricted to *brightness*, the perceived attribute of luminance. Other sensations aroused by visual stimuli like hue and saturation, which are attributes of color, will not be dealt with herein.

Most of the studies in perception have been carried out only for *threshold perception*, which refers to the perception of "just-noticeable" levels of noise (perturbation). The reason is that responses to suprathreshold perturbations are difficult to standardise and therefore general conclusions cannot be drawn. The earliest experiments were carried out in uniform background fields with circular test spots of varying area and duration. Based on these studies, a number of laws of threshold vision have been empirically derived.

The first law, known as Ricco's law, states that for small test spots, less than 2 minutes of arc in diameter, the luminance difference threshold is inversely proportional to the stimulus area. As the size of the test stimulus is increased beyond 2 minutes of arc, the degree of spatial summation falls off gradually and eventually reaches a point where the threshold is virtually independent of the stimulus area. Analogous to Ricco's law for spatial summation is Bloch's low for temporal summation. It states that for stimulus duration up to about 40 ms the luminance difference threshold is inversely proportional to the stimulus duration. For longer durations the temporal summation falls off in the same way as the spatial summation.

The variations of threshold with background luminance depends on the size of the stimulus and its duration. Over a wide range of luminance, and in particular over the range normally encountered in television, there are two limiting modes of behavior. For small stimuli of short duration (size less than 5 minutes of arc and duration less than 10 mSec) the thresholds are proportional to the square root of the background luminance (DeVries-Rose's law). For large stimuli and long duration the thresholds are directly proportional to the background luminance (Weber's law). The point of transition between the two modes of behavior gets progressively lower with increasing background luminance. In this way Weber's law covers an increasing range of stimulus size and duration as the background luminance is increased. The ratio between the threshold level and the background intensity level is very nearly a constant except for very low and very high levels of background intensities. It is called the *Weber fraction* and its value is about 0.02 [Pratt 1978].

The visibility of a perturbation as a function of its spatial frequency has been an important issue in the study of perceptual effects. In particular, researchers have tried to find a simple model that would describe the response of the HVS to stimuli of various spectral content. Linear systems, or nonlinear systems that are operating in a linear range, can be described by a curve (function) that is called the *modulation transfer function* (MTF) of the system. Once the MTF is specified, the response of the system to any stimulus can be computed by applying Fourier techniques. Unfortunately, physiological studies have confirmed that the human visual system contains nonlinearities (see previous discussion about the retina). Thus, simple ways in which these nonlinearities can be added to a model of the HVS have been sought. One simple way is shown in Fig. 3.19. It is assumed that after a logarithmic transformation which takes place at an earlier stage of the HVS, the rest of the system is linear, at least for constant background luminance.

Many researchers have thus measured the human threshold contrast sensitivity to periodic patterns (sine- waves, square-waves, etc.) viewed at a range of spatial frequencies (good reviews of this work can be found in [DePalma 1962], [Levy 1970] and [Kelly 1977]). By taking the reciprocal of such a contrast sensitivity curve, one arrives at a curve akin to the spatial frequency response function of the visual system. This curve has been called the MTF of the visual system.

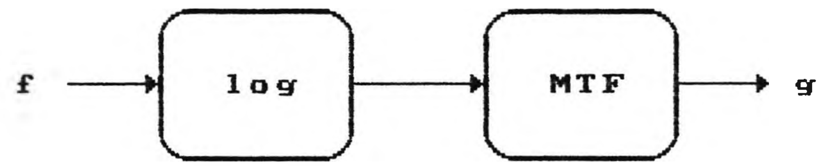


Figure 3.19: *A simple nonlinear model of the HVS.*

It has been shown that for threshold stimuli the visual system can be modeled by a band-pass filter with maximum response in the range of 2-8 cycles per degree (cpd) falling off at lower and higher frequencies (see Fig. 3.20). The lower cutoff is due to the lateral inhibition phenomenon described earlier, and the upper cutoff is due to the finite aperture of the pupil, the size and density of the receptors, and the scattering of light within the eye. The band-pass filter effect obtained with threshold stimuli becomes less pronounced as the stimuli become more visible (suprathreshold stimuli) [Hammerly 1977].

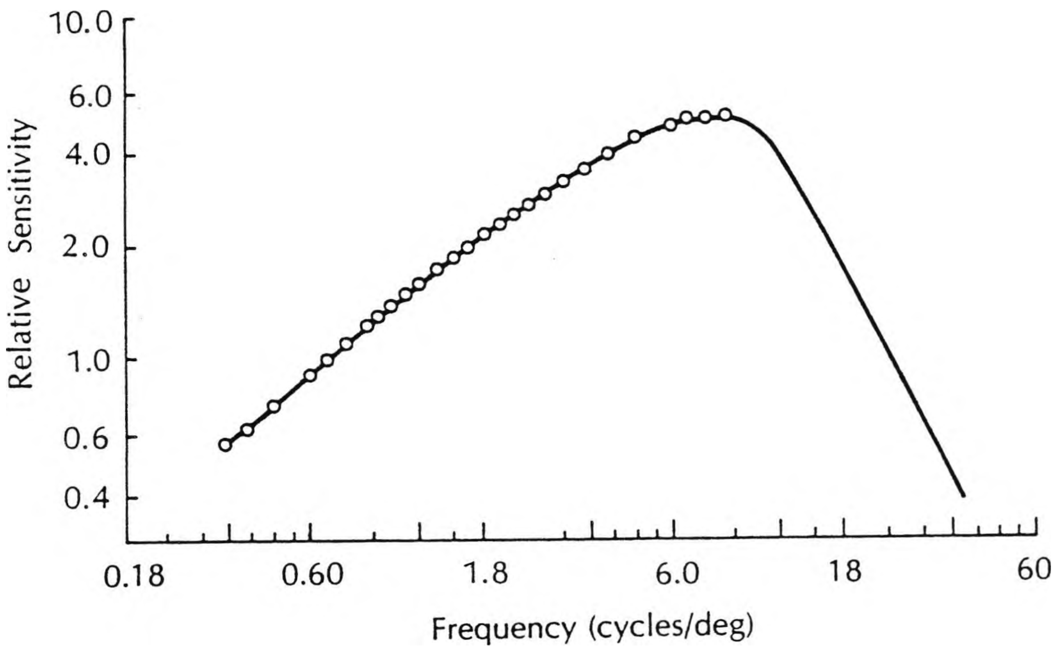


Figure 3.20: *Spatial frequency response of the HVS* (after [Cornsweet 1970]).

Two points should be taken into consideration when trying to employ the model described above. First, it should be remembered that the HVS is not isotropic, i.e., the response of the HVS to a rotated contrast grating is a function of the grating's frequency as well as its angle of orientation. It has been shown that the sensitivity of the HVS is greatest and nearly equal for vertical or horizontal gratings and decreases to a minimum at 45 deg where the sensitivity is about 3 decibles (dB) less (at a frequency of 30 cpd) [Hall 1977]. However, the effects of this anisotropy are relatively small over much of the frequency range to which the eye is sensitive (for spatial frequencies equal to or less than 10 cpd) thus, for many applications, a single MTF is sufficient [Cornsweet 1970, p. 330].

Secondly, the HVS is not spatially homogeneous, neither in optics nor receptors. For example, it has been mentioned earlier that the densities of cones and rods vary strongly with retinal position. Nevertheless, optical spatial invariance is a good assumption near the optic axis, and thus, the MTF of the HVS may be used to find the properties of retinal images [Cornsweet 1970, p. 329].

The structure shown in Fig. 3.19 can also be used to explain the psychophysical phenomenon of *brightness constancy* [Cornsweet 1970, pp.353-355] which will be explained next. Fig. 3.21*a* depicts the intensity profiles of two hypotetical images. The difference between the two profiles is that the intensity of the right-hand profile is six times greater everywhere. A human observer would not detect this difference, and in fact, the two images would be perceived as having the same brightness. This phenomenon is called brightness constancy.

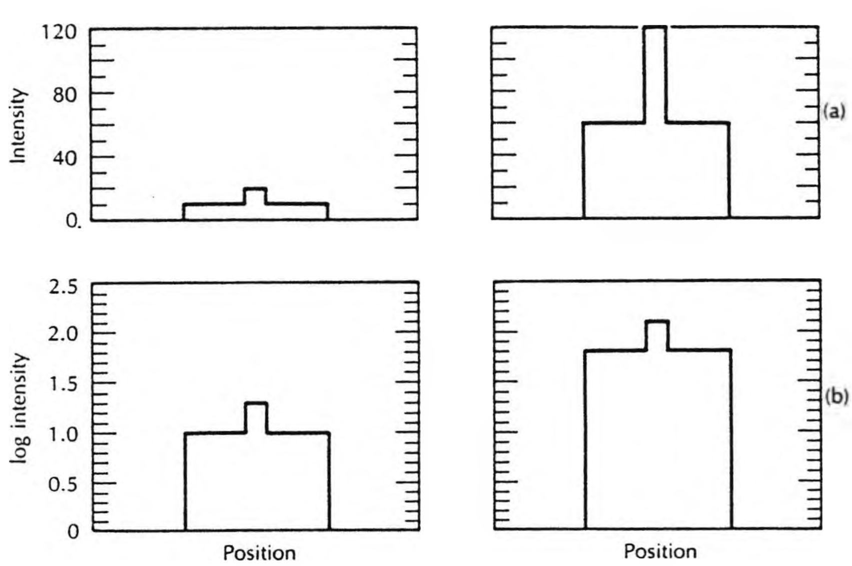


Figure 3.21: *The paradigm for brightness constancy* (after [Cornsweet 1970]).

When a logarithmic transformation is assumed to take place at an earlier stage of the HVS, the inputs to the rest of the system (see Fig. 3.21*b*) differ only by the average intensity. It should be noticed that the step in intensity is equal in amplitude for both intensity profiles. After high-pass filtering, this constant amplitude step is the only thing that passes through the system and thus the two outputs should be very similar. In other words, a combination of a logarithmic transformation and the measured MTF of the visual system lead to the prediction that brightness constancy should occur. Stockham has successfully applied a version of this model to image processing [Stockham 1972]. In addition, Mannos and Sakrison have found the same basic model to be appropriate for image coding [Mannos 1974]. They have also found that images which were coded using a cube root function were judged subjectively as being "better" than those coded via a logarithmic function.

The last perceptual effect to be discussed hereby is localized in the spatial domain. It is well known that while television pictures typically may contain areas that are uniform or quasi-uniform, they also contain detail and large changes of luminance (edges) which affect drastically the visibility of any impairments. These changes inhibit the ability of the eye to detect impairments spatially adjacent to the changes. This phenomenon is referred-to as *spatial visual masking.*

Spatial masking has been studied by means of a number of paradigms. The foremost of these has been to measure increment thresholds for small test stimuli (a thin line parallel to the edge) as a function of distance from a luminance edge. Using this paradigm it has been found that for edges of high contrast, thresholds rise sharply on both sides of the edge and reach a peak at the edge itself [Fiorentini 1966]. The magnitude of the effect increases with the contrast ratio; with a contrast ratio of 20:1 and a line test stimulus 1 minutes of arc in width, the thresholds measured at the edge were up by a factor of between 3 and 4 relative to the steady-state values on the bright side of the edge [Lukas 1980]. The spread of the effect is very narrow, though, most of it occurring within 6 minutes of arc of the edge on either side.

The visibility of the noise is defined to be inversely proportional to the level of just-visible noise. The visibility as a function of the distance from the edge has been modeled on the basis of psychovisual experiments [Hentea 1984] as an exponential function. The visibility function $vi(x)$, where x is the distance from the edge, is given by

$$vi(x) = 1 - a^{|x|} . \tag{3.1}$$

The constant a lies between 0.0 and 1.0, and its value depends on the ratio of the bright and dark intensities of the edge [Hentea 1984]. The higher the ratio, the larger the value of a. This visibility function, also called *spatial masking function*, is depicted in Fig. 3.22.

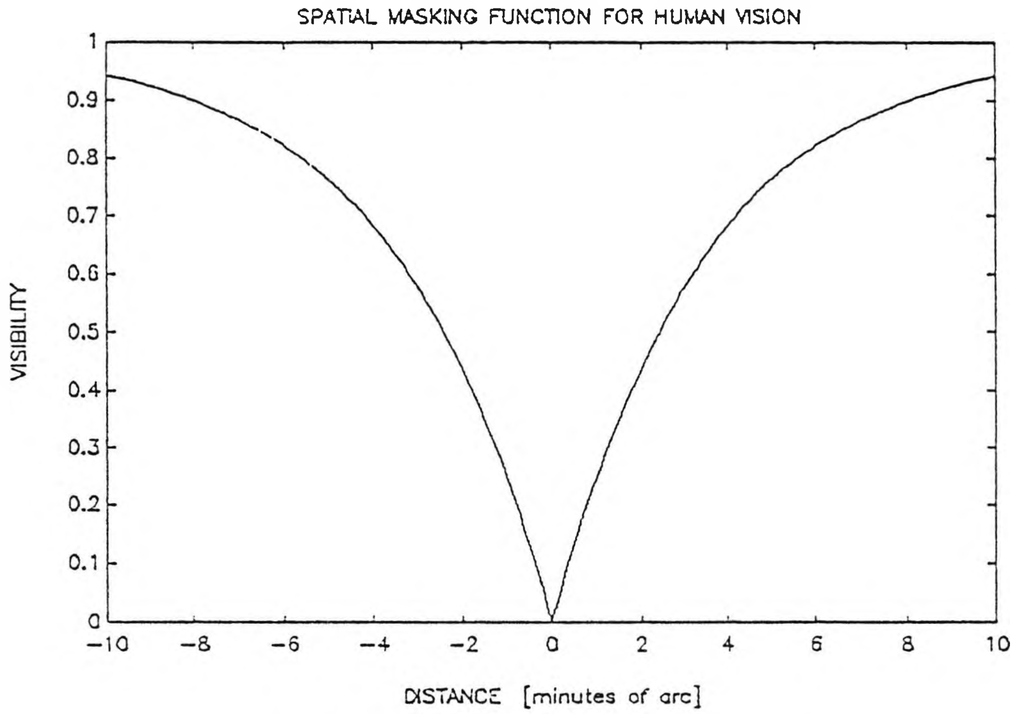


Figure 3.22: *Spatial masking function (for a=0.75).*

One further approach to the study of masking has been in terms of the spatial frequency domain. This has been done by experiments with gratings where narrow-band noise centred at various spatial frequencies has been presented against a narrow-band background [Sakrison 1975], [Stromeyer 1972]. It has been found that the visibility of the noise is lowest around the centre frequency of the background and it increases for other background frequencies. Thus, the visibility of noise is masked most effectively in those areas of the picture where it has similar spatial frequency content to the picture itself. High-frequency noise is least visible in the high-detail areas of the picture and low-frequency noise is least visible in the homogeneous low-detail areas. This important conclusion will be used in devising ways for post-processing of block coded images.

With that we complete our short review of the human visual system. The perceptual features listed above are the most important ones from the coding point of view. All of them ought to be exploited by any coder in order to distribute the noise (coding distortion) in a manner that is least visible. All of them will indeed be used to devise a new coding system which is the purpose of this research.

Chapter 4

Directional Classified Gain-Shape Vector Quantisation

See the sun in the midst of the rain;
Scoop clear water from the heart of the fire.

4.1 Introduction

A new VQ-based system for image coding, named *Directional Classified Gain-Shape Vector Quantisation* (DCGSVQ), is introduced in this chapter. It combines vector quantisation with *transform coding* (TC) techniques and exploits various properties of the human visual system (HVS) to produce reconstructed images with good subjective quality at low bit-rates.

Transform coding has been widely used, and shown to be an effective approach for image data compression. In TC, blocks of statistically dependent picture elements (pixels) are converted into blocks of relatively less correlated elements so that *statistical redundancy* can be removed more efficiently. Among the many transforms commonly used, the discrete cosine transform (DCT) has become widely recognised as an almost optimum transform method when compared with other transforms on the basis of energy compaction and decorrelation between pixels [Ahmed 1974, Perkins 1988]. In particular, it has been shown that the DCT is a limiting case of the Karhunen-Loéve (KL) transform, which is an optimum transform for image compression in the mean square error sense [Shanmugan 1975, Ahmed 1982]. Unlike the KL transform, which requires *a priori* knowledge of the source correlation for implementation (information which is seldom available), the DCT is a deterministic transform not requiring such knowledge. In addition, the all-real cosine transform can be rapidly computed by applying fast DCT algorithms [Chen 1977], [Makhoul 1980], [Haque 1985].

The cosine transform can be applied to the entire image or to sub-images of various sizes (typically blocks of 8×8 or 16×16 pixels). Applying transforms to sub-images rather than to the entire image as a whole has the advantage of reducing computational complexity, and of allowing adaptivity to different scene parts. Therefore, the general method of DCT coding involves dividing the image into small non-overlapping sub-images, i.e., blocks of pixels, and then transforming the blocks to obtain equal-sized blocks of transform coefficients. Since transformation has the effect of compacting most of the energy within the block into some of the low-frequency coefficients, data compression can be achieved by quantising and transmitting (or storing) only a subset of these coefficients. Usually, some high-frequency coefficients may be discarded, employing either *zonal sampling* or *threshold sampling* techniques [Wintz 1972] without causing significant distortion in the reconstructed image. The remaining coefficients are then independently quantised using a fixed nonuniform scalar quantiser, and transmitted. This system is a *nonadaptive coder* where the quantisation levels are defined by a bit assignment matrix, which is either available at the receiver or transmitted for each image as side information [Chen 1977a].

In *adaptive transform coders*, several bit assignment matrices are employed. Each block (sub-image) is classified into one of several classes according to the activity content of the block and then coded by the corresponding bit assignment matrix. Various *activity indices* have been defined either in the spatial domain or in the transform domain (the one most commonly used has been the total AC energy, calculated in the transform domain [Chen 1977a]). As a rule, normalisation of the coefficients has to be carried out to reduce their dynamic range, and then each coefficient is nonuniformly quantised and coded. The scalar quantisers are designed using the Max algorithm [Max 1960] assuming a well-defined probability density function for the transform coefficients.

Various probability density functions have been suggested. A Gaussian distribution has been assumed in reference [Chen 1977a] for all the transform coefficients, where in reference [Reininger 1983] a Laplacian distribution has been proposed for the AC coefficients while proposing a Gaussian distribution for the DC coefficient. In [Eggerton 1986], however, it has been stated that the one overall density function is much closer to being Cauchy than Gaussian or Laplacian.

Despite the fact that this method performs well on most natural scenes, and a coding rate of about 1 bits per pixel (bpp) is achievable with no apparent visual degradation, this method suffers from three major drawbacks. First, a specific probability density function, for describing the distribution of the AC coefficients, has to be defined in order to design the Max-quantisers. Since natural scenes vary enormously it has been quite difficult to agree upon a standard density function, thus leaving this matter open to dispute. Secondly, the quantisation process, which is necessary and can also contribute significantly to data reduction, is performed in a scalar manner. In a Shannon sense [Shannon 1948, Berger 1971], such a procedure is inherently sub-optimal. Improved performance in terms of rate-distortion function can be achieved if a sequence of selected transform coefficients are quantised simultaneously, i.e., as a vector.

The third drawback is that this method is based on classical information theory without considering valuable properties of the intended receiver, i.e., the human observer. For example, the sensitivity of the human visual system (HVS) to errors in the reconstructed image depends on the frequency spectrum of the error, the mean grey level, and the amount of detail in the picture in the vicinity of the error. Hence, it is possible to increase the efficiency of the coder by allowing distortions that do not degrade the subjective quality of the reconstructed image. In other words, "image coding must become a discipline not only of exploiting redundancy and mathematical information measures, but of visual science - understanding exactly what image information is made use of in perception" [Chen 1990].

The first step in the right direction has been the design of vector quantisers, firstly in the spatial domain and recently in the transform domain. Various schemes have been proposed that combine the good features of TC with the advantages of VQ to increase the coding performance [Sun 1985], [Ang 1986], [Marescq 1986], [Breeuwer 1988], and [Ho 1989]. In these schemes, activity measures (indices) have been used, defined either in the spatial domain or in the transform domain, for adaptive selection and quantisation of the transform coefficients in order to take into account the local statistics. The transform blocks are classified into several *activity classes*, each having its own bit assignment matrix. Each block is adaptively partitioned into a number of vectors which are then quantised employing VQ. However, since the HVS is highly sensitive to errors in the mean luminance, these schemes have separately quantised the DC coefficients employing a scalar quantiser.

Pointing out the major difficulties that arise in attempting to design a universal VQ in the spatial domain, i.e., one that would be good for encoding various images, the authors in [Saito 1987] have proposed the following notion. They argued that if the image signal can be transformed into a signal sequence with a standard probability distribution, then a universal VQ can be designed to match that distribution. Moreover, if a spherically symmetric probability model can be used, then a GSVQ can perform as well as an optimum VQ. Thus, they designed a GSVQ in the cosine domain based on the basic design principles laid forth in [Sabin 1984].

The GSVQ was designed for a Pearson-type VII distribution which is a spherically symmetric density function. They argued that the Pearson-type VII distribution is a more suitable model for the normalised AC coefficients than the Gaussian or the Laplacian distributions provided that a certain parameter, in that probability density function, is properly selected. Several vectors with various dimensions were defined by grouping AC coefficients of approximately equal variance values into one vector. These vectors were normalised and coded by several shape codebooks that were designed with a training sequence of shape vectors, derived from a random source with the assumed distribution function. The gain values were coded by a nonuniform Max-type scalar quantiser, optimised for the assumed distribution function. The DC coefficient was coded by an 8-bit uniform scalar quantiser. The coding results were better than the conventional DCT coding scheme [Chen 1977a] however, they were slightly inferior compared to a 16-dimensional VQ operating in the spatial domain. So, to improve the coding performance, an adaptive version of the above scheme has been proposed in [Saito 1988].

The systems mentioned above have achieved good coding results but they share several basic drawbacks. First, no systematic method of designing the various codebooks has been proposed, i.e., the size of each of the codebooks was determined empirically. In [Marescq 1986] for example, the AC coefficients were all grouped into one vector and employing the LBG algorithm and the *splitting* method [Linde 1980], codebooks were designed for each of the various classes. The size of each of the codebooks was determined empirically by increasing the number of the codevectors until the training image blocks in each class were coded with no visible distortion. The same design problem has been encountered in [Breeuwer 1988].

Secondly, a specific distribution function has to be used in order to describe the distribution of the transform coefficients. As mentioned earlier, various distribution functions have been proposed, including the absurdity of fine-tuning a parameter of such a function to the image under test [Saito 1987], and it seems that this matter will still be open to further dispute. Thirdly, no attempt has been made to incorporate well-known properties of the HVS into the above coding schemes. One could intuitively expect that if a suitable model of the HVS could be successfully combined with the compression process, an improvement in compression performance would result. More specifically, the removal of *subjective redundancies* [McLaren 1991] from DCT-coded images, through psychovisual thresholding and quantisation, can result in better reconstructed images.

Image coders that take the characteristics of the HVS into account have been called *psychovisual coders* [Huang 1966] or *psychophysical coders* [Schreiber 1967]. Several such coding schemes have been proposed, employing certain psychovisual properties of the HVS like frequency sensitivity [Sakrison 1977, Eggerton 1986, Ngan 1989], luminance dependence [McLaren 1991], and masking effects [Netravali 1977, Ngan 1989]. In addition, a new class of coding schemes, termed *second generation* has been proposed recently [Kunt 1985]. These schemes have included hierarchical algorithms imitating the multiple channel characteristic of visual receptive fields and methods for contour-texture detection and description.

To tackle the various drawbacks, mentioned above, a new image coding system, termed *directional classified gain-shape vector quantisation* (DCGSVQ), is introduced in this chapter. A new algorithm for designing the various codebooks, needed for the DCGSVQ, is proposed based on the *classified nearest neighbour clustering* (CNNC) algorithm described in chapter 2 and in [Kubrick 1990]. In addition, an optional simple method for feature enhancement, based on inherent properties of the proposed system, is proposed enabling further image processing at the decoder.

4.2 DCGSVQ - System Description

Fig. 4.1 depicts the basic block diagram of the proposed DCGSVQ system. The various stages are described in detail in the following subsections.

4.2.1 Cosine Transformation

The input image is divided into non-overlapping blocks of NxN pixels ($N = 8$) and each is transformed employing the DCT. The transformation is defined by

$$F(u,v) = \frac{4Q(u)Q(v)}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \cos\left[\frac{(2i+1)u\pi}{2N}\right] \cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (4.1)$$

for $u, v = 0, 1, \cdots, N-1$ where $f(i,j)$ denotes the image signal in the spatial domain, and

$$Q(r) = \begin{cases} \frac{1}{\sqrt{2}}, & r = 0 \\ 1, & r = 1, 2, \cdots, N-1\ . \end{cases}$$

The inverse transform is given by

$$f(i,j) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} Q(u)Q(v)F(u,v) \cos\left[\frac{(2i+1)u\pi}{2N}\right] \cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (4.2)$$

for $i, j = 0, 1, \cdots, N-1$.
The cosine transform is used for its superior properties, as described in the introduction of this chapter.

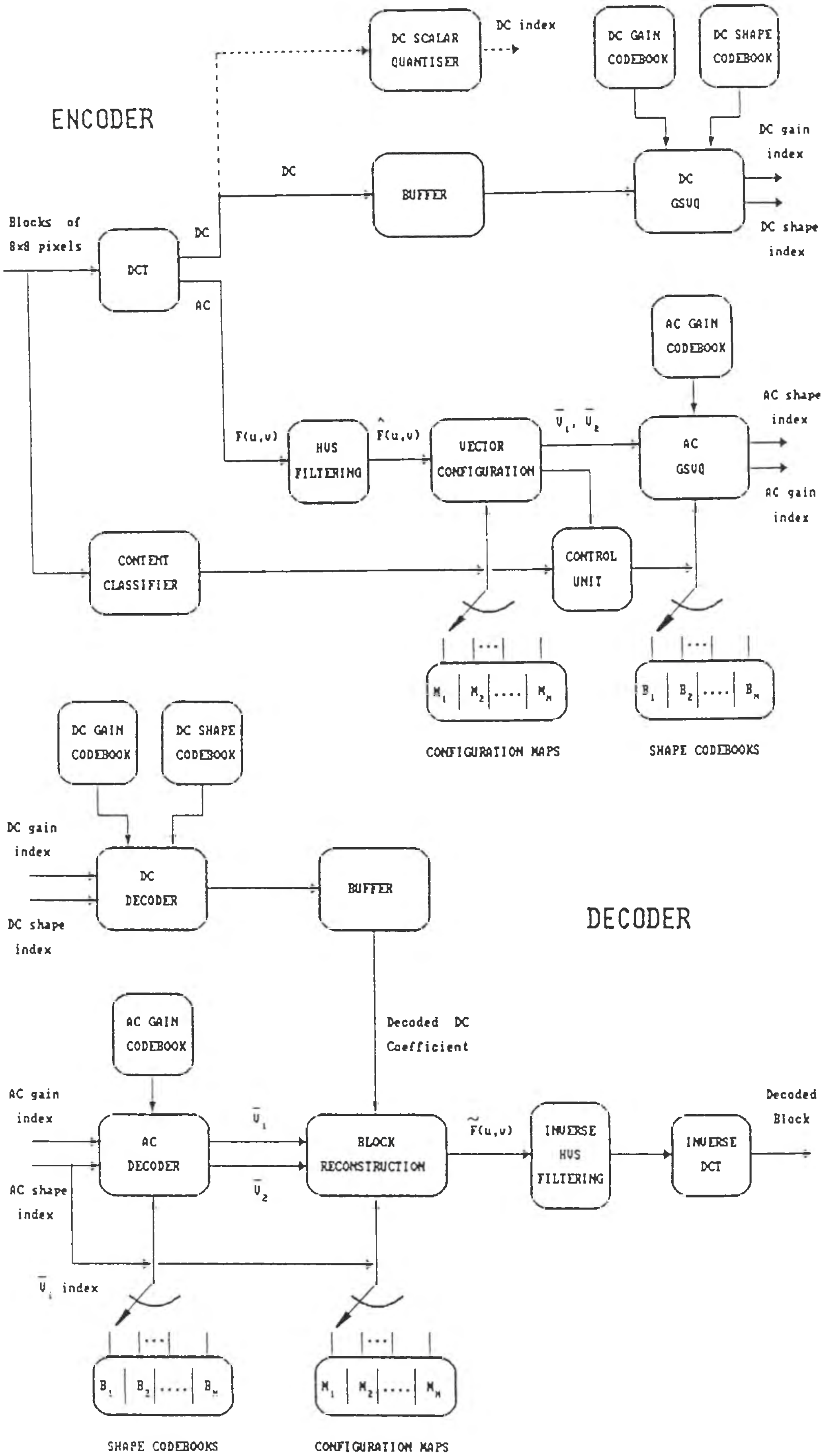


Figure 4.1: *DCGSVQ - Block Diagram*

4.2.2 Content Classification

Prior to carrying out the DCT, each block is analyzed by a *content classifier*. The classifier allocates each image block into one of seven classes which represent various image patterns (edges in various directions, monotone areas, complex texture, etc.). Accordingly, the classifier controls two switches that define which *vector configuration map*, and which shape codebooks should be used in the cosine domain for encoding the block under test.

The content classifier is similar to the one described in [Ramamurthi 1986] and [Aravind 1986] with several changes which result in more accurate decisions about the blocks' content. The classification algorithm is implemented in two steps: an edge enhancement step, followed by a decision tree which extracts the edge description from the enhanced version. The seven classes are defined as follows :

a) A *shade* class, consisting of monotone image blocks which contain no significant gradients.

b) A *midrange* class, consisting of image blocks which contain moderate gradients but no definite edges.

c) Four *edge* classes consisting of image blocks having a distinct edge running through them, and

d) A *mixed* class consisting of all the complex structured image blocks. These blocks contain fine details or complex edges which can not be treated as simple edges.

The distinction between shade and midrange blocks is based on the well-known fact that intensity changes against a uniform background are visible only if they are greater than the Weber fraction.

Four edge orientations have been defined: horizontal, vertical, and two diagonals. This definition is in accord with findings presented in [Keskes 1979]. There, an edge orientation histogram accumulated on 11 real images has shown quite clearly that horizontal and vertical edges are more probable than edges in other orientations. In addition, distinct peaks of the histogram have been found in 45 and 135 degrees for particular images. These observations support the classification of image blocks into the mentioned seven classes, and are also in accord with findings about the early visual mechanism described in chapter 3. There, it has been shown that there are special cells in the brain that are sensitive to edges. These cells are divided into groups, each group being sensitive only to edges whose orientations are within a certain range of angles. The presence of these special cells implies that edges are very important perceptually, and justifies the notion of edge-oriented classification, used herein.

Any other content classifier may be used for the classification process as long as it can divide the image blocks into classes which are perceptually distinct. For example, a simple classifier, operating in the cosine domain, has been proposed in [Kim 1989] and [Kim 1991]. It uses information about several DCT coefficients (like their polarities and magnitudes) in order to perform the desired classification. However, this classifier was used to classify image blocks of 4x4 pixels and the development of a classifier for 8x8 blocks, which is the size of the blocks in our work, seems not to be straightforward. In addition, since we will compare coding results of the DCGSVQ system with CVQ systems, operating in the spatial domain, it seems reasonable to employ the same classifier in both systems. Nevertheless, as will be shown later on, we have adopted an idea from the mentioned references in order to subdivide further the mixed class due to the large number of image blocks that were classified as mixed blocks.
Finally, it should be noticed that certain thresholds have to be set for the decision tree in the classification algorithm. Those were found through visual tests employing various image blocks which were tested by the classifier and then judged by several viewers.

4.2.3 Incorporating HVS Properties

A number of models have been proposed to describe the HVS based on threshold measurements. In general, they are based on a linear filtering of the stimulus over the dimensions involved followed by a detection mechanism. Mannos and Sakrison [Mannos 1974] have argued that after an initial nonlinear transformation, the remainder of the visual system may be considered linear over a moderate range of intensities. Thus, they have proposed the following circularly symmetric MTF for the HVS

$$H(\omega) \;=\; 2.6\,[0.0192 + 0.114\omega]\exp[-(0.114\omega)^{1.1}] \qquad (4.3)$$

having a peak of value 1.0 at the radial frequency $\omega = 8$ cycles per degree (cpd). A simpler modulation transfer function has been proposed by Nill [Nill 1985], based on the work of DePalma and Lowry [DePalma 1962] and on his own work. The new function is defined as follows :

$$H(\omega) \;=\; [0.2 + 0.45\omega]\exp(-0.18\omega) \qquad (4.4)$$

having a peak at $\omega = 5.2$ cpd.

Actually, it should be noted that the human visual system is not isotropic as assumed above (see chapter 3). Nevertheless, the linear operation, described by the above MTF, was taken to be isotropic in order to make things simpler, i.e.,

$$H(f_x, f_y) \;=\; H(\omega)\;, \qquad \text{for} \;\; \omega \;=\; \sqrt{f_x^2 + f_y^2} \qquad (4.5)$$

where f_x and f_y are the spatial frequency coordinates which span the two-dimensional Fourier domain.

Incorporating a MTF in a coding system means to weight each transform coefficient by an appropriate value of this function. Thus, more importance can be given during the coding process to those coefficients that are more important to the human observer. Unfortunately, the functions described above are defined in the Fourier domain whereas it is sought to employ such functions in the cosine domain. In order to find the equivalent functions in the cosine domain, Nill has pointed out that an even extension of the original scene has to be created [Nill 1985]. This necessary alteration takes the form of forcing a symmetry onto a normally asymmetrical original scene. Forcing a scene to be symmetrical allows application of the cosine transform in place of the Fourier transform with no loss of information. That is, the scene can be exactly reconstructed from just the cosine transform. However, such scene alternation causes the loss of physical meaning since the human observer is not viewing this altered scene. To overcome this problem, Nill proposed the introduction of a function $\mid A(f_x) \mid$ which takes the form of

$$\mid A(f_x) \mid = \left\{ \frac{1}{4} + \frac{1}{\pi^2}\left[\log_e\left(\frac{2\pi}{\alpha} f_x + \sqrt{\frac{4\pi^2}{\alpha^2} f_x^2 + 1}\right)\right]^2 \right\}^{1/2} \qquad (4.6)$$

with $\alpha = 11.636$ deg^{-1} for the HVS model used herein.

He used the two-dimensional version of this function and proved that the following function

$$\hat{H}(\omega) = \mid A(\omega) \mid H(\omega) = \begin{cases} 0.05 \exp[\omega^{0.554}] \, , & \text{for } \omega < 7 \\ \exp[-9(\mid \log_{10} \omega - \log_{10} 9 \mid)^{2.3}] \, , & \text{for } \omega \geq 7 \end{cases} \tag{4.7}$$

can be used in image cosine transform applications in the same manner as $H(\omega)$ would be treated in image Fourier transform applications. In other words, $\hat{H}(\omega)$ may be used to weight the DCT coefficients to give

$$\hat{F}(f_u, f_v) = \hat{H}(f_u, f_v) F(f_u, f_v) \, , \tag{4.8}$$

where

$$\hat{H}(f_u, f_v) = \hat{H}(\omega) \quad \text{for } \omega = \sqrt{f_u^2 + f_v^2} \, , \tag{4.9}$$

and $\hat{F}(f_u, f_v)$ denotes the weighted DCT coefficients. The variables f_u and f_v are the spatial frequency coordinates which span the two-dimensional cosine domain. This process of weighting the DCT coefficients is denoted as *HVS Filtering* in Fig. 4.1. The functions $H(\omega)$ and $\hat{H}(\omega)$ are depicted in Fig. 4.2.

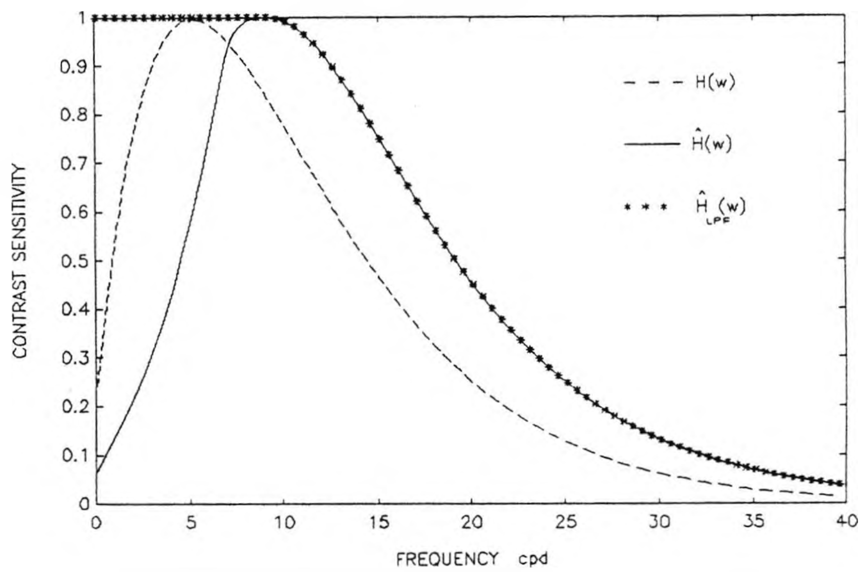


Figure 4.2: *MTF in Fourier and cosine domains*

It could be noticed in Fig. 4.2 that the effect of $\mid A(\omega) \mid$ is to translate $H(\omega)$ into the more positive frequency direction, resulting in a higher frequency peak (at $\omega = 9$ cpd). This implies that the higher spatial frequencies in the cosine domain play a more important role in the corresponding human-observed image quality than they do in the Fourier domain. However, this does not imply that the low-frequency coefficients are less important. It has been known that harsh quantisation of the low-frequency coefficients causes the *blocking effect* (block boundaries becoming visible). That is why the DC coefficient is not weighted by the MTF and special attention is given to the coefficients $F(0,1)$ and $F(1,0)$ [Carrioli 1988, Ngan 1989]. In fact in [McLaren 1991], all the low-frequency coefficients (below 5 cpd in Fourier domain) were treated with special attention.

In [Limb 1979], Limb sought to determine the specifications of the spatial filtering, masking, and error summing operations that take part in the HVS when a human observer is asked to assess the quality of an impaired image. Various types of distortions were added to a set of test images and a comparison was made between various distortion measures and ratings given by human subjects. The results have proved that a small amount of low-pass rather than band-pass filtering can improve the fit between the ratings and the distortion measures which were used. Limb argued that the noise added to the test images was generally above threshold and that, consequently, suprathreshold rather than threshold sensitivity functions might be operating. In other words, since the frequency response of the HVS is flatter for more visible stimuli, low-pass filtering (LPF) is more appropriate. Consequently, since image coding can be regarded as a process which introduces various types of noise (similar to those tested by Limb) into the processed image, the MTF used herein will be a LPF rather than a BPF. The LPF, referred to as $\hat{H}_{LPF}(\omega)$ in Fig. 4.2, is defined as follows

$$\hat{H}_{LPF}(\omega) = \begin{cases} 1.0, & \text{for } \omega < 9 \\ \hat{H}(\omega), & \text{for } \omega \geq 9 \ . \end{cases} \tag{4.10}$$

Another HVS property, which can be exploited to further improve the subjective quality of coded images, is the *spatial visual masking* phenomenon described in chapter 3. It is well-known that noise content in the direction of the luminance change, i.e., across the edge, are masked by the HVS provided that the noise has similar spatial frequency content to the picture signal itself. Thus, regions of high activity can be quantised more harshly than regions of low activity, provided that it is done only in the direction across the edge and not in the direction parallel to the edge direction. This notion will be further clarified in the next section.

4.2.4 Vector Configuration

After weighting each DCT coefficient by an appropriate value of the MTF, the AC coefficients are further processed by the next stage, the *vector configuration stage.* The AC coefficients are zonal sampled in order to discard some low-energy coefficients, and then grouped into two 16-dimensional *directional vectors* according to special *vector configuration maps.* It has been noticed that the distribution of energy over the transform coefficients is related to the content of the pixel block [Breeuwer 1988]. For example, for horizontal edges other coefficients should be quantised and transmitted than for vertical edges. Thus, the division of the AC coefficients into two vectors should be performed adaptively to the *direction* of the spatial activity in the pixel blocks.

Fig. 4.3 depicts the six vector configuration maps employed herein. They are based upon statistical data gathered from the distribution of energy over AC coefficients, derived from eight test images used in this research.
The DC coefficient, in Fig. 4.3, is referred to as 'd', where the nonzero table-values indicate to which vector a coefficient is assigned. The zeros in the table indicate which coefficients are discarded. Since the distributions of energy over the coefficients in the 45 degree and the 135 degree blocks are similar, their configuration maps are the same. The same argument holds for the shade and midrange classes.

It should be noted that unlike other researchers (see for example [Breeuwer 1988], [Ho 1989]), who have proposed grouping AC coefficients which have nearly equal variances into the same vector, we propose grouping the coefficients according to their *direction* in the cosine domain. Denoting the vertical and horizontal coordinates in the cosine domain as u and v respectively, the following notion of *directional vectors* is introduced. Coefficients in the u direction (generally those below the line $u = v$) are grouped into one vector while those in the v direction (generally those above the line $u = v$) are grouped into another vector. This definition results in a more meaningful vector configuration because it enables, for example, the coefficients, which represent the signal spectrum across an edge, to be in one vector (referred to as $\bar{v}_1$-type vector) while the coefficients, which represent the signal spectrum in the parallel direction to the edge, can be in another vector (referred to as $\bar{v}_2$-type vector).

d 1 1 1 1 1 1 0	d 1 1 1 1 1 1 0	d 1 1 1 1 1 1 1
2 1 1 1 1 1 0 0	2 1 1 1 1 1 0 0	2 2 1 1 1 1 1 0
2 2 1 1 1 1 0 0	2 2 1 1 1 1 0 0	2 2 2 1 1 1 0 0
2 2 2 2 1 0 0 0	2 2 2 2 1 0 0 0	2 2 2 2 1 0 0 0
2 2 2 2 0 0 0 0	2 2 2 2 0 0 0 0	2 2 2 2 0 0 0 0
2 2 2 0 0 0 0 0	2 2 2 0 0 0 0 0	2 2 2 0 0 0 0 0
2 2 0 0 0 0 0 0	2 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 0 0 0 0 0 0 0
Shade Class	Midrange Class	Vertical Edge Class

d 1 1 1 1 1 0 0	d 1 1 1 1 1 0 0	d 2 2 2 2 2 0 0
2 1 1 1 1 1 0 0	2 1 1 1 1 1 0 0	1 2 2 2 2 2 0 0
2 2 1 1 1 0 0 0	2 2 2 1 1 1 0 0	1 1 2 2 2 2 0 0
2 2 2 1 1 0 0 0	2 2 2 1 1 1 0 0	1 1 1 2 2 0 0 0
2 2 2 2 1 0 0 0	2 2 2 2 2 0 0 0	1 1 1 1 0 0 0 0
2 2 2 0 0 0 0 0	2 2 2 0 0 0 0 0	1 1 1 0 0 0 0 0
2 2 2 0 0 0 0 0	2 0 0 0 0 0 0 0	1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0
Mixed Class	45 degrees Class	Horizontal Edge Class

Figure 4.3: *Vector configuration maps*

As can be expected, the coefficients across the edge contain most of the signal energy (having large magnitudes even at high spatial frequencies) while those in the parallel direction contain less energy (having small magnitudes even at low spatial frequencies). So, the two directional vectors $\bar{v}_1$ and $\bar{v}_2$, so configured, are different despite being derived from the same image block. The magnitude (norm) of $\bar{v}_1$ is larger than the magnitude of $\bar{v}_2$, and the *shape* of $\bar{v}_1$ is more "noisy" than that of $\bar{v}_2$. Consequently, these vectors actually belong to different classes and therefore should be encoded by different shape codebooks. It should be noted, though, that the above observation is true only for vertical and horizontal edge-blocks. Diagonal edges manifest as meaningful coefficients in both directions, u and v, and along the line $u = v$. Therefore, the two directional vectors, derived from a diagonal edge-block, are similar and thus belong to the same class. The same is true for the mixed blocks where the two vectors belong to the same class which is, of course, different from the diagonal vectors' class.

The notion of *directional vectors* paves the way for incorporating various image processing techniques during the reconstruction of the image blocks. For example, feature enhancement can be carried out at the receiver by simply multiplying the decoded gain value by an *enhancement factor*. This process maintains the vector's shape while "stretching" its gain value, i.e., stretching the dynamic range of the coefficients which are included within that vector. Feature enhancement may be carried out only in the direction across a vertical or an horizontal edge (i.e., only for $\bar{v}_1$-type vectors). However, for the mixed vectors and the diagonal-edge vectors it may be done for both vectors, $\bar{v}_1$ and $\bar{v}_2$. Finally, the notion of *directional vectors* also ensures that the two largest coefficients, $F(0,1)$ and $F(1,0)$, are always assigned to different vectors, thus reducing the diversity of the shape vectors and reducing the dynamic range of the *gain* values, making the encoding procedure more efficient and resulting in better reconstructed images.

4.2.5 The Control Unit

The *control unit* controls a switch that defines which shape codebook should be used for encoding the directional vector undergoing quantisation. The unit performs its decision based on the following data:

a) The content classifier's decision about the class which the block under test belongs to, and

b) The polarity of the first vector element $v_1(1)$ or $v_2(1)$ if the block is a mixed block.

Employing the polarity of the first vector element for vector classification is in accord with the notion of block classification in the cosine domain as described in reference [Kim 1991]. There it has been shown that the polarity of the coefficients $F(1,0)$ and $F(0,1)$ play a major role in the classification process. Since the first element in a directional vector is either $F(1,0)$ or $F(0,1)$, it is checked by the control unit and used to sub-divide the mixed vector class due to the large number of those vectors.

The following vector classes are defined by the control unit:

1) A *shade* class consisting of all directional vectors derived from shade blocks.

2) A *midrange* class consisting of directional vectors derived from the midrange blocks, and of the $\bar{v}_2$-type vectors derived from blocks having an horizontal or a vertical edge running through them.

3) A *0-degree* class consisting of the $\bar{v}_1$-type vectors derived from blocks having an horizontal edge running through them.

4) A *90-degree* class consisting of the $\bar{v}_1$-type vectors derived from blocks having a vertical edge running through them.

5) A *diagonal edge* class consisting of directional vectors, derived from blocks having a diagonal edge running through them.

6) A *positive mixed* class consisting of directional vectors, derived from mixed blocks and having a positive first element.

7) A *negative mixed* class consisting of directional vectors, derived from mixed blocks and having a negative first element.

Classes *0-degree* and *90-degree* are actually very similar because they consist of directional vectors that contain coefficients in the direction across an edge. However, they are kept separated in order to enable the decoder to trace the content classification of the image blocks. In other words, the receiver uses the index of the vector $\bar{v}_1$ to identify the content class thus identifying which vector configuration maps should be used in order to reconstruct the block under test. A different shape codebook has to be designed for each of the above classes to accommodate the variety of shape vectors. The design algorithm of the shape codebooks will be described in the next section.

4.2.6 Encoding the AC Vectors

After constructing the directional vectors, they are normalised by their magnitudes (norm) in order to convert them into shape vectors. These shape vectors are then encoded by the *AC GSVQ* stage (see Fig. 4.1) employing appropriate shape codebooks that were specifically designed for each class of directional vectors. In this way it is ensured that proper shape encoding takes place. The *gain* value is defined as the correlation factor between the directional vector which is being encoded, and the best shape codevector, selected by the encoder from the appropriate shape codebook (see chapter 2). A scalar quantiser, which can be either a uniform or a Max-type quantiser, is used to encode the gain values. A special gain codebook may be designed for each class of vectors but this was found to be unnecessary in practice. Instead, one gain codebook was employed to encode all the gain values, thus making the process simpler. The indices of the shape codevector and the selected gain value are sent to the receiver where they can be decoded employing the same set of codebooks.

4.2.7 Encoding the DC Coefficient

Due to its importance for the reconstruction quality, the DC coefficient (representing the mean luminance of the block) has always been carefully quantised using a scalar quantiser. However, neighbouring DC coefficients are well correlated. In other words, the average luminance of one block is in general similar to those of its neighbours. Therefore a more efficient way of encoding the DC coefficient seems to be as follows. Four neighbouring DC coefficients can be grouped into one vector and encoded employing a VQ system. A similar idea has been proposed in [Tu 1987] which reports good results for a bit-rate of 4 bits per DC element. In [Ho 1989], the authors have proposed an equivalent notion employing interpolative VQ (IVQ) in order to exploit inter-block correlation and efficiently remove local mean luminance values using a small number of bits.

Based on these results we explored various methods for the encoding of the DC coefficients. Due to the vast diversity of DC vectors, a GSVQ is proposed instead of a VQ. By dealing separately with DC shape vectors and DC gain values, better coding results can be achieved. Therefore, as can be noticed in Fig. 4.1, the DC coefficient can be encoded employing either a scalar quantiser or a GSVQ. Both methods have been tested and results will be presented in a later section.

4.2.8 The Decoder

The various stages of the decoder perform basically the reverse operations of the encoder's stages. The DC Decoder in Fig. 4.1 uses the indices of the DC shape and the DC gain to retrieve the DC vector and DC gain value from the appropriate codebooks. If a scalar quantiser is employed for coding the DC coefficient, then only one index is transmitted to the receiver where a scalar decoder is used to decode the DC value.

The AC Decoder uses the indices of the AC vectors and the AC gain values to retrieve the AC vectors and the AC gain values from the appropriate codebooks. The index of the vector $\bar{v}_1$ is used to identify the block's class and accordingly the shape codebooks and the vector configuration maps that should be used in order to reconstruct the image block. The *Block Reconstruction Unit* recreates the cosine transform domain by arranging the decoded DCT coefficients in a proper way based on the vector configuration maps. Finally, inverse HVS filtering is carried out in order to compensate for the LPF, which took place at the encoder, and the inverse DCT is calculated to produce the decoded pixel block.

4.3 Codebooks Design

The proposed DCGSVQ system is actually a CVQ system with a special structure imposed on its codebooks, i.e., it is a *classified GSVQ*. Instead of using a different VQ for each of its classes, as generally is the case with CVQ, it employs a different GSVQ for each of them. A different shape codebook is employed for each class while employing either a different gain codebook for each class or using one common gain codebook. In this section, a new algorithm for designing the classified GSVQ codebooks is introduced. It will be used for designing the codebooks for the directional vectors as well as for designing the shape codebook, needed to encode the DC coefficients when they are encoded as a vector.

Designing codebooks for CVQ systems has been a difficult task because it has been mainly based on trial and error methods [Ramamurthi 1986]. Recently, however, a new algorithm aimed at solving this problem has been proposed in

[Kubrick 1990]. The *classified nearest neighbour clustering* (CNNC) algorithm is described in chapter 2 where it was used to design simultaneously the M codebooks, needed for the M different classes defined for a CVQ system.

In principle, the CNNC algorithm can be used to design the various shape codebooks needed for the proposed DCGSVQ system; however, it should be changed to deal with shape vectors instead of regular vectors. More specifically, the merging criterion, used for deciding which pair of clusters should be merged during each step of the algorithm, should be replaced by a new criterion which is more meaningful with respect to the shape vectors. When two shape vectors, each representing one cluster, are merged they are replaced by a *unified shape vector* which represents the *unified cluster*. This new representative shape vector should preserve, as closely as possible, the shape patterns defined by the original two shape vectors. Therefore, the new merging criterion should ensure shape preservation, i.e., the cross-correlation between the unified shape vector and each of the original shape vectors should be maximised.

A new merging criterion is developed in Appendix E and proved to ensure shape preservation when employed within the CNNC algorithm. Consequently, a modified version of the CNNC algorithm (employing the new merging criterion) can be used to design the shape codebooks, needed for the DCGSVQ. Naturally, this new algorithm can also be used for designing one shape codebook for a classical GSVQ, which is a special case of the classified GSVQ. In what follows, a detailed explanation is given in order to clarify the design algorithms for the various codebooks used herein.

4.3.1 The AC Codebooks

Three different algorithms for designing the product codebook of a GSVQ were tested in [Sabin 1984]. The *Individually Optimised Algorithm* was found to be the logical choice despite the fact that all three algorithms produced similar results. Basically, according to this algorithm, the shape codebook is designed first and then the gain codebook is designed using it. A similar approach is proposed herein using instead the modified CNNC algorithm for the design of the shape codebooks.

A training set of directional vectors, derived from a number of training images, is used to design the various AC shape codebooks. The vectors are grouped into various classes employing the same content classifier and control unit used by the encoder (see Fig. 4.1). The appropriate vector configuration maps are used too. The vectors are normalised by their magnitudes to obtain shape vectors and then used by the modified CNNC algorithm to produce the required AC shape codebooks.

As explained in chapter 2, a *weighted merging criterion* may be used in order to assign more importance to any particular class. By artificially increasing the merging error, calculated for a pair of clusters that belong to a preferred class and are on the merging list, the chance of merging these clusters may be reduced while increasing the chance of merging another pair of clusters, that belong to a less important class. Thus, the modified CNNC algorithm adaptively assigns the available bits so that a class of complex vectors (like for example, the *positive mixed* class) could be assigned more bits than a class of less complex vectors (like for example, the *shade* class).

In addition, the modified CNNC algorithm can also ensure that directional vectors, derived from a certain training image, can have more effect on the final classified GSVQ codebook than other training vectors. As explained above, the training vectors are first assigned to various classes where each one of them represents one cluster consisting of only one vector (itself). When two clusters S_i and S_j are being merged a unified centroid is calculated using the equation

$$\tilde{X}_{ij} = \frac{m_i \tilde{X}_i + m_j \tilde{X}_j}{m_i + m_j} \tag{4.11}$$

where m_i and m_j denote the number of vectors in clusters S_i and S_j respectively, and $\tilde{X}_i$ and $\tilde{X}_j$ are the centroids of these clusters (see Appendix B).
As can be noticed, one way of influencing the outcome of this process is by changing artificially the value of either m_i or m_j in that equation. Thus, by claiming that certain training vectors actually represent several vectors instead of just themselves, these vectors can become more influential than other training vectors. In other words, during the initial phase of assigning the vectors to the different classes instead of setting the variable m_i to a value of 1 it could be set to any other value provided that the vector was derived from a preferred image. This notion along with the idea of a weighted merging criterion were tested in

this research, and results will be presented in the next section.

The design algorithm for the AC gain codebook is explained next. Each training vector is encoded employing the classified shape codebooks, designed earlier. The encoding procedure ensures that for each vector $\bar{x}$ a proper shape vector $\bar{a}_i$ is found so that $\sigma = \bar{x}^T \bar{a}_i$ is maximised. The *gain* value, in our context, is exactly that σ (see chapter 2). So, a training set of σ values can be created in this way and used for the design of a codebook employing the LBG algorithm [Linde 1980]. The problem with this method is that the gain codebook may be quite good for images that are part of the training set, but not necessarily appropriate for images out of the training set. Therefore, a more universal method is proposed herein. Many natural images can be encoded using the classified shape codebooks, previously designed, and statistics about their appropriate gain values can be gathered and used to design either a uniform or a nonuniform scalar quantiser. By knowing the dynamic range of the possible gain values and designing accordingly the scalar quantiser, better coding results can be achieved, as will be shown in the next section.

4.3.2 The DC Codebooks

Two methods for encoding the DC coefficients are proposed. The first method is the traditional one where each coefficient is encoded using a scalar quantiser. The scalar quantiser can be a uniform quantiser, properly designed to accommodate the whole dynamic range of the DC coefficients, or a nonuniform quantiser, designed with a set of training DC coefficients. The second method is more complicated, employing a GSVQ to encode a DC vector, consisting of four neighbouring DC coefficients. A 4-dimensional shape codebook can be designed by the modified CNNC algorithm (where only one class is defined) and then a gain codebook may be designed in a similar way as it is done for the AC gain codebook. Both methods of encoding the DC coefficients were tested and appropriate results will be presented in the next section.

4.4 Simulation Results

We applied the DCGSVQ system to encode monochrome images of size 256×256 pixels with 256 grey levels. Coding results with DCGSVQ at various bit-rates employing codebooks designed with the modified CNNC algorithm are presented and compared with coding results obtained by other coding techniques like the VQ and the CVQ. The training set of directional vectors, used for the codebook design, was obtained from two 256×256 images, 'Lena' and 'Baboon' (Fig. 2.6), which represent images of different characteristics and statistics. The signal to noise ratio (SNR) in dB was used as a rough indicator of the quality of reconstructed images. In addition, since it is well known that the SNR does not correlate well with subjective (human) quality assessments, many subjective tests have been carried out too.

The viewing conditions, especially the distance between the screen and the viewer, are very important in the assessment procedure. For example, in [Watanabe 1968] it has been shown that the MTF shifts to the higher frequency region and that the cut-off spatial frequency rises monotonically with increase of the observation distance. Since a certain MTF is being used within the DCGSVQ system to weight the DCT coefficients, the observation distance should be carefully defined. The observation distance also affects the *conversion factor* between the coordinates of a DCT coefficient (u, v) and the equivalent spatial frequencies (f_u, f_v) in cycles per degree (see [Carrioli 1988] and [McLaren 1991]). Therefore, various subjective tests were carried out employing images that were encoded by a DCGSVQ, employing different values of the mentioned conversion factor. An observation distance of five times the image height (when displayed on a monitor of a Sun Sparcstation) was found satisfactory and was used throughout the subjective tests in this research.

Initial coding results with DCGSVQ codebooks at different rates, 0.48-0.625 bpp, have shown that the directional vectors, derived from the 'Baboon' image, have had more influence on the resulting codebooks than those derived from 'Lena'. The reason for this phenomenon lies in the characteristics of most of the 'Baboon' vectors. The vectors, derived from 'Baboon', are more diverse ("noisy") than those derived from 'Lena'. Therefore, it is more likely that the latter vectors will get merged during the codebook design procedure rather than the former vectors. In order to overcome this problem and balance the influence of the training images on the resulting codebooks, the initial value of the variable m_i was arbitrarily set to 5 instead of 1 (see discussion in the previous section) for each training vector that was derived from 'Lena'.

The gain values of the directional vectors were encoded by a common gain codebook. Due to the masking effect that takes place in the HVS, high activity regions in the image may be quantised more harshly than low activity regions. Therefore, a nonuniform scalar quantiser has been designed for encoding the gain values. It performs fine quantisation for small gain values (representing low-activity vectors) while performing coarse quantisation of large gain values (representing high-activity vectors). A 4-bit nonuniform scalar quantiser has proved satisfactory and was used throughout this research (unless otherwise stated) for encoding the AC gain values. The DC coefficient was encoded by a uniform 6-bit scalar quantiser, designed to accommodate the appropriate dynamic range.

4.4.1 A Comparison with Other Coding Techniques

SNR results for coding 'Lena' and 'Baboon' at various bit-rates are presented in Table 4.1 where they are compared with coding results, produced by a full search VQ and a CVQ. The VQ and the CVQ operate in the spatial domain employing codebooks designed by the LBG algorithm (with the *splitting* method) and the CNNC algorithm respectively. The training images for designing these codebooks were the same as those used for designing the DCGSVQ codebooks.

	R=0.625 bpp		R=0.562 bpp		R=0.5 bpp	
Encoder	Lena	Baboon	Lena	Baboon	Lena	Baboon
DCGSVQ	33.61	27.34	31.99	24.92	28.48	23.28
VQ	31.15	25.67	29.68	24.59	28.42	23.80
CVQ	29.70	25.28	27.77	23.41	26.34	22.25

Table 4.1: SNR results for the training images

Two reconstructed images of 'Lena' at 0.562 bpp are shown in Fig. 4.4. The image in Fig. 4.4*a* was produced by a CVQ system whereas the image in Fig. 4.4*b* was produced by the DCGSVQ. As can be noticed in Fig. 4.4, the image produced by the DCGSVQ is less blocky with a quite natural appearance whereas the image produced by the CVQ looks blocky; the edges are jagged and the background (which should be smooth) also appears quite blocky.

a *b*

Figure 4.4: *'Lena' coded by CVQ and DCGSVQ at 0.562 bpp*
a. CVQ *b.* DCGSVQ

Because the DCGSVQ encoder must ultimately operate on other input vectors that are not part of the training set, it is important to test its performance outside the training set, i.e., with images which were not used for codebook design. Three images from the USC database were used as test images: 'House' (USC15.IMG), 'Tree' (USC416.IMG), and 'Splash' (USC21.IMG). These images were coded by the DCGSVQ system employing the same codebooks that were used for coding 'Lena' and 'Baboon'. SNR results for these images at various bit-rates are presented in Table 4.2.

Two reconstructed images of 'House' at 0.5 bpp are shown in Fig. 4.5. The

Encoder	R=0.625 bpp			R=0.562 bpp		
	House	Tree	Splash	House	Tree	Splash
DCGSVQ	31.65	26.27	32.08	30.72	25.41	30.48
VQ	29.74	25.28	29.17	29.29	24.89	28.65
CVQ	28.50	23.82	25.19	27.71	22.97	24.57

Encoder	R=0.500 bpp		
	House	Tree	Splash
DCGSVQ	30.00	24.69	29.76
VQ	28.73	24.38	28.17
CVQ	27.05	22.33	24.12

Table 4.2: SNR results for the test images

image in Fig. 4.5*a* was produced by a CVQ system whereas the image in Fig. 4.5*b* was produced by the DCGSVQ system. Despite the fact that the SNR figures for a VQ system are higher than the appropriate results for a CVQ system (see Table 4.2), the subjective quality of the images, coded by the CVQ system, is better. That is why these images are compared with images produced by the DCGSVQ system. As can be noticed in Fig. 4.5, the image that was produced by the DCGSVQ system again looks less blocky and has a more natural appearance than the image that was produced by the CVQ system.

The coding performance of the DCGSVQ system was also compared with the performance of a transform coding classified VQ (TC-CVQ) system. The TC-CVQ operates in the cosine domain employing a CVQ to encode AC vectors which belong to various classes of activity. A content classifier is employed in the spatial domain to divide image blocks of 8×8 pixels into two classes: a *shade* class and a *general* class. The *shade* class consists of monotone image blocks, which contain no significant gradients, while the *general* class consists of all the other image blocks. A second division is performed by assigning the image blocks to one of four activity classes in a similar way as in [Chen 1977a].

Three 14-dimensional AC vectors are derived from each transformed image block consisting of AC coefficients which have nearly equal variances. Thus,

a *b*

Figure 4.5: *'House' coded by CVQ and DCGSVQ at 0.5 bpp*
a. CVQ *b.* DCGSVQ

taking into account that the lowest activity class consists of only *shade* image blocks while the highest activity class consists of only *general* image blocks, a total of eighteen different classes are defined for the TC-CVQ. A second system, referred to as TC-VQ, was defined by employing a full search VQ instead of a CVQ for encoding the AC vectors. In other words, the eighteen codebooks were merged to produce one codebook, which was then used to encode the AC vectors.

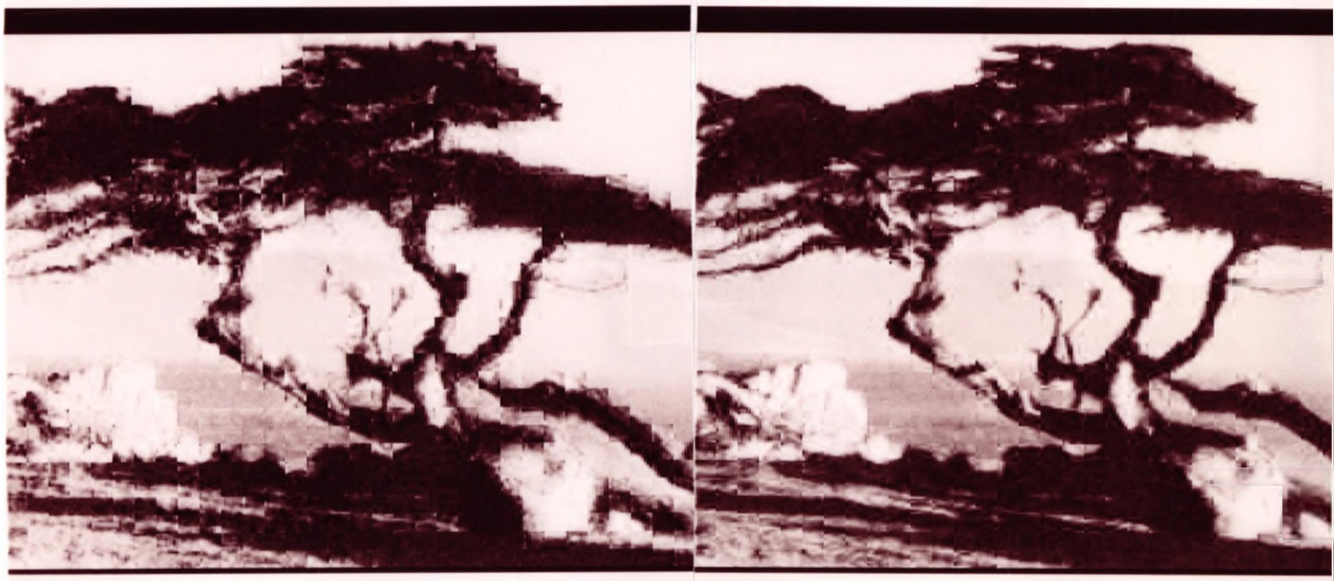
The CNNC algorithm was employed for designing the various codebooks, needed for the TC-CVQ, using the images 'Lena' and 'Baboon' as the training images. The DC coefficients were encoded by an 8-bit uniform scalar quantiser so that the equivalent bit-rate of the TC-CVQ system was 0.6875 bpp. SNR results for three test images, encoded by these systems, are presented in Table 4.3 and compared with coding results produced by the proposed DCGSVQ operating at a bit-rate of 0.625 bpp.

As can be noted from Table 4.3, the DCGSVQ system significantly outperforms

Encoder	Rate bpp	House	Tree	Splash
TC-CVQ	0.6875	27.37	22.60	28.27
TC-VQ	0.6875	28.12	23.13	28.79
DCGSVQ	0.6250	31.65	26.27	32.08

Table 4.3: SNR results for TC-CVQ and DCGSVQ systems

the other systems. Subjective tests of reconstructed images, produced by these systems, have confirmed the above findings. Two reconstructed images of 'Tree' are shown in Fig. 4.6 as an example of the systems' performances. The image in Fig. 4.6*a* was produced by the TC-CVQ system at 0.6875 bpp whereas the image in Fig. 4.6*b* was produced by the DCGSVQ system at 0.625 bpp. The images speak for themselves.



a *b*

Figure 4.6: *'Tree' coded by TC-CVQ and DCGSVQ*
a. TC-CVQ *b.* DCGSVQ

4.4.2 Vector Configuration Strategies

To test the proposed vector configuration strategy, the test images were coded by two systems. The first system was the proposed DCGSVQ operating with *directional vectors* that were configured according to the configuration maps, presented in Fig. 4.3. The second system was the proposed DCGSVQ operating with *variance vectors*, i.e., AC vectors that were configured by grouping DCT coefficients of similar variances into the same vector. Two 16-dimensional AC vectors were derived from each block in both systems. The DC coefficient was encoded by a 6-bit uniform scalar quantiser while encoding the AC gain values by a 5-bit uniform scalar quantiser. SNR results for coding the three test images by those systems at a bit-rate of 0.625 bpp are presented in Table 4.4.

Encoder	House	Tree	Splash
DCGSVQ - Directional Vectors	31.33	25.83	30.96
DCGSVQ - Variance Vectors	30.12	24.92	29.67

Table 4.4: SNR results for two vector configuration strategies

It could be noted from Table 4.4 that an improvement of more than 0.9 dB has been achieved by just grouping the DCT coefficients according to the new proposed vector configuration strategy, i.e., using directional vectors. The reason for this is mainly due to the fact that the two largest coefficients, $F(0,1)$ and $F(1,0)$, are assigned to different vectors according to the proposed directional vector notion. By having them apart, the shape vectors become less "noisy" and the effective dynamic range of the expected gain values is reduced. Consequently, the coding procedure becomes more effective and better SNR results are achievable. Subjective tests of reconstructed images, produced by both systems, were also carried out. The images, produced by the proposed system, look less blocky and more natural than those produced by the DCGSVQ system with the variance configuration maps.

4.4.3 Encoding the DC Coefficients

A 4-dimensional GSVQ is proposed as an alternative method for encoding the DC coefficients. Four neighbouring DC coefficients were grouped into one DC vector and then encoded. The training set of DC vectors was derived from four training images (including 'Lena' and 'Baboon'). However, in order to increase the diversity of the training vectors, the way in which the DC coefficients were grouped into a vector has been changed. Instead of just ordering the four coefficients by scanning them from left to right and from top to bottom, they were ordered in four different ways. In that way, four different DC vectors were created from a given neighbourhood of four DC coefficients ensuring a more diversed training set of vectors. The DC shape codebook was designed by the modified CNNC algorithm (for one class only), and consisted of 2048 shape vectors. The DC gain values were encoded by an 8-bit uniform scalar quantiser, designed to accommodate the effective dynamic range of prospective DC gain values. Consequently, the equivalent bit-rate for the encoding of the DC coefficients was $R_{DC} = 4.75$ bits per DC coefficient instead of the 6 bits in the scalar coding scheme.

SNR results are presented in Table 4.5. There it can be noticed that similar results have been achieved despite the fact that the overall bit-rate of the DCGSVQ for vector quantising the DC coefficients was 0.48 bpp whereas its bit-rate for scalar quantising the DC coefficients was 0.5 bpp. Subjective tests of the encoded images, produced by both systems, have indicated that they were of equal quality. As an example of the coding performance, a reconstructed image of 'Splash' is shown in Fig. 4.7. It has been coded by the DCGSVQ at a bit-rate of 0.48 bpp employing a GSVQ for encoding the DC coefficients.

Encoder	Lena	Baboon	House	Tree	Splash
DC-Vector Quantiser	28.53	23.25	29.87	24.71	29.83
DC-Scalar Quantiser	28.48	23.28	30.00	24.69	29.76

Table 4.5: SNR results - Two DC-encoding methods.

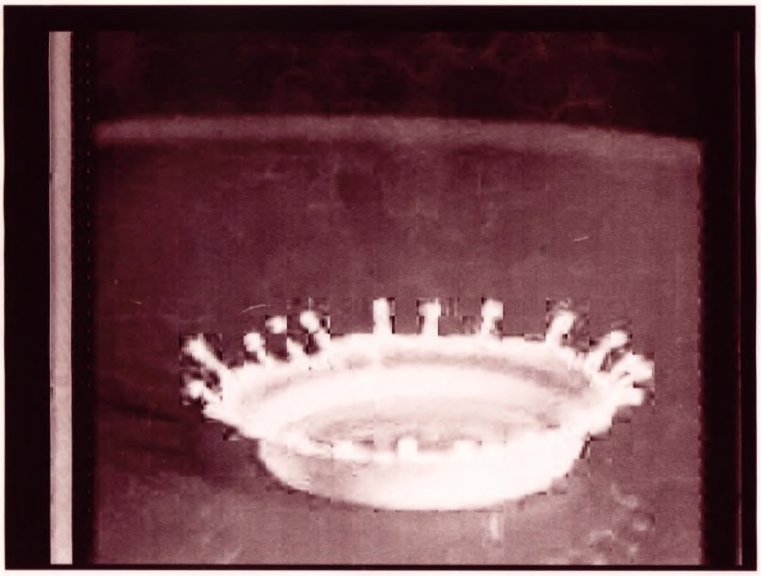


Figure 4.7: *'Splash' coded by DCGSVQ at 0.48 bpp*

4.4.4 Perceptually-based Codebook Design

The modified CNNC algorithm proposed herein gives the system designer full freedom to decide which training vectors and which class of vectors are more important so that they can be treated more wisely during the process of codebook design. As an example of this flexibility in design, we present in Table 4.6 coding results obtained by DCGSVQ systems that employed various codebooks at 0.593 bpp. The DCGSVQ employed a 6-bit uniform scalar quantiser for encoding the DC coefficients and a 5-bit uniform scalar quantiser for encoding the AC gain values.

The first codebook, referred to as *Original*, was designed without giving special importance to any training vector or any class of vectors. The second codebook, referred to as *Weighted I*, was designed with a training set of preferred vectors. Vectors derived from 'Lena' were given more importance than those derived from 'Baboon' (the initial value of the variable m_i was set to 5 for the preferred vectors). The third codebook, referred to as *Weighted II*, was designed with the same policy of preferred training vectors as the second codebook however, certain classes were given more importance. Instead of using the merging error as the criterion for merging clusters of vectors, a *weighted merging error* was employed.

Type	Lena	Baboon	House	Tree	Splash
Original	29.99	26.15	30.65	25.28	30.26
Weighted I	32.15	24.95	30.77	25.45	30.53
Weighted II	32.32	25.18	30.86	25.50	30.55

Table 4.6: SNR results for various types of codebooks at 0.593 bpp

The following policy of class weighting has been adopted. The two mixed classes, *positive mixed* and *negative mixed*, were assigned a weighting factor of 1. The vertical and horizontal edge classes were assigned a weighting factor of 0.75 while assigning a weighting factor of 0.5 to the diagonal edge class keeping in mind that the HVS is less sensitive to diagonal edges than to vertical or horizontal edges. The *shade* and the *midrange* classes were each assigned a weighting factor of 0.25. It should be emphasized that no attempt has been made to optimise these weighting factors and that the results presented herein are merely given as an example of the flexibility of the design algorithm. Subjective tests of the reconstructed images have confirmed that better coding performance could be achieved by using the proposed notion of codebook design.

4.4.5 Feature Enhancement

The inherent attribute of feature enhancement in DCGSVQ systems was explored. As explained earlier, the decoded gain value at the receiver can be multiplied by an *enhancement factor* in order to "stretch" the dynamic range of the decoded vector. When done correctly, i.e., applied only to mixed vectors or to v_1-type vectors in the case of image blocks that contain vertical or horizontal edges (see the previous section), "crisper" reconstructed images can be obtained. In Fig. 4.8, two reconstructed images of 'Baboon' at 0.5 bpp are shown. Both of them were produced by the same DCGSVQ with a 6-bit uniform scalar quantiser for encoding the DC coefficients and a 4-bit nonuniform scalar quantiser for encoding the AC gain values. However, the image in Fig. 4.8*b* was produced by multiplying the appropriate decoded gain values by an enhancement factor of 1.2. The effect of this process can easily be noticed; the edges are sharper and the face shows more detail. Nevertheless, it should be stated that the proposed process enhances not only the desired features in the reconstructed image but also the quantisation noise. Therefore, the enhancement process should be applied with appropriate care.

a *b*

Figure 4.8: *Feature enhancement - 'Baboon' at 0.5 bpp*
a. DCGSVQ *b.* DCGSVQ + Enhancement

4.4.6 HVS Filtering

As explained earlier in section 4.2, HVS filtering is being carried out in the cosine domain by weighting the DCT coefficients according to the MTF defined in equation (4.10). A LPF, rather than a BPF, was used in this work to describe the MTF of the HVS. To test this decision, we have compared coding results for a LPF with coding results for a BPF (Table 4.7). Three test images were coded by a DCGSVQ at 0.625 bpp employing a uniform 6-bit scalar quantiser for the DC coefficients, and a uniform 5-bit scalar quantiser for the AC gain values.

HVS Filtering	House	Tree	Splash
LPF	31.33	25.83	30.96
BPF	31.28	25.77	30.73

Table 4.7: SNR results for LPF and BPF at 0.625 bpp

As can be noticed from Table 4.7, marginally better coding results were achieved when a LPF was employed for weighting the AC coefficients. Subjective tests of reconstructed images have confirmed this finding. The reason for these findings is that the low-frequency coefficients are better preserved when a LPF, rather than a BPF, is being employed (see discussion in section 4.2).

In the proposed DCGSVQ system, inverse HVS filtering is carried out at the decoder to compensate for the HVS filtering which took place at the encoder. In general, HVS filtering reduces the influence of high-frequency "noisy" coefficients on the selection of a proper shape vector by the encoder. In other words, the low-frequency coefficients dominate the encoding process allowing harsh quantisation of the high-frequency coefficients. However, when inverse HVS filtering is carried out at the decoder, the quantisation errors which are present at high-frequency coefficients are amplified causing unnecessary degradation of the reconstructed images.

SNR results for reconstructed training and test images at 0.562 bpp are presented in Table 4.8. An improvement of more than 0.3 dB can be noted when inverse HVS filtering is being omitted during the decoding of the test images. On the other hand, a small degradation is caused when inverse HVS filtering is being omitted during the decoding of the training images.

Inverse HVS Filtering	Lena	Baboon	House	Tree	Splash
With	31.99	24.92	30.72	25.41	30.48
Without	31.81	24.82	31.02	25.73	30.86

Table 4.8: SNR results with and without inverse HVS filtering at 0.56 bpp

Subjective tests of reconstructed images, produced without inverse HVS filtering, have shown some loss of fine details; however, some of the high-frequency quantisation noise has been reduced too so that, in general, the perceived quality of the test images has been actually improved. Therefore, it is recommended that HVS filtering should be carried out at the encoder, to ensure better shape-preservation during encoding, but inverse HVS filtering could be omitted at the decoder when test images are concerned.

4.4.7 Nonlinear HVS Transform

There is sufficient direct physiological evidence to support the hypothesis that the relationship between the light intensity input to the visual receptors and the neural output level is approximately logarithmic (see chapter 3). Therefore, various researchers have suggested a point nonlinear transformation as the first stage in any digital image processing system. For example, Mannos and Sakrison have proposed the exponential function, $\mathcal{T}[I(x,y)] = I^{0.33}(x,y)$, as being best for image coding and transmission [Mannos 1974]. Stockham, on the other hand, has proposed a logarithmic transformation prior to carrying out any further image processing [Stockham 1972]. We have tested this notion by applying such a nonlinear transformation, referred to as *HVS transform*, to the intensity values prior to encoding the images with the DCGSVQ encoder.

SNR results are presented in Table 4.9 for DCGSVQ-coded images at 0.625 bpp. A uniform 8-bit scalar quantiser was employed for coding the DC coefficients, while using a nonuniform 4-bit scalar quantiser to encode the AC gain values.

HVS Transform	Lena	Baboon	House	Tree	Splash
Logarithmic	32.80	26.76	30.74	25.05	30.77
Exponential	33.98	27.03	31.15	25.57	31.12
Linear	34.41	27.19	31.47	26.13	31.12

Table 4.9: SNR results for various HVS transforms at 0.625 bpp

Based on these results and on subjective tests of the coded images, it is quite evident that better coding results are obtained without applying the HVS transform. These results are in accord with results presented in [Ramamurthi 1985], and could be explained in the following way.

A television display using a cathode-ray tube (CRT) transforms the electrical signal ν to a light display in a nonlinear way. This nonlinear characteristics of the CRT is described by its *gamma factor*, γ, and is defined by

$$L = c\nu^{\gamma} + L_0$$

where L is the displayed luminance, c and L_0 are positive constants, and $2 \leq \gamma \leq 3$ for most CRT's. Therefore, in practical image transmission systems, the signal of TV cameras output is γ precorrected to correct the γ characteristics of the CRT display. That is, the electrical voltage $\hat{\nu}$, produced by the image sensor, is nonlinearly transformed according to a γ precorrection curve defined by

$$\nu = \hat{\nu}^{1/\gamma} .$$

Thus, if one wishes to apply the mentioned HVS transform prior to encoding an image, the γ precorrection effect should be cancelled first (by inverse γ precorrection transformation). The results in Table 4.9 reflect the fact that such inverse transformation has not been applied. It should be noted, though, that if γ equals 3.0, the γ precorrection curve is identical to the exponential HVS transform proposed in [Mannos 1974] ($\gamma = 2.8$ in PAL systems). Thus it could be claimed that the electrical signal applied to the encoder is already in the *perceptual domain*, i.e., the domain where signal processing is carried out by the human eye and brain, and therefore should not undergo the proposed HVS transform.

4.5 Conclusions

A new cosine transform based scheme for image coding, *directional classified gain-shape vector quantisation* (DCGSVQ), has been introduced in this chapter. It combines vector quantisation with transform coding techniques and exploits various properties of the HVS like frequency sensitivity, the masking effect, and orientation sensitivity to produce reconstructed images with good subjective quality at low bit-rates. A new algorithm for designing the various codebooks, needed for the DCGSVQ, has been proposed. It is based on the CNNC algorithm but employs a new merging criterion which is more suitable for shape codebook design. The modified CNNC algorithm designs the various codebooks simultaneously giving the designer full freedom to assign more importance to certain classes of vectors or to certain training vectors. Coding performances of the DCGSVQ were compared with coding performances of full search VQ, CVQ, and TC-CVQ. The DCGSVQ outperformed the other systems producing nicer coded images at low bit rates.

The new design algorithm was also used to design a classical GSVQ, employed for encoding the DC vectors, and proved to be an effective alternative to traditional design algorithms. The new notion of *directional vectors* combined with the basic approach of using a classified GSVQ to encode them has paved the way for an effective coding procedure and for a simple technique of feature enhancement, which can be applied during the decoding process to improve the reconstructed images.

Close inspections of DCGSVQ-coded images at rates lower than 0.5 bpp have revealed that the blocking effect and the staircase effect are still noticeable (although less visible in comparison with VQ or CVQ-coded images). It seems that further processing of the coded images at the decoder is needed to improve their quality. A post-processing method that takes care of such processing at the decoder is proposed in the next chapter and shown to improve dramatically the perceived quality of reconstructed images.

Chapter 5

Postprocessing of Block Coded Images

Clouds gone,
The mountain shows.

5.1 Introduction

Vector quantisation provides many attractive features for image coding with high compression ratios. However, studies of image coding with vector quantisation have revealed several key drawbacks. Three basic drawbacks have been described in chapter 2 : (*i*) the computational complexity, (*ii*) the staircase effect (staircase noise), and (*iii*) the blocking effect (grid noise). The latter two drawbacks can be appreciated by examining Fig. 5.1 where a magnified reconstructed image of 'Lena', coded at 0.5 bpp using a classified vector quantiser (CVQ), is shown. The staircase noise is visible along diagonal edges such as the boundary line of the girl's shoulder and the black arch in the background. The grid noise is visible especially in monotone areas such as the girl's shoulder, her face and several areas in the background.

The two-dimensional Fourier amplitude spectra of monotone subimages derived from a monotone area in the original image of 'Lena' and the CVQ-coded image (Fig. 5.1) are shown in Fig. 5.2 and Fig. 5.3, respectively. Each of these two subimages is composed of four blocks of 4×4 pixels, so that the effects at

Figure 5.1: *'Lena' coded by CVQ at 0.5 bpp.*

the block boundaries are included. It should be noted that the means of the subimages have been subtracted in order to reduce the dynamic range of the spectrum.

In Fig. 5.2, it can be noticed that a large fraction of the signal's energy is packed in relatively few low frequency Fourier transform coefficients. In other words, the signal spectrum in monotone areas has a low bandwidth and will be assumed to be nearly isotropic. In Fig. 5.3, it can be noticed that most of the grid noise is out-of-band i.e., it is spread among high frequency Fourier coefficients. This observation is of great importance paving the way toward simple ideas for grid noise suppression.

In a similar way, Fig. 5.4 and Fig. 5.5 show the two-dimensional Fourier spectra of an *edge subimage* (a subimage containing an edge in it) derived from the original image of 'Lena' and the CVQ-coded image, respectively. The four lowest frequency coefficients in the spectrum have been set to zero in order to reduce the dynamic range, thus emphasizing the distribution of the signal's energy among high-frequency coefficients. The edge runs from the top right corner of the subimage to the bottom left. It can be noticed that the signal spectrum has high bandwidth in the direction which is perpendicular to the

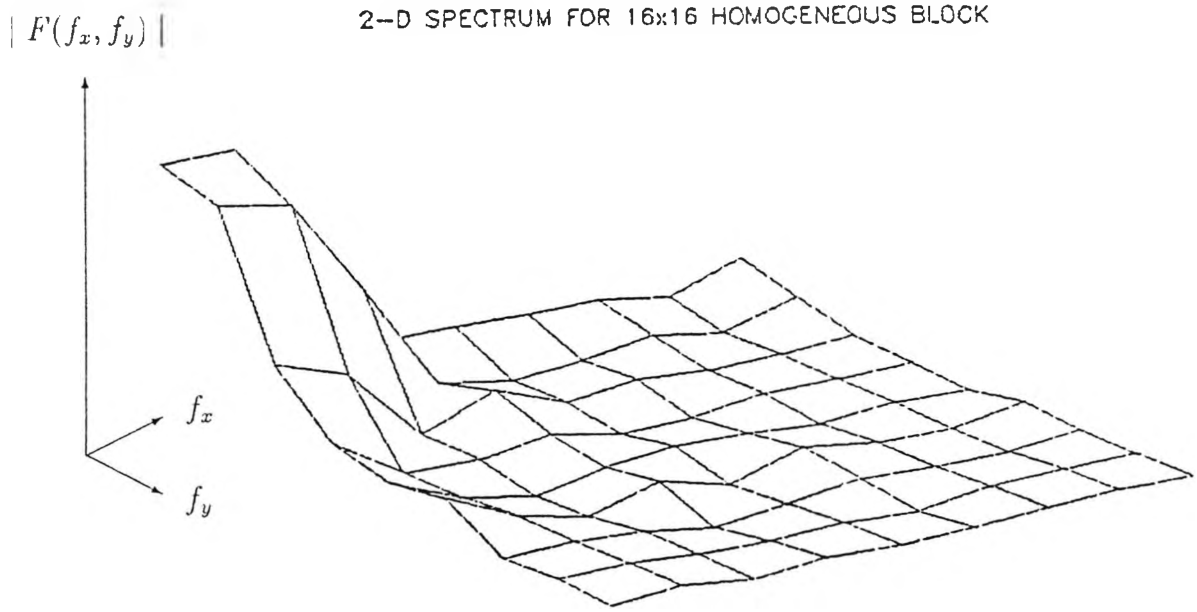


Figure 5.2: *Two-dimensional spectrum for a 16 × 16 monotone subimage derived from the original image of 'Lena'.*

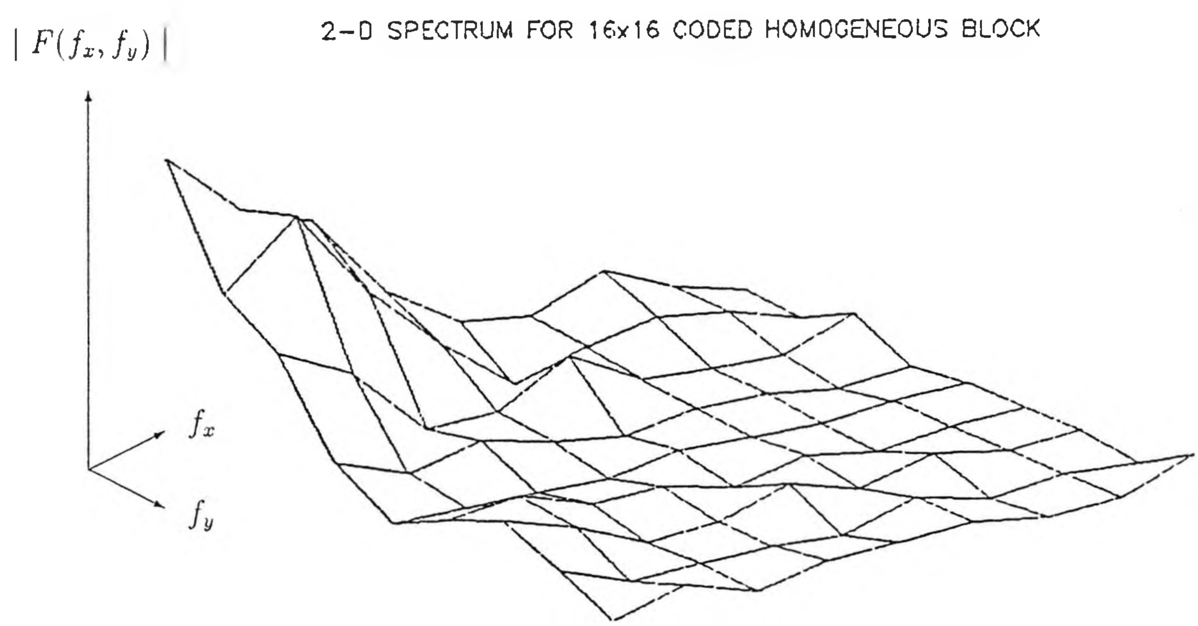


Figure 5.3: *Two-dimensional spectrum for a 16 × 16 monotone subimage derived from the coded image of 'Lena' at 0.5 bpp.*

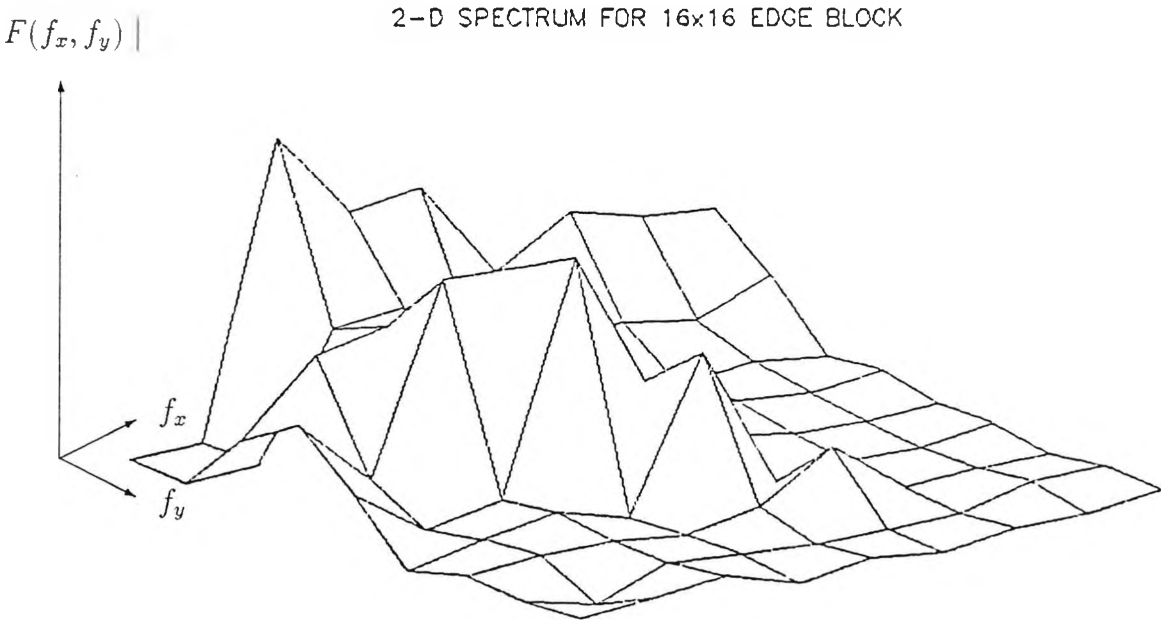


Figure 5.4: *Two-dimensional spectrum for a 16 × 16 edge subimage derived from the original image of 'Lena'.*

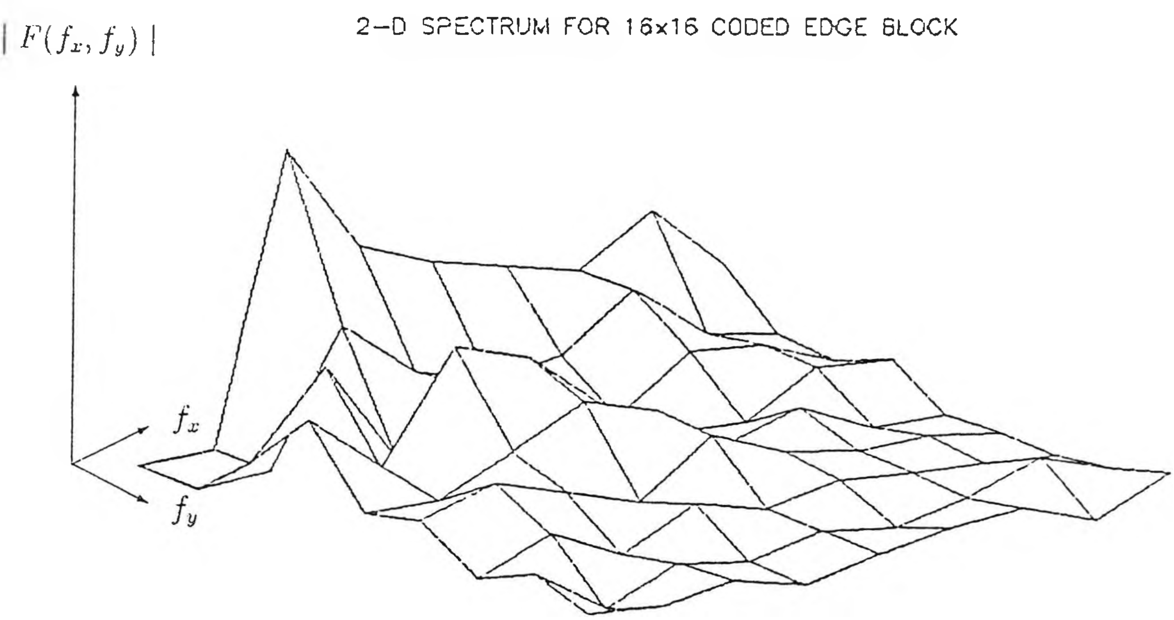


Figure 5.5: *Two-dimensional spectrum for a 16 × 16 edge subimage derived from the coded image of 'Lena' at 0.5 bpp.*

edge direction. The staircase noise, on the other hand, has significant energy out-of-band in the direction parallel to the edge. In addition, it has a large fraction of energy well within the signal's bandwidth (perpendicular to the edge direction). This knowledge about the spectrum of the signal and the staircase noise can be utilized in developing effective noise suppression techniques.

A variety of methods for noise suppression in block coded images have been suggested in the literature. Several methods for noise suppression (filtering), and several coding systems that produce less visible noise in coded images are described in this chapter. In addition, a new algorithm for postprocessing of block coded images is developed and tested employing various coded images.

5.2 Noise Suppression Techniques

The variety of methods proposed in the literature for reducing noise in coded images or *contaminated images* (images degraded by noise) fall into two categories: (*i*) source coding techniques which produce less visible noise or which are intrinsically free of blocking effects, and (*ii*) postprocessing of contaminated images. Several coding methods and coding considerations which fall into the first category are described next.

The required characteristics of an orthogonal transform function in order to obtain the best image quality, mainly in suppressing the blocking effect, have been clarified in [Miyahara 1985]. It has been demonstrated theoretically and experimentally that among most popular unitary transform functions (Hadamard, Slant, cosine and Legendre), the discrete Legendre transform is the best. New measures of degradation in block coded image quality have been suggested and used in assessing the performance of the mentioned transform functions. It should be noted though that the blocking effect is reduced by choosing the best transform function, but it still exists due to the segmentation process which is inherent in block coding.

A new measure of block distortion in coded images (employing the DCT) has been developed in [Carrioli 1988]. This new measure stresses the importance of properly coding the low frequencies, which are shown to be dominant in reducing blocking effects. It has been shown that for an 8x8 block the three lowest spatial frequencies should be coded with at least 6 bits per coefficient in order to reduce the blocking effect.

An *overlapping* method has been proposed in [Reeve 1983]. According to the overlapping method, instead of forcing the image blocks to be exclusive of each other, the image blocks overlap slightly around the block boundaries. Boundary pixels are then coded in two or more adjacent blocks so that redundant information is transmitted for these pixels. In reconstructing the image, the decoder averages the reconstructed pixels from neighbouring blocks, in the overlapping areas. Thus, abrupt boundary discontinuities caused by coding are reduced and the blocking effect becomes less visible. The disadvantage of this approach is the increase in the total number of pixels to be processed, and thus an increase in the bit rate.

A new and computationally simple approach, which abandons the idea of rectangular image blocks, has been suggested in [Pearson 1984]. Nonrectangular, overlapping blocks are used as receptive fields in the source image. These blocks, termed *interleaved blocks*, are converted into conventional rectangular, nonoverlapping blocks which are then coded using TC. It should be noted that when an image is divided into interleaved blocks, each pixel in the image is uniquely associated with one block only. However, the receptive field of any particular block intermingles with those of its neighbours thus ensuring tone continuity. At the decoder the inverse process takes place. Since the reconstructed blocks overlap, they automatically interpolate between themselves thus concealing the blocking effect.

A new class of transforms for block coding, introduced in [Cassereau 1985] and [Malvar 1989], has the same benefits of the overlapping method cited above, but without an increase in the bit rate. These new transforms, collectively referred to as the *Lapped Orthogonal Transform* (LOT), are characterized by the fact that the basis functions overlap adjacent blocks. However, the number of transform coefficients is kept equal to the original block size, so that no data overhead is incurred. A fast LOT procedure, introduced in [Malvar 1989], allows the implementation of block coding systems at low bit rates with much less noticeable blocking effects than traditional DCT-based systems.

In [Hinman 1984], the *Short-Space Fourier Transform* (SSFT) has been used instead of the discrete cosine transform (DCT). The SSFT of a block depends on the whole image thus, it is intrinsically free from blocking effects. Unfortunately, the SSFT introduces ringing around edges which is very objectionable.

Although different ways for reducing block distortion (grid noise) have been suggested, *edge integrity* [Aravind 1986] across adjacent blocks has not been ensured. Edge integrity is a notion that refers to the accurate reproduction of edges in orientation and location in coded images. In [Aravind 1986], a new *Finite-State Vector Quantiser* (FSVQ) has been introduced for image coding, addressing the problem of preserving edge integrity and tone continuity across adjacent blocks. A FSVQ is a vector quantiser with memory which exploits the two dimensional spatial correlation between adjacent image blocks. It uses a finite collection of memoryless vector quantisers, each with its own codebook. Each successive input block (vector) is quantised with a codebook determined by the *current state.* The current state is determined by the image blocks adjacent to the current block and by the previous channel index (binary word transmitted to the decoder). Employing a new distortion measure during codebook design and a perceptually based classifier for block classification and state definition, good perceptual quality of coded images at 0.375 bpp has been obtained. In addition, two new FSVQ schemes called *Side-Match Vector Quantisation* (SMVQ) and *Overlap-Match Vector Quantisation* (OMVQ) have been introduced in [Kim 1986]. By exploiting the two-dimensional correlation between adjacent image blocks, these schemes reduce substantially the blocking effect.

Although a proper coding technique can be chosen in order to ensure good quality of coded images, one has to know how properly to process coded images, already degraded by grid and staircase noise. Thus, postprocessing techniques which fall into the second category of methods, aimed at reducing noise in coded images, are described next.

A *filtering* method has been proposed in [Reeve 1983]. According to the filtering method, the coding procedure is not changed, i.e., image blocks remain mutually exclusive. Instead, an image filtering procedure is applied at the decoder. Since the boundaries of the image blocks are predetermined and known to the decoder, low-pass filtering of image pixels at or near these boundaries can smooth the unwanted discontinuities. Although this method does not increase the bit rate, it blurs the signal across block boundaries as can be seen in Fig. 5.6.

Figure 5.6: *A block coded image after postfiltering.*

A 3x3 Gaussian spatial domain filter, proposed in [Reeve 1983], was used to obtain the image in Fig. 5.6. The filtering procedure was carried out on the coded image, shown in Fig. 5.1. A different filtering method which avoids blurring by incorporating a prefilter prior to coding the image has been suggested in [Malvar 1987].

A number of edge preserving noise-cleaning nonlinear filters have been proposed in the literature [Chin 1983]. The median filter [Rosenfeld 1976], the K-nearest neighbour mean filter [Davis 1987], the gradient inverse weighted filter [Wang 1981], the sigma filter [Lee 1983] and the symmetric nearest neighbour filters [Harwood 1987] are some examples. These methods involve some type of local operation such as selective averaging, weighted averaging, etc. The focus of these approaches is on the reduction of noise while retaining edges and other details in the processed image. These nonlinear methods are not based on detailed knowledge of the image to be processed or the noise pattern and are therefore simple to implement.

Although being quite effective for some applications, such as suppressing additive white Gaussian noise or reducing speckle noise (caused by transmission errors), these methods are unsuitable for reducing grid or staircase noise. If applied to block coded images, these methods would interpret the staircase noise or the grid noise as being part of the signal and would leave them unsmoothed. Thus, speckle noise near a jagged edge would be smoothed, but not the jagged edge itself. As an example, a filtered block coded image is presented in Fig. 5.7. The filtering method used was a modified version of the symmetric nearest neighbour filter (SNN) applied to the coded image shown in Fig. 5.1. It can be noticed that the jagged edges have been left unchanged as postulated above.

Figure 5.7: *A filtered block coded image using a SNN filter.*

Space-variant nonlinear filtering techniques have also been used for noise suppression (see for example [Newman 1973], [Ingle 1979], [Sauer 1988], and [Ramamurthi 1986a]). In space-variant filtering, the filter response is varied to suit the local signal and noise characteristics. In [Newman 1973], an average gradient is calculated for each pixel and compared with a predetermined threshold constant. If the magnitude of the average gradient is greater than the threshold constant the pixel is defined as an *edge pixel* (i.e., it lies on an edge). The direction of the gradient is perpendicular to the edge direction thus, a narrow rectangle (mask) oriented perpendicularly to the gradient will in fact be aligned parallel to the edge on which the pixel lies. A new pixel value is then computed by averaging the pixel values inside the mask. Basically, this filtering procedure implements an edge-oriented one-dimensional filter aligned parallel to the edge. Unfortunately, if applied to block coded images, the filter would align itself parallel to the block boundaries leaving the grid and staircase noise unsmoothed.

In [Ingle 1979], a composite source model for image generation has been introduced. In this model, image blocks are classified into several classes each generated by a separate subsource. Five subsources have been defined: four correspond to edges with four orientations ($0\,\text{deg}, 45\,\text{deg}, 90\,\text{deg}, 135\,\text{deg}$), and one to monotone image blocks. The mean and covariance of each subsource is assumed to be known. A switch is employed to select at each time instant the output of one of the subsources as the overall image source output. Thus, at one instant, a monotone block may be selected while at the next, an edge block may be selected. Each pixel in the image is assumed to be produced by one of these five subsources. Based on noisy observations in a window surrounding the pixel under test (in the coded image) and a decision rule, the window is classified and the pixel is smoothed by a Kalman filter. The filter is specifically designed to suit the assumed statistics of the subsource which, according to the decision rule, has generated the pixel under test. Unfortunately, this algorithm is not suitable for reducing grid or staircase noise in block coded images. It is quite effective in suppressing random speckle noise, which is uncorrelated with the signal, but fails to reduce noise which is correlated to the signal. In order to effectively suppress grid and staircase noise, the correlation between these types of noise and the signal should be exploited as proposed in [Ramamurthi 1986a].

In [Ramamurthi 1986a], the notion of a composite source model for images has been adopted as in [Ingle 1979]. No assumptions have been made about the statistics of these subsources, hence it is not possible to apply Kalman filters. Instead, two spatial domain filters have been used: (*i*) a two-dimensional low pass filter has been applied to monotone image blocks, and (*ii*) a one dimensional low-pass filter has been applied parallel to the edge in edge blocks. A perceptually based classifier has been used for block classification. The classifier performs an analysis of a neighbourhood that is larger than the coded block and estimates the "true" orientation of the edge in the original image block from the noisy image. It is postulated that the monotone/edge and edge orientation decisions remain the same for all the pixels in one coded image block. Thus, these decisions are made once for a block of pixels and not for each pixel. Good results have been obtained by applying this algorithm to the coded image of Fig. 5.1, as can be seen in Fig. 5.8. However, it should be noticed that although the reproduced edges are less jagged they are still slightly blurred.

Figure 5.8: *A filtered block coded image using the algorithm of* [Ramamurthi 1986a].

To conclude this section, we refer to a recently proposed filtering algorithm aimed at reducing grid and staircase noise [Sauer 1988]. The first step in the algorithm is the location and identification of those edges in the coded image which are perceptually important. These edges will be those whose length is great enough, and curvature low enough so that patterns of distortion are easily observable. The edge location problem has been divided into (*i*) the local detection of an edge and (*ii*) tracing of the edge from points of detection. Local detection is performed along a set of equidistant vertical and horizontal lines (grid lines). Edge detection is based on the behaviour of the signal gradient on the grid lines (via one-dimensional differential filtering) followed by a thresholding process. Edge line tracing is performed assuming that the actual edge location is implicitly shifted by no more than one pixel per row of trace. The output of the edge tracing algorithm is a list of parameters describing detected and accepted edges. A one-dimensional low-pass filter is applied along the approximated edge, followed by a nonlinear contrast enhancement procedure in a direction perpendicular to the edge direction. The one-dimensional filter has been designed under the assumption that only DC along the edge need be passed. Therefore, a simple windowing (averaging) process has been used.

The proposed algorithm was applied to images coded by a FSVQ system or by the *Block List Transform* (BLT) [Haskell 1985]. It should be emphasized though, that the perceived quality of images, coded by these methods, is quite good (even without postprocessing) since spatial correlation between adjacent image blocks is being exploited. Therefore, the good results presented in [Haskell 1985] are not surprising. Moreover, in order to evaluate properly the performance of the proposed algorithm, images coded by memoryless vector quantisers or transform coders should be used. In addition, it should be noted that this algorithm performs individual operations on the entire decoded image sequentially, thus requiring storage capabilities for the entire image frame .

None of the postprocessing techniques described above have solved satisfactorily the complicated problem of grid and staircase noise removal along with edge preserving. We believe that by incorporating properties of the human visual system in the filtering algorithm, further improvement in the perceived quality of filtered coded images can be achieved. Such a new algorithm for grid and staircase noise suppression is introduced in the following section.

5.3 Description of the Algorithm

A new postprocessing algorithm aimed at reducing grid and staircase noise while retaining edge integrity and edge sharpness is described in this section. The proposed algorithm is based on observations made in section 5.1, concerning characteristics of the signal spectrum and the noise spectrum. In addition, various properties of the human visual system have been incorporated in the filtering algorithm in order to improve the perceived quality of filtered block coded images. In section 5.1, we have shown that most of the grid noise is out-of-band, while the staircase noise is out-of-band only in the direction parallel to the edge. The out-of-band portion of the grid and staircase noise can be reduced by filtering [Ramamurthi 1986a]. However, space-invariant filtering is inadequate for this purpose since it blurs edges, thus modifying the signal while smoothing the noise. Accordingly, we propose a space-variant filtering algorithm followed by an edge enhancement procedure.

In space-variant filtering, the filter response is varied to suit the local signal and noise characteristics. To perform space-variant filtering, some features of the local signal spectrum should be estimated from local observations of the signal plus noise. The local signal features, so estimated, are then used to select an appropriate filter for noise suppression. Nonlinear processing methods are very effective in estimating these features and are thus employed within the proposed algorithm. Consequently, the overall filtering algorithm is both space-variant and nonlinear.

The filtering algorithm consists of three steps: (*i*) grid noise removal, (*ii*) staircase noise removal, and (*iii*) edge enhancement. We describe next each of these steps and explain the rationale behind it. The first step in the proposed algorithm is based on the following two observations. First, we have shown in section 5.1 that for a monotone subimage a large fraction of the signal's energy is packed in relatively few low frequency coefficients. In other words, the signal spectrum in monotone areas has low bandwidth and is assumed to be nearly isotropic. Second, the locations of the image block boundaries are predetermined during the segmentation process, which takes place at the encoder. These locations are known to the decoder too and thus, it is reasonable to expect that by low-pass filtering the coded image at or near the subimage boundaries, unwanted grid noise would be smoothed. However, the filter, due to its low-pass nature, can degrade edge content in the image. Therefore, the filter should be applied only at pixels directly adjacent to subimage boundaries, and only if they are not *edge pixels*, i.e., pixels that lie on or near an edge. Following this rationale, we describe next the first step in the proposed filtering algorithm, aimed at reducing grid noise.

Each pixel in the coded image, $i(x, y)$, is first tested in order to determine whether it is a *boundary pixel*, i.e., a pixel directly adjacent to subimage boundaries, or not. If it is a boundary pixel, it is passed to a pixel classifier (see Fig. 5.9) in order to determine whether it is an edge pixel or not. If the pixel under test is found to be an edge pixel, no filtering is carried out. Otherwise, a two-dimensional low-pass filter is applied in order to reduce grid noise. Consequently, non-boundary pixels as well as boundary-edge pixels are left unsmoothed at this stage.

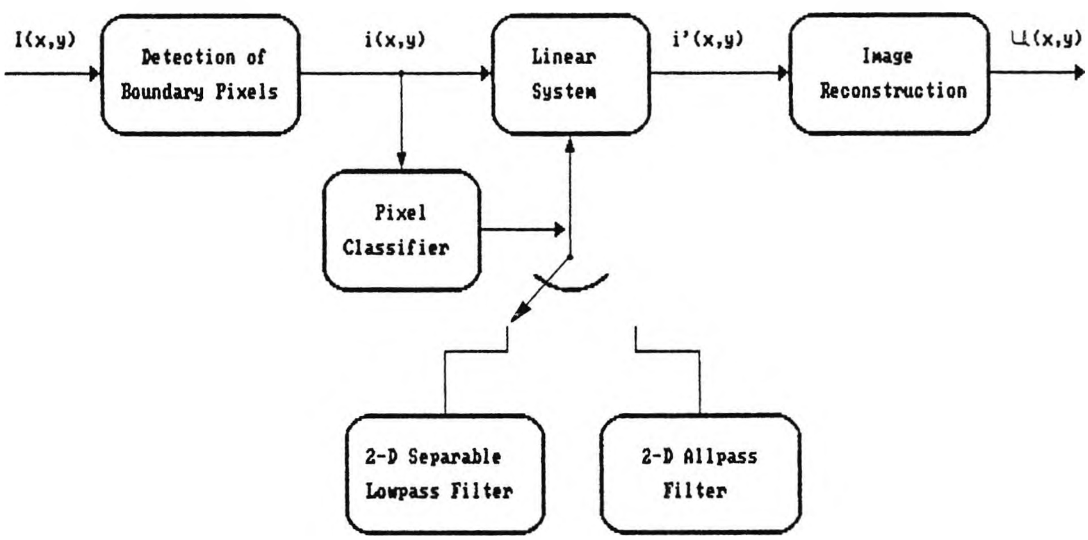


Figure 5.9: *First step - grid noise filtering.*

The pixel classifier is described in detail in appendix F. It is based on the analysis of a small neighbourhood, surrounding the pixel under test, and on known properties of the human visual system. The pixel classifier, in Fig. 5.9, controls a switch which is used to select one of two filters: (i) a two-dimensional low-pass filter, or *(ii)* a two-dimensional all-pass filter which does not affect the signal. The selected filter is then used to linearly process the pixel under consideration. At the conclusion of this step a new image, $U(x, y)$, is obtained. This image will be passed to the second stage of the proposed filtering algorithm. The filter's specifications should be chosen so that as much of the out-of-band noise as possible should be reduced without degrading the signal content. A new filter design method, which incorporates perceptual properties of the human visual system, is presented in appendix G. It has been employed for designing the various filters, used for postprocessing the coded images throughout our research. Practical filter parameters are described in section 5.4 along with the rationale behind their selection.

The second step in the proposed filtering algorithm is aimed at reducing staircase noise. It is based on the following observations. We have shown in section 5.1 that the signal spectrum has high bandwidth in the direction which is perpendicular to the edge direction. The staircase noise, on the other hand, has significant energy out-of-band in the direction parallel to the edge. Thus, by applying a one-dimensional low-pass filter aligned parallel to the edge the staircase noise can be reduced. Since there is no smoothing in the direction perpendicular to the edge direction, the edges are not blurred. The effect of the in-band portion of the staircase noise (in the direction perpendicular to the edge direction) on the perceived quality of the coded image is reduced by a phenomenon called *spatial visual masking*, described in chapter 3. That is, noise content in the direction of the luminance change are masked by the human visual system despite the fact that they are left unsmoothed by the applied filtering process.

In order to perform edge-oriented filtering, certain features of the signal should be estimated from noisy observations of the signal plus noise. Particularly, the direction of the edge should be estimated in order to select a proper filter. The correlation between the signal and the noise in the coded image is critical to the accurate estimation of edge orientation. The fact that the inclination of the staircase noise follows very closely the true orientation of the edge in the original image can be utilized in determining the edge direction in the coded image. Thus, by analysing a large enough neighbourhood in the coded image, it is possible to detect the existence of an edge along with its direction.

Following this rationale, we describe next the second step in the proposed filtering algorithm, aimed at reducing staircase noise.

Each pixel in the coded image, $U(x, y)$, is first passed to a pixel classifier in order to determine whether it is an edge pixel or not. If the pixel under test is found to be a non-edge pixel, it is smoothed by a two-dimensional low-pass filter. Otherwise, the pixel under test along with a predetermined number of surrounding pixels are formed into a subimage block and passed to a *block classifier* (see Fig. 5.10). The block classifier controls a switch which is used to select one of five filters, specifically designed to suit the signal content of the block under consideration. The selected filter is then used to linearly process the pixel under test. At the conclusion of this step a new image, $V(x, y)$, is obtained. This image will be passed to the third stage of the proposed filtering algorithm.

The block classifier is described in detail in appendix H. It is based on the analysis of coded image blocks and on known properties of the human visual system. Five classes of image blocks have been defined: (i) a monotone class consisting of monotone image blocks containing no significant gradient, and (ii) four edge classes consisting of image blocks having a distinct edge running through them. Four edge orientations have been defined: horizontal, vertical, and two diagonals. This definition is in accord with findings about the statistics of edges found in real images and about the early visual mechanism, as described in chapter 4.

It should be noted that the block classifier in Fig. 5.10 controls a switch which is used to select one of five filters. Four filters are one-dimensional low-pass filters applied parallel to the detected edge. The fifth filter, on the other hand, is a two-dimensional low-pass filter selected only when the block classifier detects either a monotone image block or a complex structured image block. A complex structured block contains fine details or complex edges which can not be treated as simple edges. Since the coding quality is not high enough to preserve fine details in images, it is difficult to estimate the features of the signal spectrum correctly. As a result, smoothing the signal and the noise is perceptually preferable in such cases, since the noise is more objectionable than the smoothed signal [Ramamurthi 1986a].

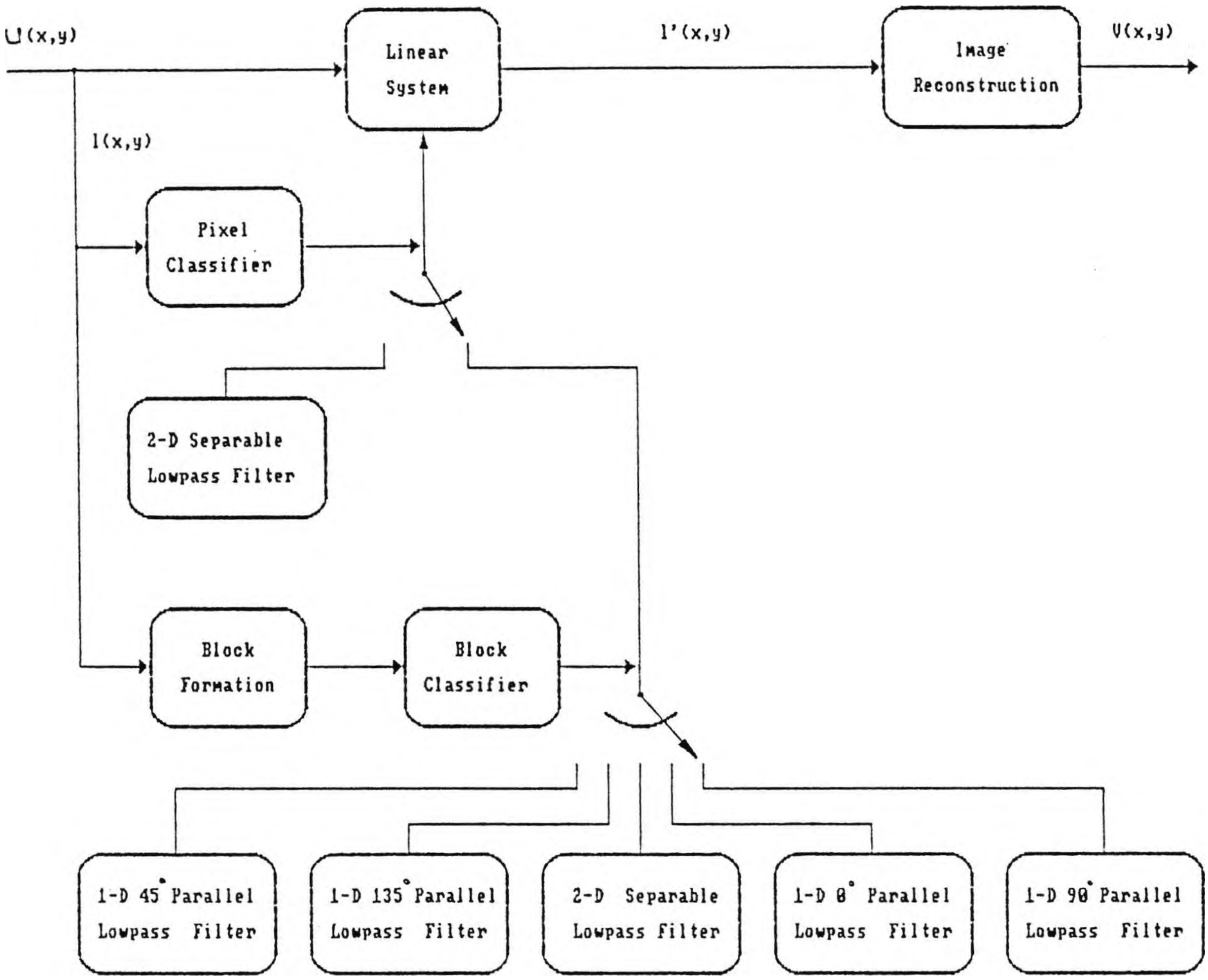


Figure 5.10: *Second step - staircase noise filtering.*

As explained earlier, an edge pixel along with a predetermined number of surrounding pixels are formed into a subimage block and passed to the block classifier for further processing. The size of the formed subimage block is a very important parameter in the proposed algorithm. If it is too small, the edge orientation decision will be highly localized and the applied filter will tend to follow the "steps" of the staircase noise. If it is too large, the simple estimate of edge orientation, made by the classifier, will be a poor one for curved edges. Thus, we have adopted the concept proposed in [Ramamurthi 1986a] and have defined a variable block size depending on the quality of the coded image.

Generally, edges are reproduced well in high quality coded images and do not need much smoothing. In that case, the size of subimage blocks can be small (about 5x5 pixels), thus ensuring close tracing of the edge orientation. For poorly coded images, edges are usually degraded by staircase noise appearing very jagged in the reconstructed image. In that case, a slightly larger block size (about 7x7 pixels) is appropriate for detecting the true orientation of the edge by exploiting the inclination of the staircase in the coded image block. Examples of coded images of different quality will be presented in section 5.4 along with filtering results for various image block sizes.

To conclude the description of the proposed filtering algorithm, we describe next its third step - the edge enhancement procedure. We have shown earlier that the out-of-band portion of the staircase noise, in the direction parallel to the edge, can be reduced by applying a one-dimensional low-pass filter aligned parallel to the detected edge. The in-band portion of the staircase noise, in the direction perpendicular to the edge direction, can be left unsmoothed due to the masking effect which takes place near edges. Despite the fact that there is no smoothing in the direction perpendicular to the edge direction, reproduced edges are still blurred. A close comparison between the spectrum in Fig. 5.4 and the spectrum in Fig. 5.5 reveals that high frequency elements in the coded subimage (in the direction perpendicular to the edge direction) have been attenuated during the coding process. In order to restore the edge gradient to its approximately original value, the mentioned high frequency elements must be restored. Such restoration can be obtained either by nonlinear contrast enhancement or by applying an high-pass filter perpendicularly to the detected edge.

The proposed restoration procedure should be carried out only for edge pixels. Therefore, each pixel, $V(x, y)$, is first passed to a pixel classifier in order to determine whether it is an edge pixel or not (see Fig. 5.11). If it is found to be a non-edge pixel, it is left unprocessed. Otherwise, an image block is formed consisting of the pixel under test and a predetermined number of surrounding pixels. A block classifier is then employed in order to determine the edge orientation. Based on the statistics on edges reported in [Keskes 1979], we apply high frequency restoration only to diagonal edges (at 45 and 135 deg) assuming that vertical and horizontal edges are usually well reproduced.

The high-pass filter employed for edge enhancement will be described in detail in section 5.4. In addition, an alternative edge enhancement procedure has

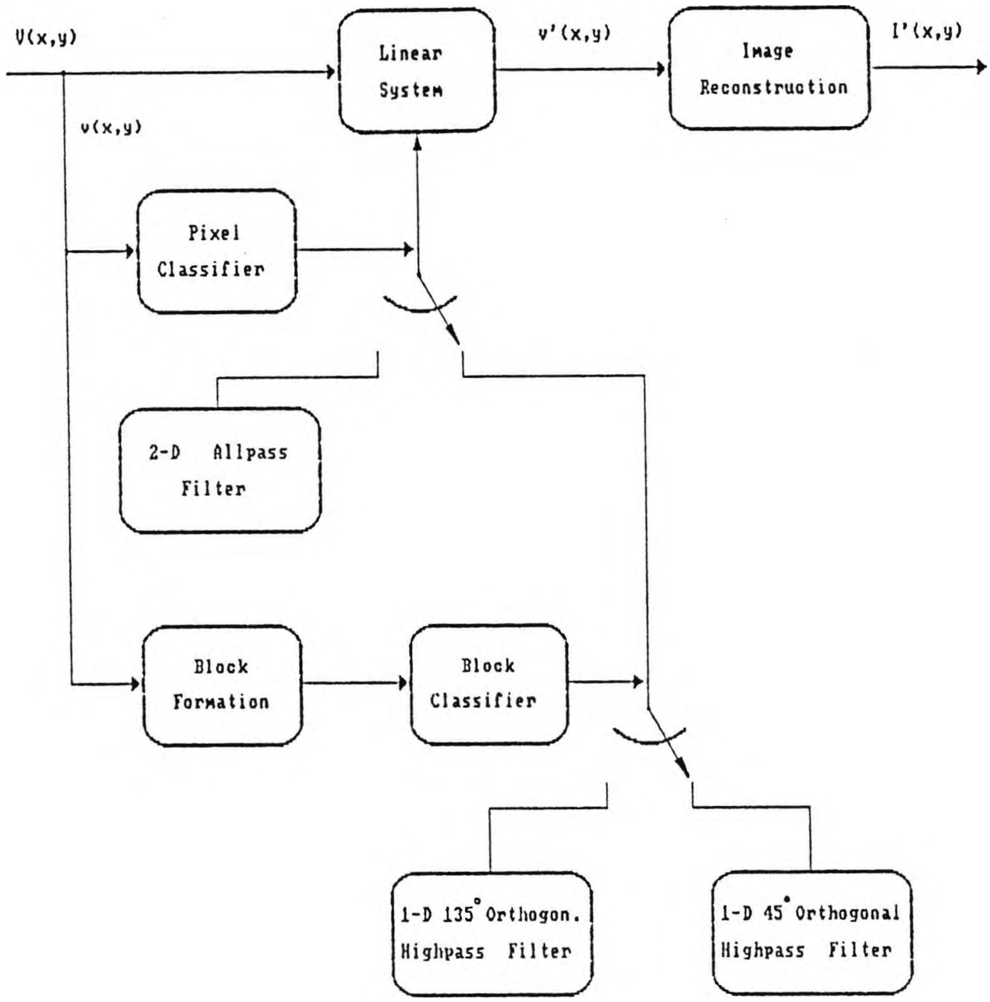


Figure 5.11: *Third step - edge enhancement.*

been suggested in [Sauer 1988] and tested by us. It is a nonlinear transformation defined as follows for a set of grey levels between v_{min} and v_{max} :

$$\mathcal{H}(v) = \begin{cases} v_{max} - \tau\delta^{\alpha}(v) & \text{for } v \geq (v_{max} + v_{min})/2 \\ v_{min} + \tau\delta^{\alpha}(v) & \text{for } v < (v_{max} + v_{min})/2 \end{cases}, \tag{5.1}$$

where

$$\tau = \frac{v_{max} - v_{min}}{2},$$

$$\delta(v) = 1 - \frac{1}{\tau} \mid v - \frac{v_{max} + v_{min}}{2} \mid ,$$

and α is a constant. The proposed enhancement procedure is adaptive to the local signal content. A predetermined number of pixels, in the direction perpendicular to the edge direction, define a *decision window* centered on the pixel under consideration. The grey levels of the pixels within the decision window are tested and the values of v_{max} and v_{min} are defined. The exponent α controls the degree of contrast enhancement as can be noticed in Fig. 5.12.

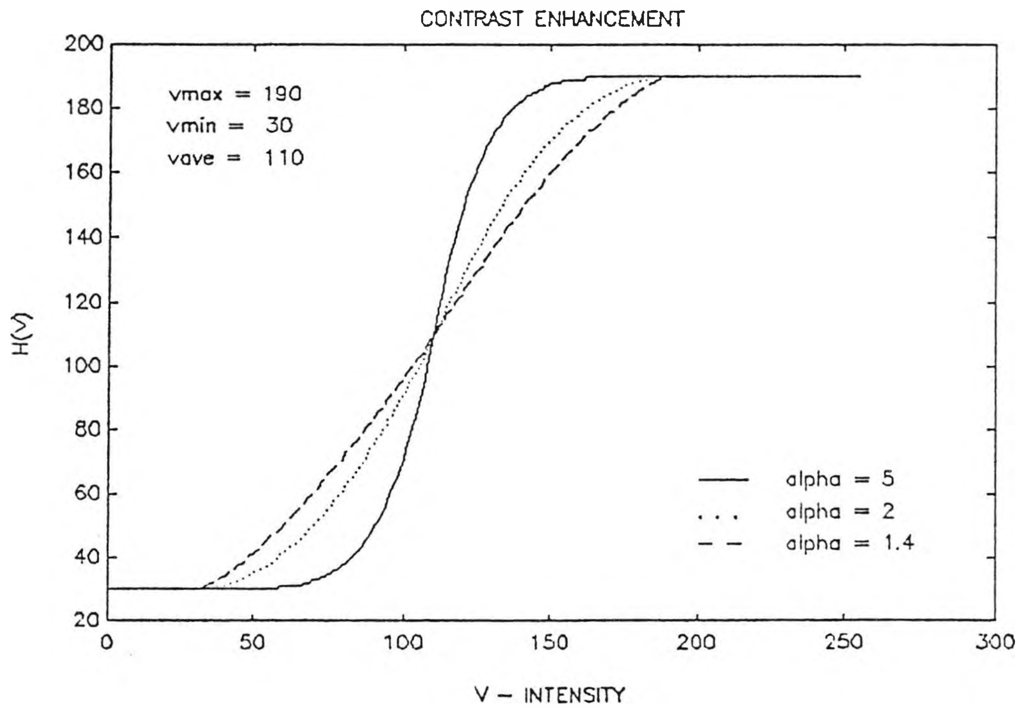


Figure 5.12: *Contrast enhancement.*

Both parameters, α and the size of the decision window, affect the performance of the proposed enhancement procedure and thus should be properly defined (as will be explained in section 5.4).

The basic concept behind our algorithm is its modularity and the adaptive way in which it can be applied. Each step stands alone and can be applied only if necessary. For example, all three steps should be applied for postprocessing of poorly coded images, degraded by grid and staircase noise, while applying just one step for high quality coded images. This concept differs substantially from other concepts reported in the literature ([Reeve 1983], [Ramamurthi 1986a], and [Sauer 1988]). Moreover, our algorithm treats each pixel in the coded image in a fully adaptive way. In [Reeve 1983] for example, all boundary pixels are smoothed by a two-dimensional low-pass filter without checking whether they are edge pixels or not. Such filtering strategy causes edge blurring as shown earlier in Fig. 5.6. Our algorithm, on the other hand, employs a pixel classifier which ensures that edge pixels are not smoothed, thus ensuring grid noise reduction without degrading edge content unnecessarily.

Another unique feature of our algorithm is the way in which decisions are made concerning the detection of an edge and its direction within an image block. In [Ramamurthi 1986a] for example, a block classifier is employed for testing predetermined image blocks of fixed size. The block classifier is applied to a large window of pixels (8x8 pixels) containing the image block which is under consideration. It is postulated that monotone/edge and edge orientation decisions, made once for a block of pixels, remain the same for all the pixels in that block. Our strategy differs from this strategy by the fact that each pixel is particularly tested without the constraint of assigning it to a certain predefined image block. In order to refine the edge detection process, we have applied the block classifier to subimage blocks which have been formed adaptively by gathering the pixel under test along with a predetermined number of surrounding pixels. Thus, for each edge pixel a more accurate decision can be made, based on noisy observations of the signal and noise found in a decision window centered on the pixel under test.

Finally, special care has been taken of high frequency elements in the coded image degraded by the coding process. We propose an edge enhancement procedure which is applied adaptively only to certain edges, i.e., edges in certain directions which are known to be more vulnerable to coding degradations. Particularly, we propose high-pass filters designed by a new technique which incorporates known properties of the human visual system. This new method ensures the good quality of filtered coded images, and has also been used for designing the low-pass filters, employed throughout steps 1 and 2 of the proposed filtering algorithm.

5.4 Experimental Results

In this section we present experimental results obtained by postfiltering block coded images which have been coded by various VQ systems. In addition, we describe the parameters of the various filters, employed within the proposed postfiltering algorithm, and the rationale behind their selection. Three 256x256 coded images have been used as test images. The various VQ systems, employed for coding these images, processed image blocks of 4x4 pixels (16-dimensional vectors). The first image, 'Lena1', is depicted in Fig. 5.1. It has been coded at 0.5 bpp using a CVQ system. The staircase noise is visible along diagonal edges such as the boundary line of the girl's shoulder and the black arch in the background. The grid noise is visible especially in monotone areas such as the girl's shoulder, her face and several areas in the background.

The second image, 'House1', is depicted in Fig. 5.13 (magnified by two). It has been coded at 0.56 bpp using a CVQ system. The perceived quality of this image is similar to the first image. Both types of noise, grid and staircase noise, are apparent making it look very unpleasant.

Figure 5.13: *'House1' coded by CVQ at 0.56 bpp.*

The third image, 'Lena2', is depicted in Fig. 5.14 (magnified by two). It has been coded at 0.625 bpp using a full search VQ system. It can be noticed that the perceived quality of the image in Fig. 5.14 is quite good. Most of the edges have been reproduced satisfactorily except for a few diagonal edges (for example, one side of the black triangle at the top right side of Fig. 5.14), and areas which contained fine details (i.e., on the woman's hat). However, monotone areas such as the girl's shoulder and her face have been contaminated by grid noise.

Figure 5.14: *'Lena2' coded by a full search VQ at 0.625 bpp.*

The proposed postprocessing algorithm was applied to these test images in an adaptive way. Poorly coded images were processed by applying all three steps of the proposed algorithm, whereas high quality coded images were processed by applying just one step of the proposed algorithm. The signal to noise ratio (SNR) was used as a rough indicator of the quality of the filtered images. In addition, since it is well known that the signal to noise ratio does not correlate well with subjective (human) quality assessments, numerous subjective tests have been carried out too.

The viewing conditions during the subjective tests were close to those recommended by the CCIR [CCIR 1974]. The viewing distance was set at five times image height, the screen luminance varied from 0.4 mL at its lowest level to 63 mL at its highest, and the ambient illumination was approximately 2 ft-candles. A nine inch Ikegami monitor was used with aspect ratio 4:3. The image contained 480 lines with 512 pixels per line and was presented with 2:1 interlace at a rate of 25 frames per second.

In order to obtain an idea of the typical bandwidth of monotone areas in images, an average spectrum for 1302 8x8 monotone blocks, derived from a real image, was computed. The following procedure was employed to obtain the 8x8 blocks. The block classifier, described in section 5.3, was used with a window of 8x8 pixels to test all the 4x4 blocks in the image under test. All groups of four contiguous monotone 4x4 blocks were then selected to give the 8x8 blocks. Then, the discrete Fourier transform (DFT) was applied to each monotone block of 8x8 pixels and an average spectrum was calculated. The energy of the coefficients in the first column (vertical direction) of the average DFT is depicted in Fig. 5.15. This was the column with the maximum bandwidth among all the rows and columns of the average DFT matrix. As in [Ramamurthi 1986a], we will use this one-dimensional spectrum to represent the assumed isotropic spectrum of blocks of pixels in monotone areas. It is also reasonable to assume the same spectrum to be typical for image areas near edges in the direction parallel to the edge.

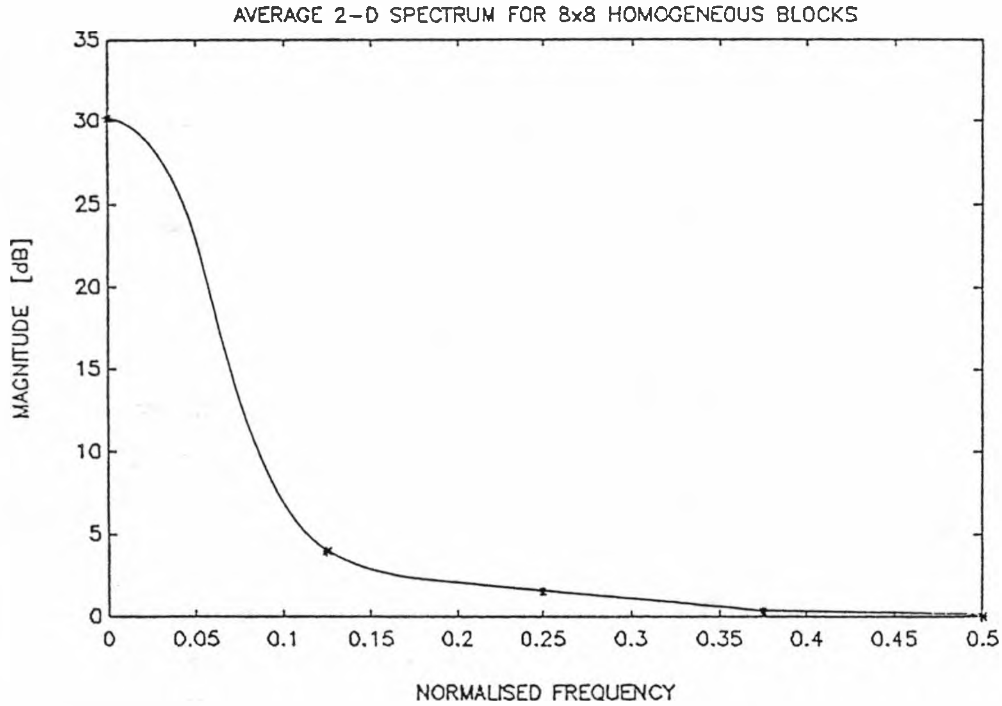


Figure 5.15: *Average one-dimensional spectrum for monotone* 8×8 *blocks.*

Three different FIR filters have been used within the proposed filtering algorithm: (*i*) a one-dimensional low-pass filter is applied parallel to the edge if the pixel under test is found to be an edge pixel, (*ii*) a two-dimensional low-pass filter is used if the pixel under test is found to belong to a monotone area, and (*iii*) a one-dimensional high-pass filter is applied perpendicularly to the edge if the pixel under test is found to be an edge pixel and the edge is a diagonal edge. In order to remove as much of the noise as possible without smoothing the signal, the passbands of the desired low-pass filters should be wide enough to accommodate the high energy portions of the spectrum in Fig. 5.15. It can be noticed that most of the signal energy is concentrated below the normalised spatial frequency of 0.25. Thus, various low-pass filters were designed having passbands below this frequency.

Since grid noise has low energy [Ramamurthi 1986a], very little stopband rejection is needed to reduce the noise amplitude below the visibility threshold. A two-dimensional 3x3 separable FIR filter with identical responses in the horizontal and vertical directions was found to be adequate for grid noise removal. That is, if $h_{gr}(x,y)$ denotes the two-dimensional filter then

$$h_{gr}(x,y) = h(x)h(y) \tag{5.2}$$

where $h(x)$ is a one-dimensional filter and $h(y) = h(x)$.
Various $h(x)$ filters were designed by employing the filter design procedure, described in appendix G. Then, the filters were tried on a test image and the subjective quality of the filtered images along with SNR results were compared. Best results were obtained for filters having a 3 dB attenuation at a normalised corner frequency of 0.205.

The frequency response of the one-dimensional filter, $h(x)$, of order 3 is depicted in Fig. 5.16 along with its coefficients. This filter was designed according to the method described in appendix G, employing the frequency weighting function proposed by Mannos and Sakrison (described in equation (G.1)). Also depicted in Fig. 5.16, is the *perceptual response* (designated "overall") which is the overall frequency response of the system consisting of the designed FIR filter and the filter which represents the human visual system (HVS). It should be noted that $h(x)$ represents a one-dimensional cross section of the separable two-dimensional filter, $h_{gr}(x,y)$, which has identical frequency response in the x and y directions.

For the one-dimensional filter $h_{st}(x)$, employed for staircase noise removal, a sharper rolloff is desirable due to the higher energy of this type of noise. An FIR filter of order 5 was chosen for this purpose in order to reduce computation and to prevent the smoothing window from extending out of the analysis image

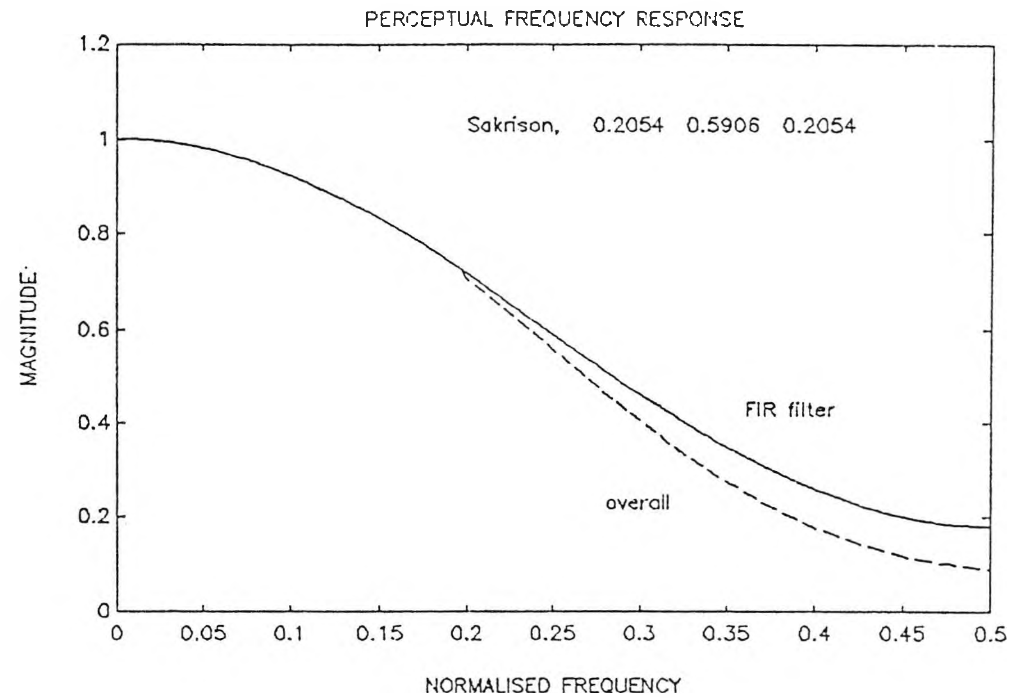


Figure 5.16: *Frequency response of the filter $h(x)$ of order 3 based on the MTF proposed by Mannos and Sakrison*

block (the 7x7 block tested by the block classifier). A similar procedure to the one described above for designing $h(x)$, was applied for designing $h_{st}(x)$. Best results were obtained for filters having a 3 dB attenuation at a normalised corner frequency of 0.195.

The frequency response of the one-dimensional filter, $h_{st}(x)$, of order 5 is depicted in Fig. 5.17 along with its coefficients. This filter was designed according to the method described in appendix G, employing the frequency weighting function proposed by Mannos and Sakrison (equation (G.1)). In a similar way, we have designed various low-pass filters, $h(x)$ and $h_{st}(x)$, by employing the frequency weighting function proposed by Nill (equation (G.2)). Frequency responses of these filters are depicted in Fig. 5.18 and Fig. 5.19, respectively. Also depicted in these figures are the perceptual responses (designated "overall") which are the overall frequency responses of the systems consisting of the designed FIR filters and the filter which represents the HVS.

In order to decide which of the frequency weighting functions is more suitable, we carried out the following experiment. Steps 1 and 2 of the proposed

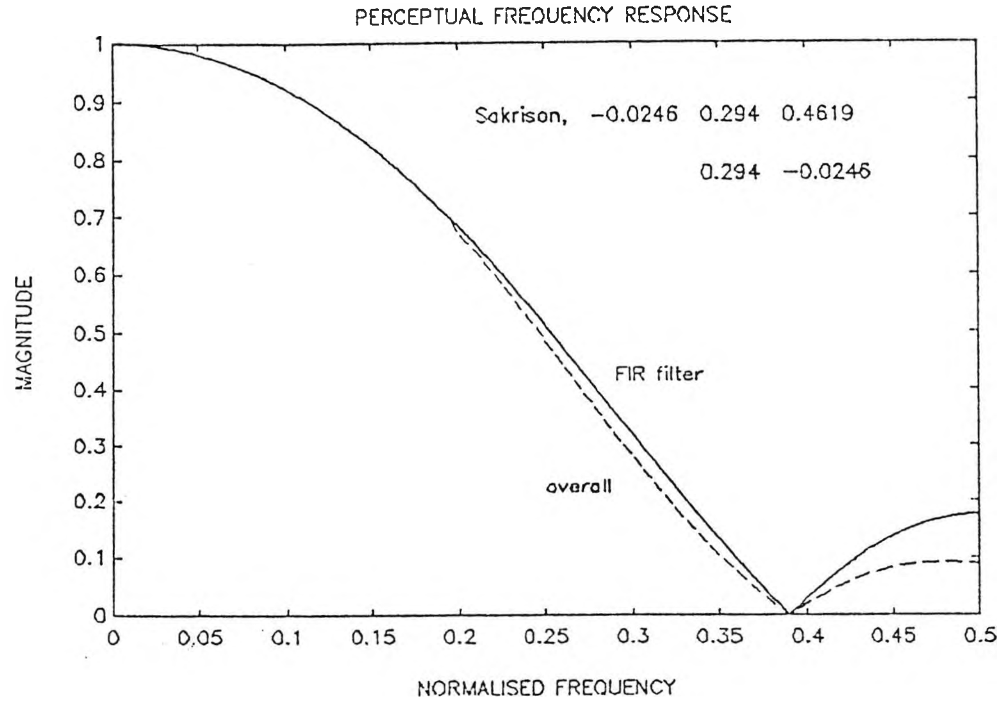


Figure 5.17: *Frequency response of the filter $h_{st}(x)$ of order 5 based on the MTF proposed by Mannos and Sakrison.*

filtering algorithm, aimed at reducing grid and staircase noise, were applied to the original (uncoded) image of 'Lena'. Due to the high quality of that image, small subimage blocks (5x5 pixels) were formed and tested by the block classifier during step 2 of the filtering algorithm (see section 5.3 for discussion).

In addition, the low-pass filter employed for reducing staircase noise, $h_{st}(x)$, was taken to be equal to the filter $h(x)$. The threshold m, used within the block classification procedure (see appendix H), was set experimentally to 2. Two sets of low-pass filters were designed. One set was designed based on the frequency weighting function, proposed by Mannos and Sakrison, while the second set was based on the function proposed by Nill. The original image was processed once by the former set of filters and once by the second set of filters. SNR results for the two filtered images are summarized in Table 5.1.

Based on these SNR results and on subjective tests of the processed images, we have selected the weighting function proposed by Nill as being more suitable. Therefore, throughout the remainder of this summary, we will present only results obtained by applying filters designed with Nill's weighting function. The

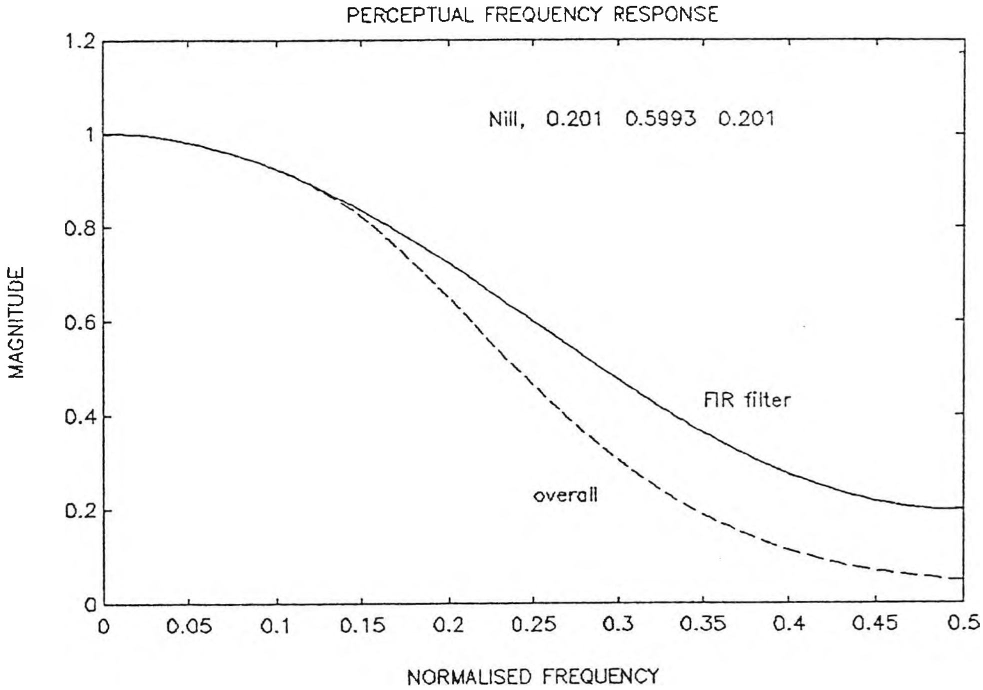


Figure 5.18: *Frequency response of the filter $h(x)$ of order 3 based on the MTF proposed by Nill.*

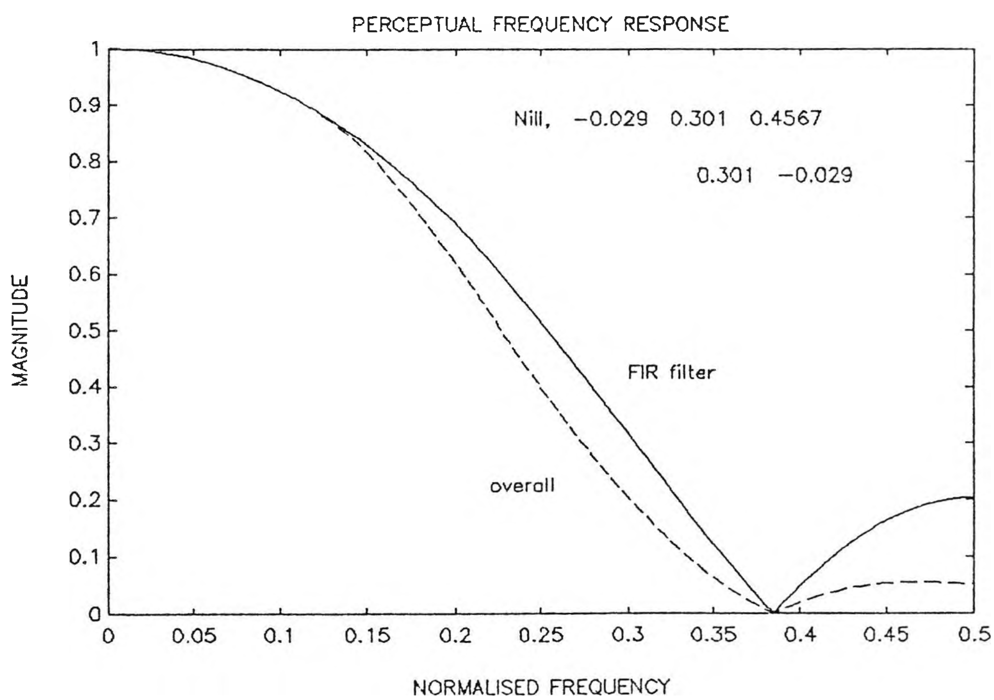


Figure 5.19: *Frequency response of the filter $h_{st}(x)$ of order 5 based on the MTF proposed by Nill.*

FREQUENCY WEIGHTING FUNCTION	SNR dB
Mannos & Sakrison	38.73
Nill	38.89

Table 5.1: SNR results for original 'Lena' processed by the proposed algorithm (steps 1 and 2)

original image, 'Lena', processed by the proposed filtering algorithm (steps 1 and 2) is depicted in Fig. 5.20 (magnified by two). It has been processed by the set of filters based on Nill's weighting function. It can be noticed that edges have been left unblurred although some fine details have been smoothed. This confirms that the filters have been adequately designed and that the proposed adaptive algorithm has been correctly applied.

Figure 5.20: *Original image after filtering it by a set of filters based on Nill's weighting function.*

The third step of the proposed filtering algorithm, the edge enhancement step, has been described in section 5.3. Two alternative edge enhancement procedures have been suggested: (*i*) applying a nonlinear transformation (defined in equation (5.1), and (*ii*) applying a high-pass filter perpendicularly to the detected diagonal edge. We present next results obtained by applying the non-linear transformation to the coded images, depicted in Fig. 5.1 and Fig. 5.13, which were previously processed by steps 1 and 2 of the proposed algorithm. Various values for both parameters, α and the size of the decision window (see section 5.3 for discussion), were tested in order to determine appropriate values which may produce best SNR results and best perceived quality. SNR results obtained by processing the coded image 'Lena1', depicted in Fig. 5.1, are presented in Table 5.2. In Table 5.3, we present SNR results obtained by processing the coded image 'House1', depicted in Fig. 5.13.

α	decision window = 3x1	decision window = 5x1	decision window = 7x1
	SNR dB	SNR dB	SNR dB
1.4	26.90	26.93	26.94
2	26.88	26.88	26.84
5	26.62	26.38	26.17

Table 5.2: SNR results for coded 'Lena1' filtered by three steps of the proposed algorithm.

α	decision window = 3x1	decision window = 5x1	decision window = 7x1
	SNR dB	SNR dB	SNR dB
1.4	27.39	27.44	27.45
2	27.39	27.45	27.47
5	27.30	27.33	27.31

Table 5.3: SNR results for coded 'House1' filtered by three steps of the proposed algorithm.

Based on these results and on subjective tests of the filtered images, we have selected proper values for the parameters, i.e., α=2 and a decision window of size 5x1. These values have produced filtered images of best perceived quality along with good SNR results.

We have also tested the alternative edge enhancement procedure, i.e., applying an high-pass filter perpendicularly to diagonal edges. Various high-pass filters were designed employing the design method described in appendix G. Those filters were applied during the third step of the proposed algorithm to diagonal edges in an adaptive way. Both SNR results and subjective assessments were compared in order to determine the proper parameters of the high-pass filter. The frequency response of the selected one-dimensional high-pass filter, $h_{en}(x)$, of order 5 is depicted in Fig. 5.21 along with its coefficients.

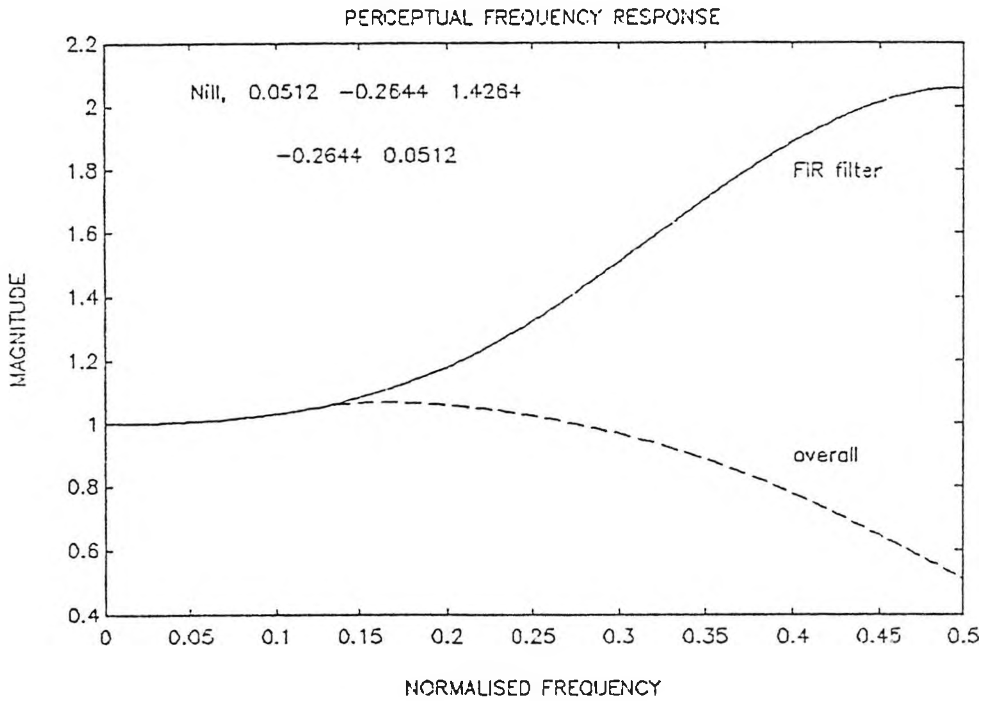


Figure 5.21: *Frequency response of the filter $h_{en}(x)$ of order 5 based on the MTF proposed by Nill.*

Also depicted in Fig. 5.21 is the perceptual response (designated "overall") which is the overall frequency response of the system consisting of the designed FIR filter and the filter which represents the HVS. It can be noticed that the perceptual response is quite "flat". In other words, the overall response of the system represents almost an all-pass filter. The low-pass nature of the HVS at high spatial frequencies compensates for the imposed enhancement of the high frequencies, thus producing high frequency restoration without degrading the perceived quality of the processed image.

Postprocessing results for the three test images, defined at the beginning of this section, are presented next. The proposed filtering algorithm was applied step by step to each of these images in an attempt to monitor changes caused by each of the algorithm's steps. SNR results were calculated and subjective tests of the processed images were carried out at the conclusion of each step. These results are presented in Table 5.4 and in the following figures. Moreover, SNR results obtained by applying the method proposed in [Ramamurthi 1986a] to the test images are compared in Table 5.4 to SNR results obtained by applying our filtering algorithm. In Table 5.4, step 1 means grid noise filtering, step 2 means staircase noise filtering, and step 3 means edge enhancement. The best SNR results are printed in boldface letters.

ALGORITHM'S STEPS	Lena1 SNR dB	House1 SNR dB	Lena2 SNR dB
Unfiltered	26.10	26.76	31.15
Step 1	26.23	26.86	**31.23**
Steps 1 & 2	26.86	27.38	30.81
Steps 1 & 2 & 3 with $\alpha = 2$	**26.88**	**27.45**	30.72
Steps 1 & 2 & 3 with HPF	26.74	27.32	30.16
Filtering method according to [Ramamurthi 1986a]	26.80	27.35	30.67

Table 5.4: SNR results for the filtered test images.

Three conclusions can be drawn from the results presented in Table 5.4 and from the following figures. First, the gradual improvement in image quality, achieved in a step-by-step manner by the proposed filtering algorithm, is quite apparent. It can be noticed that all three filtering steps should be applied to poorly coded images such as 'Lena1' and 'House1' in order to achieve best results. On the other hand, step 1 alone should be applied to high quality coded images such as the third image, 'Lena2'. These results prove the correctness of our basic concept, i.e., the modularity of the proposed filtering algorithm and the adaptive way in which it can be applied. Each step of the proposed filtering algorithm is carried out independently and can be applied only if necessary. This feature offers great flexibility which can not be found in other filtering methods (mentioned in section 5.2), and ensures good filtering results.

Second, it is quite apparent that the nonlinear edge enhancement procedure, defined in equation (5.1), offers better results than the high-pass filtering method. Third, it can be noticed that for all three test images better results have been obtained by applying our method rather than the one proposed in [Ramamurthi 1986a]. Particularly, a degradation of 0.48 dB, rather than an improvement, has been obtained by applying the method proposed in [Ramamurthi 1986a] to the third image, 'Lena2'.

The SNR for the filtered image of 'Lena1' is 26.88 dB, representing an improvement of 0.78 dB. The SNR for the filtered image of 'House1' is 27.45 dB, representing an improvement of 0.69 dB. These gains in SNR do not fully reflect the subjective improvement in quality due to the filtering process. This is not surprising since the masking of noise near edges, exploited during filtering, is not accounted for by the SNR figure. The in-band noise near edges, which is left unsmoothed by the proposed algorithm, contributes to the mean squared error (thus decreasing the SNR figure), but does not degrade the perceived quality. In order to demonstrate the dramatic change in the perceived quality of the filtered images, we present next the filtered images. The filtered images are presented in a step-by-step manner showing the gradual improvement in their perceived quality.

Magnifications of the filtered images, shown in Fig. 5.22*d* and Fig. 5.23*d*, are shown in Fig. 5.24 and Fig. 5.25, respectively. It can be seen that most of the staircase noise has been removed by smoothing. Moreover, most of the edges have been satisfactorily reproduced appearing quite sharp except in high detail areas such as the woman's hat which have been degraded by the coding process itself. Grid noise has also been almost completely smoothed by the filtering procedure. A magnification of the filtered image of 'Lena2', is shown in Fig. 5.26. This image has been obtained by applying just step 1 of the proposed filtering algorithm (grid noise filtering) to the coded image, 'Lena2'. The subjective improvement due to filtering is quite apparent when comparing Fig. 5.26 to the unfiltered image shown in Fig. 5.14.

a. Unfiltered image b. After step 1 c. After steps 1+2

d. After steps 1+2+3 nonlinear transformation

e. After steps 1+2+3 highpass filtering

Figure 5.22: *Various filtered images obtained while carrying out the filtering of 'Lena1' in a step-by-step manner.*

a. Unfiltered image b. After step 1 c. After steps 1+2

d. After steps 1+2+3 nonlinear transformation

e. After steps 1+2+3 highpass filtering

Figure 5.23: *Various filtered images obtained while carrying out the filtering of 'House1' in a step-by-step manner.*

Figure 5.24: *Magnification of the filtered image (Fig. 5.22d).*

Figure 5.25: *Magnification of the filtered image (Fig. 5.23d).*

Figure 5.26: *Magnification of the filtered coded image 'Lena2'.*

Fig. 5.27 and Fig. 5.28 show the two-dimensional spectrum of a 16x16 monotone subimage taken from the coded image, 'Lena1', before and after filtering, respectively. The reduction of the out-of-band grid noise is apparent while most of the signal content has been retained. Fig. 5.29 shows the two-dimensional spectrum of an edge subimage taken from the coded image, 'Lena1'. Fig. 5.30 shows the two-dimensional spectrum of the same edge subimage after applying all three filtering steps to the coded image (employing the nonlinear transformation for edge enhancement). Fig. 5.31 shows the two-dimensional spectrum of the same edge subimage after applying all three filtering steps to the coded image (employing high-pass filtering for edge enhancement). The reduction of the out-of-band noise in the direction parallel to the edge is apparent, while the in-band noise has been mostly retained. Also noticeable, is the restoration of high frequency elements in a direction perpendicular to the edge direction.

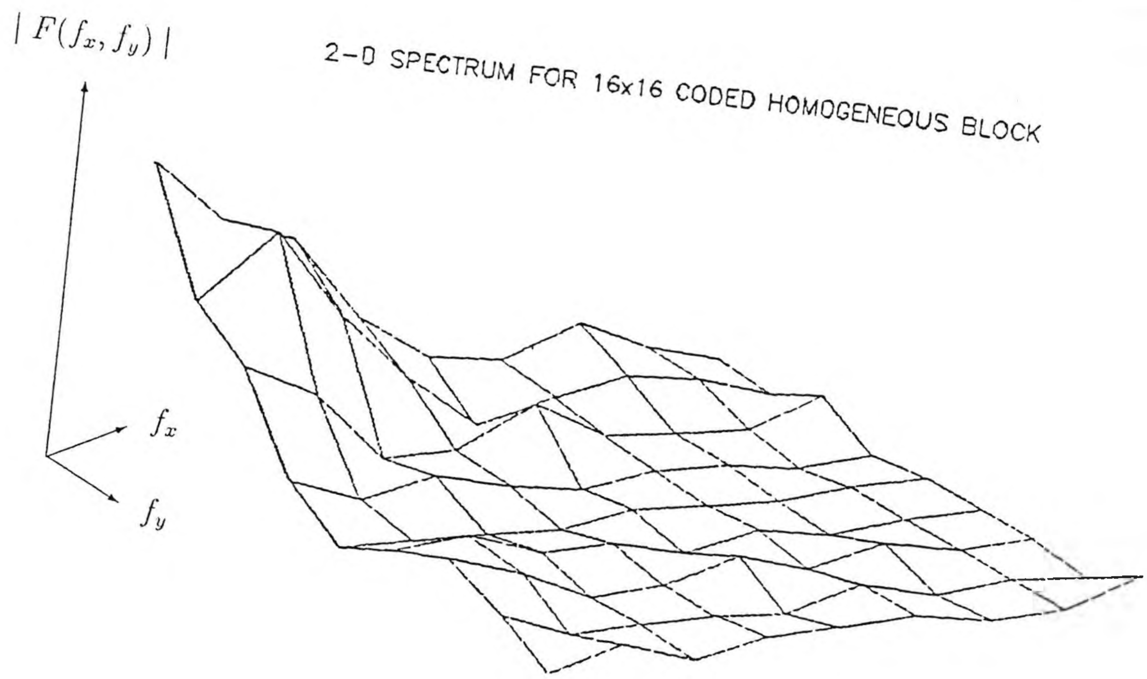


Figure 5.27: *Two-dimensional spectrum for a 16 × 16 monotone subimage from the coded image 'Lena1'.*

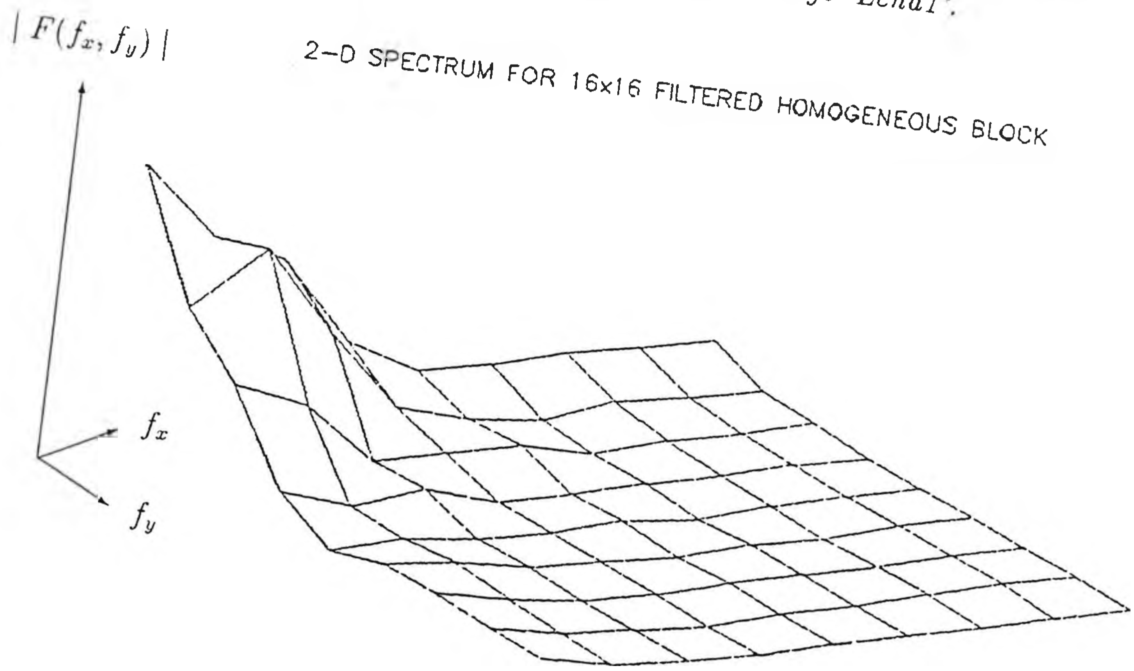


Figure 5.28: *Two-dimensional spectrum for a 16 × 16 monotone subimage from 'Lena1' after filtering.*

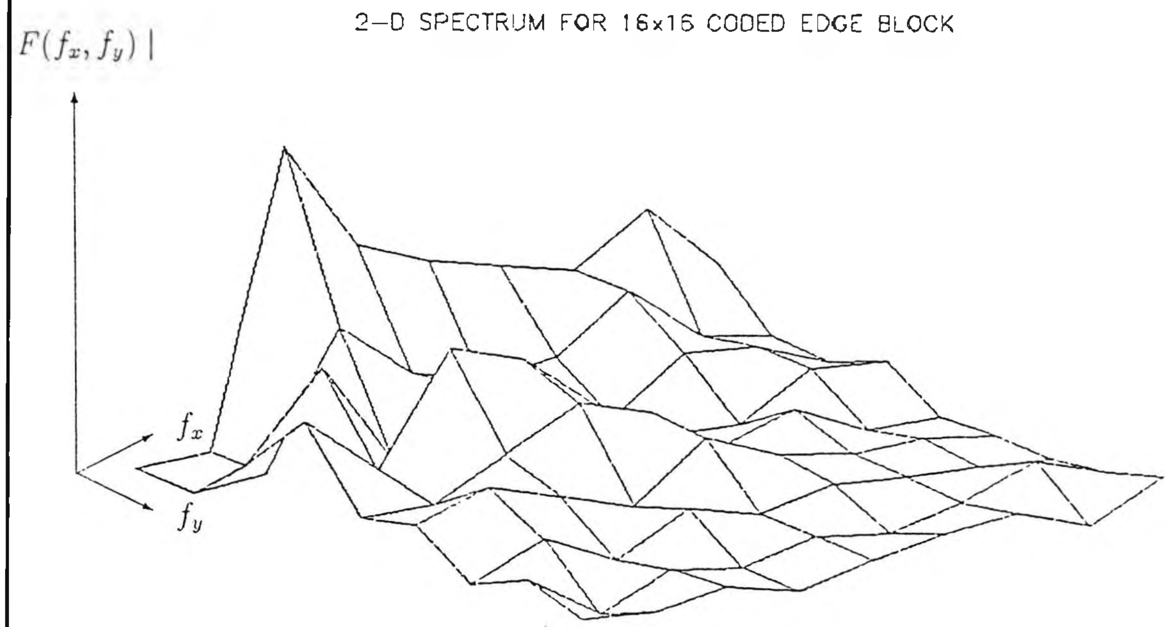


Figure 5.29: *Two-dimensional spectrum for a* 16×16 *edge subimage from the coded image 'Lena1'.*

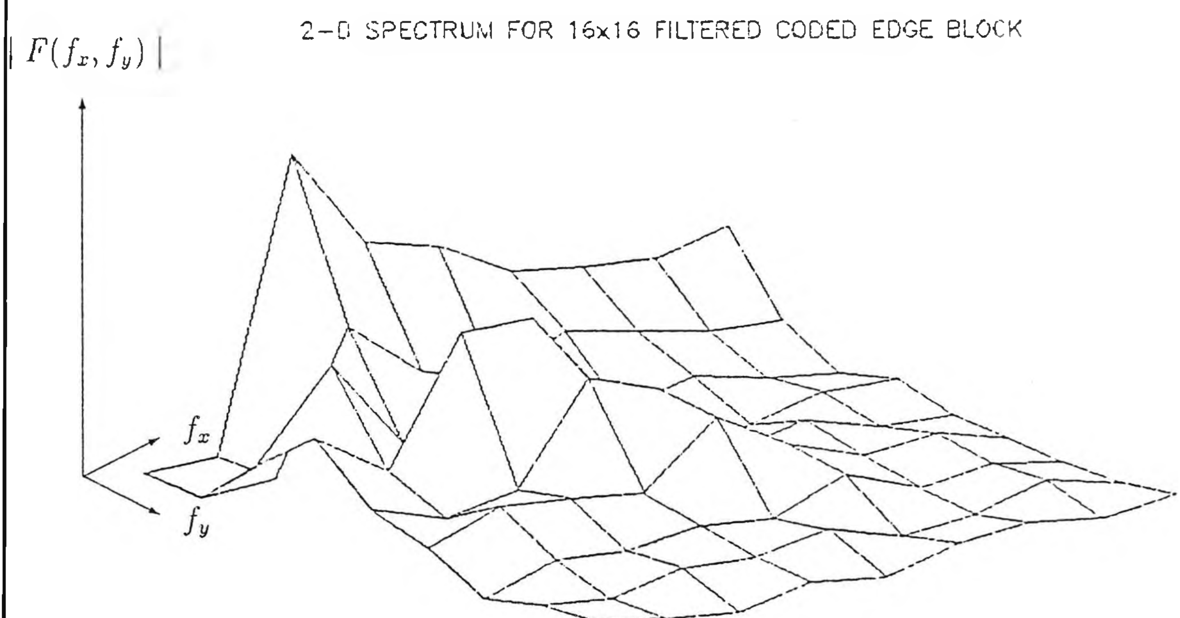


Figure 5.30: *Two-dimensional spectrum for a* 16×16 *edge subimage from 'Lena1' after filtering (employing nonlinear transformation for edge enhancement).*

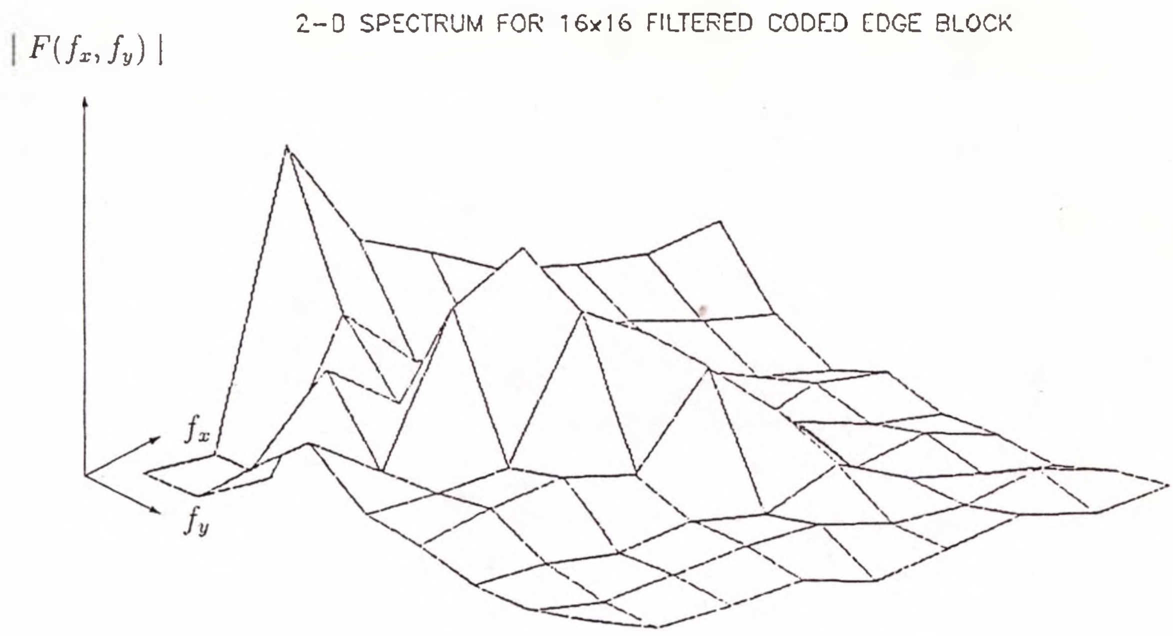


Figure 5.31: *Two-dimensional spectrum for a* 16×16 *edge subimage from 'Lena1' after filtering (employing high-pass filtering for edge enhancement).*

The proposed filtering algorithm was also applied to various images, coded by the new DCGSVQ described in chapter 4. All three filtering steps were applied using the nonlinear transformation with $\alpha = 2$ as the third step. Edge enhancement was carried out for all detected edges, i.e., vertical, horizontal, and diagonal edges. The threshold T employed by the pixel and block classifiers was set experimentally to 0.07 based on subjective tests. The grid noise in DCGSVQ-coded images may appear along the boundaries of 8x8 subimage blocks (not 4x4 blocks as in VQ-coded images); therefore, step 1 of the filtering algorithm has been appropriately modified. SNR results for the coded unfiltered and filtered images of 'House', 'Tree', and 'Splash' are presented in Table 5.5.

An improvement of 0.64 dB was achieved for the 'Tree' image (which is the "noisiest" image among the tested images) while achieving an improvement of 1.15 dB for the 'Splash' image. These gains in SNR do not fully reflect the subjective improvement in quality due to the filtering process. This is not surprising since the masking of noise near edges, exploited during filtering, is not accounted for by the SNR figure. The in-band noise near edges, which is left unsmoothed by the proposed algorithm, contributes to the mean squared error (thus decreasing the SNR figure), but does not degrade the perceived quality.

Image Type	R=0.500 bpp		
	House	Tree	Splash
Unfiltered	30.00	24.69	29.76
Filtered	30.80	25.33	30.91

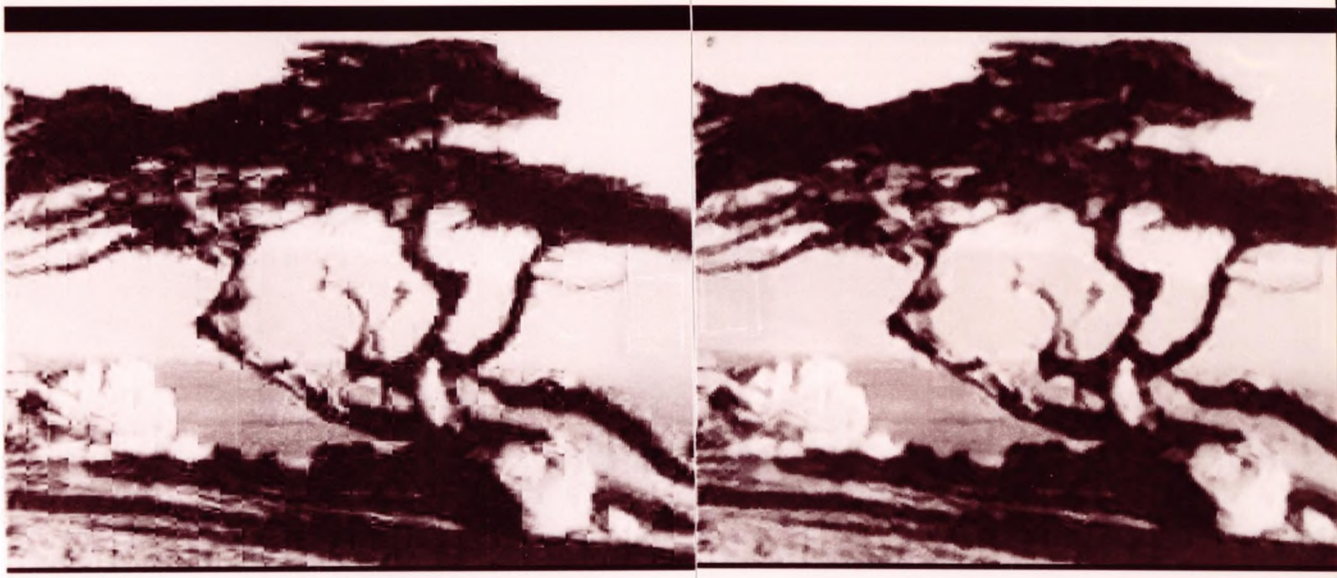
Table 5.5: SNR results for DCGSVQ-coded unfiltered and filtered images at 0.5 bpp.

To demonstrate the dramatic change in the perceived quality of the filtered images, we present next the following images. The DCGSVQ-coded image of 'House' at 0.5 bpp is shown in Fig. 5.32*a*. The general appearance of the coded image is good compared to images produced by full search VQ or CVQ at the same rate; however, some grid and staircase noise can be noticed especially along diagonal edges. The same image, after filtering, is shown in Fig. 5.32*b*. Although some fine details have been smoothed out, the edges have been cleaned of grid and staircase noise appearing nice and sharp.

a *b*

Figure 5.32: *'House' coded by DCGSVQ at 0.5 bpp*
a. before filtering *b*. after filtering

The DCGSVQ-coded image of 'Tree' at 0.5 bpp is shown in Fig. 5.33*a* while its filtered version is shown in Fig. 5.33*b*. The good performance of the proposed filtering algorithm is quite apparent.



a *b*

Figure 5.33: *'Tree' coded by DCGSVQ at 0.5 bpp*
a. before filtering *b*. after filtering

5.5 Summary

A new postprocessing algorithm, aimed at reducing grid and staircase noise in block coded images while retaining edge integrity and edge sharpness, has been described. The proposed algorithm is based on characteristics of the signal spectrum and the noise spectrum. In addition, various properties of the human visual system have been incorporated in order to improve the perceived quality of the filtered block coded images. The proposed filtering algorithm is both space-variant and nonlinear consisting of three steps: (*i*) grid noise removal, (*ii*) staircase noise removal, and (*iii*) edge enhancement. The proposed algorithm is modular and can be applied in an adaptive way depending on the quality of the block coded image. Each step stands alone and can be applied only if necessary. For example, we have shown that all three steps should be applied for postprocessing of poorly coded images while applying just one step for high quality coded images. This feature offers great flexibility, which can not be found in other filtering methods proposed in the literature, and ensures good filtering results.

Various FIR filters have been used within the proposed filtering algorithm. These filters were designed according to a new filter design method which is based on a weighted least squares design procedure. This new technique incorporates perceptual properties of the human visual system as part of the optimization process, and has been shown to be better suited for the design of image processing filters. A comparison between two frequency weighting functions, employed within the filter design method, was performed. These functions represent the modulation transfer function of the human visual system and have been used widely by other researchers. We have shown that better SNR results and better perceived quality of filtered coded images were obtained if the function proposed by Nill was used rather than the function proposed by Mannos and Sakrison.

We have also shown that better filtering results were obtained if the nonlinear edge enhancement procedure was used rather than applying high-pass filtering during step 3 of the proposed algorithm. Finally, we have shown that the proposed filtering algorithm outperforms the filtering algorithm proposed recently in [Ramamurthi 1986a]. Particularly, we have shown that a degradation of 0.48 dB rather than an improvement has been obtained by applying the algorithm of [Ramamurthi 1986a] to a coded image of moderate quality.

Employing various ways of presentation (SNR results, pictures, and figures of signal spectrum), we have demonstrated the striking improvement in image quality due to filtering. It should be remembered though that the subjective improvement with filtering is not reflected sufficiently by the SNR figures because the SNR does not account for the masking effect. The proposed filtering algorithm can be applied to coded images obtained by various block coders: VQ, BTC, or even TC if the coding block size is small. However, the filtering parameters, including the parameters of the block classifier, have to be carefully selected depending on the quality of the coding process. The proposed algorithm offers a simple way of improving the subjective quality of coded images without increasing the bit rate, and thus can be used independently at the decoder end. Moreover, it can be applied in an on-line mode while receiving the coded image (requiring just a small frame memory) or can be applied in an off-line mode depending on the user's decision.

Chapter 6

Image Quality Prediction

Strolling together along the mountain side,
Each eye sees different wind and mist.

6.1 Introduction

Various image compression systems have been implemented in the course of our research, and we have often faced the problem of deciding which is best. Identification of useful fidelity criteria for image compression system design and analysis has been a persistent difficulty for researchers. The image quality measure, actually a measure of quality degradation, that has most been used in digital image compression research is the mean square error (MSE) between the original image and the reconstructed image. However, it has often been empirically determined that the MSE or functions of it, such as the signal to noise ratio (SNR), do not correlate well with subjective (human) quality assessments [Mannos 1974], [Hall 1978]. The MSE criterion does not adequately track the types of degradation caused by digital image compression systems and it does not adequately "mimic" what the human visual system (HVS) does in assessing image quality. There is a need, therefore, for accurate measures of subjective impairment which can be used to predict image quality.

Our aim in this research is to determine such distortion measures and to test how well these measures are able to predict image quality for a set of still test images containing various coding impairments, introduced by vector quantisers. As a basis for subjectively relevant distortion measures, we would wish to test elaborate models of the processes that govern the visibility of impairments in images. We need to know how visible any arbitrary impairment is, given its location in the image, and how the visibility of all the impairments should be combined to obtain an overall quality rating.

The subjective impairment resulting from a given distortion is difficult to quantify although one reference point is available: that is the amplitude at which a degradation reaches threshold (i.e., the point at which an impairment becomes just visible). If we could accurately define when a given distortion was at threshold, we would be able to estimate the quality of images accurately, at least when the image quality is high (i.e., when coding impairments are small). Threshold vision and various factors that affect it have been described in chapter 3. This data, both for uniform and nonuniform background fields, should be accounted for by the visual model which is to form the basis of a subjectively relevant distortion measure.

6.2 Subjective Distortion Measures

Based on previous studies of threshold contrast sensitivity and on rate-distortion theory, Mannos and Sakrison investigated various distortion measures in order to find one which is in good accord with subjective evaluation [Mannos 1974]. Summarising basic properties of the HVS, they argued that after an initial nonlinear transformation, the remainder of the visual system may be considered linear over a moderate range of intensities. Taking into consideration this assumption, they described the following mathematical model.

Let $I(x,y)$ denote the intensity of a monochrome image as a function of position (x,y) in the spatial domain and let $\hat{I}(x,y)$ denote the intensity of the reconstructed image. Based on what is known physiologically and from psychophysical measurements, the authors defined the following transformation

$$L(x,y) = \mathcal{T}\left[I(x,y)\right] \quad \text{and} \quad \hat{L}(x,y) = \mathcal{T}\left[\hat{I}(x,y)\right] \tag{6.1}$$

where $\mathcal{T}[\cdot]$ is restricted to be monotonically increasing and convex $\cap$. Let $g(x,y)$ and $\hat{g}(x,y)$ be the results of operating on $L(x,y)$ and $\hat{L}(x,y)$ respectively with a linear spatially-invariant filter. The filter is defined by a transfer function

$H(f_x, f_y)$ in the spatial frequency domain or by $h(x,y)$, its impulse response in the spatial domain, i.e.,

$$g(x,y) = L(x,y) * h(x,y) \quad \text{and} \quad \hat{g}(x,y) = \hat{L}(x,y) * h(x,y) \tag{6.2}$$

where $*$ denotes convolution. Then the following distortion measure has been defined

$$d(I,\hat{I}) = \int_x \int_y [g(x,y) - \hat{g}(x,y)]^2 \, dx dy \ . \tag{6.3}$$

The linear operation, described by the transfer function $H(f_x, f_y)$, was taken to be isotropic in order to make things simpler (see discussion in chapter 3), i.e.,

$$H(f_x, f_y) = H(\omega) \ , \qquad \text{for } \omega = \sqrt{f_x^2 + f_y^2} \tag{6.4}$$

where f_x and f_y are the spatial frequency coordinates which span the two-dimensional Fourier domain.

To fit the data obtained from psychophysical measurements of sine-wave gratings, Mannos and Sakrison considered functions for $H(\omega)$ of the form

$$H(\omega) = \left[c_1 + \left(\frac{\omega}{\omega_0}\right)^{k_1}\right] \exp\left[\left(-\frac{\omega}{\omega_0}\right)^{k_2}\right] \tag{6.5}$$

where c_1, k_1 and k_2 are parameters, and ω_0 is the frequency at which the curve reaches its peak value. The authors investigated distortion measures of the form (6.3) for $\mathcal{T}[\cdot]$ monotonically increasing and convex $\cap$with different choices of the parameters ω_0, c_1, k_1, and k_2. At the conclusion of their subjective experiments, they proposed the following transfer function, $H(\omega)$, and nonlinear function, $\mathcal{T}[\cdot]$, as being best appropriate for image coding and transmission:

$$H(\omega) = 2.6\,[0.0192 + 0.114\omega] \exp[-(0.114\omega)^{1.1}] \tag{6.6}$$

having a peak of value 1.0 at $\omega_0 = 8.0$ cycles per degree (cpd) and,

$$\mathcal{T}\,[I(x,y)] = I^{0.33}(x,y) \ . \tag{6.7}$$

Functions $H(\omega)$ and $\mathcal{T}[I(x,y)]$ are shown in Fig. 6.1. The bandpass form of $H(\omega)$, with a central peak at 8 cpd and the rapid decrease on either side of this peak, is typical of the contrast sensitivity functions obtained from psychophysical experiments.

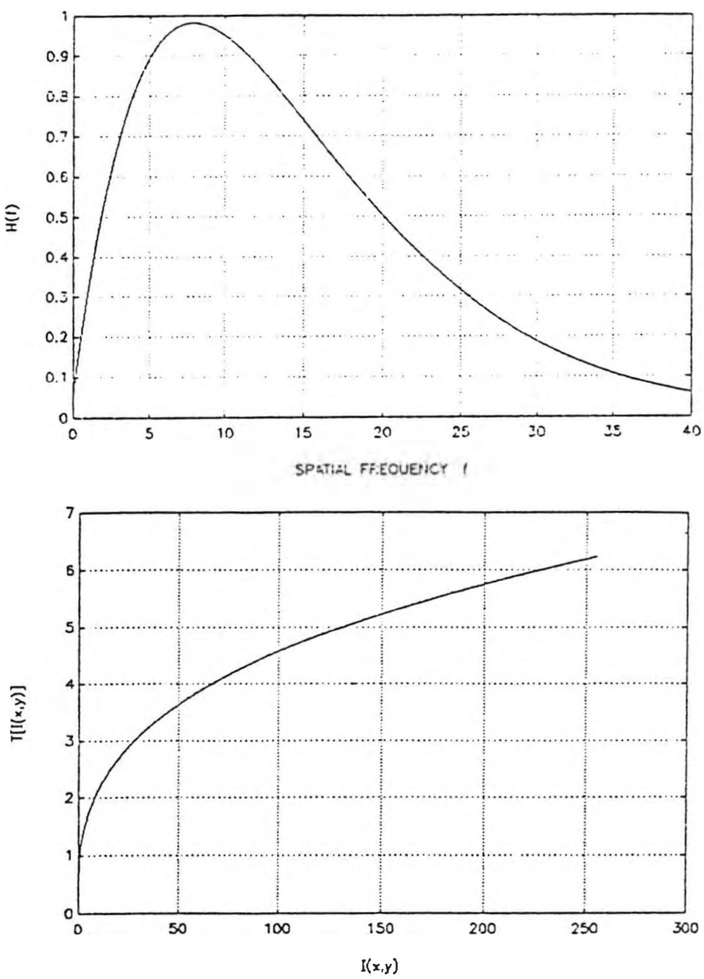


Figure 6.1: *Transfer function $H(\omega)$ and nonlinear function $T[I(x,y)]$*

A new model for the spatial frequency characteristics of the HVS was presented in [Hall 1977]. The major implication of the new model is that the system is analogous to a variable bandwidth filter which is controlled by the contrast of the input image. As input contrast increases, the bandwidth of the system decreases and vice versa. Based on this new model, Hall and Hall [Hall 1977] proposed that the function in equation (6.5) should be modified to

$$H(\omega) = \left[c_1 + \left(\frac{\omega}{\omega_0}\right)^{k_1}\right] \exp\left[\left(-\frac{\omega}{m(c)\omega_0}\right)^{k_2}\right] \tag{6.8}$$

where they have added $m(c)$, a function of contrast responsible for modifying the high-frequency roll-off.

Based on the work of Mannos and Sakrison [Mannos 1974] and DePalma and

Lowry [DePalma 1962] and on his own work, Nill [Nill 1985] proposed a third transfer function for the HVS which is a simplification of the previous proposed functions. The new function was defined by Nill to be

$$H(\omega) \; = \; [0.2 + 0.45\omega] \exp(-0.18\omega) \tag{6.9}$$

having a peak at $\omega = 5.2$ cpd. However, Nill argued that even after incorporating a visual response function in a quality measure, a further refinement is in order to mimic more closely how a human assesses quality. He added that in many cases an observer will base his/her judgement of overall scene quality on the higher structural (activity) regions contained in the scene. Thus, an improvement to an overall scene quality measure should become apparent by incorporating a weighting factor that puts more emphasis on high structure subimage areas and less emphasis on low structure subimage areas. Subimages are defined by dividing the image into contiguous non-overlapping square blocks (subimages) of 16x16 pixels. Each subimage is then transformed employing the Fourier transform and an *activity index*, that is a measure of the variability of the signal in each transformed subimage, is calculated for it. According to the value of this activity index, each transformed subimage is assigned to one of four equally populated classes, each having a weighting factor w_i associated with it.

Bringing together the preceding concepts of a visual response function and subimage structure weighting, Nill proposed the following quality measure which is defined (for now) in the 2-D discrete Fourier spatial frequency domain

$$D_v \; = \; \frac{1}{K} \sum_{i=1}^{B} w_i \sum_{f_x} \sum_{f_y} H^2(\omega) \left[F_i(f_x, f_y) - \hat{F}_i(f_x, f_y) \right]^2 \tag{6.10}$$

where

B = The number of subimage blocks,

K = A normalisation factor such as total energy,

$H(\omega)$ = A rotationally symmetric MTF of the HVS, $\omega = \sqrt{f_x^2 + f_y^2}$ where f_x and f_y are the spatial frequency coordinates which span the two-dimensional Fourier domain.

$F_i, \hat{F}_i$ = Fourier coefficients of the original and the reconstructed subimage i, respectively,

w_i = A weighting factor that is proportional to the variability of the signal in each subimage.

$w_i = 1.0$ for maximum structure subimage.
$w_i \rightarrow 0.0$ for minimum structure subimage.

Remarks :

1) An initial nonlinearity is sometimes introduced into the HVS model by preprocessing the image with a logarithmic or power function as in equation (6.7) above. However, such nonlinearity has not been introduced by Nill in the quality measure in equation (6.10) because he was particularly interested in low contrast images whereby he assumed to be working in a linear region of an overall nonlinearity.

2) The weights w_i in equation (6.10) could be assigned to the different subimages in a similar way to the one described in [Chen 1977a]. An activity index, that is a measure of the variability of the signal in each transformed subimage, could be calculated and according to its value each transformed subimage could be sorted into one of several equally populated classes. Then for each class a weight, w_i, could be defined and used within equation (6.10).

Considering the importance of the cosine transform (see discussion in chapter 4), Nill addressed the problem of how to correctly combine the MTF (given in the Fourier domain) with the cosine transform of imagery for image quality assessment. Nill tried to find a simple function that can be directly applied to the HVS model in the Fourier domain and to the image cosine transform, such that combining the two becomes both a theoretically correct procedure and a practically useful one. Applying various mathematical transforms he obtained the following function (mentioned earlier in chapter 4)

$$\hat{H}(\omega) = \begin{cases} 0.05 \exp[\omega^{0.554}] , & \text{for } \omega < 7 \\ \exp[-9(|\log_{10}\omega - \log_{10} 9|)^{2.3}] , & \text{for } \omega \geq 7 \end{cases} \tag{6.11}$$

This function can be treated in image cosine transform applications in the same manner as $H(\omega)$ would be treated in image Fourier transform applications.

Thus, for quality assessment, one simply substitutes function (6.11) for $H(\omega)$ in equation (6.10), when dealing with image cosine transforms instead of image Fourier transforms.

We have employed the cosine transform [Rosenfeld 1976] and equations (6.10) and (6.11) to obtain the following quality measure in the discrete cosine domain

$$D_q = \frac{1}{K}\sum_{i=1}^{B} w_i \sum_{f_u}\sum_{f_v} \hat{H}^2(\omega)\left[F_i(f_u,f_v) - \hat{F}_i(f_u,f_v)\right]^2 \qquad (6.12)$$

where

B = The number of subimage blocks,

K = A normalization factor such as total energy,

$\hat{H}(\omega)$ = The function defined in equation (6.11), and $\omega = \sqrt{f_u^2 + f_v^2}$ where f_u and f_v are the spatial frequency coordinates which span the two-dimensional cosine domain.

$F_i, \hat{F}_i$ = DCT coefficients of transformed original and reconstructed subimage i, respectively.

w_i = A weighting factor that is proportional to the variability of the signal in each subimage.

$w_i = 1.0$ for maximum structure subimages.
$w_i \rightarrow 0.0$ for minimum structure subimages.

The quality measure, defined in equation (6.12), has been used as a basis for the development of various quality prediction procedures, which will be described next.

6.3 Quality Prediction Procedures - *Type A* and *Type B*

The first procedure has been denoted *Type A* and is based on Nill's [Nill 1985] quality measure with the following three modifications. First, we have employed the function in equation (6.7) or a logarithmic function prior to performing the DCT on each subimage so that HVS nonlinearities are taken into consideration. Second, we have defined eight activity classes (and eight weighting factors, w_i) instead of the four classes proposed in [Nill 1985]. This decision is based on [Ngan 1982] where the author claims that he has found experimentally that better coding results are obtained with eight activity classes than with four classes as proposed in [Chen 1977a]. Third, we have defined the following two activity indices for subimage i :

$$AI1_i = \sum_{u=0}^{U-1} \sum_{v=0}^{V-1} F_i^2(u,v) - F_i^2(0,0) \tag{6.13}$$

where u and v are the orthogonal coordinates that span the cosine domain, U and V denote the number of the DCT coefficients in the u and v directions, respectively, and

$$\begin{aligned} AI2_i = \frac{1}{4(U-2)(V-2)} \sum_{j=1}^{U-2} \sum_{q=1}^{V-2} & \{[L_i(j,q) - L_i(j,q-1)]^2 + \\ & [L_i(j,q) - L_i(j,q+1)]^2 + \\ & [L_i(j,q) - L_i(j-1,q)]^2 + \\ & [L_i(j,q) - L_i(j+1,q)]^2\} \ . \end{aligned} \tag{6.14}$$

As can be noticed, $AI1_i$ is defined in the cosine domain and $AI2_i$ is defined in the spatial domain. We have tested both activity indices and compared their application within quality assessment measures.

The block diagram of *Type A* procedure is depicted in Fig. 6.2 where the processing carried on the i-th subimage is being described. In Fig. 6.2, the original image $I(x,y)$ and the reconstructed image $\hat{I}(x,y)$ are first transformed (pixel by pixel) employing a function $\mathcal{T}[\cdot]$. The transform applied is taken to be either $I^{0.33}(x,y)$ or $\ln[I(x,y)+1]$ or simply $I(x,y)$. Then, each transformed image is partitioned into subimages of 16x16 pixels which are then transformed employing the DCT. A weighting factor w_i is assigned to subimage i based on an activity index calculated for that subimage. The weighting factors w_i, defined for *Type A* procedure, are taken to be directly proportional to the activity indices that are calculated either in the spatial domain or in the spatial frequency domain. The application of this quality prediction procedure to our research has been investigated by computing D_q for images that are produced by various compression systems (vector quantisers) and comparing D_q values with human assessments of those images.

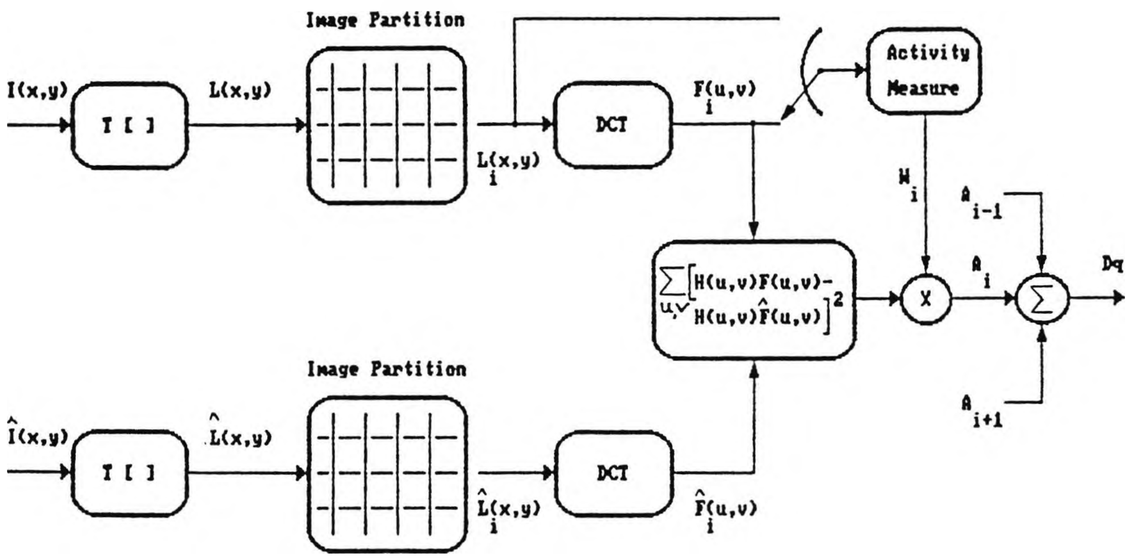


Figure 6.2: *Block diagram of Type A quality prediction procedure.*

Procedure *Type A*, described above, is based on a basic argument postulated by Nill in [Nill 1985]. Nill argued that better quality prediction results could be achieved by incorporating a weighting factor that puts more emphasis on high structure subimage areas and less emphasis on low structure subimage areas (i.e. $w_i = 1$ for subimages with high activity indices, and $w_i \rightarrow 0$ for subimages with low activity indices). Nill based this argument on a well known principle used for bit allocation within adaptive coding techniques such as the one described in [Chen 1977a]. We, on the other hand, argue that a different approach should be considered for quality prediction problems.

High activity subimages (associated with high activity indices) consist of image areas where sharp transitions of luminance occur, i.e., areas where spatial masking effects take place thus affecting the visibility of any impairment. As previously explained, these sharp transitions inhibit the ability of the eye to detect impairments that are spatially adjacent to the transitions and thus, reducing the effects of these impairments on an overall quality measure. On the other hand, impairments that occur in low activity subimages. i.e., homogeneous image areas, are more pronounced and thus should be given more emphasis within an overall quality measure. In other words, we argue that the weighting factor w_i in equation (6.12) should be inversely proportional to the activity index, calculated for subimage i, instead of being directly proportional to it, as proposed by Nill.

In order to test these arguments we have defined a second quality prediction procedure which we named *Type B*. This procedure is based on *Type A* procedure with the following modification: (i) the weighting factors are taken to be inversely proportional to the activity indices, i.e., larger weighting factors correspond to smaller activity indices and vice versa, (ii) the function $\hat{H}_{LPF}(\omega)$, defined in chapter 4, is used within equation (6.12) instead of using the function $\hat{H}(\omega)$. Both procedures, *Type A* and *Type B*, were implemented and tested by us.

6.4 Quality Prediction Procedure - *Type C*

The modulation transfer functions, described above in equations (6.6), (6.8) and (6.9), were sought to fit data obtained from psychophysical measurements of sine-wave gratings. No attempt was made to optimise the visual model for the specific task of quality prediction. In [Limb 1979], on the other hand, various two-dimensional filter functions are tested within a visual model testing their ability to predict quality. In that paper an attempt is made to determine a model of the human viewer which could then be used to predict the amount of subjective impairment for any arbitrary degradation that may be introduced in an image by digital processing methods such as image compression. An experiment is described in which different types of distortions are added to a set of still images which are then rated by a group of subjects according to the visibility of the resulting impairment. In addition, various measures of the distortion are calculated for the images. The measure yielding data points that lie closest to a smooth monotonic function on a plot of impairment ratings versus an objective measure is assumed to most accurately reflect the operation of the visual system and the human viewer. A full description of these experiments, carried out by Limb, can be found in [Limb 1979], while a summary of Limb's main results will be given in the following paragraphs.

Limb sought to determine the specifications of the spatial filtering, masking, and error summing operations that take place in the HVS by calculating various distortion measures for a set of test images. The first distortion measure, evaluated by Limb, was E_p defined by

$$E_p = \left[\frac{1}{m}\sum_x\sum_y \mid e(x,y) \mid^p\right]^{1/p} \tag{6.15}$$

where m denotes the total number of pixels in an image, and

$$e(x,y) = I(x,y) - \hat{I}(x,y)$$

with $I(x,y)$ the value of the pixel in the original image and $\hat{I}(x,y)$ the distorted pixel value in the reconstructed image. The distortion measure E_p was evaluated for $p = 1, 2, 3, 4, 6$. Varying the power, to which the error is raised before summing, changes the relative importance attached to small errors versus large errors. Limb found out that E_p is a very good estimate of impairment ratings where the type of distortion added to the images is additive white noise. Best results were found for $p = 2$ where E_2 in equation (6.15) is the well known root mean square error (RMS).

A second measure of image impairment was obtained by weighting the pixel by pixel distortion, $e(x,y)$ (raised to a power), by a weighting function designed to reflect the masking effect on the signal, i.e.,

$$EM_p = \left[\frac{1}{m}\sum_x\sum_y \frac{\mid e(x,y) \mid^p}{W(x,y)}\right]^{1/p} \tag{6.16}$$

where $W(x,y)$ is the value of the weighting function at pixel (x,y) derived from an activity function that is a measure of the variability of the signal in the neighbourhood of the pixel (x,y).

Three different forms of activity function were investigated. These functions are ad hoc guesses as to what actually happens since there is very little psychophysical literature on which to base a model. The first function A_{max} measures the maximum signal change between any pair of pixels in a neighbourhood consisting of the pixel being evaluated, (x,y), plus the eight surrounding pixels, i.e.,

$$A_{max}(x,y) = \max\{\mid I(x+k,y+l) - I(x+n,y+t) \mid\} \tag{6.17}$$

for $k, l, n, t = -1, 0, +1$.

The second function A_{ave} sums the deviations of the same neighbourhood of pixels from the neighbourhood average $\bar{I}$, i.e.,

$$A_{ave}(x,y) = \sum_{k=-1}^{1} \sum_{l=-1}^{1} \mid I(x+k, y+l) - \bar{I} \mid \quad \text{where} \tag{6.18}$$

$$\bar{I} = \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} I(x+k, y+l) \ . \tag{6.19}$$

The third function A_{df} is the weighted sum of the magnitude of the surrounding pixel differences (slopes) in both the horizontal and vertical directions and is very similar to a measure employed by Netravali and Prasada [Netravali 1977]. For this function the neighbourhood consists of the pixel being evaluated and 24 surrounding pixels (on a 5x5 grid).

$$A_{df}(x,y) = \sum_{k=x-2}^{x+2} \sum_{l=y-2}^{y+2} 0.35^{\|(k,l)-(x,y)\|} 0.5 \left\{\mid s_h(k,l) \mid + \mid s_v(k,l) \mid\right\} \tag{6.20}$$

where $\| \ (k,l) - (x,y) \ \|$ is the Euclidean distance between pixels (x,y) and (k,l),

$$\begin{aligned} s_h(k,l) &= I(k,l) - I(k-1,l) \ , \quad \text{and} \\ s_v(k,l) &= I(k,l) - I(k,l-1) \end{aligned}$$

where $I(k,l)$ is the luminance intensity of pixel (k,l). As can be noticed, the effect of surrounding pixels on the activity function decreases exponentially with the distance from the central grid point (x,y).

In all three cases, $W(x,y)$ was obtained from the activity function $A(x,y)$ so as to span a range of from 1.0 to approximately 10.0 as suggested by previous experiments, i.e.,

$$W(x,y) = \begin{cases} 10A_n(x,y) & \text{for} \quad A_n(x,y) \geq 0.1 \\ 1.0 & \text{for} \quad A_n(x,y) < 0.1 \end{cases} \tag{6.21}$$

where

$$A_n(x,y) = \frac{A(x,y)}{\max\{A(x,y)\}} \ ,$$

and $A(x,y)$ is the chosen activity function. The relation described above in equation (6.21) is depicted in Fig. 6.3.

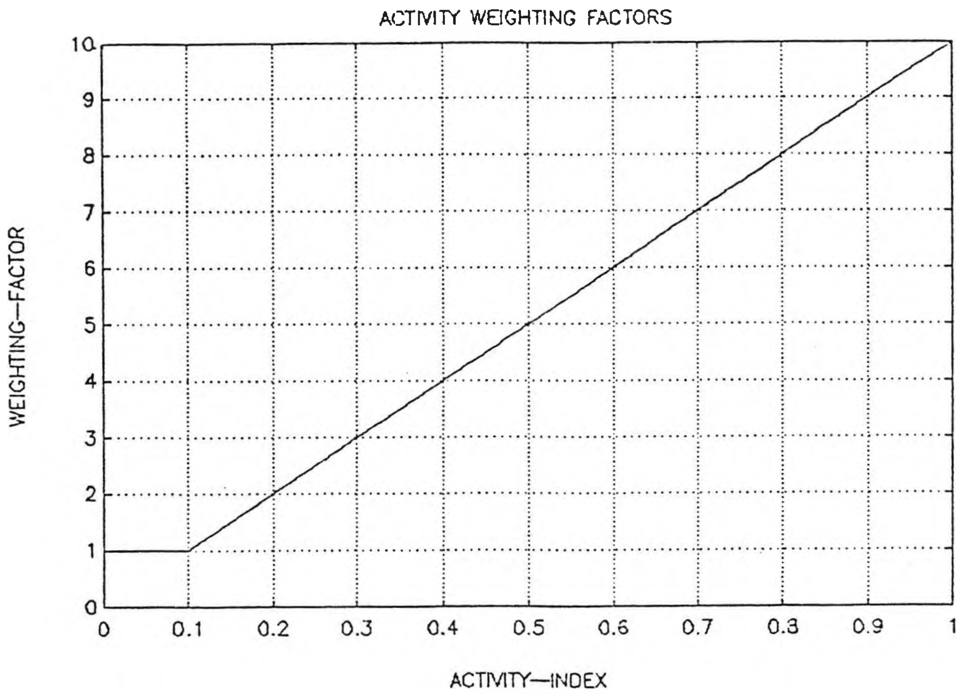


Figure 6.3: *Activity weighting factors* (from [Limb 1979])

An important result, pointed out by Limb based on these tests, is that when the spatial masking process is incorporated, the measure of picture impairment gets worse rather than better. This observation is particularly true for images where noise was added primarily at edges (as in Set 2 of the images used in [Limb 1979]). Similar results were obtained by Lukas and Budrikis in their study [Lukas 1982]. The reason for this seemingly incongruous result is based on image statistics and viewer behavior. Masking occurs largely at sharp luminance transitions, that is, at edges. Statistically, the number of pixels where masking is operative represent a small fraction of the pixels in an image. Yet, because of the nature of images, edges are precisely the areas of highest interest and so are most critical.

In rating images, subjects do not take an average of the errors over the whole image. Rather, they focus their attention on the worst areas and base their quality rating on these. This general behavior should be taken into account within a measure of image impairment, but is not. Instead, E_p and EM_p, defined in equations (6.15) and (6.16) respectively, consist of a global averaging procedure that totally belies the importance of local structures. It would seem that the solution should be to average errors over local areas of the image rather than globally.

A local measure of image impairment such as this could be calculated by summing the error in a local neighbourhood of pixels for each pixel in the image. This procedure would be very demanding computationally, thus a somewhat simpler procedure was used. The image was partitioned into a rectangular array of squares (subimages) and a local measure was calculated for each square. The size of each square was selected to be approximately the same size as the human fovea, that is 1 degree of subtended arc. The final measure of image impairment was taken to be the greatest value over all squares or the average of the two greatest values.

For the set of images, distorted by noise that was added primarily at edges, E_p and EM_p were re-calculated in a local sense following the proposed ideas. The results with and without masking were compared for different values of p and the following conclusions were drawn. First, it was shown that incorporating masking significantly improved the fit between the local measure of image impairment and the subjective impairment rating. Second, it was shown that the best results were obtained for $p = 2$. It should be noted that we are particularly interested in the results obtained for this set of images because the impairments, introduced in them, are very similar to those introduced by vector quantisers.

A third measure of picture impairment was obtained by raising to a power the weighting function $W(x,y)$ in equation (6.16), i.e.,

$$LEM_p = \left[\frac{1}{m_l}\sum_x\sum_y \left| \frac{e(x,y)}{W(x,y)} \right|^p\right]^{1/p} \tag{6.22}$$

where m_l denotes the number of pixels in a local neighbourhood.
Based on test results, Limb postulated that the errors $e(x,y)$ are normalized more correctly if the form $[e(x,y)/W(x,y)]^p$ is used rather than $e^p(x,y)/W(x,y)$, thus LEM_p was used for the rest of his experiments.

The performances of the activity functions A_{max}, A_{ave}, and A_{df} were re-evaluated using local measures. In addition, special attention was given to testing the following activity function

$$A_{avr}(x,y) = \sum_{k=-1}^{1} \sum_{l=-1}^{1} \mid I(x+k, y+l) - \bar{I} \mid^{r} \tag{6.23}$$

where $\bar{I}$ is defined as in equation (6.19). A_{avr} was evaluated for different values of r and the local measure LEM_p was calculated employing $W(x,y)$ weights derived from this activity function. Best results were obtained for $r = 2$, suggesting that visual masking is a function of the local AC power of the signal. The local measures, evaluated so far, were determined by the maximum of E_p, EM_p or LEM_p over all 1 degree neighbourhoods. By taking the average of these local measures for the worst two neighbourhoods, a small further improvement was obtained.

Various two-dimensional separable nonrecursive filters were explored by Limb in order to improve the fit between a distortion measure and subjective impairment rating. The final model of threshold vision, defined by Limb at the conclusion of his tests, is depicted in Fig. 6.4. There the error signal $e(x,y)$ is filtered first and then weighted by a weighting factor $W(x,y)$. The weighting factor is derived from the original image and calculated for each pixel (x,y) based on the activity index A_{avr} (with $r = 2$) calculated on a 3x3 neighbourhood. Then local measures of image impairment are calculated for all 1 neighbourhoods and the square root of the average of the worst two local measures is taken as the impairment measure IM.

For the set of images, in which the spectral distribution of the error varied greatly, a small amount of low-pass rather than band-pass filtering improved the fit slightly relative to no filtering. However, for the set of images in which noise was added primarily at edges, best fits were obtained with no filtering. Thus, Limb postulated that under the viewing conditions of his experiment and for the set of distortions employed, visual filtering plays a rather less important role in determining overall quality than does masking. He suspected that the cause to these findings is the type of noise added to the images under test. He argued that the noise was generally above threshold and that, consequently, suprathreshold rather than threshold sensitivity functions might be operating. In other words, since the frequency response of the visual system is flatter for more visible stimuli. filtering is less important compared to masking.

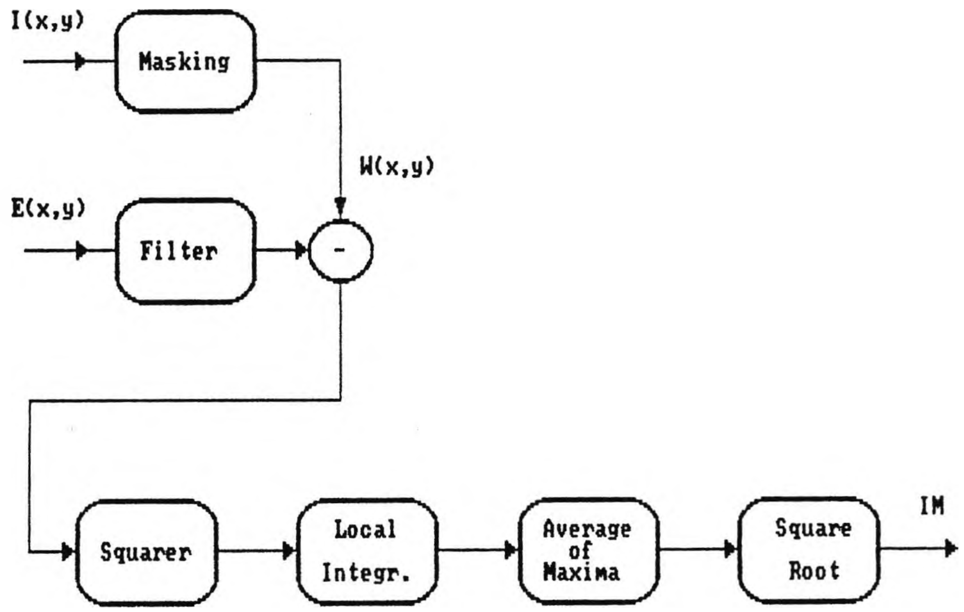


Figure 6.4: *Limb's model of threshold vision* (from [Limb 1979]).

Lukas and Budrikis [Lukas 1982] have developed a different model of vision which can be used to predict the subjective quality of moving monochrome television pictures containing arbitrary impairments. Their model, depicted in Fig. 6.5, was developed in two stages. The first stage is a nonlinear spatiotemporal model of the visual filter which describes the threshold characteristics on uniform background fields. Their model of the visual filter consists of parallel excitation and inhibition paths, each of which is separately linear but which combine in a nonlinear way. The inhibition accounts for the low-frequency decline in sensitivity in the threshold characteristics. and the nonlinearity acts as gain control, enabling the model to adapt to changes in the level of background luminance. The second stage of the visual model extends its application to nonuniform background fields by incorporating a masking function. The output of the model is the filtered error, weighted according to image content in a similar way to Limb's model. Various error powers. averaged over the image were taken as predictors of overall image quality.

In Fig. 6.5. $I(x,y.t)$ and $\hat{I}(x.y.t)$ are the original and reconstructed images respectively. $E(x.y.t)$ is the error signal based on the outputs of two nonlinear

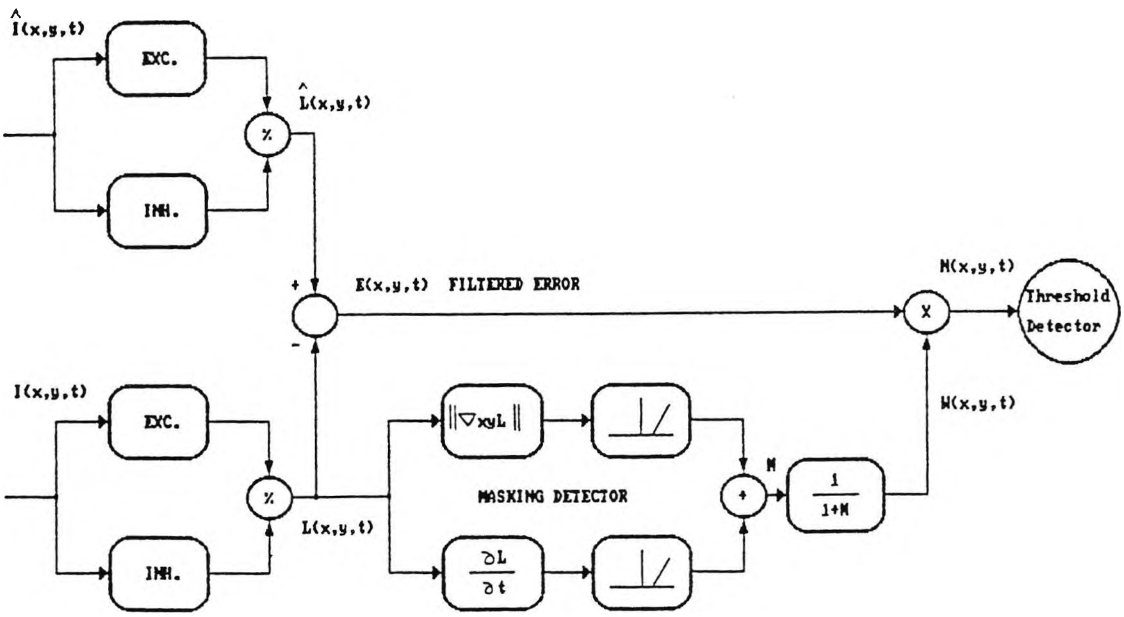


Figure 6.5: *Lukas and Budrikis' model of threshold vision* (from [Lukas 1982]).

spatiotemporal filters. $W(x, y, t)$ is the weighting function which is inversely proportional to the masking signal $M(x, y, t)$ whereas $N(x, y, t)$ is the product of weighting the error signal by the activity weighting function $W(x, y, t)$. In general, Lukas and Budrikis obtained results similar to Limb's results and the same basic conclusions have been drawn.

Encouraged by the results described so far, we have evaluated the model proposed by Limb (Fig. 6.4) and tested its performance within a quality prediction procedure, employing images produced by various vector quantisers. In addition, an extended model has been developed following Limb's suspicion that "...while the present study strongly indicates a local measure for masking effects, perhaps this should be combined with a global measure incorporating spatial filtering for additive error effects" [Limb 1979]. The extended model has been incorporated in a new quality prediction procedure named *Procedure Type C*, which is depicted in Fig. 6.6.

In Fig. 6.6, both images, the original image $I(x, y)$ and the reconstructed image $\hat{I}(x, y)$, are first processed by a nonlinear operation $\mathcal{T}[\cdot]$. We have used two

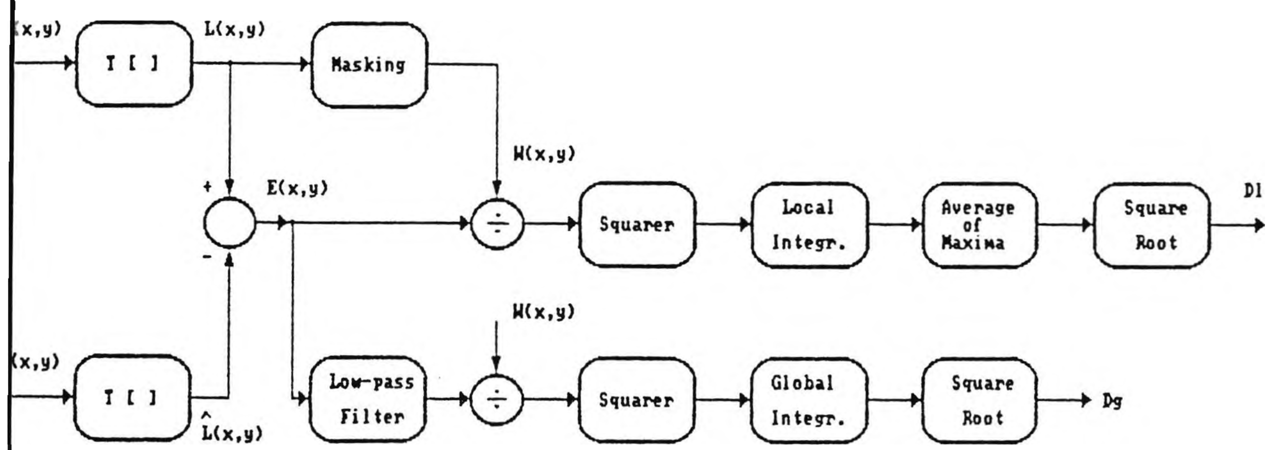


Figure 6.6: *Block diagram of Type C quality prediction procedure.*

functions to represent $\mathcal{T}[\cdot]$: a logarithmic operation, which enables the model to adapt to changes in the level of background luminance [Sakrison 1977], or an exponential function as defined above in equation (6.7). The difference between the two outputs of the nonlinear blocks is taken to be the error signal $E(x, y)$. The error signal is then processed along two parallel paths. In the first path it is weighted from point to point according to the amount of spatial activity in the original transformed image and then raised to a power of two. Local measures are calculated for small neighbourhoods and the square root of the average of the worst two local measures is taken as the measure of image impairment for masking effects, D_l. In the second path. the error signal is filtered by a two-dimensional low-pass filter. Then it is weighted from point to point according to the amount of spatial activity in the original image and then raised to a power of two. A global measure for additive error effects, D_g.is calculated and then combined with D_l to form the desired quality measure. The two measures, D_l and D_g, could be combined as follows :

$$D_{v1} = \frac{1}{2}(D_l + D_g) \quad \text{. or as} \qquad (6.24)$$

$$D_{v2} = \sqrt{D_l^2 + D_g^2} \quad . \tag{6.25}$$

A fourth procedure for quality prediction, based on different ideas described throughout this report, has been developed and tested . We named it *Procedure Type D* and will describe it next.

6.5 Quality Prediction Procedure - *Type D*

According to procedure *Type B*, an activity index is calculated for each subimage and based on its value the subimage is assigned to one of eight activity classes. Then, a weighting factor associated with each activity class is used within equation (6.12) and D_q is calculated. The weighting factors have been defined so as to span a range of from 0.0 to 1.0 with low values (w_i= 0.125, 0.250 etc.) assigned to high activity classes and high values (w_i = 0.875, and 1.0) assigned to low activity classes. These weights were defined in an adhoc way whereas in procedure *Type C* they were defined according to previous subjective experiments. Thus, instead of assigning subimages to activity classes, as in procedures *Type A* and *Type B*, we have used equation (6.21), within procedure *Type D*, to find a weighting factor for each subimage. In addition, a third activity function, defined in equation (6.23) with $r = 2$, has been tested within procedure *Type D*.

6.6 Subjective Tests and Simulation Results

Among the great variety of known psychometric methods, three main types have been used for television. One is the *comparison method* in which the magnitude of one kind of impairment is varied until it is judged to be equal in effect to a fixed magnitude of another kind of impairment adopted as a reference. The second is the *discrimination method* which seeks to establish either the threshold magnitude at which an impairment just becomes visible, or else the sequence of small changes of magnitude which produce just noticeable differences. The third is the *opinion-rating method* in which, in the simplest case, several different magnitudes of an impairment are applied in random order to an image and the observer rates each by selecting one of a predetermined set of opinions arranged to form a rating scale. Each of the methods has its proper field of application and each has many possible variations both in experimental technique and in analysis and presentation of results.

In the field of visual communications much work has gone into digital image coding aimed at reducing the bit rate requirements for image transmission or image storage. Experience has shown that each coding scheme is subject to its own unique set of impairments that are often difficult to characterize. The complicated way in which these impairments depend upon image content and quantisation method makes it difficult to optimise the quantisation characteristics for each of the impairments separately. Therefore, in general, design approaches of coding schemes tend to be heuristic, and lengthy, expensive and time-consuming subjective (psychometric) tests are often required to optimise parameters. Consequently, it is often difficult to compare results obtained in different laboratories because of the diversity of the methods employed. This state of affairs is correctly described in [Prosser 1964] where it is stated that: "...it almost seems that there must exists a psychological law to the effect that no two psychometrists will ever knowingly use the same method".

To avoid incomparable results, we have used throughout this research a method similar to the methods used by Lukas and Budrikis [Lukas 1982], by Limb [Limb 1979] and by Prosser, Allnatt and Lewis [Prosser 1964]. The method belongs to the opinion-rating class with *quality* chosen as the subject, because the ultimate judgement of a system is given by the viewer's opinion of the overall quality of the image produced by the system under test. The major difference between our experiment and theirs lies in the amount of resources available for carrying out such elaborate large scale tests that only few organisations could afford to undertake. For example, forty subjects took part in the subjective tests carried out by Lukas and Budrikis while thirty three subjects took part in Limb's tests. On the other hand, only three subjects took part in our subjective tests due to the limited resources available to us. Therefore, results presented herein should be considered carefully keeping in mind that more work should be carried out in order to derive well established conclusions.

6.6.1 Choice of Test Images

The source image for this experiment was 'Lena' which is shown in Fig. 6.7. From the one source image a set of 17 test images were generated. These test images were designed to contain a wide range of types and amounts of impairments typical to the distortions introduced by low bit rate vector quantisers. Different VQ systems (full search VQ, GSVQ, and CVQ) were employed to generate reconstructed images at rates from 0.5 to 0.625 bpp. The vectors encoded by the VQ systems were 16-dimensional vectors generated from small image blocks of 4x4 pixels. The best reconstructed image at rate 0.625 bpp is shown in Fig. 6.8*a*, generated by a full search VQ, whereas the worst reconstructed image at rate 0.5 bpp is shown in Fig. 6.8*b*, generated by a CVQ system. Both images are magnified by a factor of two in order to show clearly the differences between them.

The two most typical impairments caused by VQ systems are: the *staircase effect* and the *blocking effect.* The first one refers to the poorly coded edges that appear jagged in low bit rate reconstructed images. The cause of the staircase effect is the fact that there are too few codevectors, in the codebook used by the VQ, that contain parts of edges in them to represent the great variety of edges that must be coded. Therefore, the intensity change that occurs across an edge in the original image occurs instead at the block boundaries in the coded image, making the block boundaries visible. These block boundaries form the "steps" of the staircase. The second type of impairment, blockiness, is found in image areas where the intensity changed gradually in the original image. The reason again is the small number of codevectors that contain gradual changes of intensities in them, and the fact that each block is encoded independently of its neighbours (i.e., memoryless VQ).

6.6.2 Subjective Tests

In order to determine accurately quality ratings, the standard CCIR procedure [CCIR 1974] was modified somewhat in a similar way to the experiment described in [Lukas 1982]. The subjective tests consisted of three stages: a *preview, image evaluation*, and a *review.* During the preview the subject was

Figure 6.7: *'Lena', the source image.*

a b

Figure 6.8: *The best and worst reconstructed test images.*

shown a sample of test images that included some of the best and some of the worst in the complete set. He was asked to define his quality ratings such that the best images were given quality ratings close to "10" and the worst quality ratings close to "0" on the scale shown in Fig. 6.9. Following the preview, the subject was presented with images in random order and asked to rate them according to quality. He was allowed to view each image for as long as he liked. This was called the *evaluation stage*. Upon completion of the evaluation stage, the images were sorted and then represented in ascending order of quality rating. This gave the subject the opportunity of comparing successive images and reviewing his ratings. Finally, the ascending presentation was followed by a presentation of the images in descending order to complete the test.

10 9	VERY GOOD
8 7	GOOD
6 5 4	FAIR
3 2	BAD
1 0	VERY BAD

Figure 6.9: *Quality scale* (from [Lukas 1982])

The quality rating scale is not standard in the field. It is, however, the familiar 0-to-10 scale that is used whenever something is rated "out-of-ten". The larger number of categories allows for finer precision of quality rating, provided that subjects are able to discriminate that many categories. Other grading scales could be found in [Limb 1979] and [Prosser 1964] where *quality* and *impairments* are the two possible subjects. *Quality scales* are suitable for judging the overall quality of an image irrespective of the number of different impairments that may be present. *Impairment scales*, on the other hand, are suitable for judging a single impairment, which must be named if others are also present. For the present purpose. quality is the subject of chief interest therefore the scale in Fig. 6.9 has been used in our experiments.

The viewing conditions were close to those recommended by the CCIR. The viewing distance was set at five times the image height, the screen luminance varied from 0.4 mL at its lowest level to 63 mL at its highest, and the ambient illumination was approximately 2 ft-candles. A nine inch Ikegami monitor was used with aspect ratio 4:3. The image contained 480 lines with 512 pixels per line and was presented with 2:1 interlace at a rate of 25 frames per second.

6.6.3 Regression and Functional Relationship

Having determined quality ratings for the set of test images, our aim in this research was to develop a procedure for predicting these ratings as closely as possible. That is, we have sought a quality prediction procedure such that a smooth functional relationship would exist between it and the quality rating, regardless of the type or amount of distortion in the image. The selection of a method for fitting a smooth monotonic curve to the data points relating subjective quality rating to an objective error measure is somewhat arbitrary in that the notion of smoothness is not clear enough. Nevertheless, we have followed Limb [Limb 1979] and Lukas and Budrikis [Lukas 1982] and considered the following four functions as candidate functions :

1) the quadratic $Q(z) = az^2 + bz + c$,

2) the exponential $Q(z) = a\exp(bz)$,

3) the logistic function $Q(z) = \frac{a}{1+\exp[(z-a)/b]}$, and

4) the gaussian $Q(z) = a\exp(bz^2)$.

The logistic function has been strongly advocated to fit image quality data in the past [Prosser 1964]. It has the attractive property of being initially flat for low values of error, then falling over a midrange, and finally asymptoting to a value of 0 at high values of error. However, it leads to nonlinear regression procedures and thus has been dropped. The other three functions, on the other hand, lead to simple linear regression procedures and thus have been used. Perhaps because it has an extra degree of freedom, the quadratic function was found to fit the data better than the exponential function and the gaussian function (also pointed out in [Lukas 1982]). Therefore, we will describe in detail the results for quadratic regressions and will compare the best results obtained for it with results obtained for exponential and gaussian fits.

The root mean square (RMS) error between the original image and the reconstructed image is a benchmark against which other quality measures may be compared. As an example, therefore, a plot of quality ratings versus RMS errors for all the test images is shown in Fig. 6.10. The full line curve is a minimum mean-square fit of a quadratic function to all data points which are marked by *. It should be noted that in Fig. 6.10 subjective quality ratings are being plotted against normalised quality measures (errors). Normalisation can improve the accuracy of the final results and thus will be used throughout our presentation of results. It is carried out by dividing any error value (computed quality measure) by the greatest error value found for the set of data points used for the regression procedure. The *mean square deviation* (MSD) about the regression line is used as the performance index for the quality prediction procedures evaluated in this research. Results for the four procedures described earlier in this chapter will be presented next and the best quality prediction procedure will be determined based on MSD values calculated for each procedure.

6.6.4 Simulation Results

More than 540 different computer runs were carried out in order to evaluate the procedures under test. Different parameters within each procedure were changed and, for each combination of parameters, quality measures were calculated for every test image. For a given combination of procedure parameters, the set of data points used for regression consisted of quality measures calculated for each test image versus subjective quality ratings. Then, a quadratic regression was carried out for each set of data points and the MSD about it was calculated and used as the performance index.

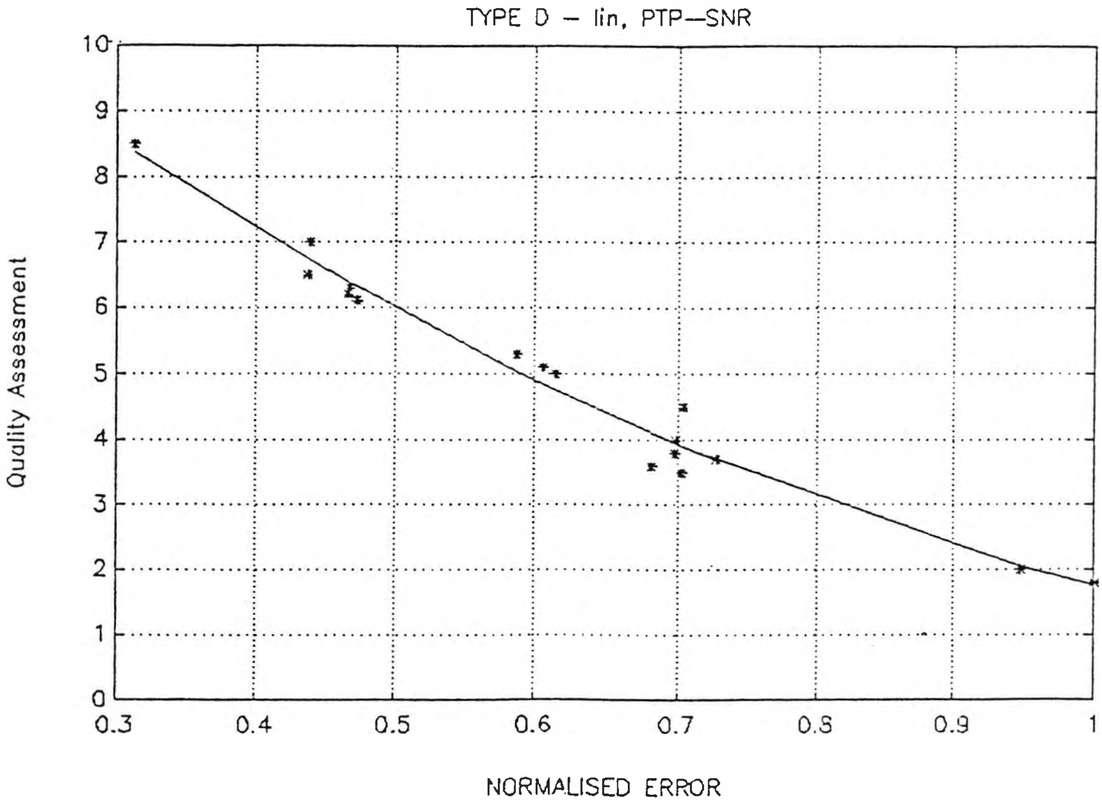


Figure 6.10: *Quality Rating Versus RMS Errors - Quadratic Regression.*

Type A Procedure

As shown in Fig. 6.2, the original image $I(x,y)$ and the reconstructed image $\hat{I}(x,y)$ are first transformed (pixel by pixel) employing a transform function $\mathcal{T}[\cdot]$. The transform applied is taken to be either exponential, $I^{0.33}(x,y)$, or logarithmic, $\ln[I(x,y)+1]$, or linear, $I(x,y)$, which means no transformation. In addition, two activity indices are used, $AI1$ in the cosine domain, and $AI2$ in the spatial domain. Simulation results, MSD values about a quadratic regression line, for the possible combinations of transform functions and activity indices are given in Table 6.1.

Three observations could be made from the results in Table 6.1. First, it is noticed that better prediction is obtained when $AI2$ and not $AI1$ is used as the activity index within *Type A* quality prediction procedure. This observation is true for all the transform functions employed within *Type A* procedure. Second, it is noticed that the best prediction results are obtained when a logarithmic

Transform Function	Activity Index	
	$AI1$	$AI2$
Linear	0.1552	0.1386
Exponential	0.0963	0.0848
Logarithmic	0.0801	0.0743

Table 6.1: MSD values for *Type A* procedure - quadratic regression.

transform function is used prior to performing the DCT. The exponential transform function yields the second best results for the two activity indices employed. The third observation is that the worst results are obtained when no transform function is employed, prior to performing the DCT, and when $AI1$ is used as the activity index. This combination of parameters is the combination proposed by Nill in [Nill 1985], which has been the basis of *Type A* procedure.

The full line curve in Fig. 6.11 is a minimum mean-square fit of a quadratic function to all data points obtained for the best combination of parameters, i.e., logarithmic transform with $AI2$ as the activity index. In Fig. 6.12, a similar curve is plotted for the worst combination of parameters, i.e., linear transform function with $AI1$ as the activity index. The solid line curve in Fig. 6.13 is a minimum mean-square fit of an exponential function to all data points obtained for the best combination of parameters, i.e., logarithmic transform with $AI2$ as the activity index. The dashed line curve in Fig. 6.13 is a similar plot employing a gaussian function to best fit the data points obtained for the best combination of parameters. As can be noticed from Fig. 6.11 and Fig. 6.13, the best fit is obtained with a quadratic function. Detailed presentation of the comparison between the regression functions involved will be given at the conclusion of this chapter.

Type B Procedure

As previously explained, this procedure is basically identical to *Type A* procedure with only two modifications. The first modification is based on our argument that the weighting factors w_i in equation (6.12) should be inversely proportional

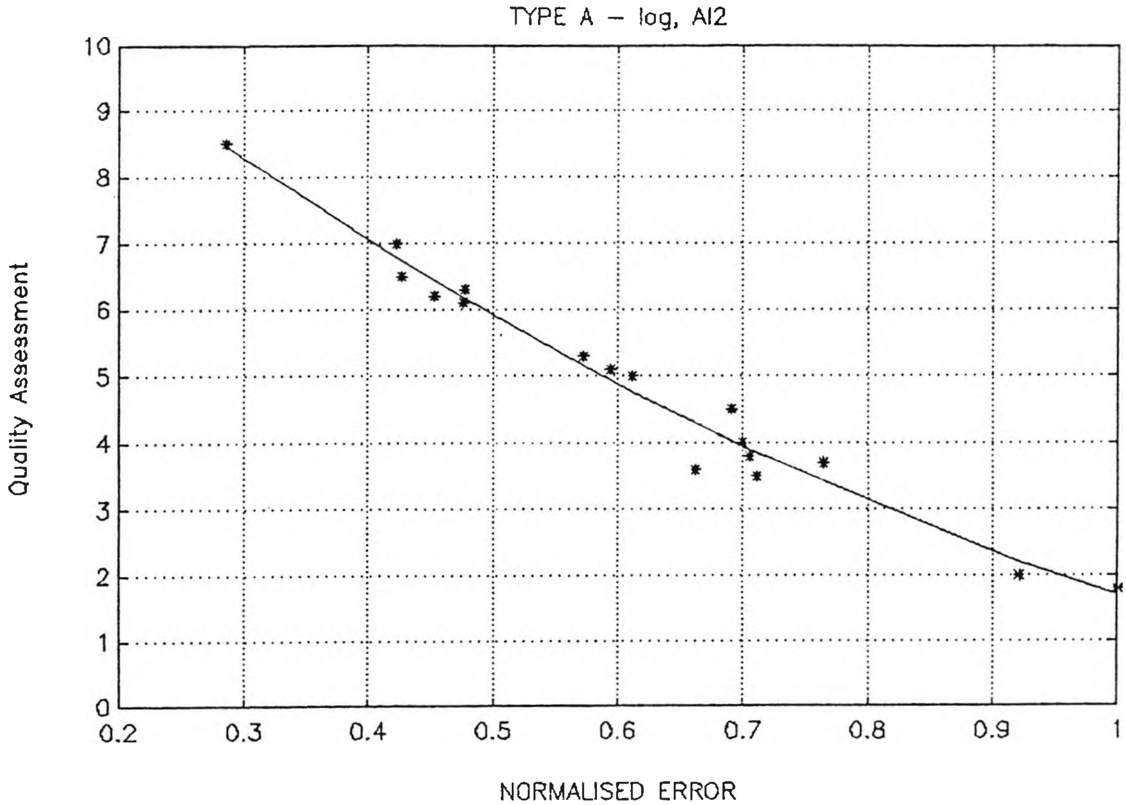


Figure 6.11: *Best quadratic fit for Type A procedure.*
(logarithmic function and AI2)

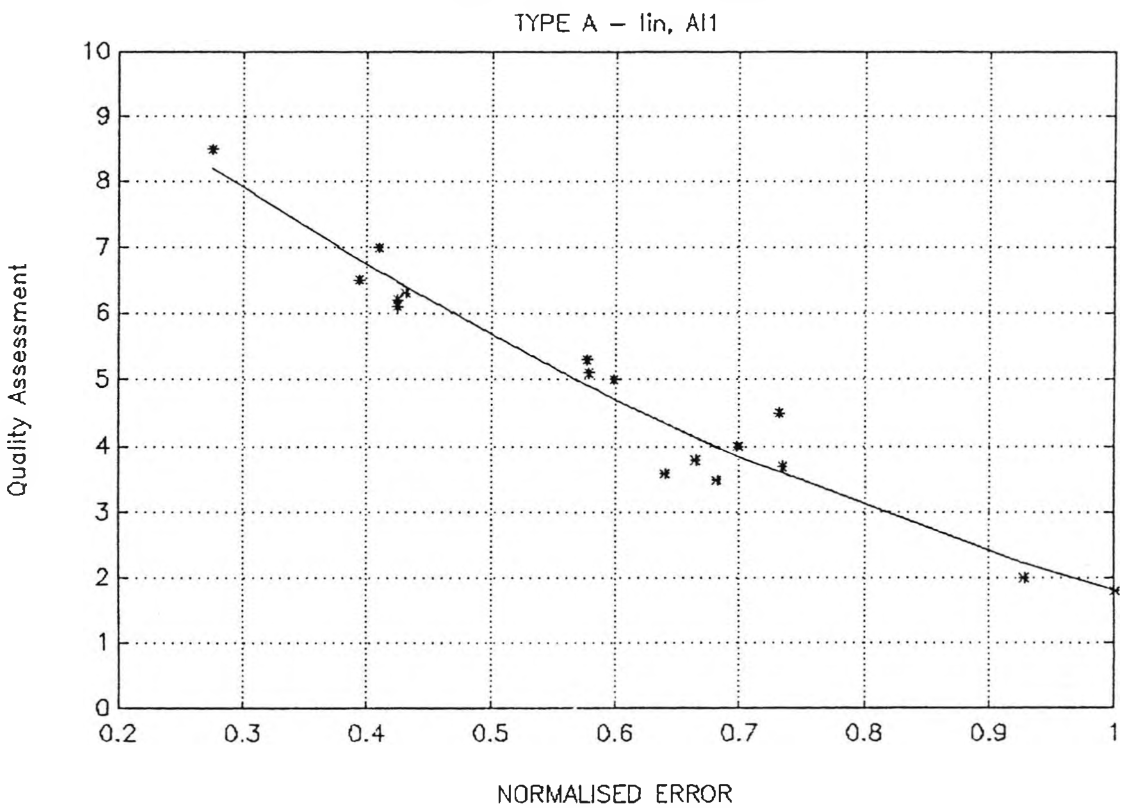


Figure 6.12: *Worst quadratic fit for Type A procedure.*
(linear function and AI1)

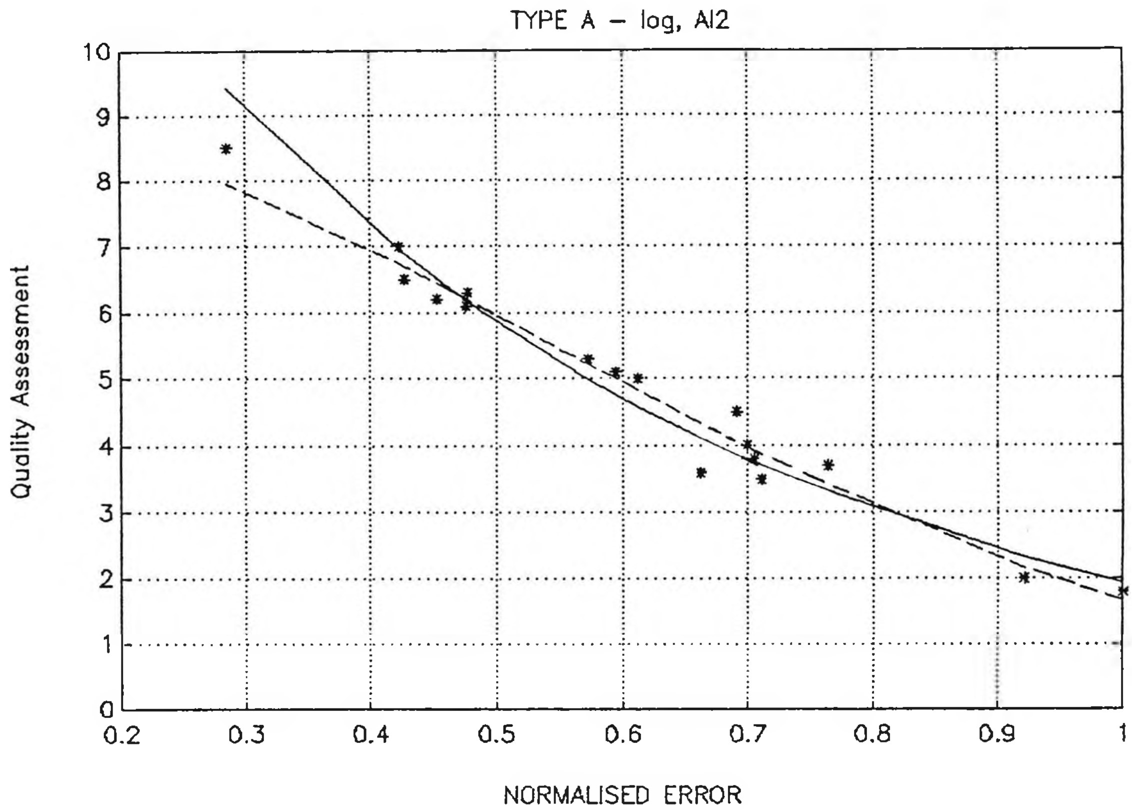


Figure 6.13: *Best exponential and gaussian fits for Type A procedure. (logarithmic function and AI2)*

to the activity indices instead of being directly proportional to them, as proposed in *Type A* procedure. The second modification is the employment of a LPF rather than a BPF to weight the errors in the cosine domain (see equation (6.12). Simulation results, MSD values, for the possible combinations of transform functions and activity indices are given in Table 6.2.

Three observations could be made from the results in Table 6.2. First, it is noticed that better prediction is obtained when $AI1$ is used as the activity index within *Type B* quality prediction procedure. This observation is true for all the transform functions employed within *Type B* procedure. Second, it is noticed that the best prediction results are obtained when an exponential transform function is used prior to performing the DCT. The logarithmic transform function yields the second best results for the two activity indices employed. The third observation is that the worst results are obtained when no transform function is employed prior to performing the DCT, and when $AI2$ is used as the

Transform Function	Activity Index	
	$AI1$	$AI2$
Linear	0.0228	0.0386
Exponential	0.0123	0.0149
Logarithmic	0.0137	0.0143

Table 6.2: MSD values for *Type B* Procedure - quadratic regression.

activity index. This observation is in accord with a similar observation made for *Type A* procedure (except that the worst results are obtained for $AI1$ within *Type A* procedure).

A comparison between the results in Table 6.1 and the results in Table 6.2 shows that better prediction results are obtained if the weighting factors w_i in equation (6.12) are taken to be inversely proportional to the activity indices. as postulated by us. In fact, it was found that results obtained for *Type A* procedure are even worse than those obtained by setting all the weighting factors equal to 1 within *Type B* procedure (i.e., ignoring masking effects). For example. setting all w_i values to 1 and employing an exponential function as the transform function within *Type B* procedure, yields a MSD value of 0.0353 which is less than any MSD value found in Table 6.1.

The full line curve in Fig. 6.14 is a minimum mean-square fit of a quadratic function to all data points obtained for the best combination of parameters, i.e., exponential transform with $AI1$ as the activity index. To conclude our findings for *Type B* procedure, the following plots are shown in Fig. 6.15. The solid line curve there is a minimum mean-square fit of an exponential function to all data points obtained for the best combination of parameters. i.e., exponential transform with $AI1$ as the activity index. The dashed line curve in Fig. 6.15 is a similar plot employing a gaussian function to best fit the data points obtained for the best combination of parameters. As can be noticed from Fig. 6.14 and Fig. 6.15, the best fit is obtained with a quadratic function. Detailed presentation of the comparison between the regression functions involved will be given at the conclusion of this chapter.

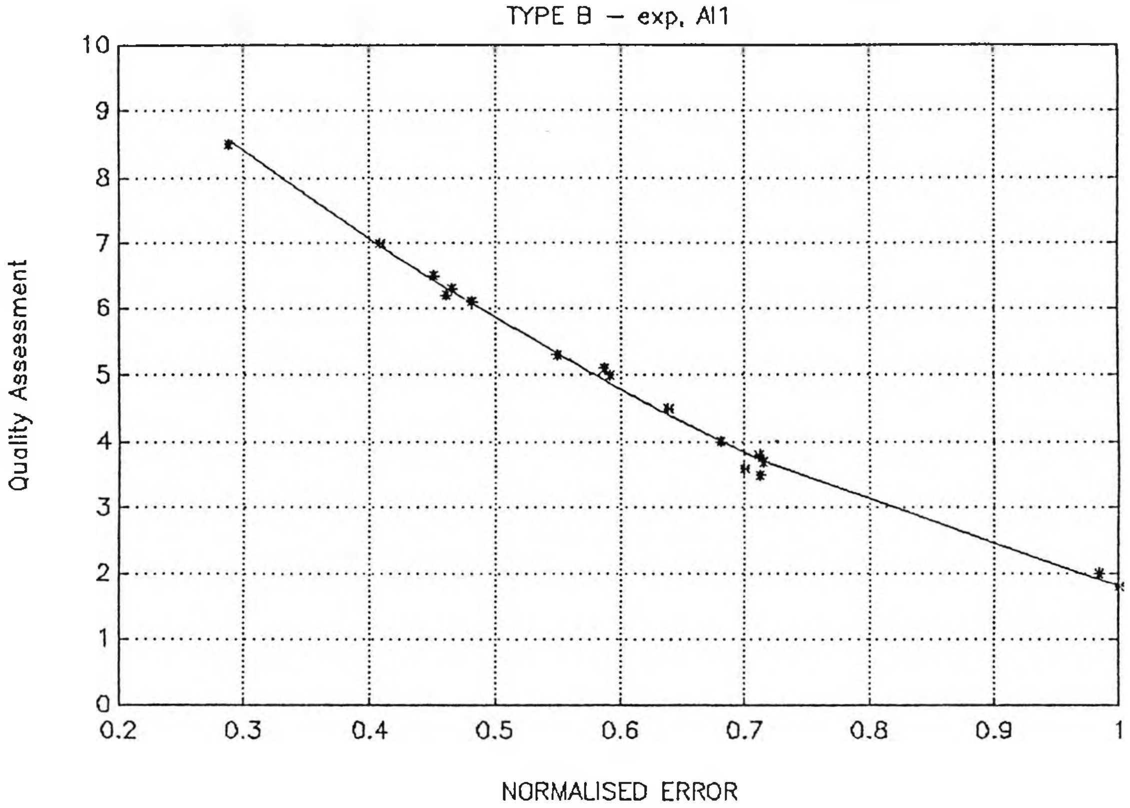


Figure 6.14: *Best fit for Type B procedure - quadratic regression.*
(an exponential transform and AI1)

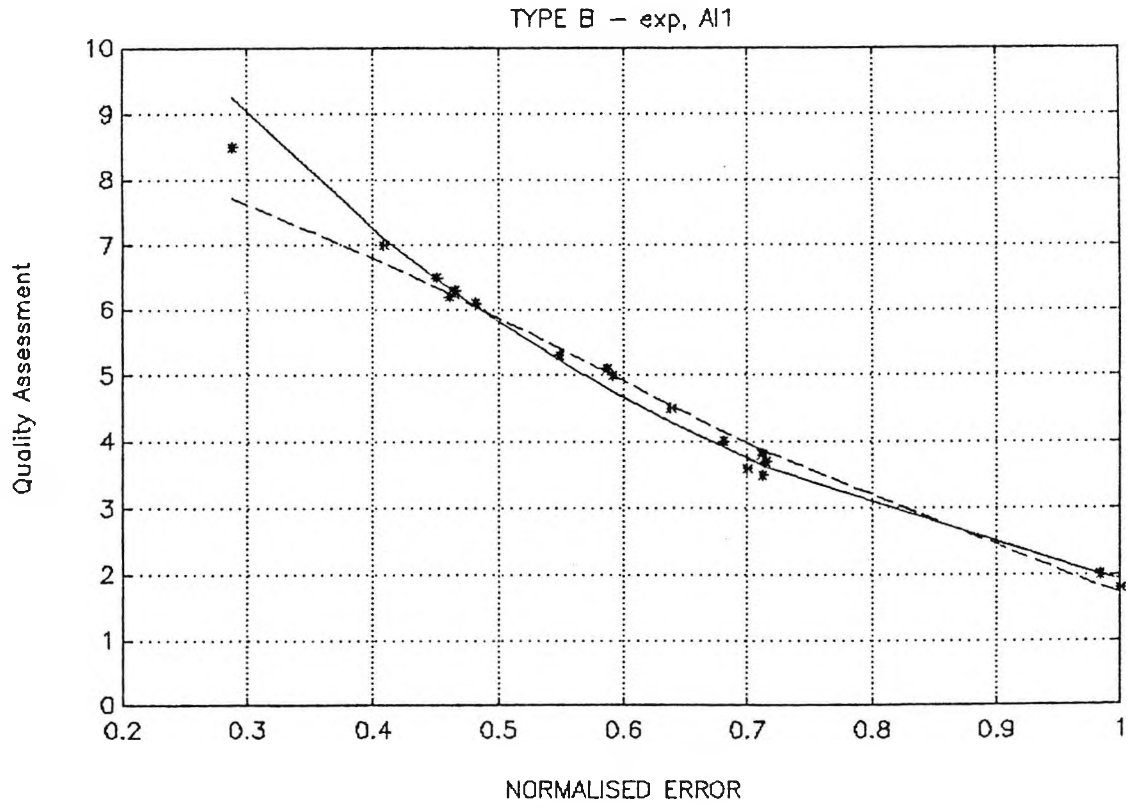


Figure 6.15: *Best exponential and gaussian fits for Type B procedure.*
(an exponential transform and AI1)

Type C Procedure

As shown in Fig. 6.6, the original image and the reconstructed image are first processed by a transform function $\mathcal{T}[\cdot]$ as in previous procedures. The difference between the two outputs of the transform blocks is taken to be the error signal which is then processed along two parallel paths. In the first path, it is weighted from point to point by a weighting factor derived from an activity index that is calculated for each point. The activity index which has been used within *Type C* procedure is A_{avr} defined in equation (6.23) with r=2. It is calculated for each pixel in the original transformed image on a 3x3 neighbourhood as proposed by Limb [Limb 1979]. It should be noted that the weighting process, carried out within *Type C* procedure, implies an inverse relation between the weighting factors and the activity indices. Local measures of image impairment are calculated for all 1 degree neighbourhoods (30x20 pixels) employing equation (6.22) and the following measures are defined as impairment predictors. The first one, denoted MAX in Table 6.3, is equal to the greatest local impairment measure calculated for the image under test. The second one, denoted D_l, is the square root of the average of the greatest two local impairment measures.

In the second path, the error signal is filtered by a two dimensional low-pass filter. The filter used is a two dimensional separable nonrecursive filter proved by Limb (Filter no. 4 in Table V in [Limb 1979]) to produce the best results in his experiments. The filtered error signal is then weighted from point to point by the same weighting factors employed in the first path, and then raised to a power of two. A global measure, denoted $LPF - D_g$ in Table 6.3, is then calculated and used as impairment predictor. If no filtering is employed in this path, the impairment measure (predictor) is denoted D_g. Simulation results, MSD values about a quadratic regression line, for different combinations of procedure parameters have been calculated and are presented in Table 6.3.

Three observations could be made from the results in Table 6.3 and the results obtained by taking different combinations of D_l and D_g as defined in equations (6.24) and (6.25). First, it is noticed that the best prediction results are obtained when an exponential transform function is being used prior to calculating the error signal. Second, it is noticed that a small amount of low-pass filtering improves the fit slightly relative to no filtering for the two nonlinear

Transform Function	IMPAIRMENT PREDICTORS			
	MAX	D_l	D_g	$LPF - D_g$
Linear	0.1005	0.1854	0.0324	0.0563
Exponential	0.1271	0.1118	0.0372	0.0236
Logarithmic	0.1225	0.1305	0.0490	0.0335

Table 6.3: MSD values for *Type C* Procedure - quadratic regression.

transform functions. The third observation seems to be the most important one. It is noted that for the three transform functions, evaluated in this research, better prediction results are obtained when global impairment measures rather than local measures are being used. These results differ from the results reported by Limb [Limb 1979], who strongly advocated local impairment measures. We suspect that the cause to these findings is the type of noise found in our test images which were generated by VQ systems. These low bit rate reconstructed images contain quantisation noise which is generally above threshold level, therefore, suprathreshold rather then threshold vision models should be used within quality prediction procedures. It should be noted that, to our best knowledge, no other experiment, similar to the one reported herein, has been carried out employing test images produced by VQ systems. Therefore, no other reference to similar results is available and more research work should be carried out in order to clear this point.

Finally, best regression results for *Type C* procedure are plotted in Fig. 6.16. The full line curve there is a minimum mean- square fit of a quadratic function to all data points obtained for the best combination of parameters, i.e., exponential transform function with a global measure of image impairment, $LPF - D_g$.

Type D procedure

Basically, this procedure is similar to *Type B* procedure with one major modification. Instead of assigning subimages to different activity classes, each associated with a weighting factor which is then used within an overall quality measure, a weighting factor is derived, for each subimage, based on equation (6.21) and the

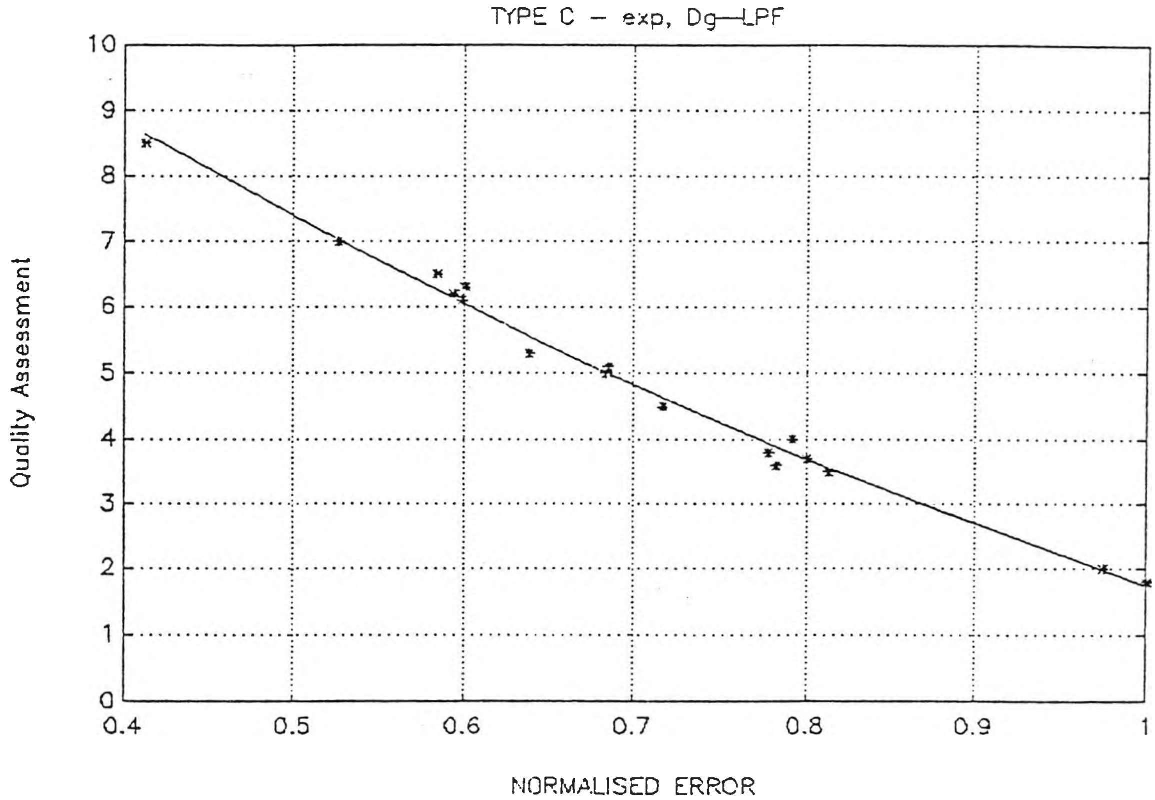


Figure 6.16: *Best fit for Type C procedure - quadratic regression. (an exponential transform and $LPF - D_g$ predictor)*

activity index calculated for that subimage. Simulation results, MSD about a quadratic regression line, for different transform functions and activity indices are given in Table 6.4.

Three observations could be made from the results in Table 6.4. First, it is noticed that in general better prediction results are obtained when $AI1$ is used as the activity index within *Type D* quality prediction procedure. This observation is in accord with similar observations made for *Type B* procedure. Second, it is noticed that the best prediction results are obtained when an exponential transform function is used prior to performing DCT. The logarithmic transform function yields the second best results for the two activity indices employed. The third observation is that the worst results are obtained when no transform function is employed, prior to performing DCT, and when $AI2$ is used as the activity index. This observation is too in accord with similar observations made for *Type B* procedure.

Transform Function	Activity Index	
	$AI1$	$AI2$
Linear	0.0242	0.0388
Exponential	0.0115	0.0122
Logarithmic	0.0134	0.0132

Table 6.4: MSD values for *Type D* Procedure - quadratic regression.

A third activity function A_{avr}, defined in equation (6.23) with r=2, has been tested in addition to the two activity functions used thus far. A_{av2} is calculated for each subimage and a weighting factor is derived based on the relation defined in equation (6.21). This activity function is in fact a measure of the local AC power of the signal, calculated for subimages of 16x16 pixels. Since the DCT is a unitary transform, the AC power of the signal, calculated in the spatial domain, equals the AC power calculated in the cosine domain, i.e., $A_{av2} = AI1$. A comparison between the MSD results for both activity functions has proved the above statement.

To conclude our findings for *Type D* procedure, the following plots are presented. The first one in Fig. 6.17, is a minimum mean-square fit of a quadratic function to all data points obtained for the best combination of parameters, i.e., exponential transform function with $AI1$ as the activity function. The second plot in Fig. 6.18, consists of two line curves: the solid line curve is the best exponential fit found for the best procedure parameters, while the dashed line curve is the best gaussian fit found under the same conditions.

6.6.5 Summary of Results and Conclusions

A summary of the best results obtained for the quality prediction procedures, evaluated in our study, is given in Table 6.5. This comparison between the best prediction results proves that *Type A* procedure yields the worst results among the procedures evaluated in this study. We suspect that the reason to this finding is the way in which spatial masking effects have been incorporated within *Type A* procedure. Nill, who has underlined the basic concept of this procedure,

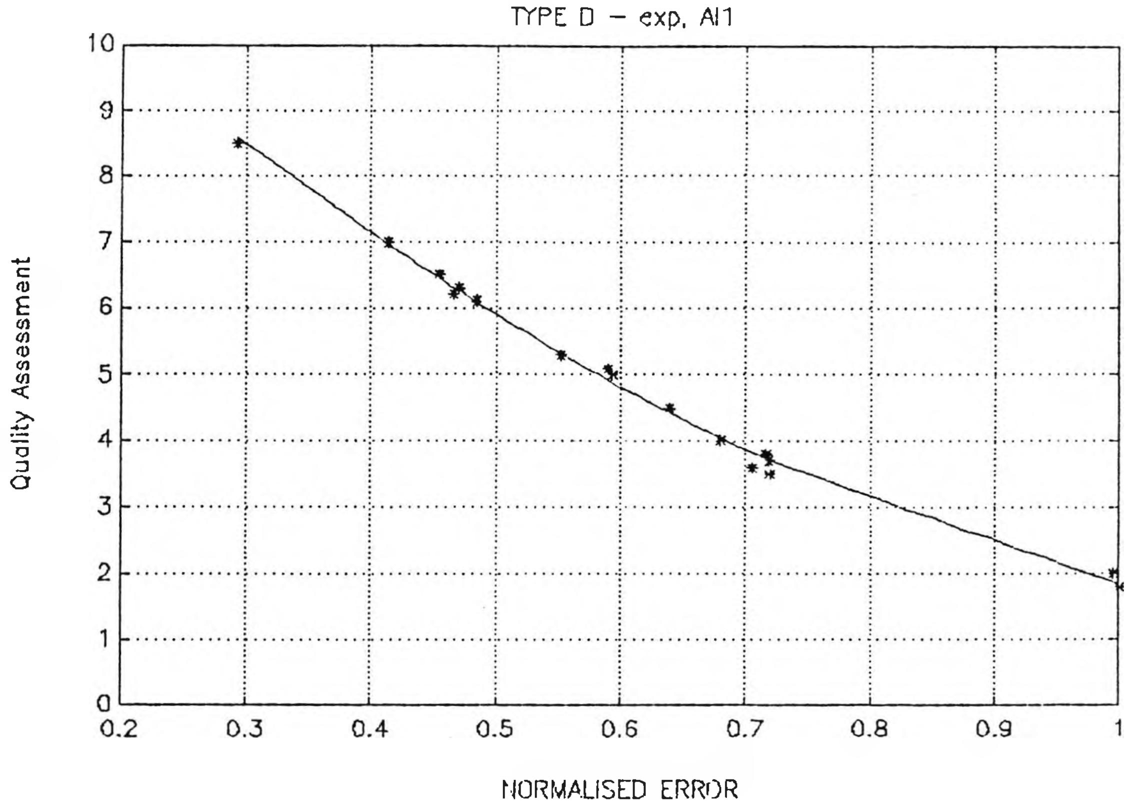


Figure 6.17: *Best fit for Type D procedure - quadratic regression. (an exponential transform and AI1)*

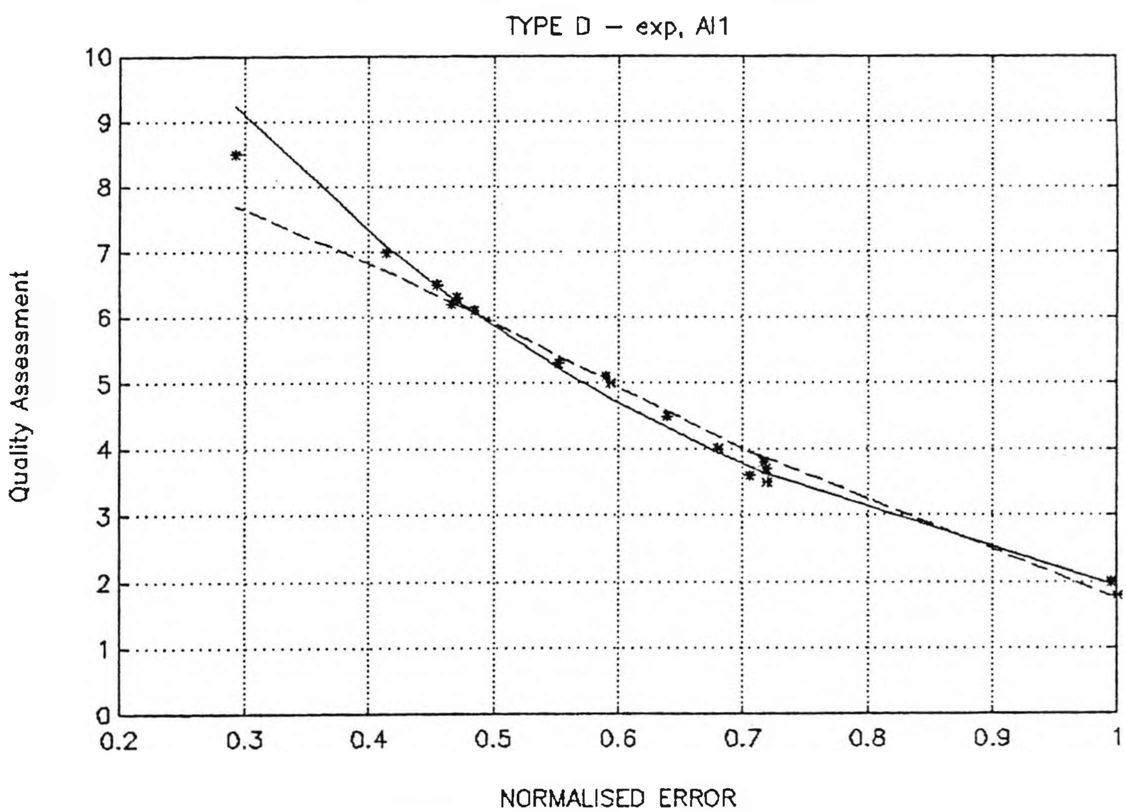


Figure 6.18: *Best exponential and gaussian fits for Type D procedure. (an exponential transform and AI1)*

argued that high structure subimages should be given more emphasis within an overall image quality measure whereas low structure subimages should be given less emphasis. We, on the other hand, have argued that a different approach should be considered for image quality prediction, i.e., giving less importance to high structure subimages due to masking effects which are operative in such subimages while giving more importance to low structure subimages. The good results, obtained for *Type B* and *Type D* procedures, prove our argument and in fact support results obtained by Limb [Limb 1979] and Lukas and Budrikis [Lukas 1982].

Procedure	Transform Function	Activity Function	Best Result	Second Best Results
Type A	logarithmic	$AI2$	0.07432	- - -
	logarithmic	$AI1$	- - -	0.0801
Type B	exponential	$AI1$	0.0123	- - -
	logarithmic	$AI1$	- - -	0.0137
Type C	exp. $LPF - D_g$	A_{av2}	0.0236	- - -
	linear D_g	A_{av2}	- - -	0.0324
Type D	exponential	$AI1$	0.0115	- - -
	exponential	$AI2$	- - -	0.0122
SNR	0.0731			

Table 6.5: A summary of best prediction results

The results in Table 6.5 strongly advocate the use of an exponential transform function prior to processing the image under test. Best prediction results are obtained by quality prediction procedures which employ this transform. These findings are in accord with results obtained by Mannos and Sakrison [Mannos 1974] who strongly advocated the use of an exponential transform function. Among the different activity functions, used to detect spatial masking effects, $AI1$ proved to be the best.

The mean square error (MSE) between the original image and the reconstructed image or functions of it, such as the signal to noise ratio (SNR), have most been used as quality measures in digital image compression research.

Therefore, a comparison between the SNR and other quality prediction measures, evaluated in this study, is in order. SNR values were calculated for every test reconstructed image and a minimum mean-square fit of a quadratic function to the set of data points, consisted of these values and subjective quality ratings, was sought. The mean square deviation (MSD) about the regression line is used as a performance index, and is given in Table 6.5. It is noted that the SNR yields the second worst quality prediction results among the best quality predictors mentioned in this table. Only one quality predictor yields results that are worse than the SNR. It is the predictor defined as *Type A* procedure (see Table 6.1) which is the original measure proposed by Nill in [Nill 1985].

A Comparison between the SNR to the predictors defined as *Type C* procedures (see Table 6.3) shows that it outperforms the local measures MAX and D_l, but is worse than the global measures D_g and $LPF - D_g$. Since the SNR measure is a global measure by it self, these results may indicate that global measures, incorporating local masking effects which are detected in small subimages of 16x16 or 30x20 pixels, are more appropriate for the type of images used in this study .

The error signal $E(x, y)$ in *Type C* procedure (see Fig. 6.6) is weighted from point to point, for each location (x, y) in the image, according to an activity index found at this location. The activity index A_{av2}, defined in equation (6.23), is calculated on a 3x3 neighbourhood as proposed by Limb [Limb 1979]. On the other hand, activity indices within *Type B* and *Type D* procedures are calculated for subimages of 16x16 pixels. Thus the results in Table 6.5 may indicate that masking effects should be evaluated in a wide local sense i.e., for neighbourhoods of about 1 degree of subtended arc, also found to be the size of the human fovea [Limb 1979].

In addition to the quadratic function used for regression, two other functions have been tested i.e., an exponential function and a gaussian function. Mean-square deviation (MSD) values about quadratic regression curves found for the best prediction procedures, *Type B* and *Type D* procedures, are presented in Table 6.6. These results are compared with results obtained for exponential and gaussian regression functions. It could be noted that the quadratic regression function yields the best fit results and thus, seems to be appropriate for the type of test images used in this study.

Procedure Parameters	Quadratic	Gaussian	Exponential
Type B, exponential, $AI1$	0.0123	0.0654	0.0520
Type D, exponential, $AI1$	0.0115	0.0680	0.0482

Table 6.6: A comparison of three regression functions - best predictors.

Finally, it should be noted that the results obtained for *Type D* procedure are slightly better than the results obtained for *Type B* procedure. Basically these procedures are very similar differing in one aspect ; the way in which a weighting factor is associated with each subimage. Activity classes, each associated with a different weighting factor, are being used within *Type B* procedure whereas, in *Type D* procedure, equation (6.21) is being used to derive weighting factors for each subimage, based on an activity index found for that subimage. Due to the straightforward algorithm, we strongly recommend using *Type D* procedure for image quality prediction.

Various models of the human visual system have been proposed by vision researchers. We have tested a small number of models, some of which have been proposed in the literature and some of which have been developed by us. The results obtained by this study indicate that good fits could be found between quality measures and subjective quality ratings. More work should be carried out, employing a larger number of test images and observers, in order to further establish the results presented in this report. New models of human visual system like the multi-channels model described by Sakrison [Sakrison 1977] or the use of two dimensional (2-D) Gabor elementary functions as models for *simple cell* receptive field profiles [Daugman 1985], should be investigated and applied within quality prediction procedures. We believe that at the end of the day an objective quality measure could be found and made standard in the field of vision research.

Chapter 7

Summary

A good explanation;
Never explains everything.

7.1 Conclusions

The main purpose of this research has been the development of a new VQ-based coding system that integrates valuable knowledge about the human visual system (HVS) with emerging VQ techniques. It has been postulated that if the coding scheme could be matched to the HVS and could attempt to imitate its functions, at least for the known part of it, high compression ratios along with good quality of reconstructed images could be achieved. The new coding system was developed in a step by step manner employing basic VQ systems as valuable "building blocks".

First, the basic notion of vector quantisation along with codebook design methods have been studied in chapter 2. Various drawbacks of full search VQ have been discussed, particularly the computational complexity caused by the search problem, and the degradation in the perceived quality of coded images. Gain-shape VQ (GSVQ) and classified VQ (CVQ) have been proposed in the VQ literature to overcome these drawbacks. GSVQ is capable of being used at higher dimensions and rates than full search VQ because of its reduced complexity, both computational and storage (having fewer codevectors to be stored and searched), thus it has widely been employed for voice, waveform, and image coding. CVQ, on the other hand, has been proposed as a coding method that is particularly suited for preserving perceptual features while retaining simple VQ distortion measures.

We have developed a new algorithm for CVQ codebook design as an alternative to the empirical method proposed in [Ramamurthi 1986]. The new algorithm provides a simple and systematic method for codebook design and reduces considerably the total number of required mathematical operations during codebook design in comparison with the brute-force method, described in [Ramamurthi 1986]. We have named this new algorithm *Classified Nearest Neighbour Clustering* (CNNC). In addition, a fast search algorithm has been developed in Appendix C to reduce computational efforts during CVQ codebook design.

Coding results for CVQ at low bit rates (less than 0.562 bpp) have shown that the *blocking effect* and the *staircase effect* are still noticeable in the coded images. Similar observations have been made when a GSVQ was used at low bit rates. Nevertheless, the CVQ concept and the notion of GSVQ have been consistenly applied in our work toward the development of a new VQ-based coding system. It seemed that the separation of *gain* coding from *shape* coding, which characterizes GSVQ, combined with the notion of perceptually-based coding, as realized in CVQ, can pave the way to the development of a better coding system.

A review of the human visual system has been presented in chapter 3. Particularly, properties of the HVS like frequency sensitivity, the masking effect, and orientation sensitivity, which can be used in image processing and coding, have been described. Armed with this knowledge about the HVS and the basic VQ systems, we have developed a new image coding system [Chapter 4]. We have named this system *Directional Classified Gain-Shape Vector Quantisation* (DCGSVQ). It combines vector quantisation with transform coding techniques and exploits various properties of the HVS to produce reconstructed images with good subjective quality at low bit rates.

A new algorithm for designing the various DCGSVQ codebooks has been proposed in chapter 4. It is based on the CNNC algorithm but employs a new optimisation criterion which is more suitable for shape codebook design. We have called it the *modified CNNC algorithm*. The new algorithm designs the various shape codebooks simultaneously giving the designer full freedom to assign more importance to certain classes of vectors or to certain training vectors.

A new vector configuration strategy for defining AC vectors in the cosine domain has been proposed in chapter 4. We have named the vectors, so configured, *directional vectors*. Directional vectors are obtained by grouping the AC coefficients into two vectors according to their *direction* in the cosine domain, i.e., adaptively to the direction of the spatial activity in the image block. To test the proposed strategy, the test images were coded by two systems. The first system was the proposed DCGSVQ operating with directional vectors. The second system was the proposed DCGSVQ operating with *variance vectors*, i.e., AC vectors that were configured by grouping AC coefficients of similar variances into the same vector (the traditional way). Coding results have shown that an improvement of more than 0.9 dB has been achieved by just grouping the AC coefficients according to the new proposed vector configuration strategy. Subjective tests of reconstructed images have confirmed this finding.

It seems that the reason for this improvement is mainly due to the fact that the two largest AC coefficients, $F(0,1)$ and $F(1,0)$, are assigned to different vectors according to the proposed directional vector notion. By having them apart, the shape vectors become less "noisy" and the effective dynamic range of the expected gain values is reduced. Consequently, the coding procedure becomes more effective and better SNR results and perceived quality are achievable.

The new notion of directional vectors, and the basic approach of using a classified GSVQ to encode them, have paved the way to a simple technique of feature enhancement which can be applied during the decoding process to further improve the reconstructed images. It has been shown that by multiplying the decoded *gain* values by an *enhancement factor*, the dynamic range of decoded vectors can be "stretched", and "crisper" reconstructed images can be obtained. Feature enhancement may be applied to vectors derived from image blocks that contain diagonal edges or complex features, and only in the direction across a vertical or a horizontal edge. However, it should be stated that the proposed method enhances not only the desired features in the reconstructed image but also the quantisation noise. Therefore, the enhancement procedure should be applied with appropriate care.

Usually, coding systems that operate in the cosine domain have quantised separately the DC and AC coefficients. Due to their influence on the perceived quality of coded images, the DC coefficients (representing the mean luminance of the blocks) have been carefully quantised using a scalar quantiser. However, by scalar quantising those coefficients, the spatial correlation which exists between neighbouring DC coefficients is not being exploited. Therefore, a more efficient way of encoding the DC coefficient has been proposed in chapter 4. Four neighbouring DC coefficients were grouped into one vector and encoded by a GSVQ. A GSVQ, rather than a VQ, was employed because we assumed that better coding results could be achieved by dealing separately with DC shape vectors and DC gain values. The modified CNNC algorithm was used to design the DC shape codebook, and proved to be an effective alternative to traditional GSVQ design algorithms.

In general, this notion works satisfactorily when the four neighbouring image blocks are *monotone* blocks; however, if one or more blocks are *edge* blocks, the *blocking effect* becomes apparent. Therefore, special care has to be taken when designing GSVQ codebooks for the DC coefficients. In particular, the training set of DC shape vectors, used for codebook design, should contain vectors of various types, especially vectors derived from edge blocks.

In the proposed DCGSVQ, the AC coefficients in the cosine domain are weighted according to a modulation transfer function (MTF) that represents the filtering characteristics of the HVS. We have named this process *HVS filtering*. It has been shown that better coding results are achieved when a low-pass filter (LPF), rather than a band-pass filter (BPF) is employed for weighting the AC coefficients. Subjective tests of reconstructed images have confirmed this finding. We conclude that the low-frequency coefficients are better preserved when a LPF, rather than a BPF, is being employed thus reducing substantially the blocking effect.

Inverse HVS filtering is carried out at the decoder to compensate for the HVS filtering which took place at the encoder. In general, HVS filtering reduces the influence of high-frequency "noisy" coefficients on the selection of a proper shape vector by the encoder. In other words, the low-frequency coefficients dominate the encoding process allowing harsh quantisation of the high-frequency coefficients. However, we have found out that when inverse HVS filtering is carried out at the decoder, the quantisation errors, which are present at high-frequency coefficients, are amplified causing unnecessary degradation of the reconstructed images. As anticipated, subjective tests of reconstructed images, produced without inverse HVS filtering, have shown some loss of fine details; however, some of the high-frequency quantisation noise has been reduced too so that, in general, the perceived quality of the test images has been improved. Therefore, it is recommended to carry out HVS filtering at the encoder, to ensure better shape preservation during encoding, but to consider omitting inverse HVS filtering at the decoder when test images are concerned.

There is sufficient direct physiological evidence to support the hypothesis that the relationship between the light intensity input to the visual receptors and the neural output level is approximately logarithmic. Therefore, various researchers have suggested a point nonlinear transformation as the first stage in any digital image processing system. We have tested this notion by applying such a nonlinear transformation, referred to as *HVS transform*, to the intensity values prior to encoding the images with the DCGSVQ encoder.

Based on SNR figures and on subjective tests of the coded images, it is quite evident that better coding results are obtained without applying the HVS transform. It seems that the nonlinear transformation, normally applied to the output voltage from the image sensor in TV cameras, is responsible for this finding. This transformation (known as *gamma precorrection*) is applied in order to correct the nonlinear characteristics (described by the *gamma factor*) of cathode-ray tubes (CRT) used in TV displays. Therefore, if one wishes to apply the mentioned HVS transform prior to encoding an image, inverse gamma precorrection transformation should be applied first.

Coding results, obtained by the proposed DCGSVQ, were compared with coding results obtained by a full search VQ and a CVQ. The images, produced by the DCGSVQ, look less blocky and have a quite natural appearance whereas images, produced by a full search VQ or a CVQ, look blocky and the edges appear jagged. SNR results for the coded test images at various bit rates have proved that the proposed DCGSVQ significantly outperforms the other systems. The coding performance of the DCGSVQ system was also compared with the coding performance of a transform coding classified VQ (TC-CVQ) system. The TC-CVQ operates in the cosine domain employing a CVQ to encode AC vectors which belong to various classes of activity. The DCGSVQ system has been shown to significantly outperform this system too.

To further improve the perceived quality of coded images, a new postprocessing algorithm that can be applied at the decoder without increasing the bit rate has been developed in chapter 5. The proposed algorithm is based on various characteristics of the signal spectrum and the noise spectrum, and exploits various properties of the HVS. The new algorithm is a general-purpose algorithm that can be applied to block-coded images produced by various systems like VQ, transform coding (TC), and block truncation coding (BTC).

The proposed filtering algorithm is both space-variant and nonlinear consisting of three steps: (*i*) grid noise removal, (*ii*) staircase noise removal, and (*iii*) edge enhancement. The algorithm is modular and can be applied in an adaptive way depending on the quality of the block-coded image. Each step stands alone and can be applied only if necessary. For example, we have shown that all three steps should be applied for postprocessing of poorly coded images while applying just one step for high quality coded images. This feature offers great flexibility, which can not be found in other filtering methods proposed in the literature, and ensures good filtering results. The new filtering algorithm has been tested on VQ-coded images and proved to improve dramatically the perceived quality of such images.

Various image compression systems have been implemented in the course of our research, and we have often faced the problem of deciding which is best. Identification of useful fidelity criteria for image compression system design and analysis has been a persistent difficulty for researchers. The image quality measure, actually a measure of quality degradation, that has most been used in digital image compression research is the mean square error (MSE) between the original image and the reconstructed image. However, it has often been empirically determined that the MSE or functions of it such as the signal to noise ratio (SNR), do not correlate well with subjective (human) quality assessments.

The MSE criterion does not adequately track the types of degradation caused by digital image compression systems and it does not adequately "mimic" what the HVS does in assessing image quality. There is a need, therefore, for accurate measures of subjective impairment which can be used to predict image quality.

In chapter 6, quality predictors that incorporate simplified models of the HVS have been proposed and tested on a large set of VQ-coded images. Quality predictors in the form of some subjectively weighted error measures were sought such that a smooth functional relationship exists between them and quality ratings made by human viewers. Two such predictors have been shown to be better suited than the commonly used MSE measure. However, due to the limited resources available to us (in particular, the small number of viewers who took part in the subjective tests), we recommend that more work should be carried out, employing a larger number of test images and viewers, in order to further establish the results presented in this chapter.

7.2 Future Work

7.2.1 Perceptually-based Codebook Design

Close inspections of CVQ-coded images at rates lower than 0.562 bpp have shown that the staircase effect and the blocking effect are still noticeable. It seems that some improvement may be achieved if the perceptual importance of the various classes, defined for the CVQ system, could be taken into consideration. For example, by increasing artificially the importance of vectors that belong to certain classes during codebook design, the number of codevectors found for each class at the conclusion of the CNNC algorithm can be influenced without changing the clustering process or the encoding process.

This approach has an intuitive appeal when applied to the problem of edge degradation, which was the major reason for developing CVQ systems, since it permits us to control the relative sub-codebook sizes and thus ensures edge integrity at the expense of SNR results. This approach is also applicable to designing DCGSVQ codebooks, as shown in chapter 4 where we have named it *perceptually-based codebook design.* The perceptual importance of the various classes of vectors, defined for CVQ and DCGSVQ systems, should be carefully studied and a compromise between SNR results and subjective quality of coded images should be sought to further improve the coding performance of these systems.

7.2.2 HVS Models

Incorporating HVS models in coding systems has recently become a common practice. While being helpful in some cases, applying this notion in a brute-force manner could be destructive in other cases. As explained in chapter 2, the visual system is most often described and investigated at *threshold* behavior. Unfortunately, it turns out that the visual system is far from linear, and properties of threshold vision cannot generally be applied to the *suprathreshold* (visible impairment) case. The main part of research on the visual system is thus primarily applicable to very good quality image coding, where the main problem concerns coding with imperceptible artifacts. At low rates, however, visible degradations will always be present, and care must be taken when employing simple HVS models within the coding system. Future work should tackle the visual modelling problem further in order to find more appropriate models for suprathreshold vision.

7.2.3 Entropy Coding

In general, coding N vectors with $\log_2 N$ bits each is actually the maximum rate needed for coding. The minimum average achievable rate to encode the codevectors $\{\bar{y}_i\}$ is given by the *entropy* of $\{\bar{y}_i\}$ [Gallager 1968]. The entropy $\mathcal{H}(\bar{y})$ of the discrete-amplitude variable $\bar{y} = \{\bar{y}_i\ ;\ i = 1, 2, \ldots, N\}$ is defined by

$$\mathcal{H}(\bar{y}) \;=\; -\sum_{i=1}^{N} \Pr(\bar{y}_i) \log_2 \Pr(\bar{y}_i) \tag{7.1}$$

where $\Pr(\bar{y}_i)$ is the discrete probability of $\bar{y}_i$. If each vector $\bar{y}_i$ is coded using $B_i = -\log_2 \Pr(\bar{y}_i)$ bits so that vectors with different probabilities will have different codeword lengths, then the resulting code will be a *variable-length* code with an average rate equal to the entropy $\mathcal{H}(\bar{y})$. This type of coding is known as *entropy coding.*

Entropy coding is one form of ***noiseless coding*** in that the coding does not introduce any additional noise or distortion beyond that introduced by the quantisation process. It merely takes advantage of the probability distribution of the codewords to minimise the bit rate. A well-known straightforward method for implementing entropy coding is the Huffman code [Huffman 1952]. We believe that a 20% reduction in bit rate, without any degradation in image quality, can be achieved by designing an appropriate Huffman code for the DCGSVQ codevectors. To design such a code, we suggest to use a large set of source images in conjunction with the CNNC algorithm. Based on a statistical study of the codevectors, employed for encoding the source images, a Huffman code can be designed and used to further improve the coding performance of DCGSVQ.

7.2.4 Lapped Orthogonal Transform

A new class of transforms for block coding has been recently introduced in [Cassereau 1985] and [Malvar 1989]. These new transforms, collectively referred to as the *Lapped Orthogonal Transform* (LOT), are characterized by the fact that the basis functions overlap adjacent blocks; however, the number of transform coefficients is kept equal to the original block size so that no data overhead is incurred. Consequently, fast LOT procedures, like the one introduced in [Malvar 1989], allow the implementation of block coding systems at low bit rates with much less noticeable blocking effects than traditional DCT-based systems.

The energy compaction performance of several lapped orthogonal transforms in comparison with the DCT has been studied in [Akansu 1992]. It has been shown that the performance of the LOT is superior to that of the DCT especially for high correlation sources. In addition, visual tests of coded images have confirmed that the blocking effect is indeed less visible when the LOT is employed. Therefore, we suggest that the applicability of the LOT to the proposed DCGSVQ should be studied. In particular, the notion of directional vectors in the LOT domain should be understood before attempting to replace the DCT by the LOT. We anticipate that better coding results could be achieved if the

LOT would be employed within the DCGSVQ.

7.2.5 Interframe Coding

Video compression techniques can be divided into two broad classes : *intraframe* and *interframe* coding. Intraframe coding is used to remove redundancy in single-frame images exploiting the strong correlation between neighbouring pixels. Interframe coding takes advantage of the strong correlation in video frame sequences to reduce the picture redundancy between frames. In general, both coding techniques should be used together to achieve the highest compression ratio.

Motion in video conferencing scenes is usually low. In such a case, interframe coding techniques can reduce the information redundancy in video sequences. Combined with intraframe coding, a high compression ratio can be achieved by incorporating motion detection techniques and then sending only the motion information through the communication channel. According to this technique, the picture is segmented into two parts : the *stationary* and the *moving* areas. Only the information about the moving area is transmitted in the form of a difference pattern between the current and previous frame images. It was suggested in the early '70s that this scheme can be improved by estimating the displacement of objects in the current frame based upon the previous frame image and using the estimated displacement for more efficient predictive coding [Rocca 1972]. This type of scheme is called *motion compensated coding* [Netravali 1979].

Several methods of estimating the object's displacement in a video sequence have been proposed. Generally, they can be classified into two types : *pixel-recursive algorithms* (PRA) [Rocca 1972] and *block-matching algorithms* (BMA) [Kappagantula 1983]. PRA depends on the temporal and spatial differential signals between the pixel intensity in the previous frame and the pixel intensity in the current frame. BMA matches blocks in the previous frame with corresponding blocks in the current frame. Usually, PRA can track complex motion more accurately than BMA. This is because individual pixels in a block may undergo different translating movement. However, the hardware implementation for BMA is much simpler than PRA. We believe that the DCGSVQ technique, proposed in this thesis, can be successfully combined with interframe coding techniques that employ motion detection and compensation techniques to provide a good coder for teleconferencing applications.

7.2.6 Image Compression Standards

Lack of open standards for video compression techniques could slow the growth of this technology and its applications. Therefore, during the last few years progress has been made on digital video standards. The three main ones concern still-picture compression, video teleconferencing, and full-motion compression on digital storage media. They have been proposed by the Joint Photographic Expert Group (JPEG), the International Telegraph and Telephone Consulative Committee (CCITT), and the Moving Picture Expert Group (MPEG), respectively.

The JPEG standard is an algorithm for coding still images developed under the patronage of the International Organisation for Standardisation (ISO). It is a general-purpose compression standard designed to meet the needs of continuous tone, still image applications. It is applicable to such uses as photovideotex, desktop publishing, the graphic arts, color facsimile, newspaper wirephoto transmission, and medical imaging.

CCITT's Recommendation H.261 (also called $p \times 64$) specifies a method of communication for visual telephony. It is a standard for covering the entire channel capacity of the integrated services digital network (ISDN). The $p \times 64$ designation refers to $p \times 64$ kb/Sec, where p can have any value from 1 to 30. The standard is intended for use in videophone and videoconferencing.

MPEG, the third digital video standard, can be applied to such storage media as compact disc ROM (CD ROM), digital audio tape, Winchester disk, and writable optical discs and on such communication channels as ISDN and local area networks (LANs). MPEG addresses the compression of video signals at about 1.5 Mb/Sec and of digital audio signals at the rates of 64, 128, and 192 kb/Sec. It also deals with the synchronisation and multiplexing of multiple compressed audio and video bit streams. Another new phase of MPEG committee activities is addressing the need for a video compression algorithm (MPEG-2) for higher-resolution signals at bit rates up to 10 Mb/Sec.

The standards mentioned above fall under the heading of transform-based image coding. Color images, represented in the red-green-blue (RGB) system used in the computer industry, or in the luminance-chrominance (YIQ) system used in the television industry, can be coded by applying separately the standard algorithm to each of the color components. In the JPEG encoder, for example, each component of the source image is divided into non-overlapping blocks of 8x8 pixels and then transformed using the two-dimensional DCT. In the cosine domain, the AC coefficients are quantised by a scalar quantiser with a variable quantisation step size. The quantisation step size varies with frequency reflecting the fact the high-frequency coefficients are subjectively less important than the low-frequency ones and may, therefore, be quantised more coarsely (a similar idea has been implemented in the DCGSVQ system proposed by us).

Following quantisation, the AC coefficients are ordered into a one-dimensional array and after run-length coding they are losslessly encoded using Huffman coding. The DC coefficients are differentially encoded so that the DC coefficient of the previous block is used to predict the DC coefficient of the current block and the difference between these two DC terms is encoded. Huffman coding is also applied to the coded difference values.

The H.261 and the MPEG coders combine transform coding with predictive coding, in which a block in the current frame is predicted from a block in the previous frame. These standards have also an optional specification for motion compensation to increase the predictive coder's efficiency in tracking the interframe motion. These coders employ a scalar quantiser to encode the DCT coefficients due to the simplicity of this method. We believe that, in general, better encoding performance can be achieved if a VQ would be used for this purpose combined with Huffman coding of its codevectors, as proposed in a previous subsection. In particular, we believe that the good performance of the DCGSVQ system along with the simple codebook design method proposed in this thesis, may result in an alternative coding method to the JPEG standard.

The activities described above certainly demonstrate the widespread recognition of the need for image compression in many diverse applications where just few years ago compression was not considered viable. We anticipate a proliferation of image compression in the near future, and certainly hope to take part in developing better coding systems.

Appendix A

A Fast Search Algorithm

In this Appendix we address a common problem encountered in VQ systems called the *search problem*, and explain a simple algorithm which helps overcoming this problem. First, the basic search problem is stated.

Given a query vector $\bar{x} \in R^k$ and a set of codevectors $C = \{\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_N\} \in R^k$, find which codevector is closest to $\bar{x}$ in the sense of minimum squared error.

A simple, but efficient, nearest neighbour search algorithm to reduce computational efforts and solve the search problem was proposed in [Soleymani 1987]. According to this algorithm, the reduction in computational efforts and complexity is achieved by performing a test *prior* to calculating the distortion for a given codevector, thereby avoiding the distortion calculation for those codevectors which fail this test. Before describing the mentioned algorithm, the following variables are defined:

- 'R' is the square root of the minimum distortion, found thus far.
- 'I' stores the index of the closest codevector, found thus far.
- '$\bar{Y}$' stores the closest codevector to $\bar{x}$, the query vector, i.e.,

  $$\bar{Y} = \min_{\bar{y}_i \in C}{}^{-1}[d(\bar{x}, \bar{y}_i)]$$

  where the inverse minimum notation means that $\bar{Y}$ equals the codevector $\bar{y}_i$ which yields the indicated minimum distortion.

The algorithm itself can be summarised as follows:

1. Starting with the first codevector $\bar{y}_1$ calculate the distortion

$$d(\bar{x},\bar{Y}) = \| \bar{x} - \bar{Y} \|^2 = \sum_{q=1}^{k} (x_q - Y_q)^2$$

   where $\bar{Y} = \bar{y}_1$ and $I = 1$.
   Defining $R = [d(\bar{x},\bar{Y})]^{1/2}$, a hypersphere with radius R is drawn around $\bar{x}$, inscribed by a hypercube. Any codevector $\bar{y}_i$ $(i \neq 1)$ that is closer to $\bar{x}$ than $\bar{y}_1$ should lie inside the hypersphere and thus inside the hypercube inscribing it. A search for such a codevector is carried out in the next step by first checking whether it lies inside the hypercube and if it does, only then checking whether it lies inside the hypersphere.

2. For each codevector $\bar{y}_i$ $(i = 2, 3, \ldots, N)$ the following comparisons are carried out:

   a) If $R \leq | x_q - y_{iq} |$ for a dimension q $(q = 1, 2, \ldots, k)$ then $\bar{y}_i$ is outside the hypercube, inscribing the hypersphere, thus it is rejected.

   b) If $R > | x_q - y_{iq} |$ for *each* dimension q $(q = 1, 2, \ldots, k)$ then $\bar{y}_i$ is inside the hypercube but not necessarily inside the hypersphere itself. To find out whether $\bar{y}_i$ is inside the hypersphere, $d_i(\bar{x},\bar{y}_i)$ should be calculated by

$$d_i(\bar{x},\bar{y}_i) = \sum_{q=1}^{k} (x_q - y_{iq})^2 = R_i^2 ,$$

   and the following comparisons should be carried out:
   If $R > R_i$ then $\bar{y}_i$ is inside the hypersphere (i.e., $\bar{y}_i$ is closer to $\bar{x}$ than the codevector stored in $\bar{Y}$), and the following variables are updated: $R = R_i$; $\bar{Y} = \bar{y}_i$; $I = i$.
   If $R \leq R_i$ then $\bar{y}_i$ does not lie inside the hypersphere, thus it is rejected.

Step 2 is carried out until all the codevectors are checked. At the conclusion of the algorithm, $\bar{Y}$ stores the closest codevector, I defines its index and the minimum distortion is given by R^2.

Appendix B

The Merging Error

In this Appendix we derive the equations used to calculate the merging error incurred when two clusters of vectors are merged into one cluster, having a weighted average vector as its centroid. The equation used to calculate the new centroid of the unified cluster is also determined.

Let S_j and S_l denote two non-empty and disjoint clusters consisting of m_j and m_l vectors respectively ($S_j \cap S_l = \emptyset$, the empty set). Let $S_p = S_j \cup S_l$, i.e., S_p is produced by uniting the clusters S_j and S_l. For k-dimensional vectors $\bar{x} \in R^k$ and the squared error distortion measure, the *centroid* is defined by

$$\tilde{X}_q = \frac{1}{m_q} \sum_{\bar{x} \in S_q} \bar{x} \qquad \text{for} \quad q = j, l, p \,, \tag{B.1}$$

and the sum of the squared distances of the vectors in the cluster from its centroid by

$$D_q = \sum_{\bar{x} \in S_q} \| \bar{x} - \tilde{X}_q \|^2 \qquad \text{for} \quad q = j, l, p \,. \tag{B.2}$$

By squaring, D_q can be written as

$$\begin{aligned} D_q &= \sum_{\bar{x} \in S_q} \left[\| \bar{x} \|^2 + \| \tilde{X}_q \|^2 - 2\bar{x}\tilde{X}_q^T \right] = \\ &= \sum_{\bar{x} \in S_q} \| \bar{x} \|^2 + \sum_{\bar{x} \in S_q} \| \tilde{X}_q \|^2 - 2 \sum_{\bar{x} \in S_q} \bar{x}\tilde{X}_q^T = \\ &= \sum_{\bar{x} \in S_q} \| \bar{x} \|^2 + m_q \| \tilde{X}_q \|^2 - 2\tilde{X}_q^T \sum_{\bar{x} \in S_q} \bar{x} \,. \end{aligned} \tag{B.3}$$

Combining Eq. (B.1) and Eq. (B.3) we get

$$D_q = \sum_{\bar{x} \in S_q} \| \bar{x} \|^2 - m_q \| \tilde{X}_q \|^2 \quad \text{for} \quad q = j, l, p \; . \tag{B.4}$$

The centroid $\tilde{X}_p$ and the sum of squared distances D_p can be represented as functions of $\tilde{X}_j$, $\tilde{X}_l$, D_j and D_l as follows

$$\tilde{X}_p = \frac{1}{m_p} \sum_{\bar{x} \in S_p} \bar{x} = \frac{1}{m_p} \left[\sum_{\bar{x} \in S_j} \bar{x} + \sum_{\bar{x} \in S_l} \bar{x} \right] . \tag{B.5}$$

Using Eq. (B.1) and the equality $m_p = m_j + m_l$ we get

$$\tilde{X}_p = \frac{m_j \tilde{X}_j + m_l \tilde{X}_l}{m_j + m_l} . \tag{B.6}$$

Substituting Eq. (B.6) in Eq. (B.4) with $q = p$ yields

$$D_p = \sum_{\bar{x} \in S_j} \| \bar{x} \|^2 + \sum_{\bar{x} \in S_l} \| \bar{x} \|^2 - m_p \| \frac{m_j \tilde{X}_j + m_l \tilde{X}_l}{m_p} \|^2 . \tag{B.7}$$

Using Eq. (B.4) again for $q = j, l$ yields

$$D_p = D_j + m_j \| \tilde{X}_j \|^2 + D_l + m_l \| \tilde{X}_l \|^2 - \frac{1}{m_p} \left[\| m_j \tilde{X}_j + m_l \tilde{X}_l \|^2 \right] . \tag{B.8}$$

Finally, using the equality

$$\| m_j \tilde{X}_j + m_l \tilde{X}_l \|^2 = m_j^2 \| \tilde{X}_j \|^2 + m_l^2 \| \tilde{X}_l \|^2 + 2 m_j m_l \tilde{X}_j^T \tilde{X}_l$$

it can be shown that

$$D_p = D_j + D_l + \frac{m_j m_l}{m_j + m_l} \| \tilde{X}_j - \tilde{X}_l \|^2 . \tag{B.9}$$

Thus the *merging error* incurred when merging the two clusters S_j and S_l to produce S_p, a new cluster having $\tilde{X}_p$ as its centroid, is defined by

$$ME_{jl} = D_p - (D_j + D_l) = \frac{m_j m_l}{m_j + m_l} \| \tilde{X}_j - \tilde{X}_l \|^2 . \tag{B.10}$$

Appendix C

A New Fast Search Algorithm

In this Appendix, the *search problem* encountered within the CNNC algorithm is addressed, and a new algorithm is proposed to reduce the computational efforts. The search problem could be stated as follows:

Given a set of clusters $\{S_1, S_2, \ldots, S_L\}$, consisting of vectors $\bar{x} \in R^k$, and a set of centroids $\{\tilde{X}_i \in R^k \ ; \ i = 1, 2, \ldots, L\}$ representing these clusters. Find two clusters that incur minimum merging error when merged to produce one cluster.

Our new algorithm is based on a simpler algorithm proposed for solving the search problem in VQ systems (see Appendix A for discussion). The rationale behind the new algorithm is described first.

Starting with two clusters S_i and S_j, the *merging error* is defined as the additional distortion introduced by merging the two clusters and representing the two clusters worth of data with a single centroid. It is defined in Appendix B by

$$ME_{ij} = \frac{m_i m_j}{m_i + m_j} \parallel \tilde{X}_i - \tilde{X}_j \parallel^2 = T_{ij} \sum_{q=1}^{k} (X_{iq} - X_{jq})^2 \qquad \text{(C.1)}$$

$$T_{ij} = \frac{m_i m_j}{m_i + m_j} \qquad \text{(C.2)}$$

where m_i and m_j are the numbers of vectors in clusters S_i and S_j, respectively, and $\tilde{X}_i$ and $\tilde{X}_j$ are the centroids of these clusters. If instead of merging S_i with S_j cluster S_a $(a \leq L, \ a \neq i, j)$ is merged with cluster S_i, the merging error incurred is given by

$$ME_{ia} = T_{ia}\sum_{q=1}^{k}(X_{iq} - X_{aq})^2 \qquad \text{(C.3)}$$

where T_{ia} is defined as in Eq. (C.2) replacing m_j by m_a. The question now is whether $ME_{ij} > ME_{ia}$. Defining

$$R_{ia}^2 = \sum_{q=1}^{k}(X_{iq} - X_{aq})^2 \ , \qquad \text{(C.4)}$$

and using Eq. (C.3) the question is whether $ME_{ij} > T_{ia}R_{ia}^2$, or equivalently whether $R > R_{ia}$, where

$$R = \left[\frac{ME_{ij}}{T_{ia}}\right]^{1/2} \ . \qquad \text{(C.5)}$$

Thus, summarising the above discussion, cluster S_a is a better candidate than S_j for merging with cluster S_i if and only if the inequality of $R > R_{ia}$ holds.

It should be noted that R_{ia} is the Euclidean distance between the centroids $\bar{X}_i$ and $\bar{X}_a$. Calculating this distance for all possible pairs of clusters is time consuming and computationally demanding. In the proposed algorithm, the reduction in computational efforts and complexity is achieved by performing a test prior to calculating the merging error (or equivalently R_{ia}) for a given pair of clusters, thereby avoiding error calculation for those clusters which fail this test. The new algorithm can be summarised as follows:

1. Starting with a pair of clusters S_i and S_j, ME_{ij} is first calculated employing Eq. (C.1). The following variables are initialised:

   $$ME = ME_{ij} \ ; \ W = i \ ; \ \text{and} \ V = j.$$

2. Taking any centroid $\bar{X}_a$ ($a = 1, 2, \ldots, L$; $a \neq i, j$), T_{ia} and R are calculated employing equations (C.2) and (C.5), respectively. A hypersphere with radius R is then drawn around $\bar{X}_i$, inscribed by a hypercube. If centroid $\bar{X}_a$ is closer to $\bar{X}_i$ than $\bar{X}_j$, it should lie inside the hypersphere and thus inside the hypercube inscribing it. Therefore, testing consists of first checking whether $\bar{X}_a$ lies inside the hypercube and if it does, only then checking whether $\bar{X}_a$ lies inside the hypersphere. Thus, for centroid $\bar{X}_a$ (representing cluster S_a) the following comparisons should be carried out:

   a) If $R \leq \mid X_{iq} - X_{aq} \mid$ for a dimension q ($q = 1, 2, \ldots, k$) then $\bar{X}_a$ is outside the hypercube, inscribing the hypersphere, and thus it is rejected.

b) If $R > \mid X_{iq} - X_{aq} \mid$ for *each* dimension q ($q = 1, 2, \ldots, k$) then $\tilde{X}_a$ is inside the hypercube but not necessarily inside the hypersphere itself. To find out whether $\tilde{X}_a$ is inside the hypersphere, R_{ia} should be calculated according to Eq. (C.4) and the following comparisons should then be carried out:

If $R > R_{ia}$ then $\tilde{X}_a$ is inside the hypersphere (it is closer to $\tilde{X}_i$ than the centroid $\tilde{X}_j$) and the following variables are updated:

$$ME = T_{ia} R_{ia}^2 = ME_{ia} \,,$$

$$W = i \; ; \quad \text{and} \quad V = a \,.$$

If $R \leq R_{ia}$ then $\tilde{X}_a$ does not lie inside the hypersphere, thus it is rejected.

Step 2 is carried out until all combinations of centroid pairs are tested. At the conclusion of the algorithm, the indices of the closest clusters are stored in W and V, and the error incurred by merging those clusters is stored in ME.

Appendix D

Complexity Study

In this Appendix, a theoretical study of the complexity of the CNNC algorithm in comparison to the brute-force method, described in [Ramamurthi 1986], is presented. The following assumptions have been made during the complexity study:

a) M different classes are defined for the CVQ system.

b) n training vectors are used for codebook design; these vectors are equally divided to the M classes so that n/M training vectors are used in each class for designing the partial codebook for that class.

c) One iteration of the LBG algorithm is needed for designing a partial codebook for a certain class within the brute-force algorithm.

d) The dimension of an image vector is k.

e) N_i is the number of codevectors allocated to class i; every permissible allocation for the set $\{N_i\}$ such that $N_i \neq 0$ for all i and $\sum_{i=1}^{M} N_i = N$ should be checked within the brute-force algorithm.

The CNNC Algorithm

During the first step of the algorithm, a search within *each* class is conducted to find a pair of clusters which incur minimum merging error when merged. We assume that the merging error should be calculated for each pair of clusters without employing the fast search algorithm described in Appendix C. The merging error should be calculated K_0 times for each class, requiring T_0 calculations in total during the first step, i.e.,

$$T_0 \; = \; K_0 M \; = \; \frac{1}{2}\frac{n}{M}(\frac{n}{M}-1)M \; . \tag{D.1}$$

The pairs of clusters, found during the first step in the different classes, are kept on a merging list. During the next stages, pairs of clusters which incur minimum merging error among all the pairs on the merging list, are being selected in each stage and merged. After merging a pair of clusters in one class, a search is conducted in that class to find a new pair of clusters as candidates for merging in the next stage. Suppose that i pairs of clusters have already been merged in some class then, the number of clusters left in that class is $(n/M - i)$. Thus, the merging error should be calculated K_i times to find a new pair of clusters in *that* class as candidates for merging in the next stage where

$$K_i \; = \; \frac{1}{2}(\frac{n}{M}-i)(\frac{n}{M}-i-1) \; . \tag{D.2}$$

It should be noted that maximum mathematical operations are required if in each stage of the algorithm the search for a new pair of clusters is conducted in the largest class, i.e., a merge occurs within the class which contains the largest number of left clusters. Since we assume that at the starting point the number of training vectors (clusters) in each class is the same, maximum operations are required if merging is carried out in a cyclic way. Thus, we assume that after merging clusters in each of the M classes a new cycle of merges is started, and the process is carried out in this manner until the number of left clusters in the M classes accumulates to the desired number of codevectors N.

The number of *integer* cycles needed for the conclusion of the CNNC algorithm is denoted by J, i.e.,

$$J \; = \; \left\lfloor \frac{n-N}{M} \right\rfloor \; . \tag{D.3}$$

It is assumed that the algorithm concludes with a search for a new pair of clusters conducted within the last class where merging took place. Thus, for $J=0$ (incomplete merging cycle) the total number of error calculations is:

$$T_{0+} \; = \; K_0 M + K_1 S \quad \text{for} \quad J=0 \tag{D.4}$$

where $S = (n - N) < M$ is the number of classes within which one merge took place, K_1 is defined in Eq. (D.2) for $i = 1$, and '0+' denotes the first incomplete merging cycle. Thus, employing equations (D.2) and (D.1) we get

$$T_{0+} = \frac{1}{2}\left[\frac{n}{M}(\frac{n}{M} - 1)(n - N + M) - 2(\frac{n}{M} - 1)(n - N)\right] \quad \text{for } J = 0\ . \tag{D.5}$$

Suppose now that $J \neq 0$ complete merging cycles, and another $S < M$ merges took place till the conclusion of the codebook design. Then, the total number of error calculations is given by

$$T_J = K_0 M + \sum_{i=1}^{J} K_i M \quad \text{for } J \neq 0 \text{ and } S = 0\ , \tag{D.6}$$

and by

$$T_{J+} = K_0 M + \sum_{i=1}^{J} K_i M + K_{J-1} S \quad \text{for } J \neq 0 \text{ and } 0 < S < M \tag{D.7}$$

where $S = (n - N - MJ)$, and '$J+$' denotes J complete merging cycles plus one incomplete cycle during which S merges took place. It should be noted that in each of the last S classes the search is carried out among $(J - 1)$ left clusters, requiring K_{J-1} error calculations during each search.

Eq. (D.7) can be written explicitly as

$$\begin{aligned} T_{J+} &= \frac{1}{2}\frac{n}{M}(\frac{n}{M} - 1) + M\sum_{i=1}^{J}\frac{1}{2}(\frac{n}{M} - i)(\frac{n}{M} - i - 1)\ + \\ &+ \frac{1}{2}(\frac{n}{M} - J - 1)(\frac{n}{M} - J - 2)(n - N - MJ)\ . \end{aligned} \tag{D.8}$$

It can be shown that after some mathematical manipulations we get

$$\begin{aligned} T_{J+} &= \frac{M}{2}\frac{n}{M}(J + 1)(\frac{n}{M} - 1)\ + \\ &+ \frac{1}{2}(n - N - MJ)(\frac{n}{M} - J - 1)(\frac{n}{M} - J - 2)\ - \\ &- \frac{MJ}{4}(\frac{2n}{M} - 1)(J + 1) + \frac{MJ}{12}(J + 1)(2J + 1)\ . \end{aligned} \tag{D.9}$$

A good approximation can be found by replacing J with $(n - N)/M$, i.e., taking J to be a real number. It can be shown that

$$T_{J+} \approx \frac{(n - N + M)^3 + (n - N + M)[3n(N - M) - M^2]}{6M^2} \quad \text{for } J > 0\ . \tag{D.10}$$

The merging error is defined in Eq. (C.1) in Appendix C. The following mathematical operations are required each time the merging error is being calculated: $(k+2)$ multiplications, k subtractions, k additions, and one division. Thus, for example, the total number of multiplications is

$$P_C \;\propto\; \frac{(n-N+M)^3(k+2)}{6M^2} \tag{D.11}$$

where $\propto$ denotes 'proportional to'. It should be emphasized that T_{J+} is the upper bound on the required number of error calculations for the mentioned set of assumptions.

The Brute-Force Method

Every permissible allocation for the set $\{N_i\}$ of codevectors such that $N_i \neq 0$ for all i, and $\sum_{i=1}^{M} N_i = N$ should be checked to find an optimal allocation that yields the least average distortion (see section 2.3 for details). The question: "How many permissible allocations are available ?" is equivalent to the question: "How many ways exist for dividing N balls into M boxes without leaving any box empty ?". First, assuming that $N > M$, one ball is put in each box thus ensuring that no box is left empty. The question now becomes: "How many ways exist for dividing $(N-M)$ balls into M boxes ?". It can be shown that the answer is:

$$Z \;=\; \binom{(M+N-M-1)}{(M-1)} \;=\; \frac{(N-1)!}{(M-1)!(N-M)!} \tag{D.12}$$

where ! denotes factorial.

According to the brute-force method, for each allocation $\{N_i^{(t)}\}$ $t = 1, 2, \ldots, T$, M sub-codebooks are designed employing the LBG algorithm. The distortion measure used within the LBG algorithm is the squared error which is defined by

$$d(\bar{x}, \bar{y}_i) \;=\; \sum_{q=1}^{k} (x_q - y_{iq})^2 \tag{D.13}$$

where $\bar{x}$ is a training vector, and $\bar{y}_i$ is a codevector. The following mathematical operations are required each time the squared error is being calculated: k multiplications, k subtractions, and $(k-1)$ additions. Thus, for a given allocation of codevectors, $\{N_i^{(t)}\} = \{N_1^{(t)}, N_2^{(t)}, \ldots, N_M^{(t)}\}$, and for the first assignment of training vectors (in each class) to the clusters represented by the initial guess of codevectors (in each class), the number of required multiplications is

$$\begin{aligned} T_1 &= \frac{n}{M}N_1^{(t)}k + \frac{n}{M}N_2^{(t)}k + \cdots + \frac{n}{M}N_M^{(t)}k = \\ &= \frac{nk}{M}\sum_{i=1}^{M} N_i^{(t)} = \frac{nk}{M}N \ . \end{aligned} \tag{D.14}$$

Since we assume that one iteration of the LBG algorithm is needed for designing a satisfactory sub-codebook in each class for a given allocation of codevectors, the number of required multiplications is doubled. To understand our argument, one must remember that after the initial assignment of training vectors to clusters represented by a first guess of codevectors, new centroids are calculated for each cluster and defined as the new codevectors. Then a second pass of assigning training vectors in each class to clusters that are represented by those new centroids is carried out. The algorithm concludes after calculating new centroids for each cluster based on the assignment which took place during the second pass. Thus, the total number of multiplications needed for the design of a CVQ codebook is

$$P_B = \frac{2nk}{M}NZ = \frac{2nk}{M}N\frac{(N-1)!}{(M-1)!(N-M)!} \ . \tag{D.15}$$

Equation (D.15) defines a lower bound on the number of required multiplications since, in general, more iterations than one are needed for the design of a sub-codebook employing the LBG algorithm. It should be noted that mathematical operations, involved in calculating centroids within the brute-force algorithm, as well as comparisons have been ignored in our study. Mathematical operations, involved in calculating the weighted average centroid when two clusters are being merged within the CNNC algorithm, have been ignored too. Consequently, for n=8196, M=15, and N=512, approximately 6.1×10^9 multiplications are required for the CNNC algorithm while requiring 7.1×10^{33} multiplications for the brute-force algorithm.

Appendix E

A New Merging Criterion

In this appendix, a new merging criterion, to be used within the CNNC algorithm, is developed. The new merging criterion is proved to ensure shape preservation when employed in the process of designing the M shape codebooks, needed for the new DCGSVQ system.

We start by introducing the following notation and definitions :

a) S_i is a cluster of k-dimensional normalised vectors,
$S_i = \{\bar{x}_{nl} \in R^k \ ; \ \bar{x}_{nl}^T \bar{x}_{nl} = 1, \quad l = 1, 2, ..., m_i\}$, where m_i is the number of normalised vectors in cluster S_i and $(\cdot)^T$ denotes transpose.

b) $\tilde{X}_i$ is the *centroid* of cluster S_i. When the squared error is being used as the distortion measure the centroid is defined as follows [Helmuth 1980]:

$$\tilde{X}_i \ = \ \frac{1}{m_i} \sum_{\bar{x}_n \in S_i} \bar{x}_n \ . \tag{E.1}$$

It should be noted that the centroid of a set of normalised vectors is not necessarily normalised.

c) A is the set of all normalised vectors $\bar{a}$, i.e., $A = \{\bar{a} \in R^k; \ \bar{a}^T \bar{a} = 1\}$.

d) ρ_i is the total cross-correlation between all the vectors in cluster S_i and a normalised vector $\bar{a}$, i.e.,

$$\rho_i = \sum_{\bar{x}_n \in S_i} \bar{x}_n^T \bar{a} \ . \tag{E.2}$$

Secondly, we consider the following basic problem:

Given a cluster S_i of normalised vectors find a representative normalised vector, $\bar{a} \in A$, such that ρ_i is maximised.
Equivalently, the following notation may be used

$$\bar{a}_i = \max_{\bar{a} \in A}{}^{-1} \left[\sum_{\bar{x}_n \in S_i} \bar{x}_n^T \bar{a} \right] \tag{E.3}$$

where the inverse maximum means that $\bar{a}_i$ is that vector $\bar{a} \in A$ which ensures maximum ρ_i.

To solve this problem the following lemma is stated and proved.

Lemma 1: *The normalised vector, $\bar{a} \in A$, which ensures maximum ρ_i in cluster S_i is the normalised centroid of that cluster.*

Proof: Let $d(\bar{x}_n, \bar{a})$ denote the distortion measure between $\bar{x}_n$ and $\bar{a}$.
Employing the squared error as the distortion measure we have

$$d(\bar{x}_n, \bar{a}) = \| \bar{x}_n - \bar{a} \|^2 = \bar{x}_n^T \bar{x}_n - 2\bar{x}_n^T \bar{a} + \bar{a}^T \bar{a}. \tag{E.4}$$

Since $\bar{x}_n$ and $\bar{a}$ are both normalised vectors, we get

$$d(\bar{x}_n, \bar{a}) = 2 \left[1 - \bar{x}_n^T \bar{a} \right]. \tag{E.5}$$

The total squared error in the cluster S_i incurred when all the shape vectors $\bar{x}_n \in S_i$ are represented (coded) by the vector $\bar{a}$ is

$$D_i = \sum_{\bar{x}_n \in S_i} d(\bar{x}_n, \bar{a}) = \sum_{\bar{x}_n \in S_i} 2 \left[1 - \bar{x}_n^T \bar{a} \right]. \tag{E.6}$$

Equivalently equation (E.6) can be written as follows

$$D_i = 2 \left[m_i - \sum_{\bar{x}_n \in S_i} \bar{x}_n^T \bar{a} \right]. \tag{E.7}$$

Now let $(\bar{a}_i)_{min} \in A$ denote that normalised vector which ensures minimum total squared error in cluster S_i , i.e.,

$$(\bar{a}_i)_{min} = \min_{\bar{a} \in A}{}^{-1} [D_i] = \min_{\bar{a} \in A}{}^{-1} 2 \left[m_i - \sum_{\bar{x}_n \in S_i} \bar{x}_n^T \bar{a} \right]. \tag{E.8}$$

Since m_i is a given entity, the argument within the brackets in equation (E.8) can be minimised by maximising $\sum_{\bar{x}_n \in S_i} \bar{x}_n^T \bar{a}$, i.e.,

$$(\bar{a}_i)_{min} = \max_{\bar{a} \in A}^{-1} \left[\sum_{\bar{x}_n \in S_i} \bar{x}_n^T \bar{a} \right] . \tag{E.9}$$

However, equations (E.9) and (E.3) are identical. Therefore, we have shown that the normalised vector $\bar{a} \in A$, that ensures maximum total cross-correlation between all the vectors in cluster S_i and itself, is also that normalised vector that ensures the minimum total squared error in that cluster, i.e.,

$$\bar{a}_i = (\bar{a}_i)_{min} . \tag{E.10}$$

Returning to equation (E.3), it can be rewritten as follows

$$\bar{a}_i = \max_{\bar{a} \in A}^{-1} \left[\bar{a}^T \sum_{\bar{x}_n \in S_i} \bar{x}_n \right] . \tag{E.11}$$

Then substituting equation (E.1) in equation (E.11) we get

$$\bar{a}_i = \max_{\bar{a} \in A}^{-1} \left[m_i \, \bar{a}^T \tilde{X}_i \right] \equiv \max_{\bar{a} \in A}^{-1} \left[\bar{a}^T \tilde{X}_i \right] . \tag{E.12}$$

Therefore, the vector $\bar{a} \in A$ should be in the same direction as the centroid $\tilde{X}_i$, i.e.,

$$\bar{a}_i = (\bar{a}_i)_{min} = \frac{\tilde{X}_i}{\| \tilde{X}_i \|} , \tag{E.13}$$

where $\| \tilde{X}_i \|$ is the *magnitude* (norm) of $\tilde{X}_i$.

Equation (E.13) states exactly what has been put into words by lemma 1. Therefore, lemma 1 is proven.

So far we have shown how to select that shape vector which can best represent the cluster S_i of shape vectors. Next we consider the second basic problem.

Given two clusters of normalised vectors,
$S_i = \{\bar{x}_{nl} \in R^k \ ; \ \bar{x}_{nl}^T \bar{x}_{nl} = 1, \quad l = 1, 2, ..., m_i\}$, and
$S_j = \{\bar{x}_{nl} \in R^k \ ; \ \bar{x}_{nl}^T \bar{x}_{nl} = 1, \quad l = 1, 2, ..., m_j\}$ where m_i and m_j denote the number of vectors in cluster S_i and S_j respectively, and $S_i \cap S_j = \emptyset$, the empty set. A *unified cluster* S_{ij} is produced by merging cluster S_i with cluster S_j ,i.e., $S_{ij} = S_i \cup S_j$.
Find the merging error incurred when S_{ij} is being represented by a normalised vector $\bar{a}_{ij} \in A$.

The merging error ME_{ij} is defined in Appendix B by

$$ME_{ij} = D_{ij} - (D_i + D_j) \ , \tag{E.14}$$

where D_i , D_j and D_{ij} denote the total squared error incurred when all the normalised vectors in clusters S_i , S_j and S_{ij} are being coded (represented) by the representative vectors $\bar{a}_i$, $\bar{a}_j$, and $\bar{a}_{ij}$, respectively. In other words, the merging error is the extra total squared error incurred by merging the clusters and coding all the vectors by a new representative vector.
Employing equations (E.1) and (E.7), and replacing $\bar{a}$ with $\bar{a}_i$ we get

$$D_i = (D_i)_{min} = 2\left[m_i - \bar{a}_i^T \sum_{\bar{x}_n \in S_i} \bar{x}_n\right] = 2m_i\left[1 - \bar{a}_i^T \tilde{X}_i\right] \ . \tag{E.15}$$

Using equation (E.13) and remembering that $\bar{a}_i$ is a normalised vector we get

$$D_i = 2m_i\left[1 - \bar{a}_i^T \bar{a}_i \parallel \tilde{X}_i \parallel\right] = 2m_i\left[1 - \parallel \tilde{X}_i \parallel\right] \ . \tag{E.16}$$

Following equation (E.16) we have

$$D_j = 2m_j\left[1 - \parallel \tilde{X}_j \parallel\right] \ , \text{ and} \tag{E.17}$$

$$D_{ij} = 2(m_i + m_j)\left[1 - \parallel \tilde{X}_{ij} \parallel\right] \ . \tag{E.18}$$

Employing equations (E.14) and (E.16) through (E.18) we get

$$ME_{ij} = 2(m_i + m_j)\left[1 - \parallel \tilde{X}_{ij} \parallel\right] - 2m_i\left[1 - \parallel \tilde{X}_i \parallel\right] - 2m_j\left[1 - \parallel \tilde{X}_j \parallel\right] \ . \tag{E.19}$$

$\tilde{X}_{ij}$ has been defined in Appendix B as

$$\tilde{X}_{ij} = \frac{m_i \tilde{X}_i + m_j \tilde{X}_j}{m_i + m_j} \tag{E.20}$$

therefore, by substituting equation (E.20) in equation(E.19) and arranging the arguments we get :

$$ME_{ij} = 2m_i \parallel \tilde{X}_i \parallel + 2m_j \parallel \tilde{X}_j \parallel - 2 \parallel m_i\tilde{X}_i + m_j\tilde{X}_j \parallel . \tag{E.21}$$

Nice formula as it is, equation (E.21) seems to have no meaning with respect to our shape vectors. However, as will be shown next, it does have a great meaning.

Using $\bar{a}_i$ as the representative shape vector for cluster S_i, and employing equations (E.1) and (E.13), equation (E.2) can be rewritten as follows :

$$\rho_i = \bar{a}_i^T \sum_{\bar{x}_n \in S_i} \bar{x}_n = m_i \bar{a}_i^T \tilde{X}_i = m_i \bar{a}_i^T \bar{a}_i \parallel \tilde{X}_i \parallel .$$

Since $\bar{a}_i$ is normalised we finally get

$$\rho_i = m_i \parallel \tilde{X}_i \parallel . \tag{E.22}$$

Similarly,

$$\rho_j = m_j \parallel \tilde{X}_j \parallel , \text{ and} \tag{E.23}$$

$$\rho_{ij} = (m_i + m_j) \parallel \tilde{X}_{ij} \parallel = \parallel m_i\tilde{X}_i + m_j\tilde{X}_j \parallel . \tag{E.24}$$

Using these results, equation (E.21) becomes

$$ME_{ij} = 2\left[(\rho_i + \rho_j) - \rho_{ij}\right] . \tag{E.25}$$

Thus, ME_{ij} is actually a measure of the damage, in terms of total cross-correlation, caused by merging cluster S_i with cluster S_j and having $\bar{a}_{ij}$ as the new representative shape vector for the unified cluster S_{ij}. The smaller the merging error the smaller will be the damage in terms of shape preservation, and vice versa. That is why the merging error, as defined in equation (E.21), can be used as a meaningful merging criterion within the CNNC algorithm to design the M shape codebooks, needed for the proposed DCGSVQ system.
To conclude, the following remarks should be remembered :

a) The vectors, used as the *training set* for designing the shape codebooks, should all be normalised before the algorithm is started.

b) At the conclusion of the algorithm, all the unified centroids should be normalised in order to get the required shape codebooks.

Appendix F

The Pixel Classifier

In this appendix, we describe the *pixel classifier*, employed within the postprocessing algorithm for detecting *edge pixels*, i.e., pixels which lie on or near an edge. It is based on the analysis of a 3x3 neighbourhood, surrounding the pixel under test, and on known properties of the human visual system.
Let $I_{ave}(x, y, q, p)$ denote the average intensity of two adjacent pixels, $I(x, y)$ and $I(x + q, y + p)$, i.e.

$$I_{ave}(x, y, q, p) = \frac{I(x, y) + I(x + q, y + p)}{2} \tag{F.1}$$

for $q, p = -1, 0, 1$ excluding the case $q = p = 0$.
The gradients in different directions are defined as follows:

$$d(x, y, q, p) = \mid I(x, y) - I(x + q, y + p) \mid \ . \tag{F.2}$$

Let $d_n(x, y, q, p)$ denote the gradients in various directions normalised by the average intensity of two adjacent pixels. It is defined as follows:

$$d_n(x, y, q, p) = \frac{d(x, y, q, p)}{I_{ave}(x, y, q, p)} = \frac{2 \mid I(x, y) - I(x + q, y + p) \mid}{I(x, y) + I(x + q, y + p)} \ . \tag{F.3}$$

This definition is in accord with the well known fact that the sensitivity of the human visual system is proportional to the normalised gradient and not to the gradient itself [Pratt 1978].

Two counters m and n are defined and set to zero. These counters are incremented according to :

$$\begin{aligned} m &= m+1 \quad \text{if } d_n(x,y,q,p) \geq T \\ n &= n+1 \quad \text{if } d_n(x,y,q,p) < T \end{aligned} \tag{F.4}$$

for $q,p = -1,0,1$ excluding the case $q = p = 0$.
The threshold T is used for detecting most of the edges and avoiding false detection. It was determined by experiments and found to be a function of the average intensity of the two pixels under consideration when I_{ave} was low (under 50 on an intensity scale between 0-255). The threshold T is defined by

$$T = T(x,y,q,p) = \begin{cases} \frac{10}{I_{ave}(x,y,q,p)} & \text{if } I_{ave}(x,y,q,p) < 50.0 \\ 0.2 & \text{otherwise .} \end{cases} \tag{F.5}$$

A pixel is defined as a *non-edge pixel* if $m \leq 1$ or $n = 0$, otherwise it is defined as an *edge pixel.* It should be noticed that $n = 0$ when the normalised gradients between the pixel under test and each of the pixels surrounding it are greater than the threshold T. In that case we assume that the pixel under test is contaminated by some random speckle noise, thus classifying it as a non-edge pixel rather than an edge pixel.

Appendix G

Filter Design Based on Properties of HVS

In this appendix, we describe a new filter design method which incorporates perceptual properties of the human visual system. This method has been employed for designing the various filters, used for postprocessing the block-coded images throughout our research.

G.1 Introduction

Typical frequency design specifications of a low-pass filter used for noise removal are:

a. δ_1 - the deviation of the magnitude characteristics from that of an ideal filter in the passband;

b. δ_2 - the amount of attenuation in the stopband;

c. $(f_h - f_l)$ - the width of the transition band; and,

d. δ_3 - the deviation of the group delay from that of the ideal filter's constant group delay in the passband.

While these frequency specifications are quite simple and many design methods that incorporate them have been developed, they are not adequate for the design of image processing filters where the visual appearance of the resulting image is of prime importance (see [Hentea 1984] for discussion). This follows from the fact that an "ideal" low-pass filter, approximated by these design specifications, has itself impulse and step responses that are visually poor approximations of the ideal pulse and step functions. Two undesirable visual effects are observed when "ideal" low-pass filters are employed in image processing: (i) the blurring of edges and other sharp details, and (ii) the occurrence of significant ripples in the vicinity of edges.

A new technique which is better suited for the design of image processing filters has been proposed by Hentea and Algazi in [Hentea 1984]. This technique is based on a weighted least squares design procedure which incorporates perceptual properties of the human visual system (HVS) as part of the optimization process. This technique has been successfully used for designing FIR anisotropic filters for image enhancement [Algazi 1986] and FIR low-pass filters for decimation and interpolation [Algazi 1989].

Two perceptual properties of human vision have been incorporated into the new filter design procedure: contrast sensitivity and spatial visual masking. Both properties have been described in detail in chapter 3. The following two functions have been proposed in the literature for describing the contrast sensitivity, or equivalently the modulation transfer function, of the HVS :

$$H_v(f) = 2.6[0.0192 + 0.114f]\exp[-(0.114f)^{1.1}] \text{ and,} \tag{G.1}$$

$$H_v(f) = [0.2 + 0.45f]\exp(-0.18f) \ . \tag{G.2}$$

The first function has been proposed by Mannos and Sakrison having a peak value at a spatial frequency $f = 8.0$ cycles per degree [Mannos 1974]. The second function has been proposed by Nill having a peak value at a spatial frequency $f = 5.2$ cycles per degree [Nill 1985]. These functions have been used alternatively by us within the filter design procedure, and are depicted in Fig. G.1.

The spatial visual masking effect has been usually modeled by a visibility function, also called the *spatial masking function.* The visibility function, $vi(x)$, has been defined in chapter 3 by

$$vi(x) = 1 - a^{|x|} \tag{G.3}$$

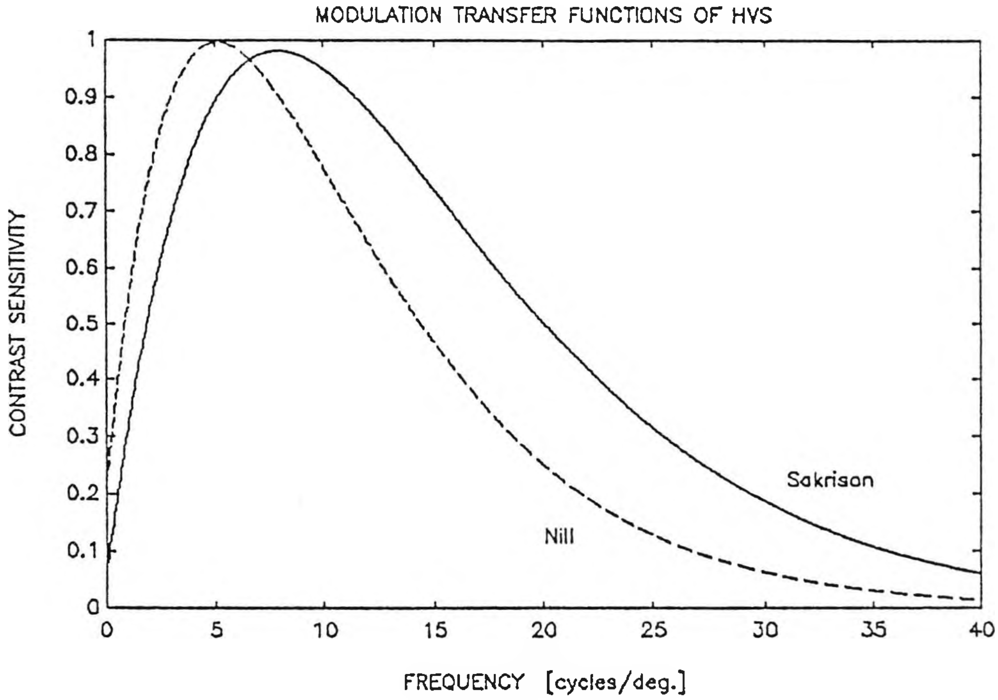


Figure G.1: *Modulation transfer functions of the HVS.*

where x is the distance from an edge, and the constant a lies between 0.0 and 1.0 (depending on the ratio of the bright and dark intensities of the edge). Following Hentea and Algazi [Hentea 1984], we will use the value $a = 0.75$ which was found to be best suited for our purposes.

We present next a general formulation of the design problem and describe the design technique which incorporates the perceptual properties mentioned above.

G.2 Problem Formulation and Filter Design

Consider the following block diagram, depicted in Fig G.2, which models a broad class of image processing problems. The image formation system $h_s(x, y)$ introduces spatial degradation of the image $I(x, y)$. We denote by $n(x, y)$ some additive, unwanted noise or distortion introduced, for example, by a block coding system. The role of the processing filter, $h(x, y)$, is to produce an approximation of the true image, $I(x, y)$, and to reduce noise corrupting $I'(x, y)$. The cascade of the two systems $h_s(x, y)$ and $h(x, y)$ will be, in general, a low-pass filter that

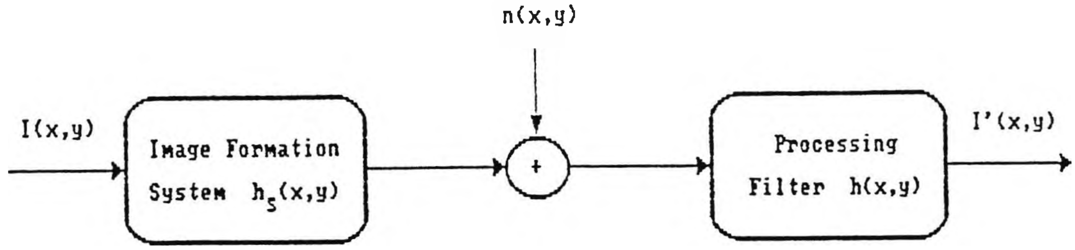


Figure G.2: *Block diagram of an image processing system* (from [Hentea 1984]).

achieves a compromise between noise suppression and signal preservation.

We are interested in the design of one-dimensional digital FIR filters thus, we will consider the system described in Fig. G.2 to be one-dimensional too. One can easily obtain a two-dimensional filter from a one-dimensional filter by McClellan's transformation [McClellan 1973], provided that the two-dimensional filter has the following properties:
(i) its frequency response is approximately circularly symmetric, and
(ii) its line spread function in the horizontal or vertical directions is identical to the impulse response of the one-dimensional filter.

The following notation is introduced. Let $u(x)$ denote an ideal unit step function to be approximated, $u'(x)$ denote the actual step response of the complete system (Fig. G.2), and $w_1(x)$ denote a spatial domain weighting function. In addition, let $H_{id}(f)$ denote an "ideal" frequency response of the filter to be approximated, $H(f)$ denote the actual frequency response of the designed filter, and $W_2(f)$ denote a frequency weighting function.

The following integrals are defined and used for filter design:

$$\mathcal{I}_1 = \int_{-\infty}^{\infty} w_1^2(x) \mid u'(x) - u(x) \mid^2 dx \quad \text{, and} \tag{G.4}$$

$$\mathcal{I}_2 = \int_{-\infty}^{\infty} W_2^2(f) \mid H(f) - H_{id}(f) \mid^2 df \quad . \tag{G.5}$$

$w_1^2(x)$ in equation (G.4) can be taken to be the masking function for an edge, thus assigning the errors (ripples near an edge) a weight proportional to their visibility. $W_2^2(f)$ in equation (G.5) can be taken to be the frequency response of the HVS, thus assigning the errors a weight proportional to their perceptual importance in the frequency domain.

The step response of the complete system, $u'(x)$, assuming $n(x) = 0$, is given by:

$$u'(x) = u_s(x) * h(x) \tag{G.6}$$

where $*$ denotes convolution and $u_s(x) = u(x) * h_s(x)$.
The filter design objective is to determine $h(x)$ which minimizes $\mathcal{I}_1$ subject to $\mathcal{I}_2$ being a fixed given value. Employing a Lagrange multiplier β, the design problem becomes: find $h(x)$, which minimizes $\mathcal{I}_1 + \beta\mathcal{I}_2$ for a given value of β.

We consider the design of FIR digital filters having an impulse response of the form

$$h(x) = \sum_{i=-N}^{N} h_i \delta(x - i\Delta x) \tag{G.7}$$

where Δx is the sampling interval in the spatial domain, and $\delta(x)$ is the unit impulse function defined as:

$$\delta(x) = \begin{cases} 1 & \text{for } x = 0 \\ 0 & \text{for } x \neq 0 \end{cases} \quad . \tag{G.8}$$

By substituting $u'(x)$ from equation (G.6) into equation (G.4) we obtain

$$\begin{aligned} \mathcal{I}_1 &= \int_{-\infty}^{\infty} w_1^2(x) \left[u_s(x) * h(x)\right]^2 dx - \\ &- 2\int_{-\infty}^{\infty} w_1^2(x) \left[u_s(x) * h(x)\right] u(x) dx + \int_{-\infty}^{\infty} w_1^2(x) u^2(x) dx \quad . \end{aligned} \tag{G.9}$$

Defining convolution between $u_s(x)$ and $h(x)$ as

$$u_s(x) * h(x) = \sum_{i=-N}^{N} h_i u_s(x - i\Delta x) \tag{G.10}$$

and substituting equation (G.10) into equation (G.9) we obtain

$$\begin{aligned} \mathcal{I}_1 &= \int_{-\infty}^{\infty} w_1^2(x) \left[\sum_{i=-N}^{N} h_i u_s(x - i\Delta x)\right]^2 dx - \\ &- 2\int_{-\infty}^{\infty} w_1^2(x) u(x) \left[\sum_{i=-N}^{N} h_i u_s(x - i\Delta x)\right] dx + \\ &+ \int_{-\infty}^{\infty} w_1^2(x) u^2(x) dx \quad . \end{aligned} \tag{G.11}$$

Taking partial derivatives with respect to h_k for $k = -N, \ldots, -1, 0, 1, \ldots, N$ we obtain

$$\begin{aligned} \frac{\partial \mathcal{I}_1}{\partial h_k} &= \int_{-\infty}^{\infty} w_1^2(x) \left[2\sum_{i=-N}^{N} h_i u_s(x - i\Delta x)\right] u_s(x - k\Delta x) dx - \\ &- 2\int_{-\infty}^{\infty} w_1^2(x) u(x) u_s(x - k\Delta x) dx \quad . \end{aligned} \tag{G.12}$$

Rearranging equation (G.12) we obtain

$$\frac{\partial \mathcal{I}_1}{\partial h_k} = 2 \sum_{i=-N}^{N} h_i a_{ik} - d_k \tag{G.13}$$

where

$$a_{ik} = \int_{-\infty}^{\infty} w_1^2(x) u_s(x - i\Delta x) u_s(x - k\Delta x) dx \quad \text{and,} \tag{G.14}$$

$$d_k = 2 \int_{-\infty}^{\infty} w_1^2(x) u(x) u_s(x - k\Delta x) dx \tag{G.15}$$

for $i, k = -N, \ldots, -1, 0, 1, \ldots, N$.

Equation (G.13) can be written in matrix form as

$$\frac{\partial \mathcal{I}_1}{\partial \mathbf{h}} = 2\mathbf{A}\mathbf{h} - \mathbf{d} \tag{G.16}$$

where

$$\begin{aligned}
\mathbf{h}^T &= [h_{-N}, \ldots\ldots, h_{-1}, h_0, h_1, \ldots\ldots, h_N] \ , \\
\mathbf{d}^T &= [d_{-N}, \ldots\ldots, d_{-1}, d_0, d_1, \ldots\ldots, d_N] \ , \quad \text{and} \\
\mathbf{A} &= \{a_{ik}\}, \quad \text{for} \quad i, k = -N, \ldots, N \ .
\end{aligned}$$

It should be noted that $\mathbf{A}$ is a symmetric positive definite matrix and $\mathbf{d}$ is a constant vector.

Turning back to $\mathcal{I}_2$, by Parseval's theorem equation (G.5) can be written as

$$\mathcal{I}_2 = \int_{-\infty}^{\infty} \{w_2(x) * [h(x) - h_{id}(x)]\}^2 \, dx \tag{G.17}$$

where $w_2(x)$ is the inverse Fourier transform of $W_2(f)$.

Equation (G.17) can be written as follows:

$$\mathcal{I}_2 = \int_{-\infty}^{\infty} \left\{ \sum_{i=-N}^{N} h_i w_2(x - i\Delta x) - w_2(x) * h_{id}(x) \right\}^2 dx \ . \tag{G.18}$$

Taking partial derivatives with respect to h_k for $k = -N, \ldots, -1, 0, 1, \ldots, N$ we obtain

$$\begin{aligned}
\frac{\partial \mathcal{I}_2}{\partial h_k} &= 2 \sum_{i=-N}^{N} h_i \int_{-\infty}^{\infty} w_2(x - i\Delta x) w_2(x - k\Delta x) dx - \\
&- 2 \int_{-\infty}^{\infty} w_2(x - k\Delta x) \left[w_2(x) * h_{id}(x) \right] dx \ . \qquad \text{(G.19)}
\end{aligned}$$

Equation (G.19) can be written as follows:

$$\frac{\partial \mathcal{I}_2}{\partial h_k} = 2\sum_{i=-N}^{N} h_i b_{i-k} - c_k \qquad \text{(G.20)}$$

where

$$b_{i-k} = \int_{-\infty}^{\infty} w_2(x - i\Delta x) w_2(x - k\Delta x) dx \quad \text{and,} \qquad \text{(G.21)}$$

$$c_k = 2\int_{-\infty}^{\infty} w_2(x - k\Delta x) \left[w_2(x) * h_{id}(x)\right] dx \qquad \text{(G.22)}$$

for $i, k = -N, \ldots, -1, 0, 1, \ldots, N$.

Equation (G.20) can be written in matrix form as

$$\frac{\partial \mathcal{I}_2}{\partial \mathbf{h}} = 2\mathbf{B}\mathbf{h} - \mathbf{c} \qquad \text{(G.23)}$$

where

$$\mathbf{h}^T = [h_{-N}, \ldots\ldots, h_{-1}, h_0, h_1, \ldots\ldots, h_N] \ ,$$

$$\mathbf{c}^T = [c_{-N}, \ldots\ldots, c_{-1}, c_0, c_1, \ldots\ldots, c_N] \ , \quad \text{and}$$

$$\mathbf{B} = \begin{bmatrix} b_0 & b_1 & \cdots & b_{2N} \\ b_1 & b_0 & \cdots & b_{2N-1} \\ \vdots & \vdots & \vdots & \vdots \\ b_{2N} & b_{2N-1} & \cdots & b_0 \end{bmatrix} .$$

It should be noted that $\mathbf{B}$ is a symmetric positive definite Toeplitz matrix and $\mathbf{c}$ is a constant vector. According to [Algazi 1975], the evaluation of $\mathbf{B}$ and $\mathbf{c}$ is more conveniently done using the Fourier transform domain. Thus, we show next alternative equations that will be used later for calculating these matrices.

Starting with b_{i-k} we first make the following change of variables :

$$\mu = x - k\Delta x \quad \text{and thus,} \quad x = \mu + k\Delta x \ .$$

Substituting x into equation (G.21) results in the following equation

$$b_{i-k} = \int_{-\infty}^{\infty} w_2(\mu) w_2(\mu - i\Delta x + k\Delta x) d\mu \ . \qquad \text{(G.24)}$$

Defining $\epsilon = (i - k)\Delta x$ and substituting it into equation (G.24) we obtain

$$b_{i-k} = \int_{-\infty}^{\infty} w_2(\mu) w_2(\mu - \epsilon) d\mu \ . \qquad \text{(G.25)}$$

Using Fourier transform $w_2(x)$ can be defined as

$$w_2(x) = \int_{-\infty}^{\infty} W_2(f) \exp[j 2\pi f x] df \tag{G.26}$$

where $j = \sqrt{-1}$ and $W_2(f)$ is the frequency weighting function employed in equation (G.5).

Equation (G.26) can be written as

$$w_2(x) = \int_{-\infty}^{\infty} [W_2(f) \cos(2\pi f x) + j W_2(f) \sin(2\pi f x)] \, df \ . \tag{G.27}$$

If we define $W_2(f)$ to be a real non-negative even function i.e., $W_2(f) \geq 0$ and $W_2(f) = W_2(-f)$, equation (G.27) becomes

$$w_2(x) = \int_{-\infty}^{\infty} W_2(f) \cos(2\pi f x) df \ . \tag{G.28}$$

From equation (G.28) we obtain that $w_2(x)$ is an even function too and thus equation (G.25) can be written as

$$b_{i-k} = \int_{-\infty}^{\infty} w_2(\mu) w_2(\epsilon - \mu) d\mu = w_2(\epsilon) * w_2(\epsilon) \tag{G.29}$$

Equivalently, in the Fourier domain we obtain

$$\begin{aligned} b_{i-k} &= \int_{-\infty}^{\infty} W_2^2(f) \exp[j 2\pi f \epsilon] df = \\ &= \int_{-\infty}^{\infty} \left[W_2^2(f) \cos(2\pi f \epsilon) + j W_2^2(f) \sin(2\pi f \epsilon) \right] df \ . \end{aligned} \tag{G.30}$$

Since $\sin(2\pi f \epsilon)$ is not a symmetrical function with respect to $f = 0$ (it is an odd function) we obtain

$$b_{i-k} = \int_{-\infty}^{\infty} W_2^2(f) \cos(2\pi f \epsilon) df \ . \tag{G.31}$$

Substituting $\epsilon = (i-k)\Delta x$ into (G.31), we finally obtain the following definition for b_{i-k}

$$b_{i-k} = 2 \int_{0}^{\infty} W_2^2(f) \cos[2\pi f (i-k) \Delta x] df \tag{G.32}$$

for $i, k = -N, \ldots, -1, 0, 1, \ldots, N$.

Turning back to equation (G.22), c_k can be defined as

$$c_k = 2 \int_{-\infty}^{\infty} w_2(x - k\Delta x) z(x) dx \tag{G.33}$$

where $z(x) = w_2(x) * h_{id}(x)$.
Since we have shown that $w_2(x)$ is an even function, equation (G.33) can be written as follows :

$$\begin{aligned} c_k &= 2\int_{-\infty}^{\infty} z(x) w_2(k\Delta x - x) dx = \\ &= 2z(k\Delta x) * w_2(k\Delta x) = \\ &= 2\left[w_2(k\Delta x) * w_2(k\Delta x) * h_{id}(k\Delta x)\right] . \end{aligned} \tag{G.34}$$

Equivalently, in the Fourier domain we obtain

$$\begin{aligned} c_k &= 2\int_{-\infty}^{\infty} W_2^2(f) H_{id}(f) \exp[j2\pi f k\Delta x] df = \\ &= 2\int_{-\infty}^{\infty} W_2^2(f) H_{id}(f) \cos(2\pi f k\Delta x) df + \\ &+ j\int_{-\infty}^{\infty} W_2^2(f) H_{id}(f) \sin(2\pi f k\Delta x) df . \end{aligned} \tag{G.35}$$

Finally, exploring symmetrical properties of the functions in equation (G.35) we find that

$$c_k = 4\int_0^{\infty} W_2^2(f) H_{id}(f) \cos(2\pi f k\Delta x) df \tag{G.36}$$

for $k = -N, \ldots, -1, 0, 1, \ldots, N$.

Turning back to the filter design problem, we have to find $\mathbf{h}$ which minimizes $\mathcal{I}_1 + \beta\mathcal{I}_2$ for a given value of β. The necessary condition for h_k to be an optimal solution to this problem is

$$\frac{\partial\mathcal{I}_1}{\partial h_k} + \beta\frac{\partial\mathcal{I}_2}{\partial h_k} = 0 \quad \text{for} \quad k = -N, \ldots, -1, 0, 1, \ldots, N \tag{G.37}$$

which, based on equations (G.16) and (G.23), leads to the matrix equation

$$2(\mathbf{A} + \beta\mathbf{B})\mathbf{h} = \beta\mathbf{c} + \mathbf{d} . \tag{G.38}$$

Since $(\mathbf{A} + \beta\mathbf{B})$ is a symmetric positive definite matrix, we obtain a unique solution for a given value of the parameter β, i.e.,

$$\mathbf{h} = \frac{1}{2}(\mathbf{A} + \beta\mathbf{B})^{-1}(\beta\mathbf{c} + \mathbf{d}) \tag{G.39}$$

where $(\mathbf{A} + \beta\mathbf{B})^{-1}$ is the inverse matrix of $(\mathbf{A} + \beta\mathbf{B})$.
The equations defining elements of the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{c}$ and $\mathbf{d}$ have been developed above and are summarized as follows:

$$a_{ik} = \int_{-\infty}^{\infty} w_1^2(x) u_s(x - i\Delta x) u_s(x - k\Delta x) dx ,$$

$$
\begin{aligned}
b_{i-k} &= 2\int_0^{\infty} W_2^2(f)\cos[2\pi f(i-k)\Delta x]df \ , \\
c_k &= 4\int_0^{\infty} W_2^2(f)H_{id}(f)\cos(2\pi f k\Delta x)df \ \text{ and,} \\
d_k &= 2\int_{-\infty}^{\infty} w_1^2(x)u(x)u_s(x-k\Delta x)dx \ . \qquad \text{(G.40)}
\end{aligned}
$$

for $i, k = -N, \ldots, -1, 0, 1, \ldots, N$.

Turning back to equation (G.4), if the system step response $u'(x)$ is defined to be an edge with ripples introduced by filtering, and we let

$$w_1^2(x) = vi(x) = 1 - 0.75^{|x|} \ , \qquad \text{(G.41)}$$

then $\mathcal{I}_1$ measures the visibility of those ripples . The luminance contrast sensitivity is introduced into equation (G.5) in a similar fashion. If we let, for example,

$$W_2^2(f) = H_v(f) = 2.6[0.0192 + 0.114f]\exp[-(0.114f)^{1.1}] \qquad \text{(G.42)}$$

then $\mathcal{I}_2$ measures the perceived deviation of the actual filter response from the ideal filter response in the frequency domain.
The function actually used for weighting is :

$$W_2^2(f) = \begin{cases} 1 & |f| \leq 8 \ \text{ cpd} \\ H_v(f) & \text{otherwise} \end{cases} \qquad \text{(G.43)}$$

in the case when $H_v(f)$ is defined according to Mannos and Sakrison as in equation (G.1), or

$$W_2^2(f) = \begin{cases} 1 & |f| \leq 5.2 \ \text{ cpd} \\ H_v(f) & \text{otherwise} \end{cases} \qquad \text{(G.44)}$$

in the case when $H_v(f)$ is defined according to Nill as in equation (G.2).
It should be noted that the low frequency attenuation of $H_v(f)$ is eliminated by extending the peak value of $H_v(f)$ down to zero cpd as proposed in [Algazi 1989].

To determine the ideal filter response $H_{id}(f)$, we define the low-pass filter frequency characteristic $H_{lp}(f)$ as the desired **overall** response of the system consisting of the ideal filter and the frequency weighting function representing the modulation transfer function of the HVS (see Fig. G.3 and [Algazi 1989]). This results in the relation :

$$H_{id}(f) = H_{lp}(f)/W_2^2(f) \ . \qquad \text{(G.45)}$$

The frequency response of the equivalent low-pass filter $H_{lp}(f)$ is depicted in the lower part of Fig. G.3, where F_c is the desired normalised cutoff frequency.

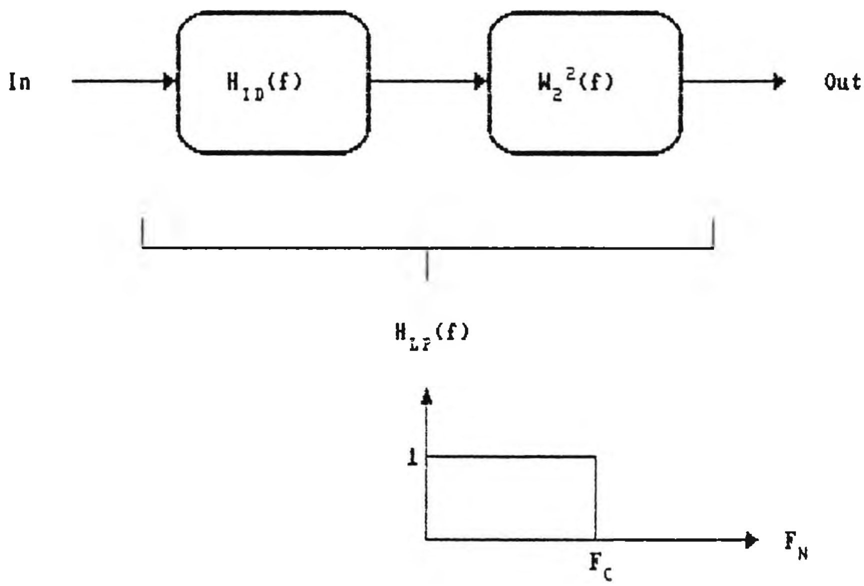


Figure G.3: *The desired overall low-pass system response $H_{lp}(f)$.*

Employing, for example, the weighting function defined in equation (G.43) and the set of equations developed above and summarized in (G.40) we show next how to calculate the matrices **A**, **B**, **c** and **d**. Substituting equation (G.43) into equation (G.40) which defines b_{i-k} we obtain

$$\begin{aligned} b_{i-k} &= 2\int_0^8 \cos[2\pi f(i-k)\Delta x]df \; + \qquad (G.46)\\ &+ \; 2\int_8^\infty H_v(f)\cos[2\pi f(i-k)\Delta x]df \quad \text{for} \quad i \neq k \;, \quad \text{and} \\ b_0 &= 2\int_0^8 df + 2\int_8^\infty H_v(f)df \quad \text{for} \quad i = k \;. \qquad (G.47) \end{aligned}$$

It should be noted that the upper limit ∞ in the integrals above stands for the highest noticeable spatial frequency. This upper limit depends on the dimensions of the TV monitor, the number of pixels along one scanned line in the tested digital image and on the distance between the observer and the monitor (see [Carrioli 1988] for discussion).

In a similar way, taking the definition of c_k in equation (G.40) and substituting equation (G.45) into it results in :

$$\begin{aligned} c_k &= 4\int_0^{f_c} \cos(2\pi f k\Delta x)df \quad \text{for} \quad k \neq 0 \ , \quad \text{and} && \text{(G.48)} \\ c_0 &= 4f_c \quad \text{for} \quad k = 0 && \text{(G.49)} \end{aligned}$$

where f_c is the desired cutoff frequency of the overall system given in units of cycles per degree.

Matrices **A** and **d** are calculated in the spatial domain employing the definitions in equation (G.40). To simplify the design procedure and without loss of generalization, we take the formation system $h_s(x)$ to be perfect (see Fig. G.2) so that equation (G.6) becomes simpler, i.e., $u_s(x) = u(x)$, and $u'(x) = u(x) * h(x)$. Substituting $u_s(x)$ into equations (G.40) we obtain :

$$\begin{aligned} a_{ik} &= \int_{-\infty}^{\infty} w_1^2(x)u(x - i\Delta x)u(x - k\Delta x)dx \ , \\ d_k &= 2\int_{-\infty}^{\infty} w_1^2(x)u(x)u(x - k\Delta x)dx && \text{(G.50)} \end{aligned}$$

for $i, k = -N, \ldots, -1, 0, 1, \ldots, N$.
The function $w_1^2(x)$ in equation (G.50) is defined in equation (G.41) and the integrals' limits are taken to be $-S\Delta x$ and $+S\Delta x$, respectively. S is an integer defining the number of pixels in the vicinity of an edge that are affected by the masking effect.

Equations (G.46) to (G.50) define the desired matrices leaving two parameters: f_c and β, at our disposal as design parameters. The parameter β is the weight given to deviations in the frequency domain relative to the weight given to errors in the spatial domain. For a desired f_c, several design iterations and image quality assessments are required in order to determine β.

Appendix H

The Block Classifier

In this appendix, we describe the *block classifier*, employed within the proposed postprocessing algorithm. Basically, we have adopted the block classifier proposed in [Ramamurthi 1986a] due to its good performance and simplicity. We describe its concept here for the sake of completeness and for easy reference.

Five classes of image blocks have been defined: (*i*) a monotone class, consisting of monotone image blocks containing no significant gradient, and (*ii*) four edge classes, consisting of image blocks having a distinct edge running through them. Four edge orientations have been defined: horizontal, vertical, and two diagonals (at 45 deg and 135 deg). A complex structured image block has also been defined. This kind of block contains fine details or complex edges which can not be treated as simple edges. Since the coding quality is not high enough to preserve fine details in images, it is difficult to estimate the features of the signal spectrum correctly. As a result, smoothing the signal and the noise is perceptually preferable in such cases. Thus, if the tested block is neither monotone nor edge block it is classified as a complex structured block and assigned to the monotone class of blocks.

The block classifier is employed for determining whether the block under consideration is a monotone block or an edge block. If the block is found to be an edge block, then the edge orientation is determined. The correlation between the signal and the noise in the coded image block is critical to the accurate estimation of the edge orientation. The fact that the inclination of the staircase noise follows very closely the true orientation of the edge in the original image can be utilized in determining the edge direction in the coded image block. Thus, by analyzing a large enough neighbourhood in the coded image it is possible to detect the existence of an edge along with its direction.

The classification procedure is defined as follows. Let $I(x,y)$ denote a pixel located at (x,y) in a block of size PxP, and let $I_{ave}(x,y)$ denote the average intensity of two vertically adjacent pixels, $I(x,y)$ and $I(x+1,y)$, defined by:

$$I_{ave}(x,y) = \frac{I(x,y)+I(x+1,y)}{2} \tag{H.1}$$

The gradients in the vertical direction are defined as follows:

$$d(x,y) = I(x,y) - I(x+1,y) \; . \tag{H.2}$$

Let $d_n(x,y)$ denote the gradients in the vertical direction normalised by the average intensity of the two vertically adjacent pixels. It is defined as follows:

$$d_n(x,y) = \frac{d(x,y)}{I_{ave}(x,y)} = \frac{2[I(x,y)-I(x+1,y)]}{I(x,y)+I(x+1,y)} \; . \tag{H.3}$$

This definition is in accord with the well known fact that the sensitivity of the human visual system is proportional to the normalised gradient and not to the gradient itself [Pratt 1978].

Two counters K and L are defined and set to zero. Counter K is updated according to the following rule:

$$\begin{aligned} K &= K+1 \quad \text{if} \;\; d_n(x,y) > T \\ K &= K-1 \quad \text{if} \;\; d_n(x,y) < -T \end{aligned} \tag{H.4}$$

for $x = 1,2,\ldots,P-1$ and $y = 1,2,\ldots,P$.
The threshold T has been defined in equation (F.5) in appendix F.

Counter L is computed in a similar way with horizontally adjacent pixels, $I(x,y)$ and $I(x,y+1)$, for $x = 1,2,\ldots,P$ and $y = 1,2,\ldots,P-1$. The tested block is then assigned to one of five classes as follows:

$$\begin{array}{ll} \text{monotone/complex} & \text{if } \mid K \mid \leq m \text{ and } \mid L \mid \leq m \\ \text{edge at 0 deg} & \text{if } \mid K \mid > m \text{ and } \mid L \mid < m \\ \text{edge at 90 deg} & \text{if } \mid K \mid < m \text{ and } \mid L \mid > m \\ \text{edge at 45 deg} & \text{if } \mid K \mid > m \,, \mid L \mid > m \,, \text{ and } K \cdot L > 0 \\ \text{edge at 135 deg} & \text{if } \mid K \mid > m \,, \mid L \mid > m \,, \text{ and } K \cdot L < 0 \end{array} \tag{H.5}$$

where m is a predetermined threshold.

It should be noted that the counters are incremented when there is a positive gradient above threshold, and decremented when there is a negative gradient above threshold. Thus, gradients due to random speckle noise are neutralized and a monotone block, contaminated by speckle noise, will be classified correctly. Moreover, an L-shape edge (such as the "steps" of the staircase noise) and a diagonal edge are not distinguished by the classifier. Both types of edges have the same effect on the counters and thus the "steps" of the staircase noise do not affect the correct estimation of the edge orientation.

Bibliography

[Ahmed 1974] N.Ahmed,T.Natarajan, and K.R.Rao, "Discrete Cosine Transform," IEEE Trans. on Computers, pp. 90-93, January 1974.

[Ahmed 1982] N.Ahmed and M.D.Flicker, "A Derivation for the Discrete Cosine Transform," Proc. IEEE, vol. 70, pp. 1132-1134, Sept. 1982.

[Akansu 1992] A.N.Akansu and F.E.Wadas, "On Lapped Orthogonal Transforms," IEEE, Trans. Signal Process., vol. 40, no. 2, pp. 439-443, Feb. 1992.

[Algazi 1975] V.R.Algazi and M.Suk, "On the Frequency Weighted Least-Square Design of Finite Duration Filters," IEEE Trans. Circuits and Systems, vol. CAS-22, no. 12, pp. 943-953, December 1975.

[Algazi 1986] V.R.Algazi, "FIR Anisotropic Filters for Image Enhancement," IEEE, ICASSP 86, pp. 2471-2474, Tokyo 1986.

[Algazi 1989] V.R.Algazi,G.E.Ford and E.Hildum, "Digital Representation and Storage of High Quality Color Images by Anisotropic Enhancement and Subsampling," IEEE ICASSP 89, pp. 1846-1849, Glasgow 1989.

[Ang 1986] Y.H.Ang and T.S.Durrani, "Adaptive Transform Coding Using Vector Quantization," IEE Proc., vol. 265, 1986.

[Aravind 1986] R.Aravind and A.Gersho, "Low-Rate Image Coding With Finite-State Vector Quantization," in Proc. Int. Conf. ASSP 1986, Tokyo, Japan pp. 137-140.

[Berger 1971] T.Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression* Englewood Cliffs, N.J. Prentice-Hall, 1971.

[Breeuwer 1988] M. Breeuwer, "Transform Coding of Images Using Directionally Adaptive Vector Quantization", IEEE ICASSP 1988, pp. 788-791.

[Buzo 1980] A.Buzo, A.H.Gray, R.M.Gray and J.D.Markel, "Speech Coding Based Upon Vector Quantization", IEEE Trans. on ASSP, vol. ASSP-28, no. 5, pp. 562 - 574, October 1980.

[Carrioli 1988] L.Carrioli and M.G.Albanesi, "Multigray Image Compression Based on the Human Eye Transfer Function," Alta Frequenza (Italy), vol. 57, no. 5, pp. 273-284, June 1988.

[Cassereau 1985] P.Cassereau, "A New Class of Optimal Unitary Transforms for Image Processing," M.S. thesis, Dep. Elec. Eng. Comput. Sci., Mass. Inst. Technol. Cambridge, MA, May 1985.

[CCIR 1974] CCIR, "Method of the Subjective Assessment of the Quality of Television Pictures," 13th Plenary Assembly, Rec. 500, vol. 11, pp. 65-68, 1974.

[Chen 1977] W.H.Chen, C.H.Smith, and S.C.Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Trans. Commun. vol. COM-25, no. 9, pp. 1004-1009, September 1977.

[Chen 1977a] W.H.Chen and C.H.Smith, "Adaptive Coding of Monochrome and Color Images," IEEE Trans. Commun., vol. COM-25, pp. 1285-1292, Nov. 1977.

[Chen 1990] D.Chen and A.C.Bovik, "Visual Pattern Image Coding," IEEE, Trans. Commun., vol. 38, no. 12, pp. 2137-2145, December 1990.

[Chen 1991] S.H.Chen and W.M.Hsieh, "Fast Algorithm for VQ Codebook Design," IEE Proc., part I, vol. 138, no. 5, pp. 357-362, Oct. 1991.

[Cheng 1984] D.Y.Cheng, A.Gersho, B.Ramamurthi, and Y.Shoam, "Fast Search Algorithms for Vector Quantisation and Pattern Matching," Proc. IEEE Int. Conf. ASSP 1984, paper 9.11.

[Chin 1983] R.T.Chin and C.L.Yeh, "Quantitative Evaluation of Some Edge Preserving Noise-Smoothing Techniques," Comp. Vision,Graphics, Image Proc. 23, pp. 67-91, 1983.

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

[Daugman 1985] J.G.Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-

Dimensional Visual Cortical Filters", J. Opt. Soc. Am. A, vol. 2, no. 7, pp. 1160-1169, July 1985.

[Davidson 1986] G.Davidson and A.Gersho, "Application of a VLSI Vector Quantization Processor to Real-Time Speech Coding," IEEE J. Select. Areas Commun., vol. SAC-4, Jan. 1986.

[Davis 1987] L.S.Davis and A.Rosenfeld, "Noise Cleaning by Iterated Local Averaging," IEEE Trans. Syst. Man Cybern. SMC-8, pp. 705-710, 1987.

[Delp 1979] E.J.Delp and D.R.Mitchell, "Image Compression Using Block Truncation Coding," IEEE Trans. Commun., vol. COM-27, pp. 1335-1342, Sept. 1979.

[DePalma 1962] J.J.DePalma and E.M.Lowry, "Sine Wave Response of the Visual System.II. Sine Wave and Square Wave Contrast Sensitivity," J. Opt. Soc. Amer., vol. 52, no. 3, pp. 328-335, March 1962.

[Dowling 1966] J.E.Dowling and B.B.Boycott, "Organization of the Primate Retina: Electron Microscopy," Proc. Roy. Soc. B, 166, pp. 80-111, 1966.

[Eggerton 1986] J.D.Eggerton and M.D.Srinath, "A Visually Weighted Quantization Scheme for Image Bandwidth Compression at Low Data Rates," IEEE Trans. Commun., vol. COM-34, no. 8, pp. 840-847, August 1986.

[Equitz 1987] W.Equitz, "Fast Algorithms for Vector Quantization Picture Coding," Proc. Int. Conf. ASSP 1987, pp 18.1.1-18.1.4

[Equitz 1989] W.H.Equitz, "A New Vector Quantization Clustering Algorithm," IEEE Trans. ASSP , vol. 37, no. 10, pp. 1568-1575, October 1989.

[Fiorentini 1966] A.Fiorentini and M.T.Zoli, "Detection of a Target Superimposed to a Step Pattern of Illumination," Atti. Fond. G. Ronchi, vol. 21, pp. 338-356, 1966.

[Friedman 1975] J.H.Friedman, F.Bashett, and L.J.Shustek, "An Algorithm for Finding Nearest Neighbours," IEEE Trans. Comput., vol. C-24, pp.1000-1006, Oct. 1975.

[Fuotes 1958] M.G.F.Fuotes, "Electric Activity of Cells in the Eye of Limulus," Amer. J. Opthal., vol. 46, pp. 210-223, 1958.

[Fuotes 1959] M.G.F.Fuotes, "Initiation of Impulses in the Visual Cells of Limulus," J. Physiology (London), vol.148, pp. 14-28, 1959.

[Gallager 1968] R.G.Gallager, *Information Theory and Reliable Communication.* New York, Wiley, 1968.

[Gersho 1992] A.Gersho and R.M.Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992.

[Gersho 1982] A.Gersho and B.Ramamurthi, "Image Coding Using Vector Quantization," Proc. Int. Conf. ASSP 1982, pp. 428-231.

[Gersho 1982a] A.Gersho, "On the Structure of vector Quantizers," IEEE Trans. Inform. Theory, vol. IT-28, pp. 157-166, Mar. 1982.

[Giusto 1990] D.D.Giusto and G.Vernazza, "Color-Image Coding by an Advanced Vector Quantizer," Proc. Int. Conf. ASSP 1990, pp. 2265-2268.

[Gray 1980] R.M.Gray, J.C.Kieffer, and Y.Linde, "Locally Optimal Block Quantizer Design," Information and Control, vol. 45, pp. 178-198, 1980.

[Gray 1982] R.M.Gray and E.Karnin, "Multiple Local optima in Vector Quantizers," IEEE Trans. Inform. Theory, vol. IT-28, pp. 256-261, Mar. 1982.

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

[Hall 1977] C.F.Hall and E.L.Hall, "A Nonlinear Model for the Spatial Characteristics of the Human Visual System", IEEE Trans. Syst. Man, Cybern., vol. SMC-7, no. 3, pp. 161-170, March 1977.

[Hall 1978] C.F.Hall, "Digital Color Image Compression in a Perceptual Space", Ph.D. dissertation, Dep. Elec. Eng., Univ. Southern California, Los Angeles, January 1978.

[Hammerly 1977] J.R.Hammerly,R.F.Quick, and T.A.Reichert, "A Study of Grating Contrast Judgement", Vision Research, vol. 17, pp 201-207,1977.

[Haque 1985] M.A.Haque, "A Two-Dimensional Fast Cosine Transform," IEEE Trans. ASSP, vol. ASSP-33, no. 6, pp. 1532-1539, December 1985.

[**Hartline 1932**] H.K.Hartline and C.H.Graham, "Nerve Impulses from Single Receptors in the Eye," J. Cell. Comp. Physiol., vol. 1, pp. 277-295, 1932.

[**Hartline 1947**] H.K.Hartline and P.R.McDonald, "Light and Dark Adaptation of Single Receptor Elements in the Eye of Limulus," J. Cell. Comp. Physiol., vol. 30, pp. 225-253, 1947.

[**Hartline 1956**] H.K.Hartline, H.G.Wagner, and F.Ratliff, "Inhibition in the Eye of Limulus," J. Genl. Physiol., vol. 39, pp. 651-673, 1956.

[**Hartline 1957**] H.K.Hartline and F.Ratliff, "Inhibitory Interaction of Receptor Units in the Eye of Limulus," J. Genl. Physiol., vol. 40, pp. 357-376, 1957.

[**Harwood 1987**] D.Harwood, M.Subbarao, H.Hakalahti and L.Davis, "A New Class of Edge-Preserving Smoothing Filters," Pattern Recognition Letters, vol. 6, no. 3, pp. 155-162, August 1987.

[**Haskell 1985**] B.G.Haskell, "Image Coding by the Block List Transform," AT&T Bell Laboratories Tech. Memo no. 11350-850903-01TM, Sept. 1985.

[**Helmuth 1980**] S.Helmuth, *Cluster Analysis Algorithms for Data Reduction and Classification of Objects* Ellis Horwood Limited (1980), England.

[**Hentea 1984**] T.A.Hentea and V.R.Algazi, "Perceptual Model and the Filtering of High-Contrast Achromatic Images," IEEE trans. Syst. Man Cyber., vol. SMC-14, pp. 230-246, Mar./Apr. 1984.

[**Hinman 1984**] B.L.Hinman,J.G.Bernstein, and D.H.Staelin, "Short-Space Fourier Transform Image Processing," in Proc. Int. Conf. ASSP 1984, San Diego, CA, pp. 4.8.1-4.8.4 .

[**Ho 1989**] Yo-Sung Ho and A.Gersho, "Classified Transform Coding of Images Using Vector Quantization," Proc. ICASSP 89, pp. 1890-1893, May 1989.

[**Huang 1966**] T.S.Huang, "Digital Picture Coding," Proc. Nat. Electron. Conf., pp. 793-797, 1966.

[**Hubel 1961**] D.H.Hubel and T.N.Wiesel, "Integrative Action in Cat's Lateral Geniculate Body," J. Physiol., no. 155, pp. 385-398, 1961.

[Hubel 1979] D.H.Hubel and T.Weisel, "Brain Mechanisms of Vision," Sci. Amer., pp. 130-146, Sept. 1979.

[Huffman 1952] D.A.Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proc. IRE, vol. 40, no. 9, pp. 1098-1101, Sept. 1952.

[Ingle 1979] V.K.Ingle and J.W.Woods, "Multiple Model Recursive Estimation of Images," in Proc. Int. Conf. ASSP 1979, pp. 642-645, April 1979.

[Itakura 1968] F.Itakura and S.Saito, "Analysis Synthesis telephony Based on the Maximum Liklihood Method," Proc. 6-th Int. Cong. Acoust. (Tokyo), pp. C-17-C20, Aug. 1968.

[Kappagantula 1983] S.Kappagantula and K.R.Rao, "Motion Compensated Predictive Coding," Proc. SPIE 27th Annual Inter. Tech. Symp., San Diego, CA, Aug. 1983.

[Kaufman 1974] L.Kaufman, *Sight and Mind, An Introduction to Visual Perception*, New York: Oxford University Press, 1974.

[Kelly 1977] D.H.Kelly, "Visual Contrast Sensitivity," Opt. Acta, vol. 24, pp. 107-130, Feb. 1977.

[Keskes 1979] N.Keskes, F.Kretz, and H.Maitre, "Statistical Study of Edges in TV Pictures," IEEE Trans. Commun. vol. COM-27, no. 8, pp. 1239-1246, August 1979.

[Kim 1986] T.Kim, "New Finite-State Vector Quantizers for Images," in Proc. Int. Conf. ASSP 1986, Tokyo, Japan, pp. 1180-1183.

[Kim 1989] J.W.Kim and S.U.Lee, "Discrete Cosine Transform - Classified VQ Technique for Image Coding," IEEE ICASSP 89, pp. 1831-1834, 1989.

[Kim 1991] D.S.Kim and S.U.Lee, "Image Vector Quantizer Based on a Classification in the DCT Domain," IEEE Trans. on Cummun., vol. 39, no. 4, pp. 549-556, April 1991.

[Koh 1988] J.S.Koh and J.K.Kim, "Fast Sliding Search Algorithm for Vector Quantisation in Image Coding," Electronics Letters, vol. 24, no. 17, pp. 1082-1083, Aug. 1988.

[Kubrick 1990] A.Kubrick and T.Ellis, "Classified Vector Quantisation of Images: Codebook Design Algorithm," IEE Proc., vol. 137, Pt. I, no. 6, pp. 379 - 386, December 1990.

[Kuffler 1976] S.W.Kuffler and J.G.Nicholls, *From Neuron to Brain*, Sunderland, MA: Sinauer Assoc., 1976.

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

[Lee 1983] J.S.Lee, "Digital Image Smoothing and the Sigma Filter," Comp. Vision, Graphics Image Proc. 24, pp. 255-269. 1983.

[Lee 1984] B.G.Lee, "FCT - A Fast Cosine Transform," Proc. IEEE ICASSP 1984, pp. 28A.3.1-28A.3.4.

[Levy 1970] L.Levy, "Vision in Communication," in Progress in Optics, vol. 8, E.Wolf, Ed. New York: Elsevier, 1970, sect. 7, pp. 345-374.

[Limb 1977] J.O.Limb and C.B.Rubinstein, "A Model of Threshold Vision Incorporating Inhomogeneity of the Visual Field", Vision Research, vol. 17, pp. 571-584, April 1977.

[Limb 1979] J.O.Limb, "Distortion Criteria of the Human Viewer," IEEE, Trans. Syst. Man, Cybern., vol. SMC-9, no. 12, pp 778-793, December 1979.

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

[Lippmann 1987] R.P.Lippmann, "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, pp. 4-22, Apr. 1987.

[Lukas 1980] F.X.Lukas, U.Tulanay-Keesey, and J.O.Limb, "Thresholds at Luminance Edges Under Stabilized Viewing Conditions," J. Opt. Soc. Amer., vol. 70, pp. 418-422, April 1980.

[Lukas 1982] F.X.Lukas and Z.L.Budrikis, "Picture Quality Prediction Based on Visual Model," IEEE, Trans. Commun. vol COM-30, no. 7, pp. 1679-1692 , July 1982.

[Makhoul 1980] J.Makhoul, "A Fast Cosine Transform in One and Two Dimensions", IEEE Trans. ASSP, vol. ASSP-28, no. 1, pp. 27-34, Feb. 1980.

[Malvar 1987] H.S.Malvar, "A Pre and Post-Filtering Technique for The Reduction of Blocking Effects," presented at the Picture Coding Symp., Stockholm, Sweden, June 1987.

[Malvar 1989] H.S.Malvar and D.H.Staelin, "The LOT: Transform Coding Without Blocking Effects," IEEE Trans. ASSP, vol. 37, no. 4, pp. 553-559, April 1989.

[Mannos 1974] J.L.Mannos and D.J.Sakrison, "The Effects of a Visual Fidelity Criterion on the Encoding of Images," IEEE Trans. Inform. Theory, vol. IT-20, no. 4, pp. 525-536, July 1974.

[Marescq 1986] J.P.Marescq and C.Labit, "Vector Quantization in Transformed Image Coding," IEEE ICASSP 1986, Tokyo, pp. 145-148.

[Max 1960] J.Max, "Quantization for Minimum Distortion," IRE Trans. Inform. Theory, vol. IT-6, pp. 7-12, March 1960.

[McClellan 1973] J.H.McClellan, "The Design of Two-dimensional Digital Filters by Transformations," in Proc. 7th Annu. Princeton Conf. Inform. Sci. and Syst., pp. 247-251, March 1973.

[McLaren 1991] D.L.McLaren and D.T.Nguyen, "Removal of Subjective Redundancy from DCT-Coded Images," IEE Proc., Pt. I, vol. 138, no. 5, pp. 345-350, October 1991.

[Miyahara 1985] M.Miyahara and K.Kotani, "Block Distortion in Orthogonal Transform Coding - Analysis, Minimization, and Distortion Measure," IEEE Trans. Commun. vol. COM-33, no. 1, pp. 90-96, January 1985.

[Nasrabadi 1988] N.M.Nasrabadi and R.A.King, "Image Coding Using Vector Quantization A Review," IEEE Trans. Commun. vol COM-36, no. 8, pp. 957-971, August 1988.

[Netravali 1977] A.N.Netravali and B.Prasada, "Adaptive Quantization of Picture Signals Using Spatial Masking," Proc. IEEE, vol. 65, no. 4, pp. 536-548, April 1977.

[**Netravali 1979**] A.N.Netravali and J.O.Robbins, "Motion Compensation Television Coding: Part I," BSTJ vol. 58, pp. 631-670, Mar. 1979.

[**Newman 1973**] T.G.Newman and H.Dirilten, "A Nonlinear Transformation for Digital Picture Processing," IEEE Trans. on Comput. vol. C-22, pp. 869-873, Sept. 1973.

[**Ngan 1982**] K.N.Ngan, "Adaptive Transform Coding of Video Signals", IEE Proc., vol. 129, Pt. F, no. 1, pp 28-40, February 1982.

[**Ngan 1989**] K.N.Ngan, K.S.Leong and H.Singh, "Adaptive Cosine Transform Coding of Images in Perceptual Domain," IEEE Trans. ASSP, vol. 37, no. 11, pp. 1743-1749, November 1989.

[**Ni 1985**] L.M.Ni and A.K.Jain, "A VLSI Systolic Architecture for Pattern Clustering," IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-7, pp. 80-89, Jan. 1985.

[**Nill 1985**] N.B.Nill, "A Visual Model Weighted Cosine Transform for Image Compression and Quality Assessment," IEEE Trans. Commun. vol COM-33, no. 6 pp. 551-557, June 1985.

[**Norman 1974**] R.A.Norman and F.S.Werblin, "Control of Retinal Sensitivity - I. Light and Dark Adaptation of Vertebrate Rods and Cones," J. Gen. Physiol., vol. 63, pp. 37-61, 1974.

[**Pearson 1984**] D.E.Pearson and M.W.Whybray, "Transform Coding of Images Using Interleaved Blocks," IEE Proc. Part F, vol. 131, pp. 466-472, no. 5, August 1984.

[**Perkins 1988**] M.G.Perkins, "A Comparison of the Hartley, Cas-Cas, Fourier, and Discrete Cosine Transforms for Image Coding," IEEE Trans., Commun., vol. COM-36, pp. 758-761, 1988

[**Pratt 1978**] W.K.Pratt, *Digital Image Processing*, New York, Wiley, 1978.

[**Prosser 1964**] R.D.Prosser,J.W.Allnatt and N.W.Lewis, "Quality Grading of Impaired Television Pictures", Proc. IEE, vol. 111, no. 3, March 1964.

[**Ramamurthi 1985**] B.Ramamurthi, *Vector Quantization of Images Based on A Composite Source Model*, Ph.D. dissertation, University of California, Santa Barbara, U.S.A

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

[Ramamurthi 1986a] B.Ramamurthi and A.Gersho, "Nonlinear Space-Variant Postprocessing of Block Coded Images," IEEE Trans. ASSP, vol. ASSP-34, no. 5, pp. 1258-1267, October 1986.

[Ran 1990] X.Ran and N.Farvardin, "Combined VQ-DCT Coding of Images Using Interblock Noiseless Coding," Proc. Int. Conf. ASSP 1990, pp. 2281-2284.

[Ratliff 1965] F.Ratliff, *Mach Bands: Quantitative Studies on Neural Networks in the Retina*, Holden-Day, San Fransisco, 1965.

[Reeve 1983] H.C.Reeve and J.S.Lim, "Reduction of Blocking Effect in Image Coding," Proc. Int. Conf. ASSP 1983 Boston, pp. 1212-1215.

[Reininger 1983] R.C.Reininger and J.D.Gibson, "Distributions of Two-Dimensional DCT Coefficients for Images," IEEE Trans. Commun. vol. COM-31, no. 6, pp. 835-839, June 1983.

[Rocca 1972] F.Rocca, "Television Bandwidth Compression Utilizing Frame-to-Frame Correlation and Movement Compensation," *Picture Bandwidth Compression*, pp. 675-693, Gordon and Breach, 1972.

[Rosenfeld 1976] A.Rosenfeld and A.C.Kak, *Digital Picture Processing*, vol. 1, Academic Press, New York, 1976.

[Rushtone 1961] W.A.H.Rushtone, "Peripheral Coding in the Nervous System," in Sensory Communication, W.A.Rosenblith, Ed. Cambridge, MA; MIT Press, pp. 169-181, 1961.

[Ryan 1988] T.W.Ryan and C.A.Cotter, "Vector Quantization Training by Self-Organizing Neural Network," Proc. SPIE, vol. 924, pp. 312-320, 1988

[Sabin 1984] M.J.Sabin and R.M.Gray, "Product Code Vector Quantizers for Waveform and Voice Coding", IEEE Trans. on ASSP, vol. ASSP-32, no. 3, pp. 474 - 488, June 1984.

[Sabin 1986] M.J.Sabin and R.M.Gray, "Global Convergence and Empirical Consistency of the Generalized Lloyd Algorithm," IEEE Trans. on Inform. Theory, vol IT-32 pp 148-155, March 1986.

[Saito 1987] T.Saito, H.Takeo, K.Aizawa, H.Harashima, and H. Miyakawa, "Discrete Cosine Transform Coding with Gain/Shape Vector Quantizer for Picture Signals," Elec. and Commun. in Japan, Pt. 1, vol. 70, no. 7, pp. 78-87, 1987.

[Saito 1988] T.Saito, H.Takeo, K.Aizawa, and H.Harashima, "Adaptive Discrete Cosine Transform Coding with Gain/Shape Vector Quantizers for Picture Signals," Elec. and Commun. in Japan, Pt. 1, vol. 71, no. 6, pp. 45-56, 1988.

[Sakrison 1975] D.J.Sakrison, M.C.Halter and H.Mostafavi, "Properties of the Human Visual System as Related to the Encoding of Images," in New Directions in Signal Processing in Communication and Control, J.K.Skwirzynski (Ed.), Noordhoff, Leyden, pp. 159-177, 1975.

[Sakrison 1977] D.J.Sakrison, "On the Role of the Observer and a Distortion Measure in Image Transmission," IEEE Trans. Commun., vol. COM-25, no. 11, pp. 1251-1267, November 1977.

[Sauer 1988] K.Sauer, "Postprocessing for Edge Enhancement in Low Bit Rate Coded Images," in SPIE, vol. 1001, Visual Commun. Image Process. pp. 658-665, 1988.

[Schreiber 1967] W.F.Schreiber, "Picture Coding," Proc. IEEE (Special Issue on Redundancy Reduction), vol. 55, pp. 320-330, March 1967.

[Sethi 1981] I.K.Sethi, "A Fast Algorithm for Recognizing Nearest Neighbours," IEEE Trans. Syst., Man, Cybern., vol. SMC-11, no. 3, pp. 245-248, Mar. 1981.

[Shanmugan 1975] K.S.Shanmugan, "Comments on Discrete Cosine Transform," IEEE Trans. on Computers, vol. C-24, p. 759, July 1975.

[Shannon 1948] C.E.Shannon, "A Mathematical Theory of Communication," Bell Syst. Tech. J. vol. 27, pp. 379-423, 623-656, 1948.

[Soleymani 1987] M.B.Soleymani and S.D.Morgera, "An Efficient Nearest Neighbour Search Method," IEEE Trans. Commun., vol. COM-35, no. 6, pp. 677-679, June 1987.

[Stevens 1970] S.S.Stevens, "Neural Events and the Psychophysical Law," Science, vol. 170, pp. 1043-1050, 1970.

[Stevens 1975] S.S.Stevens, *Psychophysics*, New York: Wiley, 1975.

[Stockham 1972] T.G.Stockham, Jr., "Image Processing in the Context of a Visual Model," Proc. IEEE, vol. 60, pp. 828-842, 1972.

[Stromeyer 1972] C.F.Stromeyer and B.Julesz, "Spatial-Frequency Masking in Vision: Critical Bands and Spread of Masking," J. Opt. Soc. Amer., vol. 62, pp. 1221-1232, 1972.

[Sun 1985] H.F.Sun and M.Goldberg, "Adaptive Vector Quantization for Image Sequence Coding," Proc. ICASSP 85, pp. 339-342, 1985.

[Tu 1987] G.Tu, G.Verbeek, J.Rommelaere and A.Oosterlinck, "A Hybrid Coding Scheme for Still Images: Adaptive Vector Quantization with Correction Information in DCT Domain," SPIE vol. 829, Application of Digital Image Processing X, San Diego, CA, 1987 pp. 119-124.

[Utall 1973] W.R.Utall, *The Psychobiology of Sensory Coding*, New York: Harper & Row, 1973.

[Vaisey 1988] J.Vaisey and A.Gersho, "Simulated Annealing and Codebook Design," Proc. Int. Conf. ASSP 1988, pp. 1176-1179.

[Wald 1965] G.Wald and P.K.Brown, "Human color vision and color blindness", Symposia on Quantitative Biology, vol. XXX, pp. 345-359, The Cold Spring Harbor Laboratory of Quantitative Biology, New York, 1965.

[Wang 1981] D.C.Wang, A.H.Vagnucci and C.C.Li, "Gradient Inverse Weighted Smoothing Schemes and the Evaluation of Its Performance," Comput. Graph. and Image Process. no. 15, pp. 167-181, 1981.

[Watanabe 1968] A.Watanabe, T.Mori, S.Nagata and K.Hiwatashi, "Spatial Sine-Wave Responses of the Human Visual System," Vision Research, Vol. 8, pp. 1245-1263, 1968.

[Werblin 1973] F.S.Werblin, "The Control of Sensitivity in the Retina," Sci. Amer., vol. 228, pp. 71-79, 1973.

[Wintz 1972] P.A.Wintz, "Transform Picture Coding," Proc. IEEE, vol. 60, pp. 809-823, July 1972.

[Yuan 1988] G.Yuan and M.Goldenberg, "A Sequential Initialization Technique for Vector Quantizer Design," Pattern Recognition Letters, pp. 157-161, Mar. 1988.

[**Zeger 1992**] K.Zeger, J.Vaisey, and A.Gersho, "Globally Optimal Vector Quantizer Design by Stochastics Relaxation," IEEE Trans. Signal Proc., vol. 40, pp. 310-322, Feb. 1992.