ADAPTIVE PREDICTION AND VECTOR QUANTIZATION BASED VERY LOW BIT RATE VIDEO CODEC

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Şennur Ulukuş
July, 1993

ADAPTIVE PREDICTION AND VECTOR QUANTIZATION BASED VERY LOW BIT RATE VIDEO CODEC

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Şennur Ulukuş

July, 1993

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Levent Onural
and
Assoc. Prof. A. Enis Çetin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Erdal Arıkan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assis. Prof. Haldun Özaktaş

Approved for the Institute of Engineering and Sciences:

Prof. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

ADAPTIVE PREDICTION AND VECTOR QUANTIZATION BASED VERY LOW BIT RATE VIDEO CODEC

Şennur Ulukuş
M.S. in Electrical and Electronics Engineering
Supervisors: Assoc. Prof. Levent Onural
and
Assoc. Prof. A. Enis Çetin
July, 1993

A very low bit rate video codec (coder/decoder) based on motion compensation, adaptive prediction, vector quantization (VQ) and entropy coding, and a new prediction scheme based on Gibbs random field (GRF) model are presented. The codec is specifically designed for the video-phone application for which the main constraint is to transmit the coded bit stream via the existing telephone lines. Proposed codec can operate in the transmission bit rate interval ranging from 8 to 32 Kbits/s which is defined as the very low bit rates for video coding. Four different coding strategies are adapted to the system, and depending on the characteristics of the image data in the block one of these coding methods is chosen by the coder. Linear prediction is implemented in the codec, and the performances of the two prediction schemes are compared at several transmission bit rates. The need for any prediction is also questioned, by implementing the same codec structure without prediction and comparing the performances of the codecs with prediction and without prediction. It is proved that the presented codec can be used in transmitting the video signal via the existing telephone network for the video-phone applications. Also, it is observed that the codec with GRF model based non-linear prediction has a better performance compared to the codec with linear prediction.

Keywords : Codec,vector quantization, prediction, Gibbs random field.

ÖZET

UYARLANIR ÖNGÖRÜ VE VEKTÖR BASAMAKLANDIRMAYA DAYALI ÇOK DÜŞÜK VERİ İLETİM HIZLARINDA ÇALIŞAN VIDEO KODLAYICI/KOD-ÇÖZÜCÜ ÇİFTİ

Şennur Ulukuş
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Levent Onural
ve
Doç. Dr. A. Enis Çetin
Temmuz, 1993

Hareket yoketme, uyarlanır öngörü, vektör basamaklandırma ve entropi kodlamaya dayalı, çok düşük veri iletim hızlarında çalışan bir video kodlayıcı/kod-çözücü çifti (KKÇ) ve Gibbs rastgele alan modeline dayalı yeni bir öngörü yöntemi sunulmaktadır. KKÇ varolan telefon hatlarından görüntü iletmeyi gerektiren görüntülü telefon uygulaması için tasarlanmıştır. Önerilen KKÇ görüntü kodlamada çok düşük iletim hızları olarak tanımlanan 8-32 Kbits/s arasında çalışabilmektedir. Görüntü parçalara bölünür ve bir karedeki her parça diğerlerinden bağımsız olarak kodlanır. Dört değişik kodlama yöntemi sistem içinde kullanılmaktadır. Görüntü parçasının karakterine göre bu dört yöntemden birinin kullanılmasına kodlayıcı tarafından karar verilir. Gibbs rastgele alan modeline dayalı öngörücü doğrusal öngörücü ile değiştirilerek KKÇ'nin başarımındaki değişiklik incelenmiştir. Ayrıca, herhangi bir öngörüye olan gereksinim de öngörü kullanmayan bir KKÇ gercekleyip, başarımını öngörüye dayalı KKÇ'lerin başarımları ile karşılaştırarak incelenmiştir. Bu çalışmada, sunulan KKÇ'nin varolan telefon hatlarından görüntü sinyali iletiminde kullanılabileceği gösterilmiştir. Ayrıca, Gibbs rastgele alan modeline dayalı öngörüyü kullanan KKÇ'nin doğrusal öngörüyü kullanan KKÇ'ye göre daha başarılı olduğu gözlemlenmiştir.

Anahtar Kelimeler : Kodlayıcı/kod-çozücü çifti, vektör basanaklandırma, öngörü, Gibbs rastgele alanı.

ACKNOWLEDGEMENT

I am indebted to Assoc. Prof. Levent Onural and Assoc. Prof. A. Enis Çetin for their supervision, guidance, suggestions and encouragement throughout the development of this thesis.

I want to express my thanks to Assoc. Prof. Erdal Arıkan and Assis. Prof. Haldun Özaktaş for their efforts in reading this thesis and valuable comments.

My special thanks are due to Engin Erzin, Gözde Bozdağı, M. Bilge Alp, Ogan Ocalı, Roy Mickos and Levent Öktem for their help and invaluable discussions.

I would like to extend my thanks to Burcu Tomurcuk Günay for her love and encouragement especially at times of despair and hardship. Also, I am grateful to my lovely cat Temir for giving me joy and happiness.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The most common way of representing an image is to use 8 bit PCM coding. This provides high quality images, and is straightforward to code and decode. It does however require a large amount of bits. In almost all image processing applications, the major objective is to reduce the amount of bits to represent the image. Image coding has two major application areas. One is the reduction of storage requirements. Examples of this application include reduction in the storage of image data from space programs (satellite images, weather maps, etc.), from medicine (computer tomography, magnetic resonance imaging, digital radiology images); and of video data in digital VCRs and motion pictures. The other application is the reduction of channel bandwidth required for image transmission systems. One reason for this is to increase the number of transmission channels as much as possible. The most well-known example of that kind of application is the digital television. Other reason is the lack of suitable transmission medium in some video coding applications. The latter reason applies to cases of video-phone, tele-conferencing and facsimile. In all of these application areas a coded bit-stream is desired to be transmitted via the existing telephone lines. Therefore the capacity of the existing telephone network puts the main constraint on the transmission bit rate.

In this thesis, research is concentrated in general on *Very Low Bit Rate* (VLBR) video coding whose application areas include the video-phone and tele-conferencing. In VLBR video coding, by very low bit rates, transmission rates from 8 to 32 Kbits/s are meant. This interval includes the transmission bit rates supported by the existing telephone lines, which may go approximately up to 20 Kbits/s.

Extensive research is going on in video-phone applications. The ongoing trend shows that, for video-phone application the image format will be standardized in the Quarter Common Intermediate Format (QCIF) [1]. QCIF has Y,U,V representation for color images. Y component has dimensions 144x176; U and V components have dimensions 72x88 and each pixel is represented by 8 bits. Also, the expected standardization for the frame rate is between 5.33 Hz and 8.00 Hz. Therefore, the raw data rate for color QCIF image ranges between 1,621 Kbits/s (for 5.33 Hz frame rate) and 2,433 Kbits/s (for 8.00 Hz frame rate). In order to reach the defined interval of VLBRs, a compression ratio between 50 (transmission rate 32 Kbits/s and frame rate 5.33 Hz) and 300 (transmission rate 8 Kbits/s and frame rate 8.00 Hz) is required. Hence compression ratios that are larger than what the traditional coding techniques can achieve are needed for the video-phone application.

Several image and/or video coding algorithms for reducing the amount of data to be stored and/or transmitted are presented in the literature. In fact, the simplest and most dramatic form of data compression is the sampling of band-limited images, where an infinite number of pixels for unit area is reduced to one sample without any loss of information, which is usually called *pulse code modulation* (PCM) [2] [3] [4]. Although PCM is nothing but a waveform sampler followed by an amplitude quantizer, it is the best established, the most implemented and the most widely used digital coding system. PCM is used either by itself (i.e., storage of images of objects with historical value that no longer exists) or in combination with other methods. In fact all waveform coders involve stages of PCM coding and decoding. In PCM image coding usually 8 bits/sample is preferred.

More complicated image coding techniques constitute a category called *predictive coding* which exploits the *redundancy* in the digital image data. Redundancy is a characteristic related to such factors as *predictability*, *randomness* and *smoothness* in the data. For example an image which has the same value everywhere is fully predictable once the pixel intensity value is known at one point of the image. Although it requires a large number of bits to represent an image in its raw format, after the reduction of the redundancy in the image, very little information is necessary to get a duplicate of the image at the decoder. On the other hand a white noise like image is not predictable at all, and all of the pixel values must be known to represent the image. Techniques such as *differential pulse code modulation* (DPCM) [5] [6] [8] and *delta modulation* (DM) [9] [10] are the examples of predictive coding schemes, which usually achieve representation (bits per sample, R) $1 < R < 4$ bits/sample In DPCM, linear prediction with constant prediction coefficients is used to take advantage of the interpixel redundancy, and prediction error at each pixel site is scalarly quantized and transmitted. A slightly complicated version of DPCM is the *adaptive DPCM* which is abbreviated as ADPCM [11] [12]. In ADPCM prediction parameters and/or quantizer characteristics are updated adaptively. DM is an important sub-class of DPCM with 1-bit prediction error quantizer.

Another class of image coding methods constitute a category called *delayed decision coding* (DDC) [13] [14], which employs encoding delays to provide run-length measurements, sub-band filtering and linear transformations to achieve $R \leq 1$ bit/sample. This kind of coding strategies include codebook, tree and trellis coding algorithms. An example for codebook coding algorithms is *vector quantization* (VQ) [15] [16]. Fractional bit rates in the range $0 < R \leq 1$ always remind an application in variable rate coding and embedded coding systems for very low bit rate coding. DDC algorithms are examples of coding with multipath search that identifies the best possible output sequence out of a set of alternatives, while the conventional coders are based on a single-path search characterized by a series of instantaneous and irrevocable choice for the component samples of the output sequence. Multipath search shows itself either coding with memory (i.e., values of the past samples are important in coding of the current sample) or coding with selection of the best matching sample value by a search of all possible values. DDC category includes *run-length coding* [17] [18], which is a time domain coding scheme that exploits the redundancy in the form of repetitions in the input sequence, *sub-band coding* [19] [20] [21] which is a frequency domain coding method that exploits the distribution of the signal energy on the frequency band. There is another class of DDC schemes which are called *transform coding* (TC) [22] [23], in which linear transformations are used to transform the image intensity signal into another domain (usually called the transform domain) where the coding process is easier and/or more efficient. Transforms are useful in concentrating the signal energy in a smaller region in the transform domain. The most well-known linear transformation is the *discrete cosine transform* (DCT) [24] [25] [26] [27]. DCT is used in many image/video compression algorithms, including H.261, JPEG, and MPEG.

There are other image compression methods which are generalizations or combinations of the methods mentioned above. Combining two or more coding methods in a single coder-decoder pair (codec) is usually called *hybrid coding.* In hybrid coding, regions of the image signal with different characteristics are coded using different coding strategies which are more appropriate. For example in the study of Maeng and Hein [28], sub-band coding, transform coding and VQ are used in combination. In that study, the incoming image is split into two bands in the frequency domain. The low frequency band is coded by 8x8 DCT with motion compensated inter-frame coding and the high band is coded by 4x4 VQ. Ghavari [29] used a hybrid scheme in which sub-band, DCT and DPCM based coding methods are used. Block truncation coding (BTC), VQ and DCT are used in combination in a hybrid video codec proposed by Wu and Coll [30]. Above three researches are examples of combining several methods to construct a hybrid coder that is superior in some sense to the coding methods constituting it. A good example of generalization of a coding method to obtain a hybrid coder is given by Ozturk and Abut [31], in which linear prediction is generalized to predicting blocks of the image using some neighboring blocks. Also in that study residual block (prediction error block) is coded by VQ.

Some video coding algorithms are specifically designed for VLBR video coding. Examples of VLBR video codecs include the one developed by Manikopoulos et. al. [32], and based on vector quantization operating on temporal domain and intraframe finite state vector quantization with state label entropy encoding. In that study 20 Kbits/s is achieved with QCIF color images and frame rate being 6 Hz with an average SNR of (average of Y, U, and V components) 32 dB. Another example is due to Schiller and Chaudhuri [33], in which a hybrid scheme based on prediction, motion compensation and DCT is entertained. The primary aim is to investigate algorithms to efficiently code all side informations so that more bits are available for transform coefficients. This codec works at 64 Kbits/s with color CIF images and 10 Hz frame rate. By a simple conversion it can be seen that this codec operates at 10 Kbits/s with QCIF color images and 6 Hz frame rate. The average SNR achieved by this codec at the mentioned bit rate is about 32 dB. Also there is a project of European countries called COST211 in which a standardization for video-phone is aimed. At each meeting they determine the new path towards the standardization and report the best performing coding algorithm proposed at the meeting. The latest VLBR codec of COST211 project is called COST-SIM2 [34]. It makes use of motion compensation, DCT and entropy coding. COST-SIM2 codec can operate at VLBRs (i.e., 8-32 Kbits/s) and it achieves an average SNR of 32 at 8 Kbits/s and 33 at 16 Kbits/s. Since COST-SIM2 codec is the best performing of all VLBR codecs proposed up to the time this thesis is prepared and since it is implemented in the computer used during this thesis it will be used in evaluation of the performance of the codec presented in this thesis, by comparing the SNR and subjective qualities of the sequences decoded by the two methods. Also, there are commercial video-phones produced by AT&T, British Telecom / Marconi, COMTECH Labs, and ShareVision. But the details of the coding methods are secure for these commercial video-phones. The only thing known is that they all use DCT based algorithms.

In this thesis, a hybrid coder-decoder pair is presented. The codec uses motion compensation, prediction, vector quantization, and Huffman coding. In fact, sub-band coding is also implemented by activating motion compensation. In motion compensation, a motion vector along with a motion compensation error (usually called displaced frame difference value) is determined. The magnitude of motion compensation error shows the degree of temporal activity. Larger motion compensation is a sign of larger temporal activity, and similarly smaller motion compensation error is a sign of smaller temporal activity. The degree of temporal activity of video signal is divided into four bands (0,1,2,3) and each band is coded using a different strategy that is suitable to the characteristic of the signal in that band. Image is divided into blocks. During the motion compensation process $(0,0)$ motion vector is favored if there is not significant reconstruction error. Blocks with least temporal activity are not coded at all. (i.e., the motion vector is $(0,0)$ and no significant residual data exists in the block) Motion compensation plays the role of the temporal prediction. In the blocks with a little bit more temporal activity, only the temporal prediction (motion compensation) is used and the prediction error is ignored. Image blocks with medium temporal activity are coded using motion information in combination with VQ. Motion compensated block difference (or residual block signal after temporal prediction) is vector quantized. Motion compensation increases the efficiency of the vector quantization based coding algorithm by decreasing the correlation between the signal to be coded and the original image signal. Signal to be vector quantized is noise-like in amplitude and has correlations (showing the edge map of the original image) in spatial domain. Predictable part of the signal is not vector quantized, but it is transmitted (coded) using motion vector. Because if predictable part is added to the signal that is vector quantized, dimensionality of the vector quantizer will increase making it less efficient. Vector quantizer codebook includes only the temporally unpredictable part of the image signal. If it included the image signal itself (without extracting the predictable part), when an image block which was not in the training sequence was encountered, vector quantizer would yield a totally unrelated output to the input block. Therefore, prediction (temporal) in combination with VQ makes the coding scheme more robust. After all, it is not meaningful to have temporal correlations in the vector quantizer codebook, since temporal correlations are intrinsic in the motion vector which is transmitted for every block of type 2. Image blocks with highest temporal activity are coded using spatial prediction accompanied with VQ. Temporal prediction (i.e., motion compensation) is not used at all. Spatial prediction is implemented in order to remove interpixel correlations in the block, and the resulting residual image block is vector quantized. Spatial prediction also makes the VQ based coding scheme more effective, by removing the signal dependency of the VQ codebook, therefore making the coding process more robust. VQ codebook used in the coding of type 3 blocks contains codewords which have some edge structures in different directions.

Two types of prediction, the linear prediction and the Gibbs random field (GRF) model based non-linear prediction methods, are implemented in the codec structure and their effects on the codec performance are examined.

Prediction is adaptive in the sense that at each block of the image, prediction parameters are updated. Prediction parameters represent the waveform characteristics of the image signal in the block. Three prediction parameters are used both in linear and non-linear prediction cases, for each block; and they are vector quantized during the coding process. Only in-block prediction is implemented in order to prevent the accumulation of error in the prediction. Also, by forcing the prediction to be constraint in the block, no information flow (transfer) is needed between the blocks. This lets independent coding of each block.

This thesis is organized as follows. In Chapter 2 and Chapter 3, structures and the activities of the coder and the decoder are presented. In Chapter 4, linear and GRF model based non-linear prediction methods are investigated in detail. Chapter 5, first introduces Linde-Buzo-Gray (LBG) vector quantizer design algorithm and then explains the construction of the training sequence to be used in the vector quantizer design. In Chapter 6, second level coding of the extracted parameters during the coding process is given. Chapter 7 gives the details of the experiments conducted. Performance features of the codec presented in this thesis, with linear prediction, GRF model based non-linear prediction and no prediction are reported in Chapter 7. Also the presented codec is compared with the COST-SIM2 codec in terms of SNR (signal to noise ratio) in that chapter. Chapter 8 includes the conclusions of this thesis and comments about the results of the this research.

Chapter 2

CODER STRUCTURE

The block diagram of the coder is shown in Fig2.1. At the coder the original image is first divided into 8x8 blocks. Motion compensation is used in order to exploit the redundancy along the temporal direction. The motion vectors are found using the *block-matching method* [35]. For each 8x8 block a motion vector, whose two integer components range from -7 to $+7$, is determined.

There are a total of four different coding strategies used in this coding system. Each block is coded independently of other blocks in the frame. A decision about one of the four coding methods is taken by looking at the value of the Displaced Frame Difference (DFD) for the block. After the motion compensation is applied, DFD of the (k,l)'th block with motion vector $\underline{u} = (u_x, u_y)$, $DFD_{kl}(u_x, u_y)$, is determined by using 2.1,

$$DFD_{kl}(u_x, u_y) = \sum_{i=8k}^{8(k+1)-1} \sum_{j=8l}^{8(l+1)-1} |X_n(i,j) - X_{n-1}(i-u_x, j-u_y)| \qquad (2.1)$$

where $X(.,.)$ represents the image intensity and the subscript n shows the frame number of the image.

Three experimentally chosen thresholds (T_m, T_1 and T_2) are used to determine the type of the coding scheme for the 8x8 image blocks.

0. *No coding :*
   $(DFD_{kl}(u_x, u_y) < T_1$ and $| \; DFD_{kl}(u_x, u_y) - DFD_{kl}(0,0) \; | \leq T_m)$

This case occurs when there is almost perfect matching in the motion compensation, and there is not much reconstruction error if $(0,0)$ motion vector is sent in stead of the real motion vector. Usually this type of image blocks are encountered in the background of the images.

1. *Coding using only the motion information :*
   $(DFD_{kl}(u_x, u_y) < T_1$ and $\mid DFD_{kl}(u_x, u_y) - DFD_{kl}(0,0) \mid > T_m)$
   This case also occurs when there is almost perfect matching in the motion compensation and there is a significant advantage in sending the real motion vector instead of the $(0,0)$ motion vector. Usually this type of blocks are encountered on the hair of the speaker.

2. *Coding using motion information in combination with VQ :*
   $(T_1 \leq DFD_{kl}(u_x, u_y) < T_2)$
   In this case the DFD is larger. Motion compensated error signal is vector quantized. This motion information provides the temporal prediction of the block. The motion data as well as the temporal prediction error are required at the receiver to reconstruct the block.

3. *Predictive coding with VQ :*
   $(T_2 < DFD_{kl}(u_x, u_y))$
   In this case the DFD is quite large, i.e., these parts of the image have the highest temporal activity. Since temporal prediction proves to be useless, motion information is not used at all. Image intensity signal itself is coded using spatial prediction and VQ. Prediction error for each block is vector quantized.

At every block of the image, first the decision parameter about the coding strategy used for that block is sent by the coder to the decoder.

If the current block is of type 0, nothing else is transmitted other than the decision parameter.

If the current block is of type 1, only the motion vector is sent by the coder to the decoder, in combination with the decision parameter.

If the current block is of type 2, motion compensated error block is vector quantized. Vector quantizer codebook (Codebook I) is searched for the codeword that is most similar to the original motion compensated image block. Similarity criterion used by the vector quantizer is given as follows,

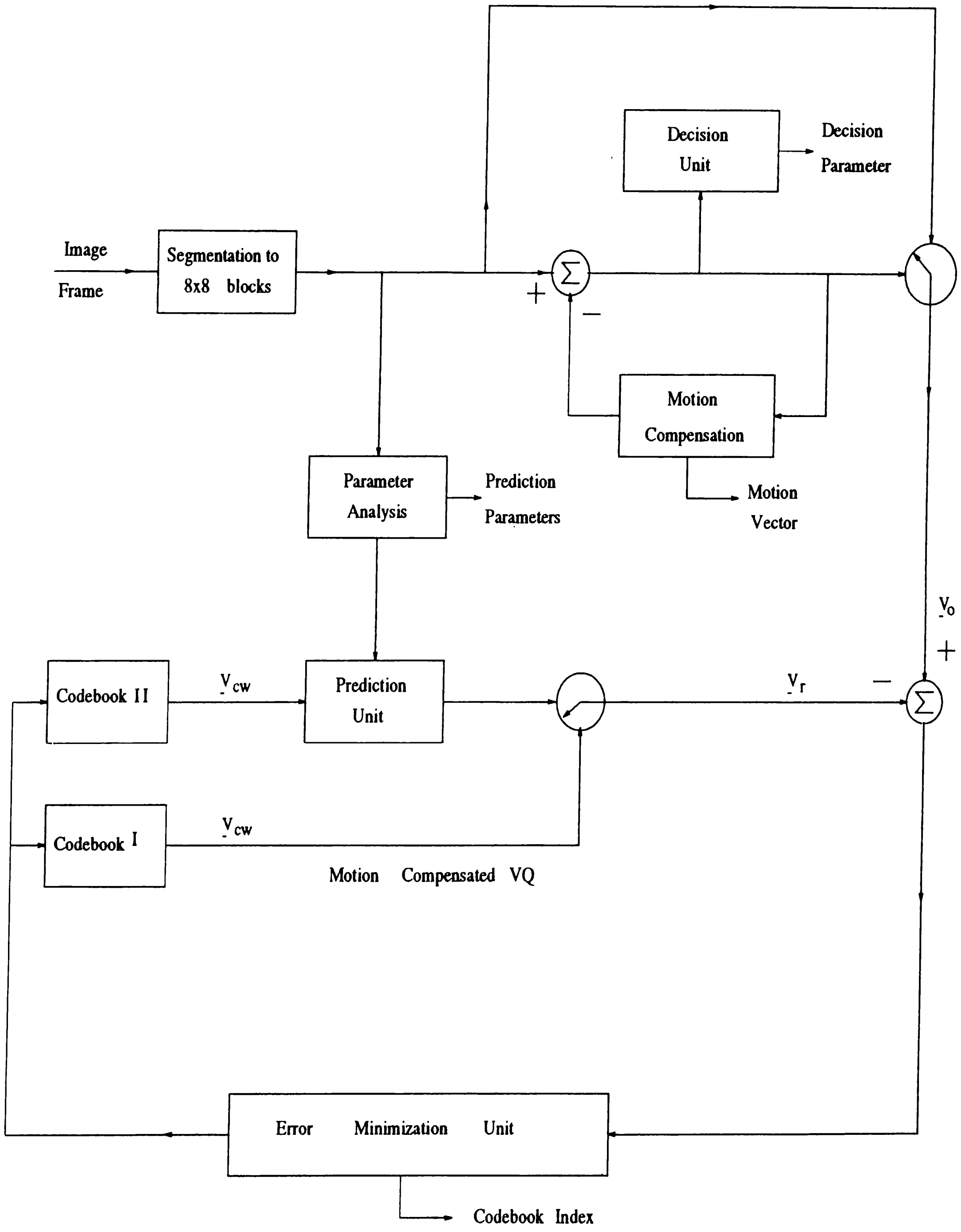

Figure 2.1: Coder structure.

$$\sum_{i=0}^{7}\sum_{j=0}^{7}|\underline{v}_{cw}(i,j) - \underline{v}_o(i,j)| \tag{2.2}$$

where $\underline{v}_{cw}$ and $\underline{v}_o$ shows the codeword read from Codebook I and the original motion compensated image block, respectively. Index of the codeword that minimizes the difference measure in 2.2, and the motion vector are the parameters that are transmitted with the decision parameter.

If the current block is of type 3, then the prediction parameters are calculated using the image data in the block. These parameters are fed into the predictor. The predictor, using the prediction parameters, constructs the prediction filter. Afterwards, the best-matching codeword is searched from the prediction based vector quantizer codebook (Codebook II) which minimizes the difference between the original image block, $\underline{v}_o$, and the reconstructed image block, $\underline{v}_r$. The reconstructed block, $\underline{v}_r$, is obtained using,

$$\underline{v}_r(i,j) = \underline{v}_p(i,j) + \underline{v}_{cw}(i,j) \tag{2.3}$$

where $\underline{v}_p$ represents the predicted block, and $\underline{v}_{cw}$ represents the codeword that plays the role of an error signal between the predicted and the original pixel values. The error criterion between the original and the reconstructed image blocks is chosen to be the sum of the absolute differences at each pixel site in the block, i.e.,

$$\sum_{i=0}^{7}\sum_{j=0}^{7}|\underline{v}_r(i,j) - \underline{v}_o(i,j)| \tag{2.4}$$

For type 3 blocks, parameters transmitted are the prediction parameters, codebook index of the best matching codeword of Codebook II and the decision parameter.

Chapter 3

DECODER STRUCTURE

The decoder is shown in Fig3.1. Inputs to the decoder are the *decision parameter* which determines the coding method used for the current block of the image, *codebook index* which determines the codeword that will be read from the codebook, *prediction parameters* which are used in the construction of the spatial prediction filter and the *motion vector* that is used in the temporal prediction.

If the current block is of type 0, previously decoded image frame is used to reconstruct the block. Let the current block be the (k, l)'th block of the n'th image frame. The reconstructed image block is obtained by just copying the block with the same position in the previously decoded image to the current image frame as shown in 3.1,

$$\hat{X}_n(8k+i, 8l+j) = \hat{X}_{n-1}(8k+i, 8l+j) \qquad 0 \leq i, j < 8 \qquad (3.1)$$

where $\hat{X}_n(i, j)$ and $\hat{X}_{n-1}(i, j)$ represents the reconstructed image intensity signals at the (i, j) pixel position for the n'th and $(n-1)$'st frames, respectively.

If the current block is of type 1, only the motion vector and the buffered previously decoded image frame is used to reconstruct the block at the decoder. Let the current block be the (k, l)'th block of the n'th image frame, and $\underline{u} = (u_x, u_y)$ show the motion vector for that block. Then the reconstructed image block is obtained using 3.2,

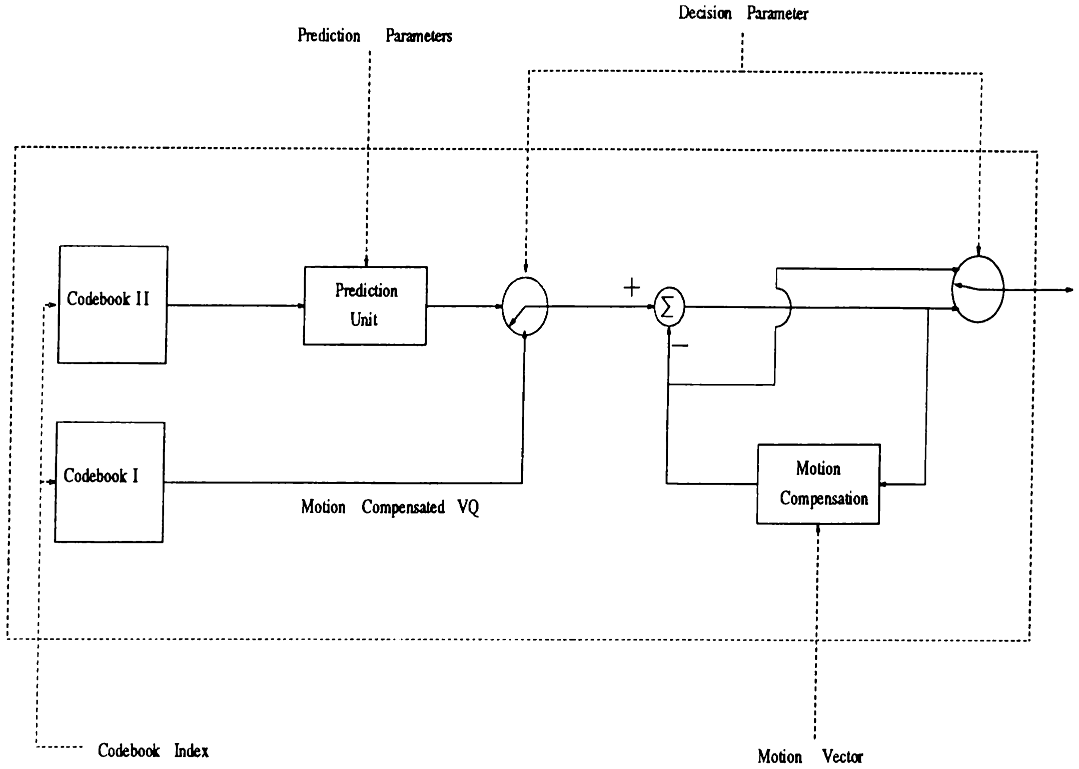

Figure 3.1: Decoder structure.

$$\hat{X}_n(8k+i,8l+j) = \hat{X}_{n-1}(8k+i-u_x,8l+j-u_y) \qquad 0 \le i,j < 8 \quad (3.2)$$

If the current block is of type 2, codebook index is used to get the vector quantizer codeword from Codebook I. Using the motion vector accompanied by the vector quantizer codeword, block is synthesized at the decoder side. With the same notation as above, reconstructed image signal for type 2 blocks is obtained using 3.3,

$$\hat{X}_n(8k+i,8l+j) = \hat{X}_{n-1}(8k+i-u_x,8l+j-u_y)+\underline{v}_{cw}(i,j) \qquad 0 \le i,j < 8 \quad (3.3)$$

It is not difficult to notice that the coding method for blocks of type 1 is the same for that of type 2 except the difference that in type 1, $\underline{v}_{cw}$ is always taken to be $\underline{0}$. Because for the blocks of type 1 motion compensated error signal is not coded.

If the block is of type 3, then the prediction filter is reconstructed by the knowledge of the prediction parameters. Suitable codeword is read from Codebook II, using the codebook index received. Via 3.4 and 3.5 reconstructed image block is obtained by the decoder.

$$\underline{v}_r(i,j) = \underline{v}_p(i,j) + \underline{v}_{cw}(i,j) \qquad 0 \leq i,j < 8 \tag{3.4}$$

$$\hat{X}_n(8k+i, 8l+j) = \underline{v}_r(i,j) \qquad 0 \leq i,j < 8 \tag{3.5}$$

Chapter 4

THE PREDICTION UNIT

The simplest model of an image is a set of discrete random variables (the pixel intensity values) in a two dimensional array, i.e., with each pixel statistically independent from any other pixel. If this model were correct, PCM coding would be perfectly adequate to encode the image concisely. Because PCM does not exploit any redundancy in the form of correlations of the pixel intensities, instead it considers every pixel to be totally independent of others. However this is not the case. A more realistic model is that of a 2D array of pixel intensities with high dependencies (or correlations) between closely neighboring pixels. By optimal selection of the prediction filter we can remove a great deal of the correlation between pixels, thus leaving a more statistically independent residual image, which is more suitable for VQ coding. Dependency of VQ on the training sequence diminishes for low correlated signal sources. This provides the compression power of multi-source coding in video applications with global-like codebooks [36]. Also by decreasing the dependency of the VQ codebook on the training sequence, the coding scheme becomes more robust to changes in the input signal. After spatial prediction, unpredictable part of the image signal is left to be vector quantized. Therefore, the VQ codebook contains several edge structures that are mostly encountered in head and shoulders type of images. It is easy to code the predictable part of the signal by just sending the prediction parameters for the block. But if the predictable part was also vector quantized this would give rise to larger overall reconstruction error.

Two different prediction methods are used in the video codec, and their effects on the codec performance are compared. Prediction methods examined are :

(i) Linear prediction, and

(ii) Gibbs random field(GRF) model based non-linear prediction.

Both of the prediction methods are causal in the sense that, in the reconstruction stage previously decoded image pixels are used in predicting other pixels. Only spatial prediction is used, i.e., for the blocks that are coded using predictive VQ, no motion compensation (temporal prediction) is used. Also prediction is adaptive, meaning that the prediction parameters are updated in each image block.

In the sections that follow, linear and GRF model based non-linear prediction methods and the calculation of the prediction parameters will be investigated in detail.

4.1 Linear Prediction

Individual pixel intensities can be thought of as a linear combination of other neighboring pixel intensities, plus a more random (or uncorrelated) residual signal. Such dependencies can be expressed as in 4.1,

$$X(i,j) = \sum_k \sum_l \alpha_{kl} X(i-k, j-l) + \varepsilon(i,j) \tag{4.1}$$

where $X(i,j)$ is the image intensity at pixel location (i,j), $\varepsilon(i,j)$ is the residual signal, and α_{kl} are the so called linear prediction coefficients, and (k,l) run over some definable region of support. The above equation in fact describes an infinite impulse response (IIR) filter, and the process is known as linear predictive (LP) filtering [37] [38] [39] [40] [41].

In our case we take the LP support to be consisting of three neighboring pixels. For the formulation that follows pixel configuration given in Fig4.1 is taken to be the basis.

Let the pixel values $X(i-1,j)$, $X(i,j-1)$ and $X(i-1,j-1)$ be given, $X(i,j)$ is desired to be predicted linearly using the pixel values that are given.

X(i-1,j-1)	X(i-1,j)
X(i,j-1)	X(i,j)

Figure 4.1: Pixel configuration for linear prediction.

Therefore prediction for $X(i,j)$, $\hat{X}(i,j)$, can be expressed as a linear superposition of $X(i-1,j)$, $X(i,j-1)$ and $X(i-1,j-1)$ as in 4.2,

$$\hat{X}(i,j) = aX(i,j-1) + bX(i-1,j-1) + cX(i-1,j) \tag{4.2}$$

Therefore the residual signal can be expressed as,

$$\begin{aligned} \varepsilon(i,j) &= X(i,j) - \hat{X}(i,j) \\ &= X(i,j) - aX(i,j-1) - bX(i-1,j-1) - cX(i-1,j) \end{aligned} \tag{4.3}$$

Total mean square error (MSE) for the block will be,

$$E = \sum_{i,j} \varepsilon(i,j)^2 = \sum_{i,j} (X(i,j) - aX(i,j-1) - bX(i-1,j-1) - cX(i-1,j))^2 \tag{4.4}$$

Prediction parameters a, b, and c are desired to be chosen such that the total prediction error E is minimized. Therefore we take the partial derivatives of E with respect to a, b and c, and equate them to zero.

$$\frac{\partial E}{\partial a} = 0 \qquad \frac{\partial E}{\partial b} = 0 \qquad \frac{\partial E}{\partial c} = 0$$

These three linear equations in three unknowns give rise to a 3x3 matrix equation stated in 4.5,

$$\begin{bmatrix} r(0,0) & r(1,0) & r(1,1) \\ r(1,0) & r(0,0) & r(0,1) \\ r(1,1) & r(0,1) & r(0,0) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} r(0,1) \\ r(1,1) \\ r(1,0) \end{bmatrix} \tag{4.5}$$

where $r(.,.)$ is the autocorrelation function which is defined as in 4.6,

$$r(m,n) = \sum_i \sum_j X(i,j)X(i-m, j-n) \tag{4.6}$$

where $X(i,j)$ is the image intensity at pixel position (i,j) and summation is over the block of the image.

4.2 GRF model based non-linear prediction

A causal Gibbs random field (GRF) based statistical model is assumed for the image intensity signal. A different notation will be used to investigate the non-linear prediction method proposed in this study. In this section following notation will be used :

L : 2D array (*lattice*) of points,

η_{ij} : set of points that are in the neighborhood of the point (i,j),

X_{ij} : random variable at point (i,j),

$X(i,j)$: realization of the random variable at (i,j),

β : any subset of L,

$\chi(\beta)$: numerical realization of subset of pixels constituting β,

P : probability assignment function.

Image intensity function is modeled as a 2D array of random variables, over a finite N_1xN_2 rectangular lattice of points (pixels) defined as

$$L = \{ \ (i,j) : 0 \leq i < N_1, \quad 0 \leq j < N_2 \ \}$$

The description of a Gibbs random field is based on the definition of a neighborhood structure on L. [42] [43]

A collection of subsets of L given by

$$\{ \ \eta_{ij} \ : \quad (i,j) \in L, \qquad \eta_{ij} \subseteq L \ \}$$

is a neighborhood system on L if and only if the neighborhood, η_{ij}, of a point (i,j) is such that :

$$(i,j) \notin \eta_{ij} \quad and$$

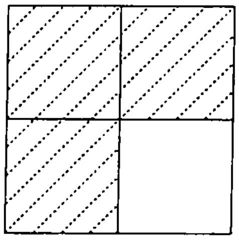

Figure 4.2: First order causal neighborhood structure used in GRF model based non-linear prediction

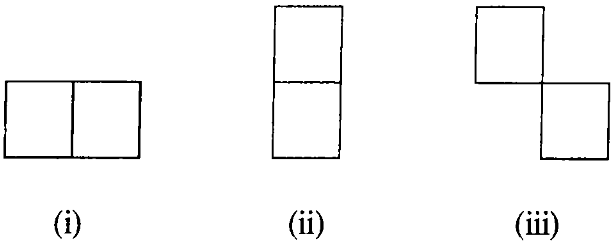

Figure 4.3: Cliques associated with the selected the neighborhood system.

$$(k,l) \in \eta_{ij} \Rightarrow (i,j) \in \eta_{kl}, \qquad \forall (i,j) \in L$$

In this study, the shape of the neighborhood of an inner point of L is assumed to be independent of the position of the point (pixel) in the image region L. A causal first order neighborhood shape, which is shown in Fig4.2, is entertained in the random field model.

The *"cliques"* associated with a lattice-neighborhood pair (L, η) is defined as follows.

A clique of the pair (L, η), denoted by c, is a subset of L such that,

(i) c consists of a single pixel, or

(ii) for $(i,j) \neq (k,l)$, $(i,j) \in c$ and $(k,l) \in c \Rightarrow (k,l) \in \eta_{ij}$

The types of cliques associated with the neighborhood shape given in Fig4.2 are shown in Fig4.3. As it is noted, each type of clique consists of two pixels in the horizontal, vertical and diagonal directions.

A Gibbs random field is a probability assignment on the elements of the set of all numerical realizations χ, subject to the following condition (Markovian property) :

$$P(X(i,j) \mid \chi(L \backslash \{(i,j)\})) = P(X(i,j) \mid \chi(\eta_{ij})), \quad \forall (i,j) \in L$$

Some preliminary definitions are given, but in order for a random field model to be totally established, joint distribution probability function must be defined.

Let η be a neighborhood system over the finite lattice L. A random field $X = \{ X_{ij} \}$ defined on L has a Gibbs distribution (GD) or equivalently is a Gibbs random field with respect to η if and only if its joint distribution is of the form,

$$P(X = x) = \frac{1}{Z} e^{-U(x)}$$

where X is the random field, x is the realization of the random field (i.e., a specific image) and $U(x)$ is the so called *energy* of the image. The *partition function* Z is simply a normalizing constant. Energy function can be expressed as,

$$U(x) = \sum_{all\ c} V_c(x)$$

where $V_c(x)$ is the potential associated with clique c. The only condition on the otherwise totally arbitrary clique potential $V_c(x)$ is that it depends only on the pixel values in the clique c.

A maximum likelihood (ML) prediction is assumed. Let the pixel values at positions $(i, j-1)$, $(i-1, j-1)$ and $(i-1, j)$ be given as $X(i, j-1)$, $X(i-1, j-1)$ and $X(i-1, j)$, respectively. GRF model based non-linear prediction tries to find $X(i, j)$ such that the following probability is maximized,

$$P(X_{ij} \mid X(k,l); k \neq i, l \neq j) \tag{4.7}$$

Since Markovian property is assumed to hold, probability expression in 4.7 is equivalent to the one in 4.8,

$$P(X_{ij} \mid X(i-1,j), X(i-1,j-1), X(i,j-1)) \tag{4.8}$$

Therefore, the prediction process is reduced to the maximization of the probability of the numerical realization of a pixel in the image, given the pixel values of its neighbors. Since the probability of an image is given in the following form,

$$P(X = x) = \frac{1}{Z} e^{-U(x)}$$

maximization of the probability is the same as the minimization of the energy function $U(x)$. Hence, the prediction is nothing but an optimization process on the energy function.

In this study, energy function $U(x)$ is chosen as shown in 4.9,

$$U(X(i,j), X(i-1,j), X(i,j-1), X(i-1,j-1)) = \mathcal{F}(D_h, D_v, D_d) \quad (4.9)$$

where,

$$D_h = \begin{cases} -1, & \text{if } \mid X(i,j) - X(i,j-1) \mid \leq T \\ +1, & \text{otherwise} \end{cases}$$

$$D_v = \begin{cases} -1, & \text{if } \mid X(i,j) - X(i-1,j) \mid \leq T \\ +1, & \text{otherwise} \end{cases}$$

$$D_d = \begin{cases} -1, & \text{if } \mid X(i,j) - X(i-1,j-1) \mid \leq T \\ +1, & \text{otherwise} \end{cases}$$

where T is a pre-selected threshold.

4.2.1 Estimation of the non-linear prediction parameters

D_h, D_v and D_d can take two different values, namely -1 and $+1$. Therefore, there are a total of 8 different configurations for the triple (D_h, D_v, D_d), as shown in Table4.1.

Eight counters are assigned to the values of the triple (D_h, D_v, D_d) : $C_0, ..., C_7$. In each block of the image 8 counters are initialized to zero. Then all of the pixels in the block are visited. Depending on the value of the triple (D_h, D_v, D_d) corresponding counter is increased by one, while the others are

Counter	D_h	D_v	D_d
C_0	-1	-1	-1
C_1	-1	-1	$+1$
C_2	-1	$+1$	-1
C_3	-1	$+1$	$+1$
C_4	$+1$	-1	-1
C_5	$+1$	-1	$+1$
C_6	$+1$	$+1$	-1
C_7	$+1$	$+1$	$+1$

Table 4.1: Counter assignment to the triple (D_h, D_v, D_d).

not effected. For example, if we encounter the following case in the block,

$$| X(i,j) - X(i-1,j) | \leq T$$
$$| X(i,j) - X(i,j-1) | \leq T$$
$$| X(i,j) - X(i-1,j-1) | \leq T$$

i.e., $(D_h, D_v, D_d) = (+1, +1, +1)$, then the counter C_7 is increased by one while the other counters are kept constant. It is not difficult to notice that,

$$\sum_{i=0}^{7} C_i = \text{number of pixels visited in the block}$$

After all of the pixel sites in the block are visited, the indices of three counters with highest values are saved and transmitted as the prediction parameters for the block. For example, after all of the pixels in a block are visited and following counter values are calculated,

$$C_0 = 2, \quad C_1 = 0, \quad C_2 = 0, \quad C_3 = 0,$$
$$C_4 = 1, \quad C_5 = 6, \quad C_6 = 10, \quad C_7 = 30$$

then the prediction parameters will be $(p_1, p_2, p_3) = (7, 6, 5)$. As it is noted, there are a total of 8x7x6 = 336 different combinations for the prediction parameters. Without any further coding, 9 bits are required to code the prediction parameters of a block.

4.2.2 Non-linear prediction process

Given the prediction parameters (p_1, p_2, p_3) and the pixel values $X(i-1,j)$, $X(i-1,j-1)$ and $X(i,j-1)$, the prediction process has the following steps :

1. Take the first prediction parameter p_1 (i.e., the index of the counter with the highest value)

2. Try to find an $X(i,j)$ such that (D_h, D_v, D_d) triple value corresponding to C_{p_1} is satisfied.
   (i.e., if $p_1 = 7$ for example, try to find $X(i,j)$ such that :

$$| X(i,j) - X(i-1,j) | \leq T$$

$$| X(i,j) - X(i,j-1) | \leq T$$

$$| X(i,j) - X(i-1,j-1) | \leq T)$$

3. a. If only one $X(i,j)$ is found, take it to be the prediction for the point (i,j)

   b. If more than one value is found, select one of them depending on the following rule :

$$\min_{X(i,j)} \; \{ \; D_h \mid X(i,j) - X(i,j-1) \mid + \\ D_v \mid X(i,j) - X(i-1,j) \mid + \\ D_d \mid X(i,j) - X(i-1,j-1) \mid \}$$

   c. If no value is found satisfying C_{p_1}, take the second prediction parameter, p_2, and try to find an $X(i,j)$ such that (D_h, D_v, D_d) triple value corresponding to C_{p_2} is satisfied. (i.e., do the steps 1 thru 3 above with p_2 instead of p_1) If the prediction proves to be unsuccessful with p_2 also, take p_3 as the prediction parameter. If a prediction for $X(i,j)$ is not found using all of the prediction parameters (which is rarely the case), prediction for $X(i,j)$ is obtained by taking the three point *median* of $X(i-1,j)$, $X(i-1,j-1)$, and $X(i,j-1)$.

Chapter 5

VECTOR QUANTIZER DESIGN ALGORITHM

In the vector quantizer design, the Linde-Buzo-Gray (LBG) algorithm is used. A brief explanation about LBG algorithm is given in this section, but for more detailed information and/or its applications to different problems see [44] [45] [46] [47] [48].

The general LBG algorithm iteratively improves a given codebook for a given source probability density function (PDF). In practice however, the source PDF is not known a priori, therefore the algorithm must use instead a sequence of training data which is assumed to be similar to the image sources that will be coded later. In this way estimate of the PDF is formed from a pre-known training set. In our case we used several "head and shoulders" type of images in training of the vector quantizer, since the codec will be used for the video-phone applications.

Starting from the given codebook, each vector in the training set is clustered around its closest match in terms of MSE, in the codebook to form a population of training vectors for each codeword. A new codebook is then formed by taking the centroid of each cluster. This procedure is then iteratively repeated, leading to a monotonically decreasing total training set error for the codebook being designed.

There are several ways of choosing the initial codebook :

(i) as a series of random vectors,

(ii) as a series of vectors equally distributed in Euclidean space, as in uniform scalar quantization,

(iii) by using the centroid of the whole training set as an initial codeword, and then splitting algorithm to increase the number of codewords to 2, then to 4, then to 8, etc.

Since the third approach is implemented in the vector quantizer design algorithm used in this thesis, codeword splitting algorithm will be explained in a little bit detail in the rest of this section.

In the codebook splitting algorithm, codebook is started off with a codeword, say $\underline{x}$, which is taken to be the centroid of the entire training set. This vector is then split into two vectors by a small perturbation vector, $\underline{\epsilon}$. Thus there are now two vectors, $(\underline{x} + \underline{\epsilon})$ and $(\underline{x} - \underline{\epsilon})$. The LBG algorithm is then applied to this new codebook to form a better two-word codebook. Each codeword is then split into two words, and LBG algorithm is used again. At each step the number of codewords is doubled, and the LBG algorithm is applied to optimize the new codebook.

5.1 Preparation of the training sequence

In the codec structure there are two different codebooks, Codebook I and Codebook II, which are used in the coding of type 2 and type 3 blocks, respectively. Therefore, two training sequences with different characteristics are generated to be used in the design of the codebooks. Both of the training sequences are made up of blocks of error-like signals.

First training sequence contains motion compensated block differences. Let $X_n(i,j)$ and $X_{n-1}(i,j)$ be the image intensity values at spatial coordinate (i,j), on the two consecutive image frames which are used in the training process. And let $\underline{u} = (u_x, u_y)$ be the motion vector associated with (k,l)'th block of the n'th image frame. Then the training sequence element, $\underline{v}_t(i,j)$ obtained from the (k,l)'th block of the n'th frame is constructed by using 5.1.

$$\underline{v}_t(i,j) = X_n(8k+i, 8l+j) - X_{n-1}(8k+i-u_x, 8l+j-u_y) \qquad 0 \leq i,j < 8 \quad (5.1)$$

Second training sequence contains the spatial prediction error signals. Let $\underline{v}_o(i,j)$ be the original image block, and $\underline{v}_p(i,j)$ be the prediction for it, then the training sequence element,$\underline{v}_t(i,j)$, is obtained using 5.2,

$$\underline{v}_t(i,j) = \underline{v}_o(i,j) - \underline{v}_p(i,j) \qquad\qquad 0 \leq i,j < 8 \qquad\qquad (5.2)$$

If prediction is not used in the codec then in generating the training sequences $\underline{v}_p$ in 5.2, is taken to be $\underline{0}$ and again two different codebooks for vector quantization of type 2 and 3 blocks are generated.

Chapter 6

SECOND LEVEL CODING

At each block of the image some parameters are extracted from the image by the coder and then those parameters are transmitted to the decoder. Decoder, using those parameters synthesizes the image at the receiver. Extraction of the parameters is called the *first level coding.* In the first level coding, greatest part of the compression is achieved. But when the parameters themselves are coded further (i.e., not transmitted in their own raw format), compression ratio is increased by a non-negligible amount. This process, coding of the parameters of the first level coding further, is called the *second level coding.*

A block of the image can be of four different types : 0, 1, 2, or 3. Parameters that must be transmitted for each type of block are known by the coder and the decoder. They are listed below :

0. decision parameter.

1. decision parameter,
   motion vector.

2. decision parameter,
   motion vector,
   codebook index.

3. decision parameter,
   codebook index,
   prediction parameters.

Event	*Approximate percentage*	*Assigned codeword*	*Number of bits*
type 0	65	0	1
type 1	20	1 0	2
type 2	10	1 1 0	3
type 3	5	1 1 1	3

Table 6.1: Huffman code table for decision parameter with no grouping.

Prediction parameters are vector quantized. The number of levels in the vector quantizer for the prediction parameters and the sizes of Codebook I and Codebook II are known a priori by the coder and the decoder. No further coding scheme is used for the codebook index and prediction parameters. The codebook and the prediction parameters' vector quantizer index are transmitted directly. Second level coding is applied to the motion vector and decision parameter only. In the following sections methods used to code the decision parameter and the motion vector will be stated in detail.

6.1 Decision parameter

Decision parameter can take four different values (0,1,2,3) showing the type of the coding scheme for the current block. Therefore, if decision parameter is transmitted in its raw format, 2 bits per block is required. Instead, Huffman coding [49] is adapted for the decision parameter. Huffman coding is an effective way of reducing the number of bits used to represent events with non-uniform probabilities. Events with larger probability are coded with fewer number of bits, on the other hand less probable events are coded with more bits. Two different Huffman coding schemes can be applied to the decision parameter.

(i) *With no grouping of events :* It is experimentally observed that, on the average 65 percent of blocks is of type 0, 20 percent of blocks is of type 1, 10 percent of blocks is of type 2, and 5 percent of the blocks is of type 3. Since the probabilities are not the same for the events Huffman coding will give rise to a reduction of the number of bits per decision parameter. Statistical results, codeword assigned to each value of decision

Event	*Approximate percentage*	*Assigned codeword*	*Number of bits*
type 0 (length 1)	14	0 1 1	3
type 0 (length 2)	8	0 0 1 1	4
type 0 (length 3)	5	1 1 1 1	4
type 0 (length 4)	3	1 1 1 0 0	5
type 0 (length 5)	5	0 0 0 1 1	5
type 0 (length 10)	2	0 0 0 0 1 0	6
type 0 (length 25)	2	0 0 0 0 1 1	6
type 0 (length 50)	0.4	0 0 0 0 0 0 0 0	8
type 0 (length 100)	0.6	0 0 0 0 0 0 0 1	8
type 1 (length 1)	18	1 0	2
type 1 (length 2)	7	0 1 0 0	4
type 1 (length 3)	4	0 0 1 0 0	5
type 1 (length 4)	2	0 0 0 0 0 1	6
type 1 (length 5)	3	1 1 1 0 1	5
type 2 (length 1)	10	1 1 0	3
type 2 (length 2)	4	0 0 1 0 1	5
type 2 (length 3)	2	0 0 0 1 0 0	6
type 2 (length 4)	1	0 0 0 0 0 0 1	7
type 2 (length 5)	2	0 0 0 1 0 1	6
type 3 (length 1)	7	0 1 0 1	4

Table 6.2: Huffman code table for decision parameter with grouping.

parameter and the number of bits used to represent each event are shown in Table6.1. With this method, average number of bits for coding the decision parameter is achieved as 1.5 bits per block.

(ii) *With grouping of events :* In this case the events are grouped, because of the experimental observation that type 0, type 1 and type 2 blocks are distributed on the image as bundles. Type 0 blocks which are usually encountered on the background of the image have larger groups (sometimes up to a few hundreds of consecutive type 0 blocks), compared to type 1 and type 2 blocks (maximum length of the groups for type 1 and type 2 blocks is around 10). On the other hand type 3 blocks are not encountered as bundles on the image. Therefore, several groping structures are selected for types 0, 1, and 2, and no grouping is assumed for type 3. Selection of events, statistics associated with, codewords assigned to and number of bits used for each event are presented in Table6.2. With this method average number of bits for coding of the decision parameter is reduced to 1.0 bit per block.

6.2 Motion vector

Motion vector is coded differentially. Let the current motion vector be $\underline{u}_c = (u_{c_x}, u_{c_y})$ and the previous motion vector be $\underline{u}_p = (u_{p_x}, u_{p_y})$. Difference motion vector,

$$\Delta\underline{u} = (\Delta u_x, \Delta u_y) = (u_{c_x} - u_{p_x}, u_{c_y} - u_{p_y})$$

is coded. Since,

$$-7 \leq u_x, u_y < 7$$

we have,

$$-14 \leq \Delta u_x, \Delta u_y < 14$$

Normally, 5 bits is required to represent Δu_x (or Δu_y) therefore 5x2=10 bits to represent the full motion vector. In order to decrease the number of bits spent for motion vector representation, Huffman coding is implemented. Events are defined to be the components of the difference motion vector. First, the statistics about the events are obtained by using some typical image sequences, and then Huffman code table is generated.

$\Delta\underline{u}$	*Approximate percentage*	*Assigned codeword*	*Number of bits*
−8 or +8	1	0 0 0 0 0 0 0	7
−7 or +9	1	0 0 0 0 0 0 1	7
−6 or +10	2	0 0 0 0 1 0	6
−5 or +11	2	0 0 0 0 1 1	6
−4 or +12	3	0 1 0 0 1	5
−3 or +13	4	0 0 0 1 1	5
−2 or +14	5	0 1 1 1	4
−1	8	0 0 1 0	4
0	50	1	1
+1	8	0 0 1 1	4
+2 or −14	5	0 1 0 1	4
+3 or −13	4	0 1 1 0	4
+4 or −12	3	0 0 0 1 0	5
+5 or −11	2	0 1 0 0 0	5
+6 or −10	1	0 0 0 0 0 1 0	7
+7 or −9	1	0 0 0 0 0 1 1	7

Table 6.3: Huffman code table for motion vector.

At each block Δu_x and Δu_y are coded independently (and consecutively). The same Huffman code table is used for both Δu_x and Δu_y. Initially, at the beginning of each frame,

$$u_{p_x} = 0 \quad \text{and} \quad u_{p_y} = 0$$

is assumed.

Via the Huffman coding, average number of bits used for one component of the motion vector is reduced from 5 bits to 2.8 bits. Therefore, 2.8x2=5.6 bits is required to code a motion vector fully.

In Table6.3 statistics, assigned codewords and the number of bits for representation of motion vector is presented.

Chapter 7

SIMULATION RESULTS

Simulations are conducted using “Miss America”, “Claire” and “Trevor” image sequences. All of these sequences are made up of the so called *head and shoulders* type of color images. All of the images are originally in the Common Intermediate Format (CIF). CIF has Y, U, V representation for color images. Y component (luminance) has dimensions 288x352, U and V components (chrominance differences) have the dimensions 144x176, and each pixel is represented by 8 bits. First frames of these three sequences are shown in Fig7.1, Fig7.2 and Fig7.3.

In the simulations Y, U and V components of the image signal are coded separately in order to have the modularity. This enables the receiver to choose only the luminance part of the coded signal and decode it to watch a black/white scene. During the coding process each image frame is divided into blocks. Size of each block is 8x8 for Y component and 4x4 for U and V components. The selection of block size for Y component is due to the frequent use of 8x8 block size in the literature. Also, since VQ coding algorithms are used, large block sizes decrease the effectiveness of VQ. On the other hand small block sizes are not appropriate to the image coding application studied in this thesis, since using small blocks decreases the compression ratio. In order to have a one to one correspondence of the blocks of Y, U, and V components of the image signal, block size is determined to be 4x4 for the color components. By this selection the number of blocks in the Y, U and V components are the same, and they can be matched in a one to one manner. Three blocks (8x8 luminance block with two 4x4 chrominance difference blocks) constitute a color (image) block.

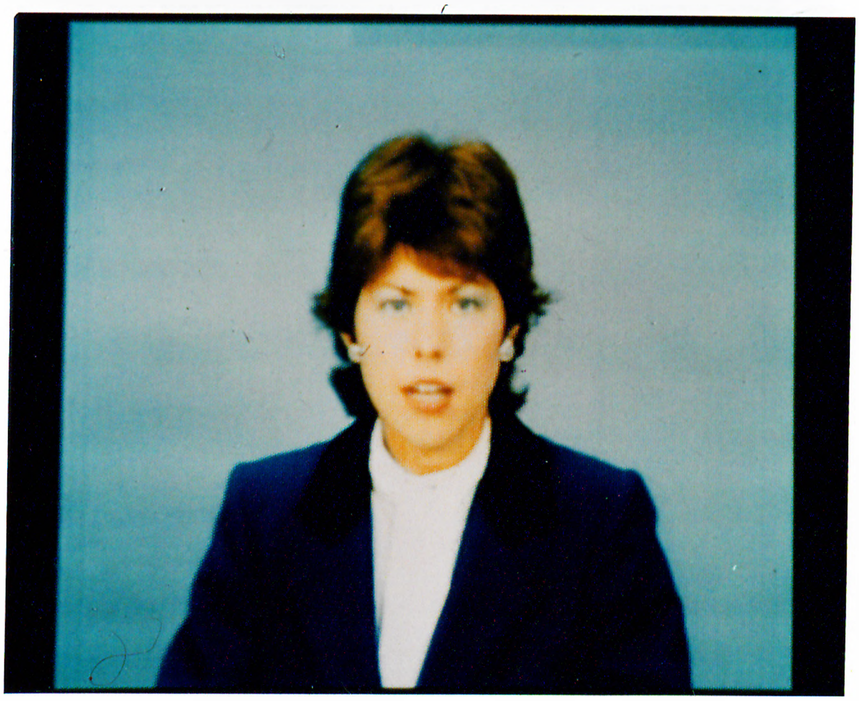

Figure 7.1: First frame of the “Claire” sequence.

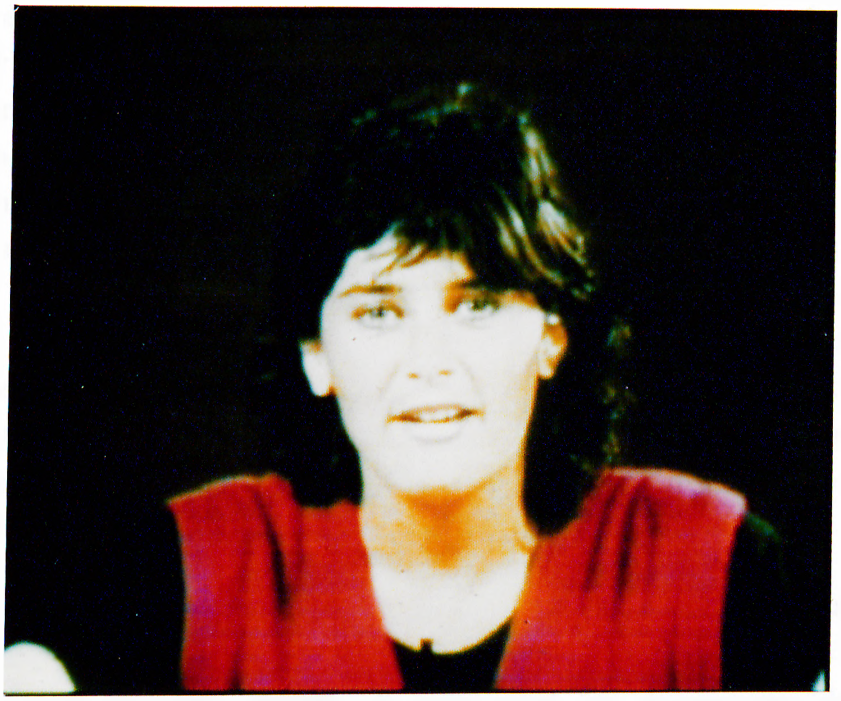

Figure 7.2: First frame of the “Miss America” sequence.

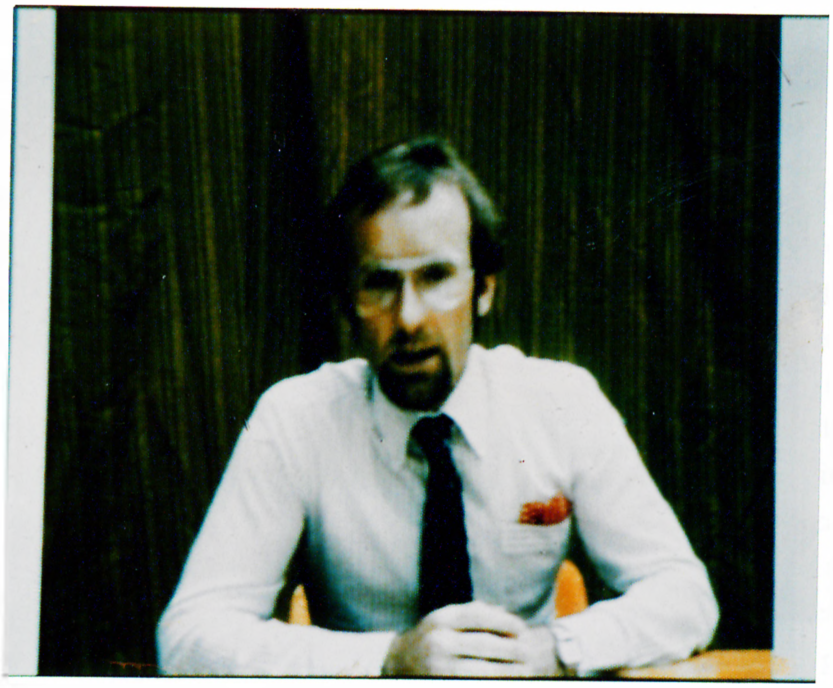

Figure 7.3: First frame of the "Trevor" sequence.

Without any coding a color block is represented by (8x8 + 4x4 + 4x4)x8 = 768 bits.

Prediction parameters are vector quantized to 256 levels (8 bits). Motion vector and decision parameter are Huffman coded to 2x2.8=5.6 bits and 1.0 bits per block, respectively. VQ codebooks used in vector quantizing the motion compensated error blocks are separately designed for the Y, U, and V components and they are the same for the codecs making use of linear and GRF model based non-linear prediction. Predictive VQ codebooks, on the other hand, are also separately designed for Y, U, and V components, but they are not the same for linear and non-linear prediction based codecs. In designing the vector quantizer codebooks LBG algorithm is used whose details are explained in Chapter 5. The reason for the selection of LBG algorithm is the fact that it is the most widely used and well performing clustering algorithm in the literature. Also, several studies in the literature concluded that the LBG algorithm is suitable for use in the video compression algorithms.

For video-phone application the images are expected to be standardized in the Quarter Common Intermediate Format (QCIF). QCIF has Y, U, V representation for color images. Y component has dimensions 144x176, U and V components have dimensions 72x88 and each pixel is represented by 8 bits. Also, expected standardization for the number of frames falling on to the screen per second (frame rate) is in between 5.00 Hz and 8.33 Hz. Throughout this section, compression ratio and transmission bit rate calculations are done assuming QCIF color images and 6.00 Hz frame rate.

In designing the codebooks "Miss America" and "Trevor" sequences are used during the training process, and "Claire" sequence is used in coding and decoding stages. VQ codebooks of several sizes are created in order to arrange experiments at different transmission bit rates. Because codebook size is one of the main parameters that determines the transmission bit rate. In order to investigate the effects of linear prediction and GRF model based non-linear prediction on the codec performance at different bit rates, several experiments are conducted by changing the transmission bit rate at which the codecs operate. A total of four codecs are used in the simulations. First three make use of the coding-decoding technique presented in this thesis. Only the prediction units of the codecs differ. Linear prediction, GRF model based non-linear prediction and no prediction cases are implemented in the codec structures. Proposed coding scheme with no prediction is equivalent to a pure split-VQ coding algorithm [48] [50]. In the split-VQ coding algorithms different sets of codebooks are designed and used for different classes of sub-images. Usually, features such as edges and continuous texture provide bases for sub-image classifications. In our case segmentation (splitting) of image regions (blocks) is done with respect to the degree of temporal activity. Fourth codec implements the COST-SIM2 coding-decoding algorithm which is an ongoing study for the standardization of the very low bit rate video codec, by the European countries. COST-SIM2 is a DCT based algorithm very similar to other DCT based algorithms such as JPEG, MPEG, etc.

Since other conditions are the same for the codecs making use of linear, GRF model based non-linear prediction and no prediction, overall performance difference of the codecs are due to the prediction performances. Signal to noise ratio (SNR) is taken to be the performance criterion. SNR is defined as in 7.1,

$$SNR = 20 \log \frac{255}{RMSE} \tag{7.1}$$

where RMSE is the root mean square error defined in 7.2,

$$RMSE = \sqrt{\frac{\sum_i \sum_j (X(i,j) - \hat{X}(i,j))^2}{N}} \tag{7.2}$$

where $X(i,j)$ and $\hat{X}(i,j)$ are the original and the decoded images respectively, and N is the number pixels in the image frame. Also for the comparison of the

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	33.670	40.939	44.792
2	28.855	35.795	36.906
3	28.440	35.541	34.352
4	28.045	34.506	36.266
5	27.376	32.740	40.012
6	27.176	32.648	40.019
7	26.855	33.801	35.814
8	26.624	32.139	36.144
Average	28.380	34.764	38.038

Table 7.1: SNR of "Claire" sequence coded at 8 Kbit/s with linear prediction.

performances of the codec proposed in this thesis and the COST-SIM2 codec, SNR is taken to be the main criterion.

Three experiments are performed at three different transmission bit rates : 8 Kbits/s, 12 Kbits/s and 16 Kbits/s. In fact, it must be noted that, for each experiment the codebook sizes are the same, not the bit rates. Since no prediction parameters are transmitted for the "no-prediction" case, transmission bit rate of the codec with no prediction is slightly less (0.5-1.0 Kbits/s less) compared to the others. (i.e., compared to the codecs with linear and non-linear prediction and the COST-SIM2 codec). For each experiment first 8 frames of "Claire" sequence is coded and decoded by the four codecs and SNRs of 8 frame individually and the average of the SNRs of 8 frames are reported in the tables.

Experiment 1. As the first experiment transmission bit rate is selected to be 8 Kbits/s. Sizes of VQ codebooks used in vector quantizing the motion compensated block differences are 32 (5 bits), 8 (3 bits) and 8 (3 bits) for Y, U and V components, respectively. Sizes of prediction based VQ codebooks are 32 (5 bits), 8 (3 bits) and 8 (3 bits) for Y, U and V components, respectively. Compression ratio for type 0 blocks 1, 2, and 3 blocks are 768, 116, 44 and 21, respectively. And the overall compression ratio for the codec is 228. SNR results of the very low bit rate video codec presented in this thesis with linear prediction, GRF model based non-linear prediction and no prediction are reported in Table7.1, Table7.2 and Table7.3, respectively. Also the SNR results of COST-SIM2 codec with 8 Kbits/s transmission bit rate is presented in Table7.4.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	34.447	41.162	45.071
2	29.729	36.153	41.739
3	29.047	35.493	41.105
4	28.454	34.914	40.806
5	27.447	34.181	40.296
6	26.853	33.964	40.185
7	27.017	33.768	40.225
8	26.683	33.615	40.179
Average	28.713	35.406	41.201

Table 7.2: SNR of "Claire" sequence coded at 8 Kbit/s with GRF model based non linear prediction.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	35.497	41.938	44.971
2	31.255	37.291	42.029
3	30.500	36.715	41.445
4	29.847	36.058	41.063
5	28.623	35.331	40.447
6	28.285	35.236	40.504
7	28.386	35.078	40.360
8	28.152	34.903	40.428
Average	30.068	36.569	41.406

Table 7.3: SNR of "Claire" sequence coded at 8 Kbit/s with no prediction.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	32.315	34.685	36.690
2	32.061	34.588	36.576
3	31.934	34.432	36.463
4	31.968	34.635	36.637
5	32.003	34.615	36.614
6	31.927	34.441	36.441
7	32.053	34.608	36.543
8	32.174	34.667	36.538
Average	32.054	34.584	36.563

Table 7.4: SNR of "Claire" sequence coded at 8 Kbit/s using COST-SIM2 codec.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	34.952	42.791	46.241
2	31.153	39.032	37.385
3	30.659	38.377	34.916
4	30.480	37.573	36.939
5	29.774	34.612	42.095
6	29.605	34.555	42.343
7	29.452	36.673	37.060
8	29.044	34.076	36.961
Average	30.636	37.211	39.242

Table 7.5: SNR of "Claire" sequence coded at 12 Kbit/s with linear prediction.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	36.170	43.498	46.444
2	31.167	40.370	43.792
3	30.387	39.608	43.174
4	30.437	39.047	43.082
5	29.284	37.726	42.490
6	28.984	37.550	42.453
7	28.932	37.697	42.399
8	28.860	36.692	42.256
Average	30.528	39.057	43.261

Table 7.6: SNR of "Claire" sequence coded at 12 Kbit/s with GRF model based non linear prediction.

Experiment 2. As the second experiment transmission bit rate is selected to be 12 Kbits/s. Sizes of VQ codebooks used in vector quantizing the motion compensated block differences are 256 (8 bits), 64 (8 bits) and 64 (6 bits) for Y, U and V components, respectively. Sizes of prediction based VQ codebooks are 256 (8 bits), 64 (6 bits) and 64 (6 bits) for Y, U and V components, respectively. Compression ratio for type 0 blocks 1, 2, and 3 blocks are 768, 116, 29 and 17, respectively. And the overall compression ratio for the codec is 152. SNR results of the very low bit rate video codec presented in this thesis with linear prediction, GRF model based non-linear prediction and no prediction are reported in Table7.5, Table7.6 and Table7.7, respectively. Also the SNR results of COST-SIM2 codec with 12 Kbits/s transmission bit rate is presented in Table7.8.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	36.725	43.015	46.550
2	32.853	39.253	43.819
3	32.035	38.603	43.183
4	31.855	38.034	43.093
5	30.333	37.009	42.554
6	30.002	36.873	42.532
7	30.065	36.867	42.326
8	29.966	36.285	43.370
Average	31.729	38.242	43.370

Table 7.7: SNR of "Claire" sequence coded at 12 Kbit/s with no prediction.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	33.094	35.767	37.639
2	32.873	35.678	37.597
3	32.701	35.448	37.325
4	32.748	35.712	37.515
5	32.803	35.657	37.526
6	32.707	35.542	37.390
7	32.849	35.720	37.566
8	32.938	35.702	37.556
Average	32.839	35.653	37.514

Table 7.8: SNR of "Claire" sequence coded at 12 Kbit/s using COST-SIM2 codec.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	37.688	44.763	47.572
2	33.725	41.764	37.946
3	33.242	41.351	35.459
4	32.915	41.119	37.793
5	32.696	36.524	44.384
6	32.633	36.500	44.557
7	32.178	40.201	37.783
8	32.255	36.355	37.274
Average	33.417	39.822	40.346

Table 7.9: SNR of "Claire" sequence coded at 16 Kbit/s with linear prediction.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	42.647	46.560	48.627
2	36.302	44.141	46.952
3	36.431	43.682	46.548
4	36.124	43.147	46.276
5	34.009	41.631	45.853
6	33.671	41.512	45.682
7	33.215	41.680	45.748
8	33.043	41.457	45.684
Average	35.680	42.976	46.421

Table 7.10: SNR of "Claire" sequence coded at 16 Kbit/s with GRF model based non linear prediction.

Experiment 3. As the third experiment transmission bit rate is selected to be 16 Kbits/s. Sizes of VQ codebooks used in vector quantizing the motion compensated block differences are 4096 (12 bits), 1024 (10 bits) and 1024 (10 bits) for Y, U and V components, respectively. Sizes of prediction based VQ codebooks are 4096 (12 bits), 1024 (10 bits) and 1024 (10 bits) for Y, U and V components, respectively. Compression ratio for type 0 blocks 1, 2, and 3 blocks are 768, 116, 20 and 13, respectively. And the overall compression ratio for the codec is 114. SNR results of the very low bit rate video codec presented in this thesis with linear prediction, GRF model based non-linear prediction and no prediction are reported in Table7.9, Table7.10 and Table7.11, respectively. Also the SNR results of COST-SIM2 codec with 16 Kbits/s transmission bit rate is presented in Table7.12.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	37.564	44.036	47.594
2	33.637	40.307	45.098
3	33.237	39.943	44.696
4	33.026	39.610	44.573
5	31.836	38.606	44.125
6	31.914	38.584	44.091
7	31.805	38.661	44.118
8	31.662	38.163	43.939
Average	33.085	39.739	44.779

Table 7.11: SNR of "Claire" sequence coded at 16 Kbit/s with no prediction.

Frame Number	*SNR for Y comp.*	*SNR for U comp.*	*SNR for V comp.*
1	33.509	36.620	38.369
2	33.310	36.498	38.332
3	33.124	36.280	38.050
4	33.149	36.533	38.301
5	33.243	36.512	38.342
6	33.145	36.361	38.143
7	33.300	36.560	38.294
8	33.372	36.560	38.323
Average	33.269	36.491	38.269

Table 7.12: SNR of "Claire" sequence coded at 16 Kbit/s using COST-SIM2 codec.

Codec type	*8 Kbits/s*	*12 Kbits/s*	*16 Kbits/s*
Codec with linear prediction	28.380	30.636	33.417
Codec with no prediction	30.068	31.729	33.085
Codec with GRF model based non-linear prediction	28.713	30.528	35.680
COST-SIM2 codec	32.054	32.839	33.269

Table 7.13: Average SNR (in dB) of Y component of the video signal with respect to bit rate for different codecs.

Codec type	*8 Kbits/s*	*12 Kbits/s*	*16 Kbits/s*
Codec with linear prediction	34.764	37.211	39.822
Codec with no prediction	36.569	38.242	39.739
Codec with GRF model based non-linear prediction	35.406	39.057	42.976
COST-SIM2 codec	34.584	35.653	36.491

Table 7.14: Average SNR (in dB) of U component of the video signal with respect to bit rate for different codecs.

Average SNRs achieved by the codecs are tabulated with respect to the transmission bit rate for Y, U, and V components of the video signal in Table7.13, Table7.14 and Table7.15, respectively.

As a result of the experiments, we noticed that the SNR criterion is not reliable for the comparison of the performances of the codecs. To show this fact, two frames coded by non-linear prediction based codec and COST-SIM2 codec, at 16 Kbits/s, are chosen. These are the fourth frames of the decoded image sequences. As it is seen on Table7.10 and Table7.12, SNR of the frame coded by non-linear prediction based codec is 3 dB higher compared to the SNR of the frame coded with COST-SIM2 codec. Fourth frames of the decoded sequence are shown in Fig7.4 and Fig7.5. And also, the absolute differences of the Y components of decoded and the original images for non-linear prediction based codec and COST-SIM2 codec are demonstrated in Fig7.6 and Fig7.7. It is easy to observe that although the non-linear prediction based coding method is superior to the DCT based coding method with respect to the SNR criterion, subjective evaluation concludes the reverse.

Codec type	*8 Kbits/s*	*12 Kbits/s*	*16 Kbits/s*
Codec with linear prediction	38.038	39.242	40.346
Codec with no prediction	41.406	43.370	44.779
Codec with GRF model based non-linear prediction	41.201	43.261	46.421
COST-SIM2 codec	36.563	37.514	38.269

Table 7.15: Average SNR (in dB) of V component of the video signal with respect to bit rate for different codecs.

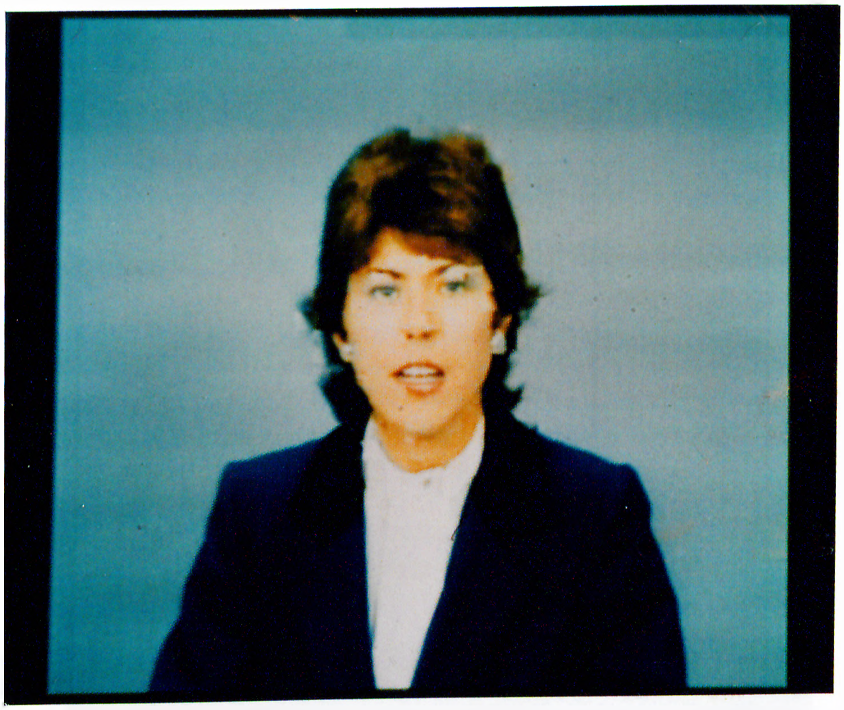

Figure 7.4: Fourth frame of the sequence coded by the non-linear prediction based codec at 16 Kbits/s. (SNRs are 36.124 dB for Y comp., 43.147 dB for U comp. and 46.276 dB for V comp.)

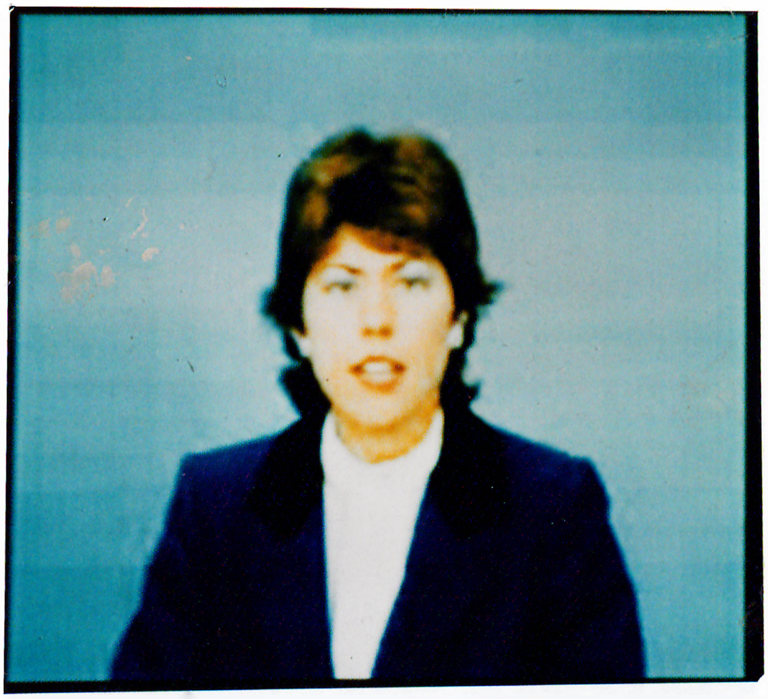

Figure 7.5: Fourth frame of the sequence coded by the COST-SIM2 codec at 16 Kbits/s. (SNR is 33.149 dB for Y comp., 36.533 dB for U comp. and 38.301 dB for V comp.)

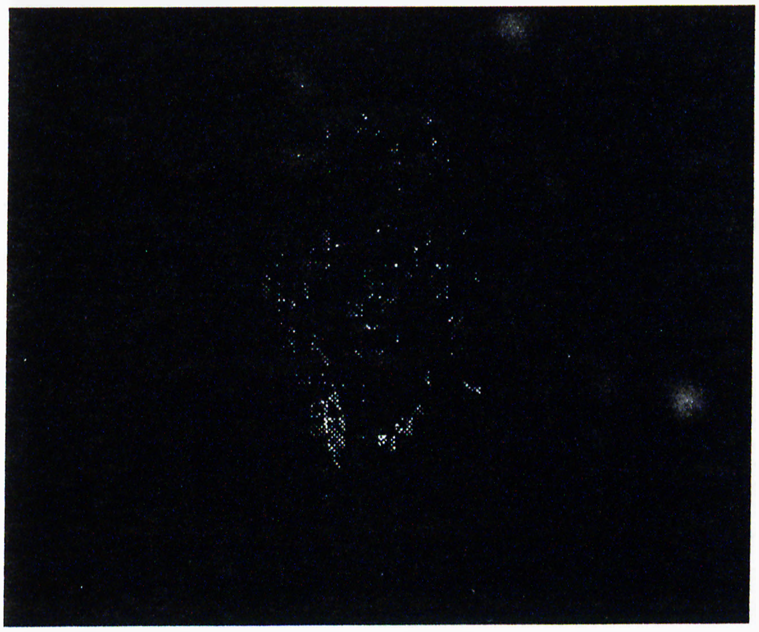

Figure 7.6: Absolute difference of the Y components of the decoded and the original frames coded by the non-linear prediction based codec at 16 Kbits/s.

Figure 7.7: Absolute difference of the Y components of the decoded and the original frames coded by the COST-SIM2 codec at 16 Kbits/s.

Chapter 8

CONCLUSION

In this thesis, an adaptive prediction and vector quantization based very low bit rate video codec is presented. The simulation examples show that this very low bit rate video codec can be used in transmitting video signals via the existing telephone lines, for the video-phone application. Although the simulations are conducted in the 8-16 Kbits/s interval, the codec may be used at higher transmission bit rates with increasing output image quality, either by designing larger size VQ codebooks or increasing the amount of bits allocated for other parameters such as motion vector and prediction parameters. By increasing the amount of bits allocated for motion vector, either the motion vector search can be done in a larger area of the previous image or interpixel accuracy may be added to the motion information. More accurate prediction parameters may be transmitted by increasing the number quantization levels of the prediction parameter vector quantizer, and hence the number of bits allocated. But in order to use the same codec structure without any change in the coding method (same number of bits allocated to motion vector and prediction parameters, as they are stated in the simulation results chapter) at the transmission bit rates above 32 Kbits/s, VQ codebooks having more than 1 million quantization levels must be designed and used. There are two major problems in using our coding scheme at bit rates above 32 Kbits/s. First problem appears as a result of the physical conditions. It is almost impossible to design and then search during coding process, a codebook of size more than 1 million, using a microprocessor which will be used in a real application. Second problem shows itself as a result of the theory. The advantage of VQ with respect to scalar quantization lies in the condition that signal to be VQ coded must be in some previously known structure. Indeed, the VQ codebooks of the codec contain certain edge structures (temporal edges in VQ codebooks for type 2 blocks, spatial edges in VQ codebooks of type 3 blocks) encountered in typical head and shoulders type of video signals. If the sizes of the VQ codebooks increase, codebooks will contain totally random, white noise like codewords in which case the advantage of using VQ vanishes. Also, when the codebook size of the VQ codebook is increased too much, scalar quantization of each pixel in the block (in stead of vector quantizing the whole block) may require less number of bits, in which case no reasonable reason survives for using VQ.

Also, a new Gibbs random field model based non-linear prediction scheme is used in the very low bit rate video codec structure. Performance of this non-linear prediction scheme is evaluated by comparing it with the linear prediction method. Also the need for "any prediction" in the codec structure is investigated by comparing the SNRs achieved by codecs that use linear and GRF model based non-linear prediction with the codec that does not make use of any kind of prediction. It is observed that at relatively low bit rates such as 8-12 Kbits/s, the improvement of the codec performance as a consequence of prediction (linear or non-linear) is negligible. The split-VQ coding algorithm which takes prediction to be always 0, gives better SNR results than the coding algorithms with prediction. This result is expected, since at lower bit rates, VQ codebook design algorithm produces very few codewords out of a very crowded training sequence. The ratio of the number of elements in the training sequence to the number of elements in the codebook is on the order of a thousand. Split-VQ algorithm gives signal vectors to the training sequence which can be easily clustered. On the other hand, for the prediction based VQ algorithms, there are edge structures in the training sequence. These training vectors of edge structures are clustered around larger number of pivot vectors, and when VQ codebook is forced to contain very few number of quantization levels, prediction based VQ system gives rise to a larger training set error in the MSE sense. Also, when a very few number of codewords are extracted from a crowded training sequence, VQ design algorithm produces very smooth and averaged out codewords which do not represent the natural clustering points of the training space. Split-VQ algorithm (therefore, the prediction based VQ algorithm with always 0 prediction) is more robust to that kind of ill-conditioned situations, because for the prediction based VQ algorithms reconstruction error accumulates in the block, giving rise to a poor overall codec performance.

But, at relatively higher transmission bit rates such as 12-16 Kbits/s, codecs with prediction have better SNR results than the codec without any prediction, with the exception that of the performance of linear prediction for the U and V components of the test image. This can also be explained, because "Claire" image sequence which is used in the experiments has sharp changes in the color components. In fact, the color components of the test image sequence used in the coding have "plateaus" (areas of constant signal level). In the "Claire" image there are five areas of almost constant color signal (green : background, black : hair, white : shirt, blue : jacket, pink : face). Between different color plateaus there are edges with infinite frequency. Linear prediction is unable to handle that kind of signals, and gives rise to larger errors in the reconstruction compared to non-linear prediction.

Experimental results show that, at those bit rates when prediction based codecs have superior performance compared to codec without any prediction, GRF model based non-linear prediction gives rise to better SNR and visual quality compared to the linear prediction.

It is observed that the very low bit rate video codec presented in this thesis with linear, non-linear and no prediction gives better SNR results compared to the COST-SIM2 codec, for the U and V components of the image. Because there are sharp edges in the color components of the test image signal, and COST-SIM2 codec blurs them giving rise to larger errors in the reconstruction. COST-SIM2 codec is a DCT based algorithm and DCT coefficients of high frequency components are forced to be quantized to zero in order to have long runs of zeros of DCT coefficients to make Huffman coding more efficient. Therefore the blurring effect of COST-SIM2 coding algorithm is expected. For the Y component of the video signal, however, if a relatively low bit rate is preferred (8-12 Kbits/s) DCT based COST-SIM2 coding scheme is superior to the coding scheme presented in this thesis, the reason being the few number of quantization levels in the VQ codebooks. But if a relatively high transmission bit rate is preferred (12-16 Kbits/s) codec presented in this thesis with GRF model based non-linear prediction has higher SNR than that of COST-SIM2 codec.

Although the coding scheme of this thesis with GRF model based non-linear prediction seems to be superior to COST-SIM2 coding method with respect to the SNR criterion, subjective evaluation shows that COST-SIM2 codec produces images with less disturbing visual degradations. The reason for this is, COST-SIM2 codec has smoothed (blurred-low pass) version of the coded image at the decoder, and it has the reconstruction error distributed uniformly all over the image frame, which is not the case for our codec. The decoded images of the codec COST-SIM2 seems blurred and low detail, but no disturbing effects are visible. On the other hand, the decoded images of the codec presented in this thesis are sharp and high detail in general, but they have some spiky errors which are disturbing.

As a brief summary, DCT based algorithms such as COST-SIM2 seems to be more suitable for very low bit rate video coding applications compared to the codec presented in this thesis when subjectively evaluated. If the coding scheme of this thesis is desired to be used in a real very low bit rate video coding application, depending on the transmission bit rate aimed, following preference of methods must be done. If the transmission bit rate is relatively low split VQ algorithm (i.e., no prediction case) based codec must be used. But, if the transmission bit rate is relatively high codec with GRF model based non-linear must be preferred.

In this thesis, the performance features of a prediction based VQ coding scheme is investigated. As a future study, a coding scheme combining DCT and prediction may be examined. (i.e., DCT takes the place of VQ. For type 0 and 1 blocks nothing changes. For type 2 blocks motion compensated block difference is coded using DCT, and for type 3 blocks prediction error block -residual block after the spatial prediction- is DCT coded instead of VQ.) It is observed that the DCT based interframe coding method (i.e., motion compensated DCT) has a better subjective quality compared to the coding method presented in this thesis. The reason is the spiky errors introduced by the vector quantization. DCT based algorithms give rise to lower SNR but nicer (smoother) visual quality. Therefore DCT in combination with prediction would most probably result in better codec performance. In order to achieve improvement in the performance of the presented codec, a couple of changes may be done to find the optimum values of the system parameters. Block size is one of those. In this study, block size is taken to be 8x8 for the Y component, and 4x4 for the U and V components of the image. One can play with the block size to determine the optimum value. There is a trade off between the compression ratio (hence the transmission bit rate) and the decoded image quality. Increasing the block size would decrease the efficiency of the VQ on the other hand give rise to higher compression ratios. Therefore, an optimum block size with an acceptable degradation in the image quality, can be found by experimentation. More than that variable block size may be used during the coding process, by increasing the block size in the smooth parts of the image and decreasing it in the detailed parts. Another modification to the coding structure introduced in this thesis may be to use another error measure in the vector quantization. In this study error criterion is chosen to be the sum of the absolute differences in the block. Taking the maximum value of the absolute differences in the block to be the error measure would increase the output image quality by forbidding the spiky errors. Finally, pre-filtering and post-filtering may be cascaded to the system to increase the overall performance of the coding method and the image quality at the output. Optimum design of the pre-filter and post-filter is an important issue, in the implementation of the codec that will be used in a real very low bit rate video coding application.

REFERENCES

[1] Project COST211bis, Redundancy Reduction Techniques for Coding of Broadband Video Signals, Final Report.

[2] B. M. Oliver, J. R. Pierce and C. E. Shannon, "The Philosophy of PCM", *Proc. IEEE*, pp. 1324-1331, November 1948.

[3] A. N. Netravali and J. O. Limb, "Picture Coding", *Proc. IEEE*, pp. 366-406, March 1980.

[4] CCIR, "Encoding Parameters of Digital Television for Studies", Rec. 601, 1982.

[5] P. Pirsch, "Design of DPCM Quantizers for Video Signals Using Subjective Tests", *IEEE Trans. on Comm.*, pp.990-1000, July 1981.

[6] B. S. Atal, "Predictive Coding of Speech at Low Bit Rates", *IEEE Trans. on Comm.*, pp.600-614, April 1982.

[7] F. Lukas and F. Kietz, "DPCM Quantization of Color Television Signals", *IEEE Trans. on Comm.*, pp.927-932, July 1983.

[8] D. Anostassiou, J. L. Mitchell and W. B. Pennebaker, "A High Compression DPCM-based Scheme for Picture Coding", *Proc. Inter. Conf. on Comm.*, pp. B.4.5.1-B.4.5.5, June 1983.

[9] H. R. Schindler, "Delta Modulation", *IEEE Spectrum*, pp. 68-79, October 1980.

[10] B. D. Agrawal and K. Shenoi, "Design Methodology for $\Sigma\Delta$M", *IEEE Trans. on Comm.*, pp. 360-370, March 1983.

[11] M. L. Honig and D. G. Messerschmitt, "Comparison of Adaptive Linear Prediction Algorithms", *IEEE Trans. on Comm.*, pp. 1775-1785, July 1982.

[12] W. Zschunke, "DPCM Picture Coding with Adaptive Prediction", *IEEE Trans. on Comm.*, pp. 1295-1301, November 1977.

[13] C. C. Cutler, "Delayed Encoding Stabilizer for Adaptive Coders", *IEEE Trans. on Comm.*, pp. 898-904, December 1971.

[14] H. G. Fehn and P. Noll, "Multipath Search Coding of Stationary Signals with Applications to Speech", *IEEE Trans. on Comm.*, pp.687-701, April 1982.

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

[16] N. Nasrabadi and R. King, "Image Coding Using Vector Quantization : a review", *IEEE Trans. on Comm.*, pp. 957-971, August 1988.

[17] T. S. Huang, "Coding of Two-Tone Images", *IEEE Trans. on Comm.*, pp. 1406-1424, November 1977.

[18] H. Meyr, H. G. Rosdolsky and T. S. Huang, "Optimum Run-length Codes", *IEEE Trans. on Comm.*, pp. 826-835, June 1974.

[19] J. W. Woods and S. D. O'Neil, "Sub-band coding of Images", *IEEE Trans. ASSP*, vol. ASSP-34, October 1986.

[20] T. A. Ramstad, "Sub-band Coder with a Simple Bit Allocation Algorithm : a Possible Candidate for Digital Mobile Telephony", *Proc. ICASSP*, pp. 203-207, May 1982.

[21] C. D. Heron, R. E. Crochiere and R. V. Cox, "A 32-band Sub-band/Transform Coder Incorporating Vector Quantization for Dynamic Bit Allocation", *Proc. ICASSP*, pp. 1276-1279, April 1983.

[22] N. S. Jayant and P. Noll, "Digital Coding of Waveforms", Prentice-Hall, 1984, pp. 510-576.

[23] A. K. Jain, "Fundamentals of Digital Image Processing", Prentice-Hall, 1989, pp. 132-177.

[24] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform", *IEEE Trans. on Computers*, C-23, pp. 90-93, January 1974.

[25] W. A. Pearlman, "Adaptive Cosine Transform Image Coding with Constant Block Distortion", *IEEE Trans. on Comm.*, vol. 38, no. 5, May 1990.

[26] V. A. Vaishampayan and N. Farvardin, "Optimal Block Cosine Transform Image Coding for Noisy Channels", *IEEE Trans. on Comm.*, vol. 38, no. 3, March 1990.

[27] S. Singhal, D. L. Gall and C. Chen, "Source Coding of Speech and Video Signals", *Proceedings of the IEEE*, vol. 78, no. 7, July 1990.

[28] J. Maeng and D. Hein, "A Low Bit-Rate Video Coding Based on DCT/VQ", *Visual Comm. and Image Proc. IV*, vol. 1199, pp. 267-273, 1989.

[29] H. Ghavari, "Multilayer Subband-Based Video Coding", *IEEE Trans. on Comm*, vol. 39, no. 9, September 1991.

[30] Y. Wu and D. C. Coll, "BTC-VQ-DCT Hybrid Coding of Digital Images", *IEEE Trans. on Comm.*, vol. 39, no. 9, September 1991.

[31] Y. Ozturk and H. Abut, "Multichannel Linear Prediction and Applications to Image Coding", *Archive fur Elektronik und Ubertragungstechnik*, vol. 4, pp. 312-320, Sept./Oct. 1989.

[32] C. Manikopoulos, H. Sun and G. Antoniou, "Low Bit Rate Teleconferencing Video Signal Data Compression", *SPIE Visual Communications and Image Processing*, vol. 1199, pp. 504-513, 1989.

[33] H. Schiller and B. B. Chaudhuri, "Efficient Coding of Side Information in a Low Bitrate Hybrid Image Coder", *Signal Processing*, vol. 19, no. 1, pp. 61-73, January 1990.

[34] Simulation Model for Very Low Bitrate Image Coding : SIM2, COST211 ter, Simulation Subgroup, 1992.

[35] R. F. Gonzales and P. Wintz, Addison-Wesley Publications, pp. 375-376, 1987.

[36] A. N. Akansu and J. H. Chien, "Adaptive One Dimensional DCT-VQ for Motion Compensated Video Coding", *Visual Comm. and Image Proc. IV*, vol. 1199, pp. 496-503, 1989.

[37] N. B. Chakraborti and R. Misra, "A Hybrid Hadamard LPC Scheme for Picture Coding", *IEEE Trans. on ASSP*, vol. ASSP-35, no. 3, March 1987.

[38] M. Kanefsky and C. Fong, "Predictive Coding Techniques Using Maximum Likelihood Prediction for Compression of Digitized Images", *IEEE Trans. on Information Theory*, vol. IT-30, no. 5, September 1984.

[39] C. H. Hsieh, P. C. Lu and W. G. Liou, "Adaptive Predictive Image Coding Using Local Characteristics", *IEE Proc.*, vol. 136, Pt. 1, No. 6, December 1989.

[40] V. Cuperman and A. Gersho, "Vector Predictive Coding of Speech at 16 Kbits/s", *IEEE Trans. on Comm.*, vol. COM-33, July 1985.

[41] S. Singhal and B. S. Atal, "Amplitude Optimization and Pitch Prediction in Multipulse Coder", *IEEE on ASSP*, vol. 37, no. 3, pp. 317-327, March 1989.

[42] H. Derin, "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields", *IEEE Trans. on. Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 1, January 1987.

[43] L. Onural, "Vector Gibbs Random Fields and Colored Textures", *The 3. International Symposium on Computer and Information Sciences*, pp. 455-462, 1988.

[44] Y. Linde, A. Buzzo and R. M. Gray, "An Algorithm for Vector Quantizer Desing", *IEEE Trans. on Comm.*, vol. COM-28, pp. 84-95, January 1980.

[45] A Gersho and B. Ramamuthi, "Image Coding Using Vector Quantization", *Proc. ICASSP*, pp. 428-431, May 1992.

[46] A. Gesrho and V. Cuperman, "Vector Quantization : A Pattern-Matching Technique for Speech Coding", *IEEE Comm. Mag.*, December 1983.

[47] A. Gersho, "On the Structure of Vector Quantizers", *IEEE Trans. on Inf. The.*, vol. IT 28, pp. 157-166, March 1982.

[48] B. Ramamurthi and A. Gersho, "Classified Vector Quantization of Images", *IEEE Trans. on Comm.*, vol. 34, no. 11, pp. 1105-1115, November 1986.

[49] N. S. Jayant and P. Noll, "Digital Coding of Waveforms", Prentice-Hall, 1984, pp. 148-150.

[50] B. Ramamurthi and A. Gersho, "Image Coding Using Segmented Codebooks", *Proc. International Picture Coding Symposium*, pp. 105-106, March 1983.