CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

INTERFRAME IMAGE CODING BY ADAPTIVE VECTOR QUANTIZATION

by

Huifang Sun

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Ph.D.
in
Electrical Engineering

OTTAWA, Ontario, 1986

ISBN 0-315-33253-0

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

The University of Ottawa requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

Rate distortion theory, image coding and vector quantization concept are first introduced and reviewed. Two new interframe coding techniques based upon adaptive vector quantization are then presented: three-dimensional block vector quantization and two-dimensional block vector quantization with replenishment. For the three-dimensional case, best results are obtained by first performing the Hadamard transform on a three-dimensional block basis, and then vector quantizing the transformed coefficients. The second algorithm has as its basis two-dimensional block vector quantization, at the frame level, onto which is grafted the concept of adaptive vector quantization and frame replenishment. Each frame is assumed to be represented by a codebook and a codeword label map, and both are selectively updated or replenished. Codebook replenishment involves updating the codebook to track the changes in local statistics. Three efficient strategies are proposed and tested: mean shift, selective codeword replacement and frame adaptive migrating mean (FAMM). Experimental results are provided for a picturephone sequence and a sequence of angiographic x-ray images. Good quality images are obtained at bit rates as low as 0.6 bits/pixel, when all overhead components have been included.

ACKNOWLEDGMENT

I wish to express my gratefulness and appreciation to my thesis advisor, Dr. Morris Goldberg. I cannot thank him adequately enough for the guidance, encouragement and support which he has as generously given me during the course of this research.

The considerable help and encouragement from Dr. W. Steenaart is also greatly appreciated. I would like to express my deep grief at the death of Dr. A. Smith, the member of my advisory committee.

In addition, I wish to thank my collegues, Steve Reed, Li-min Wang, Li Peng and Zhang Jinyun for their encouragement and help. Thanks to the staff of the Department of Electrical Engineering at University of Ottawa for their help. Thanks, too, to Shanghai Aeronautical Radio Research Institute for giving me time on leave study.

A special note of thanks goes to my family. It is hard to imagine that I could finish this long-term work without their support.

Finally, I would like to acknowledge the financial support from NSERC grant, and the test image sequences from CCETT and Compagnie Generale de Radiologie (CGR France).

LIST OF SYMBOL

$A(x,y,t;u,v,w)$	three-dimensional transform kernel
a_3	skewness of histogram
Bc	average number of bits per codeword
Bm	number of bits for block mean term
BR	bit rate
BRl	bit rate for label
BRm	bit rate for block mean
BRo	bit rate for codebook overhead
BRnl	bit rate for new label
BRnm	bit rate for new cluster mean (codeword)
Bs	bit rate for side information
BTC	block truncation coding
Cov	covariance
CT	computation time
c(i)	LPC coefficient
D	distortion
DPCM	differential pulse code modulation
DVQ	direct vector quantization
d(x,y)	distortion between x and y
d_{imax}	maximum distortion of a Voronoi region
$d^j(x,y)$	difference image of j-th and (j-1)-th frames
E	expectation

$E(u,v,w)$	energy distribution in the space u,v and w
FAMM	frame adaptive migrating mean
$F(u,v,w)$	three-dimensional Hadamard transform
$f(x,y,t)$	a digital image sequence matrix
f^j	the intensity value of j-th frame
$g(x,y)$	a continuous image matrix
HT	Hadamard transform
HTVQ	Hadamard transform followed with vector quantization
L	a set of pixels to be used for prediction
$I_N(x,y)$	mutual information between x and y for a block length of N
$I(x,y)$	information portion of a signal
IS-A	a sequence of noise x-ray images
IS-B	a sequence of translation x-ray images
IS-C	a sequence of occlusion x-ray images
j	frame number
K	dimension of vector
LPC	linear predictive coding
LR	label replenishment
M	mean value
Ms	number of training vectors
M_n	n-th order moment
N	codebook size
NMSE	normalized mean-squared error
Nb	number of pixels per block

Nc	number of new labels
Nf	number of pixels per frame
Np	number of pixels in the input source
NTSC	National Television Systems Committee
N(x,y)	noise portion of a signal
O	a region including the vectors lying out of the maximum distortion (d_{imax}).
P(x)	probability of x
P	a constant
p	number of pixels greater than the mean value in BTC coding
Q	a mapping from a vector set to a finite codebook
Q(Y/X)	conditional probability of Y about X
q	number of pixels smaller than the mean value in BTC coding
q(x,y,t,u,v,w)	a parameter of Hadamard transform
R(D)	rate distortion function
R(i)	correlation function
R_i	i-th Voronoi region
r_j	correlation coefficient between j-th and and (j-1)-th frame
S	S transform
T(Y)	probability of the input message set
V	vector set
Var	variance
VQ	vector quantization

$v(n)$	current pixel in predictive coding
W	codebook
w	codeword
X	input source
X	intensity value of coded image
Y	output message
Y_0	a coded value in BTC coding
Y_1	a coded value in BTC coding
3-D	three-dimension
ϕ	central moment
σ^2	variance
ε	symbol for "element of"

LIST OF FIGURES

LIST OF TABLES

CONTENTS

Chapter 1

INTRODUCTION

A digital image is obtained by quantizing a continuous image both spatially and in amplitude. Digitization of the spatial coordinates is called image sampling while amplitude digitization is referred to as gray-level quantization. Suppose that a continuous image is denoted by $g(x_i, y_i)$, where the value or amplitude of g at spatial coordinates (x_i, y_i) is the intensity (brightness) of image at that point. The digital image is then obtained by

$$f(x,y)=g'(x_0+x\Delta x,\ y_0+y\Delta y), \tag{1}$$

where g' is the quantized value of g, x and y are the discrete values 0, 1, 2, ..., and Δx , Δy are the sampling intervals. If this sampling process is extended to a third temporal dimension, a sequence, f(x,y,t), is obtained,

$$f(x,y,t)=g'(x_0+x\Delta x,\ y_0+y\Delta y,\ t_0+t\Delta t) \tag{2}$$

where t are the values 0, 1, 2, ..., and Δt is the sampling interval. The individual images are called frames, and coding schemes which exploit the temporal dimension are called interframe coding. The frames are normally presented at a regular interval of time so that the eye can perceive motion. For ex-

ample, the National Television Systems Committee (NTSC), specifies a temporal sampling rate of 30 frames per second and interlace 2 to 1. At this frame rate, there is considerable correlation between successive frames of the picture; this redundancy can be exploited to achieve image compression.

There are a number of applications where such coding could be used. These include point-to-point transmission of commercial television programs, satellite transmission, video teleconferencing and medical image transmission. At present, a large body of well organized knowledge has been accumulated about the coding of image sequences for bandwidth compression [1-2]. Such coding techniques exploit the following factors:

1. An image is normally highly correlated spatially; that is, there is a strong dependence between the values of neighboring pixels. This dependence can be regarded as statistical redundancy of the image scene.
2. There is a high temporal correlation between successive frames; that is, similarity exists between successive frames.
3. It is possible to reduce the data rate by introducing larger degradations in those areas of the picture where human visual sensitivity is low and reproducing accurately those areas where the visual sensitivity is high.

Coding techniques fall into two general categories, intraframe coding and interframe coding. Intraframe coding schemes, such as differential pulse-code modulation (DPCM), transform are easily extended to interframe coding with some modifications [1]. Some coding schemes, for example frame replenishment coding [3], motion-compensated predictive coding [4], are only applicable to interframe coding.

Most coding techniques are based upon scalar quantization; that is, each component is individually quantized. However, a fundamental result of Shannon's rate distortion theory indicates that better performance can always be achieved by vector quantization, in which several components are quantized simultaneously [5]. In this thesis, new algorithms for image sequence coding based upon vector quantization are proposed. In vector quantization, the image data is first processed to yield a set of vectors. Then, a codebook of representative vectors (codewords) is generated using an iterative clustering algorithm such as K-means [6], or the generalized Lloyd clustering algorithm proposed by Linde, Buzo and Gray [7]. The vectors in the set are then individually quantized to the closest codeword of the codebook. Compression is achieved by replacing the representative vector (codeword) by a label. Reconstruction of the images can be implemented by table look-up techniques; the label is simply used as an address to a table containing the representative vectors (codewords). Vector quantization techniques have recently been used with some suc-

cess in speech coding [8-10], for example, for a given distortion, the required bit rates can be decreased by 73% compared with scalar quantization [8]. Reports on their use for image coding, both intraframe and interframe, have also appeared [11-24].

In this thesis, two new algorithms based upon vector quantization are proposed: three-dimensional block vector quantization, and two-dimensional block vector quantization with replenishment. The first algorithm is an extension of the two-dimensional block vector quantization proposed in [16]. The second algorithm has as its basis two-dimensional block vector quantization, at the frame level, onto which is grafted the concept of adaptive codebook replenishment and frame replenishment.

The rest of this thesis is organized as follows. Chapter two presents the mathematical basis of image coding and reviews research in image coding. Chapter three then describes the general principle of vector quantization, and presents some results on intraframe image coding using vector quantization.

Three-dimensional block vector quantization is then introduced in chapter four. Results on using spatial-temporal domain vector quantization on a 2×2×2 and 3×3×2 block basis are first reported. There follows a scheme in which the three-dimensional Hadamard transform is first applied on a 4x4x4 block

basis and vector quantization is then performed on a subset of the transformed coefficients. In this latter scheme, the components in the vectors are decorrelated in both, spatial and temporal, directions. Improved results as compared to spatial vector quantization are obtained, however, at the expense of longer training sets and increased computation.

Chapter five presents algorithms which combine the concept of vector quantization with frame replenishment. In vector quantization, each vector is encoded by two distinct pieces of information: the label and the corresponding codeword. As both these pieces can be permitted to change independently, the concept of adaptivity can be applied to each. The underlying rationale is that a separate codebook tailored to each frame may be more efficient than one designed for the entire sequence. A smaller codebook, more representative of the local statistics, could conceivably give the same distortion at lower bit rate. However, the need for updating the codebook on a frame-to-frame basis increases the overhead and so could nullify these savings. Two efficient strategies for codebook replenishment, mean shift and mean shift with selective codeword replacement, are introduced.

In chapter six, the Frame Adaptive Migrating Mean (FAMM) codebook replenishment algorithm, is presented. The main idea behind this algorithm is that a new codebook is generated by using the previous codebook as seeds for the clustering

algorithm employed in codebook generation; the number of iterations depends upon the type of changes occurring in the input image sequence. This number is derived from one of two statistics:

(a) first-order statistical parameters of the difference images calculated from successive frames,

(b) the correlation coefficient between the successive frames.

Experimental results are provided for a sequence of angiographic x-ray images and a picturephone sequence. It is shown that good quality images are obtained at bit rate as low as 0.6 bits/pixel, when all overhead components have been included.

The last chapter briefly summarizes the main results and lists some suggestions for further investigation.

Chapter II

IMAGE CODING

This chapter first presents the mathematical basis for image data compression, rate/distortion theory, and then reviews traditional image coding techniques, for intraframe and interframe image coding.

2.1 RATE DISTORTION FUNCTION

The principal goal in the design of an image coding system is to reduce the transmission rate requirements of the image source, subject to some image quality constraint. There are only two ways to accomplish this goal: reduction of the statistical redundancy and psychophysical redundancy of the image source [1]. An image source is normally very highly correlated both spatially and temporally; that is, there is strong dependence between the values of the individual pixels. This dependence can be regarded as statistical redundancy of the image source. If the images to be coded in an image transmission system are to be viewed by a human observer, then the perceptual limitations of human vision can be exploited to reduce transmission requirements. Human observers are subject to perceptual limitations in amplitude, spatial resolution, and temporal acuity. By proper design of the coding system, it

is possible to discard information without affecting perception, or at least, with only minimal degradation. Summarizing, there are two factors that make data compression possible: the statistical structure of the source and the fidelity requirements of the end user.

This intuitive explanation of how redundancy and distortion can be used to reduce source rate is made more precise by Shannon's rate distortion theory which addresses the problem of how to characterize both the source and the distortion measure.

Consider the block diagram of a model of a typical communication system depicted in Fig.1. The data source generates information that the system must communicate to the user. Both the encoding and decoding can be separated into two distinct parts: source coding and channel coding. The function of the source coding is to remove source redundancy, whereas the function of the channel coding is to insert controlled redundancy to combat noise. The problem addressed by rate distortion theory is the minimization of the channel capacity requirement while maintaining the average distortion at or below an acceptable level.

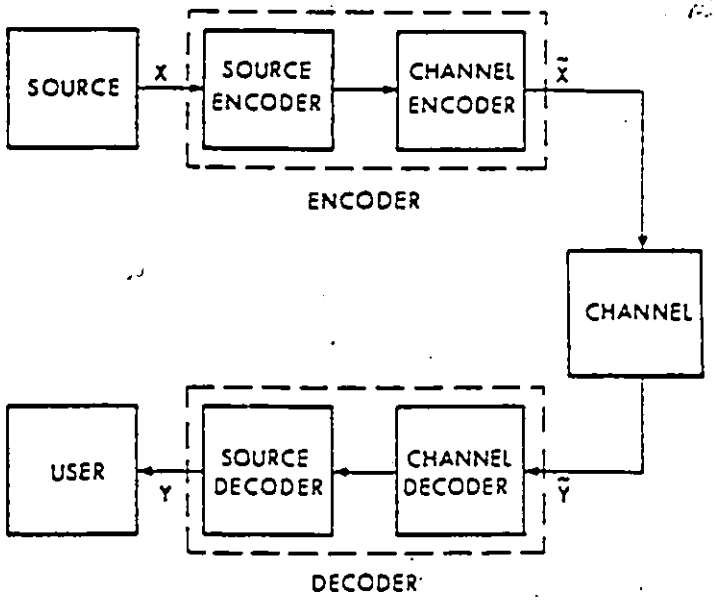


Fig.1 A Model of a typical communication system [25].

The rate distortion function R(D) is the minimum average rate (bits/element), and hence minimum channel capacity, required for a given average distortion level D. To describe this in more quantitative way, suppose that the source is a sequence of intensity values of discrete picture elements (pixels), and that these levels are encoded by successive blocks of length N. Each block of elements is then described by one of a denumerable set of messages $\{X_i\}$ with probability function $P(X_i)$. Any given system is now described mathematically by the conditional probability $Q(Y_j/X_i)$ of a message $\{Y_j\}$ at the output of the decoder given a source input, X_i. The probability of the output message set is, therefore,

$$T(Y_j) = \sum_i P(X_i)Q(Y_j/X_i) \quad (3)$$

The information transmitted is called the average mutual information between Y and X, and defined, for a block of length N, as follows:

$$I_N(X,Y) = \sum_i \sum_j P(X_i)Q(Y_j/X_i)\log\frac{Q(Y_j/X_i)}{T(Y_j)} \quad (4)$$

In the case that encoding is error free so that Y=X, then

$$Q(Y_j/X_i) = \begin{cases} 1, & j=i \\ 0, & j\neq i \end{cases} \quad (5)$$

$$T(Y_j)=P(Y_j)$$

and

$$I_N(X,Y) = -\sum_i \sum_j P(X_i)\log_2 P(X_i) = H_N(X) \quad (6)$$

the Nth-order entropy of the source. The optimum-error free coding requires at least $H_N(X)$ bits [26].

If, as in the more general situation, source coding results in error so that $Y \neq X$ at least some of the time, then

$$I_N(X,Y)=H_N(X) - H_N(X/Y) \tag{7}$$

where $H_N(X/Y)$ is the entropy of the source, given the observed decoder output Y. As the entropy is a positive quantity, the source entropy is the upper bound to the mutual information; that is,

$$I_N(X,Y) \leq H_N(X) \tag{8}$$

Let d(X,Y) be the average distortion between X and Y, then the average distortion per pixel is:

$$D(Q)=\frac{1}{N}E\{d(X,Y)\}=\frac{1}{N}\sum_i\sum_j d(X_i,Y_j)P(X_i)Q(X_i/Y_j) \tag{9}$$

The set of all conditional probability assignments, Q(Y/X), that yield average distortion less than or equal to D*, can be written as:

$$\{Q: D(Q) \leq D^*\} \tag{10}$$

The N-block rate distortion function is then defined as the minimum of the average mutual information, $I_N(X,Y)$, per pixel:

$$R_N(D^*) = \underset{Q:D(Q)\leq D^*}{\text{Min}} \frac{1}{N} I_N(X,Y) \qquad (11)$$

The limiting value of the N-block rate distortion function is called simply the rate distortion function,

$$R(D^*) = \lim_{N\to\infty} R_N(D^*) \qquad (12)$$

It should be clear from the above discussion that Shannon's rate distortion function R(D) is a lower bound on the transmission rate required to achieve average distortion D. The rate at which a source produces information subject to a requirement of perfect reproduction is called the entropy of the source. It follows that the rate distortion function is a generalization of the concept of entropy. Indeed, if the distortion measure is such that a perfect reproduction is assigned zero distortion, then R(0) is equal to the source entropy H(X). Shannon's coding theorem states that one can design a coding system with rate only negligiblly greater than R(D) which achieves the average distortion D. As D increases, R(D) decreases monotonically and usually becomes zero at some finite value of distortion [25]. A typical rate distortion function is shown in Fig.2.

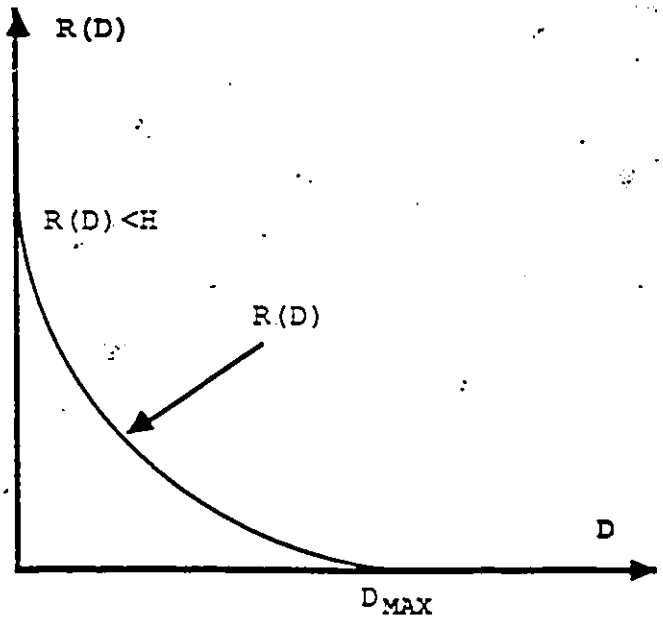


Fig.2 A typical rate distortion function.

The rate distortion function R(D) specifies the minimum achieveable transmission rate required to transmit an image with average distortion level D. The main value of this function in a practical application is that it potentially gives a measure for judging the performance of a coding system. However, this potential value has not been completely realized for image transmission. There are two reasons for this. First of all, there do not currently exist tractable and faithful mathematical models for an image source. The rate distortion function for Gaussian sources under the squared-error distortion criterion can be found, but is not a good model for images. The second reason is that a suitable distortion measure D, which accords with subjective evaluation of image quality, has not be found. In spite of these drawbacks, the rate distortion theorem is still a mathematical basis for comparing the performances of different coding systems.

2.2 TECHNIQUES FOR INTRAFRAME CODING

Historically, the techniques used to reduce the bandwidth required for transmitting digitized pictures have generally followed two paths: spatial domain encoding or transform domain encoding [27]. The goal is to process the correlated data (image picture elements) in such a way that an uncorrelated set of data results. The processed data are then quantized and coded, so that data reduction is achieved [1]. Generally, image coding can be categorized into still image (intraframe)

coding and image sequence (interframe) coding [2,27]. Among current intraframe coding techniques, there are predictive coding, transform coding, hybrid coding, block truncation coding (BTC), and interpolating coding [2,27]. Each class can be further divided into non-adaptive and adaptive depending upon whether the parameters of the coder change as a function of image data that is being coded. These coding techniques are briefly reviewed below.

2.2.1 Predictive coding

Predictive coding methods exploit the spatial redundancy in the input source. As neighbouring pixels tend to be correlated, a given pixel value can be predicted from previously transmitted pixels [1]. The prediction error, the difference between the predicted value and the actual value of the pixel, is then quantized into a set of discrete amplitude levels. These levels are represented as binary words of either fixed or variable word length and are sent to the channel coder for transmission. Thus, a predictive coder has three basic components: 1) predictor, 2) quantizer, 3) code assigner. The most common form of predictive coding is differential pulse code modulation (DPCM), the block diagram of which is shown in Fig.3. The different blocks contained are now discussed in turn.

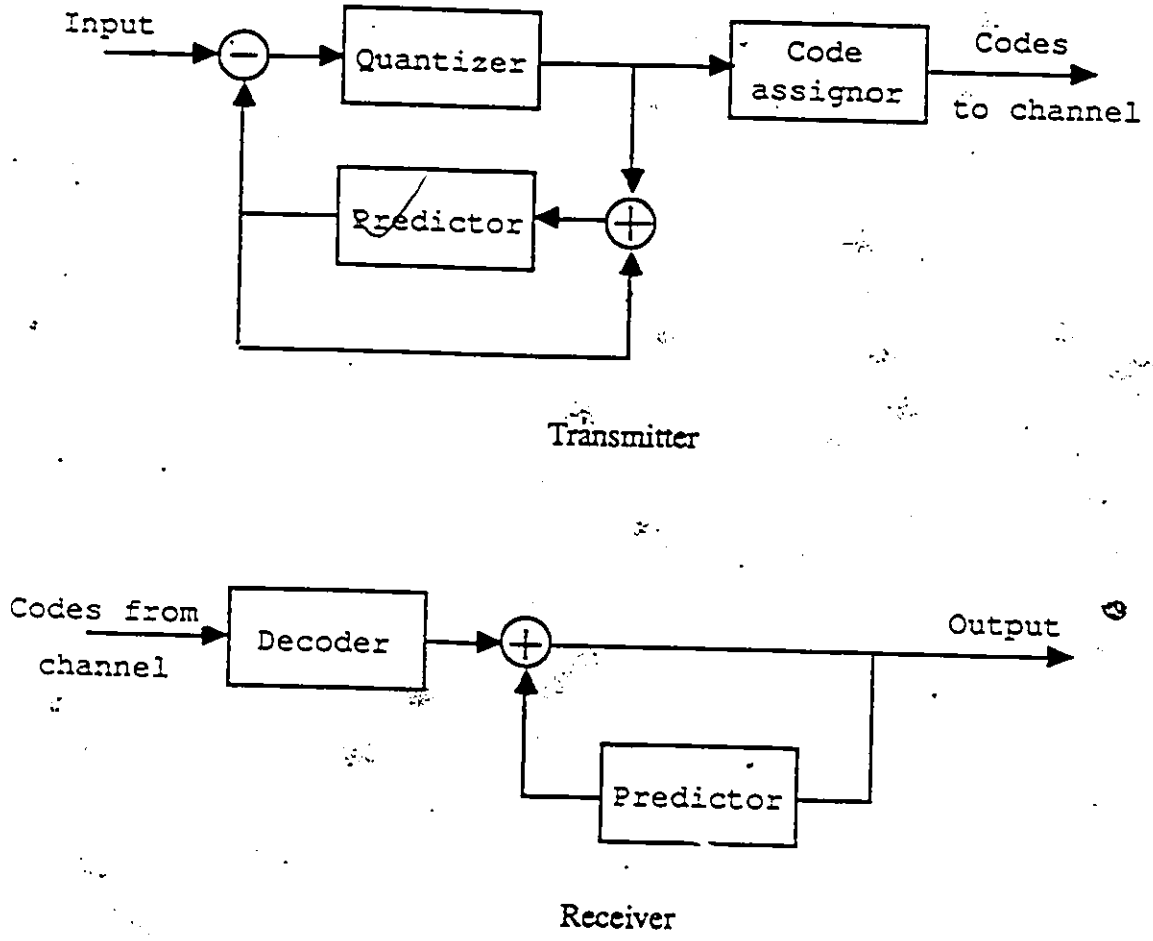


Fig.3 Block diagram of DPCM system.

Predictor

Predictors can be classified as linear and non-linear.

a) Linear predictor

Suppose that the current pixel is denoted by v(n), and {v(n-i), $i \in I$} is a set of pixels in the neighbourhood of v(n) which have been previously transmitted. The set I specifies the pixels to be used for the prediction. Fig.4 shows a typical set I, comprising pixels chosen from successive lines. A linear prediction of v(n) based on {v(n-i)} takes on the following form:

$$\hat{v}(n) = \sum_{i} c(i)\, v(n-i) \qquad (13)$$

The best mean-square predictor is obtained by choosing c(i), so as to minimize $E\{(v(n)-\hat{v}(n))^2\}$ [27]. If v(n) is a sample of a wide sense stationary process, a fixed predictor results, with coefficients given by the solution to the normal equations

$$\sum_{i \in I} c(i)\, R(i-k) = R(k) \quad , \quad k \in I, \qquad (14)$$

where R(k) is the correlation function.

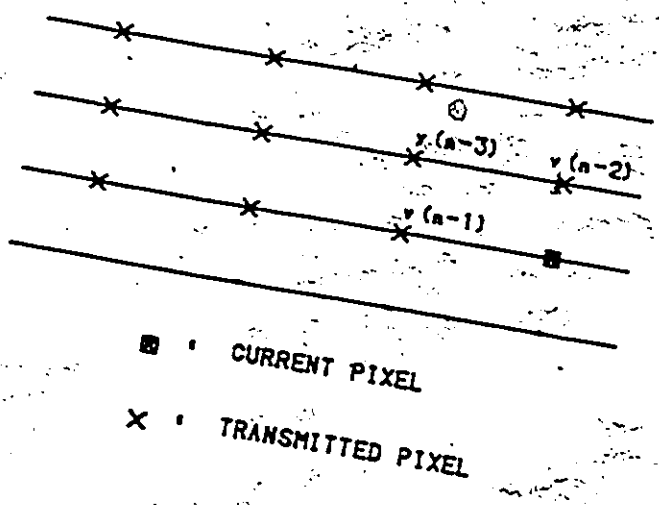


Fig.4 Pixels in the neighbourhood of the current pixel under consideration used for prediction.

b) Non-linear predictors [28]

The fact that images contain many edges and contours has led to the use of nonlinear predictors. These predictors attempt to determine the direction of contours in the image, and choose the prediction accordingly. The prediction is still a function of previously transmitted pixels, but this function is nonlinear. The basic idea is illustrated by the following simple intraframe predictor [28],

$$\hat{v}(n) = \begin{cases} v(n-1), & \text{if } |v(n-1)-v(n-3)| \geq |v(n-2)-v(n-3)| \\ v(n-2), & \text{otherwise} \end{cases} \quad (15)$$

In this predictor, either the previous pixel on the same line or on the previous line is used for prediction, and the switching is done by the surrounding line and pixel difference as shown in equation (15) and Fig.4.

Quantization

Quantization can be designed purely on a statistical basis or by using certain psychovisual measures [29]. The quantizer may be fixed or may be allowed to adapt to local picture properties. Given the type of code assignment, the general quantizer design principle is to minimize some measure of picture distortion subject to the constraint on the transmission rate. Three types of degradations can arise: granular noise, edge busyness and slope overload. Adaptive quantizers use statis-

tical or psychovisual criteria to change their characters, which involve switching between a number of fixed quantzers, or changing the basic step size of a given quantizer. For example, in areas of the picture where there is very little activity, a fine quantizer can be used, and in areas with more activity, the quantizer can be coarser.

Code assignment

Variable-word-length coding is commonly used for code assignment [4], and exploits the observation that the frequency of occurrence of the quantizer output levels is not uniform. Thus, variable length code words can be used to reduce the bit rate by representing the levels that occur more often with a smaller length word. Typically, the Huffman code [30], the average bit rate of which is very close to the entropy of the quantizer output, is used.

One of the problems with the use of variable length codes is that the output rate from the source code changes with local image content. In order to send such a signal over a constant bit rate channel, the source coder output must be temporarily held in a buffer; the inputs arrive at a nonuniform rate, but are transmitted at a uniform rate.

2.2.2 Transform coding

In transform coding [1], an image is first divided into blocks of equal size, and then a transform is applied to each block individually to yield a set of coefficients. A transform, such as the discrete cosine transform (DCT) [31], Fourier transform (FT) [32], or Hadamard transform (HT) [33], is used so that the correlation between the coefficients is decreased. The coefficients are then quantized and coded for transmission. At the receiver, the data stream is then decoded to yield the transform coefficients and the inverse transformation is applied to reconstruct an estimate of intensities of pixels. A block diagram of the process is shown in Fig.5. Compression is achieved by discarding the coefficients which contain little energy and by variably quantizing the remaining coefficients. The number of bits allocated to a coefficient is proportional to its energy (variance). In designing a transform coding system the following parameters must be selected: block size, type of transformation, the coefficients to be transmitted, the bit assignment and quantization strategy.

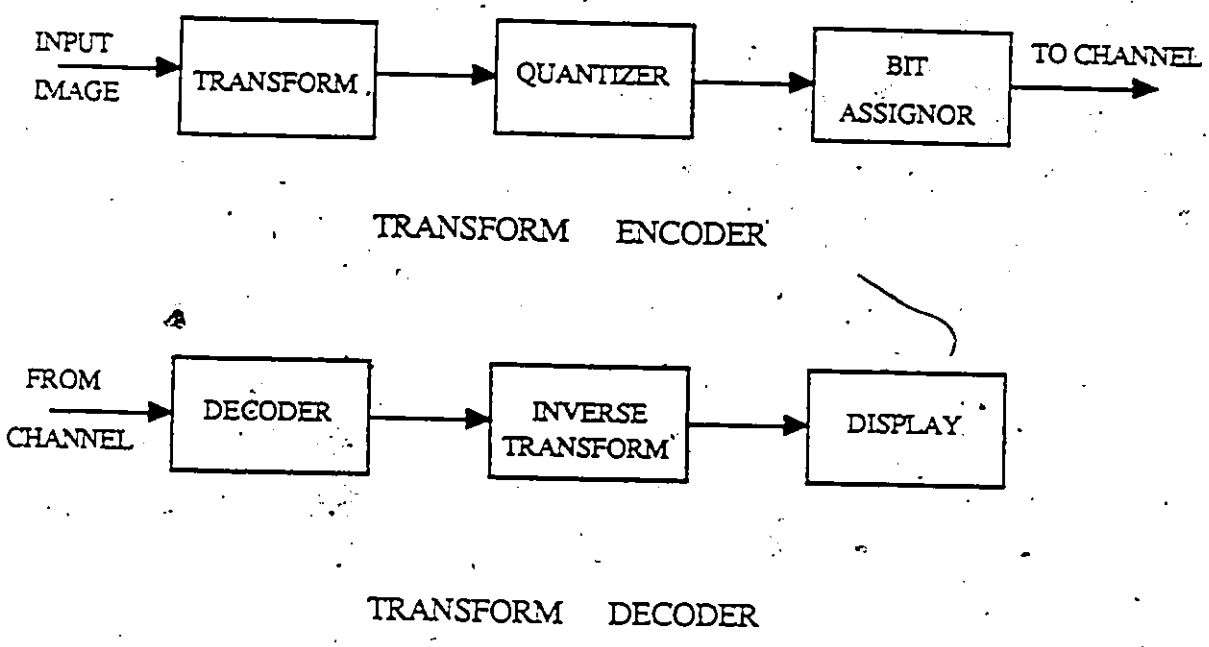


Fig.5 Block diagram of a block transform coding system.

Types of transform:

There exist many different types of unitary transforms, including the Fourier, sine, cosine, Hadamard and Karhunen-Loeve transforms [2,27]. The optimum transform, in a least-mean-square sense, is the Karhunen-Loeve (KL) transform. However, due to the large amount of computation as well as other limitations in implementing this transform, its use is currently impractical. The choice between these transforms is governed by the trade-off between implementation complexity and performance [27].

Adaptive transform coding:

The parameters of a transform coding system can be updated to match the statistics of the images being coded. Since the image statistics may be highly nonstationary, adaptation can increase the coding efficiency. A simple and effective adaptive approach is to adapt the bit assignment to changes in statistics. For example, the image blocks can be classified into several categories and more bits are allocated to the blocks which have large activity and fewer bits to the ones having lower activity. This results in a variable average rate from block to block, but gives a better utilization of the total bits over the entire ensemble of image blocks [35].

2.2.3 Hybrid transform/DPCM coding

Intraframe hybrid coding [36] exploits the horizontal and vertical correlation within an image by the use of a unitary transform cascaded with a DPCM operator. Typically, a one-dimensional transform is first taken along rows in the image, and a set of parallel DPCM linear predictive coders are then applied to the transformed coefficients in a columnwise fashion.

2.2.4 Block truncation coding

Block Truncation Coding (BTC) [37] uses a moment preserving quantizer. For a two dimensional image, this method divides the image into n by n pixel blocks (typically n=4), each pixel of which is individually quantized into two levels, Y_0 and Y_1, in such a way that the block sample mean M and variance σ^2 are preserved. If $x_1, x_2, \ldots, x_N$ are the original sampled values within a given block, then the sample mean is defined as:

$$M = \frac{1}{N} \sum_{i=1}^{N} x_i$$

and sample variance is defined as

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} x_i^2 - M^2 = \overline{x^2} - M^2$$

It has been shown [37] that , the first two moments of x, M and σ^2 , can be preserved using quantizer levels

$$Y_0 = M - \sigma\sqrt{q/p} \quad , \text{ and } \quad Y_1 = M + \sigma\sqrt{p/q} \quad ,$$

where q and p are the number of pixels above and below the sample mean, respectively. For each block, M,σ , and a bit-plane (one bit per pixel to select quantizer level Y_0 or Y_1) are transmitted.

2.2.5 Interpolative coding

In interpolative coding, a subset of pixels are transmitted and the remaining pixel values are then interpolated [38]. As an example, Fig.6 shows a case in which there is a 2:1 subsampling of pixels along each line. The subsampled pixels are staggered from one line to the next and are interpolated by a four-way average as shown by the arrows; for example, pixel A is interpolated by averaging pixels B, C, D, E. Adaptivity can be achieved, for example, by switching between horizontal and vertical averages depending upon the type of local correlation [39].

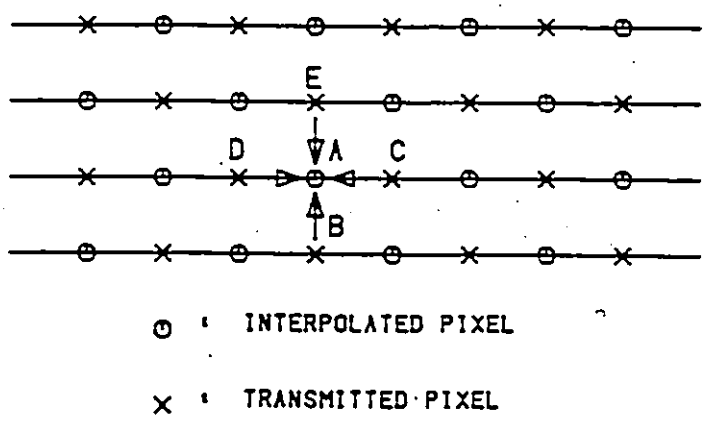


Fig.6. An example of interpolative coding using 2:1 horizontal subsampling, which is staggered from one line to the next. Pixel A is interpolated by averaging pixels B, C, D, and E.

2.3 TECHNIQUES FOR INTERFRAME CODING EXTENDED FROM INTRAFRAME CODING

In interframe coding, both spatial and temporal redundancies are exploited for data compression. In general, interframe coding subsumes some intraframe coding. Most intraframe coding schemes can be extended to interframe coding with some modifications. In addition, there are some schemes which are only for interframe coding. One such technique is based upon the concept of frame replenishment [3]. Since this technique is grafted onto vector quantization in this thesis, it is described in further detail in the next section.

2.3.1 Predictive coding with motion-compensation

In interframe predictive coding, the value of a given pixel is estimated (predicted) from a combination of pixels drawn from the present and previous frames. By transmitting only the error in the prediction, a sequence with lower entropy results [1].

One class of adaptive predictors for interframe coding takes into account object motion. A time-varying image is assumed to consist of a number of objects with different motions superimposed on a background. If the motion trajectory of each pixel can be predicted, motion-compensated prediction can be used [40-42]. As an example, a block diagram of a motion-compensated DPCM coder is shown in Fig. 7.

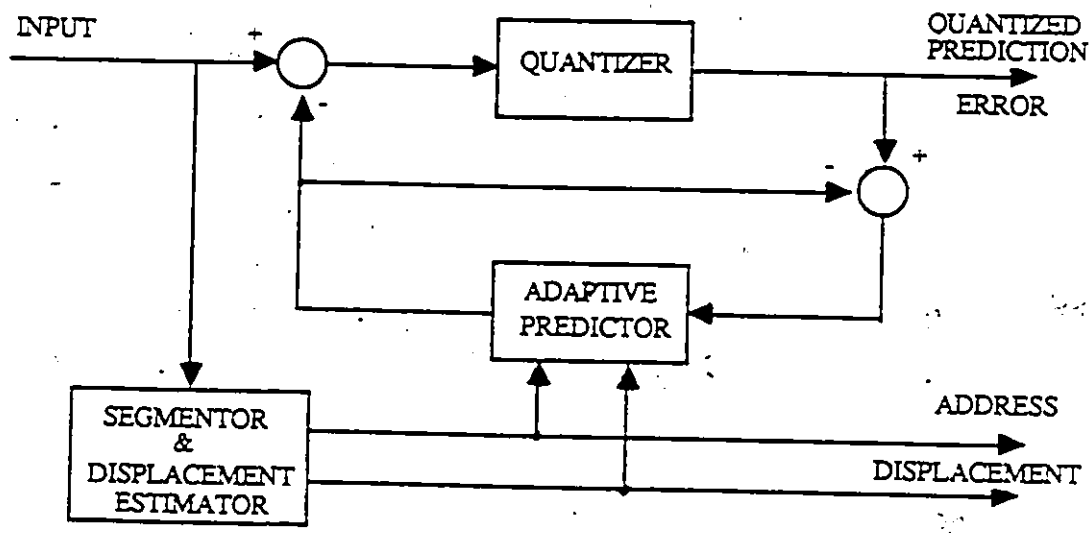


Fig.7 Block diagram of a motion-compensated DPCM coder [66].

The three steps involved in the motion-compensated predictive DPCM coding are as follows:

1. The image is first segmented into stationary and changing areas, and the displacements of moving objects in the changing areas are then estimated.
2. The motion-compensated predictor is generated by the displacement estimates.
3. The prediction error and side information are coded separately.

Approaches to estimating the motion or displacement between successive frames include correlation or matching, and differential techniques [44]. The matching techniques find displacement values which minimize some measure of the difference between the current frame and the previous frame. These techniques require considerable computation, since many possible displacement values must be evaluated. The differential techniques use the derivatives of the image sequence function with respect to time to estimate the velocity of moving objects.

A variant of motion-compensated prediction is the block-structured motion-compensated coder [40-42]. In this scheme, the scene is divided into rectangular blocks, and a single shift is estimated for each block. The block may be segmented, further, into changing and stationary areas. The previous frame is shifted, and interpolated to form the prediction for that block. A second variant operates on a pixel basis. The displacement estimate is recursively updated for each pixel.

This update is based upon only previously transmitted neighbouring pixels, so that no explicit displacement estimate need be transmitted [40].

The side information is used to address the stationary and changing areas. If the motion-compensation has been successful, many of the quantized prediction errors will be zero, and a variable length code assignment could be profitably used. [4].

2.3.2 Interframe transform coding

In interframe transform coding, a sequence of image frames, f(x,y,t), undergoes a three-dimensional transform in blocks of J×K×L pixels according to the formula

$$F(u,v,w) = \sum_{x=0}^{J-1} \sum_{y=0}^{K-1} \sum_{t=0}^{L-1} f(x,y,t)\, A(u,v,w\ x,y,t) \qquad (16)$$

to produce a corresponding sequence, F(u,v,w), in the transform domain, where A(u,v,w;x,y,t) represents a unitary transform kernel, and (J,K,L) denotes the block size in row, column and temporal directions. As for intraframe transform coding, data reduction results by selecting a subset of the transform coefficients, which are then quantized and coded for transmission.

In an image sequence, the local statistics of the input source may be continually changing, therefore, adaptive strat-

egies have been proposed to improve the performance of interframe transform coding schemes [45]. One such example is proposed by Jain [34], in which measures of spatial and temporal activity have been used for adaptive selection and quantization of the coefficients. The image blocks are classified into several categories and a larger number of bits is allocated to the blocks that have large activity.

2.3.3 Hybrid interframe transform/DPCM coding

In interframe hybrid coding [1], typically, a two-dimensional intraframe transform is first performed on spatial blocks within each frame. A linear predictive DPCM coder is then applied in the temporal direction to each resulting transform coefficient.

2.3.4 Interframe BTC coding

The two-dimensional BTC scheme can be extended to three-dimensional blocks (two spatial, one temporal) [47]. However, large movements across block boundaries cause reduced temporal correlation and degrades the performance. This problem is overcome by using a fast registration method [47]. The areas where no movement across the background are registered and coded using a three-dimensional BTC coder, while changing areas (non-registered areas) are coded by a two-dimensional BTC coder.

2.3.5 Interframe interpolative coding

In interframe interpolative coding, the interpolations operate not only in the horizonal and the vertical direction, but also in the temporal direction. Such an algorithm is proposed by Limb and Bowen [48].

2.4 FRAME REPLENISHMENT CODING

An interframe coding technique which explicitly exploits the similarity between frames is frame replenishment coding. This technique is based upon detecting the areas of changes and then updating or replenishing only these areas. In 1969, Mount [3] demonstrated a conditional frame replenishment scheme, where only the information necessary to describe the intensity of pixels that changed between successive frames is transmitted. The implementation used two frame memories, one at the receiver and another at the transmitter, to store the required reference images. The incoming signal is compared on a pixel-by-pixel basis with the reference frame. When the magnitude of the frame-to-frame difference is greater than a certain threshold, it is regarded as significant, and the new signal value is replaced in both image memories. Thus, the receiver memory is made to track the transmitter memory.

The basic steps involved in frame replenishment coding systems [3] are as follows:

1. Segmenting each frame into two parts, one part which is the same as the previous frame and the second part which has changed from the previous frame.
2. Transmission of two sets of information, (a) addresses specifying the location of the pixels in the changing area, and (b) information by which the intensity values of the changing pixels can be updated or replenished.
3. Matching the coder bit rate to the channel rate. Since the amount of information to be transmitted varies as a function of time, a buffer is required to smooth the data rate prior to transmission over a constant bit rate channel.

It is noted that, in frame replenishment coding, two distinct sets of information must be sent, the position of pixels to be replenished and the new intensity values. A number of modifications to the basic conditional frame replenishment scheme have been proposed, which address the problem of coding these sets of information. Candy et al. [40] proposed a method of more efficiently addressing the pixels in clusters, instead of individually. An alternate approach was proposed by Pease and Limb [50] in which the pixels in the changing areas are subsampled by 2:1 along the horizontal direction. Certain studies [4,51] have demonstrated that the pixels in the changing areas need not be transmitted independently, and can be coded by linear predictive coding.

2.5 COMPARISON OF IMAGE CODING METHODS

The previous sections have reviewed various image coding methods. This section presents a comparison of these image coding methods from the point of view of implementation, bit rate, complexity, and noise immunity. These are summarized in Table 1.

2.5.1 Predictive coding

The advantages of predictive coding are that the equipment complexity and the encoding delay are minimal compared to the transform systems. DPCM systems exploit both spatial and temporal correlation in the picture data, but generally require a bit rate of at least 2 bits/pixel. Reconstruction quality, however, is excellent. The overriding limitations of DPCM are the extreme sensitivity to the picture statistics, and the propagation of channel errors.

2.5.2 Transform

The unitary transform preserves the entropy of the source, while decorrelating the data. This method provides a low bit per pixel ratio, distributes the coding degradation, and is fairly insensitive to picture statistics and channel disturbances. The most significant disadvantages of the transform techniques are the amount of computation and storage required. Published results show that good image reconstruction quality

is obtained at bit rates less than 1.0 bits/pixel for intraframe coding [52], and 0.5 bits/pixel for interframe coding [53].

2.5.3 Hybrid coding

The performance characteristics of hybrid coding lies between that of transform and DPCM coding [36]. For example, it is less sensitive to channel errors than DPCM, but is not as robust as transform coding; it is also less complex than transform, but is not as simple as DPCM. Good results at moderate bit rates can be obtained by this scheme. The problems of picture statistics dependencies and error propagation still exist. This is due to the fact that the transformed coefficients are non-stationary processes, and implies that a constant set of DPCM coefficients cannot be optimal for all time and for all scenes. Adaptive predictive coding can alleviate this undesirable characteristic [36].

2.5.4 BTC coding

Block truncation coding (BTC) is a simple and effective technique for compression of image sequence [37]. Rates of 1.1 to 1.625 bits/pixel can be achieved for intraframe coding, and 0.9 to 1.4 bits/pixel for interframe coding [47]. One essential problem of BTC is that edge reproduction tends to be ragged. This can result in a blocky appearance in the recon-

structed image. For interframe coding [47], the system requires three frames of random access memory, buffering memory, and various one-bit frame memories for registration, differencing, and moving area detection.

2.5.5 Interpolative coding

The performance of interpolative coding depends upon the function of interpolation. High-order polynomial functions can improve the reconstructed quality, however, the complexity involved grows rapidly with the degree of the polynomial [48]. It appears that in terms of the number of bits required for a given quality, interpolative coding shows an improvement over nonadaptive DPCM and transform coding; but it appears to be inferior to both adaptive DPCM as well as transform coding [27]. This technique is effective in reducing discrete impulse noise since the interpolative process can be considered as a smoothing filter. Simulation studies [39,48] indicate that good quality reconstructed image can be obtained at the bit rates as low as 1.0 bits/pixel for intraframe case and 0.96 bits/pixel for interframe case, but the coding systems are quite complex.

2.5.6 Frame replenishment

Frame replenishment coding efficiently exploits the similarities between frames. It also tries to make use of subjec-

tive redundancies resulting from the inability of viewers to see spatial or temporal detail. In conditional replenishment, the data to be transmitted is proportional to the amount of movement in the scene. Thus, buffers are required to smooth the data rate prior to transmission. In the event that too much data is produced, methods are needed to reduce the data generation rate in order to prevent buffer overflow. Simulations of frame replenishment coding systems have been reported at rates of about 1.0 bits/pixel with quality rated as excellent except for periods of excessive motion [40].

Table 1 Comparison of image coding methods

Method	Typical rates (bits/pixel)	Comments
Predictive		
Intraframe DPCM	2.0-3.0	simple to implement, excellent reconstruction quality at 2 bits/pixel, sensitive to data statistics, error propagation, low channel noise immunity [1,2,27].
Infraframe DPCM (adaptive)	1.0-2.0	
Interframe DPCM	1.0-1.5	
Interframe DPCM (adaptive)	0.5-1.0	
Transform		
Intraframe	1.0-1.5	high performance, less sensitive to data statistics, channel and quantization error distributed over blocks, high channel noise immunity, high hardware complexity [1,2,27].
Intraframe (adaptive)	0.5-1.0	
Interframe	0.5-1.0	
Interframe (adaptive)	0.2-0.5	
Hybrid		
Intraframe	1.0-2.0	performance close to transform at moderate rate, complexity lies midway between transform and predictive [1,2,27,36].
Intraframe (adaptive)	0.5-1.5	
Interframe	0.5-1.0	
Interframe (adaptive)	0.25-0.5	
BTC		
Intraframe	1.1-1.625	simple to implement, bit map for each pixel is required [37,47].
Interframe	0.9-1.4	
Interpolative		
Intraframe	1.0 -2.0	performance lies midway between non-adaptive and adaptive transform and DPCM, effective in reducing impulse noise, high complexity [27,39].
Interframe	0.96-1.5	
Frame replenishment	0.25-1.0	high performance at rate 1 bit/pixel, two frame memories and buffers required [1,3]

Chapter III

PRINCIPLE OF VECTOR QUANTIZATION FOR IMAGE CODING

Vector quantization is an effective technique for performing data compression. A tutorial by Gersho and Cuperman [16] presents a review of vector quantization for speech coding. A broader and more comprehensive review of vector quantization is given by Gray [5]. In this chapter, the basic concept of image coding using vector quantization is first described in section one. Section two then briefly reviews image vector quantization. Section three follows with an example of intraframe coding which combines adaptive linear predictive coding (LPC) with vector quantization.

3.1 BASIC PRINCIPLE OF VECTOR QUANTIZATION

An N-level vector quantizer, Q, is a mapping from a K-dimensional vector set, {V}, into a finite codebook, $W = \{w_1, w_2, \ldots, w_N\}$.

$$Q: \vec{v} \rightarrow W \tag{17}$$

In other words, it assigns to an input vector, v, a representative vector (codeword), w. The vector quantizer, Q, is completly described by the codebook, $W = \{w_1, w_2, \ldots, w_N\}$, together with the disjoint partition, $R=\{R_1, R_2, \ldots, R_N\}$, where

$$R_i = \{v : Q(v)=w_i\} \qquad (18)$$

and w and v are K-dimensional vectors. The partition should ideally minimize the quantization error [46].

A block diagram of the various steps involved in vector quantization, as applied to image coding, is depicted in Fig.8. The first step in image vector quantization is the decomposition of the image into a set of vectors, {V}. In the second step, a subset of {V}, {T}, is chosen as a training set. In the third step, a codebook is generated from the training set {T}, normally with the use of an iterative clustering algorithm. The quantizer or coding step involves searching, for each input vector, the closest codeword, w, in the codebook W; the corresponding label of this codeword is then transmitted. Thus, the design decisions in implementing image vector quantization are as follows,

(a) vector formation,

(b) training set generation,,

(c) codebook generation,

(d) quantization.

In this thesis, the first three aspects are of particular concern with respect to interframe coding.

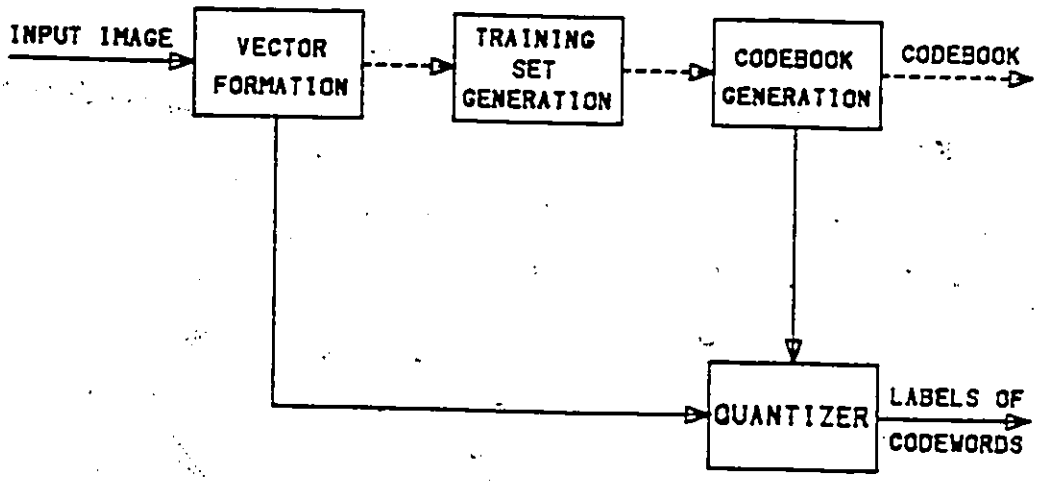


Fig.8 Principle of vector quantization. The dashed lines correspond to the training set generation, codebook generation and transmission.

3.1.1 Vector formation

The first step in vector quantization is vector formation; that is, the decomposition of the images into a set of vectors. Many different decompositions have been proposed; examples include, the colour components of a pixel [16]; the intensity values of a spatially contiguous block of pixels [11,12]; these same intensity values but now normalized by the mean and variance of the block [20]; the transformed coefficients of the block of pixels [17,19]; and the adaptive linear predictive coding (LPC) coefficients for a block of pixels [19].

3.1.2 Training set generation

An optimal vector quantizer should ideally match the statistics of the input vector source [7]. However, if the statistics of an input vector source is unknown, a training set, representative of the expected input vector source is used to design the vector quantizer. If the expected input vector source has a large variance, then a large training set (and codebook) is needed. To alleviate this problem the input vector source can be sub-divided. Gersho and Ramamurthi [11] divide the single input vector source into two more homogeneous sources corresponding to "edge" and "shade" vectors, respectively. Separate training sets are then used for each source and the resultant codebooks are then concatenated. In Boucher

and Goldberg [21], small local input sources, corresponding to portions of the image, are used as the training sets, thus the codebook can better match the local statistics.

3.1.3 Codebook generation

The key step in vector quantization is the development of a good codebook. The optimal codebook, using the mean squared error (MSE) criterion, must satisfy two necessary conditions [46]:

1. The input vector source, V; is partitioned into N closed sets or Voronoi regions, $\{R_i: 1 \le i \le N\}$, with partition determined by the minimum distance rule:

$$R_i=\{v : |v-w_i| \le |v-w_j| \text{ for all } j \ne i\}. \quad (19)$$

2. The corresponding codewords are then defined by

$$w_i=E\{v \mid v \in R_i\}, \; i=1,2,\ldots, N. \quad (20)$$

In other words, the codewords of the optimal codebook are the mean value of the corresponding Voronoi regions under the MSE measurement criterion. The K-means [6] or the closely related generalized Lloyd clustering algorithm proposed by Linde, Buzo and Gray [7] are typically used to generate the codebook. Both these clustering algorithms are iterative processes, minimizing a performance index calculated from the distances between the sample vectors and their cluster centers. These al-

gorithms have the advantage of not requiring any knowledge of the underlying statistics of the input source and additionally minimizing a criterion related to the quantization error.

3.1.4 Quantization

In vector quantization the process of quantization involves finding the closest codeword for each input vector. In this thesis, the straightforward expedient of searching the entire codebook is feasible, as the codebooks are relatively small.

3.2 A REVIEW OF VECTOR QUANTIZATION FOR IMAGE CODING

A review of image vector quantization is now presented from the following four aspects: vector formation, training sequence generation, codebook generation and quantization.

3.2.1 Vector formation

Many different approaches to vector formation have been proposed. These approaches can be classified into two categories: direct spatial/temporal and feature extraction.

(1) Direct spatial/temporal

Direct spatial/temporal is a simple approach to forming vectors from the intensity values of a spatial/temporal contiguous block of pixels in an image or an image sequence. A number of vector quantization schemes have been investigated

with this method. Hilbert [14] implemented a vector quantization for multi-speotral Landsat imagery. The vectors are formed from the intensity values associated with individual pixels in the four Landsat bands, thus exploiting the spectral correlation. Lowitz [15] described a vector quantization scheme for black and white imagery. In this scheme, the image is first divided into blocks, the vector is then formed by taking one pixel from a number of different blocks. In 1980, Yamada et al [54] reported results on using vectors formed from spatially contiguous pixels. Recent experiments on combining both spatial and spectral information were proposed by King and et al [17]. As discussed in [11], using spatially contiguous pixels introduces a problem in the reproduction of edges. One proposed solution is to first identify potential edge vectors and then to cluster these separately. Although the mean-squared error may increase, a definite visual improvement results [11].

(2) Feature extraction

An image feature is a distinguishing primitive characteristic. Some features are natural in the sense that such features are defined by the visual appearance of an image, while other so-called artificial features result from specific manipulations or measurements of an image or an image sequence. In the vector formation step some features, such as transformed coefficients, LPC parameters, block mean, can be ex-

tracted and coded separately. The practical significance of feature extraction is that it can result in the reduction in vector size, and consequently, reduce coder complexity.

(i). Unitary transform followed by vector quantization

The idea of vectors formed from a subset of transformed coefficients is similar to the hybrid transform/DPCM coding technique [36]. The transform is first applied to the blocks of pixels, then some high-order coefficients which have low energy can be discarded with little loss. The selected coefficients containing large energy are then used to form a vector. Thus, two benefits are obtained:

(a). The effective vector dimension mapped from spatial block decreases, which also reduces the computational complexity. As stated in [43], for a given bit rate, the performance achieved with vector quantization increases monotonically with increasing vector dimension and codebook size, theoretically approaching the limits given by rate distortion theory. As a first approximation, the computation required for codebook generation can be calculated as [43]:

$$CT = P \times N \times Ms \times K$$

where P -- a constant;

N -- the number of representative vectors;

Ms -- the number of training samples;

K -- the vector dimension.

Decreasing N, Ms or K, therefore, reduces the computing time, but of course affects both the bit rate and normalized mean squared error (NMSE). The transform is one effective manner of decreasing vector dimension without serious degradation of image quality.

(b). The second benefit arises indirectly from the effect of decorrelating the components in the vectors, therefore, reducing the brock effect [23]. The ideas can be made clear using the two-dimensional vector set as an example. In this case, the image is first divided into one by two pixel arrays, which are then mapped into the two-dimensional vector set $\{X: X=(x_i , x_{i+1})\}$. Since adjacent pixels are more likely to have nearly the same gray level, the vectors tend to occur in the vicinity of the line $x_i=x_{i+1}$. Additionally, the optimal codewords take the mean values of their corresponding clusters. Thus, after clustering, the resultant codewords are usually in the vicinity of the line $x_i=x_{i+1}$. However, some edge vectors (that is, those not lying near this line) are also replaced by the codewords so that a "block effect" results. This can be generalized to multi-dimensional vectors because of the correlation of adjacent pixels. To overcome this problem, the decorrelation of the components in the vectors is both necessary and valuable [23].

(ii). S-transform with vector quantization

The S-transform decomposes an image in a hierarchical manner. At each step the image is partitioned into blocks. Each block is then assembled to form a copy of the original picture at a lower spatial resolution. The non-zero order transformed coefficients from each block are aligned into a vector with dimension reduced by one. The decomposition process is reiterated on the lower resolution image. The process stops when only a single mean value remains, corresponding to the overall mean value of the original image. The results of combining S-transforms with vector quantization are reported in [13].

(iii). Adaptive LPC with vector quantization

In this scheme [18], the image is first partitioned into blocks. Separate LPC (linear predictive coding) parameters (linear predictor coefficients) are then calculated for each block, and these parameters are concatenated to form a vector and vector quantized.

(iv). Residual vector quantization

Baker and Gray [12,55] describe a residual vector quantization for intraframe coding, where the mean of each block is extracted and a difference signal is formed by subtracting the mean from the original signal. The mean is encoded by a scalar quantizer and the difference signal by vector quantization. Murakami et al. [20,56] develop a similar algorithm for interframe coding. The images is divided into blocks and each

block is first normalized with respect to its mean and variance. The normalized vector set is then vector quantized. The codebook is generated from the normalized vector set drawn from different sources..

3.2.2 Training sequence generation

The codebook is optimized on the given training sequence, which therefore must be a good representative of the input source being coded.

(1). Preprocessing

Various methods for preprocessing the training sequence before clustering have been proposed. One approach is separation of shade and edge vectors. Gersho and Ramamurthi [11] divide the single input vector source into two more homogeneous sources corresponding to "edge" and "shade" vectors, respectively. Separate training sets are then used for each source and the resultant codebooks are then concatenated. The purpose of this preprocessing is to improve the edge reproduction, by reserving more codewords for the "edge" source.

(2). Local training sequence generation

Generally, for vector quantization the codebook is optimized for a long training sequence. However, if the input image source has different local statistics, the codebook must large to maintain the fidelity; which, in turn entails more

bits for coding the label and increases coding complexity. An effective manner of decreasing the codebook size and training sequence length, thus decreasing the computational complexity, is to divide the image into smaller sections. Each section is taken as a training set and is used to generate its corresponding codebook. This approach is based upon the observation that image statistics are not spatially stationary. Thus, smaller codebooks are required to maintain the same distortion, which, in turn, implies that both codebook size and training sequence length are reduced. This algorithm is called adaptive vector quantization. One such an example is given by Boucher and Goldberg [21]. Results reported indicate that the computing times are decreased by up to a factor of four when adaptive codebook generation is used. It should be noted that in this approach, a separate codebook is generated for each image section, and the overhead is increased.

3.2.3 Codebook generation

A number of different approaches to codebook generation have been proposed.

(1). Multistep vector quantization

A multistep vector quantization process is described for speech coding in [57], and for image coding in [58]. In this technique, the error vector resulting from the first vector quantization is fed to a second vector quantization. The pro-

cess can be repeated by feeding the second step error vector into a third-step vector quantization, and so on. Note that if N_i is the codebook size at the i-th step for a m-step vector quantization, then the number of combined representative vectors is $N=N_1 \times N_2 \times \ldots \times N_m$. The computation time for codebook generation is now:

$$CT = P \times (N_1+N_2+\ldots+N_m) \times Ms \times K$$

where P is the constant, Ms is the number of training vectors, and K is the vector dimension. In a one-step approach, the computation time to obtain a codebook with the same number of representative vectors, N, is

$$CTo = P \times N \times Ms \times K = P \times N_1 \times N_2 \times N_m \times Ms \times K ,$$

For $N_i > 2$, it is clear that,

$$N_1 \times N_2 \ldots \times N_m >> N_1+N_2+\ldots+N_m$$

Therefore, CTo >> CT, and the computation time is reduced with the multi-step approach. As a result, the memory size for storing the codebook with size of N is also reduced from $(N_1 \times N_2 \ldots \times N_m)$ to $(N_1+N_2+\ldots+N_m)$.

(2). Adaptive codebook replenishment

Adaptive methods which track the local behaviour of the image statistics can improve overall performance [21]. In [21], reported results are concerned with a separate small codebook designed for each portion of the image. In this algorithm, the new codebooks increase both the overhead and computation. Instead of generating an entire new codebook, it may be more ad-

vantageous to only update or replenish the codebook, especially if the changes are not large. Such adaptive strategies are reported in [23] and [24].

(3) Initial codebook

As stated above, the codebook is generally generated by a clustering algorithm [6,7]. These clustering algorithms can only assure a local optimum, which depends upon the initial codebook or cluster seeds [6]. Two basic approaches have been developed to obtain the initial codebook. In the first [8], the starting point involves finding a small codebook with only two codewords, and then recursively splitting the codebook until the required number of codewords are generated. This approach is referred to as binary splitting. The second [59] starts with initial seeds for the required number of codewords, these seeds being generated by preprocessing the training sequence. Two examples of the latter approach are "parametric diagonal" seeding [16] and "mode-seeking" seeding [67]. In the first, the seeds are located along the diagonals of K-dimensional Euclidian hyperspace [16]. In the second, the seeds lie at the modes of the histogram [67].

(4). Mean/Reflected Residual Vector Quantization

In [55], Baker and Gray demonstrated a method for increasing the effective codebook size for a given memory space. In this technique, one label plus a 2-bit suffix is used to de-

scribe each codeword. The label describes the basic codeword and the suffix describes one of four reflections of the codeword. This does not decrease the codebook label size, but reduces the codebook storage memory size.

3.2.4 Quantization

Quantization in the context of a vector quantizers involves selecting a codeword in the codebook for each input vector. The optimal quantization, in turn, implies that for each input vector, v, the closest codeword (using the mean squared error (MSE) criterion), w_k is found:

$$|v - w_k| = \min_i \{|v - w_i|\}$$

that is, the codeword which minimizes the distortion.
A full search quantization is a exhaustive search process over the entire codebook for finding the closest codeword, as shown in Fig.9(a). It is optimal for the given codebook, but the computation is more expensive. An alternate approach is a tree-search quantization, where the search is carried out based upon a hierarchical partition. Buzo et al [8] describe a binary tree-search quantizer, as shown in Fig.9(b). At each layer of the tree, a decision is made between one of two sets of codewords. At the last layer, a representative codeword is found for the input vector. Experimental results [8] show that an 8-layer binary tree search quantization is over 6 to 8 times faster than the full search case. However, it is clear that the tree-search is suboptimal for the given codebook, and requires more memory.

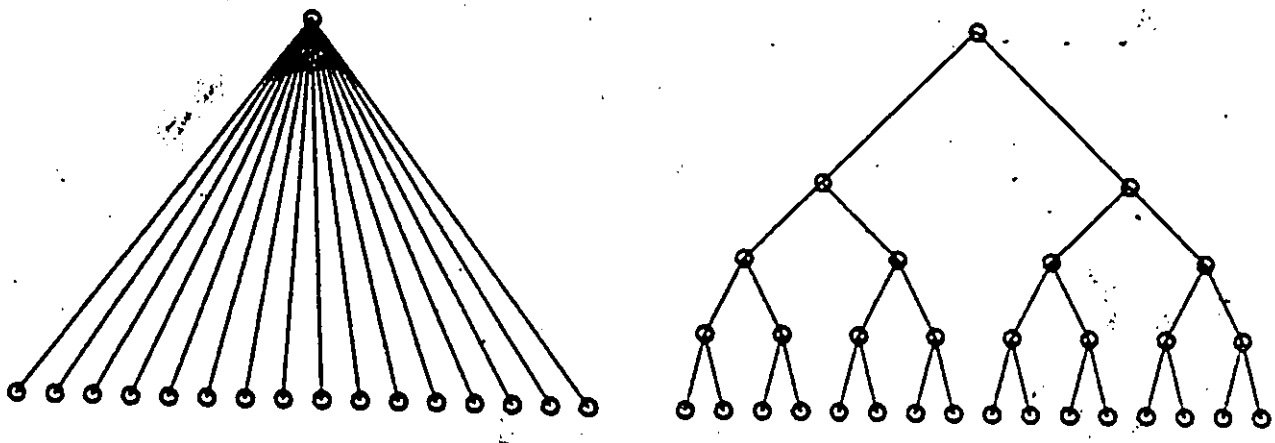


Fig.9(a) Full search quantizer, (b) Tree-search quantizer.

3.2.5 Summary of image vector quantization

Image coding by vector quantization has been reviewed from four aspects: vector formation, training sequence generation, codebook generation and quantization. This is summarized in Table 2.

Vector formation: Techniques for vector formation can be classified into spatial/temporal and feature extraction. Feature extraction includes transformed coefficients, S-transform, LPC parameters and residual vectors. Using feature extraction the vector components are decorrelated and the vector size is reduced, resulting in better coded images.

Training sequence generation: Improved performance has been achieved by the proper choice of the training sequence. For example, edge reproduction is improved by partitioning the training sequence into "shade" and "edge" vectors. Local training sequences result in codebooks which match the local statistics.

Codebook generation: Techniques for codebook generation include multistep vector quantization, adaptive codebook replenishment, and reflected residual vector quantization. Adaptive codebook replenishment updates the codebook to track the changes in local statistics of the input source and can realize an increase in performance.

Quantization: The approaches to quantization include full-search and tree-search. The full-search is optimal for the given codebook, but requires a great deal of computation. The tree-search is a faster process, but is suboptimal for the given codebook. The selection of quantization would require a compromise between performance and computation time.

Table 2 Characteristics of different vector quantization techniques

Step of VQ	Techniques	Comments
Vector formation	Direct spatial/ temporal	simple, poor edge reproduction [11].
	Feature extraction	
	Transformed coefficients	decorrelates vector components, reducing vector dimension, encoder complexity increased [17,19].
	S-transform	progressive coding, fast vector quantization [13].
	LPC parameters	reduce vector dimension, one-bit map required for differential error signal [18].
	Residual VQ	fixed codebook, increased bit rate to code mean and variance [12,54].
Training set generation	separation of shade and edge vectors	improved edge reproduction [11,68].
	Local source as training set	codebook better matches local statistics, reduces codebook size and training set length, increased overhead [43].
Codebook generation	Multistep VQ	reduces codebook size and memory size, more steps VQ required [57,58].
	Adaptive codebook replenishment	efficiently updates the codebook to track local statistics [23,24].
	Reflected residual VQ	reduces codebook size and memory size [55].
Quantization	full-search quantization	optimal for a given codebook, computation is complex.
	tree-searched quantization	reduces computational complexity, local optimum [8].

3.3 AN EXAMPLE OF INTRAFRAME CODING USING VECTOR QUANTIZATION

This section describes a technique for image coding which combines adaptive linear predictive coding (LPC) with vector quantization [18].

3.3.1 Adaptive linear predictive coding

Traditionally, LPC parameters are estimated using so-called autocorrelation methods and digitized by scalar quantization techniques, in which each parameter is individually coded. If $x(n)$ is a set of image pixels indexed according to their time occurrence and assuming them to be identically distributed with zero mean and a variance, a linear predictor for the n-th element $\hat{x}(n)$ using k previous elements $x(n-1)$, $x(n-2)$,..., $x(n-k)$ can be written as

$$\hat{x}(n) = \sum_{i=1}^{k} C(i)\, x(n-i) \qquad (21)$$

where the coefficients $C(i)$, $i=1,\ldots,k$, can be obtained by minimizing the mean-square-error (MSE) $E\{[x(n)-\hat{x}(n)]^2\}$. They must satisfy the following equations:

$$R(j) = \sum_{i} C(i)\, R(i-j), \quad 1 \le i, j \le k \qquad (22)$$

where, $R(j)=E\{x(n)x(n-j)\}$.

As stated above, in designing an adaptive LPC system, two approaches have been adopted. One method uses a predictor with parameters that change with the variance of the signals. The second approach uses a fixed predictor with a variable quantizer to accommodate the resultant differential signals. In our algorithm the former approach is adopted, since it is easier to vector quantize the resultant parameters.

3.3.2 LPC parameter extraction

Adaptivity to the local image structure is obtained by partitioning the image into M×N blocks of block size P×P, where M and N are the number of blocks in the horizonal and vertical direction, respectively. For each block, the parameters of this model can be estimated by the method of linear predictive analysis. In this method we assume the coefficients are those which minimize the mean-squared value of the predictor error signals:

$$e(l,k,m,n)=x(l,k,m,n)-\sum_i\sum_j C_{ij}(m,n)x(l-i,k-j,m,n)-C_0(m,n) \tag{23}$$

where, $1\le l,k\le P$, $1\le m\le M$, $1\le n\le N$, and i,j range over the previous elements which are used to estimate the present element. The optimal predictor coefficients satisfy the equations [60]:

$$\sum_i\sum_j C_{ij}(m,n)R(l-i,k-j,m,n)+C_0(m,n)=R(i,j,m,n) \tag{24}$$

and

$$\sum_i\sum_j C_{ij}(m,n)A(m,n)+C_0(m,n)=M(m,n) \tag{25}$$

where, $M(m,n)=\sum_{i=1}^{P}\sum_{j=1}^{P} x(i,j,m,n)/(P\times P)$ is the mean of the block indexed (m,n), and R is the correlation function. Solving these equations, the LPC parameter set of $C_{ij}(m,n)$ and C_0 is obtained, corresponding to each block (i,j range over the previous elements used in predictor).

3.3.3 Differential signal coding

From the above analysis, the differential signals of adaptive LPC system have the following form:

$$\hat{x}(l,k,m,n)=\sum_{i}\sum_{j} C_{ij}(m,n)\, x(l-i,k-j,m,n)-C_0(m,n) \qquad (26)$$

$$e(l,k,m,n)=x(l,k,m,n)-\hat{x}(l,k,m,n)=\sigma(m,n)u(l,k,m,n) \qquad (27)$$

where $\sigma(m,n)$ is the standard deviation of the block indexed (m,n) and u is a random variable, represented by a normalized Gaussian model,

$$E\{u\}=0, \text{ and } p(u)=\frac{1}{\sqrt{2\pi}}\exp\{-u^2/2\} \qquad (28)$$

If a one bit scalar quantizer is used to quantize u(l,k,m,n), then the quantized output is represented by

$$\hat{u} = \pm \frac{1}{\sqrt{2\pi}} \int_0^{\infty} up(u)du = \pm 0.8$$

3.3.4 Combination of adaptive LPC with vector quantization

It is noted that with large vector dimensions the statistical dependence of the vectors will be reduced, however, the correlation within each vector will increase. In order to exploit the redundancies, both intra-vector and inter-vector, the coding procedure can be divided into two steps:

1. The preprocessing step is used to decorrelate the components within vectors.
2. The following processing step attempts to reduce the statistical dependence between the vectors.

In this algorithm, the first step is the LPC coding. At this step, a set of LPC parameters, $C_{ij}(m,n)$ and standard deviation $\sigma(m,n)$ corresponding to the block indexed (m,n), are obtained. The $C_{ij}(m,n)$ and $\sigma(m,n)(m,n)$ are used to form a vector set. In the second step, vector quantization is applied to this vector set. The LPC parameters of each block is replaced by the closest codeword in the codebook. The coding procedure is illustrated in Fig.10.

3.3.5 Experimental results

In the experiments, a 256 by 256 image has been divided into blocks of sizes 8x8, 16x16 or 32x32. A third order two-dimensional model of LPC has been used in every block. Thus, we have

$$\hat{x}(l,k,m,n) = a(m,n)x(l,k-1,m,n)+b(m,n)x(l-1,k,m,n) + c(m,n)x(l-1,k-1,m,n)+d(m,n) \qquad (30)$$

$$x(l,k,m,n) = \hat{x}(l,k,m,n)+\sigma(m,n)\hat{u}(l,k,m,n) \qquad (31)$$

where, a(m,n), b(m,n), c(m,n), d(m,n) and σ(m,n) are obtained by using the autocorrelation estimation method for each block indexed by (m,n). A number of different block sizes and different codebook sizes have been used. The distortion is measured as the percentage normalized mean squared error (%NMSE), and defined as

$$\text{NMSE} = \frac{\Sigma\ (X_i - \hat{X}_i)^2}{\Sigma\ X_i^2}\ 100\%, \qquad (32)$$

where X_i and $\hat{X}_i$ are the original and quantized grey levels, respectively. Some results are shown in Table 3 and some pictures are shown in Fig.11.

Table 3

Block size	codebook size	NMSE(%)	bit rate
8 x8	1024	0.1322	1.625
8 x8	128	0.1796	1.188
8 x8	64	0.2072	1.102
16x16	256	0.1814	1.156
16x16	64	0.1858	1.063
16x16	32	0.1972	1.023
32x32	64	0.2457	1.039
32x32	32	0.2570	1.022

3.3.6 Conclusions

It is found that the adaptive LPC for image coding can be implemented by first partitioning an image into blocks and then processing in two steps: LPC followed by vector quantization. A small block size is good for adaptivity, but more bits are needed for coding LPC parameters. However, for small block size, the LPC parameters are more statistically dependent between the blocks. Vector quantization can, therefore, exploit this statistical dependence to reduce the bit rate. The results show that LPC followed by vector quantization produces good reconstructed images at bit rates of about 1.1 to 1.7 bts/pixel, corresponding to compression ratios between 7 to 1 and 5 to 1.

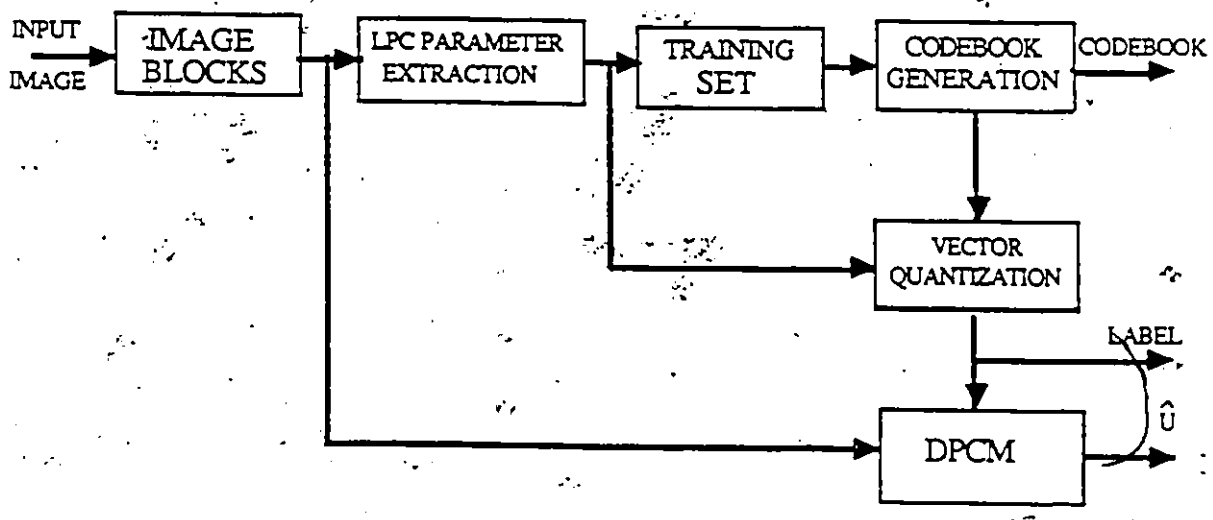


Fig.10 Schematic diagram of adaptive LPC with vector quantization.

(a)

(c)

(c)

(d)

Fig.11 Pictorial results of LPC with vector quantization, (a) original, 8 bits/pixel; (b) block size 8 by 8, 1.625 bits/pixel; (c) block size 16 by 16, 1.156 bits/pixel; (d) block size 32 by 32, 1.022 bits/pixel.

Chapter IV

THREE-DIMENSIONAL BLOCK VECTOR QUANTIZATION

In single image (intra-frame) coding by vector quantization, the images are typically divided into spatially contiguous two-dimensional blocks and vectors are then extracted from these blocks. The same principle can be extended to image sequence (inter-frame) coding, where the blocks are now three-dimensional, with the third dimension along the time axis. In other words, both the spatial and temporal redundancies are exploited simultaneously.

In this chapter, two different strategies for vector formation are tested:

i) Vectors are formed directly from the three-dimensional blocks.

ii) The three-dimensional blocks are first Hadamard transformed and vectors are then formed from a subset of the resulting transform coefficients.

4.1 DIRECT THREE-DIMENSIONAL BLOCK VECTOR QUANTIZATION

The vector formation step of direct three dimensional block vector quantization (DVQ) is shown in Fig.12. The input image sequence, $f(x,y,t)$, is first partitioned into contiguous three-dimensional blocks. Training set generation is the next step and, for the image sequence chosen, presents a problem. The first 9 frames of the sequence are fairly constant; the next 5 frames exhibit a rapid motion of the woman face; and finally, the last 6 have relatively small changes. Three different strategies for choosing the training set were tested:

(i) TS-I: the training set is formed by selecting every second vector from the entire sequence;

(ii) TS-II: the training set consists of all the vectors in the first four frames;

(iii) TS-III: the training set consists of all the vectors in frames 15 through 18.

The different training sets are then used to generate codebooks and to quantize the entire sequence.

The experimental results, given as rate distortion curves and the curves of distortion as a function of frame number for three different training sets, are shown in Fig.13(a) and 13(b), respectively. The distortion is measured as the percentage normalized mean squared error (%NMSE), and defined as

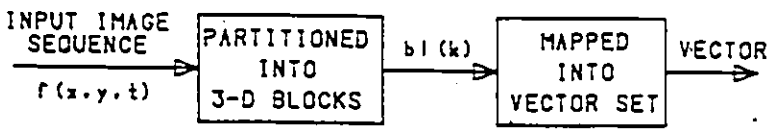


Fig.12 Vector formation for 3-D vector quantization.

$$NMSE = \frac{\Sigma (X_i - \hat{X}_i)^2}{\Sigma X_i^2} \; 100\%, \qquad (33)$$

where X_i and $\hat{X}_i$ are the original and quantized grey levels, respectively. The bit rate (bits/pixel) is calculated as follows.

$$\begin{aligned} BR &= \text{overhead} + \text{label} \\ &= BRo + BR1 \\ &= N \times Bc/Np + (\log_2 N)/Nb \end{aligned} \qquad (34)$$

where,

BRo - the bit rate for codebook overhead;

BR1 - the bit rate for the label;

N - the number of codewords in the codebook;

Np - the number of pixels in the input source;

Nb - the number of pixels per block.

Bc - the average number of bits per codeword;

In this chapter, each component of the vectors is scalar quantized with 8 bits, and Bc=8×Nb.

The rate distortion curves corresponding to the use of the three different training sets are shown in Fig.13(a). For each set, four different rates are shown, corresponding to codebooks where the number of codewords are, respectively, 64, 128, 256 and 512. Although the best overall performance is obtained by using training sequence TS-III, it is instructive

to highlight the performance on an individual frame basis. The individual frame distortion values are shown in Fig.13(b) for the three training sets, each coded at a fixed bit rate of 0.63 bits/pixel. For the first nine frames the best performance is obtained by using the training set, TS-II, drawn from the first four frames. Training set, TS-I, drawn from the entire sequence is a clear second. In frames 10 to 20, a rapid motion first occurs followed by smaller changes and best results occur by using training set, TS-III, drawn from the frames exhibiting some motion. As for the first 9 frames TS-I again ranks second in performance. These results suggest that updating the codebook on an adaptive basis could yield improved performance [23].

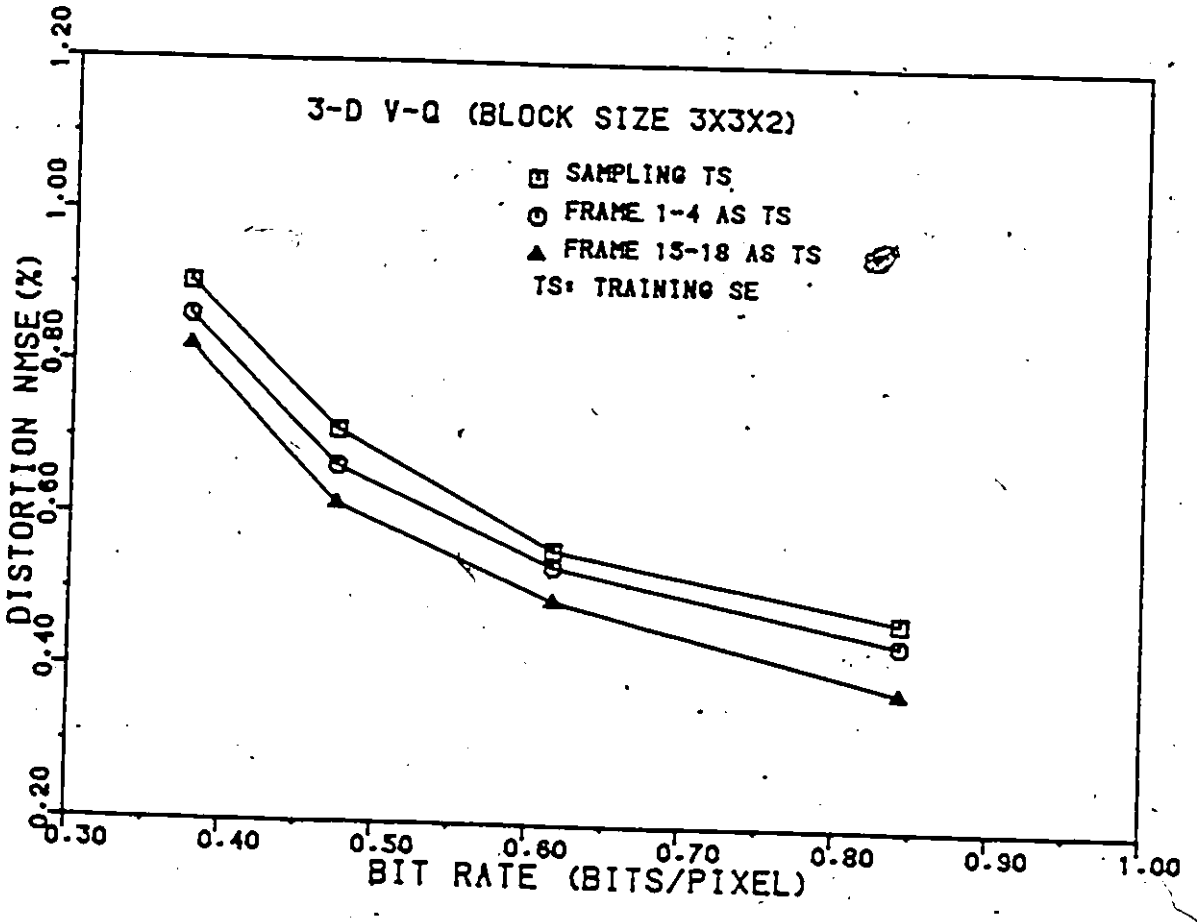


Fig.13(a) Rate distortion curves using three different training sets, and four different codebook sizes. Note the poor performance of the sampled training set. The distortion has been expressed as normalized mean squared error and has been averaged over the twenty frames.

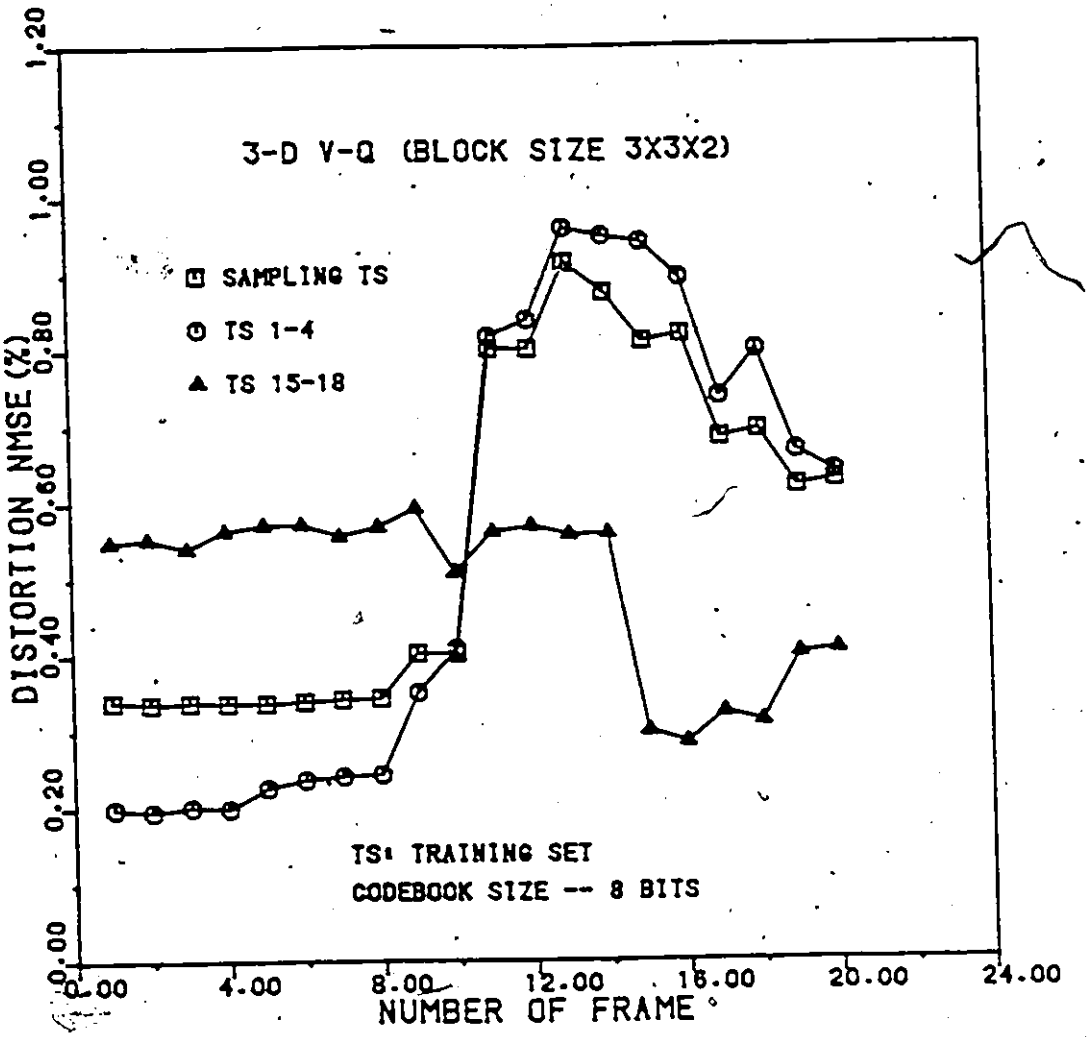


Fig.13(b) The curves of distortion as a function of frame number for three different training sets, all are for a bit rate equal to 0.63 bits/pixel. Note that training set TS 1-4 shows its best performance in frames 1-4 and similarly TS 15-18 has its best performance in frames 15-18.

4.2 THREE-DIMENSIONAL HADAMARD TRANSFORM WITH VECTOR QUANTIZATION

The vector formation step of three-dimensional Hadamard transform with vector quantization (HTVQ) is shown in Fig.14. The image sequence is first divided into three-dimensional blocks of size 4x4x4. The three-dimensional Hadamard transform is then calculated and a subset of coefficients is selected to form the vector. The Hadamard transform reduces the intra-block correlation and, therefore, compacts the energy in a small number of terms so that vectors of reduced dimension can be used.

The Hadamard transform is a non-sinusoidal orthogonal transform which is especially suited to digital processing : only addition and subtraction are necessary to calculate the transformed signals. Let $f(x,y,t)$ represent the grey level intensity values of the pixels in a three-dimensional array or block of size $p \times p \times p$, where $p=2^n$. The three-dimensional Hadamard transform, $F(u,v,w)$, of $f(x,y,t)$ is then given by

$$F(u,v,w)= \sum_{x=0}^{p-1} \sum_{y=0}^{p-1} \sum_{t=0}^{p-1} f(x,y,t)\ (-1)^{q(x,y,t,u,v,w)} \qquad (35)$$

where $$q(x,y,t,u,v,w) = \sum_{i=0}^{n-1} (u_i x_i + v_i y_i + w_i t_i)$$

and the terms u_i , v_i , w_i , x_i , y_i , and t_i are the binary representations of u ,v , w, x, y, and t, respectively [61].

If f(x,y,t) is considered to be a random three-dimensional function with given entropy, then the energy of F(u,v,w) is the same as entropy of f(x,y,t). Hence, with proper coding it should be possible to transmit, with the same channel capacity, either the original image data or its Hadamard transform.

After three-dimensional Hadamard transformation, the next step is the selection of the most significant coefficients with which to form the vector. Ideally, this is done by observing the energy distribution of the transformed coefficients within each block. However, due to the changes in the local statistics, the optimum choice will vary from block to block. Published studies indicate that the energy distribution of the transformed coefficients can be modeled as an exponential function [64]:

$$E(u,v,w) = \exp\{ -\alpha u -\beta v -\gamma w \} \qquad (36)$$

This implies that the lowest order coefficients contain the most energies; in other words, to minimize the mean squared error, the vector should be formed from the lowest order coefficients.

To test the validity of this model, the Hadamard transform is first evaluated on blocks of size 4x4x4, and the total energy distributed on the 6, 9, 12, 18 and 25 lowest order coefficients, but not including the zero-order term, is calculated. These are then compared, respectively, with the total

energy distributed on the non-zero order coefficients chosen optimally for each block. The results are presented on a slice by slice basis (4 frames being equivalent to a slice) for the entire test image sequence. The results are shown in Table 4. It can be seen that, for 12 or more coefficients, the loss resulting from using the lowest non-zero order coefficients is small.

The next step is training set generation, and the same issues for choosing the training sets, as discussed in last section arise. Three training sets similar to those in section 4.1 are used:

(i) TS-IV: the training set is formed by selecting every second vector from entire sequence;

(ii) TS-V: the training set consists of all the vectors in the first first three slices, corresponding to the first 12 frames;

(iii) TS-VI: the training set consists of all the vectors in the last three slices, corresponding to frame 9 through 20.

The different training sets are then used in generating codebooks with which to quantize the entire image sequence.

The experimental results, given as rate distortion curves and as distortion for the individual frames, for the three different training sets, are shown in Fig.15a and Fig.15b, respectively. The bit rate is calculated as follows:

$$BR = \text{overhead} + \text{label} + \text{mean term}$$

$$= BRo + BR1 + \cdots BRm$$

$$= N \times Bc/Np + (\log_2 N)/Nb + Bm/Nb \qquad (37)$$

where,

BRm - the bit rate for the block mean term;

Bm - the number of bits for the block mean term;

the others are same as for the equation (33), and Bc=8×K, it should be noted that K is now smaller than block size, Nb.

The rate distortion curves corresponding to the use of the three different training sets are shown in Fig.15(a). For each set, four different rates are shown, corresponding to codebooks where the number of codewords are, respectively, 64, 128, 256 and 512, and the vector dimension is 18. The individual frame distortion values are shown in Fig.15(b) for the three training sets, each coded at a fixed bit rate of 0.61 bits/pixel. The same conclusion as for the direct three-dimensional block vector quantization can be made. The best results on an individual frame basis is obtained by using the codebook generated with locally selected training sets. TS-V is the best on the first 12 frames, whereas TS-VI is the best, on the last 12 frames. This suggests once more that adapting the codebook as the image sequence progresses could yield reduced distortion.

4.3 COMPARISON

In the next set of experiments the vector dimensions are varied and the corresponding rate distortion curves are shown in Fig.16. For comparative purposes, the results of using a three-dimensional Hadamard transform combined with a zonal quantizer described in [64] is provided. For direct three-dimensional block vector quantization the results for blocks of size (2x2x2) and (3x3x2) are included where the appropriate training set, TS-III, is used for codebook generation. In the cases of Hadamard transform combined with vector quantization, the results for three vector dimensions are shown, respectively, 9, 12 and 18 coefficients. In all three cases, the appropriate training set, TS-VI, is used for codebook generation. The results clearly demonstrate the utility of first performing the Hadamard transform before vector quantization.

In Fig.16 the overhead component required to transmit the codebook is factored into the bit rate. For the image sequence used, the overhead component varies, from 7% for direct spatial vector quantization for a block size of 2x2x2 and a codebook size of 256, to 40% for the Hadamard transform with vector quantization for vector dimension of 18 and a codebook size of 256. For images larger than 112x96 and sequences longer than 20 frames, the overhead component will be significantly reduced. Fig.17 shows the rate distortion curves for the same cases as in Fig.16, but where only the component for

the mean and the label is included in the bit rate. These clearly demonstrate the increased performance to be expected by combining Hadamard transform with vector quantization.

The results for three frames, 1, 14 and 18 are, respectively, shown in Fig.19(a), 19(b) and 19(c). The first is in a zone of little motion, the second in a zone of rapid change and the third in a zone of moderate change. For the Hadamard transform with a zonal scalar quantizer, the bit rate is 0.7 bits/pixel and the %NMSE is 0.65. The corresponding values for the direct vector quantization on a block size of (3x3x2) are, respectively, 0.63 bits/pixel with overhead and 0.45 bits/pixel without overhead, at 0.49% NMSE distortion. In the cases of Hadamard transform followed by vector quantization the results when 18 coefficients are used are, respectively, 0.61 bits/pixel with overhead and 0.27 bits/pixel without overhead, at 0.43% NMSE distortion. Frame 1 and 18 are both reasonably reproduced, whereas frame 14 in the zone of rapid motion is more severly degraded. In all three cases, Hadamard transform with vector quantization is much better than direct block vector quantization. The reproduction of both Hadamard transform with vector quantization and Hadamard transform with zonal quantizer are similar in terms of visual quality. However, in the Hadamard transform with zonal quantizer there is a visible spatial pattern superimposed on the image, probably due to poor quantization of the higher order coefficients.

4.4 DISCUSSION

In this paper, two vector formation schemes for inter-frame image vector quantization, generalized from intra-frame image coding technique [21] are presented: direct three-dimensional block vector quantization and three-dimensional Hadamard transform followed by vector quantization of a subset of Hadamard coefficients. The best results are obtained by using the Hadamard transform with vector quantization as the components in the vectors are decorrelated in both, spacial and temporal, direction. One problem for both schemes is choosing an appropriate training set. The results demonstrate that using a training set representative of the entire sequence does not necessarily give the best results. It is demonstrated experimentally that an appropriate codebook tailored to each local source could reduce the distortion. The subsequent chapers are being pursued on various adaptive techniques for updating the codebooks [23].

Table 4. Energy distribution on the non-zero order Hadamard coefficients

Optimally chosen coefficients

	number of coefficients				
slice	6	9	12	18	25
1	0.95321	0.98392	0.99426	0.99920	0.99981
2	0.95280	0.98390	0.99464	0.99949	0.99989
3	0.88166	0.92380	0.95887	0.97739	0.99046
4	0.88832	0.92716	0.95613	0.97823	0.99115
5	0.88315	0.92646	0.95638	0.97823	0.99052

Lowest order coefficients

	number of coefficients				
slice	6	9	12	18	25
1	0.89452	0.95032	0.97216	0.98584	0.99413
2	0.89443	0.94898	0.97141	0.98513	0.99383
3	0.85931	0.92143	0.95887	0.97506	0.99032
4	0.84876	0.92041	0.95613	0.97584	0.99099
5	0.85043	0.91936	0.95638	0.97417	0.99041

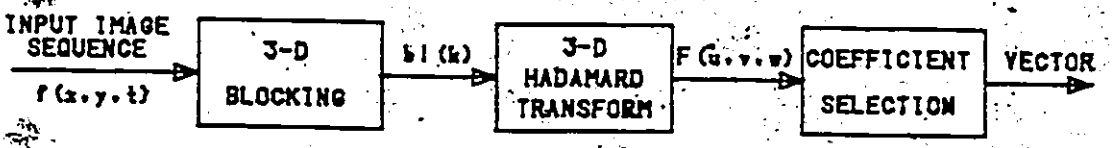


Fig.14 The three steps involved in vector formation for three-dimensional Hadamard transform with vector quantization: three-dimensional blocking, three-dimensional Hadamard transform and Hadamard coefficient selection.

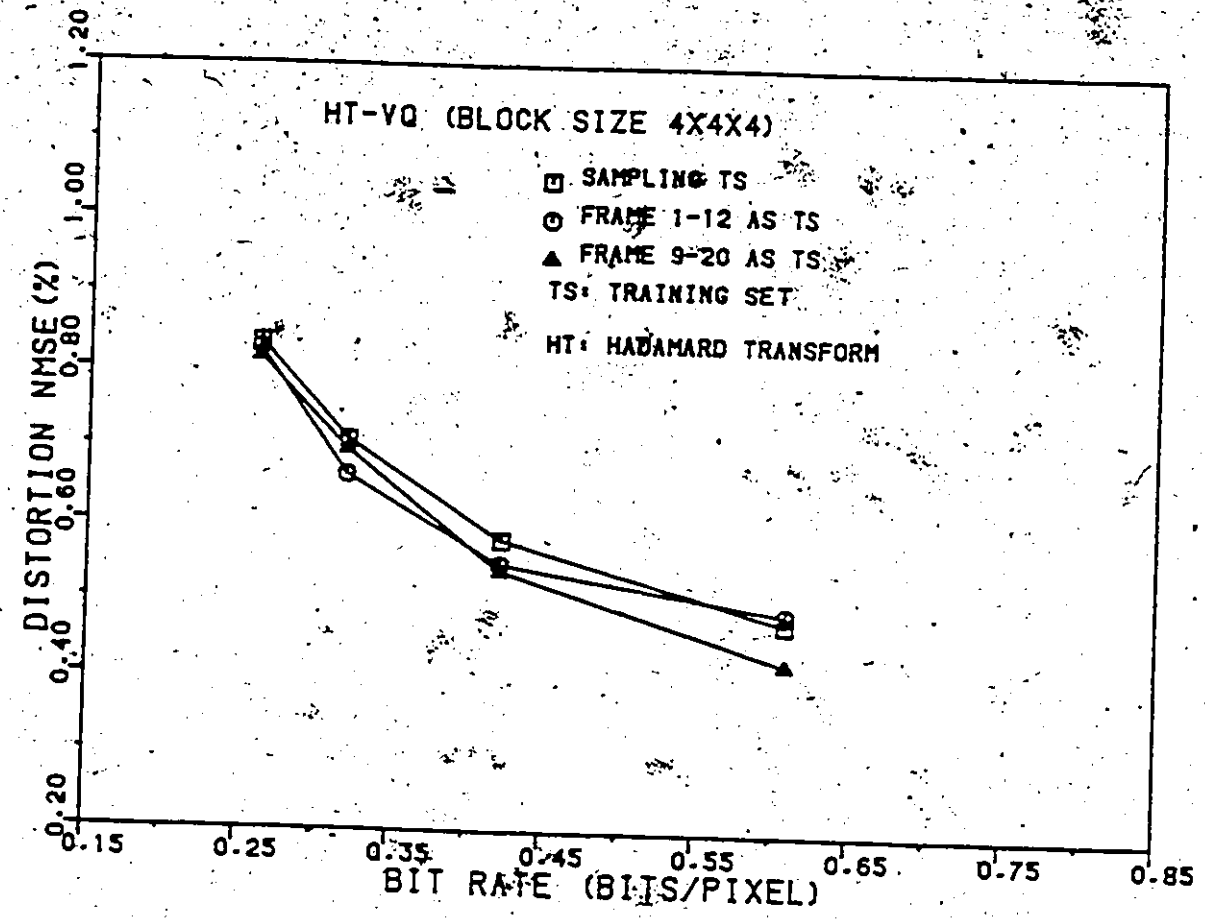


Fig.15(a) Rate distortion curves using three different training sets, and four different codebook sizes. The vectors are formed from the 18 lowest non-zero order coefficients in the transformed domain. Note the poor performance of the sampled training set. The distortion has been expressed as percentage normalized mean squared error and has been averaged over the twenty frames.

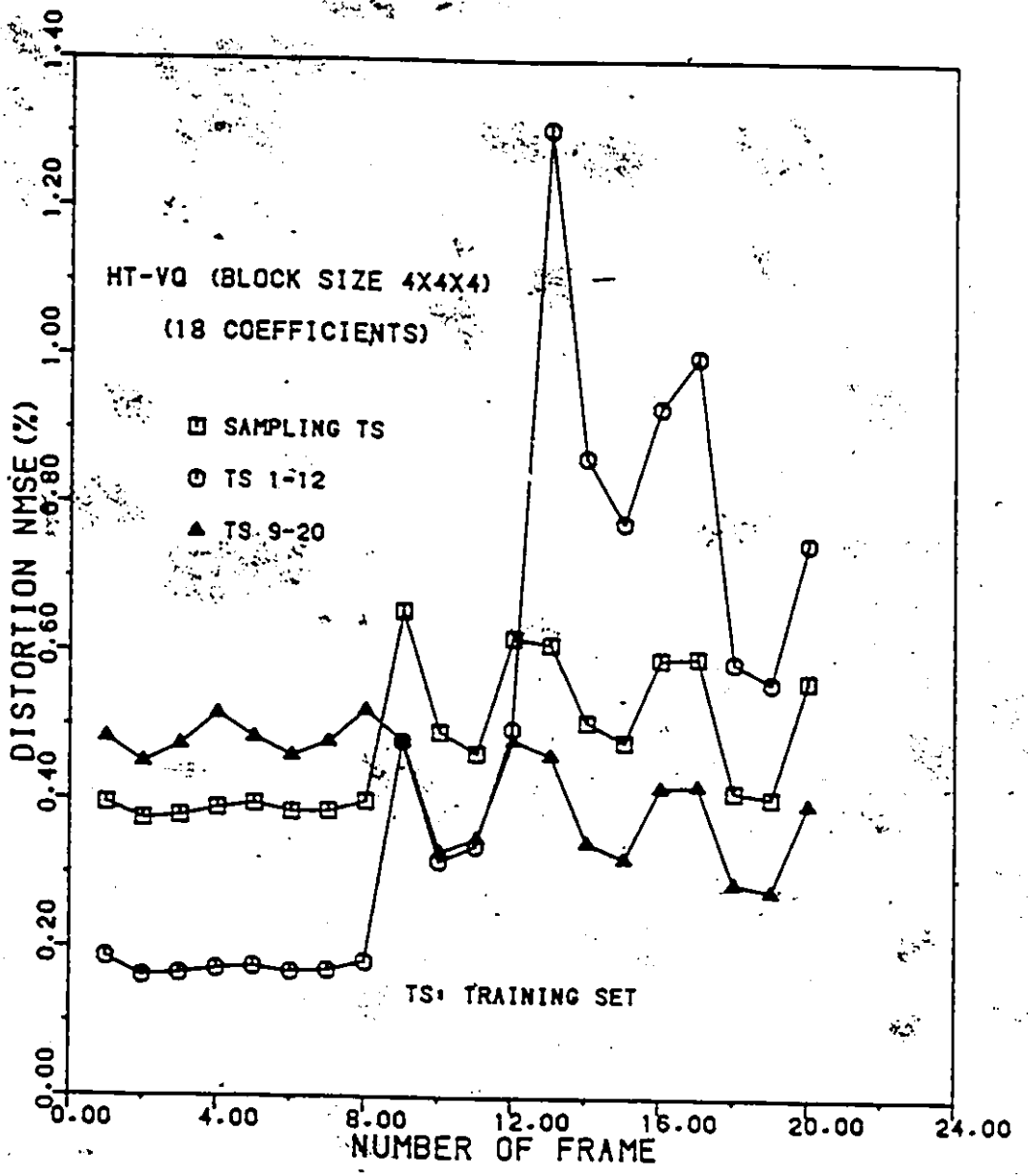


Fig.15(b) Distortion as a function of frame number for three different training sets. The bit rate is 0.61 bits/pixel. Note the locally generated training sets give the best performance on the frames used in the training sets.

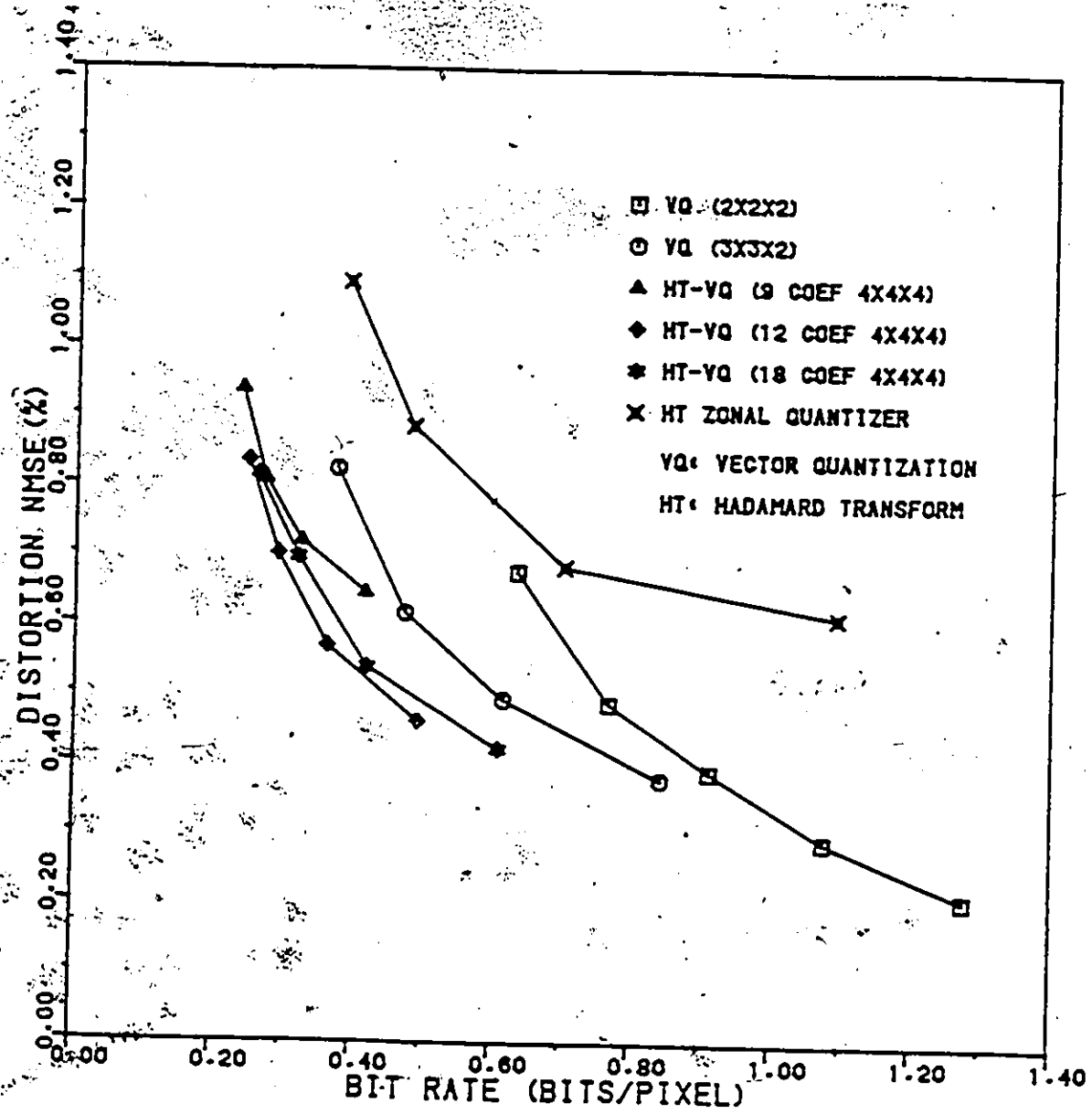


Fig.16 Rate distortion curves for different coding schemes and different vector dimensions. Here all components of the bit rate are included. Note that vector quantization combined with the Hadamard transform (HTVQ), outperforms direct three-dimensional block vector quantization (VQ), which in turn outperforms scalar quantized Hadamard transform coding. The best performance is obtained by HTVQ with 12 coefficients.

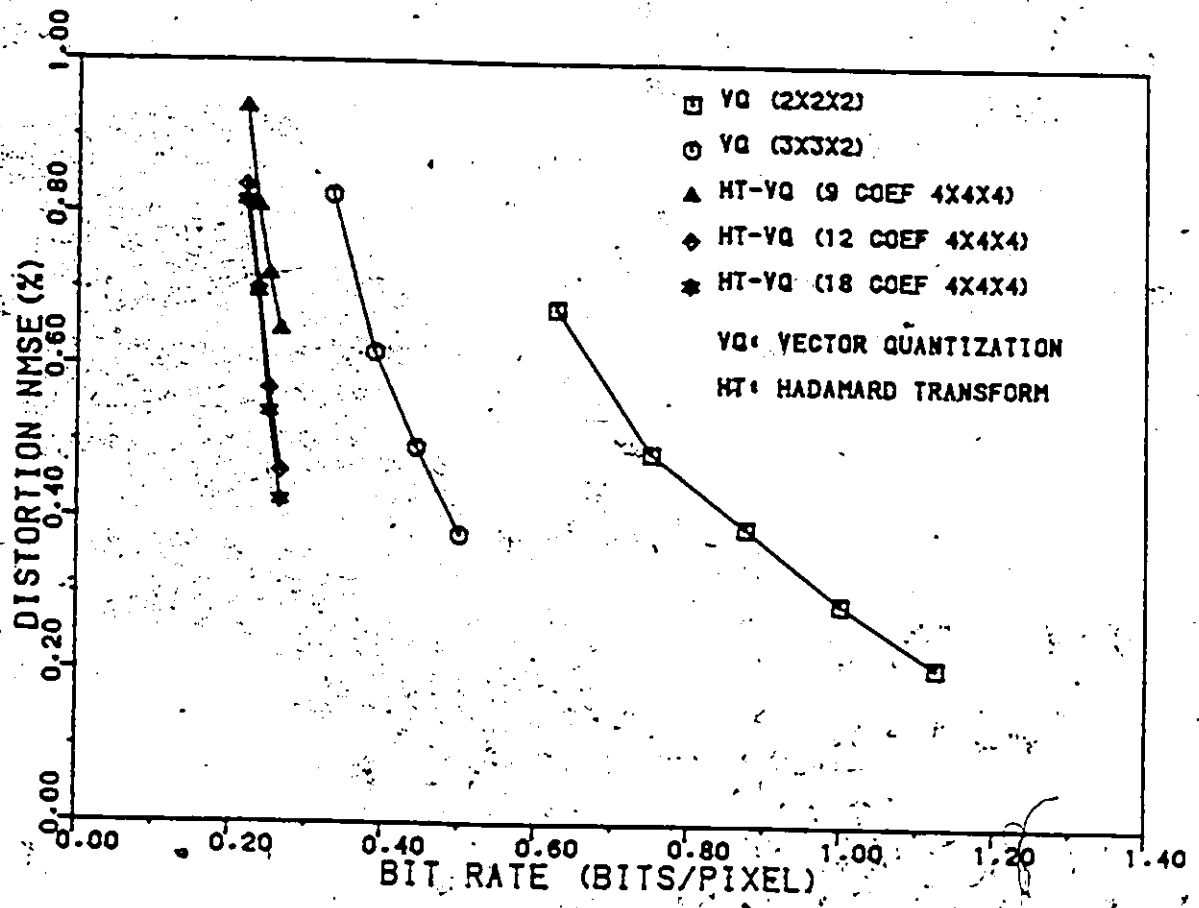


Fig.17 Rate distortion curves for different coding schemes and different vector dimensions. Only the component required for transmitting the mean value of the block and the label is included.

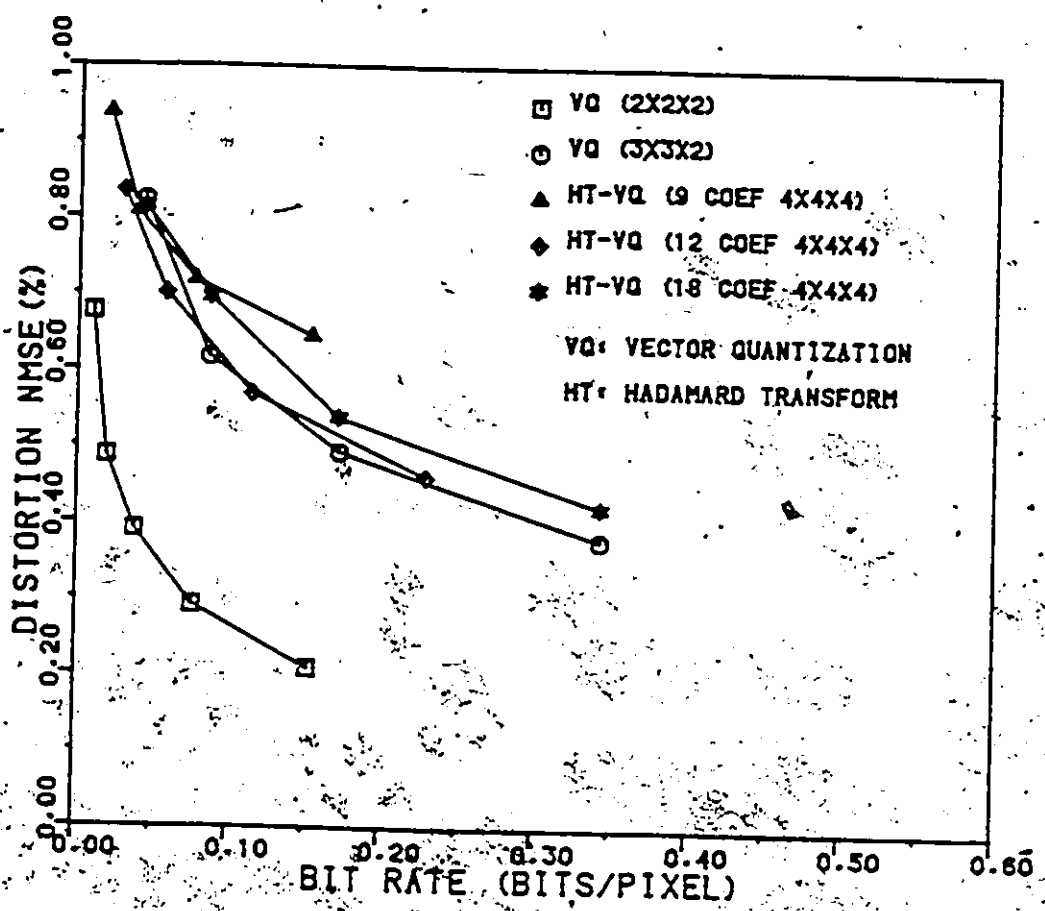


Fig.18 Rate distortion curves for different coding schemes and different vector dimensions. Here only the component required for transmitting the codebook of representative vectors, the overhead, is shown. Note that for the small block sizes, the overhead component decreases because the codewords are smaller.

Fig.19(a) The pictorial results for the first frame (the top left is the original). The three coded images correspond to, respectively, the three-dimensional block Hadamard transform with vector quantization on the 18 lowest non-zero order coefficients (top right, 0.61 bits/pixel), three-dimensional Hadamard transform by zonal quantizer (bottom left, 0.69 bits/pixel) and the direct three-dimensional block vector quantization with a block size 3×3×2, (bottom right, 0.62 bits/pixel).

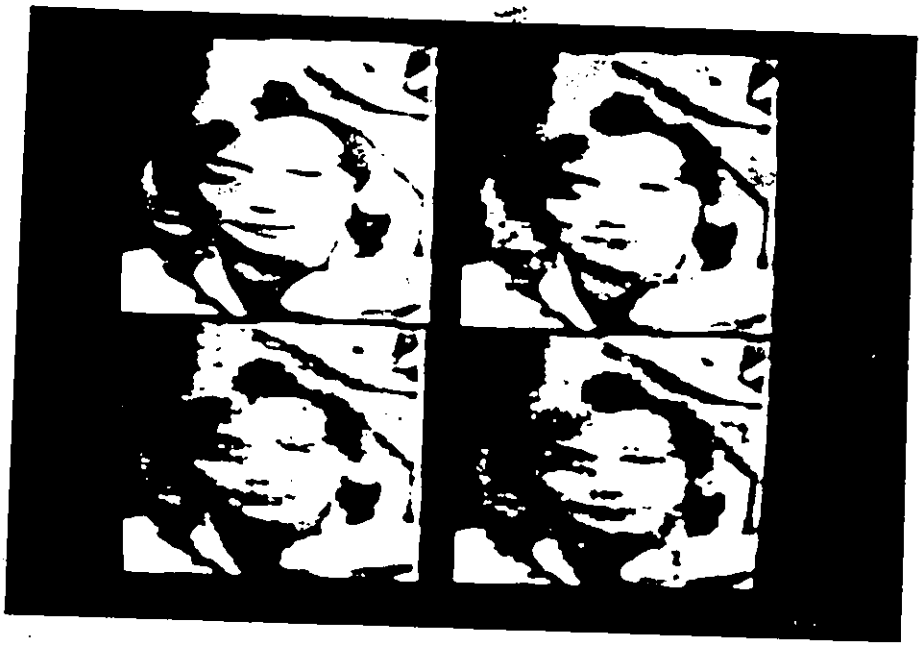


Fig.19(b) The pictorial results for the 14-th frame, which is in a zone of rapid changes. The coded images are as described in Fig.19(a).

Fig.19(c) The pictorial results for the 18-th frame, which is in a zone of moderate change. The three coded images are as described in Fig.19(a).

Chapter V

ADAPTIVE VECTOR QUANTIZATION BY REPLENISHMENT

In an image sequence, successive frames are usually very similar. A simple interframe coding technique which exploits this similarity is frame replenishment coding [3]. In this method, only the information necessary to describe the new grey level values of the pixels that change between successive frames are transmitted. The implementation uses two frame memories, one at the receiver and another at the transmitter to store the required reference frames. The incoming signal is compared pixel by pixel with the reference frame. When the magnitude of the frame-to-frame difference is greater than a given threshold, it is considered as significant, and the new signal value is replaced in both frame memories. Thus, the receiver memory is made to track the transmitter memory. Studies [4,51] indicate that the pixels of the changing area need not be transmitted independently of one another. Various techniques, such as predictive coding, motion-compensation and others have been used to exploit the correlation between these pixels [51].

In this chapter, a new algorithm which combines the concept of frame replenishment with two-dimensional block vector quantization is proposed. Each frame is first divided into two-

dimensional blocks, and vectors are then extracted from the resultant blocks, either directly in the spatial domain or indirectly through the transform domain. The vectors corresponding to the first frame are taken as the training set for generating the first codebook. In encoding subsequent frames, the operation of replenishment can be applied to both labels (label replenishment) and codewords (codebook replenishment). A more detailed description of each operation follows.

5.1 LABEL REPLENISHMENT

The basic principle of operation of label replenishment using direct two-dimensional block vector quantization, illustrated in Fig.20, consists of two phases. The first phase is codebook generation represented by the dashed line 1; where the first frame is used as a training set. The first frame is then quantized, and the labels are stored in the label memory and transmitted as indicated by dashed line 2. The next phase, represented by the solid line, corresponds to coding subsequent frames. Each vector is first quantized, the label is then compared with the one in the label memory, and if it differs, the memory is replenished. An extra bit is used as side information to indicate whether the label needs to be replenished. The bit rate for this coding scheme can be calculated as follows:

$$BR = \text{overhead} + \text{side information} + \text{new label}$$

$$= BRo + BRs + BRnl$$

$$= N \times Bc/Np + 1/Nb + Nc \times (\log_2 N)/Nf \quad (38)$$

where,

BRo - the bit rate for the codebook overhead;

BRs - the bit rate for the side information;

BRnl - the bit rate for transmitting the new labels;

Bc - the average number of bits per codeword;

N - the number of codeword in the codebook;

Np - the number of pixels in the input source;

Nb - the number of pixels per block;

Nc - the number of new labels;

Nf - the number of pixels per frame.

Note that the first two terms, the overhead required for transmitting the codebook, and the side information are fixed. The third term depends upon the number of labels changing between successive frames, and is, therefore, variable.

Returning to the vector formation step in Fig.20, each frame is first divided into two-dimensional blocks. As for the three-dimensional block vector quantization, two different strategies for vector formation are tested:

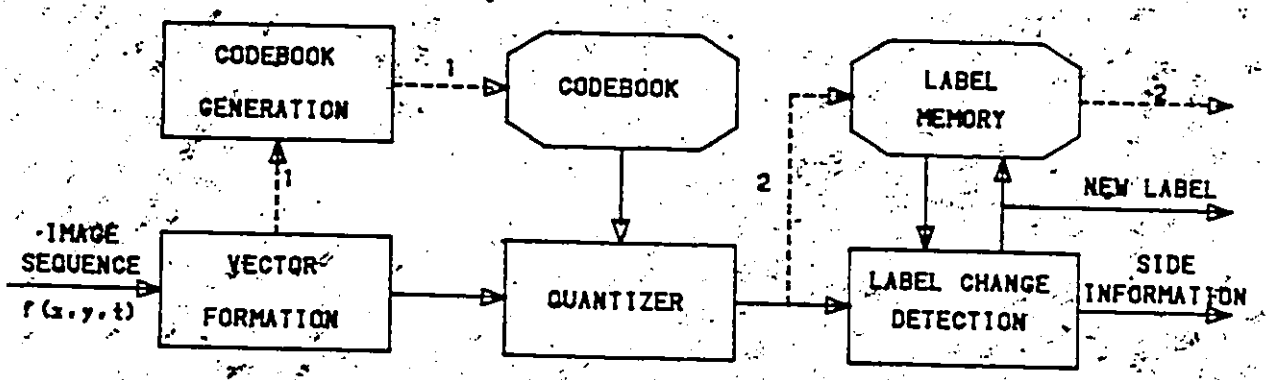


Fig.20 Schematic diagram for label replenishment on the direct two-dimensional vector quantization. Dashed line 1 corresponds to the codebook generation step using the first frame. Dashed lines 2 correspond to storage and transmission of the labels of the first frame. The remaining solid lines correspond to coding the subsequent frames.

i).Direct Two-Dimensional Block Vector Quantization,

ii) Two-Dimensional Block Hadamard Transformation with Vector Quantization.

In the first, the vectors are formed directly from two-dimensional blocks, by simply realigning the block into a linear array. Significant correlation remains between the components of the vector. In the second, the two-dimensional Hadamard transform is first applied to the block, thus decorrelating the components. The block mean, containing the most information, is scalar quantized; the remaining coefficients and then vector quantized (see Fig.21(a)). The second strategy requires a more elaborate encoder, which needs a quantizer, two separate frame memories and two bits of side information: one for the label as before and one for the block mean (see Fig.21(b)). For this scheme, the bit rate can be calculated as follows:

$$BR = \text{overhead} + \text{side information} + \text{new label} + \text{new mean}$$
$$= BRo + 2\times BRs + BRnl + BRnm \qquad (39)$$

where, the term $BRnm=Bm\times Nm/Nf$ is the bit rate for new block mean, Bm is the number of bits for block mean and Nm is the number of new block mean. The remaining terms are the same as in the equation (38). We note that in the codebook some additional reductions are obtained by scalar quantizing the components of codeword according to the energy distribution (36).

Comparing with equation (38) above, this scheme requires the same overhead, twice as much side information and an extra term for the new mean. The results shown in Fig.22 are obtained by using four different threshold values, 2, 4, 6, and 8 for replenishing the block means. For these values, four different codebook sizes, 64, 128, 256 and 512 are used as well. From the rate distortion curves shown in Fig.22, it can be seen that at low bit rates (lower than 0.6 bits/pixel) large threshold values give lower distortion, whereas, at higher bit rates (higher than 0.6 bits/pixel) small threshold values give better performance. If the threshold value is too low, insignificant changes are replenished resulting in only small additional reduction in distortion for a disproportionate increase in bit rate. A large threshold value would result in too many significant changes not being replenished. A threshold value of 4 is chosen as a compromise in the subsequent experiments.

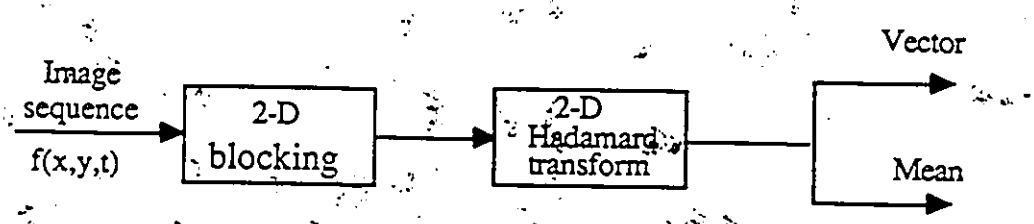


Fig.21(a) Vector formation and block mean separation step. The images are first divided into two-dimensional blocks and Hadamard transformed. The block mean (zero-order) component is separated, and the remaining coefficients are quantized and then used to form a vector.

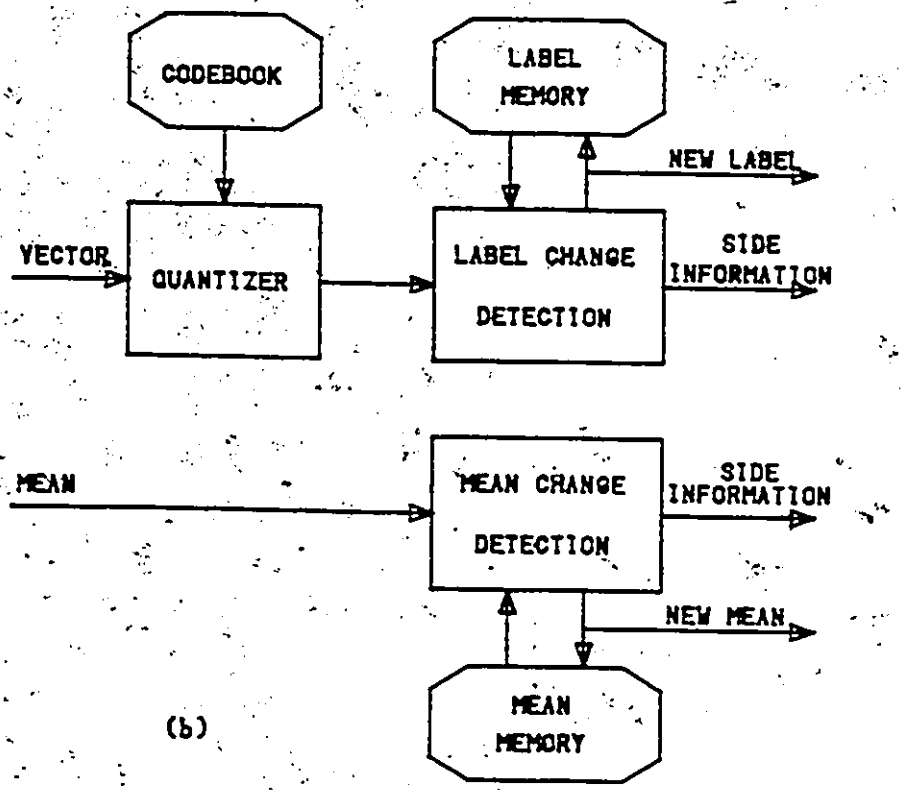


Fig.21(b) Encoder of two-dimensional Hadamard transformation with vector quantization involves a quantizer, two separate frame memories, and therefore, two bits of side information are needed: one for the label and one for the block mean.

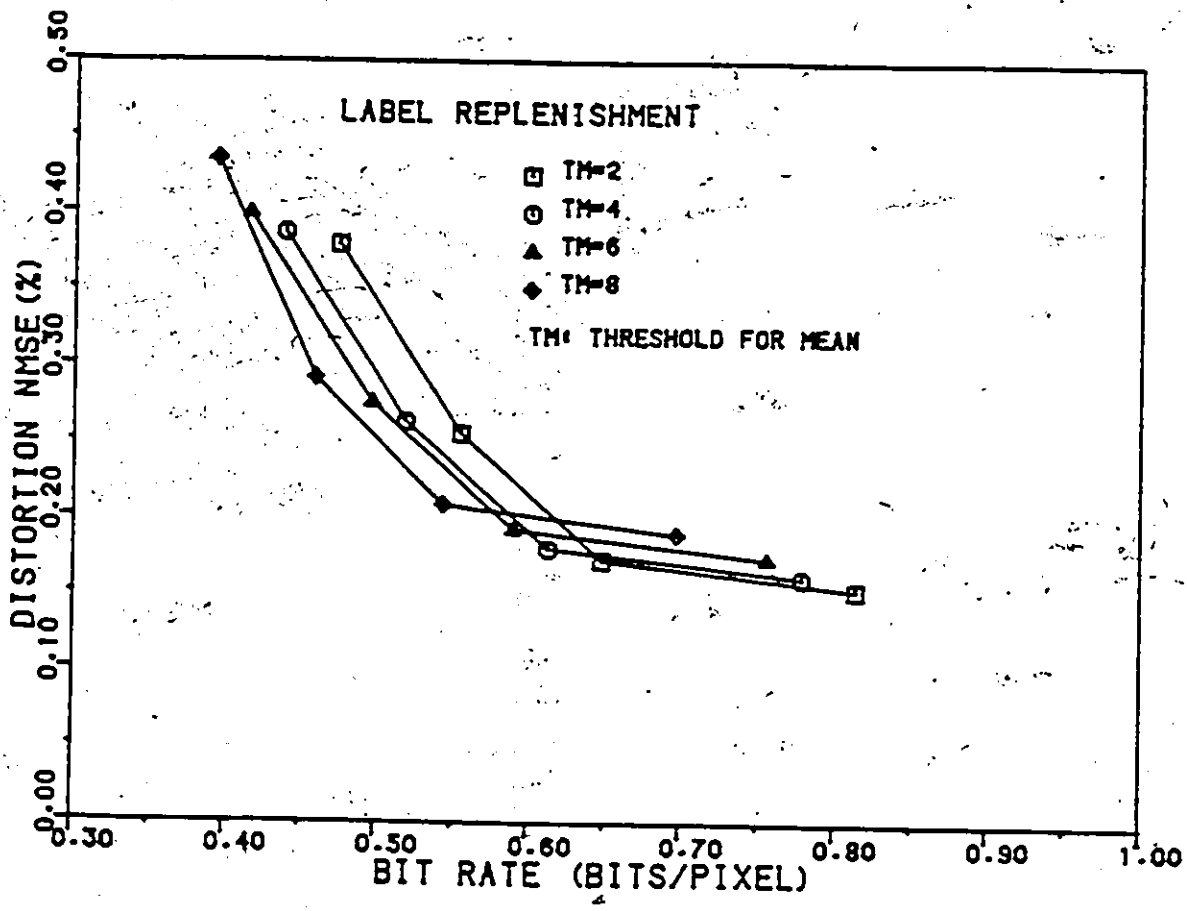


Fig.22 Rate distortion curves for different threshold values used to replenish the block mean term. Here, a block size of 4x4 is used and after Hadamard transform the mean (zero-order) component is separated and the remaining coefficients are quantized and used to form a vector. The codebook size is 256. At low bit rates (lower than 0:6 bits/pixel) large threshold values give lower distortion, whereas, for higher bit rates, small threshold values are better.

5.2 CODEBOOK REPLENISHMENT

In vector quantization each input vector is encoded by two pieces of information: the label and the corresponding codeword or representative vector. In the algorithm presented in the last section, changes in the input source were tracked by replenishing only the label component. The underlying assumption made is that vectors from the changing areas can be acceptably encoded with codewords from the existing codebook; thus, only a new label need be transmitted. The main drawback to this approach is that a codebook which is representative of the entire sequence must be found. From the point of view of implementation, this implies that the statistics of the entire sequence be known beforehand. Furthermore, as demonstrated above, if the individual frames have different statistics, then a comparatively large codebook is required, so increasing the bit rate. An alternative strategy would be, therefore, to design a codebook which more closely matches the local frame statistics. One such algorithm for intraframe coding is described by Boucher and Goldberg [13], and uses a small codebook designed for different portions of the image. However, the new codebooks increase both the overhead and computation.

Instead of generating an entirely new codebook, it may be more advantageous to only update or replenish the codebook, especially if the changes are not large. The problem then becomes how to detect the changes and replenish the codebook.

At the same time, however, if the codebook is replenished in too dramatic a fashion, the amount of label replenishment increases correspondingly. Thus, it may be more advantageous, with respect to the overall performance, to restrict the type of replenishment for the codebook. With this observation in mind, two strategies for codebook replenishment are presented: Mean Shift and Mean Shift combined with Selective Codeword Replacement.

5.2.1 Codebook Replenishment by Mean Shift

The main idea behind the mean shift algorithm is that the codebook is replenished by simply updating the individual representative vectors. Recall that from equation (20), these representative vectors are chosen to be the expected mean value over the Voronoi regions. If the local (frame) input sources are used to estimate the means, then changes in the local statistics on a frame to frame basis can be tracked. It should be noted that the mean shift algorithm has an important merit; it preserves the original codebook structure as much as possible thus reducing the need for label replenishment. This scheme was first proposed in [23]; a detailed presentation follows.

The steps involved in the mean shift codebook replenishment are shown in schematic form in Fig.23. The first frame is normally used to generate the initial codebook in the standard manner:

$$w_i^1 = E\{v^1 \mid v^1 \in R_i^1 \text{ , and } v^1 \text{ is the training set}\} \qquad (40)$$

$$i = 1, 2, \ldots, N.$$

Note that here E denotes the sample mean. For subsequent frames, in particular the j-th frame, there are three distinct steps:

1. Quantization: Each input vector v^j is assigned to the nearest Voronoi region, R^{j-1}, using the nearest distance rule:

$$|v^j - w_k^{j-1}| = \text{Min}_i \{|v^j - w_i^{j-1}|\} \quad (41)$$

Note that the codebook from the previous frame is used for the quantization. The label k is then compared with the one in the label memory.

2. Updating Representative Vectors (means): The vectors of the current frame, v^j, are used to update the representative vectors (cluster means) and corresponding Voronoi region.

$$w_i^j = E\{v^j \mid v^j \in R_i^{j-1}\}, \quad i=1, 2, \ldots, N \quad (42)$$

$$R_i^j = \{v^j : |v^j - w_i^j| \leq |v^j - w_m^j|, \text{ for all } m \neq i\} \quad (43)$$

Note that v^j are the vectors formed from the j-th frame in the sequence.

3. Codebook replenishment: The current representative vectors $\{w^j\}$ are compared with the previous set $\{w^{j-1}\}$. If

the difference is greater than a preset threshold, T_h, the codebook is replenished at both the transmitter and receiver:

$$w_i^j = w_i^{j-1} \text{ , if } |w_i^j - w_i^{j-1}| \leq T_h \tag{44}$$

and

$$w_i^j = E\{v^j \mid v^j \varepsilon R_i^{j-1}\}, \text{ if } |w_i^j - w_i^{j-1}| > T_h \tag{45}$$

It is easily demonstrated that by tracking the shift in mean, the distortion is reduced. Let Do and Ds be the distortion before and after the shift in mean, then

$$\begin{aligned} Do &= E\{(v^j - w^{j-1})^2\} \\ &= E\{(v^j - w^j + w^j - w^{j-1})^2\}. \end{aligned}$$

As the cross terms are zero,

$$E\{(v^j - w^j)(w^j - w^{j-1})\} = 0 ,$$

thus, Do can be rewritten as,

$$Do = E\{(v^j - w^j)^2\} + E\{(w^j - w^{j-1})^2\} \tag{47}$$

However, the first term corresponds to Ds,

$$Ds = E\{(v^j - w^j)^2\} \tag{48}$$

So that, we can conclude that the distortion is reduced by using the new codebook.

$$Ds \leq Do \tag{49}$$

We note in passing, that the mean shift algorithm is equivalent to using the previous codebook, $\{w^{j-1}\}$, as seeds for the clustering algorithm used for codebook generation. The new codebook, $\{w^j\}$, however, is formed by applying only one iteration of this algorithm to the input vectors, $\{v^j\}$, of the j-th frame.

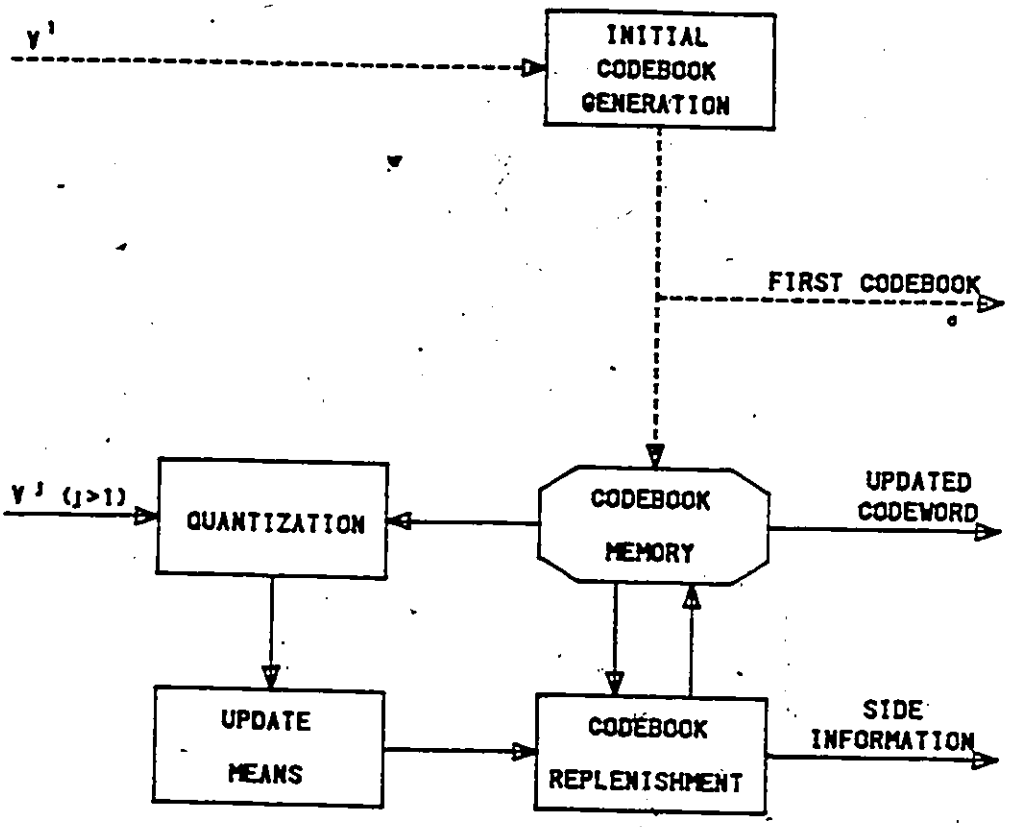


Fig.23 Codebook replenishment by Mean Shift. The dashed lines correspond to the initial codebook generation step. The solid lines correspond to coding the subsequent frames. The input vectors are first quantized by the previous codebook stored in the codebook memory, and then used to update the codewords and the corresponding Voronoi regions. The current codewords are compared with the previous ones, and replenishment carried out if necessary.

5.2.2 Mean Shift and Selective Codeword Replacement

The mean shift algorithm for replenishing the codebook is applicable if the frame to frame changes are small. However, when changes occur which alter the local statistics in a more significant fashion, simply shifting the mean may no longer be sufficient. The codebook must, therefore, be replenished in a more complex fashion. One possible strategy would be to find a completely new codebook by applying an iterative clustering algorithm with the present means as seeds. However, this could change the correspondence between the frames at the label level. An alternative strategy is to create new codewords that can represent the changes, while at the same time preserving some of the previous codewords. A measure based upon the maximum expected distortion for each Voronoi region is introduced below and used to decide whether a new codeword should be created.

Recall that the vector quantizer, Q, is completly described by the codebook, $W=\{w_1, w_2, \ldots, w_N\}$, together with the disjoint partition, $R = \{R_1, R_2, \ldots, R_N\}$, where $R_i=\{v: Q(v)=w_i\}$, and v is in the training set. We now define the maximum distortion of the region R_i:

$$d_{imax} = \underset{v \in R_i}{\text{Max}} |v - w_i| \qquad (50)$$

With respect to d_{imax}, a new set of regions, $R'=\{R'_1, R'_2, \ldots, R'_N\}$ is defined as follows:

$$R'_i=\{v:\ |v-w_i|\le d_{imax} \text{ and } |v-w_i|\le|v-w_j| \text{ for } i\neq j\ \} \qquad (51)$$

The first constraint defines a hypersphere in the K-dimensional space centered at w_i , the second constraint removes those subregions of the hypersphere which intersect with the other Voronoi regions (R_m , $m\neq i$). If the complement of R'_i with respect to R_i is taken, then a new region O_i is defined such that:

$$R_i = R'_i \cup O_i \qquad (52)$$

The test for creating new codewords will now depend upon the number of vectors lying in region O_i , that is, quantized as the i-th codeword, but with error greater than d_{imax} . An adaptive codebook replenishment strategy which combines the mean shift with selective codeword replacement is illustrated in Fig.24. The following three phases can be distinguished: -

1. Initial Codebook Generation: As before, the first frame is used as a training set to generate a codebook, $\{w^1\}$. In addition, the maximum distortion, $\{d_{imax}\}$ with respect to the training set, is evaluated and stored.
2. Quantization and Mean shift: For subsequent frames, in particular the j-th frame, each vector is assigned to the nearest codeword in the codebook $\{w^{j-1}\}$. After quantization, the same algorithm as in section 5.2.1, the mean shift, is used, that is,

$$w_i^j = w_i^{j-1} \quad , \text{ if } |w_i^j - w_i^{j-1}|\le T_h$$

and

$$w_i^j = E\{v^j \mid v^j \varepsilon R_i^{j-1}\}, \text{ if } |w_i^j - w_i^{j-1}| > T_h$$

In this step, the updated codewords and side information are not transmitted.

3. Requantization and Replenishment: During this phase the vectors are assigned to $\{O_i^j\}$, $\{R'^j_i\}$, by using equation (51). In other words, the vectors of each Voronoi region, R_i are assigned to two regions, one where the distortion is less than d_{imax}, and the second where it is greater than d_{imax}. The strategy is now to replace those w^j, which correspond to the empty or near empty regions. The new regions are chosen among $\{O_k^j\}$ and the corresponding codewords are calculated as follows:

$$w_i^j = E\{v^j \mid v^j \varepsilon O_k^j\}. \qquad (55)$$

Computer simulations have been performed for both codebook replenishment algorithms; the results are provided in the next section. The bit rate can also be calculated with equation (40), but the overhead portion, BRo, now includes the new codewords.

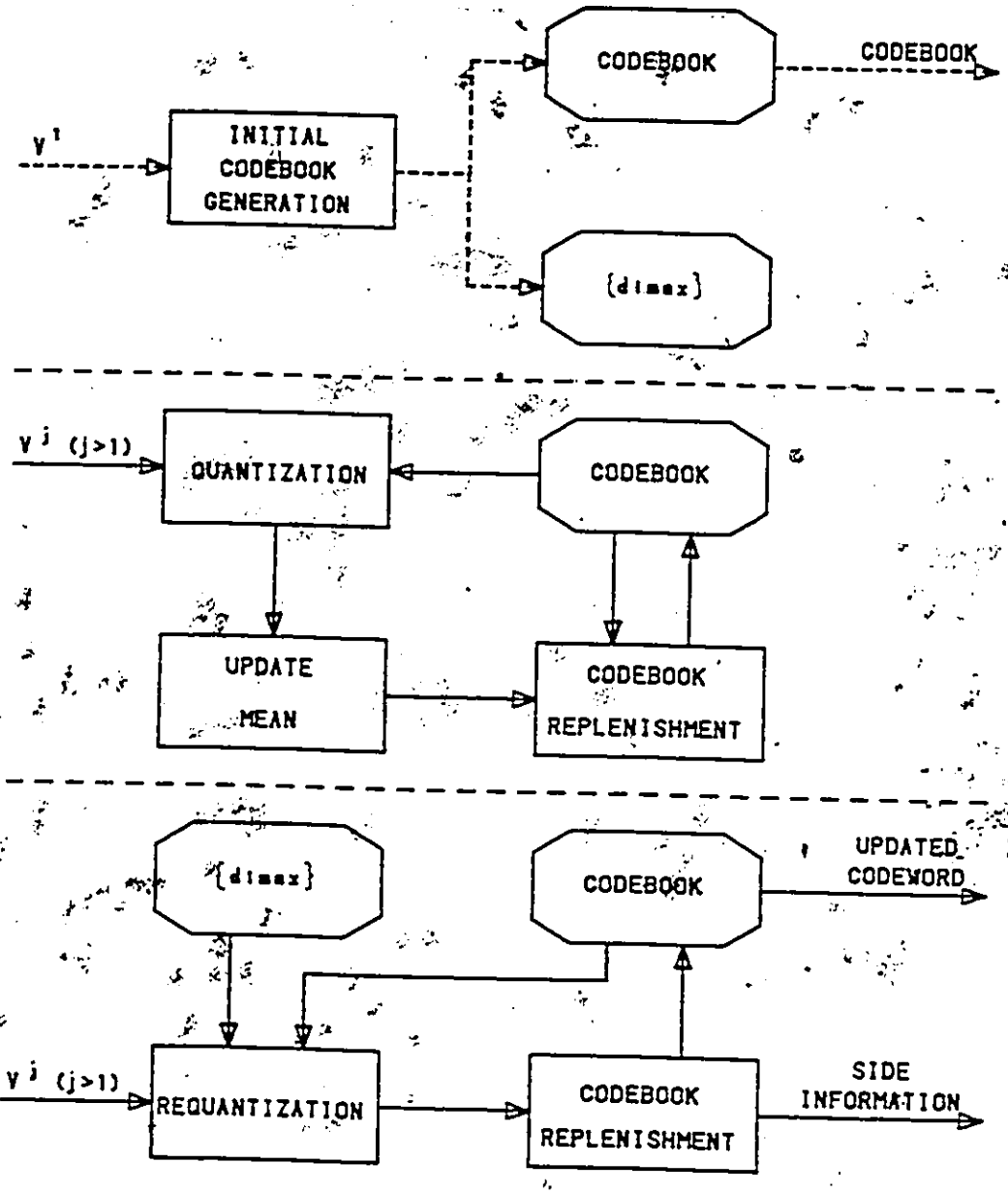


Fig.24 The schematic diagram of Mean Shift with Selective Codeword Replacement. The procedure includes: (i) Initial codebook generation: the first frame is taken as a training set to generate the codebook and the maximum distortions $\{d_{imax}\}$, (ii) Quantization and mean shift: for the subsequent frames, each vector is assigned to the nearest codeword, and used to update the region mean, the codebook is then updated by shifting the mean; (iii) Requantization and Replacement: The vectors of each Voronoi region, R_i, are assigned to two regions, one where the distortion is less than d_{imax}, and the second where it is greater than d_{imax}. The strategy is now to replace those v_i, which correspond to the empty or near empty regions, $\{R_i^j\}$, by the mean of those $\{O_k^j\}$, which are more populated.

5.3 EXPERIMENTAL RESULTS

The three replenishment -based coding schemes described above, - label replenishment, label replenishment combined with mean shift, and label replenishment combined with mean shift and selective codeword replacement - were applied to the same picturephone sequence as in the three-dimensional block vector quantization case. The corresponding rate distortion curves are presented in Fig.25, where the bit rate and normalized mean-squared error (NMSE) have been averaged over the 20 frames. For each scheme, the four values given correspond to using codebooks of size 64, 128, 256 and 512, respectively. The first three entries correspond to the use of a codebook with 256 codewords; for the fourth entry, 512 codewords were used.

In terms of the average rate distortion, at the low bit rate, the label replenishment and label replenishment with mean shift are very similar. Both are better than selective codeword replacement. At bit rates above 0.6, label replenishment combined with codebook replenishment (both mean shift and mean shift with selective codeword replacement) gives about 15% improvement in NMSE over straight label replenishment. A possible explanation is that at the lower bit rate, there are few codewords, forcing each to represent broad classes of vectors. Hence, in the bit rate calculation formula (38), the portion for new labels, BRnl, tends to be small, and

overhead (BRo) becomes a important factor in the bit rate. Whereas, at higher bit rates, there are more codewords giving a finer representation of the classes of vectors. The finer representation is, therefore, more sensitive to small interframe variation. In equation (38), the portion for new labels (BRnl) becomes dominant. Updating the codewords gives more benefits in reduction of distortion at limited expense to bit rate.. Of further note is the only marginal increase in performance achieved by selective codeword replacement compared to the mean shift. This may be due to the increase of the overhead with marginal reduction in distortion.

An important feature of codebook replenishment, illustrated in Fig.26(a), is the ability to track interframe variation and, by suitably modifying the codebook, to maintain a more constant distortion over the entire sequence. In Fig.26(a), it is seen that the three replenishment coding schemes behave identically for the first 9 frames, where there is little motion. At frame 10, an abrupt change begins and the distortion for label replenishment increases by about 40% and remains high for the rest of the sequence.. However, label replenishment combined with codebook replenishment is able to adapt to the change and maintain almost the same level of distortion throughout the sequence. In terms of bit rate, the codebook replenishment requires more bits, but as illustrated in Fig.26(b), they are allocated selectively to the latter frames which contain more changes.

An important observation is the significant improvement of two-dimensional vector quantization with replenishment as compared to using vector quantization on the three-dimensional blocks (see chapter 4) For bit rates between about 0.5 and 0.65, the NMSE is reduced by a factor at least 2. This can be seen from Table 5. The data in Table 5 are drawn from Fig.16 and Fig.25, and reproduced.

Table 5

	Bit Rate	%NMSE	Codebook size
Label Replenishment (LR)	0.584	0.180	256
LR with Mean Shift (MS)	0.645	0.154	256
LR With MS and Selective Codeword Replacement	0.658	0.152	256
LR with MS	0.790	0.135	512
HT-VQ (12 coefficients)	0.494	0.461	512
HT-VQ (18 coefficients)	0.609	0.422	512

Table 5 Comparative rate distortion values for the three-dimensional vector quantization and two-dimensional vector quantization with replenishment. The HT-VQ data is from chapter 4.

The visual results presented in Fig.27 corroborate the rate distortion curves shown in Fig.25. In Fig.27(a) are shown the coded images of the first frame using respectively, the three-dimensional block Hadamard transform with vector quantization on the 12 lowest non-zero order coefficients (from chapter 4); label replenishment with 256 codewords; and label replenishment with 512 codewords. The two coded images using label replenishment are clearly superior to the three-dimensional Hadamard transform with vector quantization. Fig.27(b) shows the images of the fourth frame coded by label replenishment at bit rates 0.584 bits/pixel and 0.73 bits/pixel, respectively for the top right and bottom right. The reproduced image at higher bit rate, 0.73 bits/pixel, shows improved appearance on the right eye, the hair and the cheek region. Fig.27(c) presents the results for frame 10, where significant interframe changes are begining. The three coded images clockwise stated from top right correspond to, respectively, label replenishment with 256 codewords at bit rate 0.584 bits/pixel, label replenishment with 256 codewords combined with mean shift at bit rate 0.645 bits/pixel, and label replenishment with 512 codewords combined with mean shift at bit rate 0.79 bits/pixel. In both cases where the mean shift is used for codebook replenishment, improved reproductions are visible over the right eye and in the highlights of the hair. Finally Fig.27(d) presents the results for the 14-th frame, which is in a zone of rapid change. The original image is blurred so

that it is better to make comparison in the zone of the background. Note that as the speaker turns her head, a new portion of the background has been exposed and that better reproduction is achieved with the mean shift. To contrast the differences between the three coded images, the corresponding error images are shown in Fig.27(e) and Fig.27(f), respectively, for the 10-th frame and the 14-th frame, where middle grey corresponds to no error. The improvement with codebook replenishment is confirmed by the error images. The error is more uniformly distributed through the image and, in particular, shows that the edges have in general been better reproduced.

5.4 DISCUSSION AND CONCLUSIONS

In this chapter, a new interframe coding technique based upon vector quantization is presented. This algorithm has as its basis two-dimensional block vector quantization, at the label level, onto which is grafted the concept of adaptive codebook replenishment and frame replenishment. For label replenishment, the underlying assumption made is that vectors from changing areas can be acceptably encoded with codewords form the existing codebook. However, if the individual frames have different statistics, the codebook, which is obtained from the first frame, must be updated to track these changes. This is implemented by label replenishment with mean shift and label replenishment with mean shift and selective codeword re-

placement. These strategies are very efficient and exploit the advantages of both frame replenishment and vector quantization. Compared with three-dimensional block vector quantization [19], at bit rates between 0.5 and 0.65, the NMSE is reduced by a factor of at least 2 by the use of the replenishment strategies. However, a practical consideration is the need to use a variable bit rate. In exchange for this variable rate, however, the scheme is highly adaptive and allocates more bits to those frames with large changes, thus, maintaining a more constant level of distortion throughout the sequence.

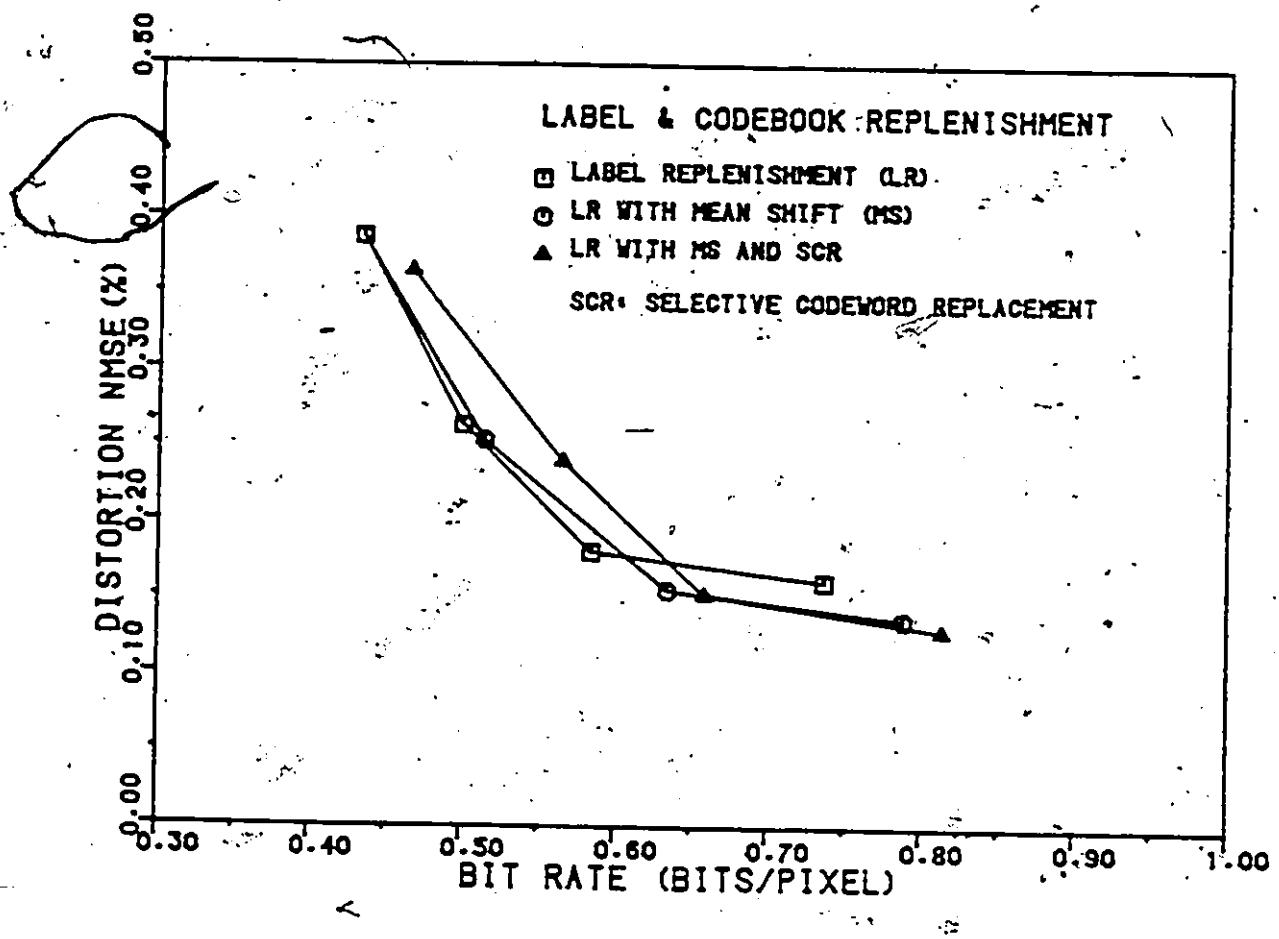


Fig.25 Rate distortion curves for label and codebook replenishment, where, the bit rates and the NMSE values are the averages over 20 frames. Three coding schemes are shown, label replenishment, label replenishment with mean shift, label replenishment with mean shift and selective codeword replacement. At a bit rate above 0.6, label replenishment combined with codebook replenishment gives a 15% improvement in NMSE over straight label replenishment.

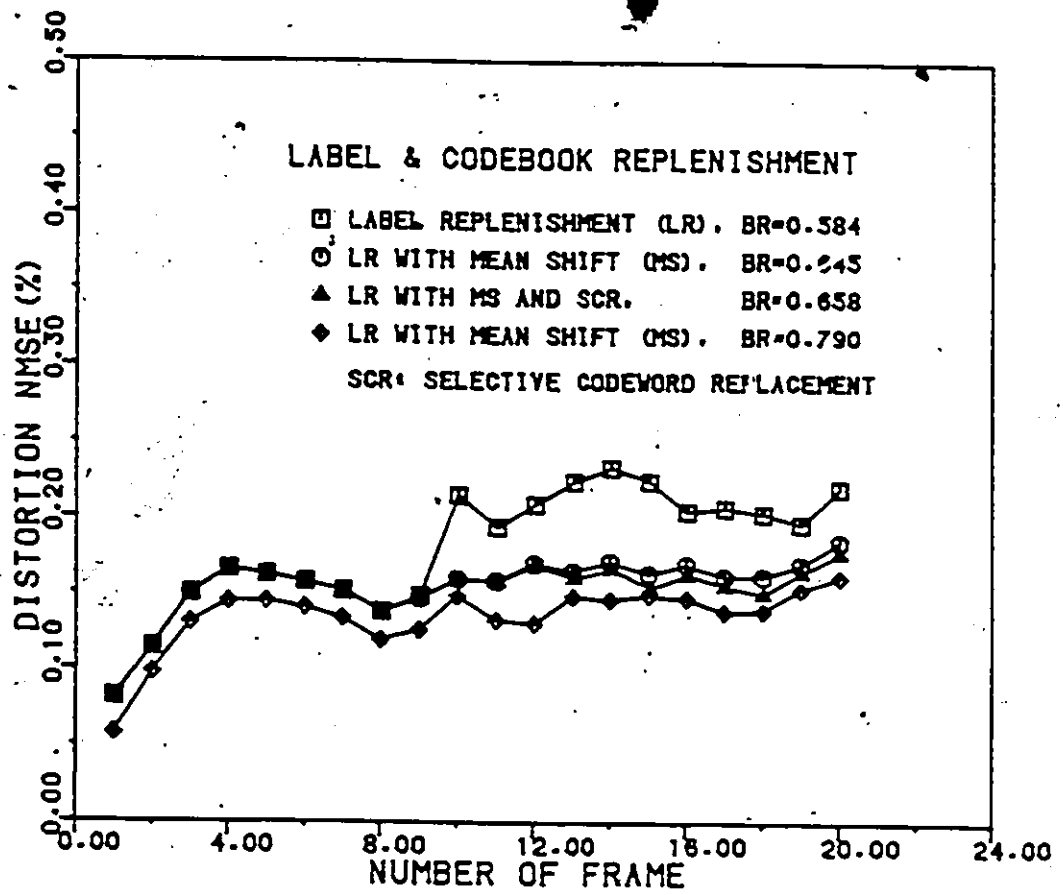


Fig.26 (a) The distortion (NMSE) as a function of the frame number for the three replenishment schemes. Note that label replenishment combined with codebook replenishment maintains a relatively constant distortion even in the frames (10-20) containing changes. The first three schemes use 256 codewords and the last 512 codewords.

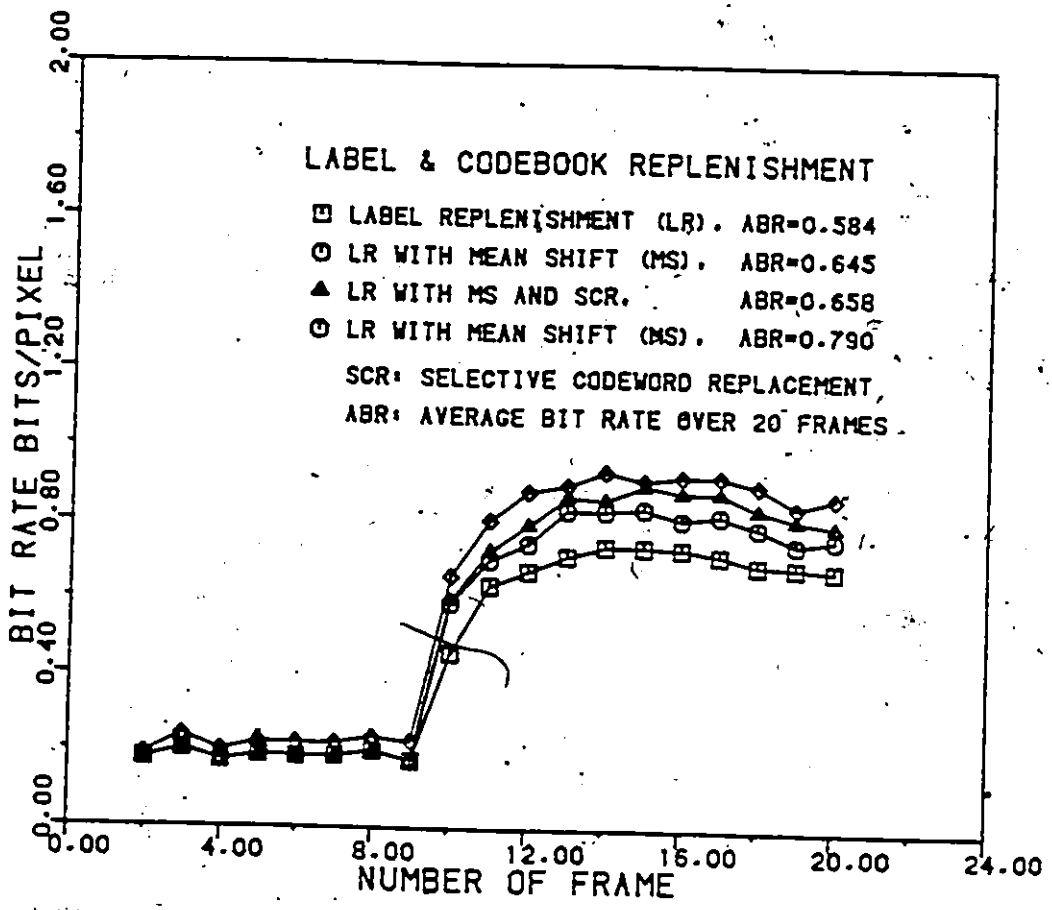


Fig.26 (b) The bit rate as a function of the frame number for the three replenishment schemes. Codebook replenishment allocates more bits to the frames (10-20) containing changes, thus resulting in the more constant distortion shown in Fig.26(a).

Fig.27(a) The pictorial results for the first frame (the top left is the original). The three coded images correspond to, respectively, the three-dimensional block Hadamard transform with vector quantization on the 12 lowest non-zero order coefficients (bottom left, 0.494 bits/pixel) label replenishment with 256 codewords (top right, 0.584 bits/pixel), and label replenishment with 512 codewords (bottom right, 0.79 bits/pixel). Note that the images coded with label replenishment are clearly superior.

Fig.27(b) The pictorial results for the fourth frame. The two coded images correspond to, respectively, label replenishment with 256 codewords (top right), and label replenishment with 512 codewords (bottom right). A slight degradation is noticeable in the area of the lips, especially for the smaller codebook.

Fig.27(c) The pictorial results for the 10-th frame. The three coded images correspond to, respectively, label replenishment with 256 codewords (top right), label replenishment with 256 codewords combined with mean shift (bottom left), and label replenishment with 512 codewords combined with mean shift (bottom right). The improvements are especially noticeable for the mean shift cases over the right eye and the highlights of the hair.

Fig.27(d) The pictorial results for the 14-th frame, which is in a zone of rapid change. The three coded images correspond to, respectively, label replenishment with 256 codewords (top right), label replenishment with 256 codewords combined with mean shift (bottom left), and label replenishment with 512 codewords combined with mean shift (bottom right). Note that as the speaker turns her head a new portion of the background has been exposed and that the better reproduction is achieved with the mean shift.

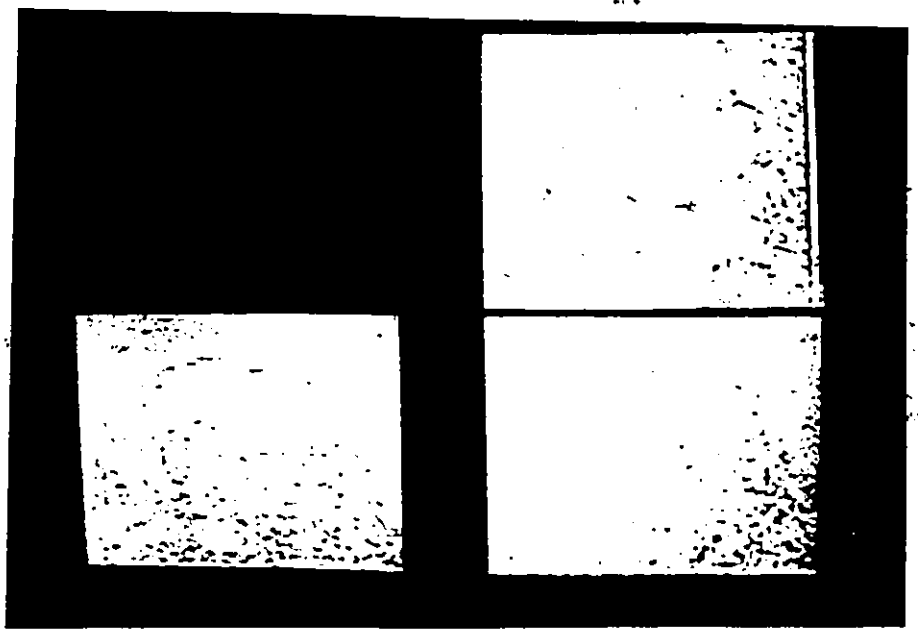


Fig.27(e) The error images corresponding to the 10-th frame. Note that the error is much uniformly distributed for the scheme with mean shift, which results in better reproduction.

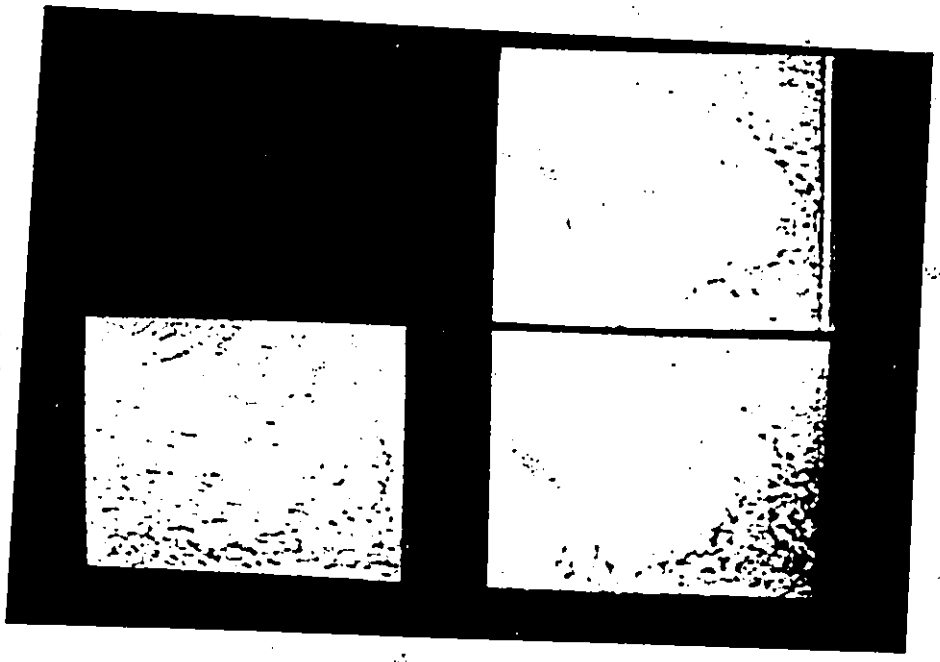


Fig.27(f) The error images corresponding to the 14-th frame. Note that the error is more uniformly distributed for the scheme with mean shift, which results in better reproduction.

Chapter VI

FRAME ADAPTIVE VECTOR QUANTIZATION

In the last chapter, two strategies for codebook replenishment have been proposed. This chapter presents a third, Frame Adaptive Migrating Mean (FAMM) codebook replenishment.

6.1 INTRODUCTION

In intraframe coding by vector quantization, the codebook can be designed optimally by providing a long training sequence drawn from the source, thus obviating the need for an accurate source model [55]. However, in the image sequence case, the problem becomes complex, since this implies that a codebook which is representative of the entire sequence must be found. As indicated in the last chapter, from the point of view of implementation, if the individual frames have very different statistics then a comparatively large codebook is required to represent the sequence, resulting in an increased bit rate. An alternative strategy could be, therefore, to design a smaller codebook which more closely represents the local frame statistics. This codebook must then be updated from frame to frame. This implies that vector quantization must include some adaptive schemes. For designing an adaptive system, two parameters are necessary. The first is used to ini-

tiate the adaptive operation, and the second, to track the variations of the local source, i.e. the individual frames.

6.1.1 Codebook generation

The adaptive operation presented updates the codebook from frame to frame by tracking the variation of the local source. In codebook generation, the K-means [6] or the generalized Lloyd clustering algorithm [7] is usually used. This clustering algorithm is an iterative process, minimizing a performance index calculated from the distances between the sample vectors and their respective cluster centers. In each iteration, two steps are included: (1) reassignment of the sample vectors to the nearest cluster center, the label reassignment step; (2) computation of the new cluster centers, the migrating (cluster) mean step. In this chapter, we propose a new algorithm, Frame Adaptive Migrating Mean (FAMM) codebook replenishment, to perform the adaptive operation. This algorithm is generalized from the mean shift codebook replenishment and label replenishment algorithm proposed in the last chapter. The main idea behind this new algorithm is that a new codebook is generated by using the previous codebook as seeds for the clustering algorithm used for codebook generation, where the number of iterations depends on the type of changes occurring in the input image sequence.

6.1.2 Statistical parameters

In this chapter, two different sets of statistics are proposed to track the type of changes occurring on a frame-to-frame basis. The first set contains first-order statistical parameter extracted from the histogram of the difference images calculated from successive frames, which is particularly efficient for medical image sequences. In the general case, the changes involved in an image sequence may be more complex. The second statistical parameter, the correlation coefficient between successive frames, is proposed to track the changes occurring in an input sequence on a frame-to-frame basis.

6.1.3 Main results

The experimental results for the angiographic sequence of x-ray images provided by CGR (France) show that at most two iterations are needed to adapt the codebook. For small changes, such as noise, only label reassignment is required. For the changes involving simple translations one complete iteration, that is, label reassignment followed by the migrating mean step is required, while for more complex and significant changes, such as occlusion, two complete iterations are needed. The first-order statistics of the difference images formed from the successive frames can be used to indicate how many iterations are needed for generating the new codebook.

The picturephone sequence is used to test the FAMM codebook replenishment algorithm. The experimental results for the picturephone sequence show that at most three iterations are required to adapt the codebook. For small and simple changes only label reassignment is required. For complex changes involving occlusion and scene changes, one to three iterations are required depending upon the extent of the changes occurring between the successive frames. The correlation coefficients between the successive frames can be used as a measure of the changes, and therefore, can be used to control the number of iterations in FAMM.

6.2 FRAME MIGRATING MEAN CODEBOOK REPLENISHMENT

The individual frames of an image sequence may have different statistics, therefore on a frame basis, the optimal codebooks must be generated from the local sources. However, it is clear that there will be redundancy between codebooks and more computations will be needed to calculate the individual codebooks. An alternative strategy, is to update or replenish a part of the codebook, so that the new codebook more closely matches the local statistics of the current frame.

Recall that, for a given training set, the codebook is obtained by using an iterative clustering algorithm such as the K-means or the generalized Lloyd algorithm proposed by Linde, Buzo and Gray [7]. In this clustering algorithm, each itera-

tion includes two steps: (1) label reassignment, (2) (cluster) mean migration. It has been proven that the iterative process is convergent, i.e., after each iteration, the distortion between the training vectors and the cluster centers will be reduced. This procedure only stops when the distortion is no longer reduced. The convergence rate depends critically upon the initial seeds [7]. Since successive frames tend to be similar, a rapid convergence may result, if the previous codebook is taken as the seeds for generating the new codebook for the current frame. The Frame Adaptive Migrating Mean (FAMM) codebook replenishment algorithm is based upon the above observation.

The steps involved in FAMM algorithm are shown in schematic form in Fig.28. The first frame is normally used to generate the initial codebook in the standard manner:

$$w_i^1 = E\{v^1 \mid v^1 \varepsilon R_i^1 \text{ , and } v^1 \text{ training set}\} \qquad i=1,2,\ldots,N \tag{56}$$

Here, the vectors, v^1, are formed from the first frame, and $\{w^1\}$ and $\{R^1\}$ are the codewords and Voronoi regions, respectively. For subsequent frames, in particular, the j-th frame, there are four distinct steps:

1. Quantization: Each input vector, v^j, is assigned to the nearest region, R^{j-1}, using the nearest distance rule:

$$|v^j - w_k^{j-1}| = \underset{i}{\text{Min}} \{|v^j - w_i^{j-1}|\} \tag{57}$$

Note that the codebook from the previous frame is used for the quantization.

2. Migrating (cluster) mean step: The vectors of the current frame, v^j, are used to update the representative vectors (cluster mean) and the corresponding Voronoi regions, R^{j-1}:

$$w_i^j = E\{v^j \mid v^j \in R_i^{j-1}\}, \quad i=1,2,\ldots,N$$

$$R_i^j = \{v^j : |v^j - w_i^j| \leq |v^j - w_m^j|, \text{ for all } i=m\}$$

Note that the vectors, v^j, are formed from the j-th frame in the sequence.

3. Convergence detection: Step 1 and 2 are repeated until the distortion between the input vectors and the current cluster centers (means) is no longer reduced by some preset threshold.

4. Replenishment: The replenishment process includes two parts; (i) Label replenishment: each vector is quantized, the label is then compared with the one in the label memory, and if it differs, the memory is replenished. (ii) Codebook replenishment: the current representative vectors $\{w^j\}$ are compared with the previous ones, $\{w^{j-1}\}$. If the difference is greater than a preset threshold value, the codebook is replenished at both the transmitter and receiver.

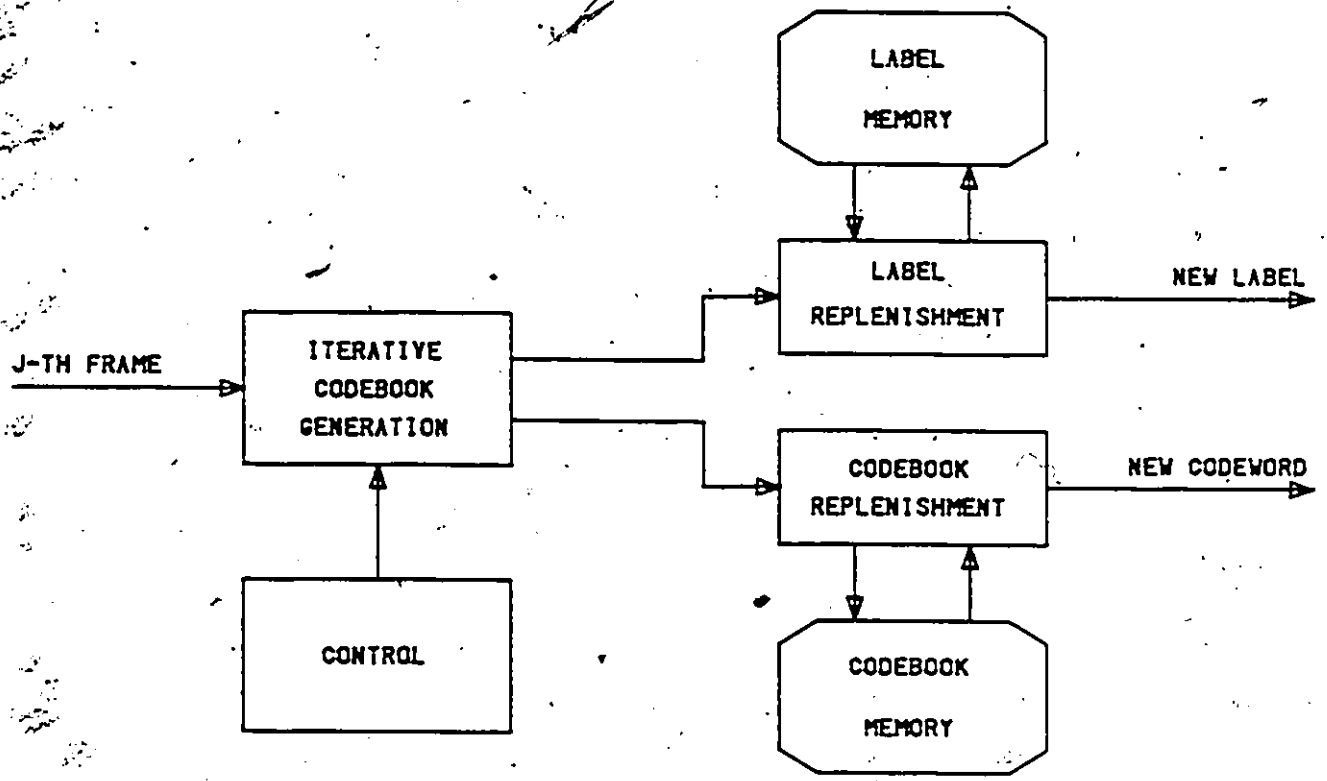


Fig.28 Schematic diagram for FAMM codebook replenishment algorithm. The initial codebook and labels in the memories are obtained from the first frame by vector quantization. The number of iterations required can be controled by parameters extracted from the first-order statistics of the difference images. After a new codebook has been generated, the codebook and label are replenished at both transmitter and receiver.

Note that in step 3, an extra iteration is required to detect the convergence. It is shown below that this step is not necessary, since the rate of convergence depends upon the changes occurring between the consecutive frames. Therefore, the number of iterations in FAMM algorithm can be predicted.

6.3 TWO CHANGE MEASURES

In this section, two change measures, the first-order statistics of the difference images and the correlation coefficient between the successive frames, are presented.

6.3.1 First-order statistics of difference images

A. First-order statistics

The first-order statistics of an image are given by the probability density function of pixel grey-level values without regard to the spatial distribution. The histogram provides an estimate of the grey level probability density function of the image. A uniqueness theorem states that if the probability density function, $P(z)$, of the image is piecewise continuous and has zero value only for a finite number of z, then the moments of all orders exist and the moment sequence (M_n) is uniquely determined by $P(z)$. The n-th order moment and its corresponding central moments are, respectively, defined as

$$M_n = \Sigma\, z^n P(z) \qquad (60)$$

and

$$\phi_n = \Sigma(z-M_1)^n P(z) \tag{61}$$

i) Measure of overall brightness: the mean of grey level of an image,

$$M_1 = \Sigma\, z\, P(z) \tag{62}$$

is a measure of the overall brightness of the image.

ii) Measure of overall contrast: the variance of the grey level,

$$\sigma^2 = \Sigma\, (z-M_1)^2 P(z) \tag{63}$$

is a measure of the overall contrast of the image, where a small value indicates that the grey level values of the image are all close to the mean value, and a large value indicates a wider range of grey levels.

iii) Degree of asymmetry: Skewness is a measure of the degree of asymmetry of a distribution. It can be measured as:

$$a_3 = \frac{\phi_3}{\sigma^3} \tag{64}$$

It is now shown how these parameters can be used to categorize image sequences.

B. Categories of difference images

The difference image $d^j(x,y)$ that results from subtraction of successive frames can be defined as follows:

$$d^j(x,y)=f^j(x,y) - f^{j-1}(x,y)$$
$$=[I^j(x,y)+N^j(x,y)]-[I^{j-1}(x,y)+N^{j-1}(x,y)] \quad (65)$$

where I^j , I^{j-1} and N^j , N^{j-1} are the information and the noise portion of frames f^j and f^{j-1} , respectively. If N^j and N^{j-1} are uncorrelated white Gaussian noise with the same variance, the mean and variance of d(x,y) can be obtained as follows:

$$M_1^j=E\{d^j(x,y)\} = E\{I^j(x,y)\} - E\{I^{j-1}(x,y)\} \quad (66)$$

and

$$\sigma_1^2=Var[d^j(x,y)] = Var[I^j(x,y)] + Var[I^{j-1}(x,y)]$$
$$- 2Cov[I^j(x,y), I^{j-1}(x,y)]$$
$$+ Var[N^j(x,y)] + Var[N^{j-1}(x,y)] \quad (67)$$

A useful model for an image sequence is a collection of objects on a uniform background. In comparing consecutive frames in the sequence, three categories of changes can be distinguished:

1. Noise: Only small changes occur between the successive frames and these can be modelled as random Gaussian noise. If no changes occur, the difference, d(x,y), between two successive frames should be zero mean with

the variance contributed only from the noise portion; that is,

$$E\{d^j(x,y)\}=0$$

$$Var[I^j(x,y)] = Var[I^{j-1}(x,y)] = Var[I(x,y)]$$

$$Var[I(x,y)] = Cov[I^j(x,y), I^{j-1}(x,y)]$$

Thus,

$$Var[I^j(x,y)]+Var[I^{j-1}(x,y)]-2Cov[I^j(x,y), I^{j-1}(x,y)]=0$$

If it is assumed that successive frames have the same noise statistics, that is,

$$Var[N^j(x,y)] = Var[N^{j-1}(x,y)]=Var[N(x,y)] =\sigma^2$$

Then from (67)

$$Var[d^j(x,y)] = 2Var[N(x,y)] = 2\sigma^2 \qquad (68)$$

Thus, the difference image can be considered as a random Gaussian noise and, therefore, the histogram is symmetrically distributed about zero mean with variance $2\sigma^2$.

2. Translations: If in addition to noise, some object translation occurs between the successive frames, the translation results in the same number of positive and negative pixels in the difference image, as explained in Fig.29. Therefore, the mean of the difference image is still zero; that is,

$$E\{d^j(x,y)\} = 0.$$

However, as the correlation between the successive frames decreases, $Cov[I^j(x,y), I^{j-1}(x,y)]$ decreases. Then,

$$Var[I^j(x,y)]+Var[I^{j-1}(x,y)]$$

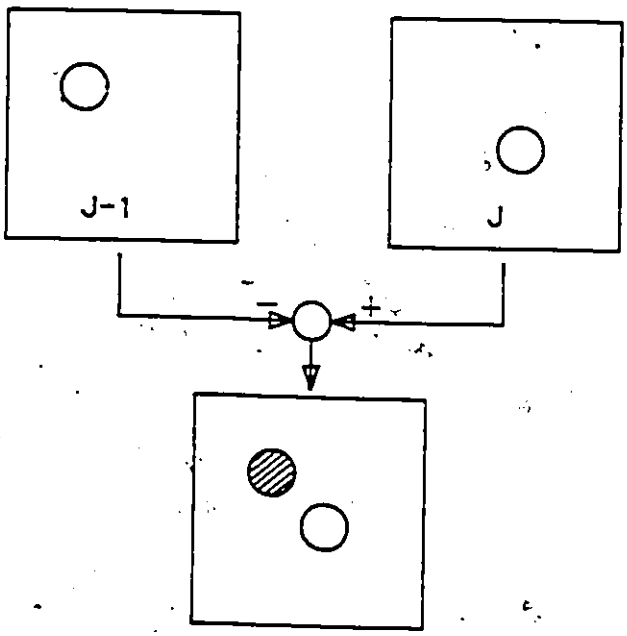
$$-2Cov[I^j(x,y), I^{j-1}(x,y)] = \sigma_t^2 > 0,$$

From (67), the variance of $d^j(x,y)$ now becomes larger:

$$Var[d^j(x,y)] = 2\sigma^2 + \sigma_t^2 > 2\sigma^2 \qquad (69)$$

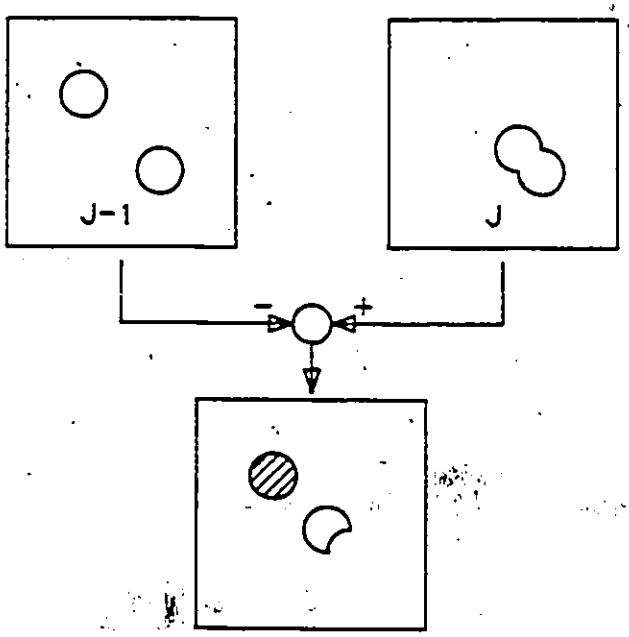
In other words, the variance involves two independent components: one from the noise portion and the second from the translation. Since translation results in the same numbers of positive and negative pixels in the difference image, the histogram is wider than in the no motion case, but is still symmetrical.

3. Occlusions: In this category, large changes are involved such as occlusion, scene translation, etc. In this situation the mean of the difference image is no longer equal to zero, as explained in Fig.30. the correlation between successive frames is decreased, and the variance of $d^j(x,y)$ increases. Furthermore, the difference image no longer contains equal numbers of positive and negative values, and the resultant histogram need not be symmetrical.



shaded area is negative, bright is positive.

Fig.29 Differencé image of translation. Note that translation results in same number of positive and negative pixels.



shaded area is negative value and bright is positive.

Fig.30 The difference image of occlusion image sequence. Note that the numbers of negative and positive values are not equal.

The above conclusions are summarized in Table 6. A narrow and symmetrical histogram implies that only the change between the successive frames is noise; a wider but still symmetrical histogram corresponds to the noise and some motion or translation; if the histogram is even wider and asymmetrical, occlusion and scene changes may exist. The behaviour of the statistical parameters: mean, variance and skewness of the histograms of the difference image corresponding to three types of image sequences is shown in Table 6.

Table 6

Type of image sequence	mean (M_1)	variance (σ^2)	skewness (a_3)
Noise	near zero	small	small
Translation	near zero	moderate	small
Occlusion	not equal zero	large	large

Table 6. Behaviour of the statistical parameters of the first-order statistics for three categories of image sequences.

C. Angiographic X-ray Image Sequence

Experiments were carried out on sequences of angiogram images supplied by CGR of France. Each image sequence is sampled

to accommodate the arrival and the wash-out of the opaque iodine agent, and the sample values are then quantized to 8 bits. The tests are performed on subimages of size 256 by 256. Image sequence IS-A consists of seven frames corresponding to the left ventricle, where only small changes and noise occur. IS-B consists of 7 frames obtained from an intravenous carotid angiogram examination with the patient swallowing, which causes a translation. IS-C consists of seven frames obtained from a cranial study, where large changes are involved.

The mean, variance and skewness of the second difference image of, respectively, IS-A, IS-B and IS-C are shown in Table 7.

Table 7

Type of image sequence	mean (M_1)	variance (σ^2)	skewness (a_3)
Noise	1.292	1.972	-0.286
Translation	-0.143	6.290	0.033
Occlusion	-6.175	25.293	-2.763

The results are representative of the other difference images in the sequence. The histograms of the difference images of above three image sequences are shown in Fig.31(a), 31(b) and 31(c), respectively. Here, the difference image is the dif-

ference of frame 2 and frame 3. These results are consistent with theoretical analysis shown in Table 6, and the numerical results can be used to initiate the adaptive operation.

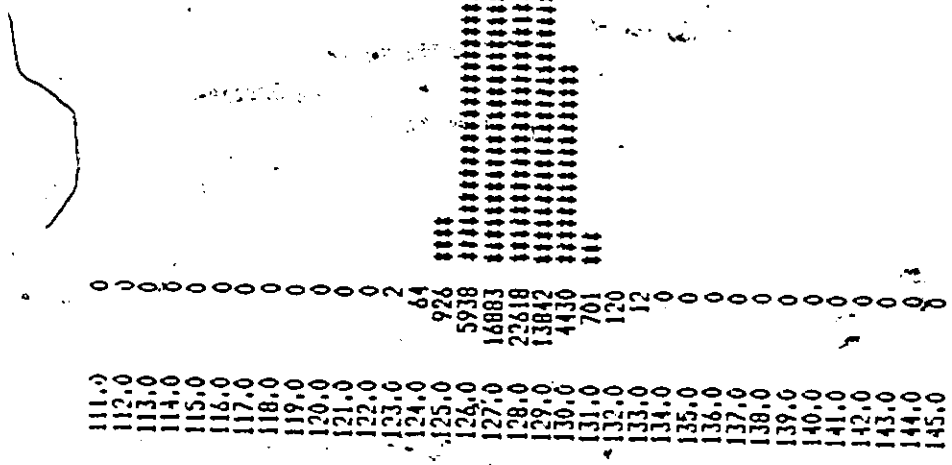


Fig.31(a) The histogram of the difference image with no motion, the histogram is sharp and almost symmetrical.

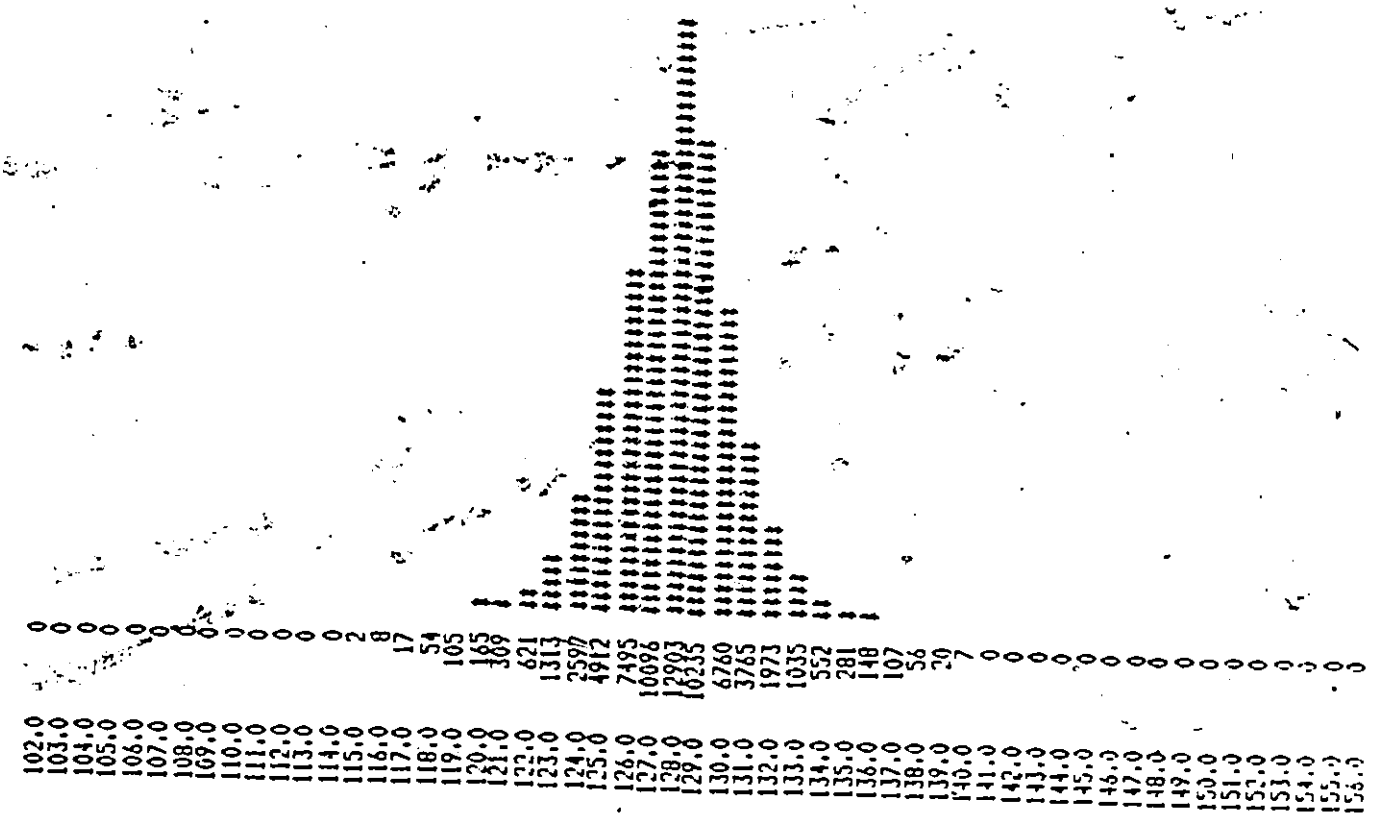


Fig.31(b) The histogram of the difference image with translation, the histogram is wide and symmetrical.

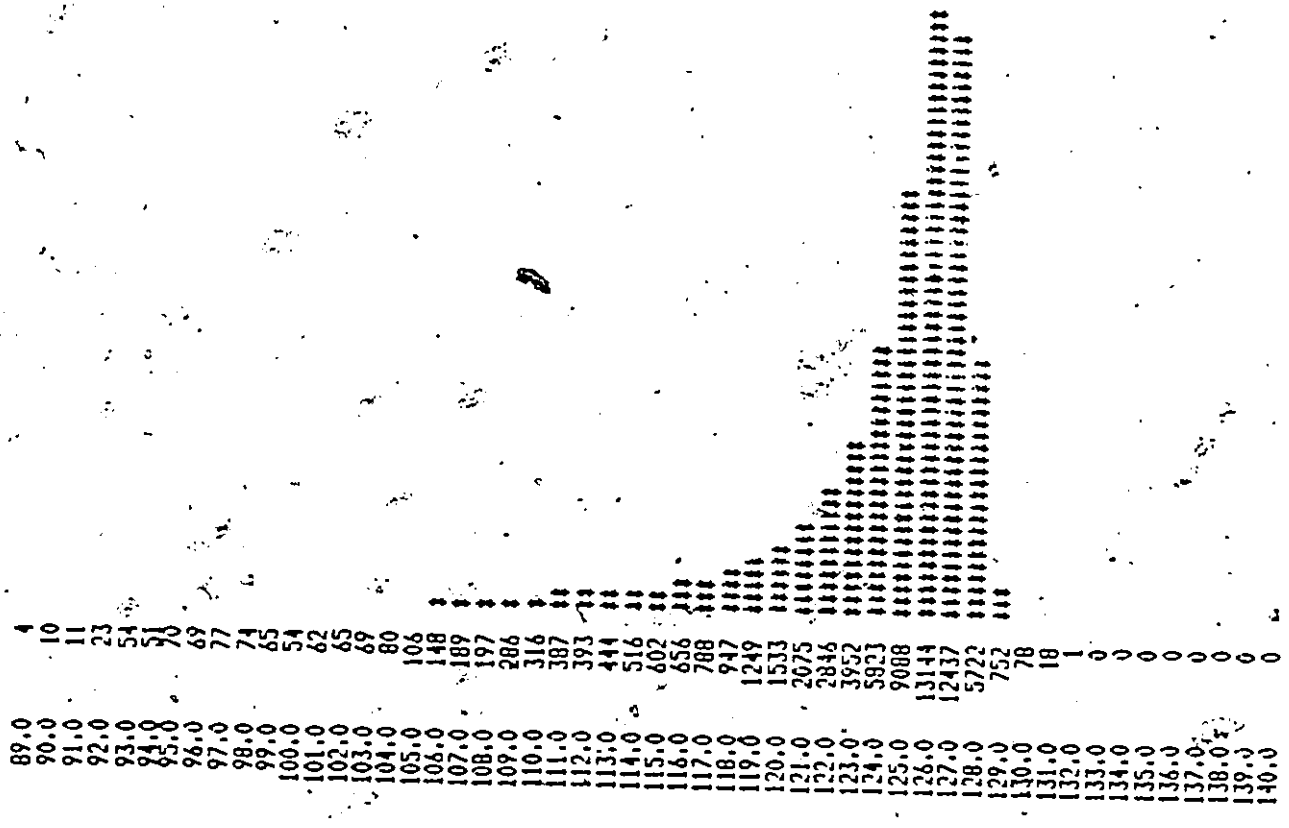


Fig.31(c) The histogram of the difference image with occlusion or scene changes, the histogram is wide and asymmetrical.

6.3.2 Correlation coefficient between the successive frames

For some complex image sequences, the changes occurring in the sequence cannot be classified into the three types proposed in the last section. In this section, we propose a correlation coefficient between successive frames to quantify the degree of change occurring in the input sequence.

A. Correlation coefficient of the successive frames

The correlation coefficient between the successive frames, f^j and f^{j-1}, is defined as [69]:

$$r_j = \frac{C_j}{\sigma_j \sigma_{j-1}} \tag{70}$$

where, C_j -- the covariance between f^j and f^{j-1},

σ_j^2 -- the variance of f^j,

σ_{j-1}^2 -- the variance of f^{j-1}

Suppose that,

M^j -- the mean value of f^j,

M^{j-1} -- the mean value of f^{j-1},

thus,

$$C_j = E\{(f^j - M^j)(f^{j-1} - M^{j-1})\} \tag{71}$$

$$\sigma_j^2 = E\{(f^j - M^j)^2\} \tag{72}$$

Therefore, the r_j can be represented as

$$r_j = \frac{E\{(f^j - M^j)(f^{j-1} - M^{j-1})\}}{(E\{(f^j - M^j)^2\}E\{(f^{j-1} - M^{j-1})^2\})} \tag{73}$$

From the Schwarz's inequality, we have

$$E^2\{(f^j - M^j)(f^{j-1} - M^{j-1})\} \leq E\{(f^j - M^j)^2\}E\{(f^{j-1} - M^{j-1})^2\}$$

and therefore,

$$r \leq 1 \tag{74}$$

The equality holds when f^j equals a constant times f^{j-1}.

Intuitively, if the changes occurring in the consecutive frame are small, the correlation between successive frames is high. The standard deviation of the difference image formed from the successive frames will be small. Therefore, the correlation coefficient can be estimated from the standard deviation of the difference image. This can be proved by the following derivation. Suppose that,

d^j -- the difference between f^j and f^{j-1},

ΔM^j -- the difference of M^j and M^{j-1},

σ_{dj} -- the variance of the difference image, d^j,

C_{dj} -- the covariance between f^j and d^j,

r_{dj} -- the correlation coefficient of f^j and d^j.

Thus, we have

$$\begin{aligned} C_j &= E\{(f^j - M^j)(f^{j-1} - M^{j-1})\} = E\{(f^j - M^j)(f^j - d^j - M^j + \Delta M^j)\} \\ &= E\{(f^j - M^j)^2\} - E\{(f^j - M^j)(d^j - \Delta M^j)\} \\ &= \sigma_j^2 - C_{dj} \end{aligned} \tag{75}$$

The variance of (j-1)-th frame can be represented as

$$\sigma_{j-1}^2 = E\{(f^{j-1}-M^{j-1})^2\} = E\{(f^j - d^j - M^j + \Delta M^j)^2\}$$
$$= \sigma_j^2 - 2C_{dj} + \sigma_{dj}^2$$

From the definition (71), $C_{dj} = r_{dj}\sigma_j\sigma_{dj}$, therefore from (70), the correlation coefficient can be rewritten as:

$$r_j = \frac{\sigma_j^2 - C_{dj}}{\sigma_j\,(\sigma_j^2 - 2C_{dj} + \sigma_{dj}^2)}$$
$$= \frac{\sigma_j^2 - r_{dj}\sigma_j\sigma_{dj}}{\sigma_j\,(\sigma_j^2 - 2r_{dj}\sigma_j\sigma_{dj} + \sigma_{dj}^2)} \qquad (77)$$

From (77), we find the partial differential of r_j about σ_{dj},

$$\frac{\partial r_j}{\partial \sigma_{dj}} = \frac{(r_{dj}^2 - 1)\,\sigma_j\,\sigma_{dj}}{\sigma_{j-1}(\sigma_j^2 - 2r_{dj}\sigma_j\sigma_{dj} + \sigma_{dj}^2)}$$
$$= \frac{(r_{dj}^2 - 1)\sigma_j\sigma_{dj}}{\sigma_{j-1}^3} \qquad (78)$$

Since r_j is smaller than 1, the partial differential is zero or negative. Thus, we prove that r_j will monotonicly decrease with each increment of σ_{dj}. The standard deviation of the difference images can be used to describe the correlation coefficient, and therefore, to estimate the degree of changes occurring in the sequence of images.

1. Simple changes: Small change or changes occurring in the successive frames such as constant translation have a small standard deviation of the difference image, therefore from (74), r is less but very close to one.

2. Complex changes: Complex changes occur between the successive frames, and d^j is no longer a constant, and the standard deviation of the difference image becomes large. From (78), we find that the correlation coefficient decreases with each increment of the standard deviation of the difference image.

Thus, the standard deviation of the difference image can be used to measure the correlation between successive frames, and therefore, to measure the changes occurring between the successive frames.

B. Picturephone sequence

The picturephone sequence of images in section 5.3 is also used in the experiments for testing FAMM algorithm. For the first 9 frames, there is little motion. At frame 10, a rapid change occurs as the woman turns her head. At frame 11, the changes become complex from the point of view of image processing, since large scene changes are involved. The correlation coefficients of the successive frames and the standard deviation of the difference images are shown in Fig.32 These results are consistent with above theoretical analysis. For the first 9 frames, only little motion exists, the correlation coefficients are close to one and the standard deviations are small. After frame 9, a rapid change occurs and correlation coefficient decreases and the standard deviation increases. In the last four frames, the changes have slowed down and the

correlation coefficients increase again and the standard deviations decrease as shown in Fig.32.

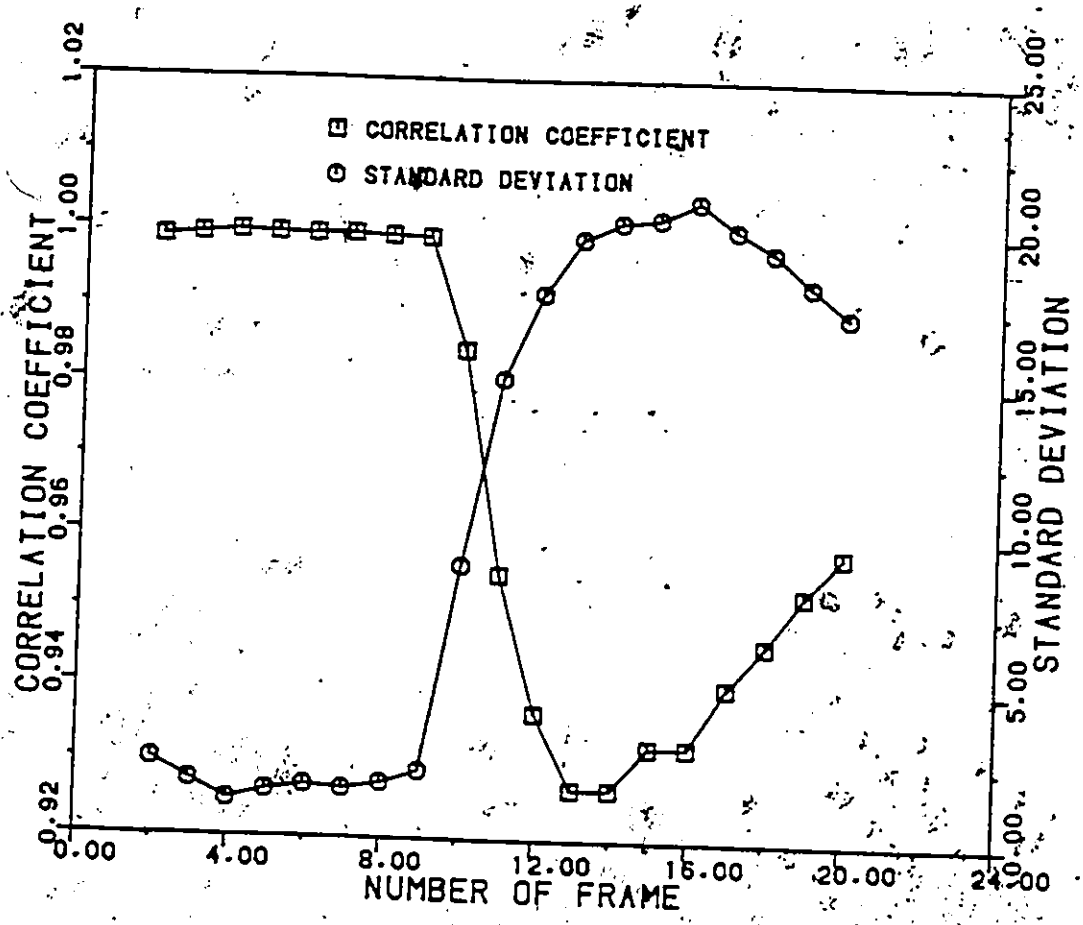


Fig.32 The curves of the correlation coefficient between successive frames and the standard deviation of the difference images as a function of the number of frames. Note that the large correlation coefficient corresponds to the small standard deviation value of the difference images; that is the correlation decreases with increase of the standard deviation.

6.4 EXPERIMENTAL RESULTS

In this section, the results of computer simulations on the sequence of angiographic x-ray images and the picturephone sequence using the FAMM algorithm are presented. The number of iterations in FAMM are controlled by, respectively, the first-order statistics of the difference image histogram and the variance of the difference image.

6.4.1 Application to angiographic x-ray images

In the last section, three types of changes occurring in angiographic sequence of x-ray images and their statistical parameters are presented. The first three moments extracted from the first-order histogram can be taken as a measure of the type of changes and used to initiate and control the adaptive operation. In this section, the convergence of the FAMM codebook replenishment as applied to these medical image sequences is investigated. The experiments are carried out on a frame drawn from the three types of sequences and the results are shown in Fig.33 (a), (b) and (c). In the experiments, we suppose that the convergence is achieved when the distortion between two connective iterations is decreased by less than 1%. For the noise sequence IS-A, in which only small changes are involved, after only label reassignment (replenishment) the iterative process is convergent (Fig.33a). For the translation sequence IS-B, convergence is achieved after one itera-

tion, i.e. label reassignment followed by mean migration (Fig.33b). For the occlusion sequence, IS-C, which involves significant changes, convergence is achieved within two iterations (Fig.33c). Hence, it is clear that for the particular medical image sequence, the number of iterations in the FAMM algorithm can be controled directly by the first-order statistics of the difference images as indicated in section 6.3.1.

The visual results are presented in Fig.34 to Fig 36, respectively, for the sequence IS-A, IS-B and IS-C. It is seen that the good reproduction is achieved by FAMM codebook replenishment.

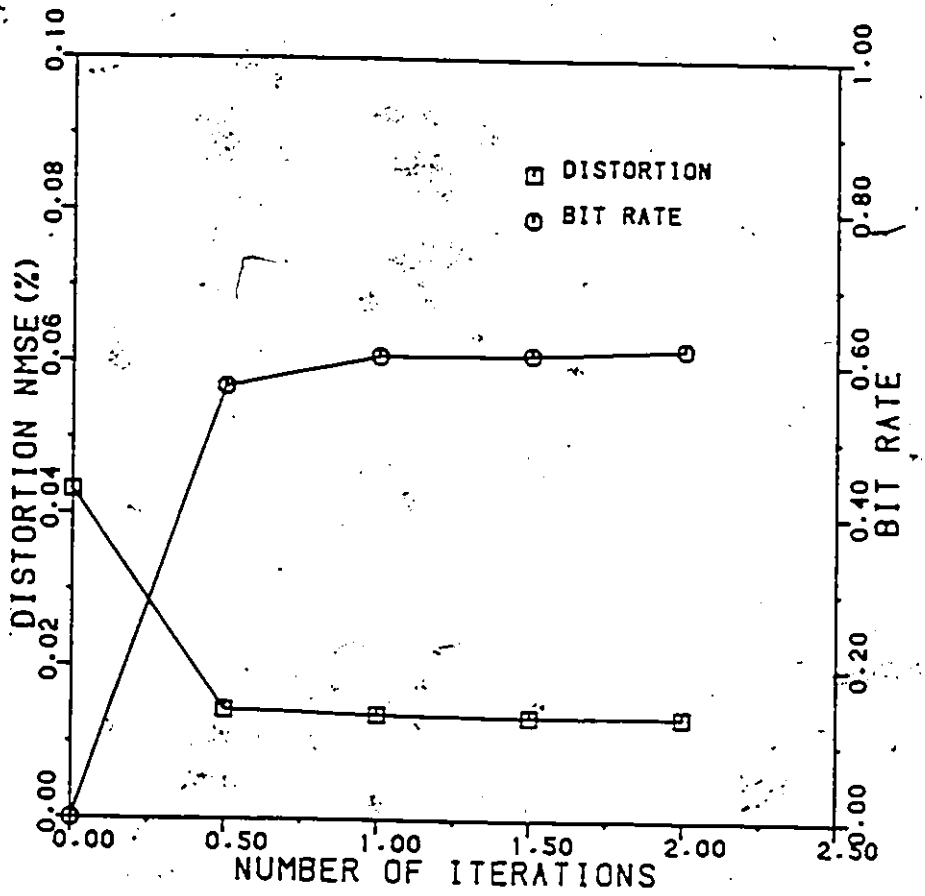


Fig.33(a) The distortion (NMSE) and the bit rates as the functions of the number of iterations for a typical frame (frame 3) in IS-A. Here, 0.5 iteration implies only the label replenishment step. Note that the convergence is achieved after one label replenishment for the noise sequence.

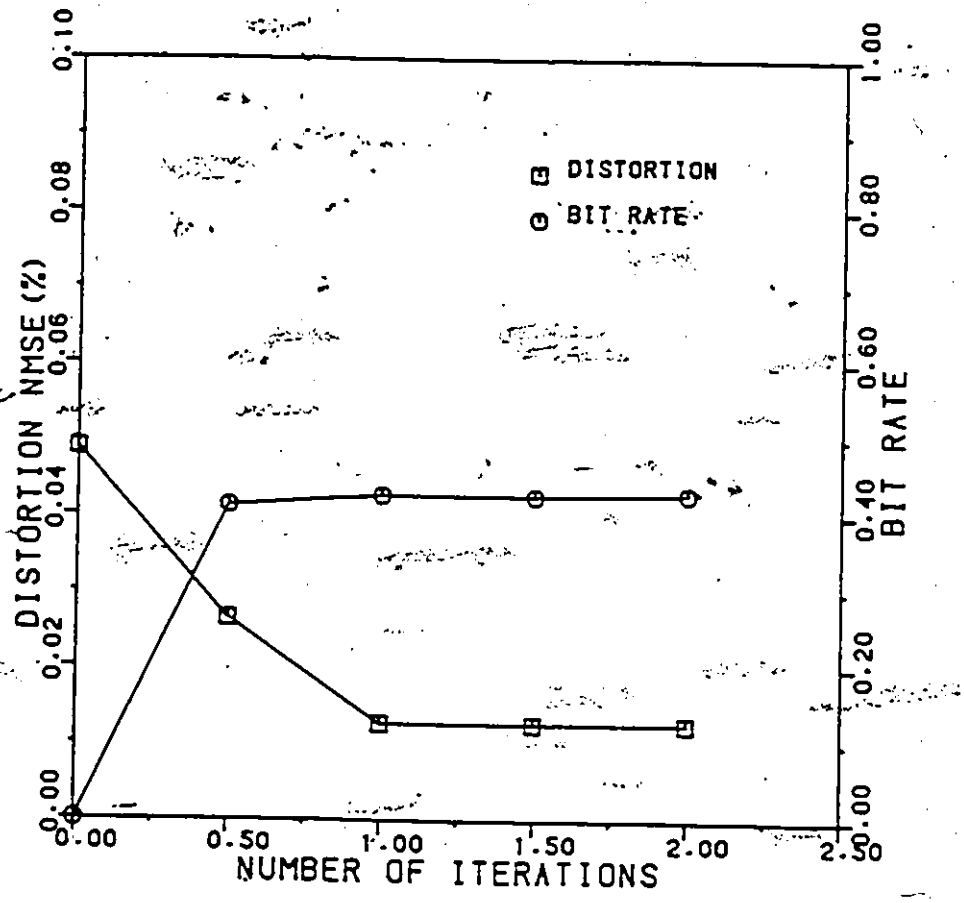


Fig. 33(b) The distortion (NMSE) and the bit rates as the functions of the number of iterations for frame 3 in TS-B. Note that the convergence is achieved after one iteration for the translation sequence.

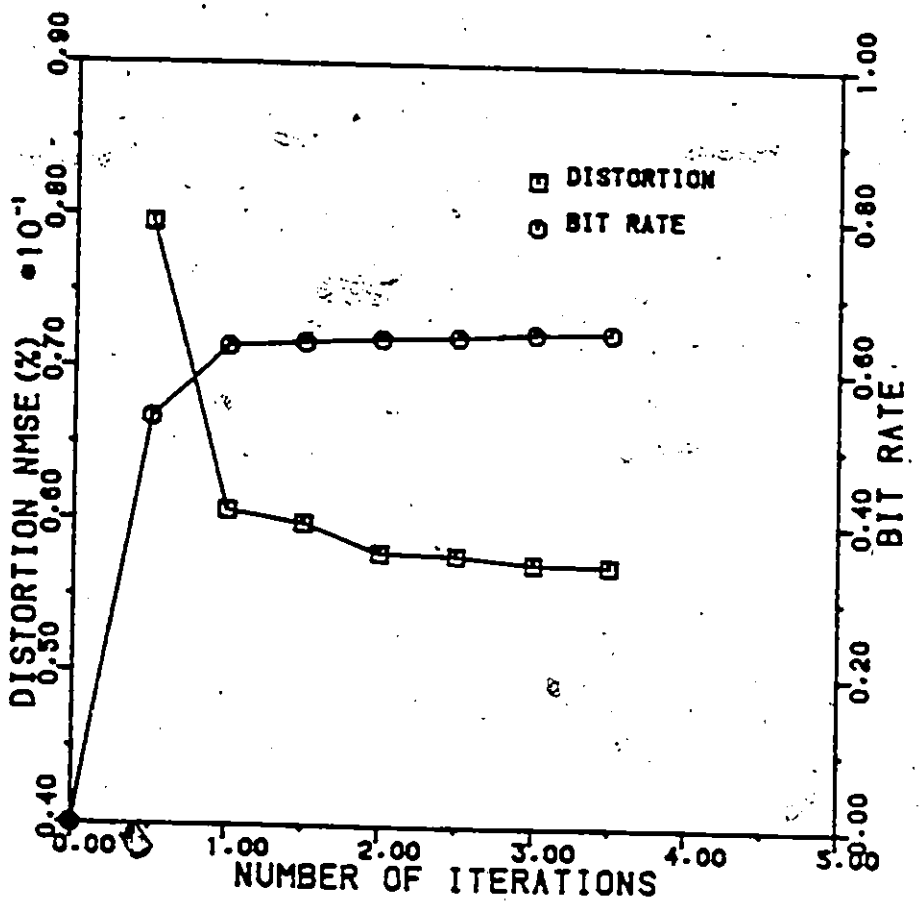
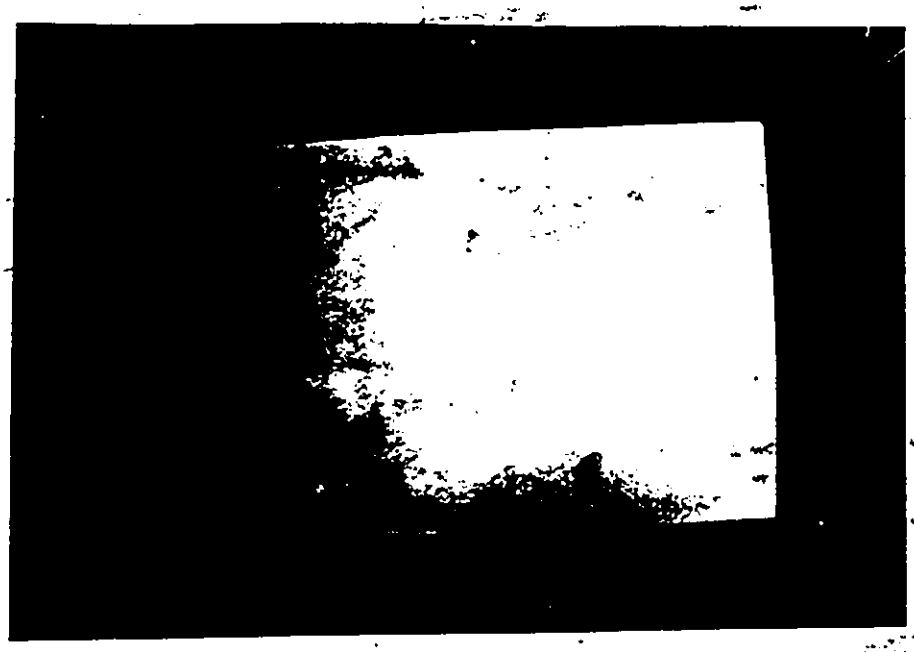
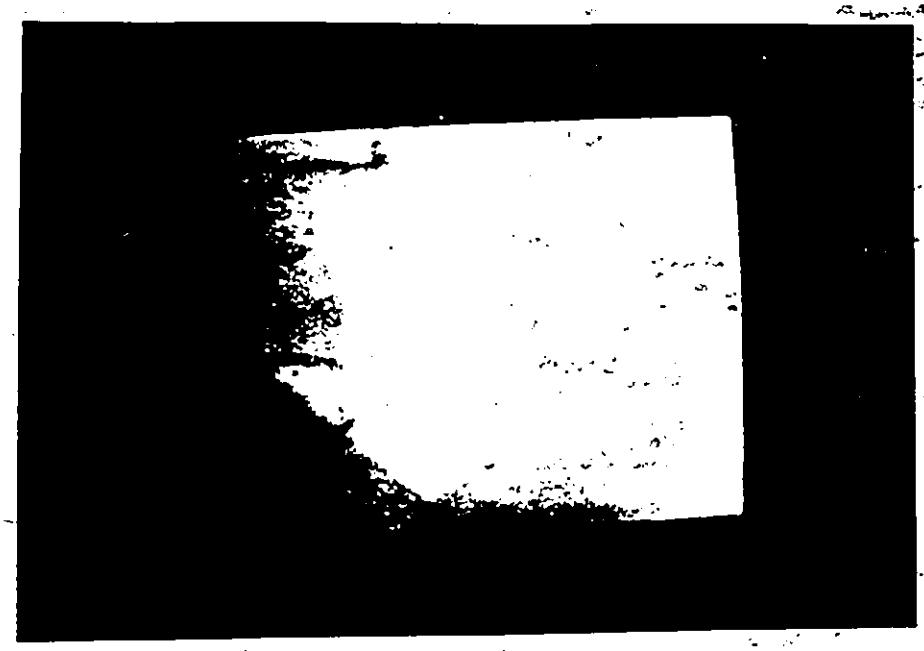


Fig.33(c) The distortion (NMSE) and the bit rates as the functions of the number of iterations for frame 3 in IS-C. Note that the convergence is achieved after two iterations for the occlusion sequence.

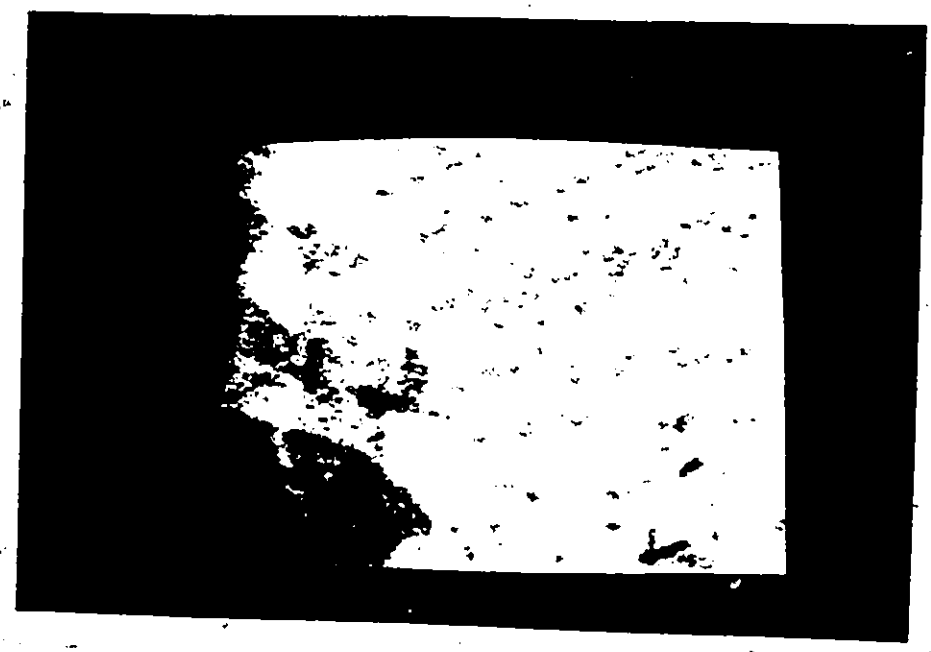


(a) original,



(b) coded

Fig.34 The pictorial result for the second frame in the sequence IS-A. The coded image corresponds to the FAMM algorithm with only label replenishment at bit rate 0.56 bits/pixel.



(a) original,

(b) coded

Fig.35 The pictorial result for the second frame in the sequence IS-B. The coded image corresponds to the FAMM algorithm with one iteration at bit rate 0.71 bits/pixel.

(a) original,

(b) coded

Fig.36 The pictorial result for the second frame in the sequence IS-C. The coded image corresponds to the FAMM algorithm with two iterations at bit rate 0.67 bits/pixel.

6.4.2 Application to the picturephone sequence

In the last section, the relationship between the correlation coefficient and the standard deviation was derived. In this section, the convergence of the FAMM codebook replenishment related to the standard deviation is investigated. The experiments are carried out on the four different frames in the picturephone sequence and the results are shown in Fig.37.

For the simple changes, such as at frame 2, the standard deviation is small (2.5), and the iterative process is convergent after a simple label replenishment. For the more complex changes exhibited in the frames 10, 11 and 14, the standard deviations are, respectively, 8.99, 17.97 and 20.47; the iterations required for convergence are, respectively, 1, 2 and 3.

The distortion curves as a function of the frame number and the bit rate curves as a function of the frame number for FAMM codebook replenishment are shown in Fig.38a and Fig38b, respectively. For comparison, the results for label replenishment and label replenishment with mean shift are also shown in Fig.38a and Fig.38b. Note that FAMM achieves better results than label replenishment with mean shift with almost the same bit rate at expense of computation.

The visual results are shown in Fig.39. In Fig.39 is shown the coded image of 18-th frame using respectively, label replenishment with 256 codewords; label replenishment with 256

codewords combined with mean shift; and label replenishment combined with FAMM with same size codebook. Note that the better reproduction is achieved with FAMM algorithm coded images.

6.5 SUMMARY

In this chapter, the Frame Adaptive Migrating Mean (FAMM) codebook replenishment has been proposed. This algorithm can be efficiently employed to update the codebooks on a frame by frame basis. In the application of the FAMM codebook replenishment algorithm to medical image sequence coding, it has been shown that parameters extracted from first-order statistics of the difference images can be successfully applied to control the number of iterations in the algorithm. For the general case, the correlation coefficient of the successive frames and the variance of the difference images can be taken as a measure to control the number of iterations in the FAMM algorithm.

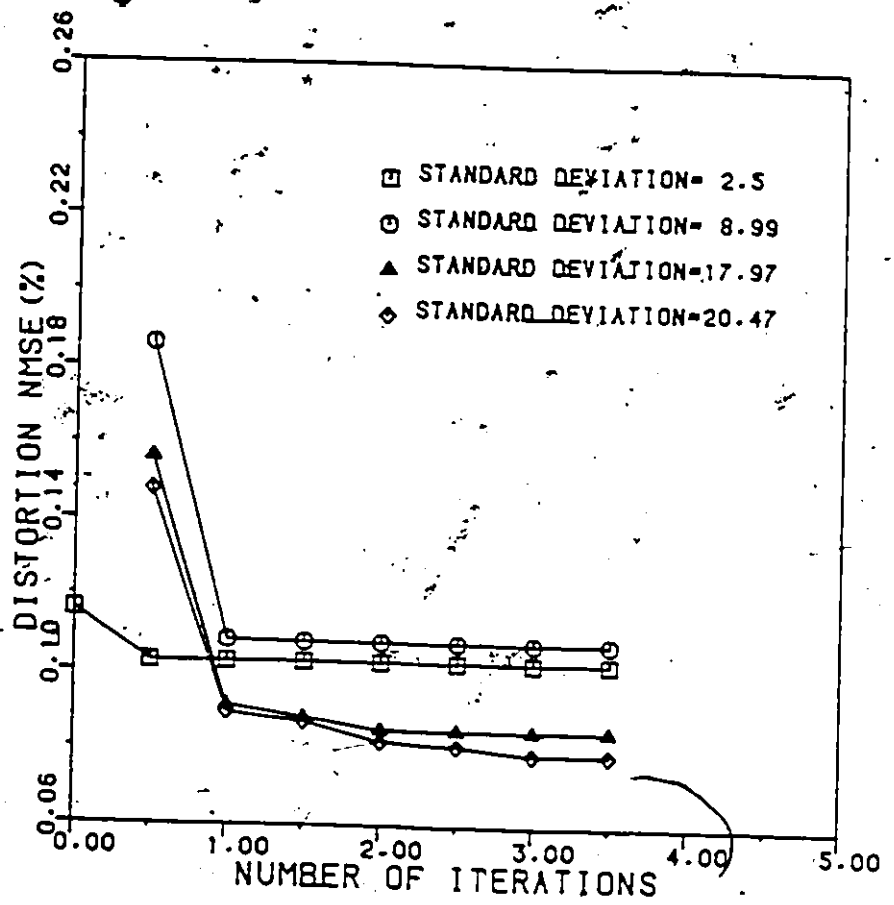


Fig.37 The curves of distortion as a function of the number of iterations in FAMM for the frame 2, 10, 11 and 14, which have different standard deviation values of the difference images. Note that the large standard deviation values need more iterations to achieve the convergence, and at most after three iterations, the convergence can be achieved even for the frames where complex changes are involved.

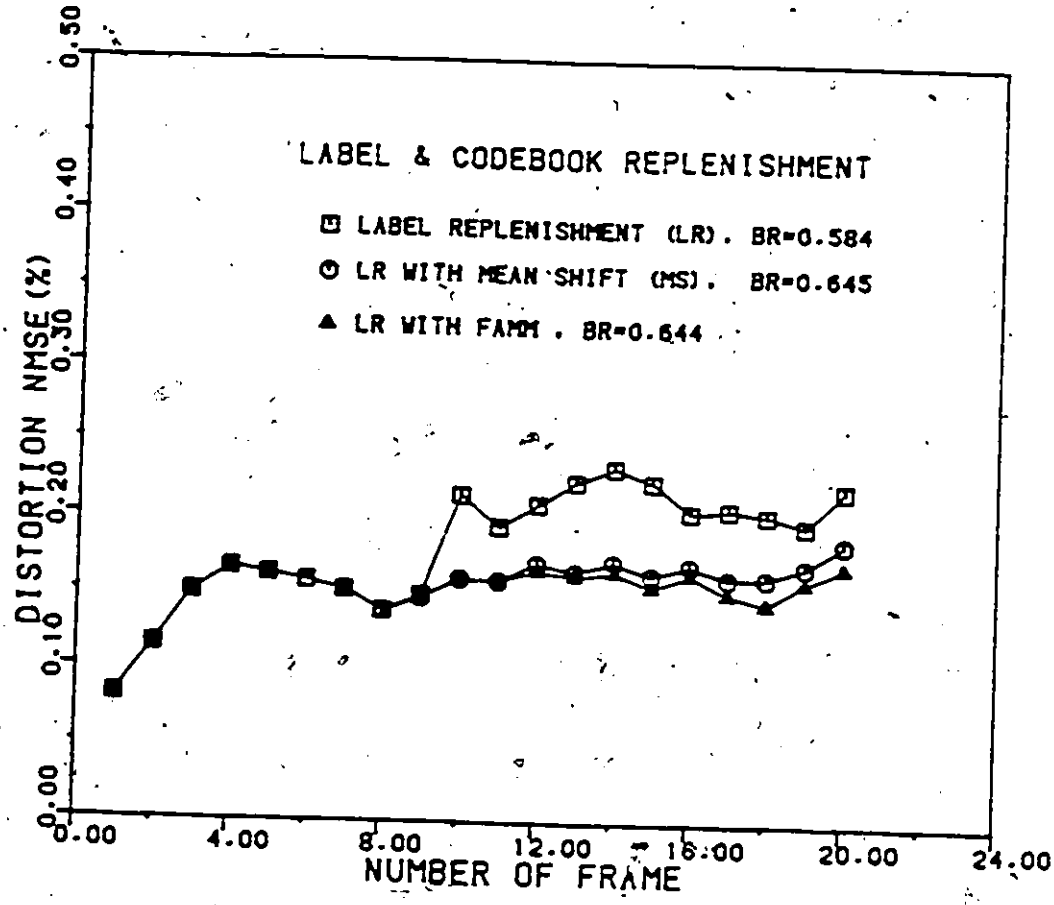


Fig.38(a) The distortion (NMSE) as a function of the frame number for label replenishment with mean shift and FAMM codebook replenishment. Note that label replenishment combined with FAMM codebook replenishment maintains a relatively constant distortion, even in the rapid change areas.

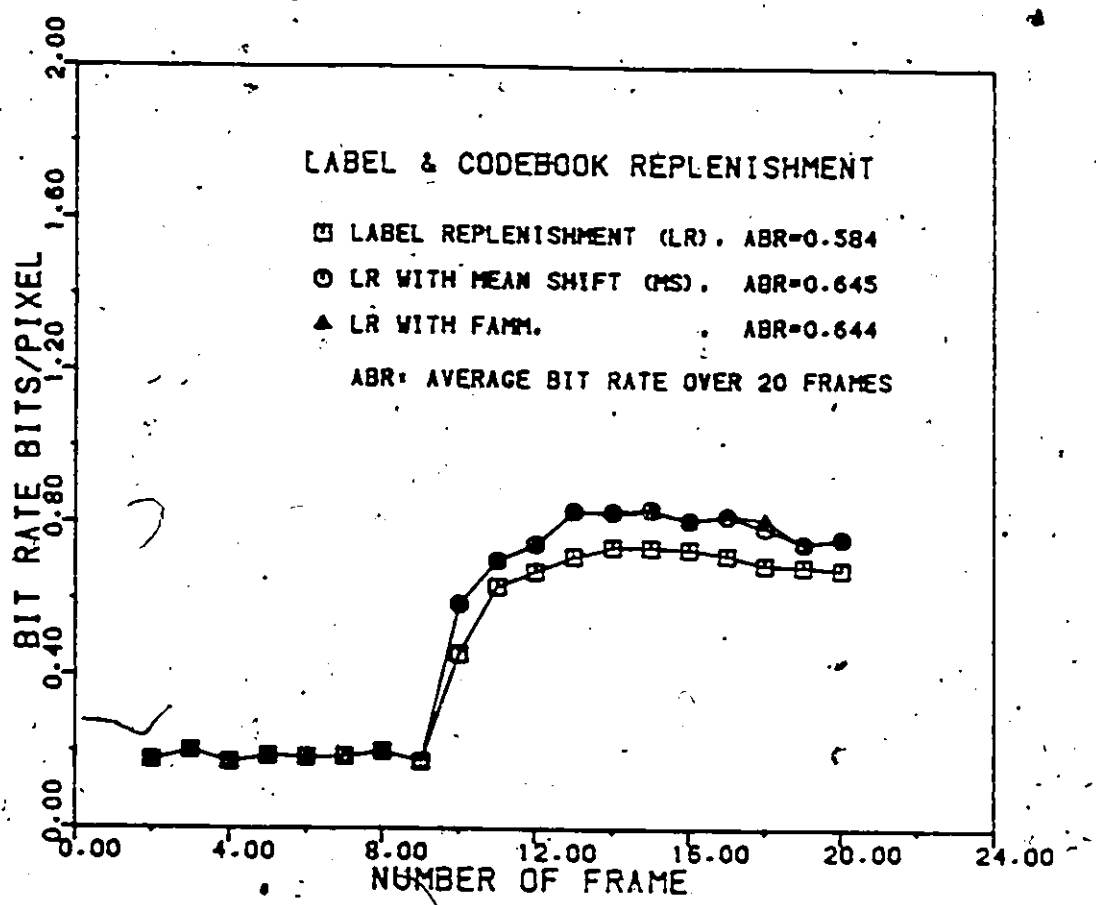


Fig.38(b) The bit rate as a function of the frame number for the label replenishment with mean shift and FAMM codebook replenishment. Note that codebook replenishment allocates more bits to frames containing changes, thus resulting in the more constant distortion shown in Fig.38a. Note that FAMM achieves better results than mean shift with almost same bit rate but at expense of computation.

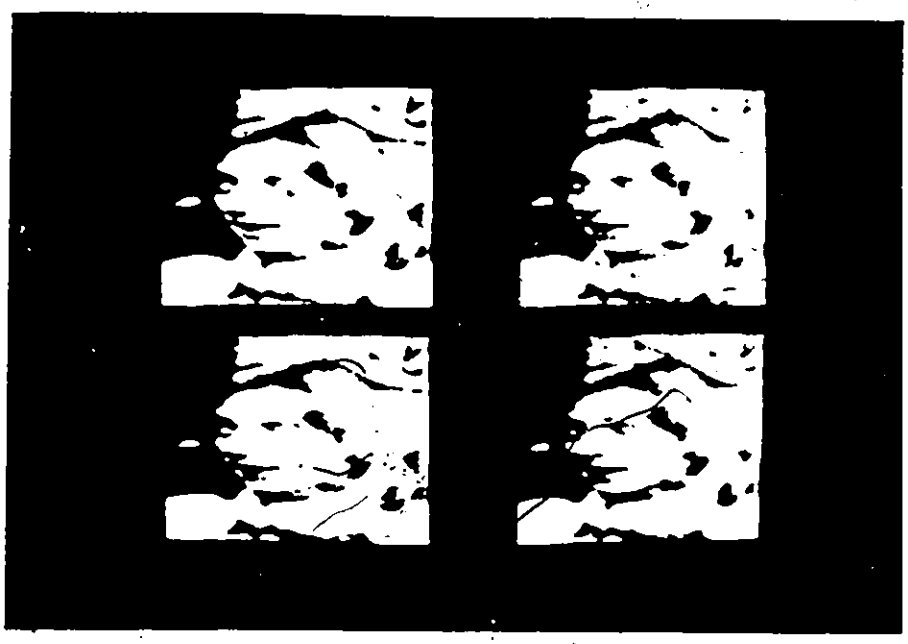


Fig.39 The pictorial results for the 18-th frame of the picturephone sequence. The coded images correspond to, respectively, label replenishment with 256 codewords (top right), and label replenishment with 256 codewords combined with mean shift (bottom left), and label replenishment with 256 codewords combined with FAMM (bottom right). The improvements are noticeable for the FAMM.

Chapter VII

SUMMARY AND SUGGESTIONS FOR FURTHER RESEARCH

In this dissertation, interframe coding using adaptive vector quantization has been investigated. The main contributions of this dissertation are as follows:

1. Three-dimensional block vector quantization: two algorithms are given for this technique. The first is an extension of two-dimensional block vector quantization to three-dimensional (spatial-temporal) blocks. The second is an algorithm in which the three-dimensional Hadamard transform is first applied on a 4x4x4 block basis and vector quantization is then performed on a selective subset of the transformed coefficients. In this algorithm, the components in the vectors are decorrelated in both spatial and temporal directions. Thus, both intraframe and interframe redundancies are reduced by, respectively, transform and vector quantization. Improved performance as compared to spatial-temporal vector quantization is obtained at the expense of longer training sets and increased complexity of transform operation. The results demonstrate that in all cases vector quantization outperforms scalar quantized Hadamard transform coding. For bit rates between

0.4 to 0.8 bits/pixel, the vector quantization gives at least 75% improvement in NMSE over scalar quantized Hadamard transform coding.

2. Label replenishment algorithm: this is a simple and efficient adaptive strategy. This algorithm is based upon two-dimensional block vector quantization, at the frame level, onto which is grafted the concept of frame replenishment and vector quantization. In vector quantization, each input vector is encoded by two distinct information: the label and the corresponding codeword. In label replenishment, the label is taken as an important information which is replenished according to the change between frames. The results have shown that this algorithm is especially efficient for the image sequence in which the change is characterized by translation.

3. Label replenishment with codebook replenishment: in the label replenishment case, the underlying assumption made is that an acceptable new codeword in the existing codebook can always be found for each input vector. Thus, only new label needs to be transmitted. However, if complex changes are involved in the image sequence, this assumption may no longer be satisfied. The closest codewords in the existing codebook yield too large error for some input vectors so that new codewords must be determined. In other words, the codebook must be up-

dated to track the changes of the local statistics. This is implemented by two codebook replenishment algorithms: mean shift and mean shift with selective codeword replacement. In the mean shift algorithm, the codebook is replenished by simply shifting the individual codewords to the positions of new mean vectors of the Voronoi regions. This algorithm is applicable if the frame to frame changes are small. The selective codeword replacement algorithm is applied in the case when significant changes of local statistics occur and simply shifting the mean is no longer sufficient. In this algorithm, the codebook is replenished by dropping certain codewords which are less important and adding new ones. The results show that significantly improved performance is obtained by two-dimensional vector quantization ~~with~~ label replenishment and codebook replenishment compared to direct vector quantization on the three-dimensional block vector quantization.

4. The Frame Adaptive Migrating Mean (FAMM) codebook replenishment is a generalization of mean shift codebook replenishment. The main idea behind this algorithm is that a new codebook is generated by using the previous codebook as seeds for the clustering algorithm used for codebook generation, where the number of iterations depends on the type of changes occurring in the input image sequence. First-order statistical parameters ex-

tracted from difference images calculated from successive frames and correlation coefficients between the successive frames are used to control the adaptive operation, that is the number of iterations. For the image sequences tested, the similarity between successive frames assures that convergence is achieved after, at most, three iterations.

An important observation is the significant improvement of two-dimensional vector quantization with replenishment as compared to using vector quantization on the three-dimensional blocks. For bit rates between about 0.5 and 0.65, the NMSE is reduced by a factor at least 2. It also demonstrates that two-dimensional vector quantization with replenishment outperforms direct frame replenishment [1].

5. An additional contribution of this thesis is adaptive LPC coding combined with vector quantization for intraframe coding. An image is first partitioned into blocks. The LPC parameters are extracted from each block using the autocorrelation estimation method and then used to form a vector. The vector quantization is then applied to the vector set to remove the redundancy.

Some further areas of research include:

1. In the applications for medical image sequences, the changes between successive frames can be caused by the contrast dye material. This kind of changes is poorly reproduced by the current vector quantization coding techniques. To solve this problem, we could use the concept of a partitioned codebook, previously used by Gersho and Ramamurthi [11], in which a portion of the codebook is specially assigned to edge information. In the application here, we divide the codebook into two parts with one part preferentially reserved for the changes.

2. In the codebook replenishment algorithms proposed in this thesis, replenishment is achieved by comparing the previous codebook and present codebook. This implies that the present codebook must be first calculated from the present frame and require one or more iterations of the clustering algorithm. To decrease the number of iterations, the concept of predictive codebook replenishment could be investigated. The idea behind this is that the new codebook can be predicted from the previous codebooks and the changes occurring between the successive frames. In this algorithm, the relationship between the changes of the codebooks and the changes in the consecutive frames probably needs to be investigated.

3. In this thesis, we exploit codebook replenishment techniques to track the changes occurring in the input image sequences. Some studies [11,12,20] try to create a universal codebook to fit different input image sources. Investigations that analyze and compare these two techniques and investigate their strengthes and weakness could be valuable.

REFERENCES

1. Pratt, W. K., (Ed) Image Transmission Techniques, Academic Press, N.Y., 1979.

2. Jain, A.K., "Image Data Compression: A Review", Proceeding of IEEE, Vol.69, No.3, pp.349-389, March, 1981.

3. Mounts, W. F., "A Video Encoding System using Conditional Picture Element Replenishment", B.S.T.J. 48, No.7, pp.2545-2554, September 1969.

4. Netravali, A.N., and Robins, J.D., "Motion-compensated television coding: Part I", B.S.T.J., Vol. 58, No.3, pp.631-670, March 1979.

5. Gray, R.M., "Vector quantization", IEEE ASSP Mag., pp.4-29, April 1984.

6. Tou, J., and Gonzalez, R.C., Pattern Recognition Principles, Addison-Wesley Publication Company, 1974.

7. Linde, Y., Buzo, A., and Gray, R.M., "An Algorithm for Vector Quantizer Design", IEEE Trans. on Commun., Vol. Com-28, No.1, pp.84-89, Jan. 1980.

8. Buzo, A., Gray, A.H., Gray, R.M., and Markel, J.D., "Speech coding based upon vector quantization", IEEE Trans. on ASSP, Vol. ASSP-28, No.5, pp.562-574, Oct. 1980.

9. Juang, B.H., Wong, D.Y., and Gray, A.H., "Distortion performance of vector quantization for LPC voice coding", IEEE Trans. on ASSP, Vol.ASSP-30, No.2, pp.294-303, April 1982.

10. Rebolledo, G., Gray, R.M., and Bury, J.P., "A Multirate voice digitizer based upon vector quantization", IEEE Trans. on Comm., Vol. Com-30, No.4, April 1982.

11. Gersho, A., and Ramamurthi, B., "Image coding using vector quantization", IEEE Proceeding ICASSP, pp.428-431, May 1982.

12. Baker, R.L., and Gray, R.M., "Image compression using non-adaptive spatial vector quantization", IEEE Conference on Circuits System-Computer, pp.55-61, 1983.

13. Boucher, P., and Goldberg, M., "Transform image coding by vector quantization" Neuvieme Colloque sur le traitement du signal et ses applications, Gretsi Nice, pp.629-632, May 1983.

14. Hilbert, E.E., "Joint pattern recognition/data compression concept for ERTS multispectral data", SPIE, Vol.66, "Efficient transmission of pictorial information", August 1975.

15. Lowitz, G.E., "Image data reduction", Proc. International Conf. on Spacecraft On-Board Data Management, Nice, pp.291-294, Oct. 1978.

16. Hilbert, E.E., "Cluster compression algorithm a joint clustering/data compression concept for ERTS multispectral data," Jet Propulsion Lab. Publication, 73-43, 143 pages.

17. King, R.A., and Nasrabadi, N.M., "Image coding using vector quantization in the transform domain" Pattern Recognition Letters, Vol.1 pp.323-329, July 1983.

18. Sun, H.F., and Goldberg, M., "Image coding using LPC with vector quantization" IEEE proceeding of International Conference on Digital Signal Processing, pp.508-512, Florence, Italy, Sept. 1984.

19. Sun, H.F., and Goldberg, M., "Image sequence coding using vector quantization" IEEE Proceeding of International Communication and Energy Conference, pp.266-269, Montreal Canada, Oct. 1984.

20. Murakami, T., Asai, K., and Yamazaki, E., "Vector quantizer of video signals", Electronic Letters 7, pp.1005-1006, Nov. 1982.

21. Boucher, P., and Goldberg, M., "Color image compression by adaptive vector quantization" IEEE Proceeding ICASSP, pp.29.6.1-4, San Diego, March 1984.

22. Nasrabadi, N.M., and King, A.K., "A new image coding technique using transforms vector quantization", IEEE Proceeding ICASSP, pp.29.9.1-29.9.4, 1984.

23. Goldberg, M. and Sun, H.F., "Image sequence coding using vector quantization," accepted by IEEE Trans. on Communication.

24. Gersho, A., and Yano, M., "Adaptive vector quantization by progressive codevector replacement", IEEE Proceeding ICASSP, pp.133-136, Florida, March 1985.

25. Berger, T., Rate Distortion Theory, Prentice-Hall, Inc. 1971.

26. Davisson, L.D., "Rate distortion theory and application", Proceeding of The IEEE, Vol.60, No.7, pp.800-808, July, 1972.

27. Netravali, A.N. and Limb, J.O., "Picture Coding: A Review", Proceeding of The IEEE, Vol.68, No.3, pp.366-407, March 1980.

28. Dubois, E., Prasada, B. and Sabri, M.S., "Image sequence codng", Image Sequence Analysis, Edtor: Huang, T.S., Chapter 3, 1982.

29. Kretz, F., "Subjectively optimal quantization of picture", IEEE Trans. on Commun. Vol.Com-23, pp.1288-1292, Nov. 1975.

30. Huffman, D.A., "A method for the construction of minimum-redundancy coder" Proc. IRE, 40, 9, pp.1098-1101, sept. 1952.

31. Roese, J. A., Pratt, W.K., and Robinson, G.S., "Interframe cosine transform image coding" IEEE Trans. on Commun. vol. COM-25, pp.1329-1339, Nov. 1977.

32. Andrews, H.C. and Pratt, W.K., "Fourier transform coding of images" Proceeding Hawaii Int. Conf. System Science, pp.677-679, Jan. 1968.

33. Pratt, W.R., Kane, J., and Andrews, A.C., "Hadamard Transform image coding" Proceeding of the IEEE, Vol.75, No.1, pp.58-68, Jan 1969.

34. Jain, J.R., and Jain, A.K., "Interframe adaptive data compression techniques for image" Sig. & Image Proc. Lab., Elect. and Comput. Eng. Dept. Univ. CA, Davis, Aug. 1979.

35. Tasto, M., and Wintz, P.A., "Image coding by adaptive block quantization", IEEE Trans. on Commun. Technol., Vol. Com-19, pp.957-972, Dec. 1971.

36. Habibi, A., "Hybrid Coding of Pictorial Data" IEEE Trans. on Comm., Vol. Com-22, No.5, pp.614-624, May 1979.

37. Mitchell, O.R. and Delp, E.J., "Multilevel Graphic Representation Using Block Truncation Coding", Proceedings of the IEEE, Vol.68, No.7, pp.868-873, July 1980.

38. Kurtman, C.M., "Redundancy reduction-A practical method of data compression", Proceedings of the IEEE, Vol.55, pp.253-263, Mar. 1963.

39. Netravali, A.N., "Interpolative coding using a subjective criterion," IEEE Trans. on Commun., Vol.Com-25, No.5, pp.503-508, May 1977.

40. Jain, J.R., and Jain, A.K., "Displacement measurement and its applications in interframe coding," IEEE Trans. on Commun., Vol. Com-29, pp.1799-1806, Dec. 1981.

41. Koga, T., Iinuma, K., Hirano, A., Fijima, Y., and Ishiguta, T., "Motioncompensated interframe image coding for video conferencing," in NTC 81, Proc., Ing, pp. G5.3.1-G5.3.5, New Orleans, LA, Dec. 1981.

42. Ninomiya, Y., and Ohtsuka, Y., "A motion-compensated interframe coding scheme for television pictures,", IEEE Trans. on Commun., Vol.Com-30, pp.201-211, Jan. 1982.

43. Goldberg, M., Boucher, P., and Shlien, S., "Adaptive Image vector quantization", IEEE Trans. on Communication, Volume Com-34, pp.180-187, February 1986.

44. Huang, T.S., and Tsai, R.Y., "Image Sequence Analysis: motion estimation", Image Sequence Analysis, Huang, T.S., (Editor), pp.1-17, 1980.

45. Habibi, A., "Survey of adaptive image coding techniques", IEEE Trans. on Commun., Vol. Com-25, No.11, pp.1275-1284, Nov. 1977.

46. Gersho, A., "On the structure of vector quantizer", IEEE Trans. on Information Theory, IT-28, pp.157-166, March 1982.

47. Healy, D.J., and Mitchell, O.R., "Digital video bandwidth compression using block truncation coding", IEEE Trans. on Commun., Vol. Com-29, No.12, pp.1809-1817, Dec. 1981.

48. Limb, J.O., and Bowen, E.G., "An interframe coder for broadcast television," in preparation.

49. Candy, J.C., Franke, M.A., Haskell, B.G., and Mounts, F.W., "Transmitting television as clusters of frame-to-frame differences," B.S.T.J., 50, No.6, pp.1889-1917, 1971.

50. Pease, R.F.W., and Limb, J.O., "exchange of spatial and temporal resolution in television coding," B.S.T.J. Vol.50, pp.191-200, Jan. 1971.

51. Haskell, B.G., Mounts, F.W., and Candy, J.C., "Interframe Coding of vediotelephone picture," Proceeding of The IEEE, Vol.60, pp.792-800, July 1972.

52. Chen, W.H., and Smith, C.H., "Adaptive coding of monochrome and color images," IEEE Trans. on Commun., Vol.Com-25, No11, pp.1285-1292, Nov. 1977.

53. Roese, J.A., Pratt, W.K., and Robinson, G.S., "Interframe cosine transform image coding," IEEE Trans. on Commun., Vol.Com-25, No.11, pp.1329-1339, Nov. 1977.

54. Yamado, Y., Fujita, K., and Tazaki, S., "Vector quantization of vedio signals," Proceeding of Annual Conference of IECE, pp.1031, 1980.

55. Baker, R.L., and Gray, R.M., "Vector quantization of digital images," submitted for publication, July 1984.

56. Murakami, T., Asai, K., and Yamazaki, E., "Interframe vector coding of color video signals," Proc. Int. Picture Coding Symp., July 1984.

57. Juang, B.H., and Gray, A.H., "Multiple stage vector quantization for speech coding," Proceeding of The IEEE International Conference on ASSP pp.597-600, Apr. 1982.

58. Nasrabadi, N.M., "Use of vector quantizers in image coding," IEEE Proceeding ICASSP 85, pp.125-128, March 1984.

59. Boucher, P., "Adaptive vector quantization of pictorial data," thesis for M.A., Dept. of Elect. Eng. Univ of Ottawa, 1984.

60. Maragas, P.A., Mersereau, R.M., and Schafer, R.W. "Some experiments in ADPCM coding of image," IEEE ICASSP, pp.1227-1230, May 1982.

61. Pratt, W.R., Kane, J., and Andrews, A.C., "Hadamard Transform image coding" Proceeding of the IEEE, Vol.75, No.1, pp.58-68, Jan 1969.

62. Conner, D.J., Brainard, R.C., and Limb, J.O., "Intraframe coding for picture transmission", Proc. IEEE 60, pp.780-791, 1972.

63. Netravali, A.N., Prasada, B., and Mounts, W., "Some Experiments in Adaptive Hadamard Transform Coding of Pictures" B.S.T.J., Vol.56, No.8, pp.1531-1547, Oct. 1977.

64. Knauer, S.C., "Real-time video compressor algorithm for Hadamard transform processing", IEEE Trans. on Electromagnetic Compatibility, Vol. EMC-18, No.1, pp.28-36, Feb. 1976.

65. Habibi, A. and Wintz, P.A., "Image coding by linear transformation and block quantization," IEEE Trans. on Commun., Vol. Com-19, No.1, pp.50-62, February 1971.

66. Musmann, H.G., Pirsch, P., and Grallert, H.J., "Advances in picture coding," Proceeding of The IEEE, Vol.73, No.4, pp.523-546, April 1985.

67. MacCall, J.R., and Chang, M.V., "Multispectral data compression using hybrid cluster coding," American Institute of Aeronautics and Astronautics Communications Satellite System conference, pp.404-407, 1980.

68. Ramamurthi, B. and Gersho, A., "Image coding using segmented codebooks," Proceedings International Picture Symposium, March 1983.

69. Rosefeld, A., and Kak, A.C., Digital Picture Processing Computer Science and Applied Mathematics, second edition, Volume 1, Academic Press, 1982