INFORMATION TO USERS

While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. For example:

- Manuscript pages may have indistinct print. In such cases, the best available copy has been filmed.

- Manuscripts may not always be complete. In such cases, a note will indicate that it is not possible to obtain missing pages.

- Copyrighted material may have been removed from the manuscript. In such cases, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or as a 17"x 23" black and white photographic print.

Most photographs reproduce acceptably on positive microfilm or microfiche but lack the clarity on xerographic copies made from the microfilm. For an additional charge, 35mm slides of 6"x 9" black and white photographic prints are available for any photographs or illustrations that cannot be reproduced satisfactorily by xerography.

Colef, Michael

MOTION COMPENSATION IN THE HADAMARD DOMAIN FOR VIDEO CONFERENCING

City University of New York PH.D. 1987

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✓.

1. Glossy photographs or pages ✓
2. Colored illustrations, paper or print ✓
3. Photographs with dark background ✓
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) _____ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Dissertation contains pages with print at a slant, filmed as received ✓
16. Other _____

_____

_____

MOTION COMPENSATION IN THE HADAMARD DOMAIN

FOR

VIDEO CONFERENCING

by

MICHAEL COLEF

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1987

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

4-29-87 — Date

Hugh Barta — Chair of Examining Committee

4/29/87 — Date

Donald R. Schilling — Executive Officer

Prof. Norman Scheinberg

Prof. Donald L. Schilling

Prof. Michael Conner

Prof. Erlan H. Feria

Supervisory Committee

The City University of New York

Abstract

MOTION COMPENSATION IN THE HADAMARD DOMAIN

FOR

VIDEO CONFERENCING

by

Michael Colef

Adviser: Professor Joseph Barba

The research presented here introduces an original technique to separate the NTSC composite signal into its components in the Hadamard transform domain, when sampled at four times the frequency of the color subcarrier and a phase angle of $Kx12^{o}$, K=1, 2,... .

A motion detection, estimation and compensation algorithm is presented which takes advantage of the separation of the NTSC composite signal into its components in the Hadamard transform domain. This algorithm involves setting proper thresholds for the coefficients in the Hadamard transform domain in order to detect the motion and estimate the displacement of the blocks classified as changed. The thresholding process involves only the transform coefficients which contain most of the energy pertaining to the Y, I and Q component signals.

The bit rates obtained in this study are 54Kbps - 318Kbps appropriate for the low bit rate video conferencing systems. The reconstructed video images were subjectively evaluated and found to be of good "video conferencing" quality.

AKNOWLEDGEMENTS

I wish to express my sincere gratitude to my mentor, Professor Joseph Barba for his expert counsel and continued encouragement that he afforded me throughout this research.

I also wish to thank the members of my Doctoral Committee: Professor Norman Scheinberg (Co-Mentor), Professor Donald L. Schilling, Professor Michael Conner and Professor Erlan H. Feria for the time and effort each has spent on the reading and criticism of this dissertation.

In addition, I wish to especially thank Professor Demos Eitzer for his continuous support and for being the first one to make me see the challange of the work towards a doctoral degree.

I also wish to thank to all my colleagues in room T502F for their encouragement and concern throughout this work.

I especially wish to thank my brother Gabriel for his help, counselling and support which helped to ease many difficult periods that arose during the course of this research.

Finally, I wish to thank my parents, Ivan and Jana-Ioana, for their love, encouragement, patience and support.

To My Family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

Fig.2.10: Position of picture elements in frame s_{k-1} used for interpolation of $s_{k-1}(x-\hat{dx}_i, y-\hat{dy}_i)$

Fig.2.11: Pel recursive displacement algorithms simplified for comparison by assuming $E[s^2_{k-1}(s,y) = = E[s^2_{k-1}(x-dx,y-dy)] = \text{constant}$

Fig.2.12: Stability ranges for various recursive estimation algorithms

Fig.2.13: 2D-logarithmic search procedure

Fig.2.14: Three-step search procedure

Fig.2.15: Conjugate direction search in simplified version

Fig.2.16: Choice of neighboring pels for prediction switching

Fig.2.17: Comparison of bit rates for different types of predictors (a,b,c,d,e,f); effect of displacement estimation on chrominance bit rate (g,h); comparison of chrominance bit rates for various methods of switching the chrominance prediction (i,j).

Fig.2.18: Map of pels surrounding the present pel in $4f_{sc}$ sampled NTSC signal

1. INTRODUCTION

A HISTORY OF VISUAL INFORMATION

The first attempts to portray real or imagined scenes as perceived by the human vision were dated from long before man knew how to use articulated speech. As mankind evoluated, the means of expressing visual perceptions became more sophisticated. It took man's understanding of scientific processes in order to go from cave paintings to the first pictures on paper, to motion pictures and finally to television. The tools used for storing and transmitting visual information became more complex.

Recently, visual information has been presented in forms which do not in themselves, stimulate the eye. To give an example, voltage variations can represent in analogous fashion the changes in brightness or color in a picture. These forms of visual information are a solution to the problem of storing, processing and transmitting of the images. Thousands of pictures could be stored on films, magnetic tapes and disks. Their transmission and processing is easier as well.

During the last couple of decades the advantages of digital representation of signals over analog representation completely changed the approach towards

storing, processing and transmission of visual information [2,5,71]. The first application of image processing techniques was in improving the digitized newspaper pictures sent by submarine cables between London and New York using the five level Bartlane cable picture transmission system, in 1920 [1]. The pictures where coded for cable picture transmission and then reconstructed at the receiver by specialized printing equipment. In 1929 the capability of the system was increased to fifteen levels [1]. Since 1929 to the present digital techniques have played a crucial role in the development of practical coding algorithms for image transmission.

A disadvantage of digital representation of video signals is the tremendous increase in the bandwidth required to transmit these images (64 MHz). Consequently, it is not surprising that a tremendous effort has gone into the development of bandwidth compression algorithms. All these algorithms have to make certain assumptions about the psychophysical nature of the video signals and take advantage of certain imperfections of the human vision depending on the nature of the application.

Early bandwidth compression algorithms were used for "half-duplex" transmission to achieve near studio quality of images over the full range of possible scenes. This resulted in a reduction of the transmission rate from 64

Mbps to 32, 34 or 44 Mbps which required dedicated transmission channels.

Over the past decade video conferencing systems have developed which attempt to transmit full duplex a restricteded range of scenes typically found in a conference room setting. Restricting the types of scenes that are to be transmitted allowed a further reduction in the bandwidth used for transmission of these scenes. Transmission rates used were 1.5 Mbps in the United States and Japan and 2 Mbps in Europe. Recently transmission rates as low as 19 Kbps, 56Kbps, 112 Kbps, 224 Kbps, 384 Kbps or 768 Kbps have been attempted.

During the last five years, technological development in fiber optics have led to the development of wide band services using fiber optic links capable to accomodate transmission rates as high as 140 Mbps. Video conferencing systems using such type of transmission media are still unavailable for extended commercial use because of the restricted area using fiber optic links for commercial communication and of the cost of such systems.

2. PICTURE CODING

2.1 A REVIEW OF BASIC PICTURE CODING TECHNIQUES

There are three important assumptions one generally makes prior to storing, processing and transmitting of the visual information in a spatial/temporal sampled representation. Firstly, the amount of visual information will be reduced because we represent only a finite number of intensities. Secondly, the amount of visual information will be reduced because the rapid intensity variations are pre-filtered. Thirdly, the amount of visual information will be reduced because we could quantize the intensity values only to a finite accuracy [2].

In order to decide upon the coding technique and subsequent transmissicn rate to be employed one should know the amount of information contained in the class of images to be transmitted [4,8,13,37,62,63,71]. The amount of information has a bearing on the minimum bandwidth required for transmission of signals as shown in APPENDIX B.

There are four major coding techniques which have been used to code video images to achieve bandwidth compression. These coding techniques are: Pulse Code Modulation, Predictive Coding, Transform Coding and Interpolative/ Extrapolative Coding. Each of these coding techniques gives bandwidth compression [2,5,8,11]. A fifth class which

consists of different schemes that do not belong to any of the four major coding approaches, such as run-length coding, etc., can be added to these major coding techniques. A classification of the coding techniques is given in Fig.2.1.

2.1.1 PCM CODERS

In PCM, a time discrete, amplitude discrete representation of picture elements, i.e. pels, is made [2,5]. This representation does not reduce the statistical or perceptual redundancy existent in the signal [2]. It is a time discrete representation because the signal is sampled at the Nyquist rate [3,APPENDIX A]. It is an amplitude discrete representation because the signal is quantized to n bits/sample [2]. A diagram of a PCM system is shown in Fig.2.2. In order to transmit NTSC signal, using PCM, one should use 8 bit/sample, [2,5,6,71]. If less bits are used, image degradation in the form of contouring effect is manifested. Contouring occurs in low detail portions of the image where the intensity does not change very rapidly, and is due to the fact that changes from one quantization level to the next occur along fairly continuous lines, or contours, which are quite visible to the human eye, [5,6]. Using 8 bit/sample and sampling at

the Nyquist rate of 8 MHz, yields a bit rate of 64 Mb/s, [5,61,64,71], which is too high to transmit over standard links (satellite links accomodate 45 Mb/s and terestrial links accomodate an even lower bit rate). It is possible to use less than 8 bits/sample for PCM encoded images and to obtain reasonable picture quality. This is possible by adding high frequency noise to the original signal before quantizing [6,71]. This noise causes the coded signal to oscillate between quantizing levels thereby increasing the frequency content of the quantizing noise. This causes the quantizing contours not to be any longer visible. The technique is called dither and was successfully used by L. Roberts to produce good quality PCM coded images for less than 8 bit/sample [2,5,6,71]. The PCM technique does not perform well when less than 5 bits/sample are required.

2.1.2 PREDICTIVE CODERS

In order to further reduce the bandwidth, other coding techniques have to be employed such as predictive coding. Predictive coding systems do not transmit the quantized amplitudes of each pel as in PCM systems. An approximate prediction of the value of the pel to be transmitted, from previously coded information that has been transmitted, is made [2,5,11,71]. This prediction is then subtracted from the actual value of the pel. The difference, known as the error, is then quantized into a set of discrete amplitude

levels. These levels are then encoded using fixed or variable word-length codes and sent to the channel coder to be transmitted, [2,5,11,71]. Thus, we may say that a major objective of the predictive coding is to generate an error signal which is small on the average and large only occasionally [2], resulting in a low signal entropy. Therefore, the better the predictor, the smaller the error and consequently the lower the entropy and the lower the bit rate required to transmit the signal.

The predictive coder has three basic components, which are the predictor, the quantizer and the encoder, [2,5,71]. The number of levels of the quantizer will differentiate between delta modulation systems and differential pulse code modulation systems. If the quantizer has only two levels, the system is a delta modulation system, otherwise the system is a DPCM system, [2,5,71].

2.1.2-1 PREDICTORS (COMPOSITE)

Predictors can be classified as linear or nonlinear predictors depending upon the prediction function used. In a linear predictor the prediction function is a linear function of the previously transmitted pels. If the prediction is a nonlinear function of the previously

transmitted pels, then the predictor is a nonlinear one, [2,5,71].

We can make a further division of the predictors based upon the location of the pels used in the prediction. If the pels used for predicting the value of the pel to be transmitted are located in the same line as the pel to be transmitted, then the predictor is a one-dimensional one. If the predictor is using values of pels from previous lines then the predictor is a two-dimensional one. The interframe (3-D) predictors use values from the previously transmitted frames [2,5,71].

The predictors used can be fixed or adaptive. The adaptive ones change their characteristics as a function of the data. The fixed predictors maintain the same characteristics independent of the data.

DM systems are attractive due to their simplicity. The DM systems have found a great use in encoding speech and video signals [2,5,71]. The slope overload and edge business effects presented by DM systems have been greatly reduced by using an adaptive step size for the delta modulator. An exponentially changing step size type of delta modulator performs better than other types of delta modulators for encoding signals with large changes in amplitude as in video signals [66]. Different ADM algorithms have been described in the literature [9,10,65,66,67,68] to obtain bandwidth compression of video

signals. Bit rates of 2 bits/pixel, 1 bit/pixel and 0.5 bit/pixel have been reported.

Two-dimensional predictors are more often used in picture coding. A proper choice of the predictor coefficients will yield an improved prediction as well as a quick decay of the effects of the transmission errors in the reconstructed picture [2,5,11]. The two-dimensional predictors, also known as intrafield predictors, reduce the error on the vertical edges [5] but increase the error in the horizontal edges [2].

When two-dimensional predictors are used for color pictures, special problems are encountered due to the presence of the modulated chrominance components. Much of the correlation in the adjacent samples is lost due to this reason [2]. This effect of the subcarrier is overcome, to a large extend, by using previous samples of the same phase [2]. That means that if the sampling rate is n times the color subcarrier frequency, then the n^{th} previous sample is used as a prediction. This results in less correlation due to the distance between the samples than if the signal was a baseband monochrome signal. In order to match the bimodal spectrum of the NTSC signal more complex predictors have to be designed [2,5].

Since the picture signal is highly nonstationary it is advantageous to change the prediction based on the local

properties of the signal. Overhead information has to be transmitted to indicate the change to the receiver. For the intrafield situation, a commonly used method is to compute some measure of directional correlation based on the local neighboring pels which have already been transmitted and use this measure to select a predictor along the direction of maximum correlation. The predictor is usually linear and is chosen such that the prediction error is very small if the signal is correlated in a certain direction [2,5,14]. For example, the Graham predictor uses the previous line (pel B in Fig.2.3) or the previous pel (pel A). The switching between them is done by the following rule:

$$X = \begin{cases} A & \text{, if } |B-C| < |A-C| \\ B & \text{, otherwise} \end{cases} \qquad (2.1.1)$$

- predictor for element X

Another variation of the adaptive prediction is to use a weighted sum of several predictors. The weights are switched from pel to pel and are chosen by observing certain characteristics of the neighbouring pels. It turns out that such adaptive predictive coding techniques do not improve the mean square prediction error neither do they significantly lower the entropy of the signals [2,5].

Interframe coders employ predictors which make use of

a combination of elements from the present as well as previous frames. The most succesful adaptive interframe predictors are those that take into consideration the motion of the different objects. If an estimate of the translation undergone by an object is known predictive coding becomes more efficient if the differences of the appropiate elements in the present and previous frames are considered for coding. These are known as motion compensated predictors. The success of a motion compensated prediction scheme depends upon the amount of motion of objects in real television scenes and the ability of an algorithm to estimate motion with the accuracy that is desirable for good prediction. One set of techniques recursively adjust the translational estimate at every pel or at every small block of pels [16]. Another set of techniques obtain an estimate of translation in a block of pels [15]. A decrease of 20 to 70 percent in the total coder bit rate can be achieved due to motion compensation. Frame difference predictors seem to be the best when the coding of a low detail and small motion scene is desired. The field difference predictors perform better than the frame difference predictors if the scene to be encoded presents higher detail and motion [2,5,7]. For even higher motion the element or line difference of frame or field difference predictors are desirable to be employed because

of their superior performance [5,7].

One of the problems encountered by DPCM systems are the propagations of transmission errors in the reconstructed pictures. Because of the way the prediction of the next sample is generated transmission errors in previous samples could affect the present sample and therefore the future ones as well. How much these errors will affect the picture depends on the type of predictor used. It is known that in the case of optimum one-dimensional linear predictors, for most types of correlations generally found in the pictures, the effect of transmission errors decays [17]. The previous element predictor, if it is not provided with a "leak", will propagate the transmission errors, and will therefore generate horizontal streaks in the reconstructed pictures. For two-dimensional predictors the situation is somewhat difficult, but there are some two-dimensional predictors for which the pattern of distortion produce by transmission errors is less annoying than for the one dimensional predictors. There are a multitude of methods of error correction or concealment. Some will replace the erroneous line by the previous line or by an average of the surrounding lines, others will provide a "leak", but the most reliable method seems to be the use of error detecting codes [5].

2.1.2-2 QUANTIZERS

DPCM systems owe their bandwidth compression characteristic, to a large extend,. to a coarser quantization of the predicted error signal than the original signal itself. An extensive amount of work has been done for quantizer optimization for the previous element DPCM coding, in which approximate horizontal slope of the input signal is quantized. Extensions to the case of the two-dimensional and interframe predictors have been made, but they did not receive enough attention [5]. There are three types of degradation that appear in images due to improper design of the quantizer of a DPCM coder. These are granular noise, edge busyness and slope overload and they are shown in Fig.2.4 [2,5,71]. The design of quantizers can be based.on statistical or psychovisual measures. The quantizers can be adaptive or fixed.

The nonadaptive optimum quantizers that are statistically based have been derived using the work of Max [72]. If the characteristic of a quantizer is the one presented in Fig.2.5, where x is the input to the DPCM quantizer with probability density p(x), then quantizer parameters can be obtained to minimize the following measure of quantization error:

$$D = \sum_{i=1}^{N} \int_{X_i}^{X_{i+1}} f(x-Y_i)p(x)dx \qquad (2.1.2)$$

where $X_1<X_2<\ldots<X_{N+1}$ are the decision levels, $Y_1<Y_2<\ldots<Y_N$ represent the representative levels and $f(.)$ is a nonnegative function. All the inputs $X_i<x<X_{i+1}$ to the quantizer are assumed to be represented by Y_i. The necessary conditions for optimality with respect to X_i and Y_i for a fixed number of levels N are given by:

$$f(X_j-Y_{j-1}) = f(X_j-Y_j), \; j=2,\ldots,N \qquad (2.1.3)$$

and

$$\int_{X_j}^{X_{j+1}} \frac{df(x-Y_j)}{dx} \, p(x)dx = 0, \; j=1,\ldots N \qquad (2.1.4)$$

assuming that $f(.)$ is differentiable. In the case of the mean square error criterion, $f(z)=z^2$, the conditions for optimality with respect to X_i and Y_j reduce to

$$X_j = (Y_j-Y_{j-1})/2 \qquad (2.1.5)$$

and

$$Y_j = \frac{\int_{X_j}^{X_{j+1}} xp(x)dx}{\int_{X_j}^{X_{j+1}} p(x)dx} \qquad (2.1.6)$$

For subjective reasons, the quantizers should not be designed based on the mean square error criterion [5,18,71]. In the case of DPCM systems, which code the quantizer outputs with fixed-length binary words, the output bit rate depends on the logarithm of the number of levels of the quantizer. Therefore, minimization of an error measure for a fixed number of levels for such systems is relevant [5]. There is a considerable advantage to using variable length words to represent the outputs of the quantizer because the distribution of occurances of different quantizer levels is highly skewed [5]. In this case, it is more relevant to minimize a measure of distortion subject to the entropy of the quantizer output being less than a given number, because the average bit rate is lower bounded by the entropy of the quantizer output. Quantizers are optimum with respect to this criterion for random variables with Laplacian density. Even though such quantizers give adequate picture quality for small step size they present an increased complexity because of the large number of levels. It seems that for a

better picture quality, quantizers have to be designed based on psychovisual criteria. It is hard though to decide which criterion to use because of the complexity presented by the human vision. One way of designing a quantizer is to minimize the number of quantizer levels or the entropy of the quantizer output such that the quantization error is at the threshold of visibility. Spatial variations of intensity mask the visibility of small perturbations in the intensity. The value by which an edge can be perturbed such that the perturbation is just visible is called the threshold at an edge. The quantization error can be constrained to be below the amplitude threshold for all slopes of the signal if one knows the relationship between the edge threshold and the slope of a single edge [69]. Whether the number of levels is minimized or the entropy is minimized, the optimum quantizers satisfying the above constraint turn out to be the same [5,69]. Another method used to design quantizers, based on a psychovisual criterion, is to minimize a weighted mean square quantization error, where the weights are derived from subjective tests. If the probability density is replaced by a weigthed function, the above optimization would be similar to the mean square quantization error. The weighting function could be the product of the probability density and a function of the amplitude threshold [5,71],

or it could be derived by measuring visibility of the noise added to certain regions of the picture as determined by the edge contrast [18,69]. These weighting functions are picture dependent, but the variation for a certain class may not be significant. The optimum quantizers designed on the basis of the above weighting functions have been quite successful, even though the optimum choice of the weighting function is still unknown. If the edges have to be reproduced faithfully, the quantizers that minimize the weighted mean square error for a fixed number of levels perform better than the minimum mean square error quantizers because they are less companded [5,18,41].

The design of quantizers for intraframe coders, which use more sophisticated predictors had been done either by minimizing the mean square error, by trial and error or by using extensions of the psychovisual quantizers discussed above. For interframe coders a lack of suitable models for both the distribution of the prediction error and the visibility of quantizing error had lead to the design of quantizers based on trial and error [5].

Adapting the DPCM quantizer to the picture statistics and the required fidelity of reproduction in different regions of the picture would lead to both the perception of the quantization noise and the statistical properties of the differential signal being uniform and stationary. This desirable adaptation is an extremely difficult task since

the perception of noise and the statistics may not be sufficiently related to each other. Psychovisual criteria or purely statistical based approximations to this goal have been made. If the processing of the signal takes place in the frequency domain the signal can be split into two frequency bands in order to exploit the detail sensitivity of the eye. A lower rate sampling but a finer quantization, because of the high sensitivity of the eye to noise in the low frequency region, is desirable for the low frequency component of the signal while the signal in the high frequency band has to be sampled at a higher rate and quantized coarser, because the eye is less sensitive to noise in highly detailed areas. In the pel domain a measure of spatial detail is found and a relationship between the visibility of noise and the measure of spatial detail is experimentally obtained. This relationship helps to segment a picture into subpictures in such a way that the unit noise would be almost equally visible in the entire subpicture. Quantizers for each subpicture could be designed using the criteria described before for nonadaptive quantizers. This together with employing variable length word coding could reduce the bit rate by about 30 percent over the previous element coder [19]. Much of this reduction is due to the coarser quantization of different segments rather than to the variable length word

coding. Delayed coding is another way of adapting the quantizer. The operation of quantization, in the case of delayed coding, is that of following a path through the encoding tree determined by the quantizer steps. The path is chosen to minimize the weighted quantization error at several samples, including some subsequent samples. The transmission of the adaptation information is not required because the quantization steps are not varied from sample to sample [5]. One may conclude that, even though the optimum rules for adaptation are not known for the intraframe and interframe coding, due to intuition or trial and error based rules, lower bit rates than for the nonadaptive systems have been obtained.

2.1.2-3 CHANNEL CODER

The output of the quantizer has to be coded before it is transmitted to the receiver. The choice of a code bears on the final bit rate of the system. Intraframe and interframe predictive coders present a nonuniform frequency of occurance of the quantizer output levels. Therefore, variable length word coding is recommended. This type of coding yields an average bit rate which is very close to the lower bound given by the entropy of the quantizer output. Some procedures require exact knowledge of the probabilities of the quantized output while others work with approximate probabilities and maintain their

efficiency even when the probabilities deviate from their approximate value. For picture type statistics optimum codes based on an average of many pictures remain quite efficient compared to the optimum codes designed on the basis of knowledge of the statistics of the individual pictures because the optimum code is not too sensitive to changes in the probability distribution [2,5]. This type of codes yield an improvement of about 6 dB S/N ratio with no change in the average transmission bit rate over the fixed length code [5]. The use of variable length codes creates a problem at transmission because the output bit rate from the source coder changes according to the local picture content and the transmission channel accomodates a constant bit rate. This problem is overcome by using a buffer which reads in data at a nonuniform bit rate and reads out data at an uniform bit rate. Since buffers have a finite length buffer overflow and underflow problems could occur. Buffer overflow could be overcome either by using a channel which accomodates a bit rate higher than the entropy of the quantizer output or by designing codes which minimize the probability of the length of the code words exceeding a certain number of bits. Sometimes buffer overflow cannot be prevented. For these situations strategies have been designed such that when the buffer begins to fill the output bit rate of the coder is gradually reduced, by

either subsampling or quantizing coarser, etc.. The case of buffer underflow is solved by increasing the output bit rate of the coder if the buffer begins to empty. These techniques find an extensive use in interframe coders [2,5].

2.1.2-4 COMPONENT SIGNAL CODING

In the case of component color coding, some predictive coding techniques use the information available in the coded form for the luminance component to derive as much useful information about the coding of the chrominance components as possible. If we assume the situation depicted by Fig.2.6 where L_i and $\overline{L}_i$ represent the luminance pels and C_i and $\overline{C}_i$ represent the chrominance pels, then if L_{n-1} gives the least mean square prediction error from all the predictors computed for the luminance pel L_n then C_{n-1} is considered to be the best predictor for C_n. Using this approach the entropy of the prediction error of the chrominance signals is deacreased by about 15 to 20 percent [2]. Another approach to coding color pictures is to separate the luminance and the chrominance signals, sample them at different sampling rates and use fixed predictors [55,70]. These predictors could be the previous element [70] or a weighted sum of previous elements in the same line and/or the previous lines [55,70]. Even though the predictors for chrominance signals are quite similar, they

are quite different from the predictor for the luminance signal. Some of these predictors minimize the entropy of the prediction error while others minimize the forth power of the prediction error. The error power for the chrominance coders is reduced by a factor of about 5 in going from a previous element predictor to the optimized third order predictor [70]. The quantizers used with these predictors are of the types described previously. Adaptive prediction techniques as well as multidimensional quantization are still in the begining stage and need to be further investigated in order to draw a valid conclusion as of their performance when used to encode color signals.

2.1.3 TRANSFORM CODERS

The major objective of transform coding is to produce statistically independent, or at least uncorrelated, transform coefficients which could be independently coded of each other and to obtain energy compaction such that a small number of coefficients need to be transmitted in order for the receiver to reconstruct the picture [2,4,5,8,20,40]. The performance of a transform coder depends on the transformation used, on the quantization strategy employed and on the subpicture size and shape [2,20]. Generally, linear unitary orthogonal transforms are employed in order to preserve the signal information in the

transform domain [1,2,4,5,8,12]. Similar to predictive coding, transform coding can be classiffied as one-dimensional, two-dimensional or three-dimensional based upon the spatial and temporal locations of the samples used in the transform [2,20]. One can also classify transform coding as nonadaptive if the parameters of the coder do not change to match the statistics of the subpicture to be coded or as adaptive if the parameters ochange to match the statistics [2,5,20]. The optimum transform is the transform that yields statistically independent transform coefficients [2,5]. This implies the knowledge of higher order (>2) statistics of the image, which one generally does not have. The higher order statistics of an image can be computed but is computationally intensive and only apply to a specific image. This is a major drawback of the optimum transforms since real time systems would have to use an extensive amount of hardware to implement such transforms [2]. Therefore, one seeks a transform which results in uncorrelated coefficients. This transform is the Karhunen-Loeve transform (KLT) [APPENDIX D]. Even though the KLT is known for its efficiency [2,4,5,35], its practical implementations presents problems [2,4,5]. Firstly, the covariance function of an image is not stationary, which means that either an average of the different covariance matrices has to be employed or different covariance matrices have to be employed for

different regions of the pictures. Secondly, in many cases the covariance matrix turns out to be singular, which means that some eigenvectors cannot be uniquely defined. Thirdly, the implementation of the transform is cumbersome because it requires N^2 multiplications by constants which sometimes are not simple powers of 2 for easy binary arithmetic [2,4,5]. All the other transforms used are suboptimum [5]. They produce less correlated coefficients but are easier to implement. Some of these transforms are the discrete Fourier transform (DFT), the discrete cosine transform (DCT) and the Hadamard transform (WHT) [APPENDIX D]. Even though these transforms are not optimum, they posses good energy compaction properties. These is due to the fact that the bases vectors they represent not only correspond to some most likely subpicture but also to some very unlikely subpictures. The coefficients obtained using these transforms are not as uncorrelated as the coefficients obtained by using the KLT but are more uncorrelated than the pels.

2.1.3-1 NONADAPTIVE TRANSFORM CODING

If the picture to be processed is assumed to be stationary with separable exponential correlation in both the horizontal and vertical direction, the mean square

error for encoding at a fixed bit rate varies with respect to the transform used and the block size. This variation is shown in Fig.2.7 [5]. The mean square error produce by transform coding for a given bit rate improves with the size of the subpicture. The improvement is not significant if the size of the subpicture is increased beyond 16x16. Subpicture of size 8x8 are commonly used, but from the point of view of the subjective picture quality no significant improvement seems to be presented by transforms using subpictures of sizes larger than 4x4. This is a conclusion based on computer simulations [5,20]. It turns out that the performances (comparing bit rates for a given mse) of the KL, Fourier and Hadamard transforms are about the same for supictures of size 4x4, but, for subpictures of sizes 8x8 and 16x16, KLT is better than Fourier transform which is better than Hadamard transform [5]. It has also been found that the two dimensional transforms do better than the one dimensional ones [2,5]. Three dimensional transforms have been used and their performance was better than the two dimensional transforms [5,22,45]. The performances were judged by comparing the bit rate for a given mean square error.

2.1.3-2 ADPTIVE TRANSFORM CODING

Adaptive transform coders match their parameters to the picture statistics. Because the statistics of the

pictures are highly nonstationary, this action could result in an increased coding efficiency [2,5,20]. There are basicaly two types of adaptations that transform coders could employ. In the first type changes in parameters are based on previously transmitted data. In the second type future data is used to compute the parameter changes. It seems that use of simple unitary transforms followed by adaptive selection of the coefficients for transmission and adaptive quantization provide most of the advantages of an adaptive transform coder [5,20].

One of the most popular transform coders in this class employs the threshold sampling technique. The threshold sampling coder selects a threshold and transmits all the coefficients above the threshold. By doing so, the threshold sampling coder adapts the number and the type of coefficients to the statistics of the subpicture. This technique requires a large amount of overhead to be transmitted to signal the changes due to adaptation [5,20]. Other transform coders take advantage of the lower sensitivity of the eye to amplitude changes in regions of high spatial detail by adapting the coefficient selection to a measure of spatial activity in the subpicture. A weighted sum of absolute values of all the coefficients gives a measure of the activity in the subpicture and based on this measure more bits are assigned to the subpictures

which have a higher activity [5]. Another technique used is zonal subsampling. This technique calculates the variances of the transform coefficients on a set of averages pictures and eliminates the coefficients which have a variance lower than a certain value (threshold). The discarded coefficients are set to zero by the receiver. This leads to a loss of resolution and a visible block structure in the picture depending on the number of the coefficients dropped from the picture. Coding overhead, when using zonal subsampling, can be reduced by transmitting the coefficients in a pre-defined order.

Other adaptive transform coding techniques have been used in which the statistics of the pictures are found and then used to code the pictures. One technique uses the variance of the transform coefficients and the bits are assigned proportional to the logarithm of the calculated variance [5]. Another one computes the "ac" energy of the pels in each subpicture and based on this classifies the subpictures into four classes. The thresholds used for classification and the bit assignment for each coefficient are selected specifically for each class [5,25].

As far as the three dimensional transforms are concerned, spatial activity and/or temporal activity could be used for an adaptive selection and quantization of the coefficients. Techniques that attempt to exploit the decreased sensitivity of the eye to perceive spatial detail

accompanied by large temporal changes use coarse sampling and quantization of the spatial coefficients whenever large temporal changes occur and transmit the spatial coefficients with high fidelity when the temporal changes are small.

2.1.3-3 QUANTIZERS

An important part of transform coders is the quantizer. The quantization of certain transform coefficients has an important bearing on the final bit rate of the coder. Once coefficients have been selected for transmission a quantizer has to be designed for each of them. If the bits are assigned to the coefficients proportional to the logarithm of their variances it means that the average quantization error is made the same for each of the coefficients. This is equivalent to minimizing the mean square error for a given total number of bits for Gaussian variables [2,5]. Optimum quantizers for the transform coefficients have been designed mainly on statistical basis. A Rayleigh density is used to model the first coefficient, which is usually an average of the pels within a block, while all the other coefficients are modeled with Gaussian density assuming a given variance. According to the central limit theorem the histograms of

the coefficients have a Gaussian shape and this is due to the fact that each coefficient is a linear combination of the picture elements. Minimizing the mean square error will yield the optimum quantization if the coefficients are quantized independently. Quantizers have to be designed based on the optimization of the picture quality for a given bit rate in order to achieve a better subjective picture quality. Such quantizers are hard to design because no accurate measurement of the picture quality exists. Subjectively matched quantizers have been designed on a trail and error basis or by employing weighting functions, which give the visibility of a unit quantization noise as a function of the amplitude of the coefficient, to minimize the weighted quantization noise. These quantizers perform better than the quantizers which minimize the mean square quantization errors [5,24,71].

2.2.4 CHANNEL ERRORS

One advantage of the nonadaptive transform coders is that the errors do not spread beyond the block in which they occured. The type of transform and the coefficient in error determine the nature of the degradation. Since the sensitivity of the eye decreases with frequency, above a certain frequency, the errors in the higher frequency coefficients are less visible than the errors in the lower

frequency coefficients. Due to the smaller number of bits required to encode the high frequency coefficients, the probability of errors taking place in these coefficients is lower than the probability of errors taking place in the lower frequency coefficients, which require more bits for encoding. The transmission errors appear as blotches in the picture. Adaptive transform coders could spread the effect of channel errors if they use the information from one block to the other either for adaptation or for decoding. If the adaptive coder uses the adaptation within the subpicture and a fixed number of bits is used for encoding then the channel error effects would be limited to subpictures at the expense of the coding efficiency [5]. The transform coding technique has been found to perform better than predictive coding if low bit rates have to be achieved.

2.1.4 HYBRID CODERS

The hybrid coding technique combines the attractive features of both transform coding and DPCM. There are three types of hybrid coders which have been reported in the literature. The first one uses a one dimensional block along the scan line and DPCM in the vertical direction. The second one employs a small two dimensional block of

transform coefficients and DPCM using coefficients of the previous block in the horizontal direction for prediction. The third type of hybrid coder uses a two dimensional block and DPCM in the temporal direction [2,5,32,71]. The first type is attractive because it requires less computations than a two dimensional transformation and the hardware required is less complex. This system tends to uncorrelate the data in one spatial direction by using a one dimensional transform and in the other spatial direction by the use of the DPCM encoders. As far as performance is concerned, this type of hybrid coder presents little degradation at small to moderate levels of channel noise, but its performance is considerably poorer at high levels of channel error. The second type of hybrid coder is attractive because of the simplicity of the coder (it employs a small block size and therefore requires less computations than a two dimensional transformation, hence simpler harware to implement the system). Its performance is better than the performance of a two dimensional transform coder as far as picture quality is concerned, but it turns out that the hybrid coder of the second type is more sensitive to channel errors [5,32]. Even though different types of predictors have been tried [5], the sesitivity of these types of hybrid coders to channel errors remains a certain fact. The third type of hybrid coder performed as well as a three-dimensional transform

coder which uses three previous frames [5,23]. This type of coder does not seem to present ill effects at bit error rates of 10^{-4}. The predictor used in interframe hybrid coding can be adapted by estimating motion and using coefficients from displaced blocks in the previous frame [5,26,27]. Recently, a minimum mean square error approach to hybrid coding has been suggested [85]. This scheme leads into simple coders. In this paper it was shown that a simpler 2x2 hybrid coder yields a performance at least 1dB better than that of an optimum 4x4 KLT coder. A simplified version of the scheme proposed, [85], has been offered, [86].

2.1.5 INTERPOLATIVE CODERS

Coders using interpolative coding techniques operate by sending a subset of pels to the receiver which interpolates to obtain the untransmitted pels [2,5,28]. Generally, the interpolative coding techniques can be classified as fixed or adaptive. Fixed interpolative coding techniques use a fixed set of pixel to be transmitted and interpolates to obtain the pixels which were not transmitted [2,5]. Different systems use different samples for transmission, i.e. alternate samples could be transmitted or one out of four samples is transmitted or

every alternate scan line or every alternate field or frame could be elected to be transmitted. Most use weighted sums of straight lines or higher degree polynomials. Between these two methods, the one using weighted sums of straight lines seems to be more effective [5]. Based upon the type of local correlation measured for the transmitted pixels the switched interpolation coder switches among horizontal, vertical, temporal or other directional averages [5]. Switched interpolation seems to be more effective than fixed interpolation.

Adaptive interpolation coders have to chose certain pixels for transmission, construct the interpolation of the nontransmitted pixels, evaluate the interpolation error and then chose less pixels for transmission if the error is less than a certain threshold or more pixels for transmission if the error is higher than the threshold [2,5]. Mean square error or subjective visual criteria are used to decide what interpolation scheme to employ. Adaptive interpolation coders have to transmit the addresses of the pixels chosen for transmission and the intensity values of these pixels [2,5]. The performance of the interpolative coders depends on the interpolation technique used and on the technique used to transmit the samples. Interpolative coders do better than nonadaptive DPCM or nonadaptive transform coders but they perform poorer than the adaptive DPCM and transform coders. These

comparisons take into account the number of bits necessary to obtain a given picture quality. These techniques are used with good results for interframe systems in conjunction with predictive coding [2,5].

Besides these four major coding techniques there are other schemes that do not belong to any of these techniques and are working only for certain types of video images. Run length coding which works very well for two-level facsimile is an example of such techniques. Contour coding, in which the picture is divided in contours and the rest of the picture and then by reproducing the edges better a crisp picture is obtained, is another example. Variations of contour coding are successfully used in facsimile systems.

The transmission of visual information at long distances using terestrial links or satellite communications is an ordinary event nowadays. What is not so ordinary are the techniques used to maintain a high quality of transmitted images using less information than that contained in the original images. Hence, the name of the game is to use less bits for the transmission of the information embeded in the original images and obtain a good subjective quality of the reconstructed video images. The transmission of TV signals using digital methods when the sampling rate greater than 8Mhz and 8 bits per sample are used will lead to a 64Mbps data rate, which is very

high. Fortunately a large amount of redundancy is enbeded in the video data. Reducing the redundant data we reduce the amount of data which is to be transmitted. Depending on the application we could also reduce the quality of the video images which will lead to an even higher data reduction. Motion detection and conditional replenishment systems make use of all of the above discussed bandwidth compression techniques in order to achieve the low bit rates required by their specific applications. These represent the topic of next chapter.

2.2 VIDEO CONFERENCING

In the past decade new applications of video communication systems have seen the development of video conferencing systems. Video conferencing systems provide means for full duplex communicating of live (moving) pictures of conference participants. This leads to an expansion of the shared visual and aural space. These systems are also capable of handling the transmission of text, graphics, photographic still images and sometimes some form of animation. From a historical point of view, the first video conferencing system was installed between two location of Bell Laboratories at Murray Hill and at Holmdel in New Jersey. This was followed by several experimental systems which include the CONFRAVISION system of British Telecom Research, the Australian Post Offices split screen system, Bell Canada's system, NTT's system and other telephone administration's systems. All these systems utilized analog TV transmission technology and standard TV cameras and monitors [64].

These systems were followed by video conferencing systems based on digital transmission of the information. In order to make these systems work efficiently bandwidth compression algorithms had to be employed. There is a wide range of bit rates that can be presently used for video

conferencing systems. The decision to use one bit rate or another depends on the transmission cost, transmission network availability, terminal cost, etc. 34-45 Mbps bit rates are usually used for short distance video conferencing and for intracity video conferencing. The complexity of the video codec is low. Lower bit rates, i.e. 1.5-2 Mbps, are desirable for international transmissions. If multipoint video conferencing is employed, then the bit rate per node are desirable to be lower than the 1.5-2 Mbps range, as low as 768, 384, 56 Kbps or lower (19 Kbps) [64]. In order to achieve such low bit rates the use of only the before mentioned coding techniques will not be sufficient. Therefore, the coding techniques mentioned in the previous chapter have to be used together with motion compensation algorithms in order to achieve very low bit rates.

The first question that arrises is why use motion detection, estimation and compensation? The answer to this question is that if one wants to reduce the amount of redundancy existent in images, in order to obtain lower bit rates for the transmission of these images, one has to transmit only those parts of the images that changed in time, which is equivalent to transmitting only the relevant information. Thus, motion compensation algorithms are important tools in the process of obtaining bandwidth reduction. If one can predict the motion of a certain

object in a scene based upon the way the object moved in the previous scenes, the amount of information that needs to be transmitted is even less [2,5,71].

So far, mainly motion models describing only the translational component of a motion have been studied with some success. The reason is the computational complexity presented by other forms of motion [2,12]. The result of a translational movement is a frame-to-frame displacement of the moving object. Once an estimate of the displacement vector of the moving object from two successive frames is generated, one could use it in motion-compensated predictive coding, motion-compensated transform coding and motion-adaptive frame interpolation [2,12]. Three types of displacement estimation algorithms are reported in the literature. These are the recursive displacement estimation algorithms, the displacement estimation by block matching algorithms and the feature-based displacement estimation algorithms. All the displacement estimation algorithms are based on the fact that in a sequence of images a moving object creates frame-to-frame luminance and chrominance changes which are used to create a mathematical models to describe the movement of the object. These models are used to improve the efficiency of the different interframe coding techniques. The most used model is the translational model because it does not require sophisticated computations which would be a major drawback in a real time

implementation of a displacement estimation algorithm. The first displacement estimation algorithms for TV coding were proposed by Limb and Murphy in 1975 [43] and by Cafforio and Rocca in 1976 [15].

Limb and Murphy used a moving edge to explain their algorithm. The displacement D is calculated using the following formula:

$$\hat{D} = \hat{dx} = \sum_{M} |FD| \; / \; \sum_{M} |ED| \qquad (2.2.1)$$

where:

M = area defined by frame differences greater than a given threshold

|FD| = magnitude of the frame difference signal

|ED| = magnitude of the element difference signal

The two summations are carried out over the entire area M see, Fig.2.8, [43]. The equation gives an estimate of the motion but does not indicate the direction of the displacement [12].

The algorithm developed by Cafforio and Rocca assumes that the object undergoing the translation does not change its luminance from frame to frame. If $z_k(x,y)$ denotes the luminance value at point (x,y) of a moving object in frame k, then a pure translational movement generates a displacement vector D with components dx,dy. The frame

differences are obtained by using the following formula:

$$\begin{aligned} FD(x,y) &= z_k(x,y) - z_{k-1}(x,y) \\ &= z_k(x,y) - z_k(x+dx,y+dy) \\ &= -\frac{\partial z_k(x,y)}{\partial x}\,dx - \frac{\partial z_k(x,y)}{\partial y}\,dy + n(x,y) \\ &= -D^T \nabla z_k(x,y) + n(x,y) \end{aligned} \qquad (2.2.2)$$

where:

FD(x,y) = frame difference

n(x,y) = higher order terms of Taylor series expansion, which are neglected

z = vector which has as components element differences (ED) and line differences (LD)

If a one-dimensional displacement in the x-direction is to be considered, see Fig.2.9, the displacement vector $\hat{D}$ is reduced to the form

$$\hat{D} = \hat{dx} = -\frac{FD(x,y)}{\partial z(x,y)/\partial x} \qquad (2.2.3)$$

The frame differences for all the pixels of the moving object can be computed if the boundaries of the moving object are known. An approximation of the displacement vector D in terms of dx and dy can be computed using linear regression and neglecting the x,y cross terms.

$$\hat{dx} = -\frac{E[FD(x,y)\partial z(x,y)/\partial x]}{E[(\partial z(x,y)/\partial x)^2]} = -\frac{\sum(FD\ ED)}{\sum(ED)^2} \qquad (2.2.4)$$

$$\hat{dy} = -\frac{E[FD(x,y)\partial z(x,y)/\partial y]}{E[(\partial z(x,y)/\partial y)^2]} = -\frac{\sum(FD\ LD)}{\sum(LD)^2} \qquad (2.2.5)$$

The statistical averages are calculated by summing over the entire area M of the moving object. This algorithm is based on the assumption that the luminance function is linear, an assumption which is valid only within a small area (M_1). As the displacement increases the area decreases and restricts the use of the algorithm only to small displacements.

New estimation algorithms have been developed which overcome the above described limitations. These algorithms are grouped as recursive algorithms, block-matching algorithms and feature-based algorithms.

2.2.1 MOTION COMPENSATION USING RECURSIVE DISPLACEMENT ESTIMATION ALGORITHMS

One could classify recursive displacement estimation algorithms as pel recursive, block recursive and coefficient recursive algorithms. The first recursive algorithm was proposed by Netravali and Robbins [16]. The purpose of this algorithm was to improve the estimation accuracy and to increase the measuring range of the

displacement vector. All recursive estimation algorithms assume an initial estimate and try to produce a better estimation by using an update term based on previous estimates. This operation is described by the following operation:

$$\hat{D}_{i+1} = \hat{D}_i + U_i \tag{2.2.6}$$

where $\hat{D}_i$ is the original estimate, $\hat{D}_{i+1}$ is the new improved estimate and U_i is the update term at iteration i. Based upon the direction in which the iteration is executed the algorithms can be classified as algorithms with horizontal, vertical or temporal recursion. The displaced frame difference is defined by

$$DFD(x,y,\hat{D}_i) = s_k(x,y) - s_{k-1}(x-\hat{dx}_i, y-\hat{dy}_i) \tag{2.2.7}$$

where $s_k(x,y)$ is the luminance value at point (x,y) of a moving object in frame k. The value of DFD can be used as a criterion for calculating the new improved estimate $\hat{D}_{i+1}$. The estimation algorithm proposed by Netravali and Robbins is based on the minimization of the squared value of the DFD recursively with i using a gradient method and is described by the following equation:

$$\hat{D}_{i+1} = \hat{D}_i - \frac{1}{2}\epsilon \nabla_{\hat{D}_i}[DFD(x,y,\hat{D}_i)]^2 \tag{2.2.8}$$

where ϵ is a positive constant and $\nabla_{\hat{D}_i}$ is the gradient operator with respect to $\hat{D}_i$. The choice of ϵ is critical, because a small value of ϵ gives a more accurate displacement but requires more iterations, while a large value of ϵ yields a quick convergence but the estimate is noisy. For motion compensated predictive coding ϵ was chosen to be 1/1024 [82]. The evaluation of $\hat{D}_{i+1}$ yields

$$\begin{aligned}\hat{D}_{i+1} &= \hat{D}_i - \epsilon DFD(x,y,\hat{D}_i)\nabla_{\hat{D}_i}[DFD(x,y,\hat{D}_i)] \\ &= \hat{D}_i - \epsilon DFD(x,y,\hat{D}_i)\nabla s_{k-1}(x-\hat{dx}_i, y-\hat{dy}_i)\end{aligned} \tag{2.2.9}$$

where

$$s_{k-1}(x-\hat{dx}_i, y-\hat{dy}_i) = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} s_{k-1}(x-\hat{dx}_i, y-\hat{dy}_i) \tag{2.2.10}$$

In order to eliminate the effect of quantization noise, the update term can be calculated over an area M using the following formula:

$$U_i = -\frac{1}{2}\epsilon \nabla_{\hat{D}_i} \sum_{j \in M} W_j [DFD(x,y,\hat{D}_i)]^2 \tag{2.2.11}$$

where W_j represent weighting coefficients with the properties that $W_j>0$ and $\sum_{j \in M} W_j = 1$.

If the displacements $\hat{dx}_i$ and $\hat{dy}_i$ are nonintegral, then an interpolation of the luminance $s_{k-1}(x-\hat{dx}_i, y-\hat{dy}_i)$ is

required in order to evaluate DFD and ∇s_{k-1}. In order to carry out this interpolation four neighboring pixels s_A, s_B, s_C and s_D have to be considered as shown in Fig.2.10. s_A is determined by making use of the integral parts of $\hat{dx}_i$ and $\hat{dy}_i$. Two-dimensional interpolation using the fractional parts δx and δy of $\hat{dx}_i$ and $\hat{dy}_i$ respectively renders the interpolated value of the luminance s_{k-1} as given by

$$s_{k-1}(x-\hat{dx}_i, y-\hat{dy}_i) = (1-\delta y)[(1-\delta x)s_A + \delta x s_B] + \delta y[(1-\delta x)s_C + \delta x s_D] \qquad (2.2.12)$$

Other algorithms use variables instead of ϵ in order to achieve a better adaptation to the local image statistics, which in turn improves the rate of convergence and also the accuracy of these recursive displacement estimation algorithms. These algorithms have been proposed by Newton and Raphson, Cafforio and Rocca, and Bergmann [12]. The update term of these algorithms differ only in the value of the denominator which takes the place of ϵ. A comparison of the displacement estimates at consecutive iteration steps for the different algorithms for a sequence of vertical black bars moving in the horizontal direction is presented in Fig.2.11.

Recursive displacement estimation algorithms for transform domain have been described in the literature [26,27]. Coders employing these algorithms are basically of

two types. The first type uses an algorithm which recursively estimates the displacement from the previously transmitted transform coefficients and therefore it does not need to transmit the displacement estimates [26,27]. The second type employs an algorithm which estimates the displacements by taking ratios of accummulated frame difference and spatial difference signals in a block using some past and future data and therefore needs to transmit the displacements estimates to the receiver [27]. The first type of recursive displacement estimation algorithm is based on the pel recursive algorithm developed by Netravali and Robbins [12,16,82]. It assumes that $s_k = (s_{1k}, s_{2k})^T$ represents the coordinate of the upper left-hand pel of the k^{th} block. The blocks are numbered from left to right with $k = 0,1,2,\ldots,$ in each row of blocks. The transpose of a vector or a matrix is denoted by the superscript T. $I(s_k,t)$ is a column vector and represents the pel intensities of block k arranged in a column-scanning manner. ϕ_n represents the n^{th} basis vector of the transform. Hence, the n^{th} coefficient of the k^{th} block of the transform of the present frame is given by:

$$c_n(k) = I^T(s_k,t)\phi_n \qquad (2.2.13)$$

while the displaced previous frame value of the same coefficient is given by:

$$\hat{c}_n(k,\hat{D}) = I^T(s_k - \hat{D},\ t - \tau)\phi_n \qquad (2.2.14)$$

with $\hat{D}$ the estimated displacement of the moving object. The column vector of intensities of the displaced k^{th} block of the previous frame is $I(s_k - \hat{D},\ t - \tau)$. An interpolation among the given previous frame pel intensities is generally required when calculating the elements of $I(s_k - \hat{D},\ t - \tau)$: The goal of the recursive displacement estimation algorithm is to minimize the prediction error in predicting $c_n(k)$ by $\hat{c}_n(k,\ \hat{D})$ by the steepest descent iteration of the form:

$$\begin{aligned}\hat{D}_{n+1}(k) &= \hat{D}_n(k) - \frac{\epsilon}{2}\nabla_{\hat{D}_n(k)} e^2{}_n(k,\hat{D}_n(k)) \\ &= \hat{D}_n(k) - \epsilon\, e_n(k,\hat{D}_n(k))\nabla I^T(s_k - \hat{D}_n(k), t-\tau)\phi_n\end{aligned} \qquad (2.2.15)$$

with $n = 0, 1, \ldots, M-2$ and $k = 0, 1, 2, \ldots$. The initial displacement estimate is given by:

$$\begin{aligned}\hat{D}_o(k) &= \hat{D}_{M-1}(k-1) - \epsilon\, e_{M-1}(k-1,\hat{D}_{M-1}(k-1)) \\ &\quad \cdot\nabla I^T(s_{k-1} - \hat{D}_{M-1}(k-1), t-\tau)\phi_o\end{aligned} \qquad (2.2.16)$$

where $e_n(k,\hat{D}_n(k))$ represents the prediction error of $c_n(k)$ and M is the number of displacement iterations performed per block. A quantized version of the coefficient

prediction error $e_n(k,\hat{D}(k))$ is transmitted to the receiver if $|e_n(k,\hat{D}(k))|>T_n$, where T_n is a given threshold, thus enabling the receiver to update its displacement estimate $\hat{D}_n(k)$ and correct its prediction of coefficient $c_n(k)$. Since only previously transmitted information is used for displacement updating, the separate transmission of the displacement is not necessary [26,27].

The second type of recursive displacement estimation algorithm for the transform domain computes the displacement within displacement blocks with varying sizes (such that the transform block is an exact submultiple of the displacement block in both dimensions). In order to avoid the need for an additional frame store, the algorithm uses coded values of intensities of the previous frame. Either the displaced coefficient from the previous frame or the nondisplaced coefficient from the previous frame, depending on which one performed better for the previous coefficient of the same block, is used for predicting the value of each transform coefficient. There is a tradeoff between the size of the displacement block size and the total number of bits required for a given picture quality. If the size of the block is large enough to average all the local variations of the displacement then the prediction might not be as good as if the size of the block was smaller. On the other hand the overhead required to transmit the displacement estimate is less when larger

blocks are used than when the small blocks are employed [27]. Typical bit rates obtained by empoying the above algorithms are in the range of 150 to 200 Kbits/frame [27]. The performance of coders employing this type of displacement estimation algorithm in the transform domain turns out to be as good as that of the coders using the coefficient recursive displacement estimation algorithm described previously. Both perform better than conventional interframe hybrid coders, achieving bit rates which are about 20 to 40 percent lower than those of the interframe hybrid coders. The performance of the pel recursive algorithms is just slightly better than that of the motion detection algorithms in the transform domain [27].

Another important feature of a recursive displacement estimation algorithm is its range of stability in which it converges to the actual displacement (the correct correlation peak). The stability constraint requires that the update vector be always directed towards the actual displacement and not opposite to it. Stability ranges for the above mentioned algorithms are presented in Fig.2.12 for a typical example of a cross-correlation function. The performance of recursive displacement estimation algorithms when applied to predictive or interpolative coding can be influenced by the presence of quantization noise superimposed to the image signal [12,16].

2.2.2 MOTION COMPENSATION USING DISPLACEMENT ESTIMATION BY BLOCK MATCHING

The algorithms which use the block matching technique for the estimation of the displacement of a point (x,y) take a block of MxN pixels centered at (x,y) in frame k and correlate it with the pixels in a search area in frame k-1 to find the best match. If the maximum displacement in both the horizontal and the vertical direction is dm pixels, then the size of the search area is given by:

$$SR = (M + 2dm)\cdot(N + 2dm) \tag{2.2.17}$$

A normalized two-dimensional cross-correlation function (NCCF) can be measured. The estimate of the displacement is obtained from the position of the correlation peak. The NCCF is given by:

$$NCCF(D) = \frac{R_{s_K s_{K-1}}(D)}{\sqrt{R_{s_K s_K}(0)\; R_{s_{K-1} s_{K-1}}(0)}} \tag{2.2.18}$$

where $R_{s_K s_{K-1}}$ is given by:

$$R_{s_K s_{K-1}}(x,y,\hat{D}) = E[s_k(x,y)s_{k-1}(x-\hat{dx},y-\hat{dy})] \tag{2.2.19}$$

A total number of $(2dm+1)^2$ evaluations of NCCF are required in order to find the correlation peak. This means that the method is computationaly intense and would represent an

important drawback if the algorithm has to be implemented in hardware. There are ways to get around the intense calculations. One of them is to consider a special correlation technique which is making use of the nonstationarity of the second moments to reach a very high accuracy of the displacement estimation. For a block size of 7x7, this method gives an estimation accuracy of 1/10 pel [12]. Another way to reduce the intense calculations is to divide the picture in a fixed number of blocks and to assume that all the pixels undergo the same displacement. The displacement estimation requires now only the calculation of a single displacement vector per block. If the matching criterion is simplified, then the computations required for the displacement estimation are less. Instead of evaluating the NCCF, the mean-square error (MSE) [12,72], or the mean of the absolute frame difference (MAD) [12,49] could be used as a matching criterion. The last criterion has the advantage that no divisions or multiplications are required. The number of shifts required to find the best match is another important characteristic of the displacement estimation by block matching algorithms. This feature can be improved by improving the search technique. Different search techniques have been proposed. The most known ones are the 2D-logarithmic search, the three-step search and modified conjugate direction search.

As the search for the best match moves away from the direction of minimum distortion D(i,j) = MSE(i,j), which is the matching criterion for the 2D-logarithmic procedure, D(i,j) is assumed to be monotonically increasing. The point (i,j) for which D(i,j) is minimum defines the direction of minimum distortion. The process of tracking down the direction of minimum distortion requires five search points at every step and it reduces the size the distances between the search points if the minimum is in the middle or at the boundary of the search area. This procedure is illustrated in Fig.2.13.

The three-step algorithm is closely related to the 2D-logarithmic search algorithm. The three-step search algorithm requires eight search points, except the starting point (i,j), at every step. The approximation of the displacement vector is found by using the MAD criterion. The procedure is repeted until the required accuracy is achieved. After three steps, for a search area with dm<6, the final displacement vector is found [12,49]. An illustration of this procedure is shown in Fig.2.14. A variation of the three-step search algorithm, reduces the number of stages based upon a thresholding sheme which employes two thresholds. The number of search points at each stage is equal to nine. This adaptive three-step algorithm saves unnecessary computations based on the

observation that if there was no significant improvement in motion search from one stage to the next, then no significant overall improvement can be achieved by going to further stages [56].

The simplified conjugate direction search procedure is based on the MAD criterion as well. This algorithm is searching for the direction of minimum distortion, D(i,j), using the MAD criterion, by first finding a minimum in one direction, i for instance, and then in the other direction. A point is considered to be a minimum if it resides between two points which yield higher values for the MAD criterion. This procedure is illustrated in Fig.2.15 [12]. Table 2.1 presents a comparison of the performance of the search procedures presented [12].

2.2.3 MOTION COMPENSATION USING FEATURE-BASED ALGORITHMS FOR DISPLACEMENT ESTIMATION

The feature-based displacement estimation algorithms are two-step algorithms. Predictive coding is used to extract the edges of the moving objects in a first step. The second step consistes in using one of the displacement estimation algorithms which were previously discussed to estimate the displacement of the edges [12]. These techniques require a fairly high amount of computations and are still to be explored and improved [12].

2.2.4 MOTION COMPENSATION USING CONDITIONAL REPLENISHMENT

Another technique used to reduce the bit rate is the conditional replenishment. This technique uses the fullness of the output buffer to control the amount of information to be transmitted. In conditional replenisment the image is devided into moving objects and stationary background. Most of the information to be transmitted is dedicated to the moving objects [2,38,39,46,61]. Therefore if the moving objects undergo a rapid movement the amount of information is suddenly increased and buffer overflow may occur. The solution is to decrease the amount of transmitted information by reducing the resolution in the moving areas of the picture [2,46,61]. Spatial resolution is reduced by the integrating efect of the TV camera and also by subsampling the moving areas while temporal resolution is reduced by low-pass filtering in the temporal frequency domain through the use of the recursive filters, or by subsampling along the temporal axis. As a result of all these operations the resulting coder is a "multimode" one [2,46,61]. This technique combined with motion estimation and a thresholding scheme could be used to obtain bit rate reduction.

2.2.5 MOTION COMPENSATION USING MOTION DETECTION AND MOTION ESTIMATION OR CONDITIONAL REPLENISHMENT APPLIED TO NTSC SIGNAL CODING

Some of the above displacement estimation algorithms have been applied to color pictures as well. Displacement estimation algorithms based on recursive techniques have been used for motion compensated component color coding [53] and motion compensated composite color coding [54]. Motion compensated component color coding using a displacement estimation algorithm based on the block matching technique has also been reported in the literature [56]. The motion compensated component color coding algorithms proposed by Netravali and Prabhu [53] are based on extensions of the pel recursive displacement estimation algorithms introduced by Netravali and Robbins for monochrome TV signals [16,82]. These algorithms use either separate predictors, predictors switching schemes and addressing schemes for each individual component or they use one of the components (luminance in this case) as a reference for the others. Three different predictors are used by Prabhu and Netravali:

1) previous frame (conditional replenishement);
2) switched predictor between previous frame and displaced previous frame (motion compensation);

3) switched predictor, between previous frame, displaced previous frame, and an intraframe predictor which is either the previous element along the same line or the element from the previous line of the same field, directly above the present pel (motion compensation-cumspatial).

The algorithm uses the predictor which yields the smallest sum of the magnitude of the prediction error for certain already transmitted neighboring elements. The neighboring elements are the ones shown in Fig.2.16. The frame difference predictor is expected to be more efficient for unchanged areas (from frame to frame). The displaced frame difference predictor should perform better for the areas whose motion can be estimated, while the spatial (intrafield) predictor would be efficient for areas containing complex unpredictable motion. The results show that by using motion compensation instead of conditional replenishment whenever the sequence of images presents an increased amount of motion yields a lower bit rate. Conditional replenishment performs as good as motion compensation whenever the amount motion is reduced and therefore large unchanged areas are present in the sequence of images. The above approach requires separate predictors for each component and therefore the hardware implementation required would be complex.

One way of reducing the hardware complexity is to use the same displacement estimators for all the components. Netravali and Prabhu tried three schemes for coding the chrominance:

1) $\hat{D}_c = \hat{D}_y$, $\hat{D}_y$ obtained by minimizing $DFD_y^2(x,\hat{D}_y)$
2) $\hat{D}_c = \hat{D}_y$, $\hat{D}_y$ obtained by minimizing
$DFD_y^2(x,\hat{D}_y) + DFD_c^2(x,\hat{D}_y)$
3) $\hat{D}_c$, obtained by minimizing $DFD_c^2(x,\hat{D}_c)$

where D represents the displacement of the object in one frame interval and DFD represents the displaced frame difference. Their results show that the performance of the third scheme yields a 10 percent lower bit rate than the other two, which have equivalent performances, for small to moderate amounts of motion in the sequence of images. Image quality was subjectively judge by the authors to be the same in all cases. For larger amounts of motion the three schemes have equivalent performances. Therefore, the authors concluded that there is no need to employ separate displacements estimators for luminance and chrominance. The coding hardware can be further simplified by switching the chrominance predictor the same way as the luminance predictor. By concatenating the luminance and chrominance samples and then use run-length coding, the number of bits necessary to transmit the location of the predictable or unpredictable luminance and chrominance samples is reduce

and thus, the efficiency of the algorithm is increased. Netravali and Prabhu analysed a simplified scheme which used all the above mentioned simplifications and obtained at least 35 percent improvement in overall bit rates than previous frame prediction, but they concluded that further bit reduction can be achieved by using a separate quantizer for the chrominance components or by transmitting fewer chrominance samples [53]. The bit rates obtained by Netravali and Prabhu are shown in Fig.2.17.

Different motion compensated composite color coding strategies have been attempted by Netravali and Prabhu [54], using extensions of the same recursive diplacement estimation algorithm presented in [16,82,53]. These extensions of the pel recursive displacement estimation algorithm accomodate the particularities presented by the NTSC composite signal due to the presence of the subcarrier. In order to prove that the samples of the composite signal are displaced differently from field to field than the samples of the component signals. The following expressions are used for the signals in the two successive fields:

$$\begin{aligned} S(x,t) &= Y(x,t) + I(x,t)\cos(2\pi f_{sc}t + \sigma) \\ &\quad + Q(x,t)\sin(2\pi f_{sc}t + \sigma) \\ &= Y(x,t) + A^{T}(x,t)C(x,t) \end{aligned} \qquad (2.2.20)$$

where

$$A(x,t) = \begin{bmatrix} \cos(2\pi f_{sc} t + \sigma) \\ \sin(2\pi f_{sc} t + \sigma) \end{bmatrix}, \text{a vector of modulation coefficients} \qquad (2.2.21)$$

and

$$C(x,t) = \begin{bmatrix} I(x,t) \\ Q(x,t) \end{bmatrix}, \text{a vector of chrominance components} \qquad (2.2.22)$$

Previous field signal is given by:

$$S(x,t-\tau) = Y(x,t-\tau) + A^T(x,t-\tau)C(x,t-\tau) \qquad (2.2.23)$$

S(x,t) can also be expressed as

$$S(x,t) = Y(x-D,t-\tau) + A^T(x,t)C(x-D,t-\tau) \qquad (2.2.24)$$

because

$$Y(x,t) = Y(x-D,t-\tau) \qquad (2.2.25)$$

$$C(x,t) = C(x-D,t-\tau) \qquad (2.2.26)$$

But

$$S(x,t) \neq S(x-D,t-\tau) \qquad (2.2.27)$$

because

$$A(x,D) \neq A(x-D,t-\tau) \qquad (2.2.28)$$

Therefore the displacement estimation algorithm described previously requires modifications in order to take into consideration the properties of the NTSC signal due to the presence of the subcarrier. Netravali and Prabhu compared the performaces of four different methods. The first method treats the samples of each subcarrier phase independently and therefore four separate iteration paths are present. In the second method, an average luminance at a pel is defined and used to calculate the displacement which actually amounts to obtaining a quantity which is independent of the phase of the subcarrier, and then use that quantity to estimate the displacement. The third method is a modified version of the first method in that it uses one displacement vector D (which ought to be phase independent) and not separate displacements for each phase. Another advantage of the third method is that the displacement D is updated at each pel location. The fourth method obtains an estimate for the Y, I, and Q components at any spatial location x from the composite signal values in a small neighborhood around x. The running estimate for Y can be obtained using the following formula:

$$\hat{Y}_k = (S_{k+2} + 2S_k + S_{k-2})/4 \qquad (2.2.29)$$

where S_k, $k = 1, 2, \ldots,$ represent the composite signal samples. The I and Q components are given by:

$$\begin{bmatrix} I \\ Q \end{bmatrix} = \begin{bmatrix} A_i{}^T \\ (A_i')^T \end{bmatrix}^{-1} \begin{bmatrix} (2S_k - S_{k+2} - S_{k-2})/4 \\ (S_{k+1} - S_{k-1})/2 \end{bmatrix} \qquad (2.2.30)$$

where A_i is the vector of modulation coefficients at the present sample k and A_i' is defined by:

$$A_i' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} A_i \qquad (2.2.31)$$

The predicted composite signal at the present location is then reconstructed by using the predicted values for Y, I, and Q while the second method is using the composite signal values at the displaced location to predict the composite signal at the present pel, given that an estimate for D was found previously. The strategy behind the fourth method is to demodulate the composite into its components, predict each component individually and then use these predictions of the components to build a prediction for the composite. The predictors used could be classified as:

1) Previous field predictor: It uses D_{01}, a sample one raster scan below as shown in Fig.2.17, because this sample has the same phase as the pel to be predicted (conditional replenishment).

2) Intrafield predictor: Samples from the present and previous lines are used by this predictor as shown in Fig.2.17 and the following formula:

$$\hat{X} = (D_{12} + D_{32} + 2D_{31})/4 \qquad (2.2.32)$$

(motion compensation).

3) Displaced previous field predictors: Are obtained from displacement estimate as given by:

$$S_i(x-\hat{D}_i, t-\tau) = S_1 + \underline{(\Delta x)}^T \underline{\nabla S_1} \qquad (2.2.33)$$

where

$$\underline{\Delta_x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad (2.2.34)$$

and

$$\underline{\nabla S_1} = \begin{bmatrix} (S_2 - S_1)/4 \\ S_1 - (S_3 + S_4)/2 \end{bmatrix} \qquad (2.2.35)$$

S_i, Δx_i and $x-\hat{D}_i$ are indicated in Fig.2.19 (motion compensation cumspatial).

Netravali and Prabhu proposed four different modes of operation. These modes are described below:

Mode A: Fixed previous field predictor for all pels.

Mode B: Switched prediction between previous field predictor and intrafield predictor.

Mode C: Switched prediction between previous field predictor and displaced previous field predictor.

Mode D: Switched prediction between previous field predictor, intrafield predictor, and displaced previous field predictor.

The absolute value of the prediction error for each of the surrounding previously coded four pels (A_{12}, C_{32}, B_{32} and A_{32} located as shown in Fig.2.18) for each of the three predictors considered is calculated in order to control the switching process. The predictor used is the one which yields the smallest sum of the prediction error magnitudes calculated at the four neighborhood samples. If two or more predictors happen to give the smallest sum then the order of preference is previous field, displaced previous field and intrafield predictor. The prediction error is transmitted only if it is larger than a certain threshold which was subjectively designated. The simulation results show that methods 2 and 3 have a superior performance compared to that of methods 1 and 4 (see Fig.2.20) [54]. Motion compensation gave a 15 percent improvement in bit rate over conditional replenishement while motion compensation cumspatial gave a 25 percent improvement in bit rate over conditional replenishement. The improvement in bit rate is not as good as the one presented by component coding because of the difficulty of estimating

the displacement from the composite signal (reason being the frequency multiplexing of the Y, I, and Q components) and of the errors introduced in the prediction of a sample by the process of interpolation. The errors which appear in the process of interpolation are due to the fact that pels of the same phase are further apart.

An algorithm which uses the block matching method to estimate the displacement and is used to encode color signals is presented by Lin and Kwatra [56]. The scheme employed by this algorithm is a sequential, nonoverlaping, block-by-block displacement search method. This algorithm is different than the one proposed in [49], in that it uses a variable stage search and not a three stage search. The scheme first defines:

B = an MxN block of data to be processed.

R = the motion search range (MSR) of B from memory.

The minimum of

$$d(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} f[b(m,n)-r(m+p_i(j),\ n+q_i(j))] \qquad (2.2.36)$$

where

i = 1, 2, 3, ...k represents the stage number

j = 1, 2, 3, ...9 represents the nine locations searched for each stage

b, r represent elements of the matrices B and R;

p, q represent the pixel location with respect to the center of the block where motion search will be performed;

f is a measuring positive function which is defined to be $f[.] = |.|$,

gives the motion trajectory. The size of matrix R at stage i is given by:

$$[R]_i = [(M+HMV_i x2)x(N+VMV_i)] \qquad (2.2.37)$$

where

HMV_i - maximum horizontal displacement for a block with respect to MSR center;

VMV_i - maximum vertical displacement for a block with respect to MSR center.

The sum of the absolute difference values of pels without any motion compensation, d(1,1), is first computed. The value of d(1,1) is compared to a threshold value T_1. If $d(1,1)<T_1$ no further search is conducted and the motion vector is determined. If that is not the case, i.e. $d(1,1)>T_1$, the minimum of d(1,j), j = 1, ..., 9, is computed. If $d(1,j)_{min}<T_1$ the search is stoped and the motion vector is computed. If $d(1,j)>T_1$ the search

continues to the second stage with location which results in $d(1,j)_{min}$ as the new center. At the second stage motion search $d(2,j)_{min}$ is computed. Then $d(1,j)_{min} - d(2,j)_{min}$ is computed and then compared to a second threshold T_2 and if $d(i,j)_{min}-d(2,j)_{min}<T_2$ then the search stops and the motion displacement vector is computed. If $d(1,j)_{min} - d(2,j)_{min}<T_2$ then the search resumes at the next stage. If the motion search continues to the last stage k, then the motion vector corresponding to $d(k,j)_{min}$ is determined. The motion vector information and the corresponding quantized difference values for the block are transmitted. In order to transmit less overhead information the size of the block has to be larger, but this would increase the number of bits used to transmit the difference values. This is due to the increased number of mismatches. Hence, the overall bit rate might be increased on the average. The algorithm proposed by Lin and Kwatra eliminates this difficulty by dividing the larger block into subblocks and classifing the subblocks as active and inactive. The classification is done based on the values of the mean and standard deviation of the pixels. If the subblock is classified as active the difference values are transmitted together with the motion vector information. On the other hand if the subblock is classified as inactive no differences are transmitted. Every element of the active subblocks is quantized before transmission using a 17 level nonuniform quantizer. The

quantizer is designed based on the observation that for over 75 percent of the active subblocks the difference values are distributed in a small range between -18 and 18. The algorithm was found to require 40 percent less computations than the nonadaptive one [49]. 90 percent of the blocks did not require any processing beyond the second stage. The algorithm resulted in a reduction of up to 1 bit/pel over DPCM coder. By using activity segmentation on the motion compensated blocks the algorithm achieves 20 percent more compression.

As a final remark, one must agree that the displacement estimation algorithms are valuable means to achive bandwidth compression whenever bit rates lower than 1.5 Mbps are required.

3 A HADAMARD TRANSFORM DOMAIN APPROACH TO VIDEO BANDWIDTH COMPRESSION

The literature survey clearly indicates that in order to design video coders for video conferencing applications operating under 1 Mbps rates some form of motion estimation and conditional replenishment should be used. One has to decide whether to operate on the component signals (Y, I, Q) or directly on the composite signal. One has also to decide on whether to implement the algorithm in the spatial domain or in the transform domain.

Various researchers have developed motion estimation algorithms that operate on the component signals [53,73]. Sawada, et al. [73], and Netravali, et al. [53] have shown that it is important to incorporate all the three components (Y, I, Q) when estimating motion. The drawback of operating on the component signals (whether in the temporal domain or transform domain) is the complexity of operating on the three signals independently. Researchers have reduced the complexity by using only the luminance (Y) signal for motion estimation and assuming that the chrominance signals behave in the same fashion. It was shown that the chrominance signals do not necessarily behave as the luminance signal, i.e. changes in chrominance are not necessarily accompanied by changes in luminance [84].

In an attempt to circumvent the complexity encountered by using the component signals, researchers have attempted motion estimation using the composite signal directly [54,56]. The resulting algorithms required less hardware and computational complexity but resulted in inferior motion estimation as a result of the subcarrier embeded in the composite signal.

Ideally, one would like to implement a motion estimation algorithm which operates on all three component signals and has the complexity of an algorithm that operates on the composite signal. Our research investigated a motion estimation algorithm which operated on the transform of the composite signal, since pel (data, or temporal) domain operation required separate processing of the three signals. We chose to use the Hadamard transform because its (hardware) implementation requires only additions and subtractions. Unlike other techniques which operate in the transform domain (and required three separate transform signals) we required only one transform by taking advantage of the properties of the Hadamard transform to separate the composite signal into its components (when sampling at $4f_{sc}$). Therefore, the hardware complexity generally required by motion detection, estimation and compensation algorithms employing the component signals is reduced. The motion detection and

estimation algorithms also make use of the chrominance signals and provide a better estimate than using only the luminance signal.

The basis of the proposed motion detection, estimation and compensation algorithm is the ability to separate the properly sampled NTSC composite signal into its components by using the Hadamard transform. This represents a major portion of the research conducted and is presented in the next section.

3.1 NTSC COMPOSITE SIGNAL SEPARATION IN THE HADAMARD DOMAIN

3.1.1 WHY TRANSFORM CODING AND WHY HADAMARD TRANSFORM ?

Technological developments in digital hardware (both complexity and speed) have made digital processing of video signals cost competitive with analog techniques. This is especially true in the area of bandwidth compression of NTSC composite video. Here we consider transform coding which has found extensive use in bandwidth compression of NTSC signals [12,70]. Transform coding is generally used to code the component signals, YIQ. This means that the coder will require additional hardware to decompose the

composite signal into its components. However, the advantage gained is that each component signal can be coded separately.

Some researchers have investigated transform coding of the NTSC composite signal directly [48,75,76]. These attempts define the data block boundaries, taking into consideration the phase of the subcarrier, so as to maximize the correlation among the samples within the block. This results in compacting most of the energy, in both the luminace and chrominance components, towards the low frequency coefficients in the transform domain. This technique requires that the phase of the subcarrier be taken into consideration but does not require the additional hardware to demodulate the NTSC composite signal. However, these techniques do not provide the flexibility of coding the components separately.

We show that it is possible to apply an NxN Hadamard transform directly to the NTSC composite signal and still maintain the flexibility of coding each component separately. By taking into consideration the sampling phase of the NTSC composite signal at four times the color subcarrier we show that the components, YIQ, are separated into different areas of the transform domain. This allows individual coding of the component signals by manipulating the bit allocation of those coefficients pertaining to a particular component. At the receiver, all that needs to

be done is to perform an inverse Hadamard transform to generate the NTSC composite signal. We concentrate on the Hadamard transform because it is easier to implement in hardware [22,77] and because the theory is easier to visualize.

In the next section we develop the mathematical theory for the mapping of the chrominance signals in the Hadamard domain. This development does not consider the effect of spectrum spreading due to the partitioning of the data into NxN blocks but shows that the effects of data trunction and the resulting spectrum spreading does not prevent component separation in the Hadamard domain. The simulation results of component separation using the Hadamard transform are presented in chapter 4.

3.1.2 HADAMARD SPECTRUM OF THE Q AND I SIGNALS

The Walsh functions $\{Wal_n(x)\}$ n=0,1,.. are defined in the interval $0 \le x \le 1$ by the iterative relationship [8]:

$$Wal_0(x) = 1 \qquad 0 \le x \le 1$$

$$Wal_1(x) = \begin{cases} 1 & x<1/2 \\ -1 & x>1/2 \end{cases} \qquad (3.1.1)$$

$$Wal_n(x) = \begin{cases} Wal_{[n/2]}(2x) & x<1/2 \\ Wal_{[n/2]}(2x-1) & x \ge 1/2,\ n \text{ odd} \\ -Wal_{[n/2]}(2x-1) & x \ge 1/2,\ n \text{ even} \end{cases}$$

[ ] = integer part of.

The sequency of a Walsh function is equal to the number of zero crossings in the interval $0<x<1$ [78]. If the Walsh functions with sequency less than or equal to 2^n-1 are sampled at $N=2^n$ uniformly spaced points, a square matrix is produced. Except for the ordering of the rows, the discrete Walsh matrix is equivalent to the Hadamard matrix of rank N. In image processing applications no distinction is generally made between these two matrices which we shall refer to as a Hadamard matrix. The Hadamard matrix of order N=8 is given by:

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \qquad (3.1.2)$$

The two-dimensional Hadamard transformation is a unitary orthogonal separable transform where the columns of the Hadamard matrix serve as the orthogonal basis vectors. In general, if the array f(x,y) represents the samples of an NTSC composite signal over an array of N^2 points, then the two-dimensional Hadamard transform, F(u,v), of f(x,y) is given by the matrix product:

$$F = \frac{1}{N^2} H_N \, [f] \, H_N \qquad (3.1.3)$$

where H is the Hadamard matrix containing the first N Walsh functions. The elements of F are given by F(u,v) where the indices u and v correspond to variations in the vertical and horizontal direction respectively. The inverse transformation is defined by the matrix product:

$$f = H_N \; [F] \; H_N \tag{3.1.4}$$

The result of the two-dimensional Hadamard transform is a NxN coefficient matrix where each coefficient represents the projection of the data matrix onto the particular Hadamard basis picture. The basis pictures are defined by:

$$B(i,j) = >x_i x_j< = x_i x_j^T \; ; i,j=0,1,2,3..N \tag{3.1.5}$$

where x_i, x_j are the i^{th} and j^{th} basis vectors with sequency i and j respectively. The Hadamard basis pictures for an 8x8 transform are shown in Fig.3.1.

The NTSC composite color signal is given by:

$$\begin{aligned} f(t) = Y(t) &+ I(t)\cos(2\pi f_{sc} t + 33^o) \\ &+ Q(t)\sin(2\pi f_{sc} t + 33^o) \end{aligned} \tag{3.1.6}$$

where: f(t)= The NTSC composite color signal.

Y(t)= The luminance signal.

I(t),Q(t)= The baseband color differential signals.

f_{sc}= 3.579MHz = (227+1/2)f_h= color subcarrier.

f_h= 15,734.26Hz = the horizontal line frequency.

The luminance and chrominance information in an NTSC composite signal are contained in the same frequency band. The luminance component contains most of the signal energy and has frequency components which are statistically clustered about harmonics of the frame rate (30Hz) with stronger components located at harmonics of the horizontal line frequency. The chrominance signals are modulated in quadrature and transmitted on the color subcarrier at a frequency of 455/2 times the line frequency. The chrominace component contains most of its energy in bands around odd multiples of half the line frequency [79]. In general, motion introduces an overlap of the chrominance and luminance spectra and the NTSC composite signal cannot be separated exactly into the original components. Transform coders that operate on the components of the NTSC video signal usually accomplish the separation by using either an analog or digital comb filter [80]-[83].

There are two properties of the composite signal which are of importance. Firstly, the color signals appear as amplitude modulated in quadrature with amplitudes equal to the baseband color differential signals. Secondly, the phase of the modulating subcarrier changes 180^{o} from line to line.

From a huristic point of view, consider that I(t) and Q(t) have constant values. Then if the NTSC composite signal is sampled at four times the color subcarrier and the sampling occurs at 12°, 102°, 192° and 282° of the subcarrier, the resulting sampled composite signal will be:

$$f(nT) = Y(nT) + I(nT)\cos(n\pi/4) + Q(nT)\sin(n\pi/4) \quad ;n=1,2,3,.... \tag{3.1.7}$$

where: $T = 1/4f_{sc}$

The sampled modulated I and Q signals are shown in Fig.3.2. If we consider an 8x8 Hadamard transformation, we notice that the sampled modulated I and Q signals shown in Fig.3.2 correspond exactly, in the horizontal direction, to the Hadamard basis vectors of sequency 3 and 4, respectively. This means that the 8x8 matrix, [f], obtained from the sampled composite signal will project the chrominance signal energy along columns 4 and 5 of the coefficient matrix, [F], which correspond to the basis pictures with horizontal sequency of 3 and 4. In the vertical direction the chrominance signal corresponds to the Hadamard basis vector of the highest sequency due to the 180° phase shift of the color subcarrier from line to line. Thus, the chrominance signal energy must be mapped along row 8 of the coefficient matrix. The result is that the D-C value of the modulated Q and I signals will be mapped onto coefficients (7,3) and (7,4) of the 8x8 Hadamard transform.

The above results can be generalized for any I(t) and Q(t). Here we make the initial assumption that the component signals, when partitioned into NxN blocks, maintain their original bandwidth, i.e., we do not initally worry about the effects of data discontinuity due to data truncation and resulting spectrum leakage. The effect of sampling the modulated I and Q signals is equivalent to spatially multiplying the baseband I and Q signals by the corresponding modulation matrix. The modulation matrices for the Q and I signals for a 4x4 picture block are:

$$M_Q^4 = \begin{bmatrix} 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \qquad M_I^4 = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix} \tag{3.1.8}$$

For transform size greater than 4x4 the corresponding modulation matrices are given by:

$$M_L^{2^k} = \begin{bmatrix} M_L^{2^{k-1}} & M_L^{2^{k-1}} \\ M_L^{2^{k-1}} & M_L^{2^{k-1}} \end{bmatrix} \quad L = Q \text{ or } I; \quad k=3,4,.. \tag{3.1.9}$$

Thus, an NxN array of the sampled chrominance signals can be written as:

$$\sqrt{2}\, D_m = D \otimes M_L^N \quad ; \; L=Q \text{ or } I \tag{3.1.10}$$

where: D = NxN sample array of the baseband chrominance signal.

D_m = NxN samples of the modulated chrominace signal.

M_L^N = NxN modulation matrix corresponding to the particular chrominace signal in Eq.(3.1.9).

⊛ = denotes spatial multiplication, i.e.

$$L_m(i,j) = L(i,j)M_L^N(i,j) \; ; \; i,j=1,2,..,N$$

The modulation matrices of Eq.(3.1.10) can be separated into two operators, one operating on the rows of the data and the other operating on the columns of the data. Thus, Eq.(3.1.10) can be written as:

$$\sqrt{2}\; D_m = {}_vM_L^N \circledast [D] \circledast {}_hM_L^N \; ; \; L=Q \text{ or } I \qquad (3.1.11)$$

where: ${}_vM_L^N$ = NxN matrix whose columns are all equal to the vector $Wal_{(N-1)}(y)$.

${}_hM_L^N$ = NxN matrix whose rows are all equal to the vectors $Wal_{(n/2)-1}(x)$ when L=Q or $Wal_{n/2}(x)$ when L=I.

To obtain a complete mapping of the modulated chrominace signals in the Hadamard domain let the NxN Hadamard transform of the baseband signal I(t) be given by:

$$F_I = \frac{1}{N^2} H_N [I] H_N \qquad (3.1.12)$$

Let the NxN Hadamard transform of the modulated signal $I_m(t)=I(t)\cos(2 \quad f_{sc}t+33^o)$ be given by:

$$F_{I_m} = \frac{1}{N^2} H_N [I_m] H_N \tag{3.1.13}$$

Equation (3.1.13) can be written with respect to the baseband signal and the modulation matrix by:

$$\begin{aligned} F_{I_m} &= \frac{1}{N^2\sqrt{2}} H_N [I \otimes M_I] H_N \\ &- \frac{1}{N^2\sqrt{2}} H_N [{}_vM_I^N \otimes I \otimes {}_hM_I^N] H_N \\ &- \frac{1}{N^2\sqrt{2}} [H_N \otimes {}_vM_I^N] I [{}_hM_I^N \otimes H_N] \end{aligned} \tag{3.1.14}$$

The spatial product of two Hadamard basis vectors results in another basis vector and is given by:

$$Wal_a(x) \otimes Wal_b(x) = Wal_{a\oplus b}(x) \tag{3.1.15}$$

where $\oplus$ = sum modulo 2 when a and b are expressed in binary form, i.e., 1+0=0+1=1, 0+0=1+1=0.

Thus, Eq.(3.1.14) can be expressed in the form:

$$F_{I_m}(u,v) = \frac{1}{\sqrt{2}} F_I(u\oplus(N-1), \quad v\oplus(N/2)) \tag{3.1.16}$$

In general, for an NxN transform, the Hadamard transform of the modulated chrominance signals are related to the Hadamard transform of the baseband signal by:

$$F_{L_m}(u,v) = \frac{1}{\sqrt{2}} F_L(u\oplus(n-1),\ v\oplus k)\ ;\ L=Q \text{ or } I \qquad (3.1.17)$$

$$k = \begin{cases} (N/2)-1 & ;\ L=Q \\ (N/2 & ;\ L=I \end{cases}$$

Equation (3.1.17) relates the mapping of the baseband chrominance signal to the mapping of the modulated chrominance signal in the Hadamard domain. Having one mapping the other can be obtained. Since the baseband Q and I signals are bandlimited to 0.5MHz and 1.5MHz respectively, the Hadamard transform should ideally have zero value for those coefficients which represent frequency variations greater than the highest frequency component of the baseband signal. The effect of truncating the data by an NxN window in performing the two dimensional transform is to introduce energy leakage into the higher frequency components of the Hadamard domain. This results in an overlap in the spectrum of the components in the Hadamard domain and prevents complete separation of the componets.

3.2 MOTION DETECTION, ESTIMATION AND COMPENSATION

We present a different approach to motion detection, estimation and compensation than the traditional techniques. The first difference comes from the fact that the detection and the estimation of the motion takes place in the transform domain and not in the data domain. Secondly, the motion detection and estimation are based on thresholding techniques applied to the Y, I and Q component signals in the transform domain. The energy pertaining to the Y, I and Q signals is concentrated by the Hadamard transform into known coefficients as shown in section 3.1.2. The motion detection, estimation and compensation algorithm studied has the block diagram shown in Fig.3.2.1.

The three distinct cases, when evaluating the motion of a block in the transform domain are: 1) no motion detected; 2) motion detected and estimated; and 3) motion detected but not estimated. These cases can generally be defined by the motion detection function, $f_{1MD}(r,s,TH)$, and a motion estimation function, $f_{2ME}(r,s,TH1(i,m))$:

$$f_{1MD}(r,s,TH) = \begin{cases} 0 & , \ SCD(i,m) < TH(i,m) \\ 1 & , \ SCD(i,m) > TH(i,m) \end{cases} \tag{3.2.1}$$

$$f_{2ME}(r,s,TH1(i,m)) = \begin{cases} 0 & , \ SQCD(i,m) > TH1(i,m) \\ q & , \ SQCD(i,m) < TH1(i,m) \end{cases} \tag{3.2.2}$$

where r and s are the coordinates of the block; i and m represent the coefficient coordinates within each block; TH(i,m) is the motion detection threshold; SCD(i,m) represents the squared frame to frame coefficient differences; q is the block displacement vector; SQCD(i,m) represents the square frame to frame coefficient differences; and TH1(i,m) represent the motion estimation thresholds.

3.2.1 MOTION DETECTION THRESHOLD SETTING

A crucial role in the motion detection algorithm is played by the threshold setting. It is very important to select the right thresholds in order to detect the motion in the picture. If the threshold setting is too low, the noise in the picture might cause the algorithm to classify blocks as moved when in fact no motion is present. On the other hand, if the threshold setting is too high, blocks which moved might be classified as not changed. In either case reconstruction errors could result.

In general the motion detection thresholds will depend on the transform coefficients being considered. The higher frequencies coefficients are more sensitive to background noise, and change even if no motion is present between frames. The lower frequency coefficients are better

candidates for detecting motion since they are less sensitive to background noise.

In order to determine the motion detection thresholds, statistics of the square frame to frame coefficient differences pertaining to the DC and surrounding frequency components for Y, I and Q signals were considered for two separate sequences. The results of the statistical measurements are presented in Fig.3.2.2. We note that the curves behave in a similar manner. This indicates that the number of square frame to frame coefficient differences for those coefficients containing the DC energies of the Y, I, and Q signals and the surrounding coefficients are of the same order of magnitude. In this study only the coefficients containing the DC energies of the Y, I and Q signals were subsequently considered in the thresholding process because inclusion of any other coefficients did not improve the motion detection results obtained by simulation. Thus, only coefficients (0,0), (7,3) and (7,4) are used for the motion detection algorithm. The results for the coefficients containing the DC energies indicate that a large number of square frame to frame coefficient differences are less than the values of 5.0 in the case of the coefficient F(0,0), 2.0 in the case of the coefficient F(7,4) and 1.0 in the case of the coefficient F(7,3). Therefore, differences less than the above mentioned values

are not considered relevant. Threshold values greater than those above mentioned may be used as thresholds for motion detection.

3.2.2 CHANGE MAP

In order for the transmitter to keep track of which block in the present frame underwent motion a change map is used. This map is generated by using the square of the frame to frame coefficient difference, the motion detection thresholds selected, and the motion detection function, $f_{1MD}(r,s,TH)$.

Let us denote the current block in the present field by B(x,y,t) and the corresponding block in the j^{th} previous field by B(x,y,t-jT), where T is the time between the two fields, and (x,y) points to a specific NxN block. The square of the difference of corresponding coefficients in the two blocks is given by:

$$SCD(i,m) = \{F_{B(x,y,t)}(i,m) - F_{B(x,y,t-jT)}(i,m)\}^2 \qquad (3.2.3)$$

where x, y = 1, 2,..., 512/N and (i, m) = {(0,0), (7,3), (7,4)}.

If all three SCD(i,m) are less than their threshold then the block is recognized as not changed. In this case

the motion detection function has a value of 0 which is entered in the appropriate (x,y) position of the change map. If any of the SCD(i,m) is greater than the threshold then the block is recognized as changed and the motion detection function has a value of 1 which is entered in the appropiate (x,y) position of the change map. At the end of this process a map of changed blocks is generated. The changed map is "median filtered" in order to remove isolated "changed" blocks surrounded by "unchanged" blocks and to remove isolated "unchanged" blocks surrounded by "changed" blocks.

3.2.3 MOTION ESTIMATION AND THRESHOLD SETTING

The result of the motion detection algorithm is a change map. This map has 0's in the position corresponding to blocks which have not changed and 1's in the positions corresponding to blocks which have changed. In the later case a search is initiated to find a similar block in the j^{th} previous field. The outcome of this search is the value of the motion estimation function $f_{2ME}(r,s,TH1(i,m))$ which represents either the block displacement vector or indicates that no corresponding block in the j^{th} previous field was identified.

If a block has been classified as moved in the change map an exhaustive sequential search is initiated. In order to cary on the sequential search, a search area is defined.

In this study, the search areas in the j^{th} previous field were selected as either a 5 block by 5 block or a 3 block by 3 block area. The block denoted by B(x,y,t-jT) is located in the middle of the search area as shown in Fig.3.2.3. The motion estimation algorithm makes use of:

1) The above defined search area.
2) The thresholds for motion estimation.
3) The motion estimation function $f_{2ME}(r,s,TH1(i,m))$.
4) The following square coefficient difference formula:

$$SQCD(i,m) = \{F_{B(x,y,t)}(i,m) - F_{B(x-D1,y-D2,t-T)}(i,m)\}^2 \qquad (3.2.4)$$

where $-2<D1<2$, $-2<D2<2$, $(i,m) = \{(0,0), (7,3), (7,4)\}$.

The variables D1 and D2 are determined by the position of the block in the search area. The three values obtained for SQCD(i,m) are compared with the thresholds TH1(i,m). If all $SQCD(i,m) \leq TH1(i,m)$ then the block is recognized as being similar, but the search in the defined search area does not stop at that point. The square difference which was found to be less than the threshold becomes the new threshold and the search continues until all the blocks in the search area have been exhausted. This ensures that the proper block is designated as the similar one. If a similar

block is designated, the value of the motion estimation function represents the displacement vector which is encoded and transmitted to the receiver. Due to the size of the search area only 5 or 4 bits, corresponding to a 5x5 or 3x3 search area, respectively, are necessary to encode the value of the displacement vector. If at the end of the search SQCD(i,m)>TH1(i,m), indicating that no match was found, the motion estimation function takes on a value of 0 and a differential coding technique is employed to transmit the differences between 10 preselected coefficients of blocks B(x,y,t) and B(x,y,t-jT), which the receiver already has. The reason for transmitting only 10 coefficients is that approximately 90% of the signal energy is contained in these coefficients.

The thresholds for motion estimation have to be set such that enough blocks are matched to allow bandwidth compression but low enough to prevent erroneous matching and reconstruction errors. The motion estimation thresholds are selected based upon the statistics shown in Fig.3.2.2. The tradeoff is between accepting errors in matching or having a block which cannot be matched and therefore has to be transmitted using a differential technique, which is the case of blocks containing large edges which move a fraction of a block.

3.2.4 BLOCK CLASSIFICATION AND BIT ASSIGNMENT FOR MOTION COMPENSATION

At the end of the motion detection and motion estimation process blocks can be classified as:

1) Blocks which did not change.

2) Blocks which changed and are matched with blocks in the transform of the previous video frame.

3) Blocks which changed and cannot be matched with blocks in the transform of the previous video frame.

Since there are three different classes of blocks , as shown in the previous section, three different approaches for bit assignment have to be employed. A fourth one is added whenever picture upgrading is employed. The bit assignment algorithm has the following four different modes of operation:

Mode 1: No motion detected. Run length coding is employed to encode the number of blocks which are recognized as unchanged.

Mode 2: Motion detected. The motion is detected by using a thresholding technique, but cannot be estimated and therefore cannot be compensated for. A differential technique of the conditional replenishment type is employed to transmit the information.

Mode 3: Motion detected, estimated and compensated. The motion is detected by using a thresholding technique, is estimated by using a block matching technique and is compensated for by transmitting the address of the matched block.

Mode 4: No motion or very little motion detected and empty output buffer will allow the upgrading of the picture by transmitting 10 preselected square frame to frame coefficient differences of all the blocks which were classified as changed between the last two processed video frames. If the output buffer is still empty and the amount of motion between the following video frames is low, other square frame to frame coefficient difference of the same blocks are transmitted and the process continues as long as the two conditions are satisfied.

It is obvious that the overhead necessary to flag any of these modes of operation is two bits. The bit assignment for each mode is described below.

3.2.4-1 BIT ASSIGNMENT FOR MODE 1

If the value of the motion detection function $f_{1MD}(r,s,TH(i,m))$ is 0 then no motion was detected. After "1-D median filtering" the change map a group of consecutive 0's and 1's are produced. A two bit header, which represents overhead, will signal the "run length" mode whenever successive 0's are encountered. The number of successive 0's is encoded and transmitted to the receiver. When the receiver receives the two bit header for Mode 1 the number which follows is added to the address of the last processed block. All the blocks between the address of the last block reconstructed and the new address are copied to the new video frame from the previous video frame. Since both the transmitter and the receiver have the previous video frame no extra information is required to be transmitted for the reconstruction of these blocks. Hence, in Mode 1, the algorithm generates an overhead to flag the mode of operation and transmits the encoded value of the count of unchanged blocks.

3.2.4-2 BIT ASSIGNMENT FOR MODE 2

When the algorithm operates in Mode 2 the value of the motion detection function, $f_{1MD}(r,s,TH)$, is 1 and the value of the motion estimation function, $f_{2ME}(r,s,TH1(i,m))$, is 0. This means that the block was classified as moved but no match was found in the previous video frame. Therefore, a differential technique is used to encode the information using a fix number of bits for each coefficient. The number of bits allocated to a particular frame to frame coefficient difference is obtained using the image statistics. The resulting bit assignments for the two sequences are presented in TABLE 3.2.1.

The differences of the corresponding coefficients in the corresponding blocks are assigned a number of bits to be encoded using the following formula:

$$b_i = \frac{1}{2} \log_2 \frac{\sigma^2(i,m)}{D} \qquad (3.2.5)$$

where D represents the maximum distortion allowed and the variance σ^2 is given by:

$$\sigma^2(i,m) = \frac{1}{N} \sum_{m=1}^{N} [x_m(i,m) - x(i,m)]^2 \qquad (3.2.6)$$

The mean frame to frame coefficient difference between two corresponding blocks is given by:

$$x(i,m) = \frac{1}{N} \sum_{m=1}^{N} x_m(i,m) \qquad (3.2.7)$$

where $x_m(i,m)$ is the frame to frame coefficient difference of the coefficient denoted by i and m and N represents the total number of blocks present in the video frame (excluding the synchronization signals).

3.2.4-3 BIT ASSIGNMENT FOR MODE 3

If the algorithm is operating in Mode 3 then the motion detection function is 1 and the motion estimation function takes on the value of the displacement vector q. A two bits header is followed by the encoded value of the displacement vector. The number of bits necessary to encode the displacement vector depends on the size of the search area. If the search area is 5 by 5 blocks then only 5 bits are necessary to encode the value of the displacement vector. If the size of the search is 3 by 3 blocks then only 4 bits are used for the encoding of the value of the displacement vector. The values of the displacement vector for the 5 by 5 blocks search area and for the 3 by 3 blocks search area are shown in Fig.3.2.4, respectively.

3.2.4-4 BIT ASSIGNMENT FOR MODE 4

Mode 4 is the mode in which the picture upgrading takes place. This mode of operation is possible only if the output buffer is almost empty and very little or no motion at all has been detected. In this case the latest change map is checked and only the frame to frame coefficient differences of those blocks which were classified as changed are encoded and transmitted. The bit assignment procedure is similar to the one used in Mode 2. The only difference is that the number of bits allocated for each frame to frame coefficient difference is less than in the case of Mode 2. Bit assignment matrices for Mode 2 and Mode 4 are shown for comparison purposes in TABLE 3.2.2. The frame to frame coefficient differences are upgraded in a predetermined order. The ten coefficients used for the transmission of the information are upgraded first. If at the next video frame the amount of motion is again small other coefficients will be upgraded and the upgrading will continue in the same manner until either the buffer fills out or the amount of motion increases. The order in which the coefficients are upgraded is shown in Fig.3.2.5.

4 SIMULATION RESULTS

4.1 NTSC COMPOSITE SIGNAL SEPARATION IN THE HADAMARD DOMAIN SIMULATION RESULTS

Five color images were used to simulate the component separation of NTSC composite signals in the Hadamard domain. Four of the images are the standard test slides SMPTE1, SMPTE2, SMPTE3, and SMPTE15. The fifth image is that of a BOY playing with toys. The NTSC composite signals were generated using the system shown in Fig.4.1.1. The RGB signals of each image were sampled at four times the color subcarrier and stored in a VAX-11/750 computer. The Y, I and Q signals are generated by linear combinations of the RGB signals. The Q and I signals are bandlimited to 0.5MHz and 1.5Mhz respectively, modulated by the subcarrier with the phase relation shown in Fig.4.1.2, and summed with the Y signal to generate the composite signal. The composite signals were used as the input to test the component separation of the Hadamard transform.

The component signal mapping in the Hadamard transform domain was shown in section 3.1.2. In order to separate the NTSC composite signal into its components we have to assign each transform coefficient to a component. Various

criteria can be used to assign coefficients or even portions of coefficient to each component. Since we wish only to demonstrate the technique of component separation in the Hadamard domain the standard deviation was used to determine the transform coefficient assignment. The three signals Y(t), I(t)cos(2 f_{sc}t+33°), and Q(t)sin(2 f_{sc}t+33°) were transformed for each of the five images. The standard deviation equals the positive square root of the variance given by:

$$\sigma^2(i,j) = \frac{1}{T} \sum_{K=1}^{T} [F(i,j) - \bar{F}(i,j)]^2 \quad i,j=0,1,..,N-1 \qquad (4.1.1)$$

where: T = Total number of transform matrices.

F(i,j) = the (i,j) coefficient.

$\bar{F}(i,j)$ = the average value of the (i,j) coefficient.

The standard deviation for the three signals Y(t), I(t)cos(2πf_{sc}t+33°) and Q(t)sin(2πf_{sc}t+33°) averaged over the active video portion of the five images are given in TABLE 4.1.1. The effect of windowing and truncating the data results in the energy of the baseband signals being spread over the Hadamard domain. A particular coefficient is assigned to the component which has the largest standard deviation for the particular coefficient. The coefficient assignment for a 4x4, 8x8 and 16x16 transform are given in TABLE 4.1.2.

We notice from TABLE 4.1.1 that the modulated Q and I coefficients are assign locations which normally represent the high frequency of the Y signal. To see the effect on the Y signal of setting these coefficients equal to zero we used the TEST patterns shown in Fig.4.1.3. Fig.4.1.3a is the original TEST patterns sampled at four times the color subcarrier. The reason the entire image is not present in the horizontal direction is that our imaging system only stores 512 samples per line. Figs.4.1.3b, c and d are processed images using a 4x4, 8x8 and 16x16 Hadamard transform respectively. These images were obtained by taking the appropriate Hadamard transform of the original image, setting to zero the coefficients assigned to the Q and I signals in TABLE 4.1.2, and performing the inverse Hadamard transform. The original and processed images were shown on a monochrome monitor to 10 unbiased viewers who have no image processing background. All ten viewers had trouble determining which images were "better". When told what to look for and with the aid of a magnifying glass they were able to see the blurring of the edges in the 4x4 processed image. This is especially evident in Fig.4.1.3b towards the center and those lines which form a 45^{o} angle with the base of the image where high frequencies are required in the image. Four of the viewers could see "some" degradation of the 8x8 transform image and none could see any degradation of the 16x16 image. Judging from

the response of the viewer and the knowledge that typical scenes do not have such high frequency content as Fig.4.1.3a, we belive that the degradation of the luminance signal when separated using the Hadamard transform will not be noticeable. In bandwidth compression, where some degradation is expected, this technique of separating the luminance signal should prove more than adequate. To see the affects on the chrominance signals, we perform the component separation in the Hadamard domain, as outlined below, and reconstruct the NTSC composite image for viewing.

The procedure to recover the baseband component signal from the Hadamard coefficents is given below:

1) Set to zero those coefficients which do not correspond to the particular signal being recovered.
2) If either of the chrominance signals is being recovered, multiply the coefficients by 2 and remap the coefficients by the inverse operation of Eq.(3.1.17).
3). Perform the inverse Hadamard transform to obtain the recovered baseband signal.

The signal-to-noise ratio (SNR) of the recovered signal is given by [4]:

$$SNR = -10 \log_{10} \left[\frac{\frac{1}{N} \sum_{j=1}^{N} (S_j - \bar{S}_j)^2}{255^2} \right] \qquad (4.1.2)$$

where: S = Original baseband signal.

$\bar{S}$ = Recovered baseband signal.

N = Total number of samples in image.

The resulting SNR obtained using the Hadamard transform are presented in TABLE 4.1.3. We notice that the component separation generally improves as the transform size increases. This results because as the window increases in size less high frequency energy is generated from convolving the signal and window spectrum.

Fig.4.1.4a shows the original BOY image and the processed image using a 4x4, 8x8, and 16x16 Hadamard transform. The processed images were obtained by properly sampling the original NTSC composite signal; taking the appropriate size Hadamard transform; performing the component separation described above; and then using the separated components to generate the composite images presented in Figs.4.1.4b-d.

The processed images were viewed on a SONY Trinitron monitor model CVM-1750. No degradation was noticeable in any of the processed images. No difference in color content was detected. The processed colors remained saturated and showed no streaking or bleeding.

4.2 MOTION DETECTION, ESTIMATION AND COMPENSATION SIMULATION RESULTS

The simulation results presented in this study were obtained using a sequence of 20 frames of a man speaking in a video conferencing enviroment (i.e. limited motion, uniform background and plain appropriate clothing) and a 15 frames sequence of a moving cube (the motion was translational and rotational). The sequence of the man speaking presents motion in the area of the head and the arm, which represent about 20 - 30%of the image. The second sequence presents motion in the area of the cube and its pedestal, which cover about 50 - 60% of the image.

The R, G and B video signals sampled at four times the color subcarrier frequency were used to generate the NTSC composite signal with the proper sampling phase as described in section 4.1. An 8x8 Hadamard transform is applied to the resulting NTSC composite image. The Hadamard transform of the image is then used in the motion detection, estimation and compensation algorithm. The transforms of the consecutive video frames were stored on the digital disk of a VAX-11/750 computer.

Statistical measurements were made on the frame to frame square differences of the Hadamard transform coefficients in corresponding blocks. The measurement of

the number of square differences corresponding to different threshold settings as presented in Fig.3.2.2 were used in determining the thresholds for motion detection and estimation. The large number of square differences less than 5.0 in the case of coefficient (0,0), 2.0 in the case of coefficient (7,4) and 1.0 in the case of coefficient (7,3) can result from lack of motion and the effect of background noise. The values of differences larger than the above mentioned values of the thresholds are a characteristic of moving edges and uncovered background. This can be explained by the fact that edges are characterized by abrupt changes in the values of the luminance and chrominance signals. As the edges move the average values change significantly and therefore large frame to frame differences are characteristic. Large frame to frame differences due to uncovered background can be explained in a similar manner. Therefore, the thresholds for motion detection were chosen such that larger changes are detected (mainly moving edges and uncovered background). The thresholds for motion estimation were chosen such that only small differences between the coefficients would be below the thresholds and therefore only very similar blocks would be chosen as matching. The three choices for the motion detection threshold were obtained by analysing the curves of the number of frame to frame square coefficient differences shown in Fig.3.2.2.

The behaviour of the three curves is similar and therefore, instead of chosing three different motion detection thresholds, one single motion detection threshold for all the three DC coefficients was chosen. The plot indicates that a reasonable number of frame to frame squared coefficient differences can be found above the motion detection threshold values. On the other hand the motion estimation thresholds have different values for each DC coefficient because an "accurate" match has to be found. Simulations were performed using only one value of the motion detection threshold for all the three DC coefficients (the values simulated were: 5.0, 20.0 and 35.0). The motion estimation thresholds values were set at 5.0 for the (0,0) coefficient, 2.0 for the (7,4) coefficient and 1.0 for the (7,3) coefficient. The number of blocks detected as changed between the odd fields of two successive frames versus the number of frame difference for the three motion detection threshold values are plotted in Fig.4.2.1. The number of "changed" blocks is larger when the motion is more "violent" and is less whenever little motion is detected.

The algorithm was applied to the active portion of the video signal and excluded the synchronization signals. This represents 1325 active blocks of video per field. Only the odd fields in each video frame were used in the process of

motion detection, estimation and compensation. The coefficients in each transform block are thresholded in order to detect motion by applying each block to a change detector. Based upon the result of this thresholding process a change map is generated as shown in Fig.4.2.2 and Fig.4.2.3. In the case of the sequence of the man talking the change map shows that an average of 20% of the blocks are recognized as moved for a motion detection threshold of 5.0. For motion detection thresholds of 20.0 and 35.0 the percentage of the blocks recognized as moved decreased to 8% and 4.5% of the total number of active video blocks. This decrease in the number of "changed" blocks is due to the fact that less square frame to frame coefficient differences are larger than the threshold value as the this value increases. In the case of the sequence representing a cube undergoing translational and rotational motion the number of blocks classified as moved represent 48.42% of the total number of blocks for a motion detection threshold of 5.0, 26.03% for a threshold of 20.0 and 19.45% for a threshold of 35.0. The higher percentage of blocks classified as moved for the moving cube sequence is due to the higher activity in the video frames. The moving cube and its pedestal, which also moves, represent approximately 2/3 of the video frame.

The change maps are used in the process of motion estimation where the blocks classified as changed are

checked for a match in the previous frame. The search for a match in the transform of the previous frame is performed as shown in Fig.4.2.4. The result of this search will be either a matched block, in which case the value of the displacement vector will encoded, or no match for the changed block. In the later case the frame to frame coefficient differences are encoded and transmitted to the receiver. Approximately 10% to 25% of the blocks recognized as moved, for each threshold, are matched for the sequence of the man speaking while 20% to 42.57% of the "changed" blocks are matched for the sequence representing the moving cube. The number of matched blocks is directly proportional to the number of blocks classified as changed and therefore inversely proportional to the motion detection threshold values. The percentage of matched blocks can be increased if the motion estimation thresholds are increased, but errors will appear in the reconstructed video frame. These errors can be minimized or completly removed when the upgrading of the video frame takes place.

Coefficient difference bit assignment is done as described in the previous chapter. Since the video frame is reconstructed at the receiver by repeating the received field (2:1 temporal subsampling), the total number of bits necessary to transmit the video frame is the total number of bits necessary to transmit the field. The total number

of bits TNBIT necessary to transmit one field is obtained by summing the number of bits generated by the four modes of operation of the bit assignment algorithm. The total number of bits generated are BRN, BC BA and BU bits for processing Mode 1, Mode 2, Mode3 and Mode 4 respectively. Therefore the total number of bits TNBIT is given by:

$$TNBIT = BRN + BC + BA + BU \qquad (4.2.1)$$

The plots of the total number of bits required to transmit a field versus the number of frame differences, for each sequence as a function of threshold values employed, are shown in Fig.4.2.5 and Fig.4.2.6 for 2:1 temporal subsampling. The processing of the first sequence, that of a man speaking, produced bit rates of 54 Kbps (0.0068 bits/pel), 84 Kbps (0.01 bits/pel) and 240 Kbps (0.0274 bits/pel) for motion detection threshold values of 35.0, 20.0 and 5.0 respectively. Processing of the sequence of the moving cube in the same fashion produced bit rates of 165 Kbps (0.021 bits/pel), 216 Kbps (0.0274 bits/pipel) and 318 Kbps (0.404 bits/pel) for motion detection threshold values of 35.0, 20.0 and 5.0. The higher bit rates obtained for the moving cube sequence are explained by the amount of motion undergone by the cube. The speed at which the cube was moving translationally was 60mm/sec. The cube was rotated 90^{o}/sec. The bit rate can be reduced even more if

further temporal subsampling combined with frame repetition is employed. This can be shown by the plots of the number of bits necessary to transmit a field versus the number of frame difference for 4:1 and 8:1 temporal subsampling presented in Fig.4.2.7 and Fig.4.2.8. The tradeoff is a lower SNR.

The plots of the SNR versus the number of frame difference, for each threshold, for the two sequences, are shown in Fig.4.2.9 and Fig.4.2.10. The plots can be compared to the plot of the SNR versus the number of the frame difference if the previous frame is repeated, i.e. if frame 1 is displayed instead of frame 2, frame 2 is displayed instead of frame 3 etc. The plots of the error/pel versus the number of frame difference for the two sequences and the three motion detection thresholds are shown in Fig.4.2.11 and Fig.4.2.12. Comparing the plots one can easily notice that the results are better when motion compensation is employed. The error/pel is smaller and the SNR is higher when motion compensation is employed. If the amount of motion is increased, the picture tends to break-up if no larger motion detection thresholds and further temporal subsampling are employed. This is due to the fact that the number of bits necessary to encode the information is too large and the output buffer rate is low resulting in the output buffer going into an overflow condition. It

is also worth noticing that the quality of the reconstructed video frames is poorer when 4:1 or 8:1 temporal subsampling was employed. This is explained by the fact that the video images have to be repeated and flicker will result if there are more than two repetitions. But this observation should not come as a surprise because the lower the bit rate, the poorer the quality of the reconstructed video frames.

The pictures of the error frames also prove the superior performance of the motion compensation algorithm. The white areas in these frames represent the errors. The number of errors when no motion compensation is employed is noticeable larger then in the case when motion compensation is employed as shown in Fig.4.2.13 through Fig.4.2.16. Most of the errors occur at the moving edges, as expected, but these errors are eliminated when the algorithm works in Mode 4 of operation, i.e. the picture upgrading mode. The results of this mode of operation of the algorithm is presented in Fig.4.2.17 through Fig.4.2.19 respectively. The quality of the reconstructed video frames is noticeable better after Mode 4 of operation is employed.

Pictures of a processed frame, for the sequence of a man speaking are also presented in Fig.4.2.20a) through Fig.4.2.20j). Fig.4.2.20a) represents the original image. Fig.4.2.20b) represents the same image after reconstruction. The motion detection threshold employed to

detect the motion between the previous iamge and the present one was 35.0 and was applied to the square differences of the three DC coefficients. The motion estimation thresholds employed had the values of 5.0 for the square differences corresponding to the (0,0) coefficient, 2.0 for the square differences corresponding to the (7,4) coefficient and 1.0 for the square differences corresponding to the (7,3) coefficient. A staircase effect at the slanted edges is slightly visible. This distortion can be eliminated by updating all the transform coefficients whenever there is a minimal amount of motion detected, i.e. Mode 4 of operation.

Fig.4.2.20c) represents the reconstruction of the same frame when all three DC coefficients were used for motion detection but only the luminance coefficient was used to estimate the motion. The same motion detection threshold of 35.0 is used in this case and the motion estimation threshold applied to the square differences corresponding to the (0,0) coefficient is 5.0. The quality of the reconstructed image is noticeably poorer than the quality of the image shown in Fig.4.2.20b). Blotches are present in places were matching error occured. These errors occured due to the fact that the chrominance was not taken into consideration in the matching process and therefore, the "matching" blocks matched only the luminance and not the

chrominance. These errors could be eliminated if a difference value for each coefficient would be transmitted together with the motion vector, but this will increase the number of bits to be transmitted considerably.

Figure 4.2.20d) represents the reconstruction of the same image when the motion detection threshold is applied only to the DC coefficient corresponding to luminance signal, i.e. the (0,0) coefficient. The threshold for motion estimation is also applied to the square differences corresponding to the (0,0) coefficient. Blotches are again present in the picture and the remedy is the one suggested above for Fig.4.2.20c) at the expense of a higher bit rate. The quality of this reconstructed image is poorer than that of the image presented in Fig.4.2.20b).

Figures 4.2.20e)-4.2.20g) represent the corresponding reconstructed images of those presented in Fig.4.2.20b)-Fig.4.2.20d) except that the motion detection threshold value was reduced to 20.0. More blotches appear in the reconstructed images but the quality of the reconstructed image, when all three components were considered in the process of motion detection and estimation, is clearly better than the quality of the other two reconstructed images.

The last set of pictures presented in Fig.4.2.20h)-Fig.4.2.20j) represent the same algorithm as above except that a threshold of 5.0 was employed for motion detection

while the motion estimation thresholds remain the same as those used in Fig.4.2.20b)-Fig.4.2.20g). The quality of the image reconstructed by using all three components in the process of motion detection and estimation is again better than the quality of the other two reconstructions.

The blotches due to errors in each frame are less visible or not visible at all when the sequence is looked as an ensemble, but they decrease the quality of the picture if they are located in the same position for more frames. It is also important to notice that the quality of the processed frames is considerably better when all three components are considered in the process of motion detection and estimation.

The same observations can be made about the second sequence. The quality of the video frames is even better when the upgrading of the video images is employed during periods of very little or no motion at all and output buffer emptiness. These conditions are met whenever the percentage of "changed" blocks is lower than 4% of the total active video blocks and the output buffer is 2% full. Pictures of a reconstructed video frame are presented in Fig.4.2.17 through Fig.4.2.19. A blocking effect is visible when only 10 coefficients are transmitted. This is the result of eliminating the higher frequency coefficients. The improvement in video frame quality is noticeable as

more of the coefficients are transmitted. The reconstructed video frame, when all 64 coefficients/transform block are upgraded, is identical to the original frame.

Another important aspect of the algorithm is the relationship between the bit rate at the input of the output buffer and the channel bit rate. This relationship is controlled by the threshold setting. Plots of the time in which the output buffer fills or empties out when a certain coder bit rate, channel bit rate and output buffer size (128 Kbytes and 64 Kbytes) are employed are shown in Fig.4.2.21 through 4.2.24. The plots indicate that the algorithm can be employed for channel bit rates less than 384 Kbits/sec. The plots for the sequence of the man speaking indicate that by varying the motion detection threshold from 5.0 to 35.0 the coder bit rate is reduced from 215 kbits/sec to 54 kbits/sec. This operation will allow the buffer to empty and therefore the algorithm can operate in the upgrading mode. The plots for the second sequence, that of a cube undergoing translational and rotational motion, indicate that the coder bit rate can be lowered from 318 Kbits/sec to 166 Kbits/sec by varying the motion detection threshold from 5.0 to 35.0. The coder bit rate can be also lowered by subsampling in time and increasing the motion detection thresholds.

These operations have the disadvantage of lowering the SNR. But this is the tradeoff in the low bit rate systems.

If channel bit rates of 44 Mbps or 32 Mbps are employed then 5.594 bpp or 4.069 bpp are reqired. The algorithm presented in this study requires between 0.0068bpp and 0.0404 bpp. This represents compression ratios of 1/128 to 1/32. The corresponding SNR indicate that the picture quality is better at lower compression rates than at higher ones, as expected. But the picture quality, at any of the compressions ratios achieved, is acceptable for video conferencing.

4.3 CONCLUSIONS

A technique by which component separation of NTSC composite signal can be achieved in the Hadamard domain when the sampling rate is four times the color subcarrier was firstly presented in this research study. The only constraint is the sampling phase relative to the color subcarrier which poses no problem in implementation. This technique does not require the additional equipment that is normally needed to demodulate the component signals prior to transform coding. If coding for bandwidth compression is to be employed then the bit allocation can be done directly to the transform matricies, transmitted, and at the receiver perform an inverse transformation to reconstruct the composite signal. The quality of the images used for components separation have demonstrated that we should expect no noticeable degradation in the reconstruction of typical images. This technique can serve as a possible comb filtering technique for future digital television systems.

Secondly, this study presented a motion detection, estimation and compensation algorithm which operates on the component signals in the Hadamard transform domain and can be used for low bit rate video conferencing systems. The algorithm is based on the separation of the NTSC composite signal into the Y,I,Q component signal when Hadamard

transformed. The mapping of the DC energies in the Hadamard domain is used in the motion compensation algorithm. Only the coefficients containing the DC energies are thresholded for motion detection and then for motion estimation. This reduces the number of computations required in the motion compensation process.

Bandwidth compression was achieved by combining the two algorithms, the one which separates the NTSC composite signal into its component signals in the Hadamard transform domain and the motion compensation algorithm. The original part of the algorithm presented in this research study consists in taking advantage of the properties of the Hadamard transform to detect, estimate and compensate motion in the transform domain by using all three components of the NTSC signal. This algorithm has the flexibility of operating on the component signals Y, I, Q and in the same time is simple because it uses only one transform matrix for all three components and not three different ones.

Bandwidth compression ratios of 1/128 to 1/32 were achieved by using this algorithm. This makes the algorithm attractive for low bit rate systems. The coder bit rates obtained prove the feasibility of this algorithm for low bit rate video conferencing systems. The quality of the reconstructed video frames prove that not only the

algorithm achives high compression ratios, but it also achieves acceptable video conferencing picture quality.

The originality of this algorithm consists of the separation technique and of the motion compensation technique, which operates on the Y, I, Q signals in the Hadamard domain.

5. FUTURE RESEARCH

Several features can be added to the basic algorithm which would increase its capabilities. One of these is the use non uniform quantizers based not on objective criteria but on subjective ones. These will allow a more efficient distribution of the quantization error and therefore a better picture quality at lower bit rates.

Another feature which can be added to the algorithm is an optimum code word assignment. This will reduce the number of bits necessary to encode the information and therefore reduce the bit rate.

A serious thought has to be given to developing an efficient search algorithm for finding the "perfect match" for the blocks which, in the data (pel) domain, underwent displacements smaller than the block size. This feature will increase the efficiency of the algorithm by matching more blocks and therefore requiring less bits to transmit the changes in the video frames.

Much lower bit rates can be achieved if more is known about the psychophysics of the human vision. The imperfections of the human vision can be taken advantage of in the process of encoding the information. This will result in lower bit rates for the same picture quality. A modeling of the video sources would also be possible and would result in lower bit rates.

6. APPENDIX

APPENDIX A

A SHORT DESCRIPTION OF THE NTSC SIGNAL

There are three basic color television systems which are used world wide. These are the NTSC system, the PAL system and the SECAM system. The system we are using in the United States is the NTSC system. Here we present a short description of the NTSC signal.

Color television systems are based on the experimental fact that a great many colors can be created by appropiate combinations of three primary colors. Any three independent colors can be chosen as primary colors. The most popular choice of red, green and blue as the primary colors is based on the experimental results that this set of primary colors can generate a wider range of colors. Colorimetry is based on two premises: 1) color is a three-dimensional property of light; and 2) the amount of three primary lights (colors) needed to match an unknown color may be used as numerical dimensions to specify the color. The three dimensions required to specify a color are hue, saturation and brightness. Hue is the characteristic by which colors can be separated into groups such as red, yellow, purple, etc. Saturation refers to the degree of

purity of a color, as to its lack of delution by white light. Low saturated colors are "pale" while highly saturated colors are vivid or "strong". Brightness refers to the degree of brightness or darkness of a color, or to its position in a scale from black to white. Hue, saturation and brightness are not completely independent. Hues tend to shift slightly with variations in brightness and saturation and it is often difficult to be sure whether two colors are different in brightness or in saturation. The human eye accuity or resolving power is consideably less for differences in hue and saturation than for differences in brightness. The resolving power of the eye for chromatic differences varies for various color combinations. The color accuity of the eye is generally greater for color differences in the yellow-green-to-purple color region and considerably less for color differences in the orange to cyan region.

In 1941 the first National Television System Committee (NTSC) recommended the 6 MHz bandwidth requirements of the monochrome television system used widely in the United States since 1946. When the color television system had to be defined, one of the requirements was that the bandwidth of the color TV signal to be transmitted match the bandwidth used by the monochrome systems. Since the R, G and B color signals have a bandwidth of about 4 MHz each, other signals had to be used to transmit the information

embeded in the R, G and B signals. These new signals are the Y, I and Q. The reason for selecting these signals are the followings: 1) the Y signal, which represents the luminance signal was already used to transmit the monochrome TV signal; 2) the I and Q signals represent the chrominance signals and they can be bandlimited because the eye is unable to perceive color in small area whereas the color information for large areas have to be transmitted. It turns out that the resolving power of the eye is stronger along the I axis (one could represent the colors in a system of coordinates with the axis I and Q) than along the Q axis. Therefore, the magnitude of the I coordinates is bigger than the magnitude of the Q coordinate. Since the eye is insensitive to color changes for higher frequency the spectra of the two chrominance signals (after they undergo a process of in quadrature amplitude modulation) can be bandlimited and used together with the spactrum of the Y signal in an interleaved manner such that the bandwidth of the composite signal matches the bandwidth of the monochrome signal (as shown in Fig.A.1).

The transformation matrix used to obtain the Y, I and Q signals from the R, G and B signals has to be such that the resultant component I and Q points in the proper direction in the color space and when R, G, B are 100% the luminance signal Y is 100% and the I and Q are 0%. The

transformation using this matrix is given below:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.569 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (A.1)$$

The composite signal obtained by using the Y, I and Q signals has to contain information about the hue, saturation and brightness. The hue information is specified by the phase of a sinusoidal superimposed (subcarrier) on the luminance signal relative to the burst (which is a reference sinusoidal waveform superimposed on the horizontal blanking interval). The amplitude of the subcarrier is used to specify the saturation. The average amplitude of the signal represents the brightness. The brightness is related to the frequency of the subcarrier, which was chosen to be an odd integer multiple of half the horizontal line frequency (f_h = 15.75 KHz) as given by equation (A.3). Because of the choice made for the frequency of the subcarrier, the subcarrier changes the phase by 180° from line to line and the eye by its integrating property is able to extract the brightness information contained in the signal.

The NTSC composite signal has the following expression:

$$f(t) = Y(t) + I(t)\cos(2\pi f_{sc}t + 33^\circ) + Q(t)\sin(2\pi f_{sc}t + 33^\circ) \qquad (A.2)$$

The frequency of the color subcarrier is:

$$f_{sc} = (227+1/2)f_h = 3.579 \text{ MHz} \qquad (A.3)$$

APPENDIX B

ENTROPY AND ITS INFLUENCE ON THE BIT-RATE

The quantity used to measure the amount of information contained in an image is the average information, which is also referred to as entropy. If event x=X occurs with probability p(x), where x is an arbitrary element of the image and X is a particular pixel value, then the self-information of that occurance is

$$i(x) = \log[1/p(x)] = -\log p(x) \tag{B.1}$$

The base of the logarithm is usually base 2.

The next step is to average the self-information per picture element over the entire image. This yields the entropy, H(x), [4,8,62,63].

$$H(x) = -\sum p(x)\log_2 p(x) \tag{B.2}$$

$$H(x) = -\sum p(x)\log_2 p(x)$$

This definition of the entropy can be extended to higher order statistics as well, [4,8,62,63]. The entropy has a direct bearing on the bit rate necessary to transmit an image or a sequence of images. The average information

represents a lower bound for the bit rate, which means that it gives the minimum number of bits necessary to transmit the information contained in the image, such that the image can be reconstructed without any distortions. In spite of this, the bit rate can be lowered beyond the limit given by the entropy, but in this case the reconstructed picture presents distortions.

Hence, once the amount of average information to be transmitted is known, the task of deciding which coding technique to employ becomes easier.

APPENDIX C

LINEAR PREDICTION

Linear predictors are used to generate an estimate of the present pel by a linear combination of previous transmitted pels. The spatial/temporal location of the previous pels used in the weighted sum determine the type of predictor, i.e. one-dimensional, two-dimensional or three-dimensional. Predictors attempt to exploit the correlation property of pels within close proximity of one another in order to generate an estimate of the current pel. Because of the low cost of memory, the difference in terms of implementation, between the predictors became less important.

Assuming that we have a block of pels (b_1, b_2, ..., b_N) and that pels b_1, b_2, ..., b_{N-1} have already been transmitted, a linear predictor for b_N can be written as:

$$\hat{b}_N = \sum_{i=1}^{N-1} \alpha_i \, b_{N-1} \qquad (C.1)$$

The coefficients (α_i) can be found by minimizing the mean square error (mse), $E(b_N - \hat{b}_N)^2$. A minimum mean square linear predictor can be obtained by differentiating $E(b_N - \hat{b}_N)^2$ with respect to each α_j and setting the result to

zero. This yields:

$$-2E\left[b_N b_j - \sum_{i=1}^{N-1} \alpha_i b_i b_j \right] = 0, \quad j=1,\ldots,N \tag{C.2}$$

Making the following notations:

$$d_j = E(b_N b_j) \tag{C.3}$$

$$r_{ij} = E(b_i b_j) \tag{C.4}$$

Where d and α are column matrices and r is a square matrix, the equation that gives the minimum mean square linear predictor becomes:

$$d - r \quad = 0 \tag{C.5}$$

The square matrix r is the correlation matrix of the pels $b_1,\ldots, b_{N-1}$. Therefore r_{ij} is the correlation between pels b_i and b_j.

If the correlation matrix is nonsingular, then,

$$\alpha = r^{-1} d \tag{C.6}$$

is the unique solution of the equation

$$d - r \quad = 0 \tag{C.7}$$

If the correlation matrix is singular then the above

equation has many solutions each having the same mean square prediction error [2].

APPENDIX D

TRANSFORM CODING

In transform coding one attempts to decorrelate the pels by using a suitable transformation. The resulting transform coefficients obtained are less correlated than the sampled pels and can be quantized and transmitted separately. Orthogonal transforms have been studied in detail and are generally used in image coding. In order to transform a pel column vector **b** of length N, we multiply the vector by a suitable transform and obtain the coefficient vector **c**.

$$\mathbf{c} = \mathbf{T}\mathbf{b} \tag{D.1}$$

For orthogonal and linear transforms the inverse is equal to the transpose:

$$\mathbf{T}^{-1} = \mathbf{T}^{T} \tag{D.2}$$

The original data vector **b** can be obtained by:

$$\mathbf{b} = \mathbf{T}^{T}\mathbf{c} \tag{D.3}$$

If the m^{th} column of the matrix **T'** is denoted by t_m, then the equality

$$T^{-1} = T^T \tag{D.4}$$

becomes

$$\langle t^T_m t_n \rangle = \delta_{mn} \tag{D.5}$$

which is the condition for the orthonormality. Using the orthonormal basis vectors of the unitary transform **T** we can write that:

$$b = \sum_{m=1}^{N} c_m t_m \tag{D.6}$$

where **cm** is the m^{th} element of **c**.

The first basis vector t_1 which corresponds to the zero spatial frequency has the following form for practically all the transforms:

$$t^T_1 = \frac{1}{\sqrt{N}} (1, 1, 1, \ldots, 1) \tag{D.7}$$

This is the reason c_1 is denoted as the DC coefficient. If b_{max} is the largest pel value, then Nb_{max} is the upper

bound for c_1, [2].

There are three ways to build the vector b. If we want the transform coding to exploit one dimensional horizontal correlation between pels, then the vector b could be constructed from N successive pels of a raster scan line [2]. If we want to exploit two dimensional correlation between pels, that is in both vertical and horizontal direction, we have to build the vector b from a two dimensional LxL array of image pels by concatenating the L-pel columns or rows end to end. In this way we built a single column vector of length $N = L^2$ [2]. Practical two-dimensional transforms are separable, i.e. the transformation in the vertical and horizontal directions can be performed independently and the transform matrix is $L^2 x L^2$. This allows us to to create a square matrix $B = [b_{i,j}]$ which denotes the LxL array of image pels. Then the transform can be performed in two steps as shown below:

$$c_{m,n} = \sum_{i=1}^{L} \sum_{j=1}^{L} b_{i,j} t^T_{n,j} t_{m,i} \qquad (D.8)$$

where t's are the elements of the transform matrix T and the c's are the elements of the matrix of the transform coefficients. The dimension of this two matrices is LxL [2].

We shall assume that the basis vectors are indexed in increasing spatial frequencies. Based on this assumption the subscript m indexes the vertical spatial frequency and the subscript n indexes the horizontal spatial frequency. In this context f_m and f_n represent the vertical and horizontal spatial frequencies of the respective basis vectors and the spatial frequency corresponding to the coefficient c_{mn} is given by [2]:

$$f_{mn} = \sqrt{f_m^2 + f_n^2} \quad \text{cycles/unit distances} \tag{D.9}$$

If the transform is a unitary one, then, the inverse transform is given by, [2]:

$$b_{ij} = \sum_{m=1}^{L} \sum_{n=1}^{L} c_{mn} t_{nj} t^T_{mi} \tag{D.10}$$

The direct and the inverse transform can be written in the matrix form as:

$$C = TBT^T \tag{D.11}$$

$$B = T^T C T \tag{D.12}$$

If one employes a separable transformation, the number of multiplication and additions required is $2L^3$ [2]. On the other hand, if the column transformation is used the number of multiplication and addition required is L^4. One could

very easilly see the advantage of using separable transformations over the column transformation [2].

If the L^{th} order transform **T** has $\{t_m\}$ as its basis vectors, then **B** can be written as:

$$B = \sum_{m=1}^{L} \sum_{n=1}^{L} c_{mn} t_m t^T_n \qquad (D.13)$$

Thus, **B** can be thought of as a linear combination of the set of basis images $\{t_m t^T_n\}$; m,n = 0, ..., L-1. Unitary transforms can be classified by the set of basis vectors (or basis images) they use in the expansion. Commononly used are Karhunen-Loeve transform (KLT), Fourier transforms (DFT or FFT), discrete cosine transform (DCT) and Walsh-Hadamard transform (WHT).

KARHUNEN-LOEVE TRANSFORM

KLT matrix ^{T} is obtained from the NxN correlation matrix of the image pel blocks **b** which have been biased to have zero mean. This correlation matrix is given by the expected value:

$$R = E\{(b - E(b))(b - E(b))\} \qquad (D.15)$$

In terms of the elements of R and b we have:

$$r(i,j) = E[(b_i - E(b_i))(b_j - E(b_j))] \qquad (D.16)$$

The basis vectors of the KLT are the real orthonormalized eigenvectors of R, i.e.,

$$Rt_m = \lambda_m t_m, \tag{D.17}$$

$$t^T_m t_n = \delta_{mn} \tag{D.18}$$

The eigenvectors are arranged according to decreasing eigenvalue, which corresponds to ordering according to increasing frequency.

The KL transformation is given by:

$$\mathbf{c} = \mathbf{Tb} \tag{D.19}$$

and

$$E(c_m c_n) = \lambda_m \delta_{mn} \tag{D.20}$$

Therefore, we may conclude that the KLT coefficients are uncorrelated, and their mean square values are equal to their respective eigenvalues. If the KLT basis vectors are ordered according to the decreasing eigenvalues a compaction of the energy into the lower index coefficients will result. With the mean square error distortion criterion, the KLT achieves the best energy compaction of any linear transform. Since image statistics are highly nonstationary, using the KL transform is computationally intensive. Thus, orthonormal suboptimum (fixed basis vectors) transforms are generally used.

DISCRETE FOURIER TRANSFORM

If T is the matrix which represents the NxN unitary, symmetric Discrete Fourier Transform, then its elements are given by:

$$t_{mi} = \frac{1}{N} \exp\left[- \frac{2\pi}{N} \sqrt{-1}\ (i-1)(m-1)\right] \qquad (D.21)$$

$$i,\ m = 1, \ldots,\ N$$

If **b** is real, then the complex coefficients of **c** present a conjugate symmetry, i.e.,

$$c_{2+q} = c_{N-q}\ ,\quad 0 < q < N-2 \qquad (D.22)$$

Therefore, the transmission of at most N real values is required in order to reconstruct the N pels of **b**.

The DFT basis vectors t_m are complex sinusoids. Thus, the DFT is a spectral decomposition with the energy at the frequency (m-1)/N being given by $c_m{}^2$. For most of images, the spectral energy decreases as a function of frequency. Thus, the DFT results in a compaction of energy into the lower index transform coefficients.

DISCRETE COSINE TRANSFORM

The matrix T has the elements given by:

$$t_{mi} = \sqrt{\frac{2-\delta_{m-1}}{N}} \cos\{\frac{\pi}{N}(i-\frac{1}{2})(m-1)\} \qquad (D.23)$$

$i, m = 1, \ldots, N$

The DCT basis vectors t_m are sinusoids with frequency index m. The similarity between the DCT and KLT basis vectors is in part the explanation for the good performance of the DCT compared to the other transforms.

The DCT can be computed from the FFT as follows:

(I) Take the FFT of the length 2N vector

$$\mathbf{b} = (b_1 b_2 \ldots b_N 000\ldots0). \qquad (D.24)$$

(II) The first N of the resulting Fourier coefficients are then multiplied by:

$$F_m = \sqrt{2(2-\delta_{m-1})}\exp[\frac{\pi\sqrt{-1}}{2N}(m-1)] \qquad (D.25)$$

(III) The DCT coefficients are given by taking the real part.

WALSH-HADAMARD TRANSFORM

The easiest way to describe the WHT transform is a recursive one. If we let $H_1 = 1$ and $N = 2^n$, we can define

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \tag{D.26}$$

The unitary, symmetric WHT matrix is given by:

$$T = \frac{1}{\sqrt{N}} H_N = T^T \tag{D.27}$$

The concept of frequency can be extended to the WHT, in which case it is called sequency. The sequency of a WHT basis vector t_m is defined as the number of sign changes within it devided by two. As with the DFT, the energy in the WHT coefficients for images decreases with the sequency.

The advantage of the WHT is that with the exception of the term $1/\sqrt{N}$, the computation requires only additions and subtractions as opposed to most of the other transforms which require multiplications as well. Thus, the WHT transform is easier to implement in hardware than other trnasforms. There is also a fast algorithm which require only $N\log_2 N$ operations. These properties make the WHT very attractive for image processing applications.

7. TABLES

Search Procedure	Required Number of Search Points		Required Number of Sequential Steps	
	a	b	a	b
2D-logarithmic	18	21	5	7
Three step	25	25	3	3
Conjugate direction (simplified)	12	15	9	12

a) For a special displacement vector $(i + 2, j + 6)$.
b) For a worst case situation.

Table 2.1: The comparison of some search procedures

BP64

4.00	2.00	1.00	1.00	1.00	1.00	1.00	1.00
3.00	2.00	1.00	1.00	1.00	1.00	1.00	1.00
2.00	2.00	1.00	1.00	1.00	1.00	1.00	1.00
2.00	2.00	1.00	1.00	1.00	0.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	0.00	2.00	2.00	1.00	1.00	1.00
1.00	1.00	1.00	2.00	2.00	1.00	1.00	1.00
1.00	1.00	1.00	2.00	3.00	1.00	1.00	1.00

2:1 Temporal Subsampling

BP4164

3.00	2.00	1.00	2.00	2.00	1.00	1.00	1.00
3.00	2.00	1.00	2.00	2.00	1.00	1.00	1.00
3.00	1.00	1.00	2.00	2.00	1.00	1.00	1.00
3.00	1.00	1.00	2.00	2.00	1.00	1.00	1.00
2.00	1.00	1.00	2.00	2.00	1.00	1.00	1.00
2.00	1.00	1.00	2.00	2.00	1.00	1.00	1.00
2.00	1.00	1.00	2.00	2.00	1.00	1.00	1.00
2.00	1.00	1.00	2.00	3.00	1.00	1.00	1.00

4:1 Temporal Subsampling
(Man Speaking Sequence)

BP4264

3.00	3.00	1.00	1.00	1.00	1.00	0.00	0.00
3.00	2.00	1.00	1.00	0.00	1.00	1.00	0.00
2.00	2.00	1.00	1.00	0.00	1.00	1.00	1.00
2.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00
1.00	1.00	1.00	1.00	2.00	1.00	1.00	1.00
1.00	0.00	1.00	1.00	3.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	3.00	1.00	0.00	0.00

4:1 Temporal Subsampling
(Moving Color Cube Sequence)

BP8164

3.00	2.00	1.00	2.00	1.00	1.00	1.00	1.00
3.00	1.00	1.00	2.00	1.00	1.00	1.00	1.00
2.00	1.00	1.00	2.00	2.00	1.00	1.00	1.00
2.00	0.00	1.00	1.00	2.00	1.00	1.00	1.00
2.00	0.00	1.00	1.00	2.00	1.00	1.00	1.00
2.00	0.00	1.00	2.00	3.00	1.00	1.00	1.00
2.00	1.00	1.00	2.00	3.00	1.00	1.00	1.00

8:1 Temporal Subsampling
(Man Speaking Sequence)

BP8264

3.00	2.00	0.00	1.00	0.00	1.00	1.00	0.00
2.00	1.00	0.00	1.00	1.00	0.00	0.00	1.00
2.00	1.00	0.00	1.00	1.00	1.00	1.00	0.00
0.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
1.00	0.00	1.00	1.00	1.00	0.00	1.00	1.00
1.00	1.00	1.00	1.00	3.00	1.00	1.00	1.00
1.00	1.00	1.00	2.00	3.00	1.00	0.00	1.00

8:1 Temporal Subsampling
(Moving Color Cube Sequence)

Table 3.2.1: Bit assignment matrices for Mode 2 when different temporal subsampling is employed

PP64

4,00	2,00	1,00	1,00	1,00	1,00	1,00	1,00
3,00	2,00	1,00	1,00	1,00	1,00	1,00	1,00
2,00	2,00	1,00	1,00	1,00	1,00	1,00	1,00
2,00	2,00	1,00	1,00	1,00	0,00	1,00	1,00
1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1,00	1,00	0,00	2,00	2,00	1,00	1,00	1,00
1,00	1,00	1,00	2,00	2,00	1,00	1,00	1,00
1,00	1,00	1,00	2,00	3,00	1,00	1,00	1,00

Mode 2

FD64

2,00	2,00	1,00	1,00	1,00	1,00	1,00	1,00
2,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
1,00	1,00	1,00	2,00	2,00	1,00	1,00	1,00

Mode 4

Table 3.2.2: Bit assignment matrices for Mode 2 and Mode 4

33.565	5.239	2.379	2.399	0.705	0.866	1.083	1.227
8.632	2.954	1.559	1.351	0.589	0.657	0.733	0.668
5.012	2.132	1.226	1.001	0.517	0.565	0.590	0.500
4.295	1.745	1.009	0.823	0.451	0.496	0.497	0.426
2.353	1.238	0.795	0.653	0.427	0.425	0.414	0.351
2.334	1.130	0.731	0.603	0.417	0.419	0.402	0.339
2.227	1.024	0.641	0.558	0.401	0.385	0.369	0.324
1.908	0.735	0.490	0.471	0.429	0.344	0.323	0.290

(a)

0.037	0.046	0.053	0.044	0.487	0.186	0.109	0.080
0.035	0.049	0.058	0.050	0.517	0.214	0.118	0.092
0.036	0.051	0.061	0.051	0.504	0.228	0.126	0.096
0.037	0.052	0.064	0.054	0.518	0.243	0.132	0.101
0.038	0.057	0.073	0.068	1.009	0.309	0.153	0.132
0.042	0.062	0.083	0.078	1.045	0.363	0.176	0.156
0.047	0.069	0.101	0.111	1.920	0.508	0.216	0.223
0.062	0.097	0.143	0.230	8.810	1.026	0.306	0.463

(b)

0.048	0.042	0.087	0.260	0.032	0.030	0.030	0.033
0.055	0.046	0.105	0.306	0.034	0.031	0.030	0.031
0.057	0.049	0.112	0.341	0.035	0.032	0.030	0.031
0.063	0.051	0.126	0.373	0.037	0.033	0.031	0.031
0.075	0.057	0.151	0.507	0.042	0.035	0.032	0.032
0.080	0.064	0.166	0.555	0.045	0.037	0.032	0.032
0.091	0.073	0.190	0.766	0.050	0.041	0.033	0.033
0.154	0.091	0.322	2.775	0.079	0.049	0.036	0.035

(c)

Table 4.1.1: Standard deviation of 8x8 Hadamard transform coefficients of NTSC component signals averaged over five images a) Y signals, b) Mod. I signals, c) Mod. Q signals

Matrix size	Mod. Q	Mod. I	Y
4x4	(3,1)	(3,2),(3,2)	all others
8x8	(7,2),(7,3),(6,3) (5,3)	(7,4),(7,5),(7,6) (7,7),(6,4),(6,5) (5,4),(4,4)	all others
16x16	(15,6)(15,7)(14,6) (14,7)(13,7)(12,7) (11,7)(10,7)(9,7) (8,7)	(15,8)(15,9)(15,10) (15,11)(14,8)(14,9) (14,10)(14,11)(13,8) (13,9)(13,10)(13,11) (12,8)(12,9)(11,8) (11,9)(10,8)(10,9) (9,8)(9,9)(8,8)(8,11) (7,8)(6,8)(5,8)(4,8) (3,8)	all others

Table 4.1.2: Coefficient assignment to component signals

		SMPTE1	SMPTE2	SMPTE3	SMPTE15	BOY
4X4	Y	40.54	39.84	40.80	41.95	39.13
	I	40.22	39.38	40.02	41.82	38.06
	Q	41.27	39.90	41.75	42.62	39.87
8X8	Y	41.37	40.03	41.53	42.73	39.57
	I	40.52	39.53	40.30	42.08	38.51
	Q	42.76	40.17	43.17	43.95	40.22
16X16	Y	41.26	40.63	41.62	42.38	39.57
	I	42.99	39.77	39.95	41.68	38.60
	Q	42.99	41.17	43.19	43.88	41.36

Table 4.1.3: SNR in dB

8. FIGURES

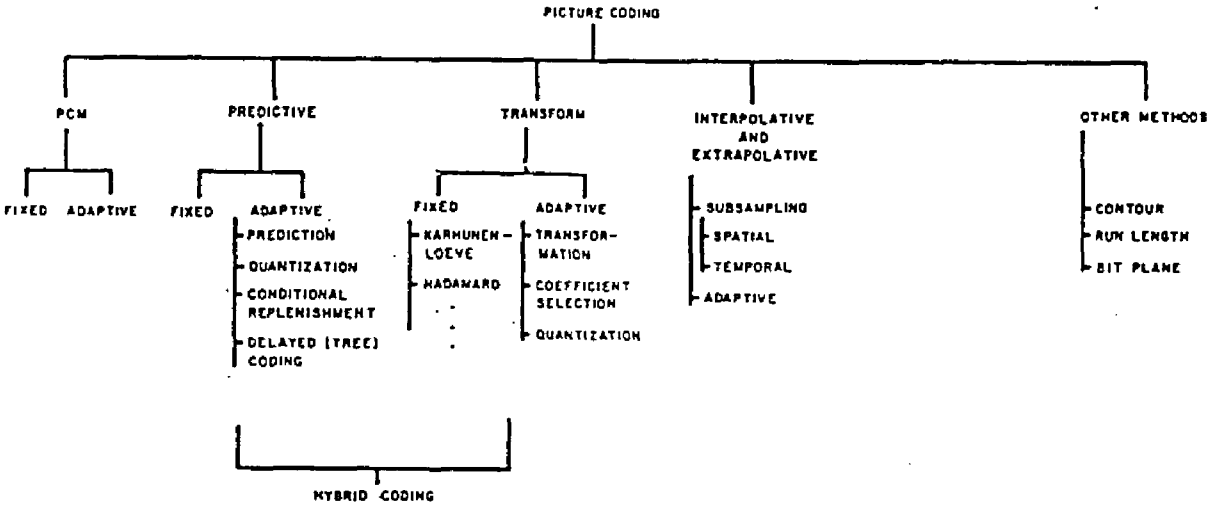


Fig.2.1: Classification of coding techniques

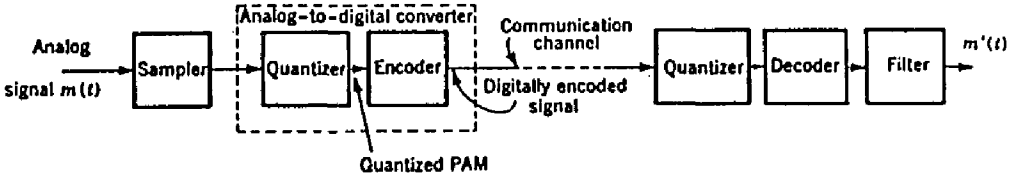
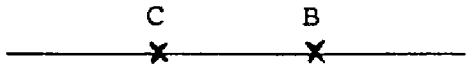


Fig.2.2: Block diagram of a PCM system



$$\hat{X} = \text{PREDICTOR FOR } X$$

$$= \begin{cases} A, \text{ IF } |B-C| \leq |A-C| \\ B, \text{ OTHERWISE} \end{cases}$$

Fig.2.3: Graham Predictor

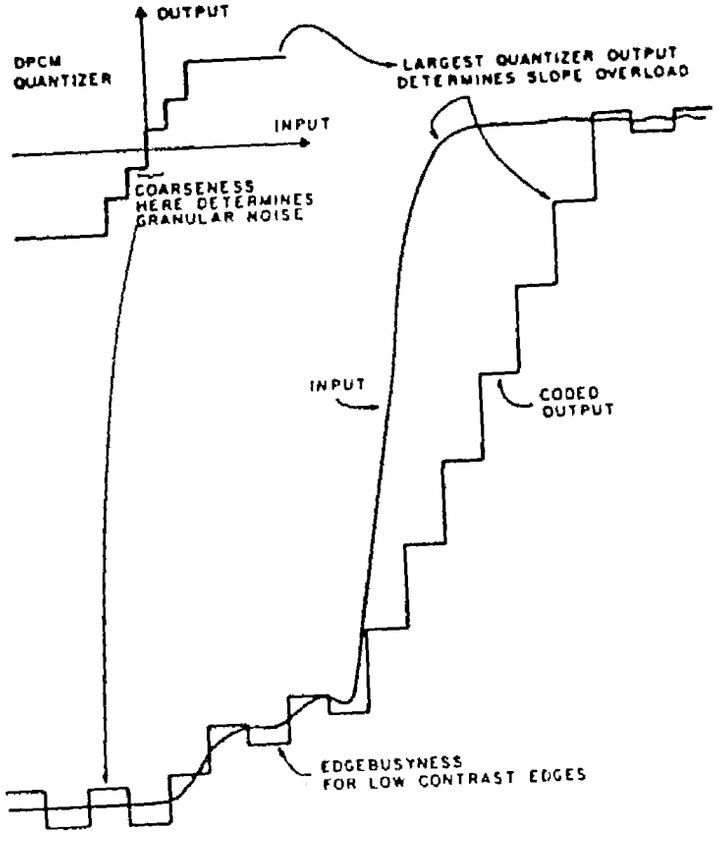


Fig.2.4: Types of degradation due to improper design of the quantizer of a DPCM coder

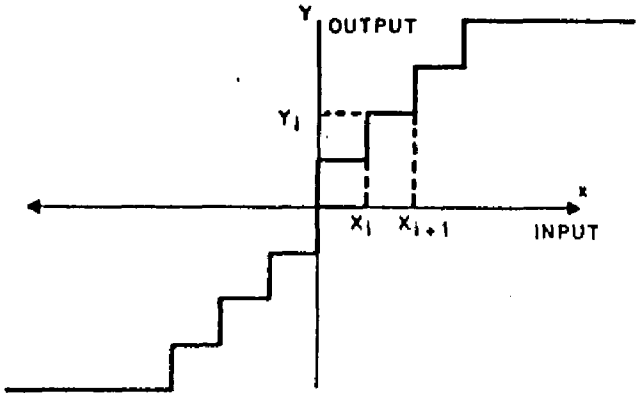
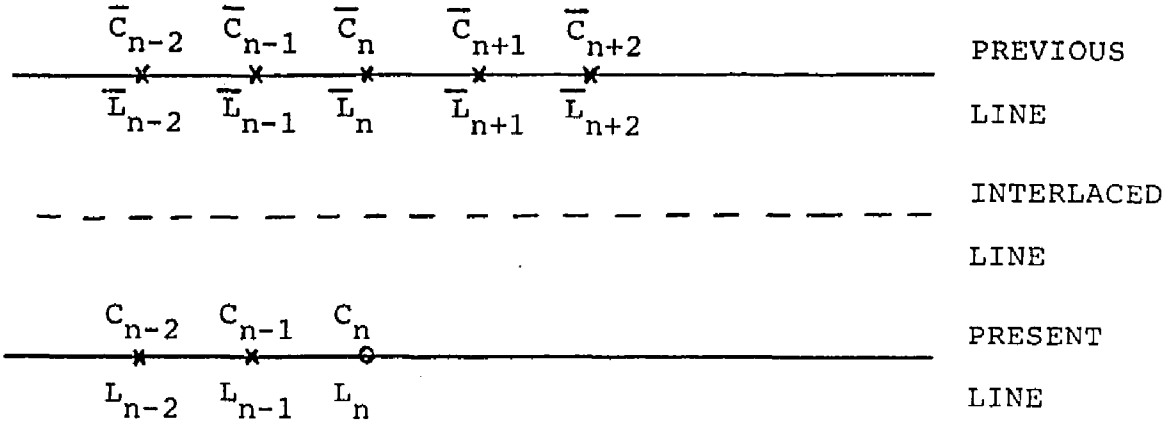


Fig.2.5: Characteristic of a quantizer



C_n = PRESENT CHROMINANCE SAMPLE

L_n = PRESENT LUMINANCE SAMPLE

Fig.2.6: Samples of luminance and chrominance signals used in component predictive coding

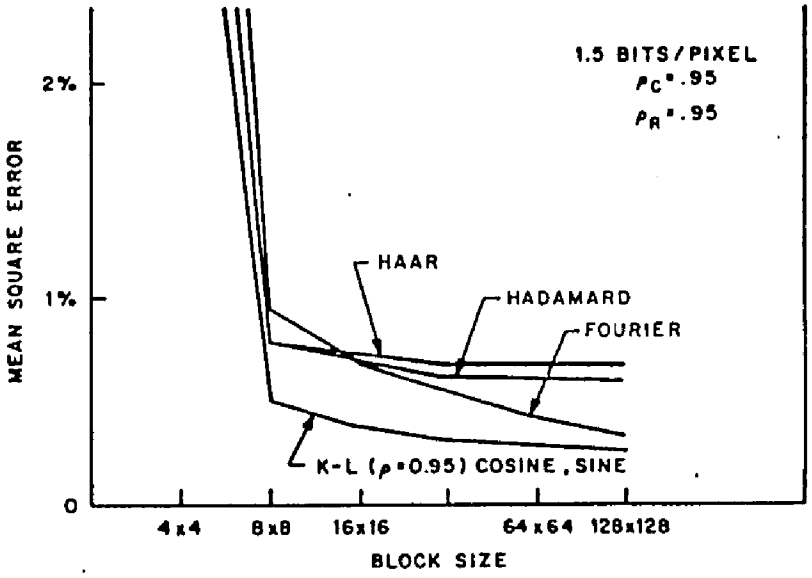


Fig.2.7: Comparison of mean square error vs. block size at a given bit rate for different transforms

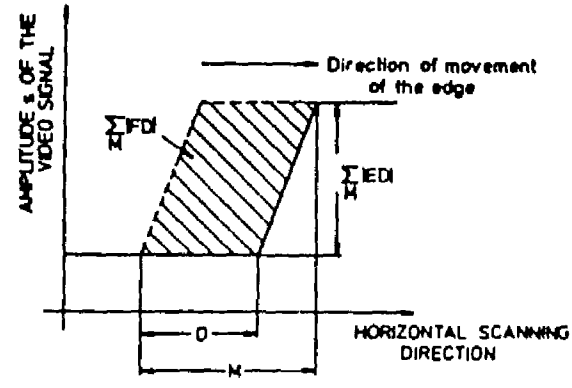


Fig.2.8: Illustration of the displacement estimation scheme proposed by Limb and Murphy

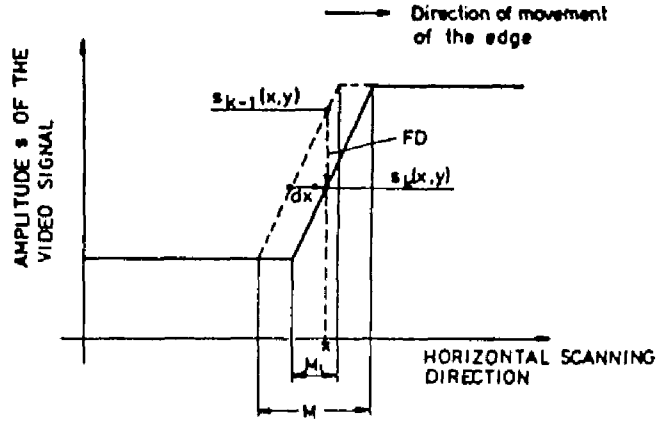


Fig.2.9: Illustration of the displacement estimation algorithm proposed by Cafforio and Rocca

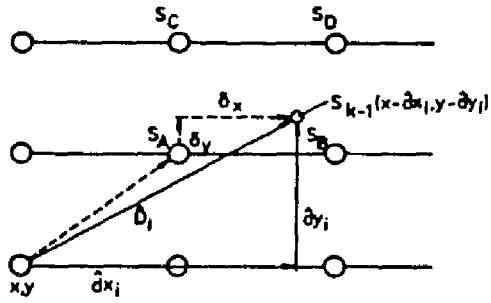


Fig.2.10: Position of picture elements in frame s_{k-1} used for interpolation of $s_{k-1}(x-\hat{dx}_i, y-\hat{dy}_i)$

Algorithm	x-Component of the Displacement Estimate $\hat{D}_{i+1}$
Netravali and Robbins	$\hat{dx}_{i+1} = \hat{dx}_i + \epsilon \frac{\partial}{\partial x} R_{s_k s_{k-1}}(x, y, \hat{D}_i), \quad \epsilon = 1/1024$
Newton–Raphson	$\hat{dx}_{i+1} = \hat{dx}_i - \frac{\frac{\partial}{\partial x} R_{s_k s_{k-1}}(x, y, \hat{D}_i)}{\frac{\partial^2}{\partial x^2} R_{s_k s_{k-1}}(x, y, \hat{D}_i)}$
Cafferio and Rocca	$\hat{dx}_{i+1} = \hat{dx}_i + \frac{\frac{\partial}{\partial x} R_{s_k s_{k-1}}(x, y, \hat{D}_i)}{\left\lvert \frac{\partial^2}{\partial x^2} R_{s_{k-1} s_{k-1}}(x, y, 0) \right\rvert + \eta^2}, \quad \eta^2 = 100$
Bergmann	$\hat{dx}_{i+1} = \hat{dx}_i - \frac{\frac{\partial}{\partial x} R_{s_k s_{k-1}}(x, y, \hat{D}_i)}{\frac{1}{2}\left[\frac{\partial^2}{\partial x^2} R_{s_k s_{k-1}}(x, y, \hat{D}_i) + \frac{\partial^2}{\partial x^2} R_{s_k s_k}(x, y, 0) \right]}$

Fig.2.11: Pel recursive displacement algorithms simplified for comparison by assuming $E[s^2_{k-1}(s,y)] = E[s^2_{k-1}(x-dx, y-dy)] = \text{constant}$

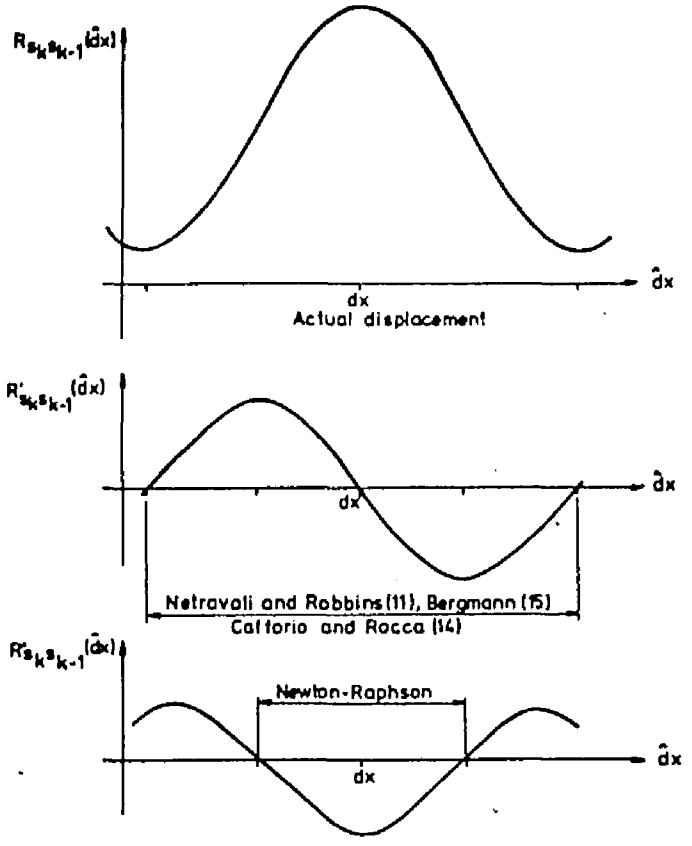


Fig.2.12: Stability ranges for various recursive estimation algorithms

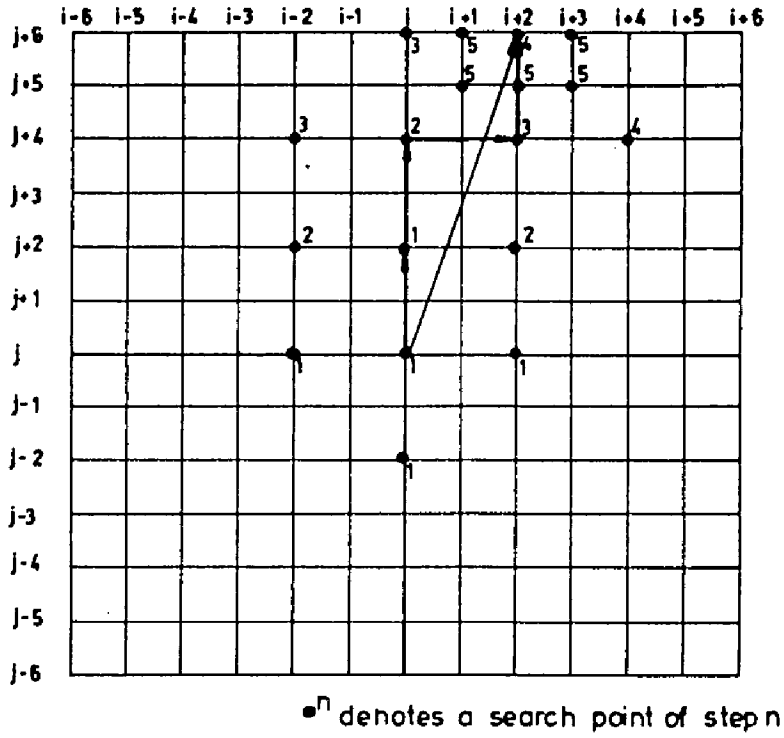


Fig.2.13: 2D-logarithmic search procedure

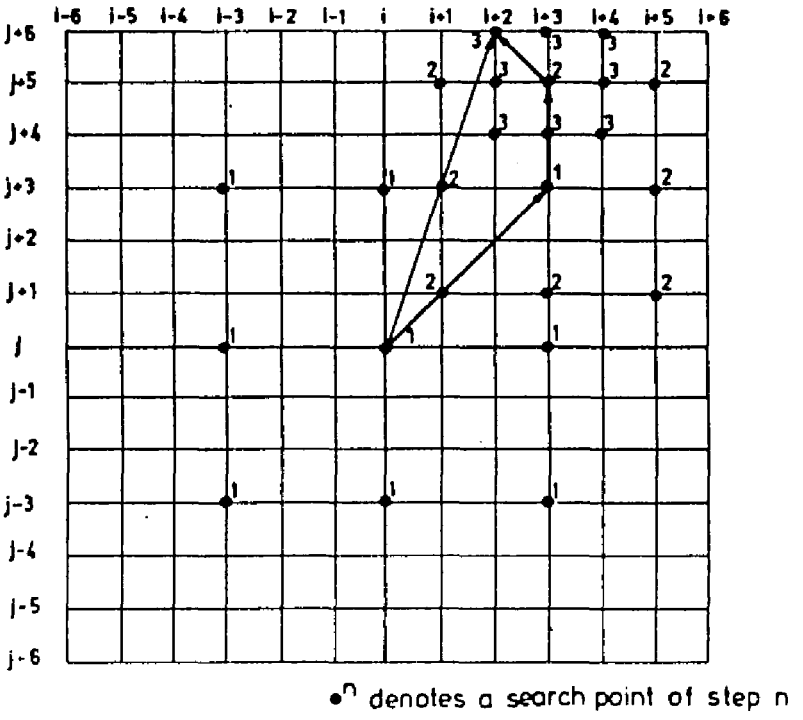


Fig.2.14: Three-step search procedure

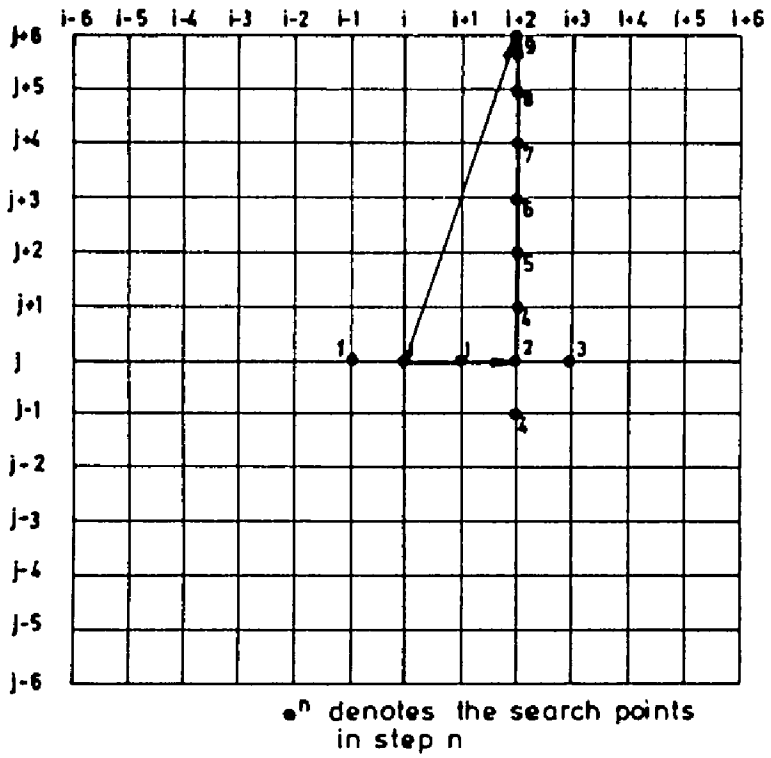


Fig.2.15: Conjugate direction search in simplified version

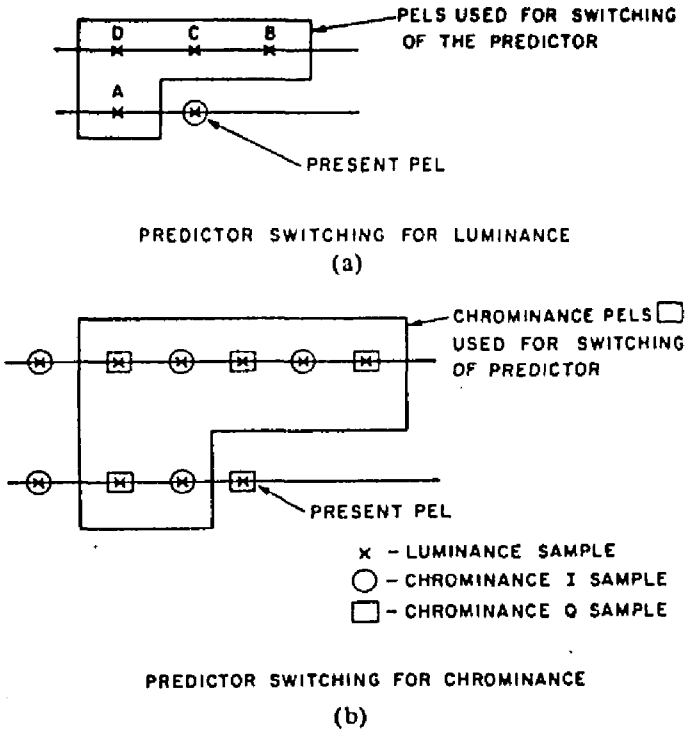
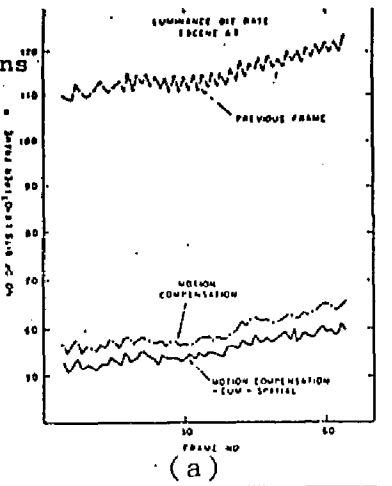
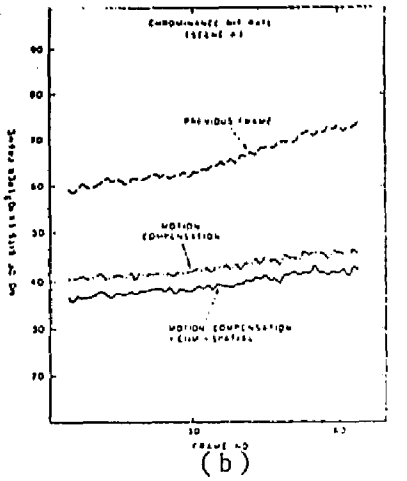


Fig.2.16: Choice of neighboring pels for prediction switching

Scene A: Two persons talking; bright colored background. Little motion.

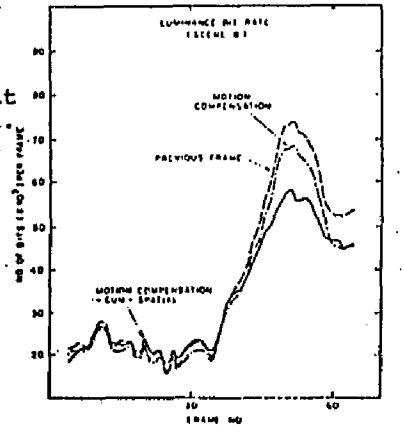


(a)

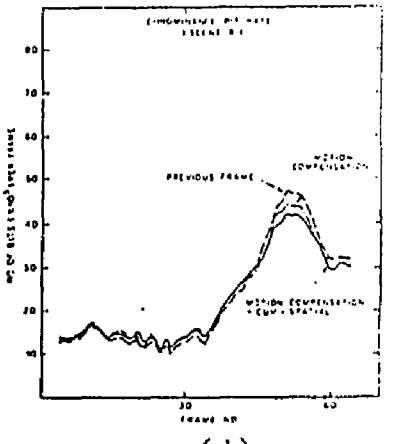


(b)

Scene B: A person with a book; bright colored background. Little to moderate motion.

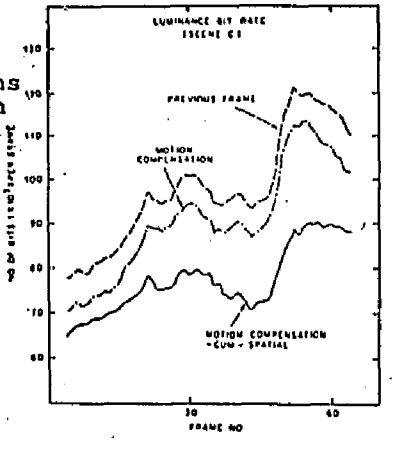


(c)

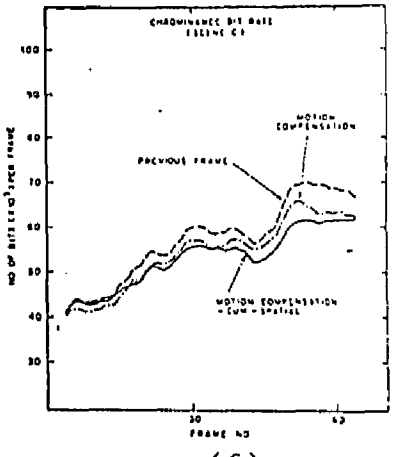


(d)

Scene C: Two persons walking around each other. High motion.

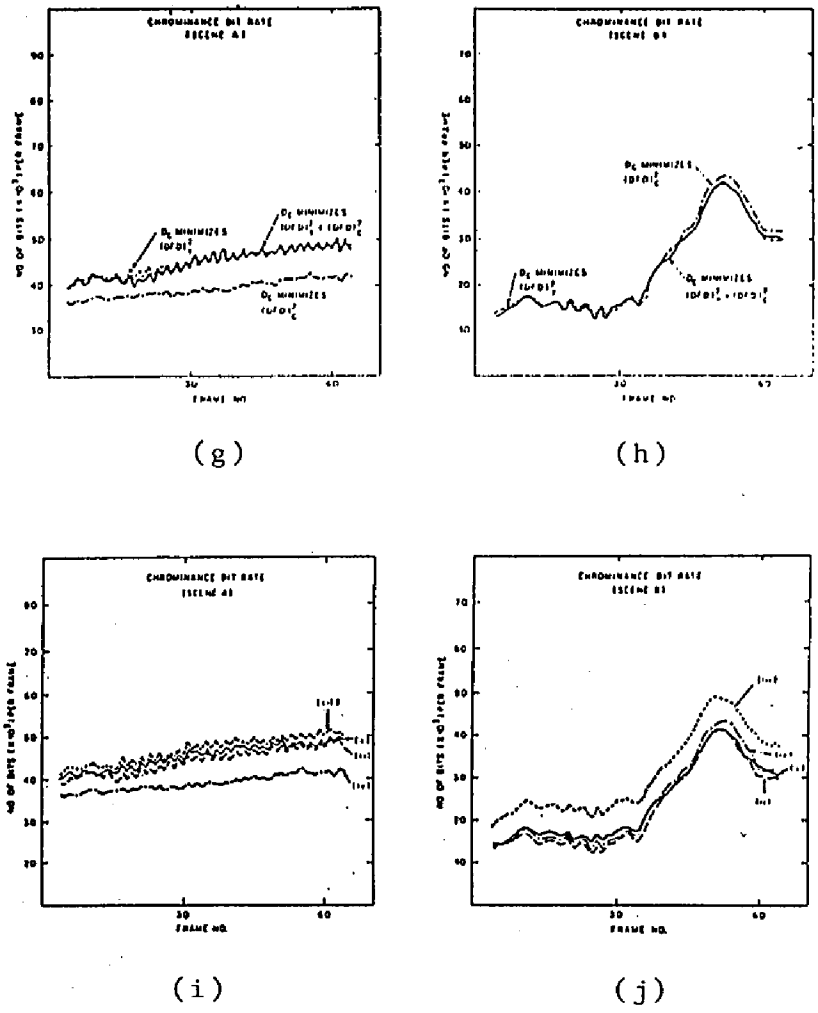


(e)



(f)

Fig.2.17: Comparison of bit rates for different types of predictors (a,b,c,d,e,f); effect of displacement estimation on chrominance bit rate (g,h); comparison of chrominance bit rates for various methods of switching the chrominance prediction (i,j).



(g) (h)

(i) (j)

Fig.2.17

(cont.)

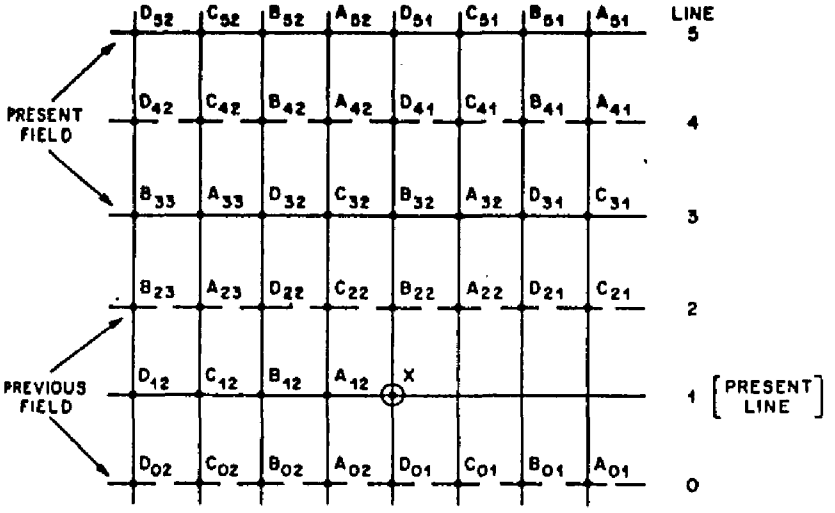
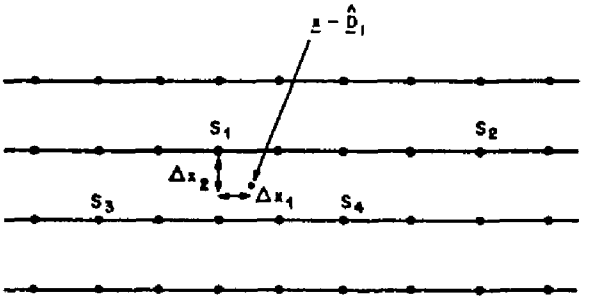


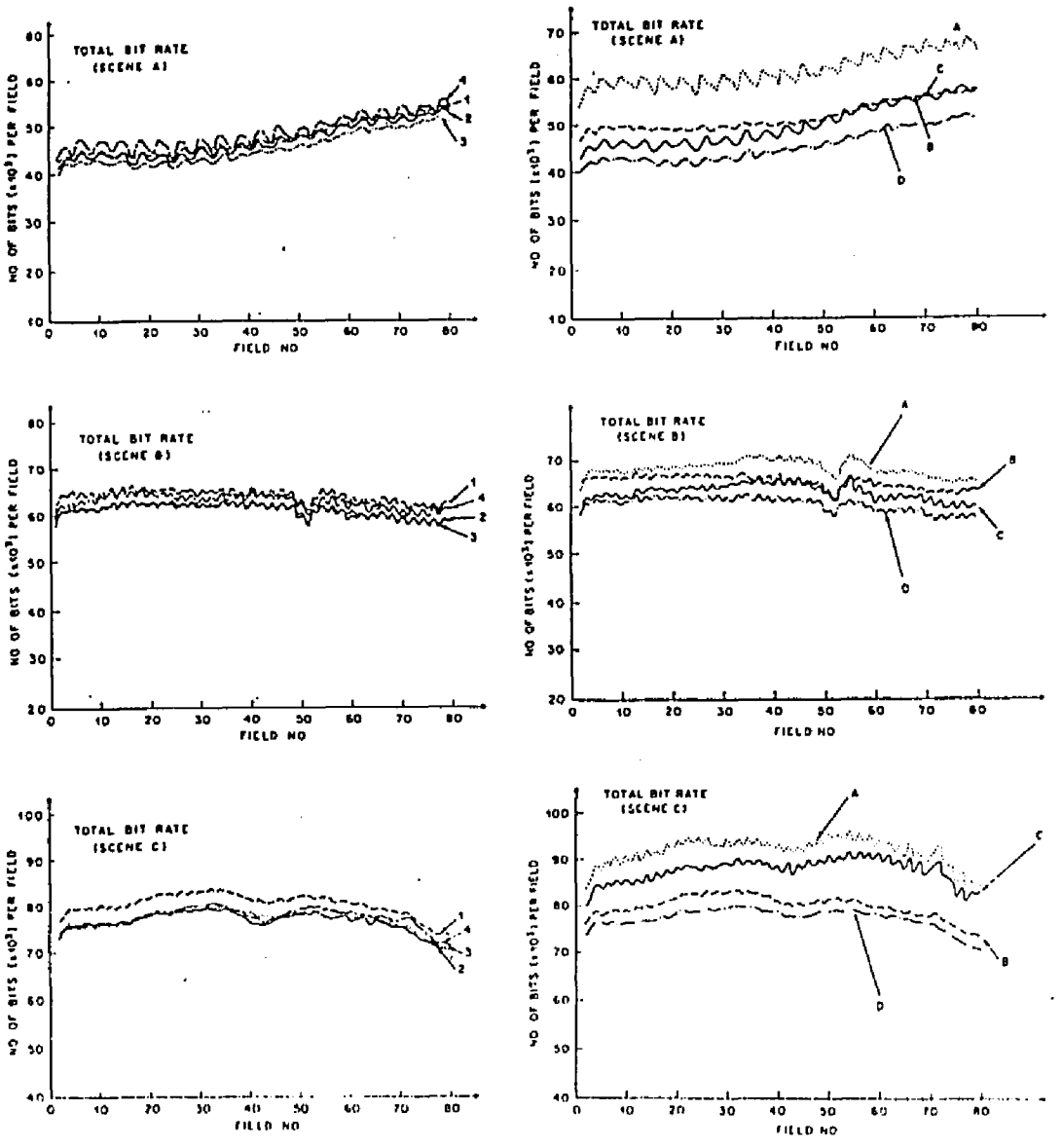
Fig.2.18: Map of pels surrounding the present pel in $4f_{sc}$ sampled NTSC signal



$$\underline{\Delta x} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}$$

$$\underline{\nabla S_1} = \begin{bmatrix} (S_2 - S_1)/4 \\ S_1 - (S_3 + S_4)/2 \end{bmatrix}$$

Fig.2.19: Displaced signal estimation



(a) (b)

Fig.2.20: Comparison of coder output for various motion compensation schemes (a); comparison of coder output for various prediction modes (b)

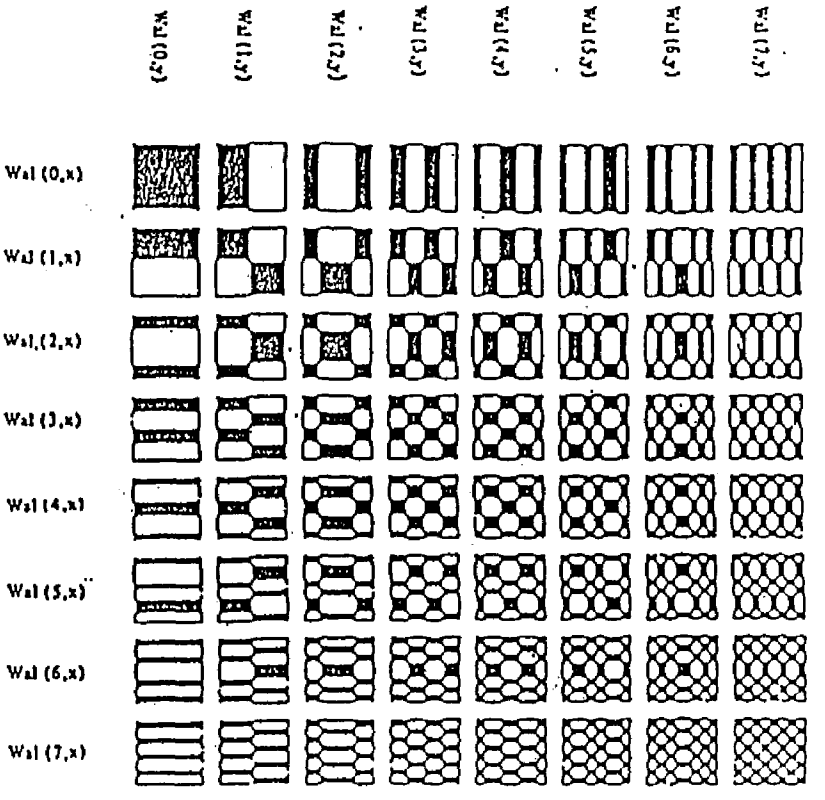


Fig.3.1: Two-dimensional Walsh-Hadamard basis matrices, Black=1, White=-1

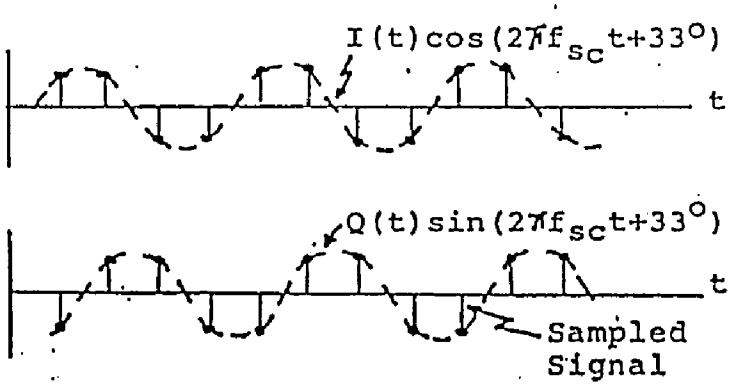


Fig.3.2: Phase relationship between the sampling clock and the modulated I and Q components

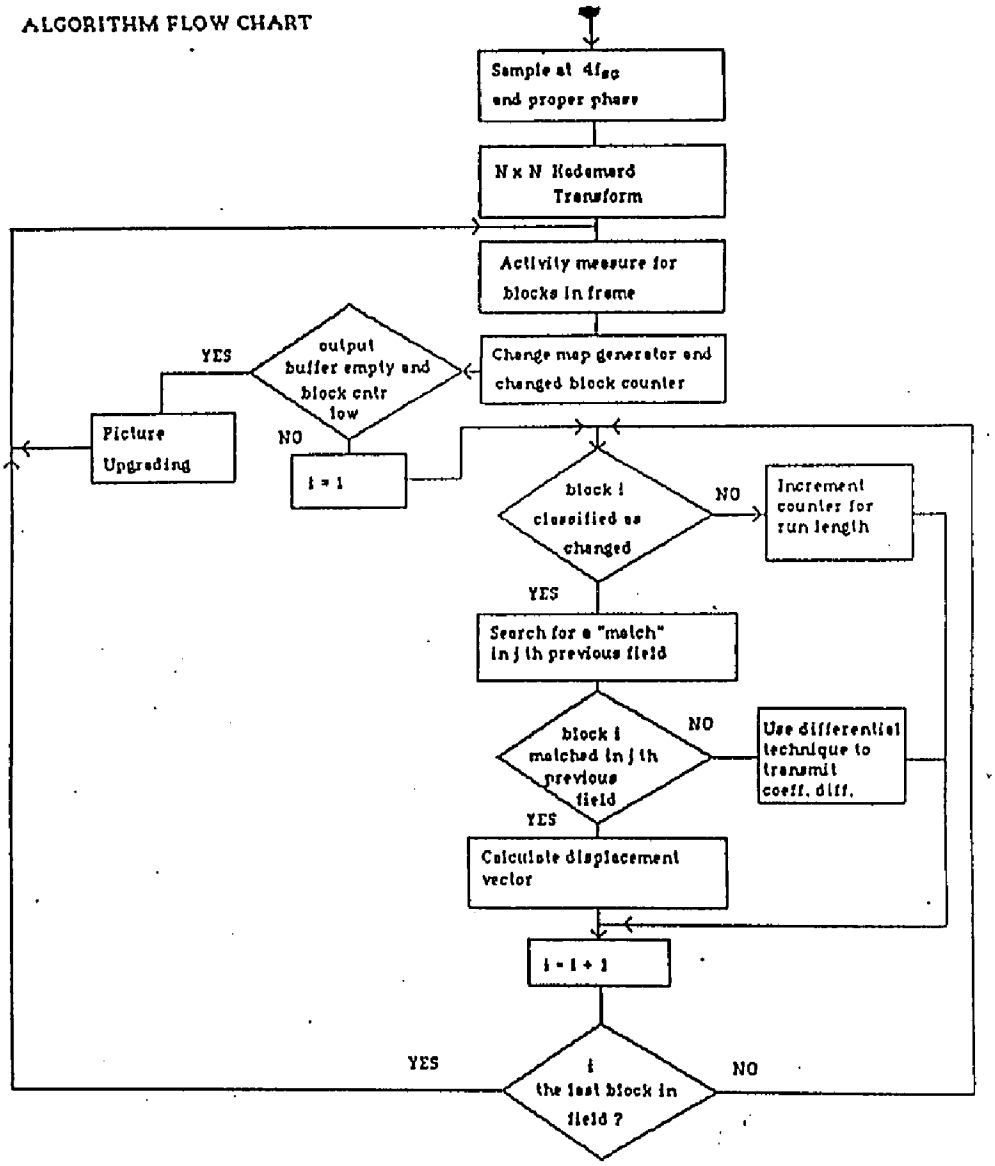


Fig.3.2.1: The flow chart of the motion detection, estimation and compensation algorithm

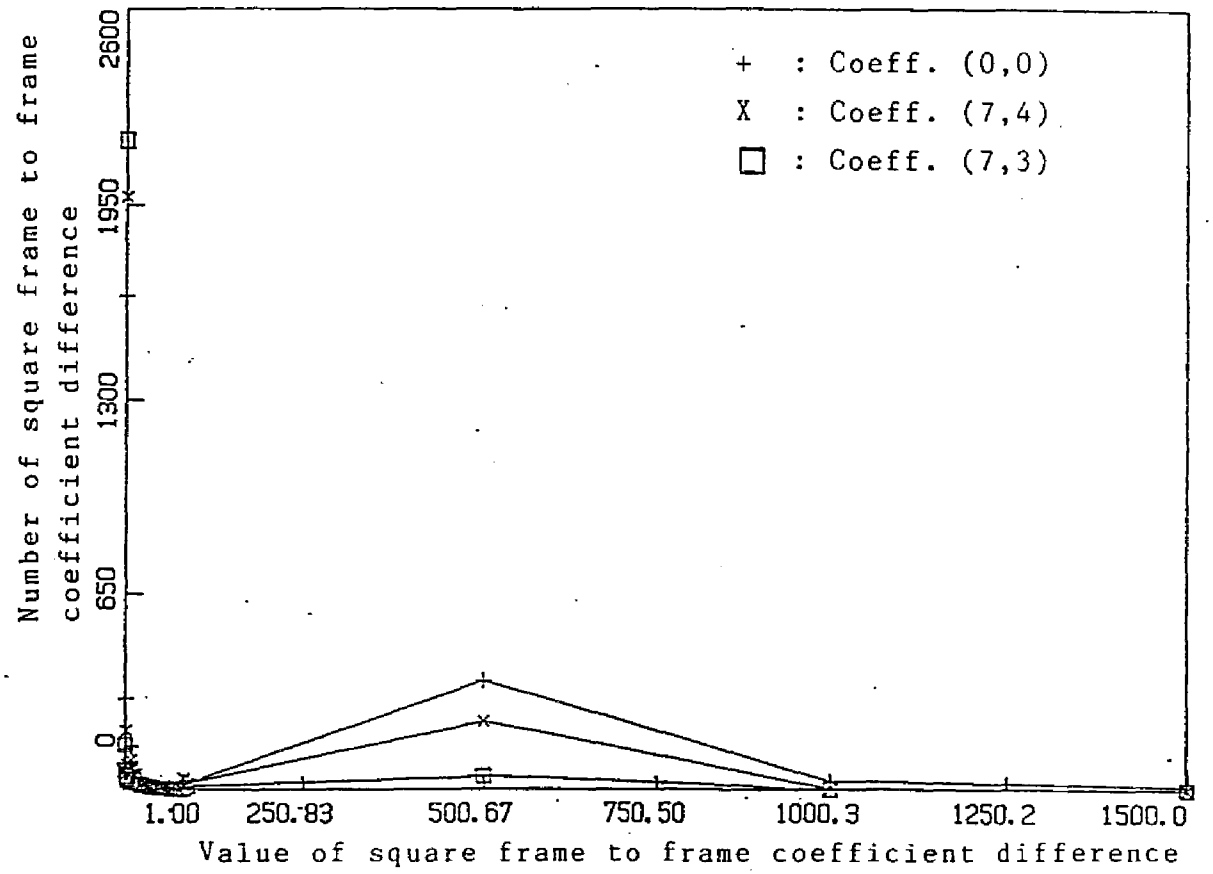


Fig.3.2.2: a) Number of squared frame to frame coefficient difference vs. value of square frame to frame coefficient difference for Y, I, Q DC coefficients averaged over the sequence of the man speaking

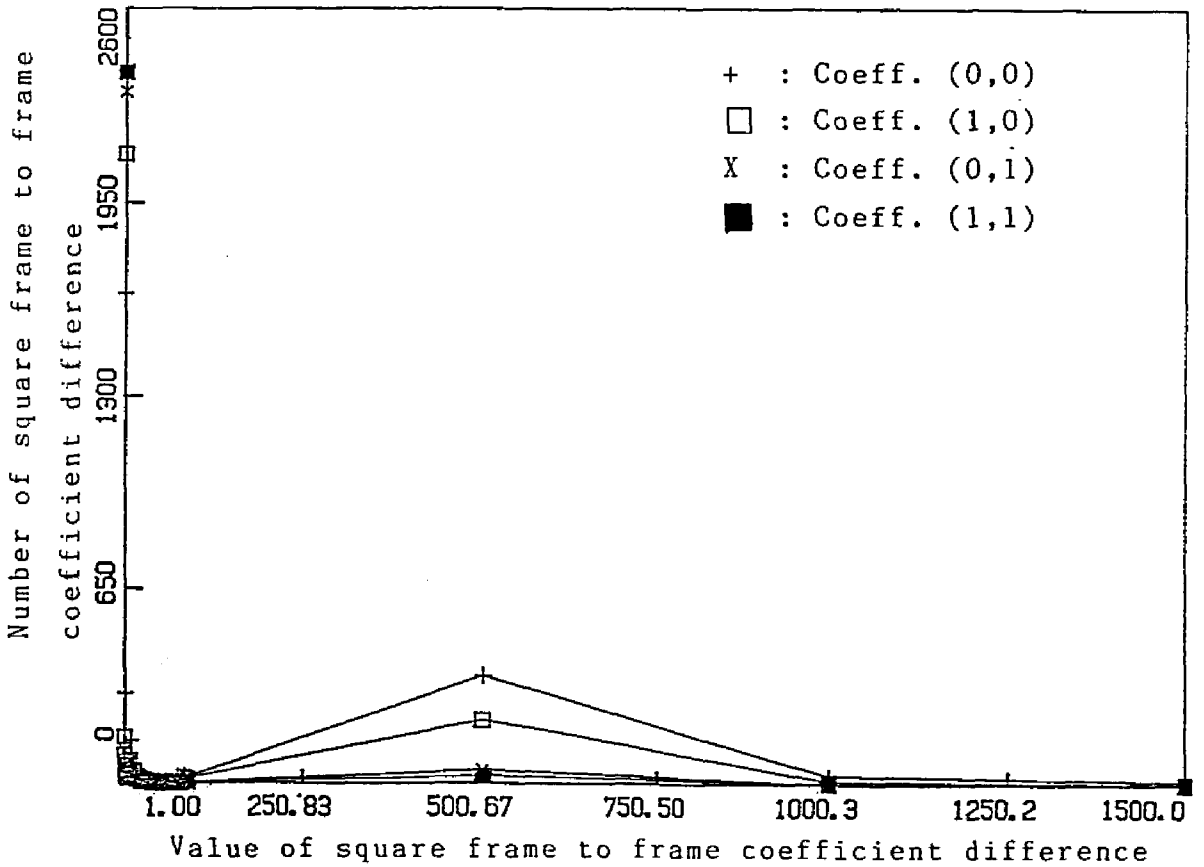


Fig.3.2.2: b) Number of squared frame to frame coefficient difference vs. value of square frame to frame coefficient difference for Y coefficients averaged over the sequence of the man speaking

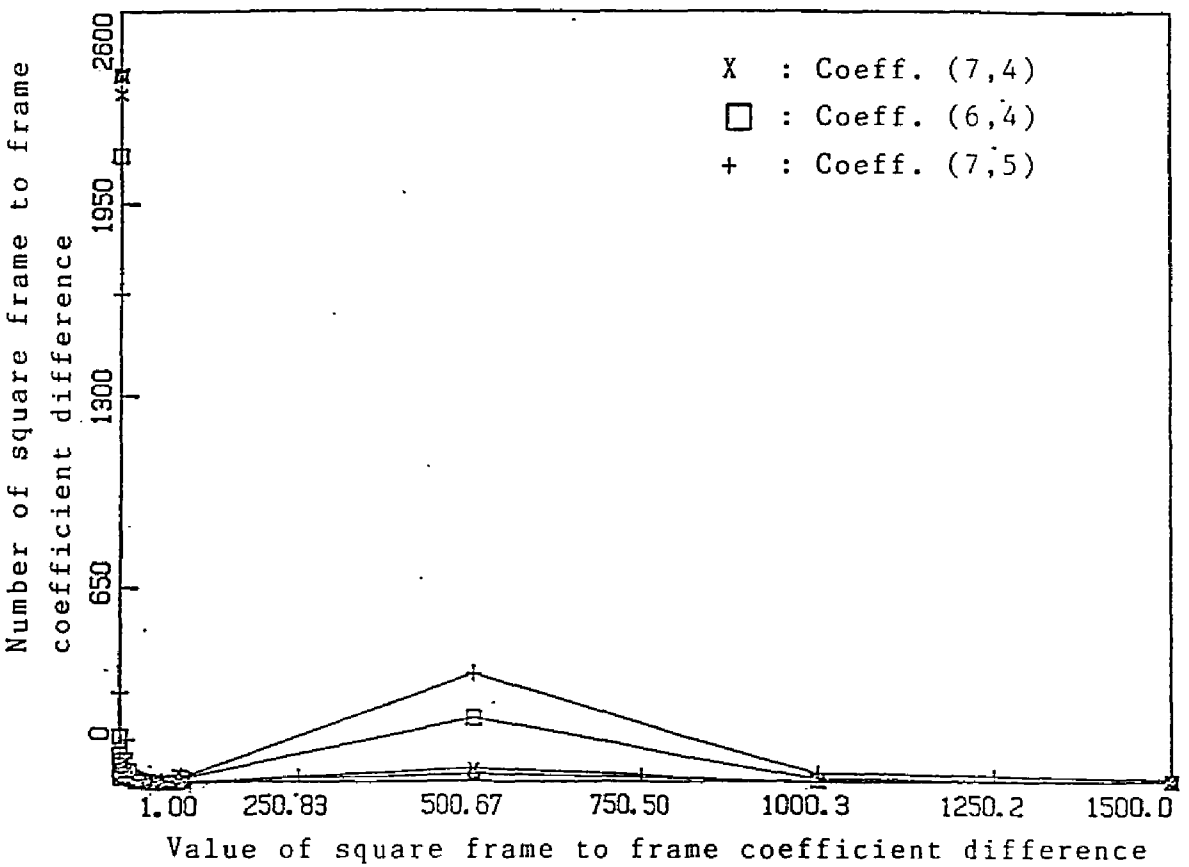


Fig.3.2.2: c) Number of squared frame to frame coefficient difference vs. value of square frame to frame coefficient difference for I coefficients averaged over the sequence of the man speaking

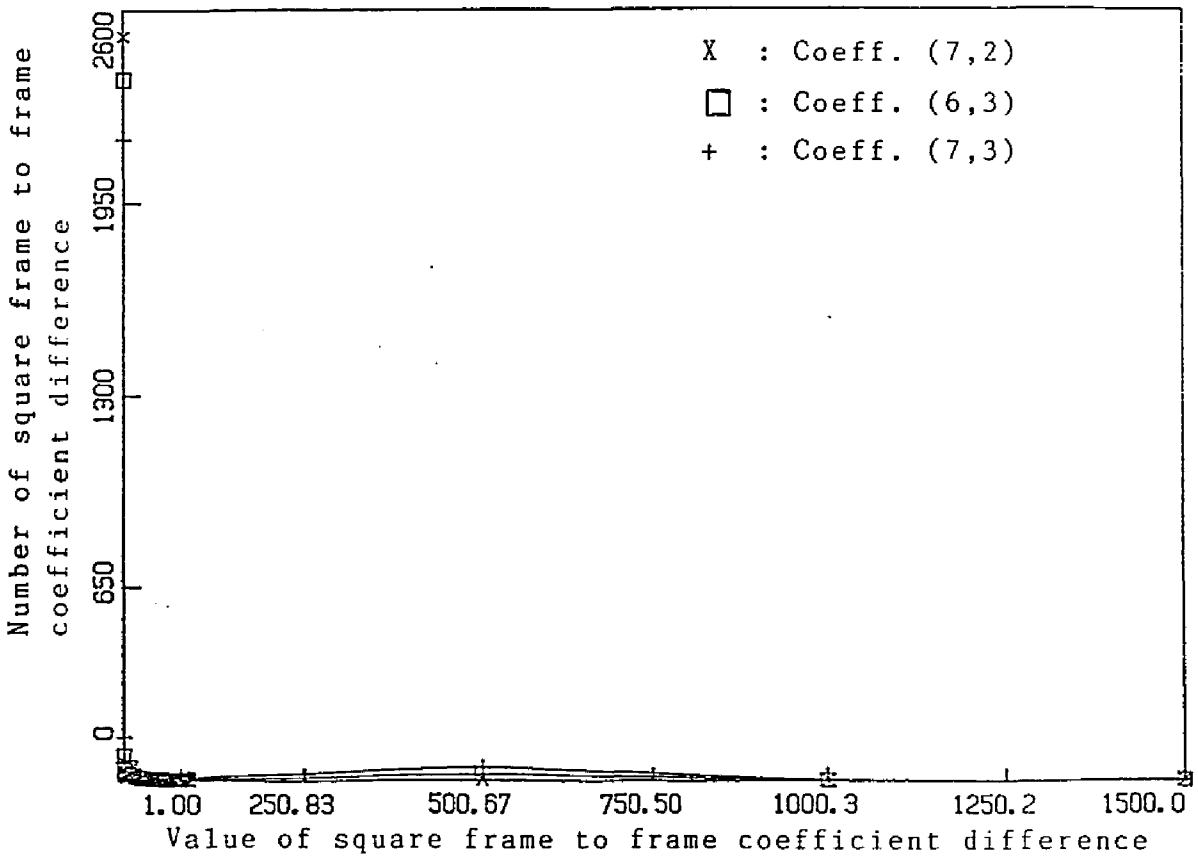


Fig.3.2.2: d) Number of squared frame to frame coefficient difference vs. value of square frame to frame coefficient difference for Q coefficients averaged over the sequence of the man speaking

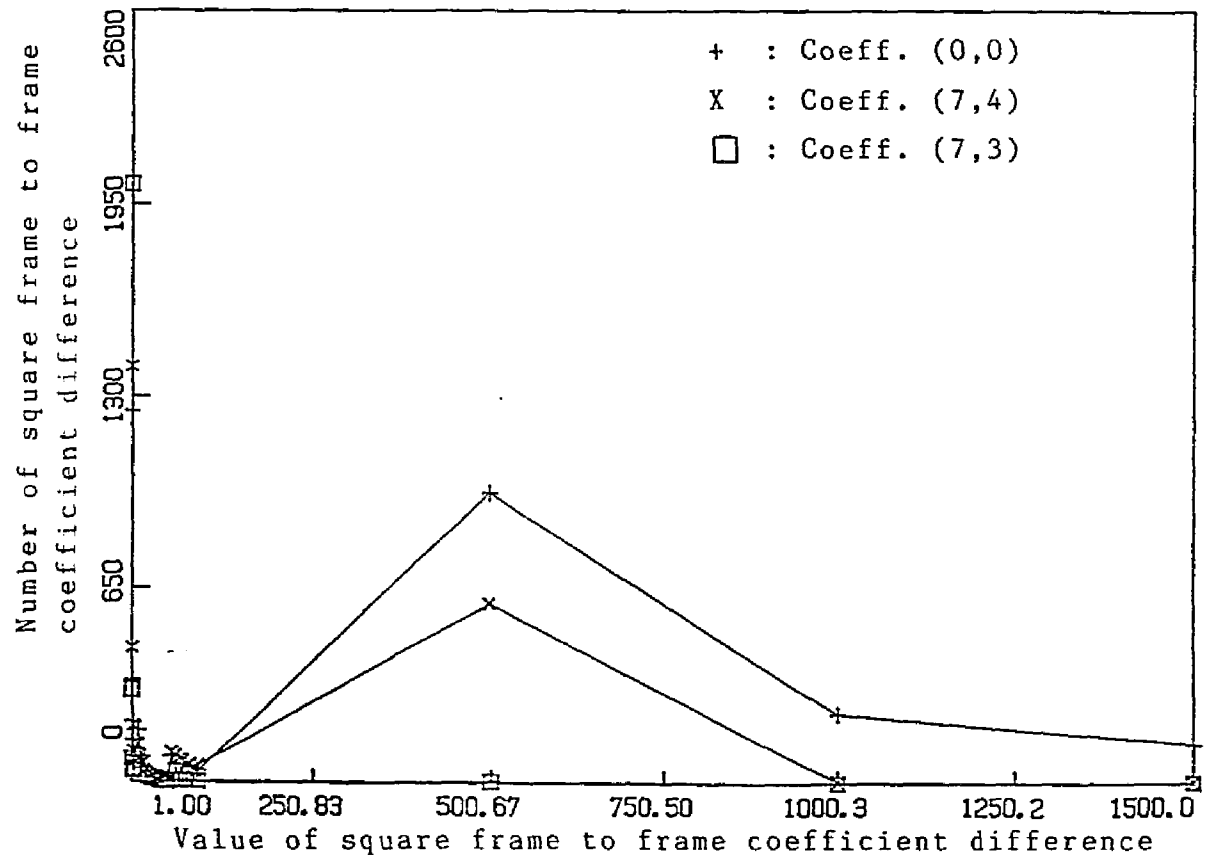


Fig.3.2.2: e) Number of squared frame to frame coefficient difference vs. value of square frame to frame coefficient difference for Y, I, Q DC coefficients averaged over the sequence of the moving color cube

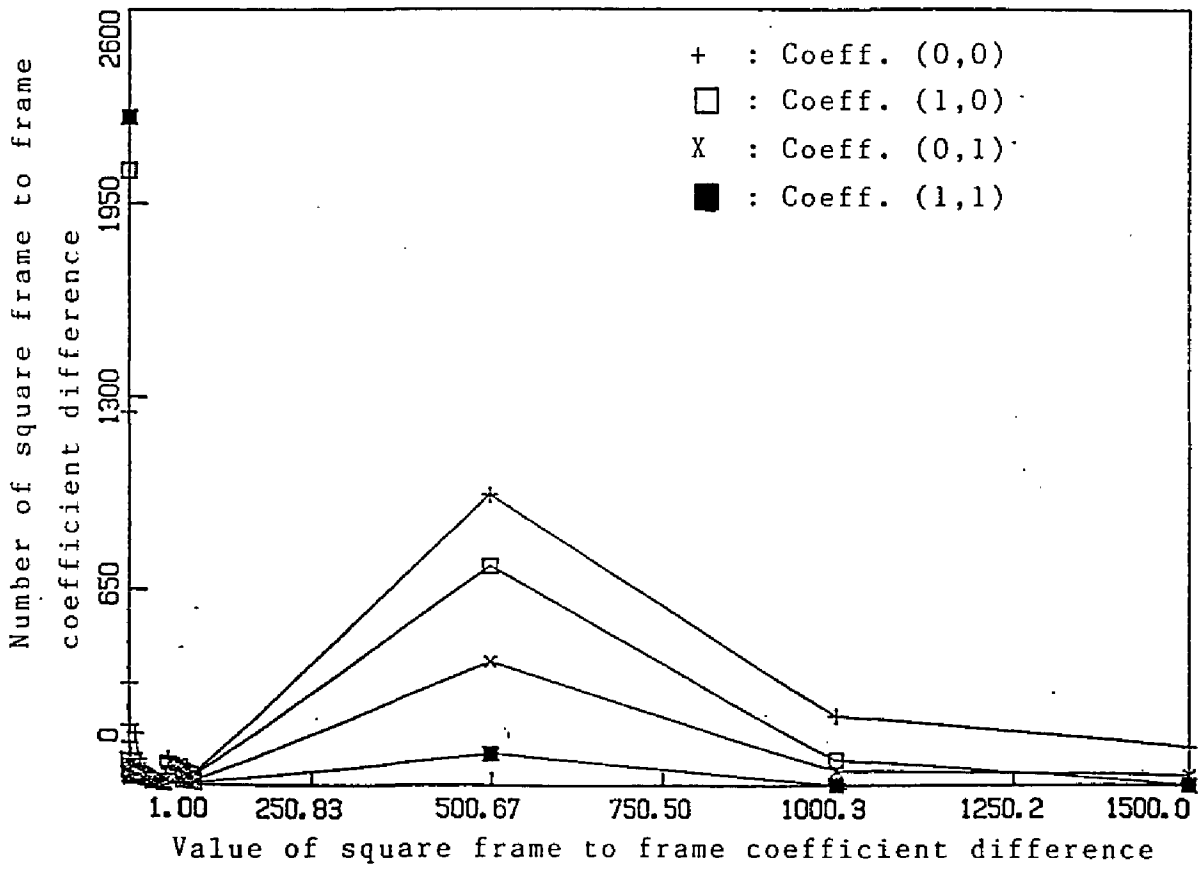


Fig.3.2.2: f) Number of squared frame to frame coefficient difference vs. value of square frame to frame coefficient difference for Y coefficients averaged over the sequence of the moving color cube

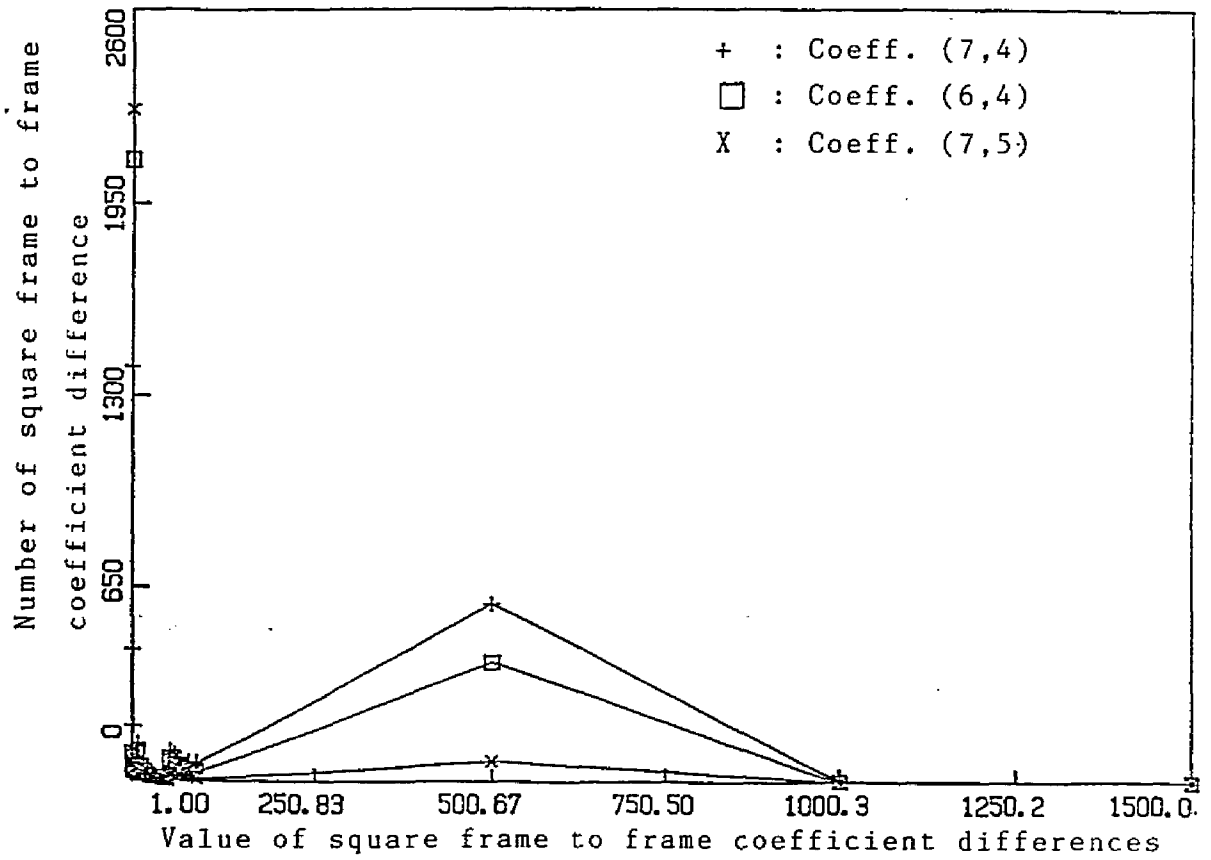


Fig.3.2.2: g) Number of squared frame to frame coefficient difference vs. value of square frame to frame coefficient difference for I coefficients averaged over the sequence of the moving color cube

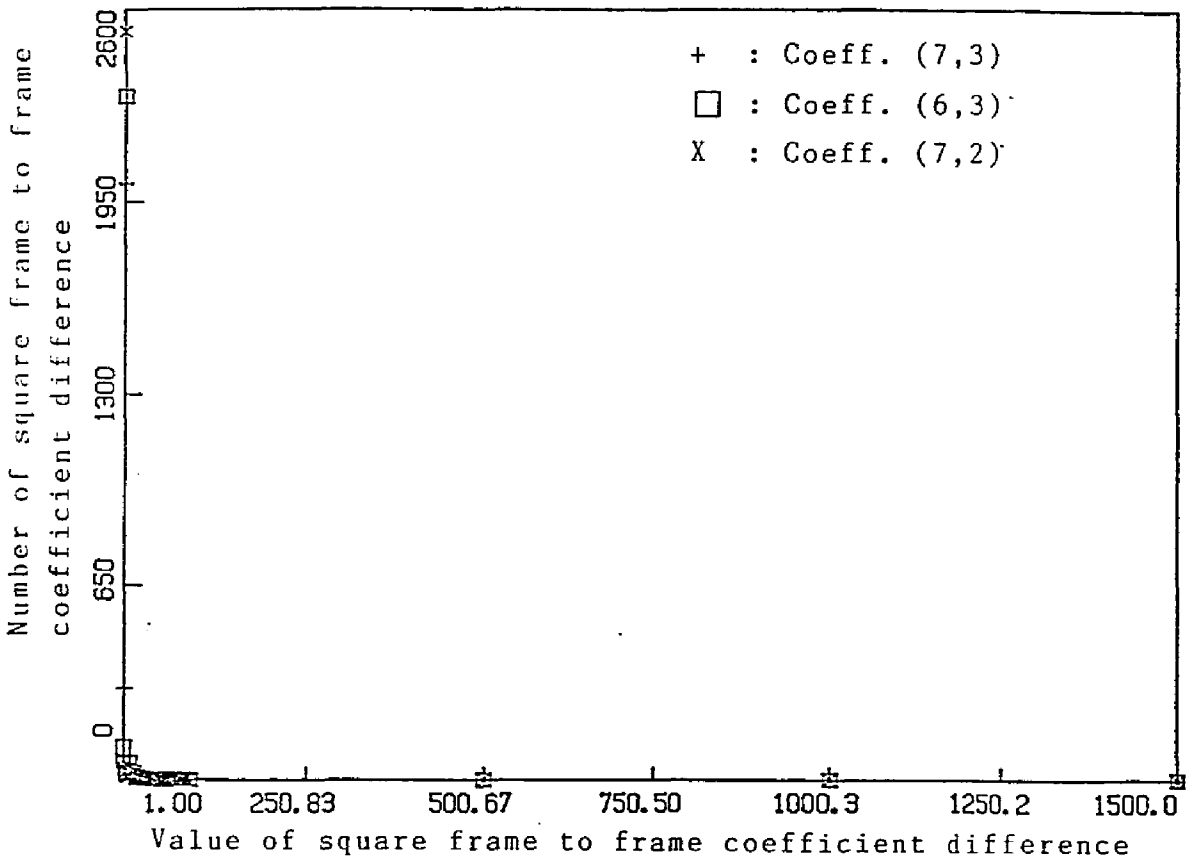


Fig.3.2.2: h) Number of squared frame to frame coefficient difference vs. value of square frame to frame coefficient difference for Q coefficients averaged over the sequence of the moving color cube

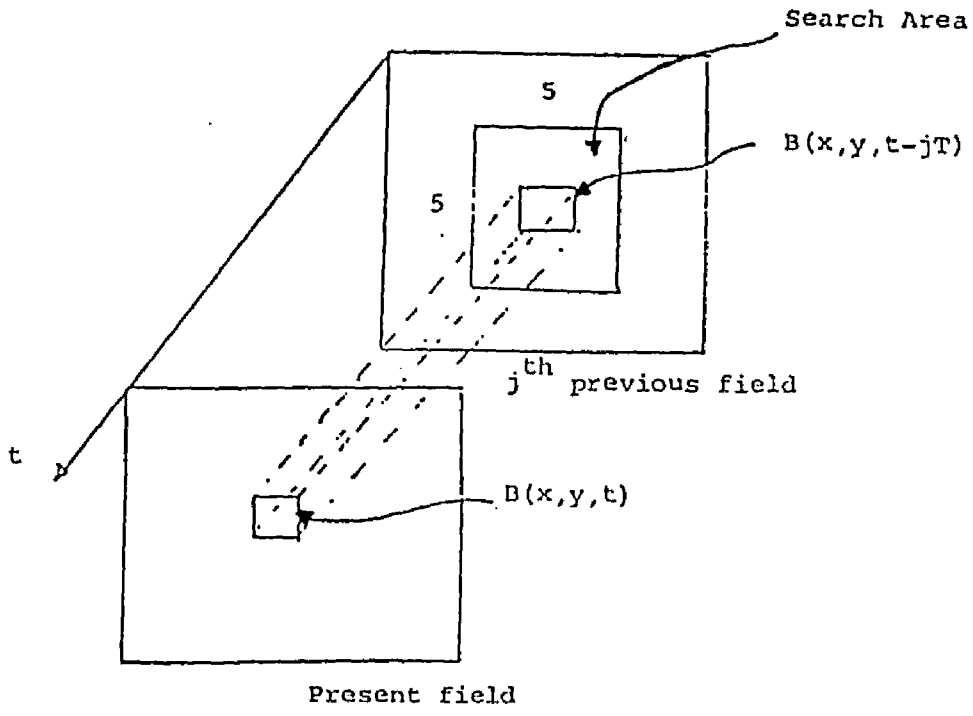
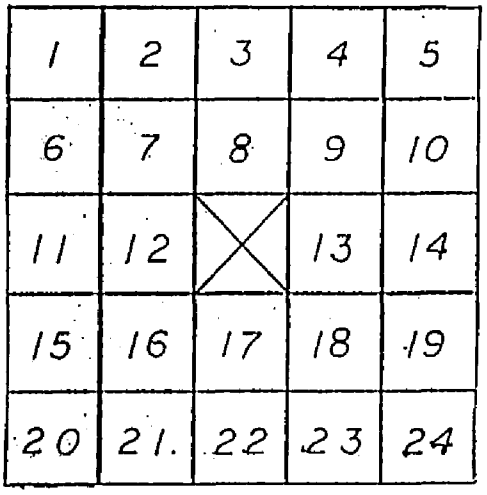
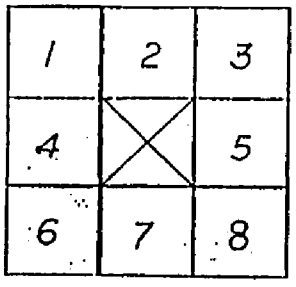


Fig.3.2.3: Illustration of search procedure



(a)



(b)

Fig.3.2.4: The values of the displacement vector for a 5x5 block search area (a) and a 3x3 block search area (b)

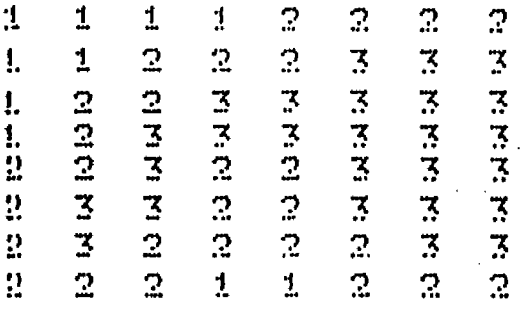
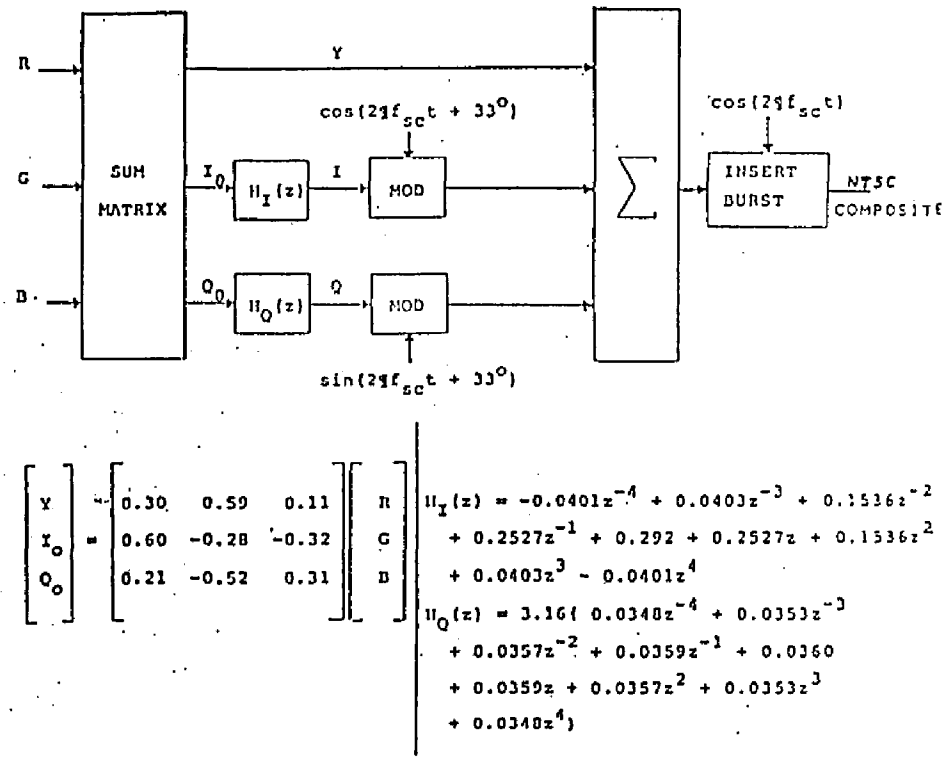


Fig.3.2.5: The order in which the coefficients are upgraded in Mode 4



$$\begin{bmatrix} Y \\ I_Q \\ Q_Q \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.28 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$H_I(z) = -0.0401z^{-4} + 0.0403z^{-3} + 0.1536z^{-2} + 0.2527z^{-1} + 0.292 + 0.2527z + 0.1536z^{2} + 0.0403z^{3} - 0.0401z^{4}$$

$$H_Q(z) = 3.16(0.0348z^{-4} + 0.0353z^{-3} + 0.0357z^{-2} + 0.0359z^{-1} + 0.0360 + 0.0359z + 0.0357z^{2} + 0.0353z^{3} + 0.0348z^{4})$$

Fig.4.1.1: The block diagram of the system used to obtain the NTSC composite signal

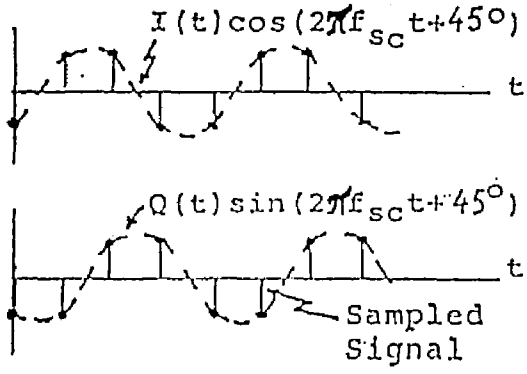
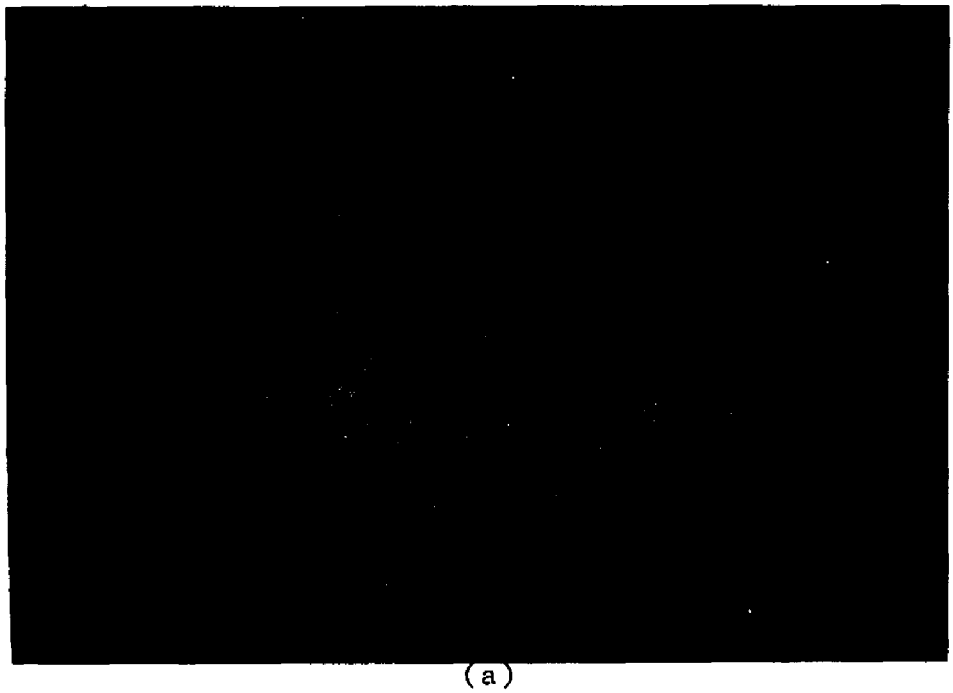
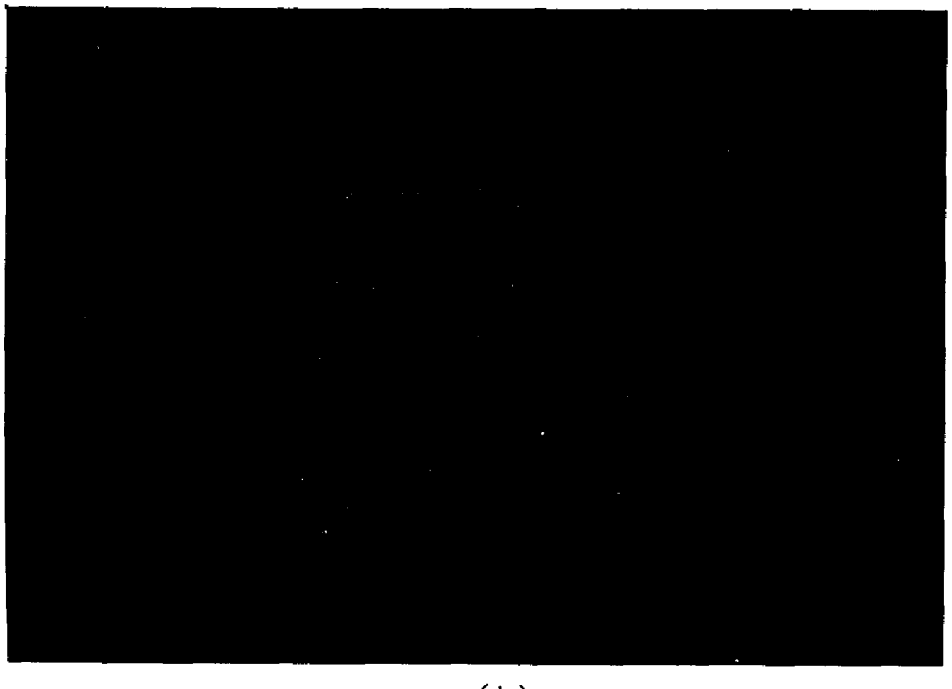


Fig.4.1.2: Phase relationship between the "proper" sampling clock and the modulated I and Q components



(a)



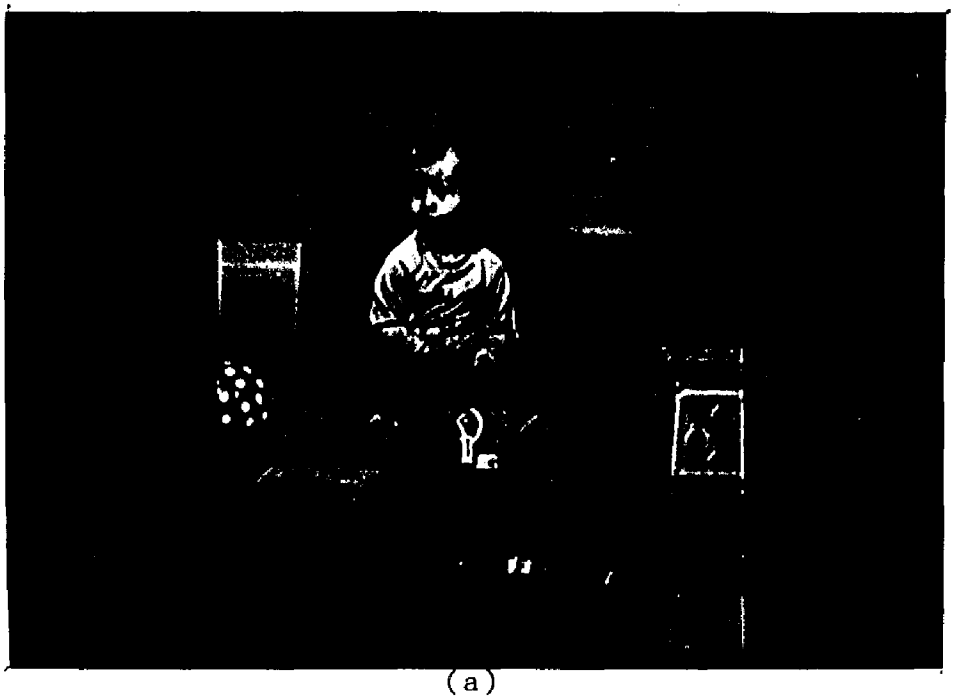
(b)

Fig.4.1.3: Inverse Hadamard transform of TEST patern with Q and I coefficients in Table 4.1.2 set to zero. a) Original TEST pattern; b) 4x4 transform; c) 8x8 transform; d) 16x16 transform

(c)

(d)

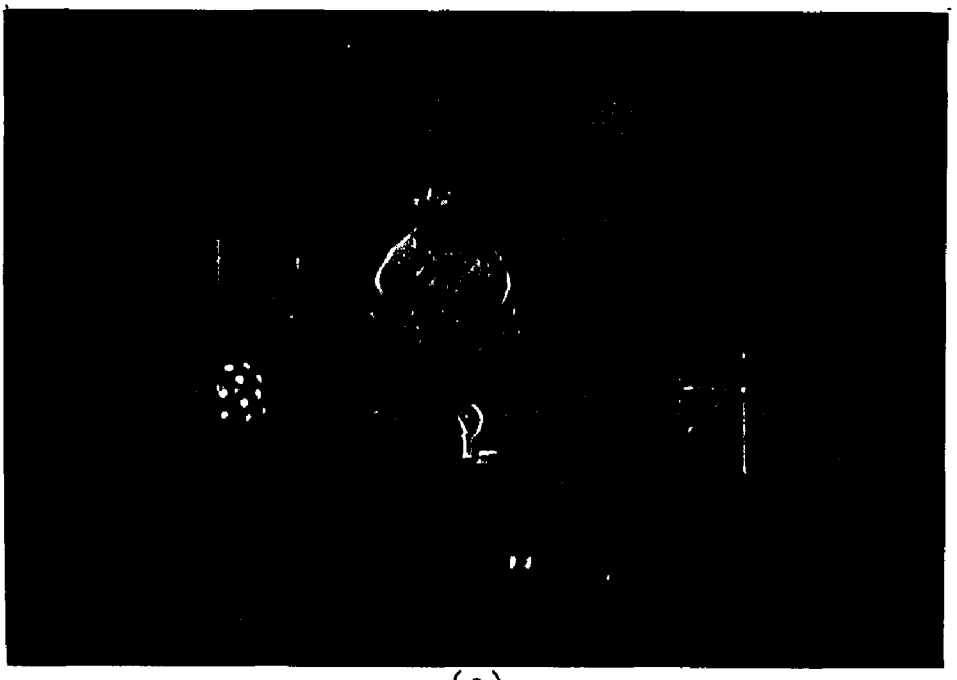
Fig.4.1.3. (cont.)



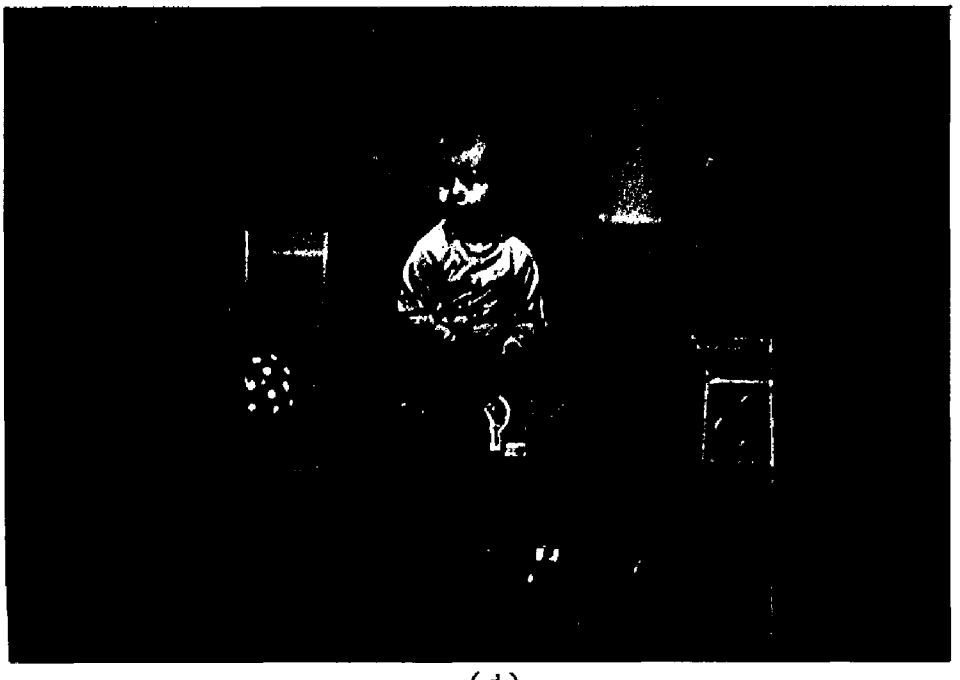
(a)

(b)

Fig.4.1.4: Reconstructed composite images using separated components with coefficient assignment of Table 4.1.2. a) Original Boy image; b) 4x4 transform; c) 8x8 transform; d) 16x16 transform



(c)



(d)

Fig.4.1.4 (cont.)

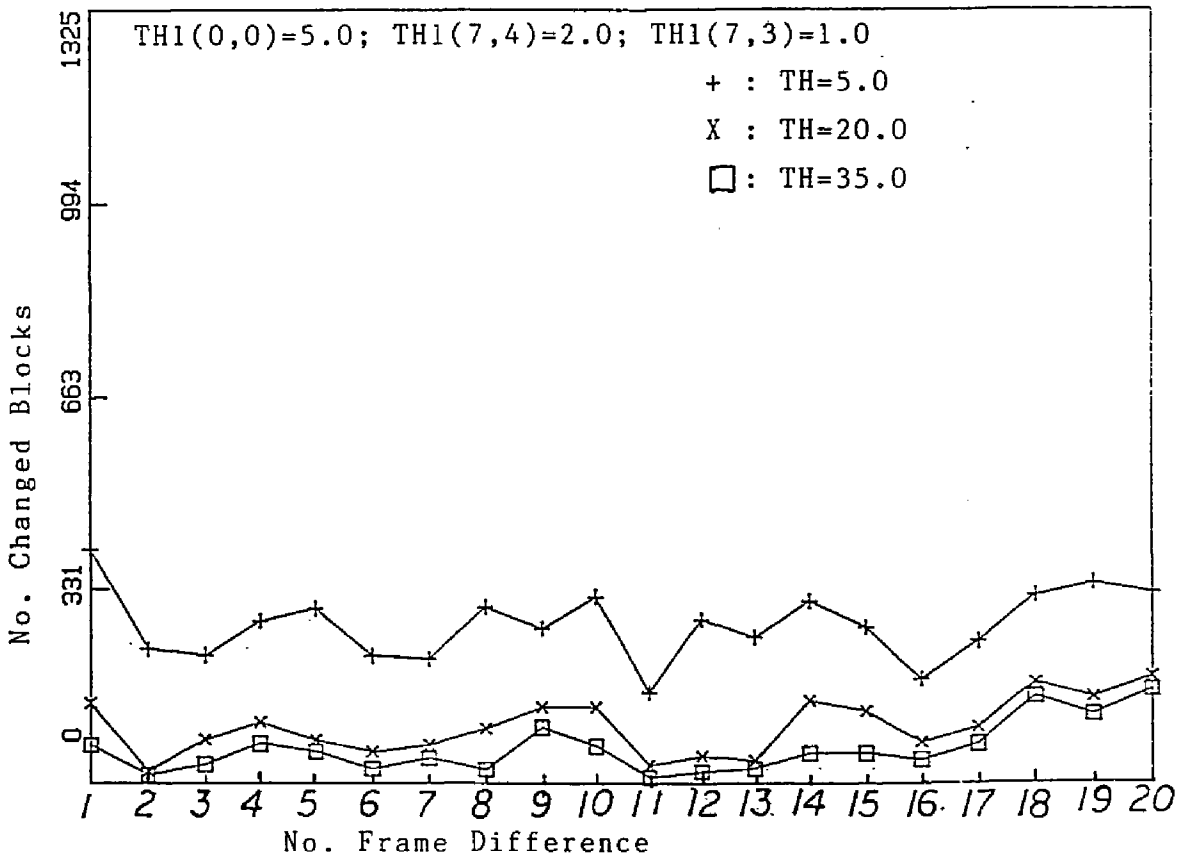


Fig.4.2.1: a) The dependence of the number of changed blocks on the motion detection thresholds in the man speaking sequence

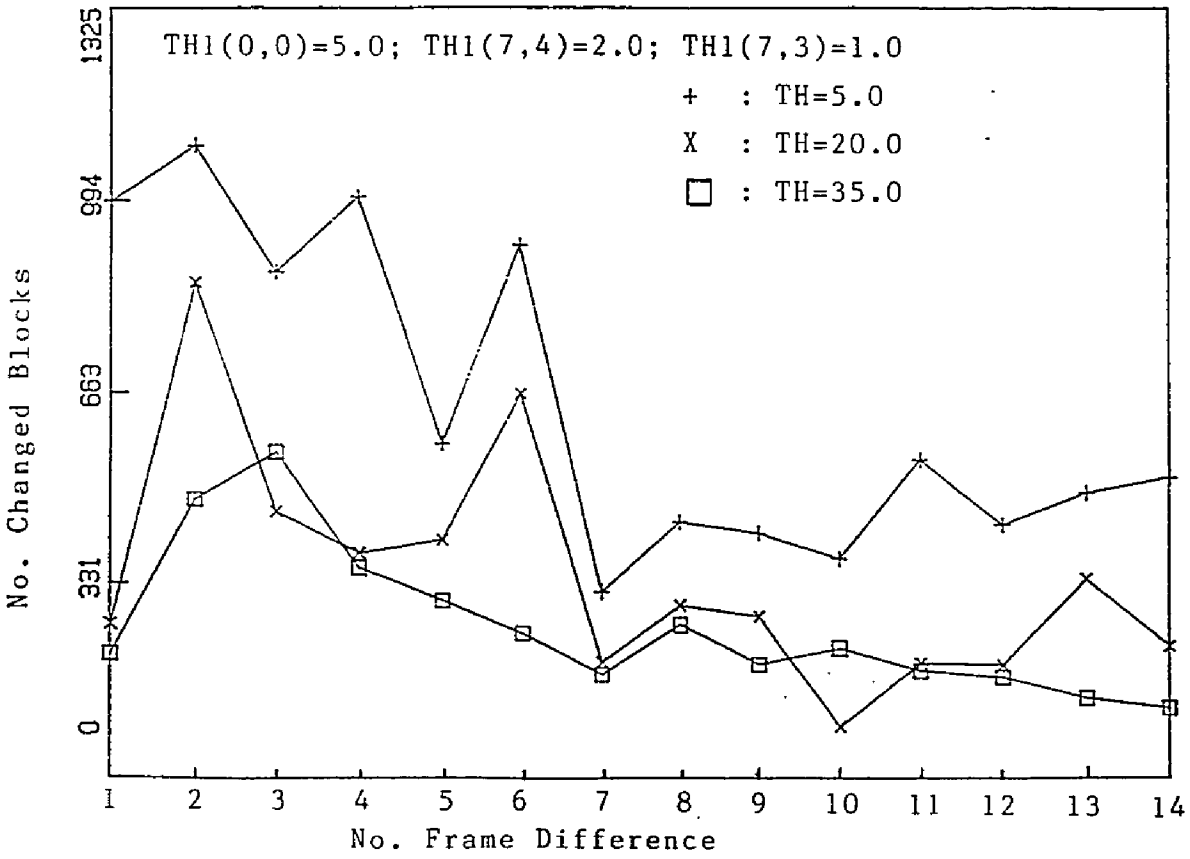


Fig.4.2.1: b) The dependence of the number of changed blocks on the motion detection thresholds in the moving color cube

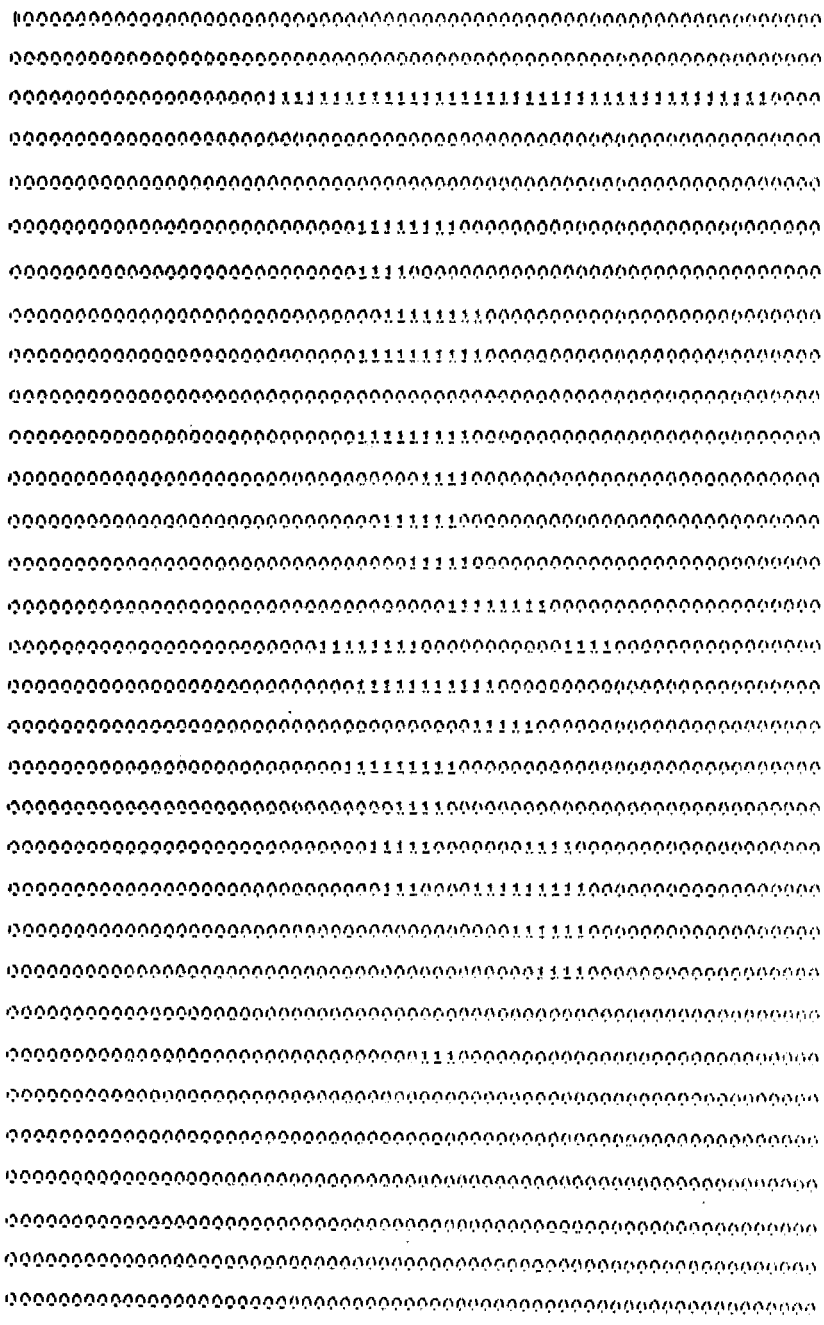


Fig.4.2.2: A "change map" for the man speaking sequence

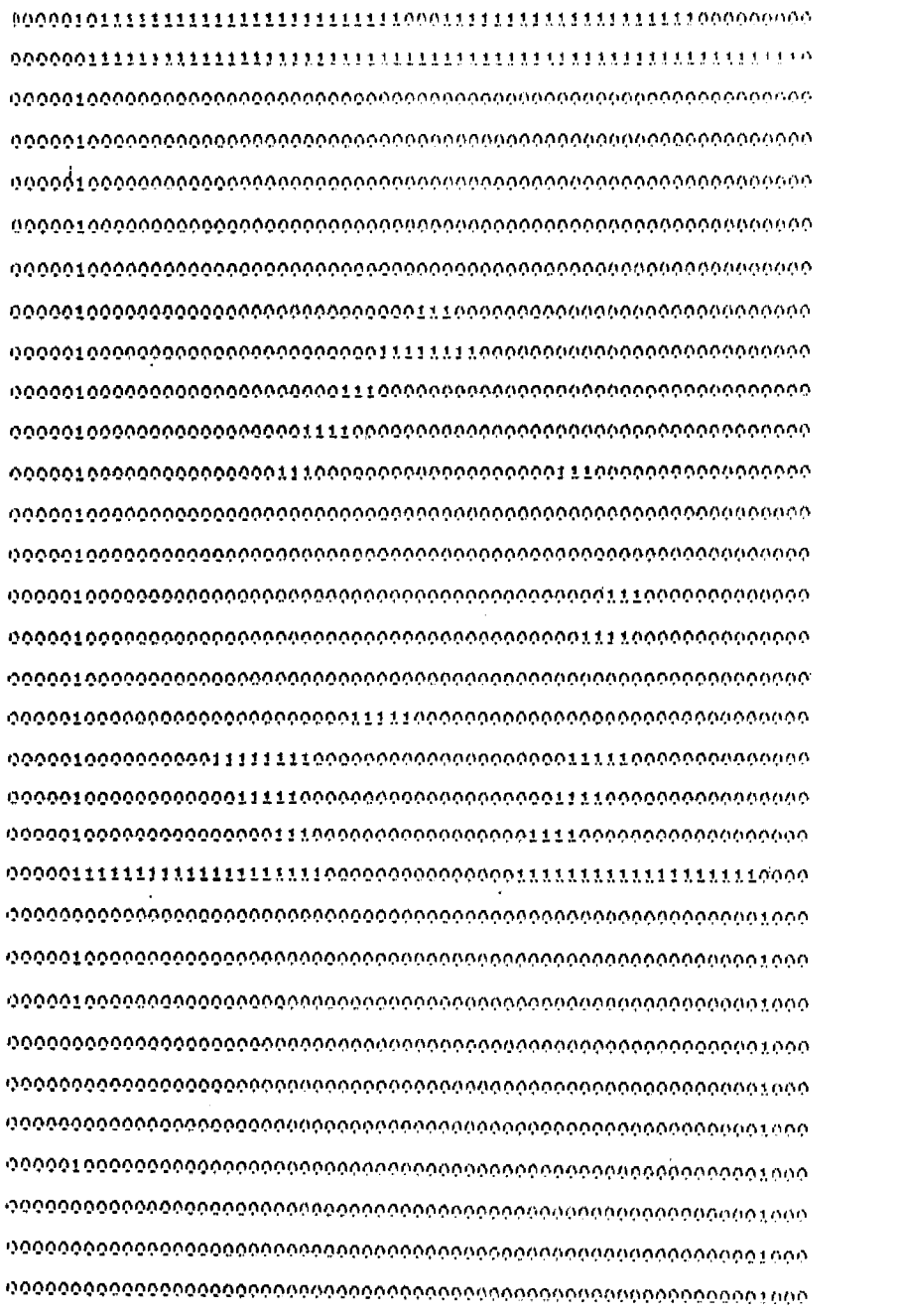


Fig.4.2.3: A "change map" for the moving color cube sequence

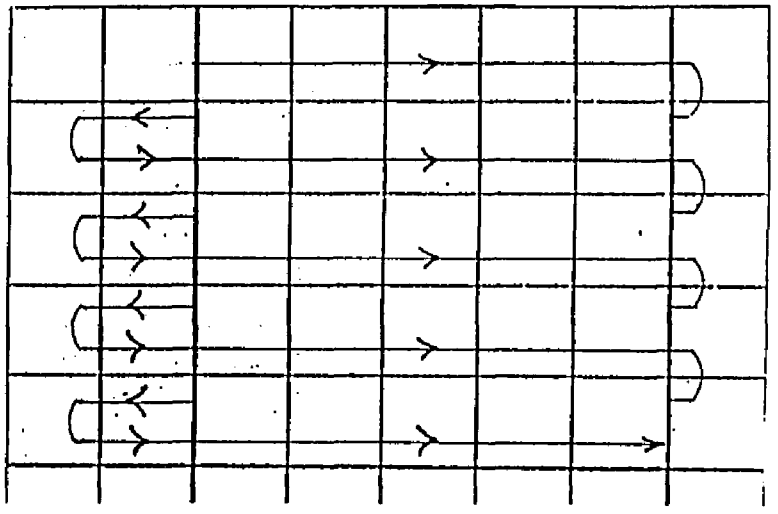


Fig.4.2.4: The direction of the search

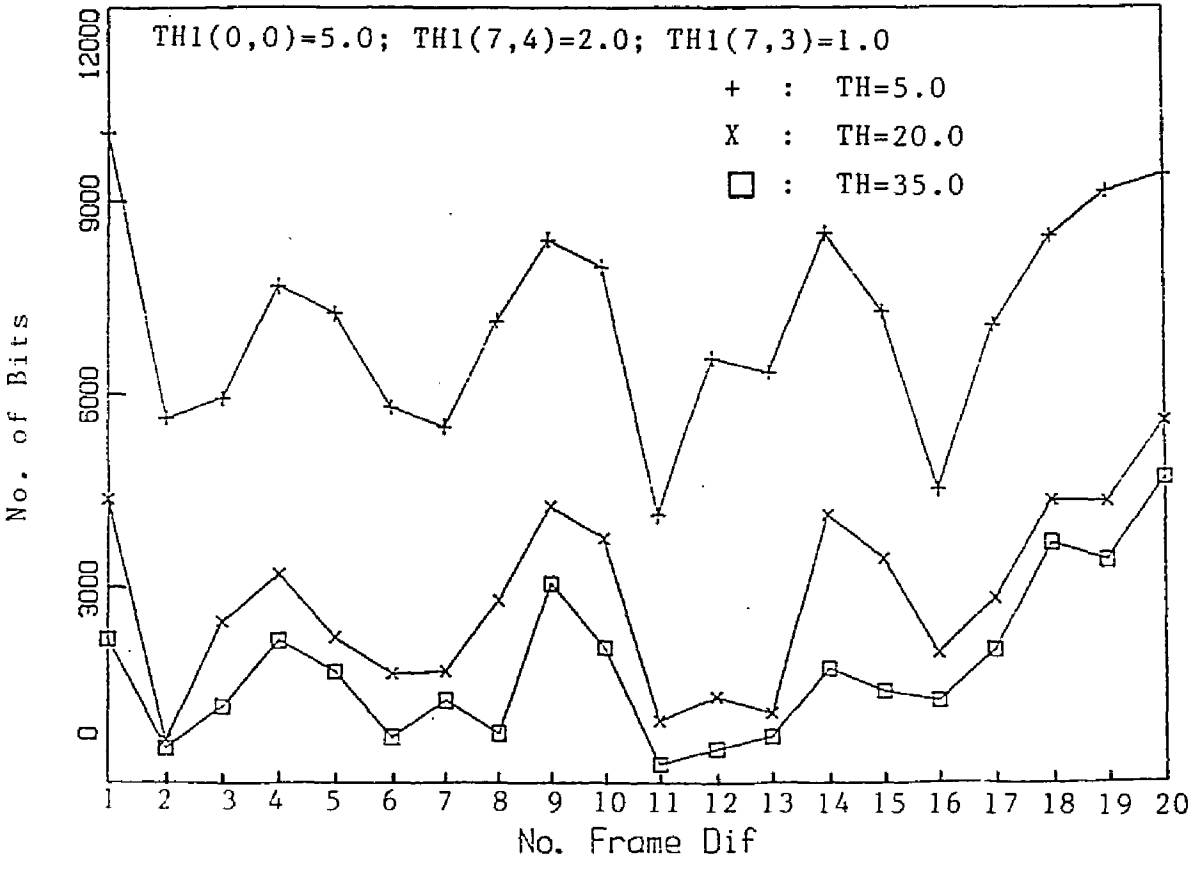


Fig.4.2.5: The dependence of the bit rate on the motion detection thresholds in the man speaking sequence

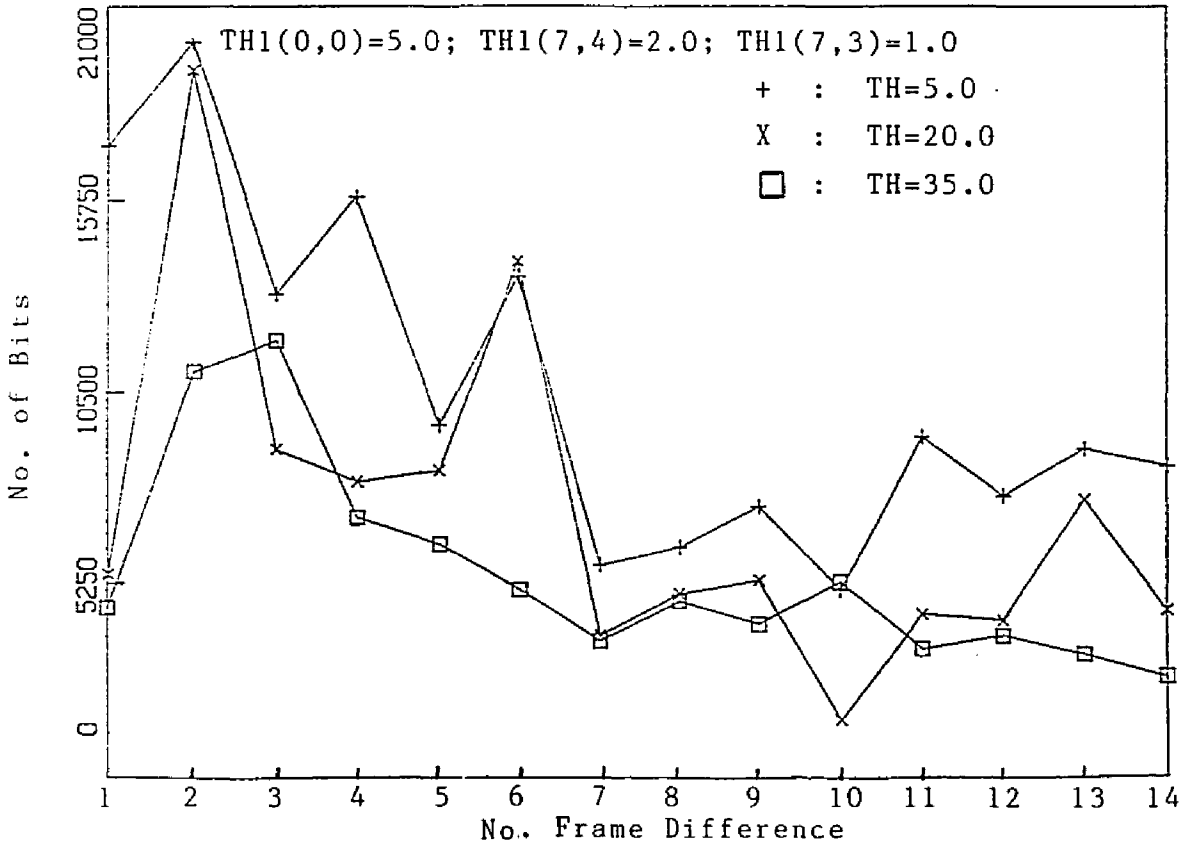


Fig.4.2.6: The dependence of the bit rate on the motion detection thresholds in the moving color cube sequence

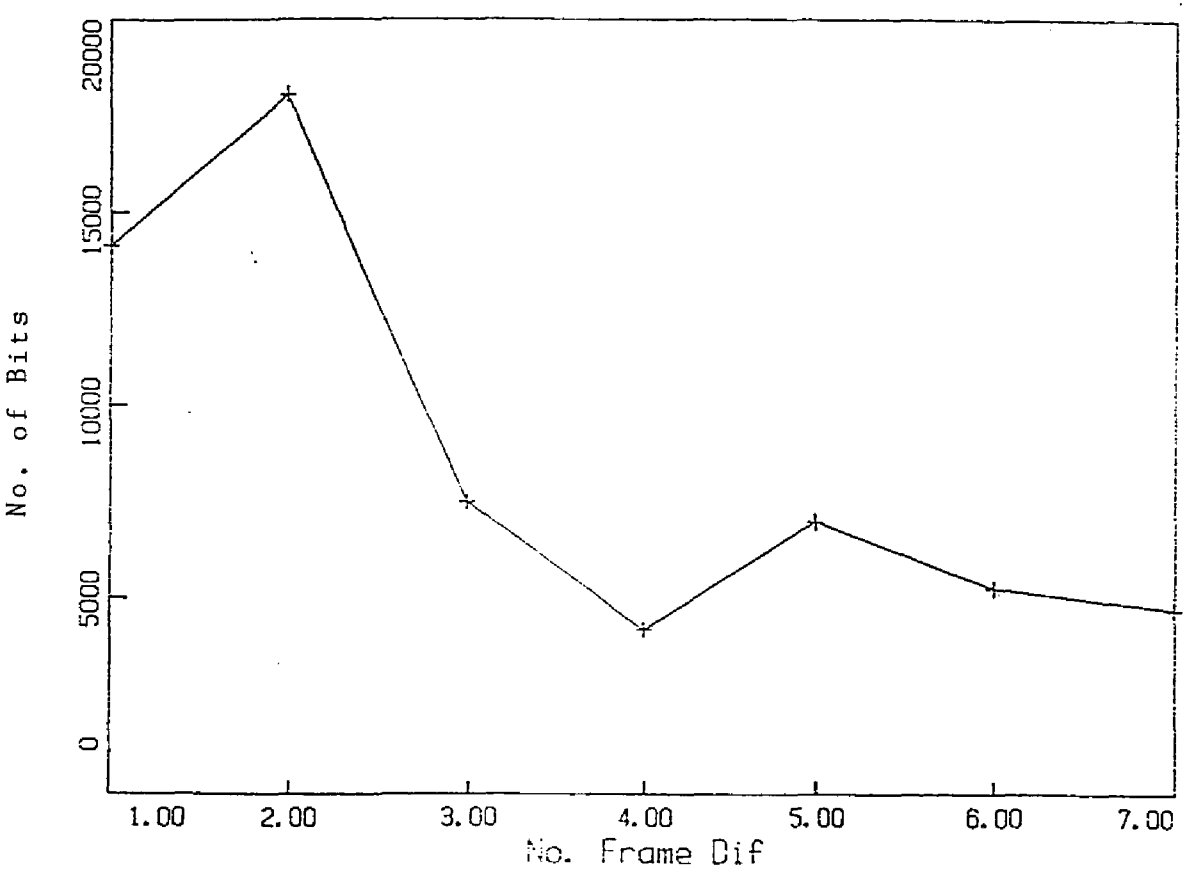


Fig.4.2.7: The plot of the bit rate for the moving color cube sequence when 4:1 temporal subsampling is employed

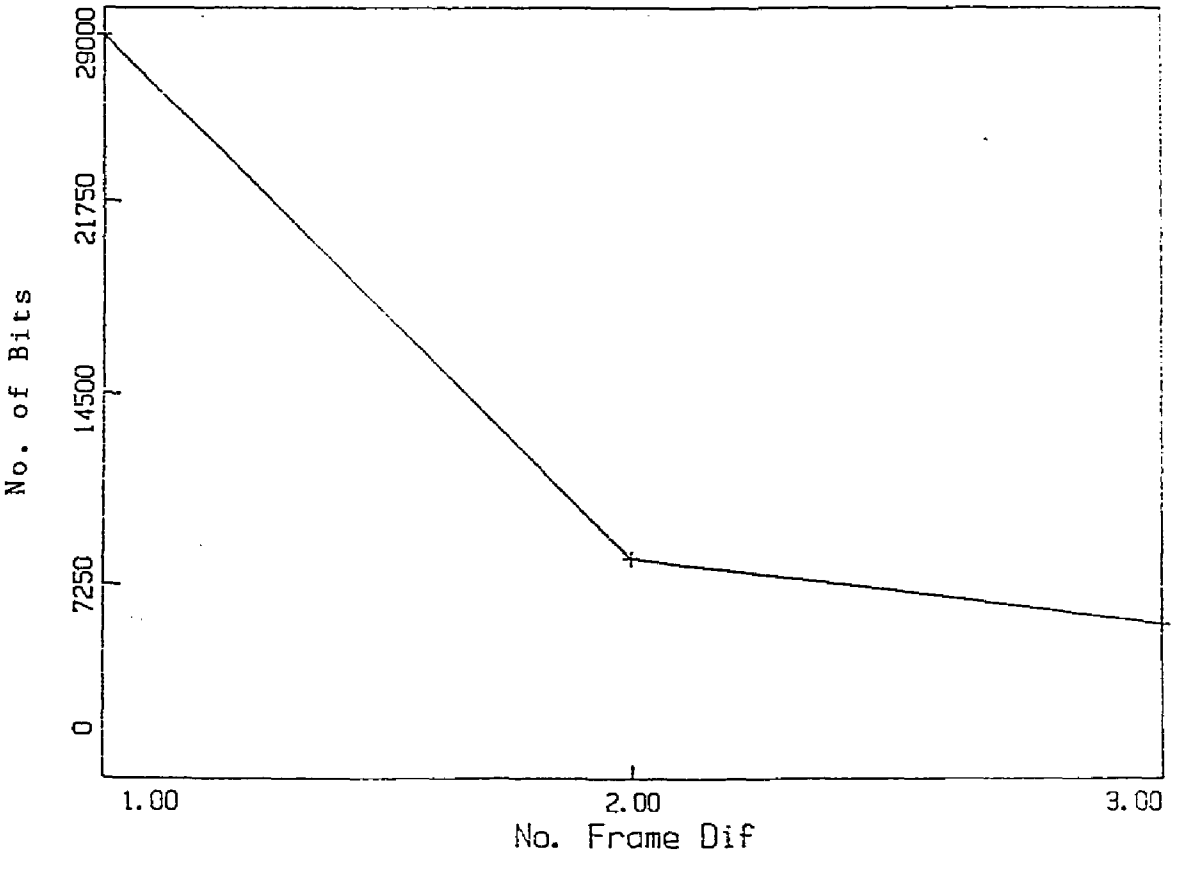


Fig.4.2.8: The plot of the bit rate for the moving color cube sequence when 8:1 temporal subsampling is employed

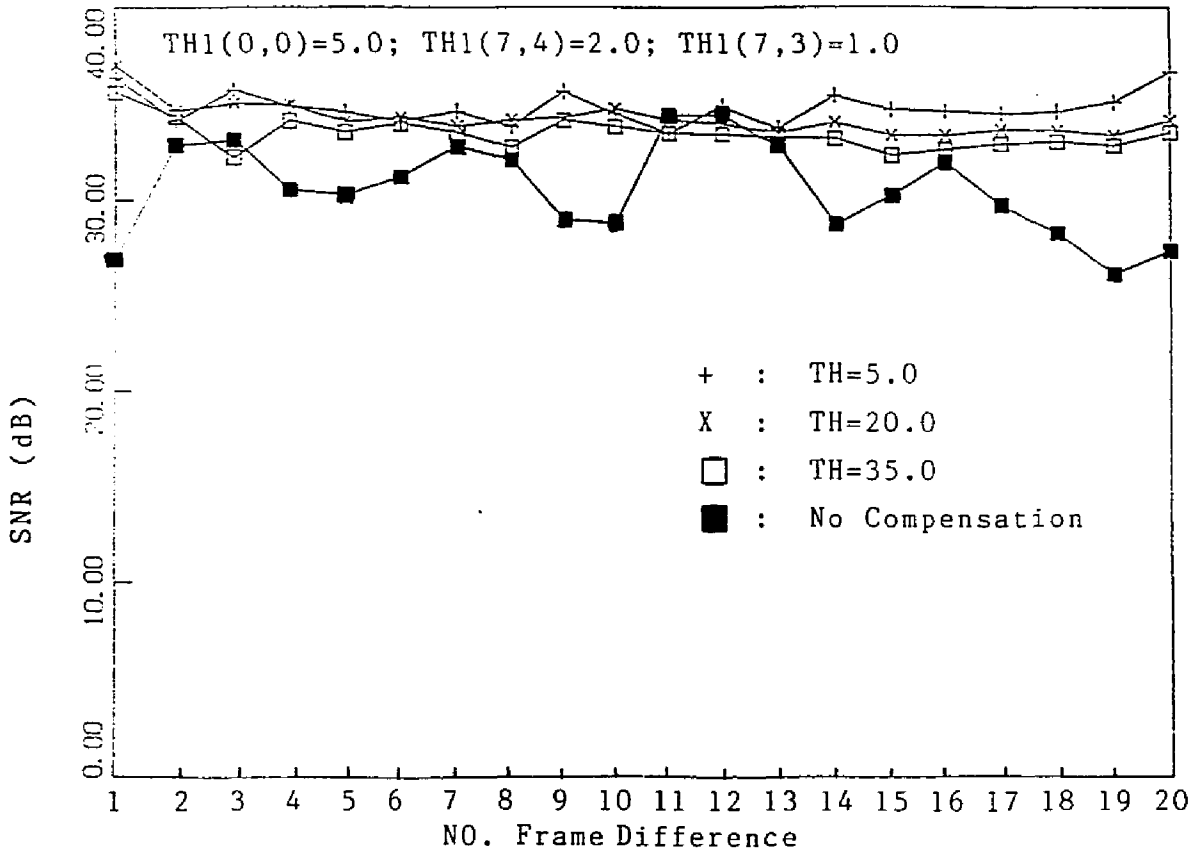


Fig.4.2.9: Comparison of SNR for reconstructed video frames for the man speaking sequence

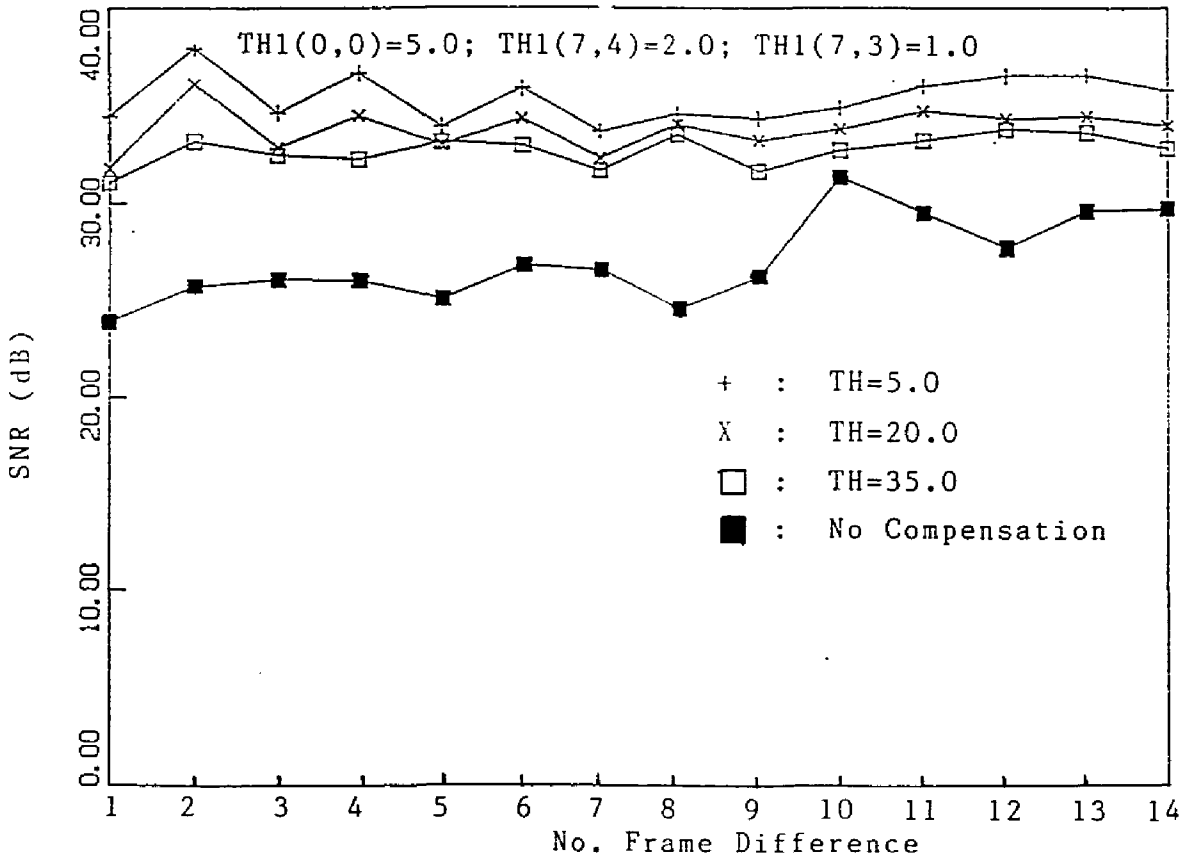


Fig.4.2.10: Comparison of SNR for reconstructed video frames for the moving color cube sequence

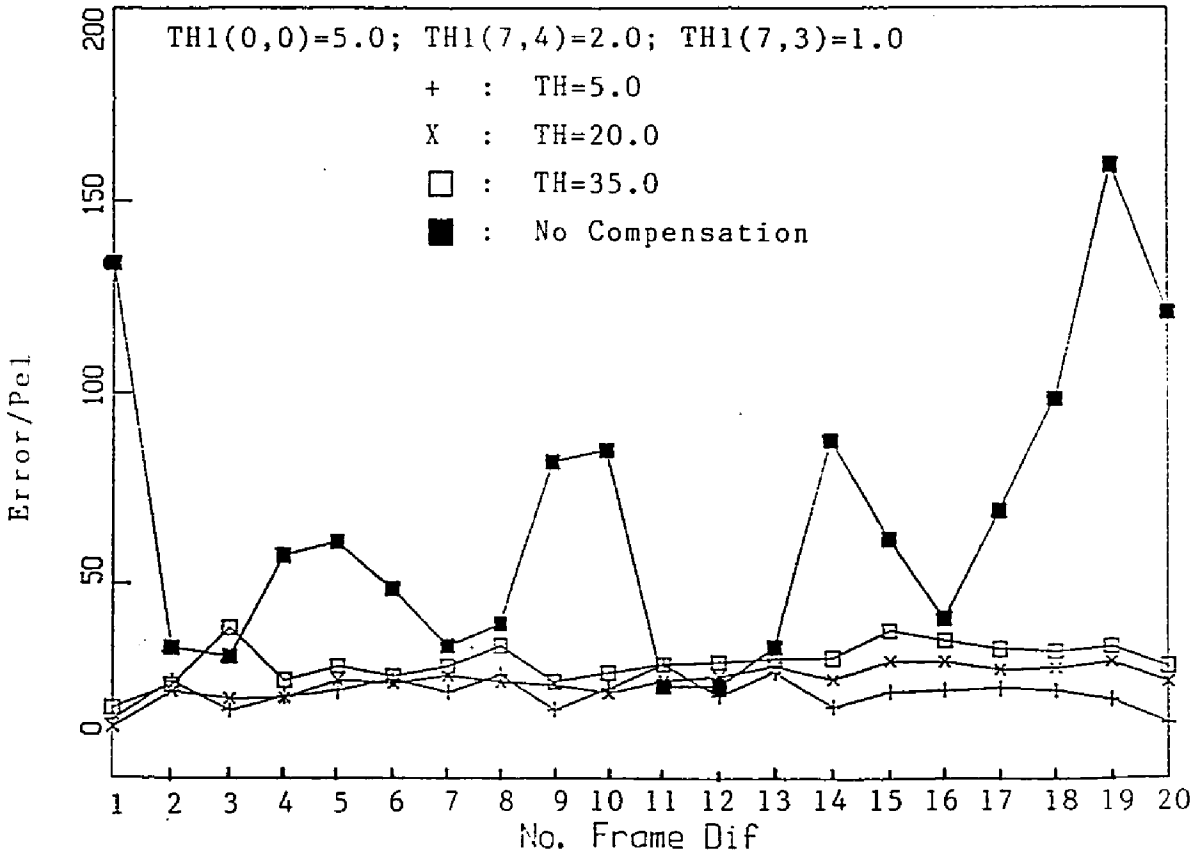


Fig.4.2.11: Comparison of error/pel for reconstructed video frames for the man speaking sequence

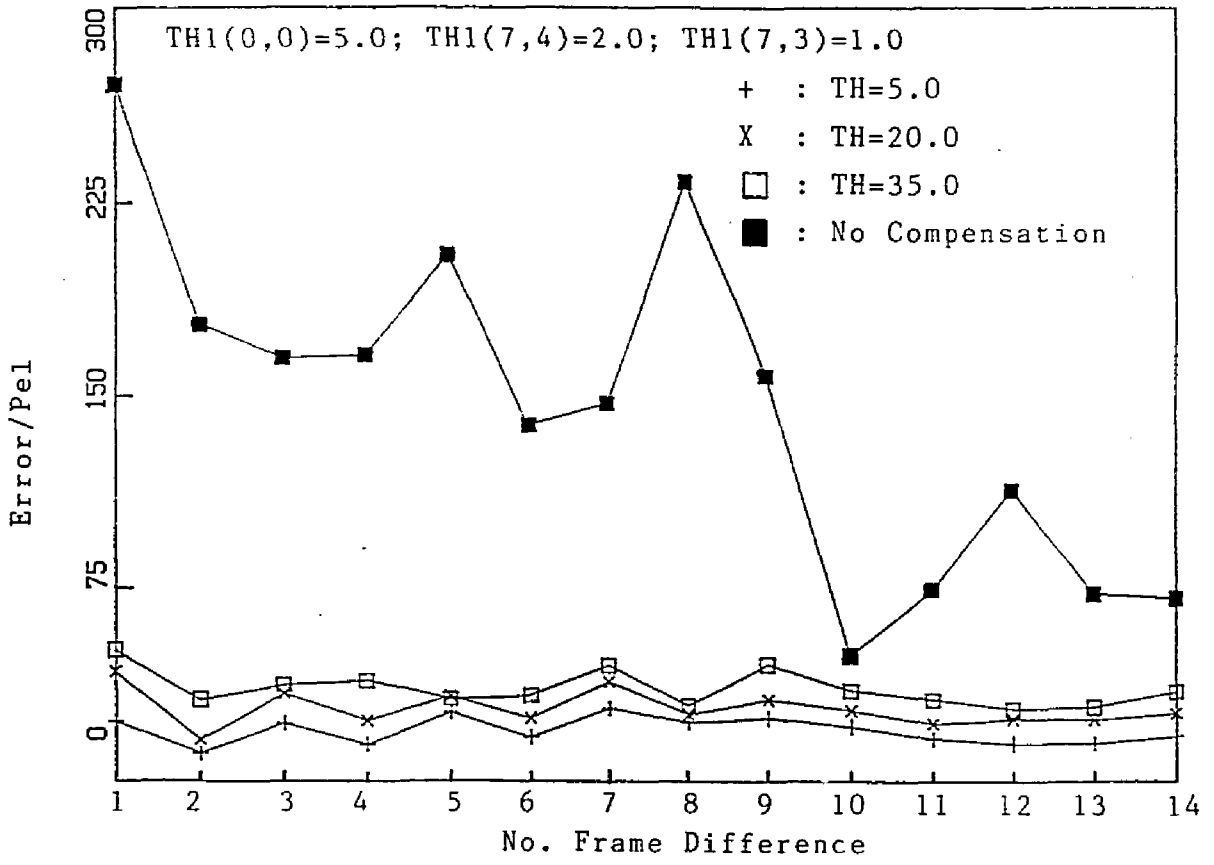
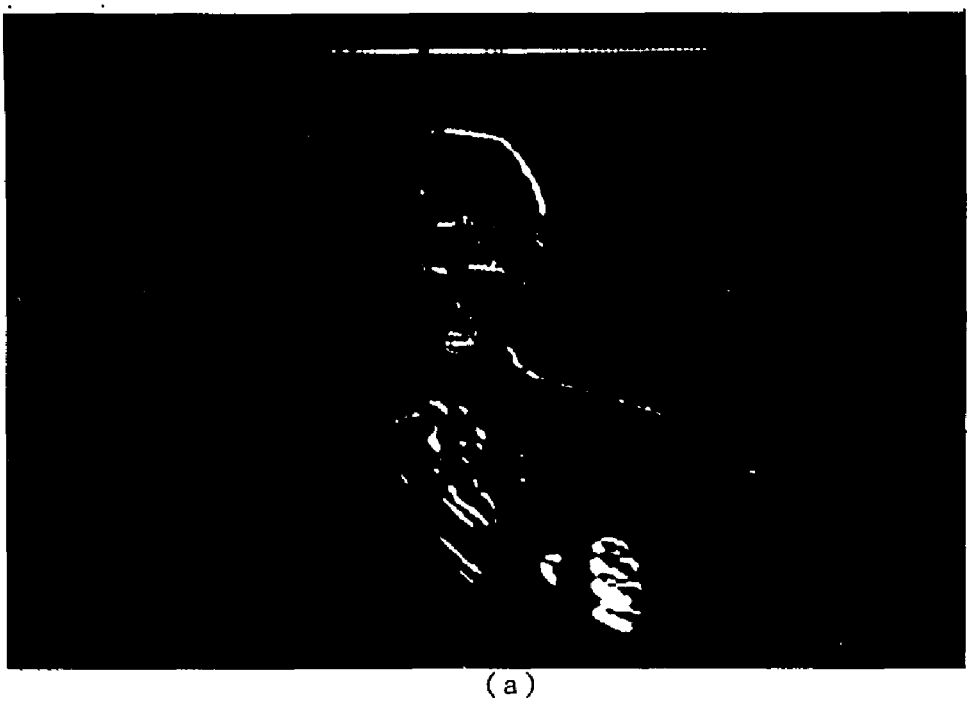
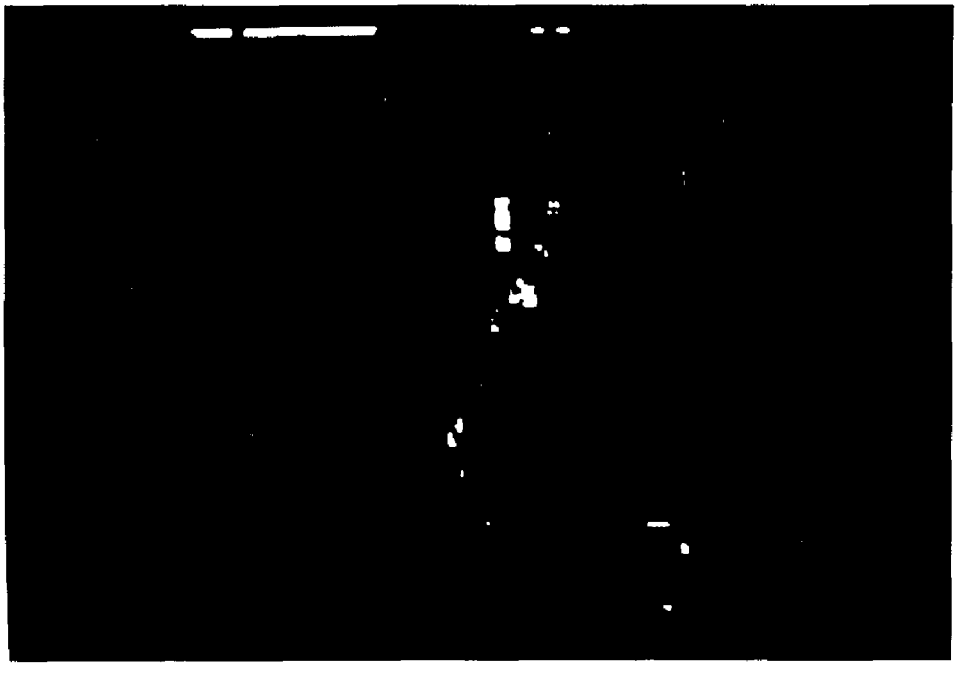


Fig.4.2.12: Comparison of error/pel for reconstructed video frames for the moving color cube sequence

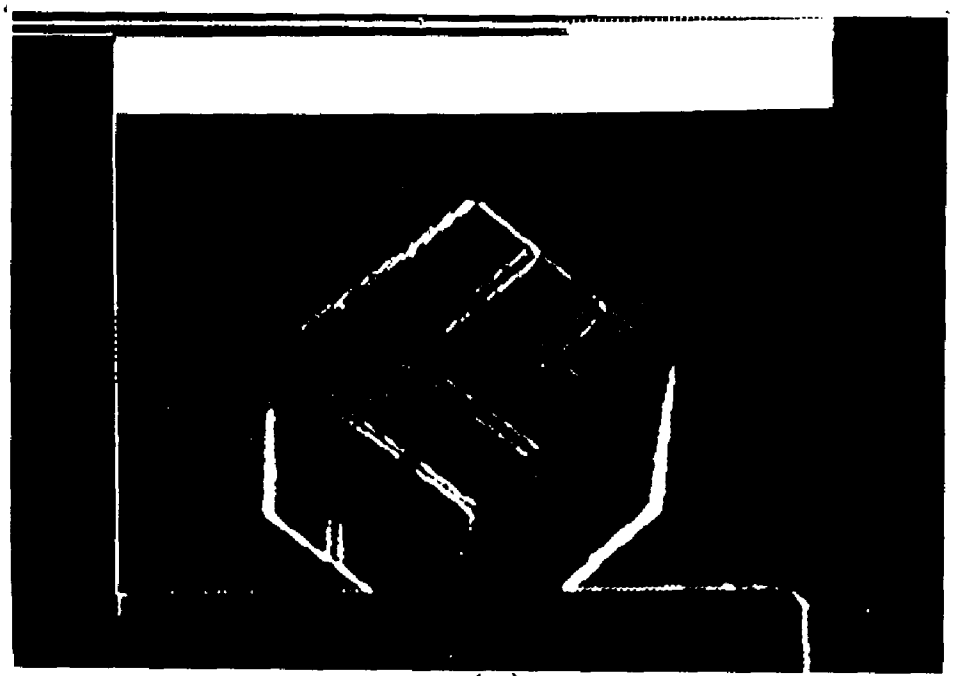


(a)



(b)

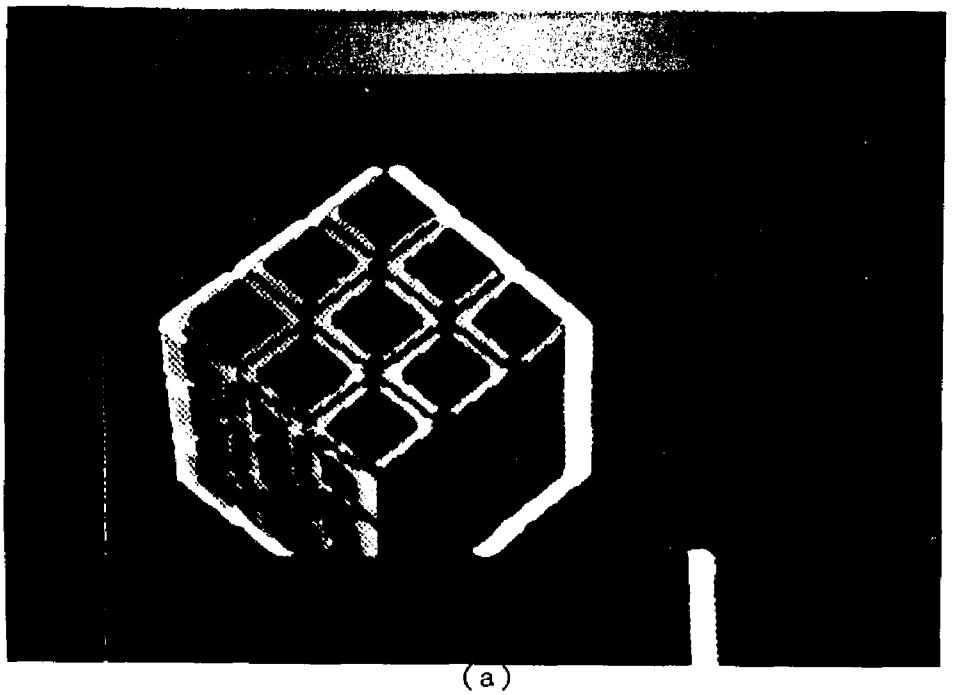
Fig.4.2.13: Error frames for the man speaking sequence when: a) the previous frame is repeted; b) motion detection, estimation and compensation algorithm in the Hadamard domain is employed



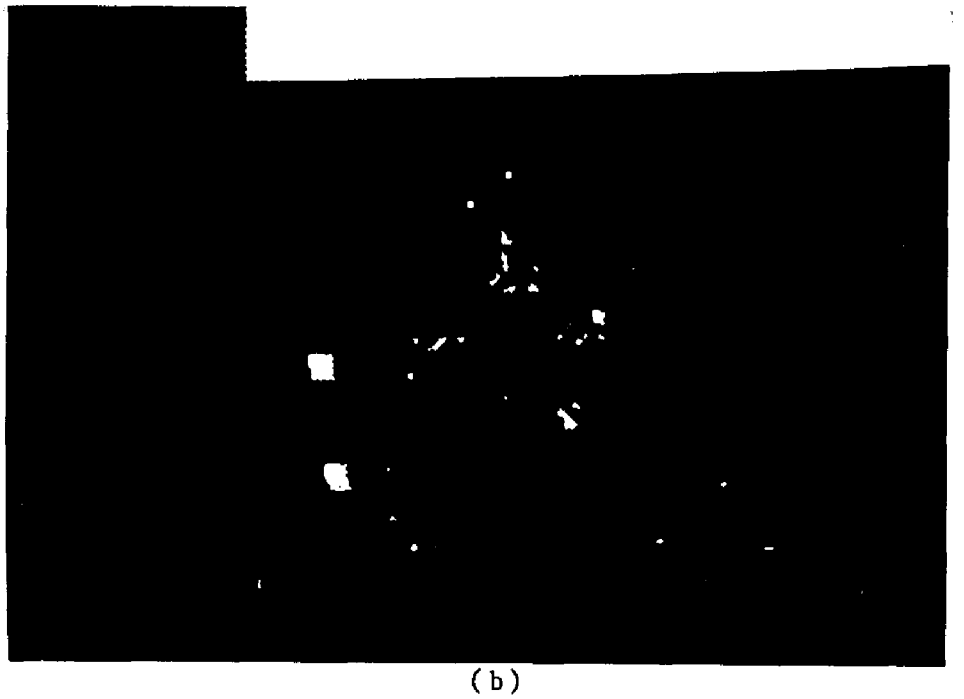
(a)

(b)

Fig.4.2.14: Error frames for the moving color cube sequence when: a) the previous frame is repeted; b) motion detection, estimation and compensation algorithm in the Hadamard domain is employed

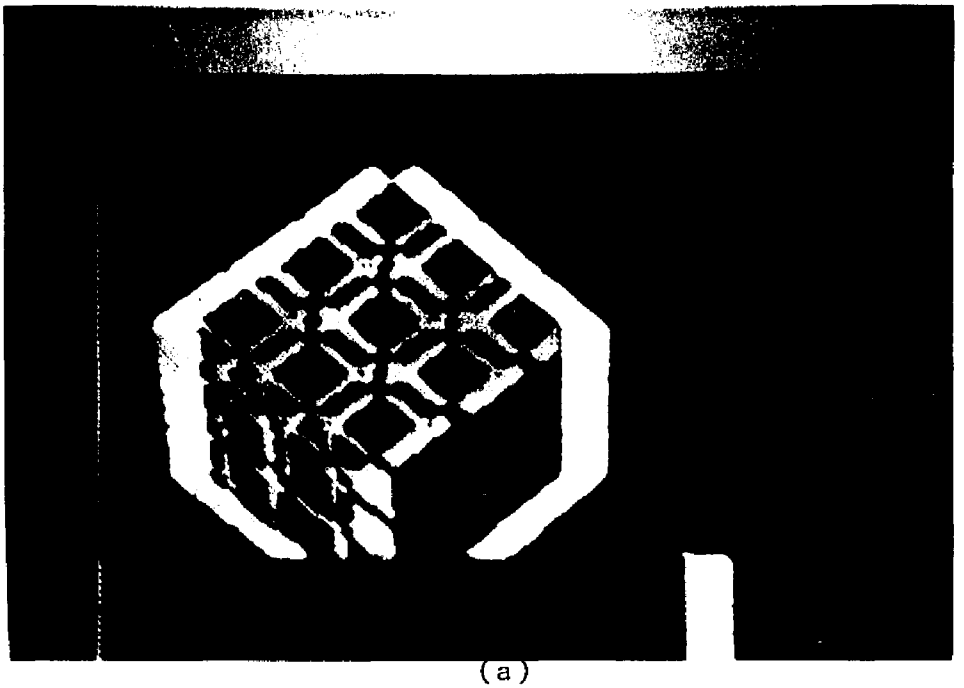


(a)

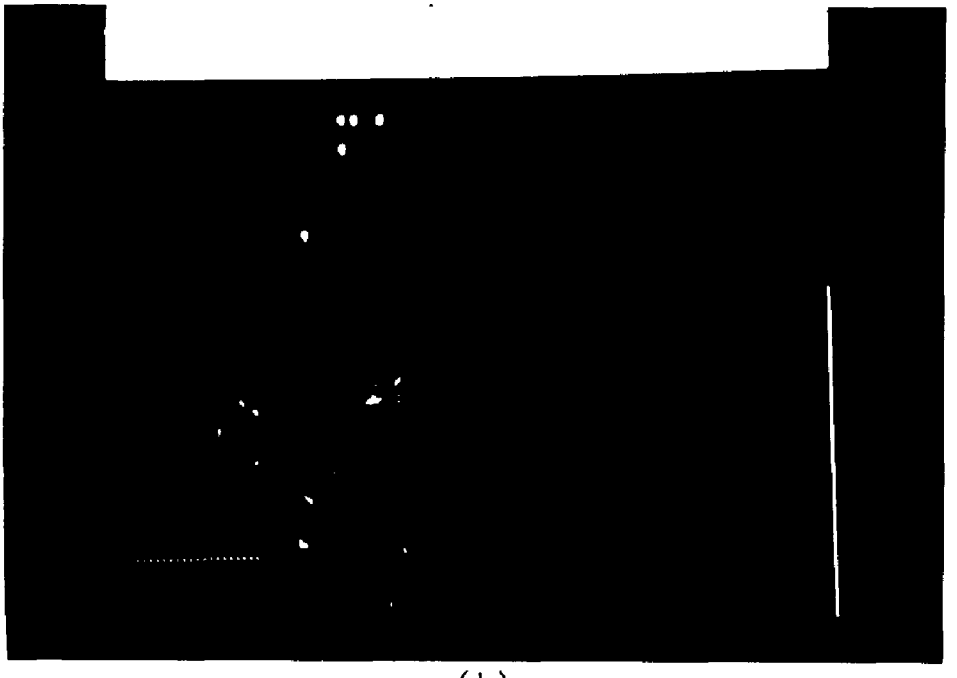


(b)

Fig.4.2.15: Error frames for the moving color cube sequence when 4:1 temporal subsampling is employed and when: a) the previous frame is repeted; b) motion detection, estimation and compensation algorithm in the Hadamard domain is employed



(a)



(b)

Fig.4.2.16: Error frames for the moving color cube sequence when 8:1 temporal subsampling is employed and when: a) the previous frame is repeted; b) motion detection, estimation and compensation algorithm in the Hadamard domain is employed

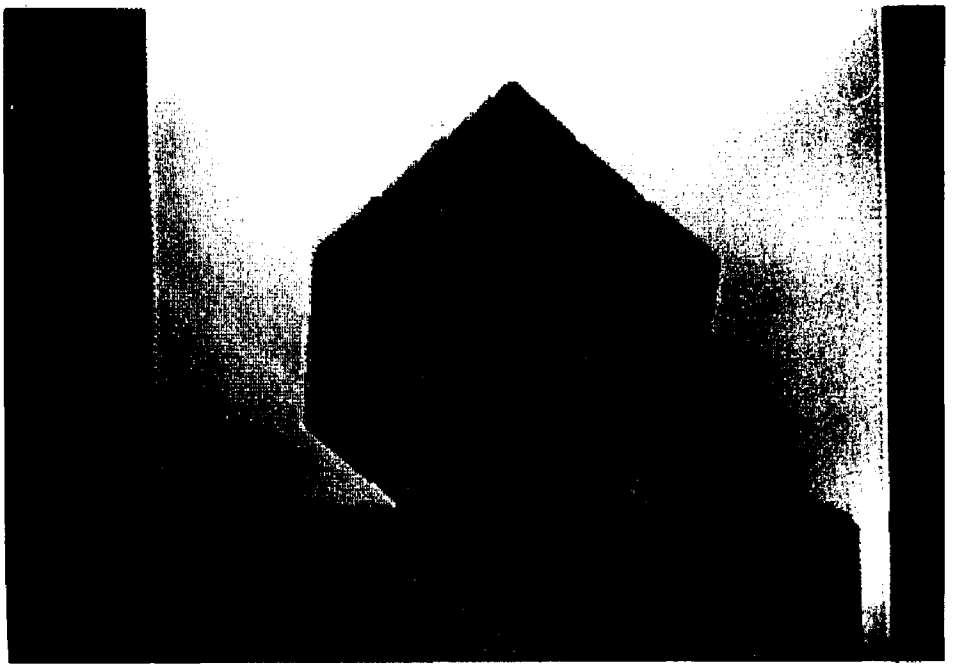


Fig.4.2.17: Reconstructed video frame in the moving color cube sequence (No coefficient is upgraded)

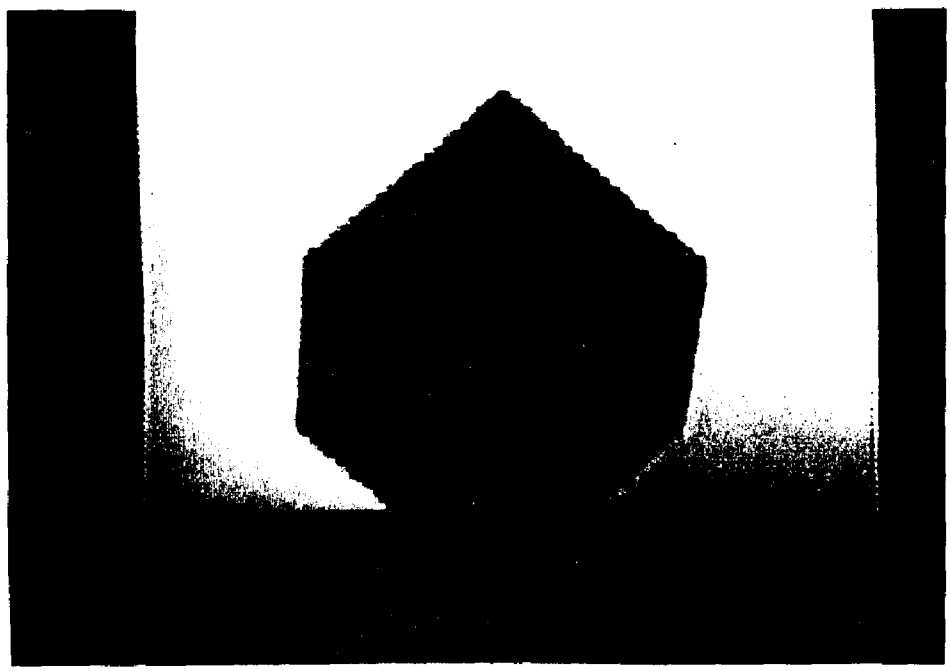


Fig.4.2.18: Reconstructed video frame in the moving color cube sequence (10 coefficients are upgraded)

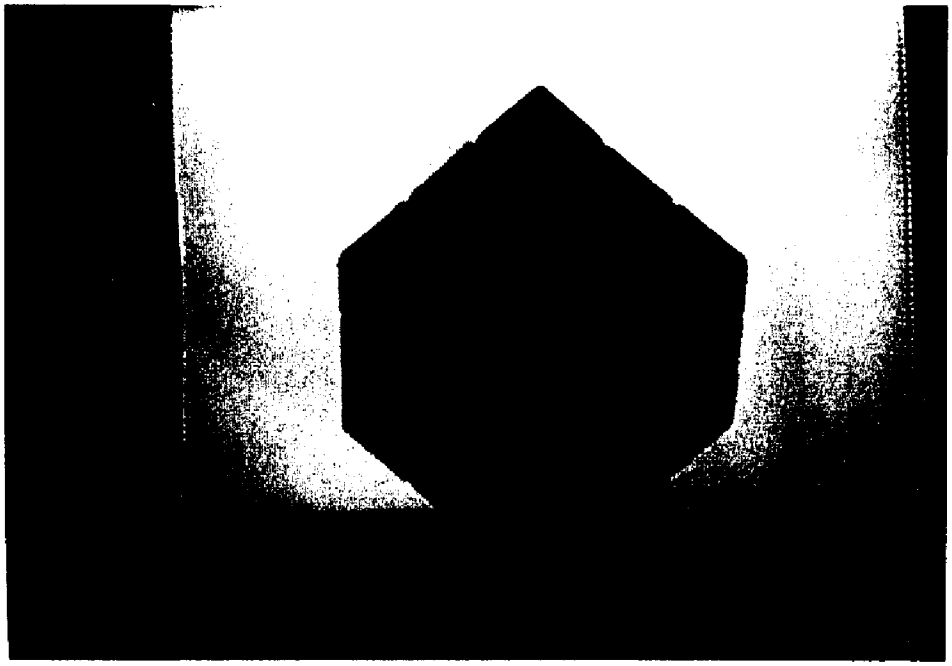
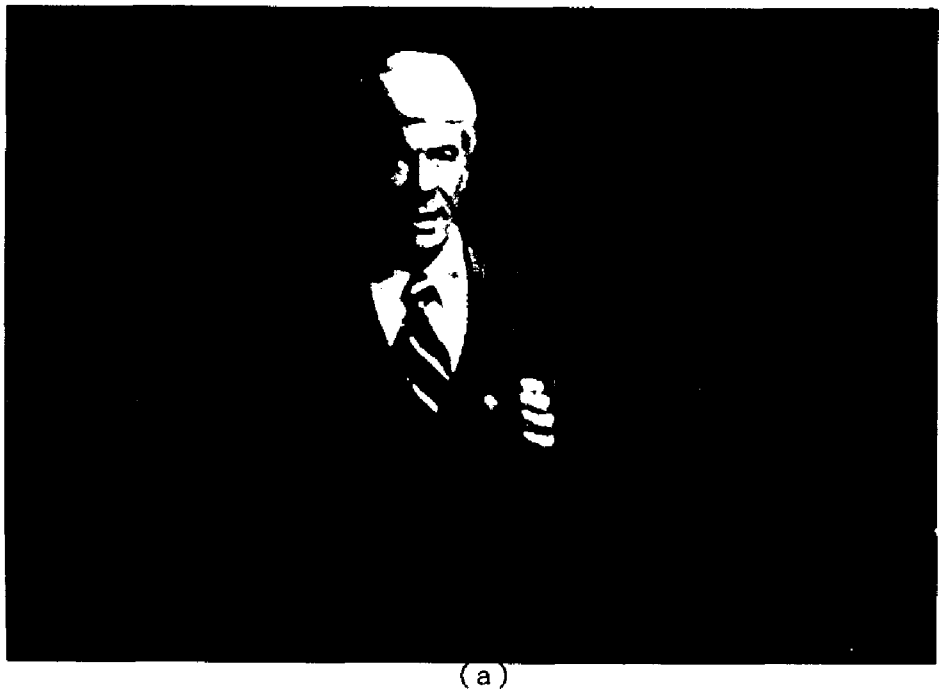
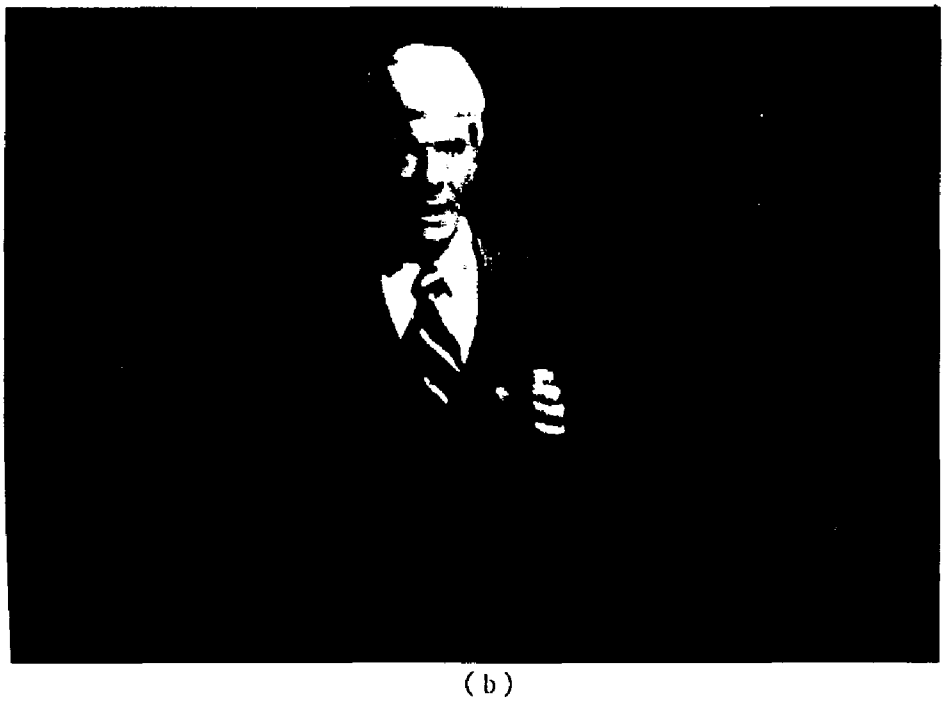


Fig.4.2.19: Reconstructed video frame in the moving color cube sequence (64 coefficients are upgraded)



(a)



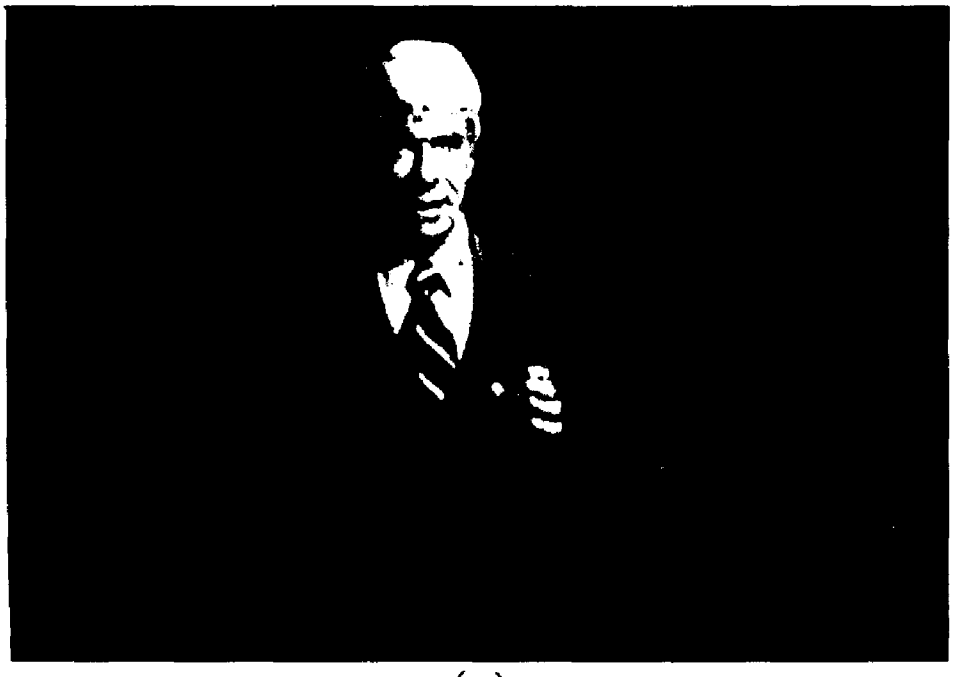
(b)

Fig.4.2.20: Comparison of video frames in the man speaking sequence: a) original; b)- j) reconstructions

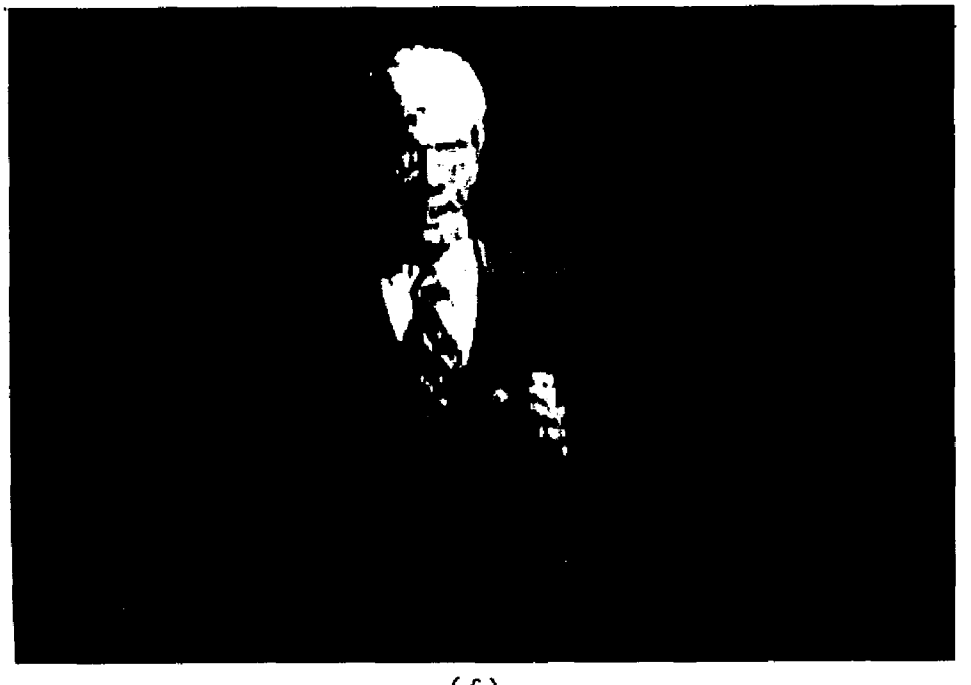
(c)

(d)

Fig.4.2.20 (cont.)

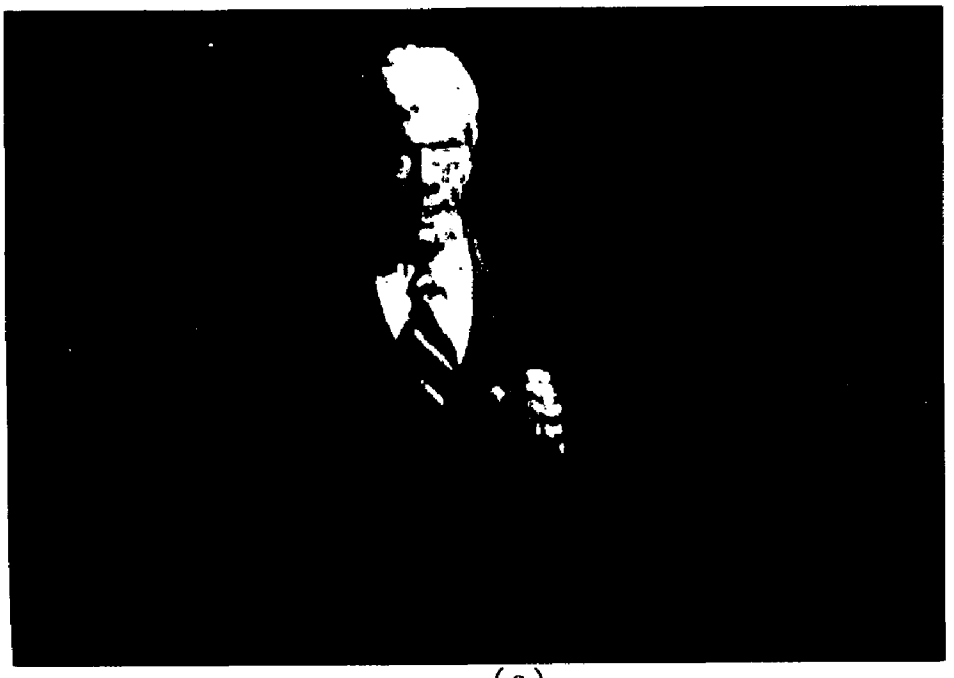


(e)

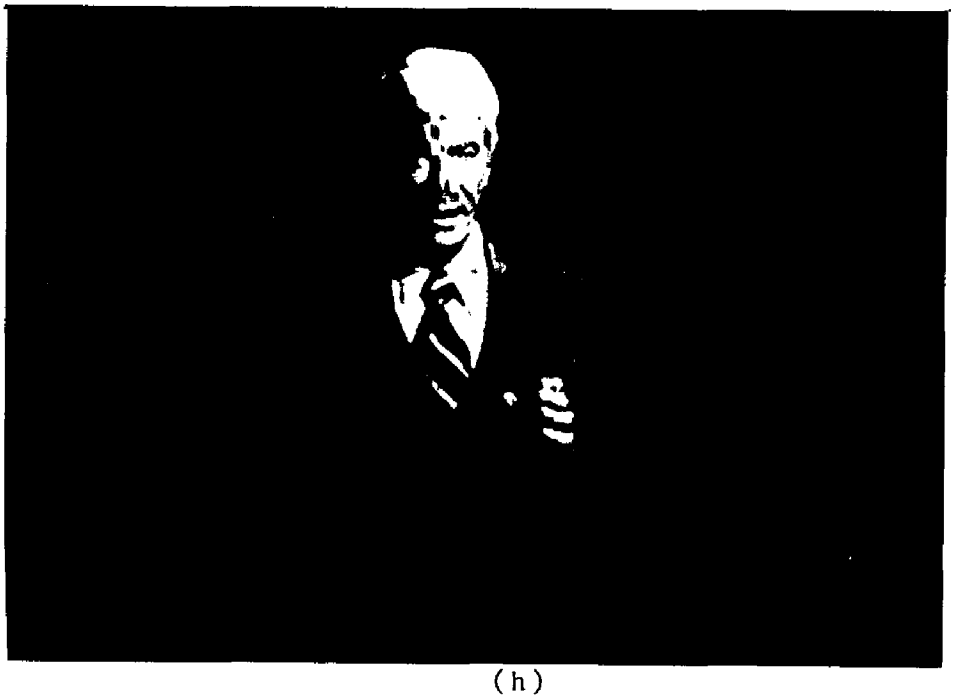


(f)

Fig.4.2.20 (cont.)



(g)



(h)

Fig.4.2.20 (cont.)

(i)

(j)

Fig.4.2.20 (cont.)

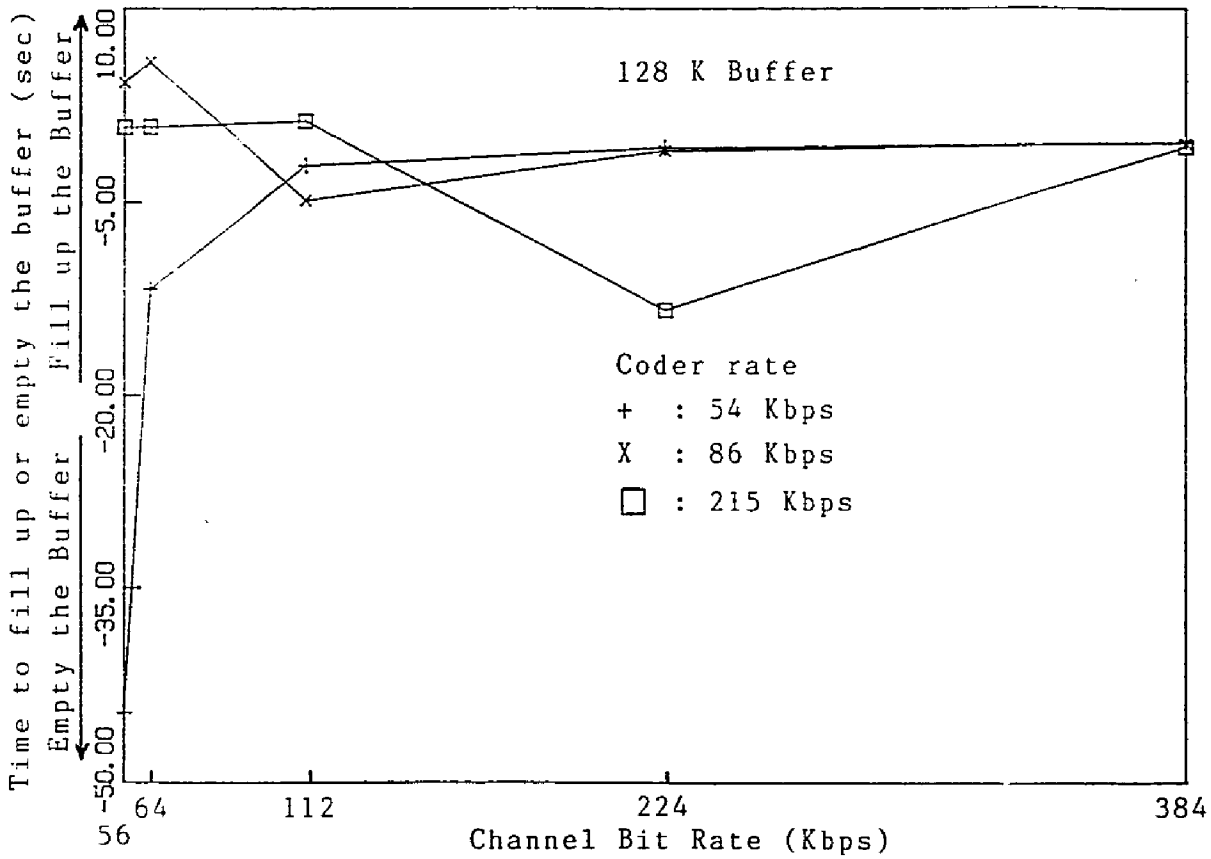


Fig.4.2.21: Plot of the time necessary to fill up or empty the buffer (128 K) for different coder and channel rates (the man speaking sequence)

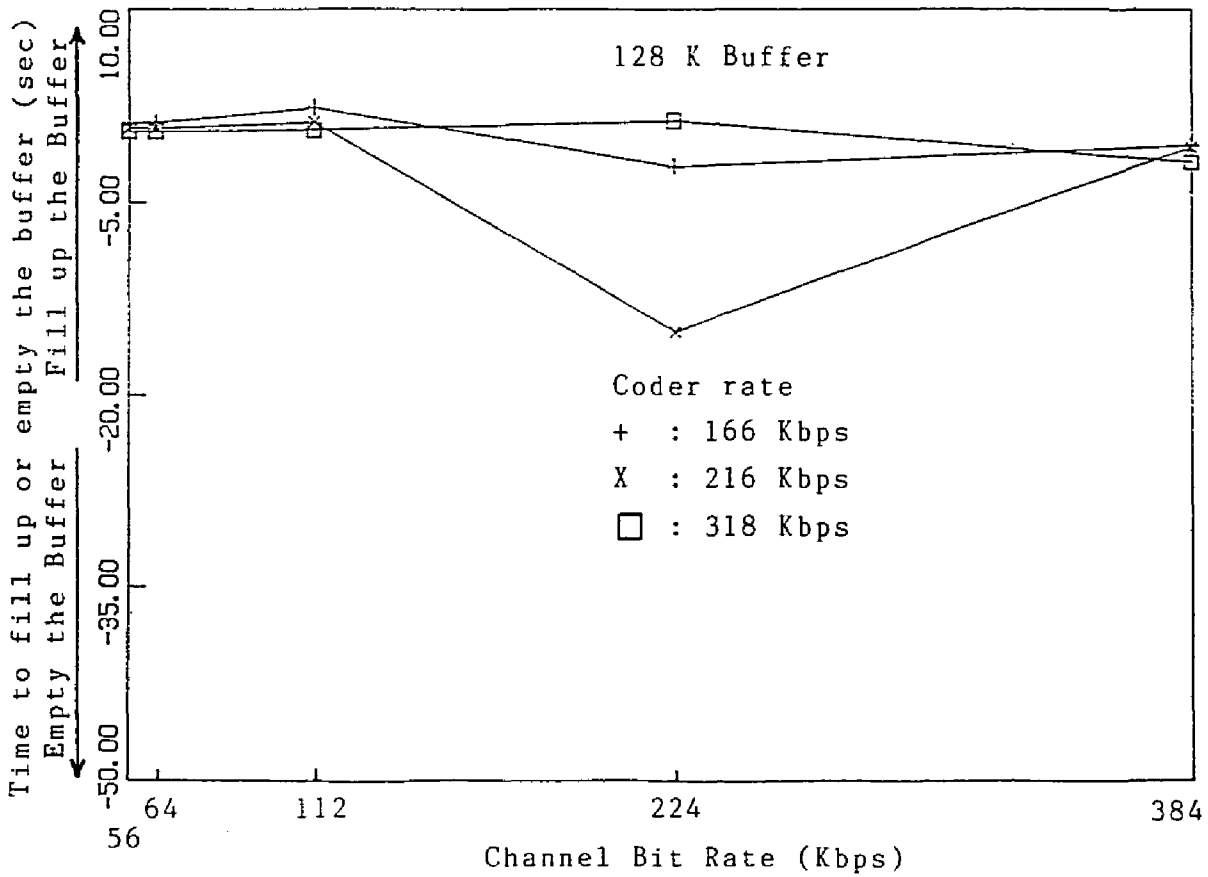


Fig.4.2.22: Plot of the time necessary to fill up or empty the buffer (128 K) for different coder and channel rates (the moving color cube sequence)

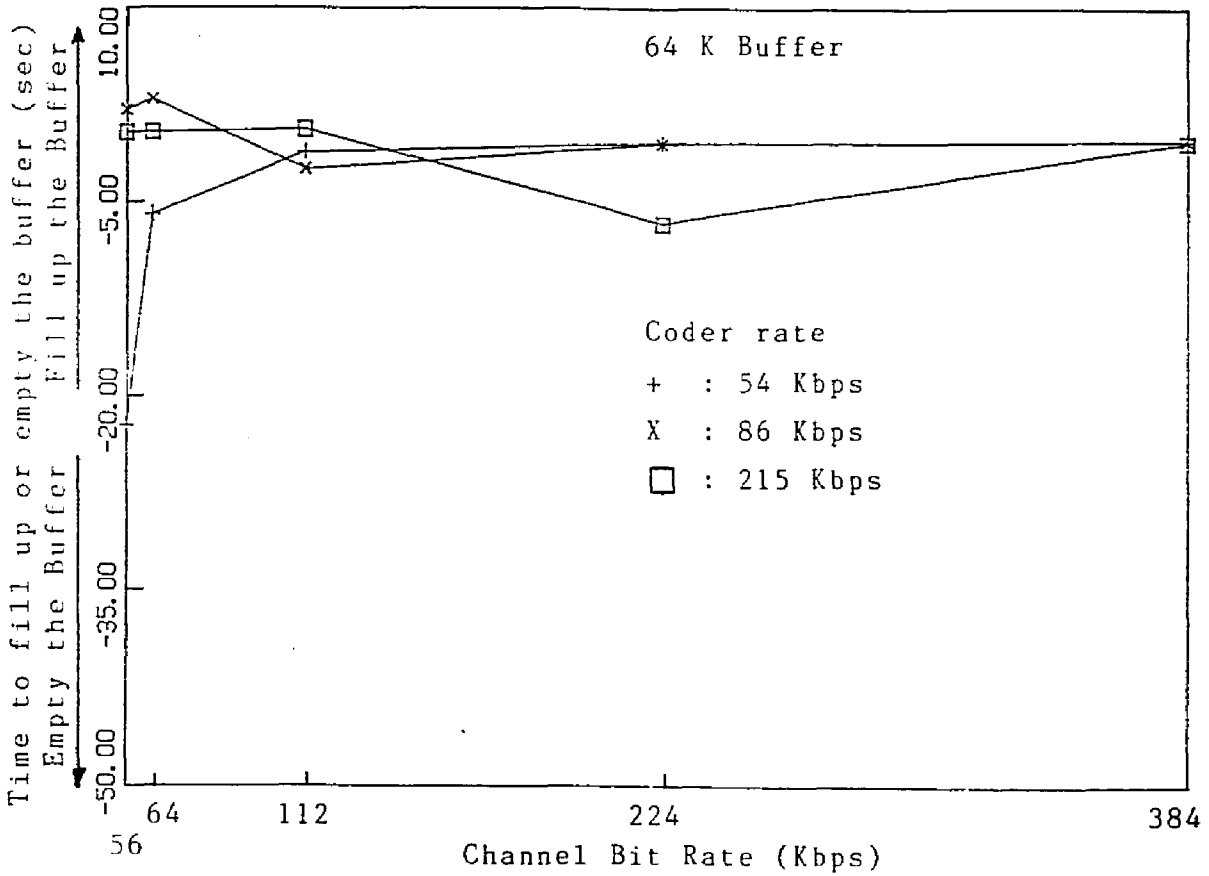


Fig.4.2.23: Plot of the time necessary to fill up or empty the buffer (64 K) for different coder and channel rates (the man speaking sequence)

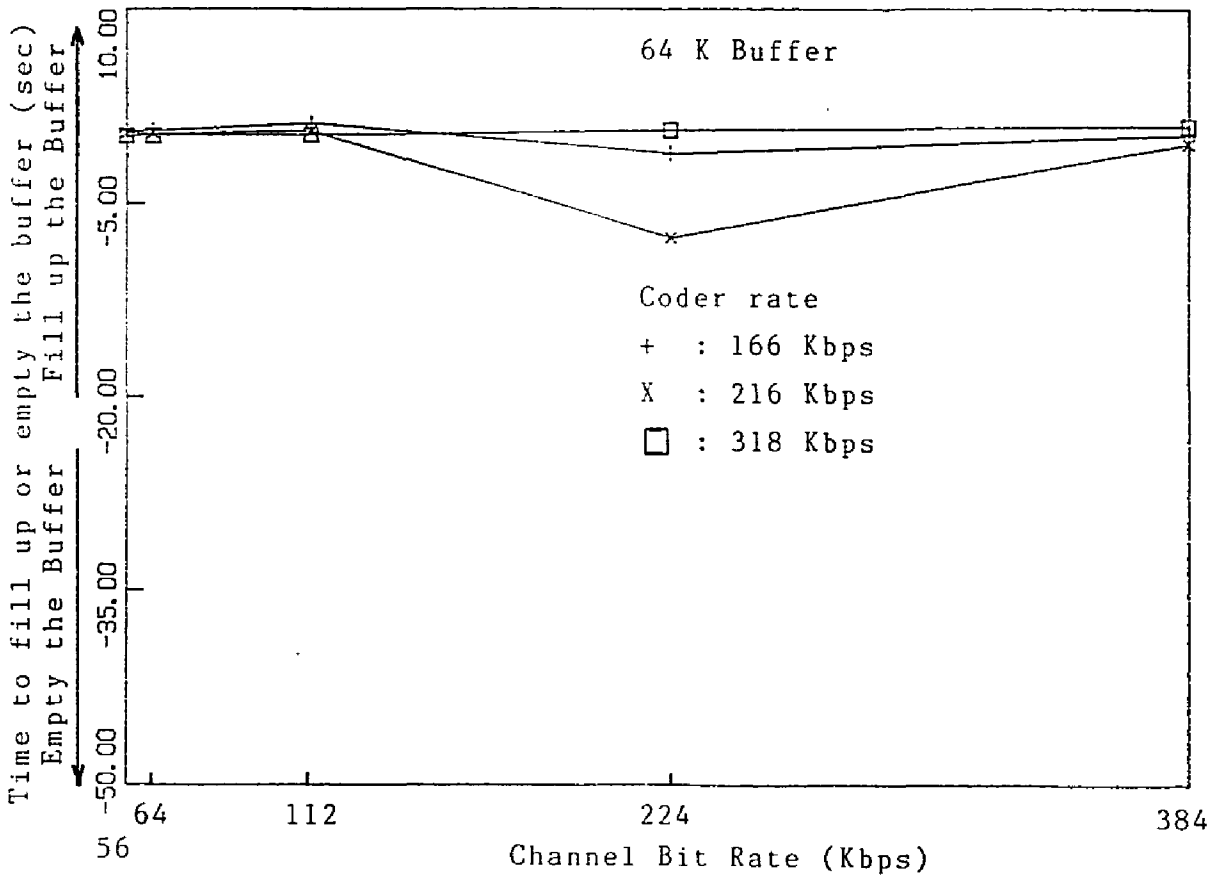


Fig.4.2.24: Plot of the time necessary to fill up or empty the buffer (64 K) for different coder and channel rates (the moving color cube sequence)

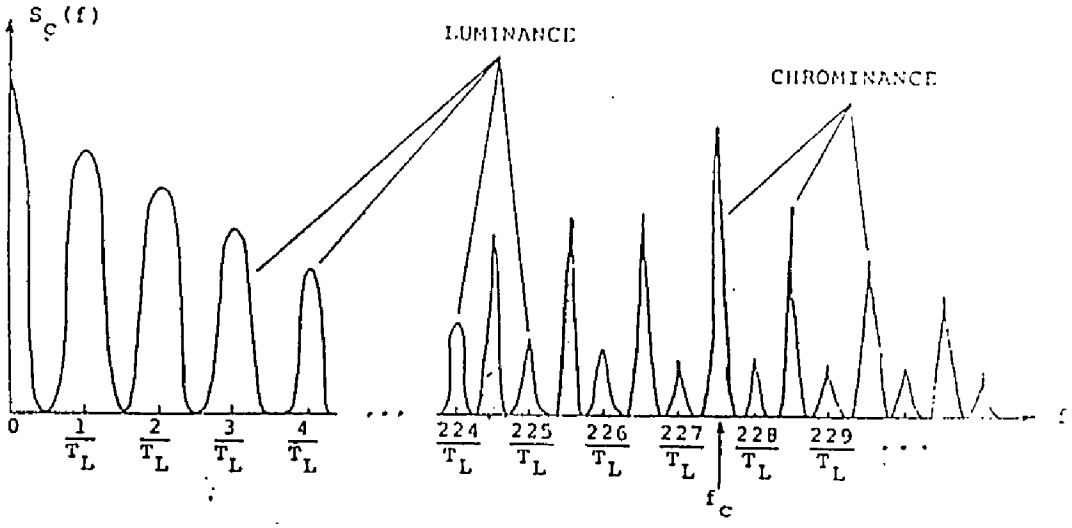


Fig.A.1: The interleaved spectrum of the NTSC composite signal

9 REFERENCES

[1] R. C. Gonzales and P. Wintz, " Digital Image Processing", Addison-Wesley Publishing Company, Inc., 1983.

[2] A. N. Netravali and B. G. Haskell, " Picture Coding, Principles and Applications ", Course Notes, C.C.N.Y., 1983.

[3] H. Taub and D. L. Schilling, " Principles of Communication Systems ", McGraw Hill.

[4] R. J. Clarke, " Transform Coding of Images ", Academic Press, 1985.

[5] A. N. Netravali and J. O. Limb, " Picture Coding: A Review ", (Invited Paper), Proc. IEEE, vol. 68, pp. 366 - 406, March 1980.

[6] W. M. Goodall, " Television by pulse code modulation", BSTJ, vol. 30, pp. 33 - 49, Jan. 1951.

[7] B. G. Haskell, " Entropy measurements for nonadaptive and adaptive, frame-to-frame, linear predictive coding of video telephone signals ", BSTJ, vol. 54, no. 6, pp. 1155 - 1174, August 1975.

[8] E. L. Hall, " Computer Image Processing and Recognition ", Academic Press, 1979.

[9] J. Barba, N. Scheinberg, D. L. Schilling, J. Garodnick and S. Davidovici, " A modified adaptive delta modulator ", IEEE Trans. Commun., vol. COM-29, no. 12, pp. 1767 - 1786, Dec. 1981.

[10] T.-L. R. Lei, N. Scheinberg and D. L. Schilling, " Adaptive delta modulation systems for video encoding ", IEEE Trans. Commun., vol. COM-25, no. 11, pp. 1302 - 1314, Nov. 1977.

[11] F. Kretz and D. Nasse, " Digital television: Transmission and Coding ", (Invited Paper), Proc. IEEE, vol. 73, no. 4, pp. 575 - 591, April 1985.

[12] H. G. Musmann, P. Pirsch and H-J. Grallert, " Advances in Picture Coding ", (Invited Paper), Proc. IEEE, vol. 73, no. 4, pp. 523 - 548, April 1985.

[13] D. J. Connor and J. O. Limb, " Properties of frame-difference signals generated by moving images ", IEEE Trans. Commun., vol. COM-22, pp. 1564 - 1575, Oct. 1974.

[14] W. Zschunke, " DPCM picture coding with adaptive prediction ", IEEE Trans. Commun., vol. COM-25, no. 11, pp. 1295 - 1302, Nov. 1977.

[15] C. Cafforio and F. Rocca, "Methods for measuring small displacements of television images ", IEEE Trans. Inform. Theory, vol. 11-22, pp. 573 - 579,Sept. 1979.

[16] A. N. Netravali and J. D. Robbins, " Motion compensated television coding - Part I ", BSTJ, pp. 631 - 670, Mar. 1979.

[17] J. Woods, " Stability of DPCM coders for television ", IEEE Trans. Commun., vol. COM-23, pp. 845 - 846, Aug. 1975.

[18] A. N. Netravali, " On quantizers for DPCM coding of pictures signals ", IEEE Trans. Inform. Theory, vol. IT-18, no. 3, pp. 360 - 370, May 1977.

[19] A. N. Netravali and B. Prasada, " Adaptive quantization of picture signals using spatial masking ", Proc. IEEE, vol. 65, pp. 536 - 548, Apr. 1977.

[20] P. A. Wintz, " Transform picture coding ", Proc. IEEE, vol. 60, no. 7, pp. 809 - 820, July 1972.

[21] W. Chen, C. H. Smith and S. Fralick, "A fast computational algorithm for the discrete cosine transform", IEEE Trans. Commun., pp. 1004 - 1009, Sept. 1977.

[22] S. C. Knauer, " Real time video compression algorithm for hadamard transform processing ", Proc. SPIE, vol. 66, pp. 58 - 69, Aug. 1975.

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

[24] F. W. Mounts, A. N. Netravali and B. Prasada, " Design of quatizers for real-time hadamard-transform of pictures ", BSTJ, vol. 56, no. 1, pp. 21 - 48, Jan. 1977.

[25] W. H. Chen and C. N. Smith, " Adaptive coding of monochrome and color images ", IEEE Trans. Commun., vol. COM-25, pp. 1285 - 1292, Nov. 1977.

[26] J. A. Stuller and A. N. Netravali, " Transform domain motion estimation ", BSTJ, pp. 1673 - 1702, Sept. 1979.

[27] A. N. Netravali, " Motion compensated transform coding", BSTJ, pp. 1703 - 1718, Sept. 1979.

[28] A. N. Netravali, " Interpolative picture coding using a subjective criterion ", IEEE Trans. Commun., vol. COM-25, pp. 503 - 508, May 1977.

[29] T. Ohira, M. Hayakawa and K. Matsumoto, "Orthogonal transform coding system for NTSC color television signal ", Proc. Int. Communications Conf., pp. 4B.3-86 - 4B.3-90, 1977.

[30] J. C. Candy, M. A. Franke, B. G. Haskell and F. W. Mounts, " Transmitting television as clusters of frame-to-frame differences ", BSTJ, vol. 50, pp. 1889 - 1919, July-Aug 1971.

[31] B. G. Haskell, " Buffer and channel sharing by several interframe picturephone coders ", BSTJ, vol. 51, no.1, pp. 261 - 289, Jan. 1972.

[32] A. Habibi, " Hybrid coding of pictorial data ", IEEE Trans. Commun., vol. COM-22, pp. 614 - 624, May 1974.

[33] M. Miyahara, " Analysis of perception of motion in television signals and its application to bandwidth compression", IEEE Trans. Commun., vol. COM-23, pp. 761 - 766, July 1975.

[34] F Kretz, " Subjectively optimal quantization of pictures ", IEEE Trans. Commun., vol. COM-23, pp 1288 - 1292, Nov. 1975.

[35] A. Jain, " A fast Karhunen-Loeve transform for a class of stochastic process ", IEEE Trans. Commun., vol. COM-24, pp. 1023 - 1029, Sept. 1976.

[36] A. Habibi, " Survey of adaptive image coding techniques ", IEEE Trans. Commun., vol. COM-25, no. 11, pp. 1275 - 1284, Nov. 1977.

[37] E. R. Kretzmer, " Statistics of Television signals ", BSTJ, pp. 751 - 763, July 1952.

[38] F. W. Mounts, " A video encoding system with conditional picture-element replenishment ", BSTJ, pp. 2545 - 2554, Sept. 1969.

[39] J. O. Limb, " Buffering of data generated by the coding of moving images ", BSTJ, vol. 51, no.1 , pp. 239 - 259, Jan. 1971.

[40] W. K. Pratt, " Spatial transform coding of color images ", IEEE Trans. Commun., vol. COM-19, no. 6, pp. 980 - 992, Dec. 1971.

[41] J. C. Candy and R. H. Bosworth, " Methods for designing differential quantizers based on subjective evaluation of edge busyness ", BSTJ, vol.51, no. 7, pp. 1495 - 1516, Sept 1972.

[42] D. J. Connor, B. G. Haskell and F. W. Mounts, " A frame-to-frame picturephone coder for signals containing differential quatizing noise ", BSTJ, vol.52, no. 1, pp. 35 - 51, Jan. 1973.

[43] J. O. Limb and J. A. Murphy, " Measuring the speed of moving objects from television signals ", IEEE Trans. Commun., pp. 474 - 478, Apr. 1975.

[44] J. J. Knab, " Effects of round-off noise on Hadamard transformed imagery ", IEEE Trans. Commun., vol COM-25, no.11, pp. 1292 - 1294, Nov. 1977.

[45] T. Raj Natarajan and N. Ahmed, " On interframe transform coding ", IEEE Trans. Commun., vol. Com-25, no. 11, pp. 1323 - 1329, Nov. 1977.

[46] B. G. Haskell, P. L. Gordon, R. L. Schmidt and J. V. Scattaglia, " Interframe coding of 525-line, monochrome television at 1.5 Mbits/s ", IEEE Trans. Commun., vol. COM-25, no. 11, pp. 1339 - 1348, Nov. 1977.

[47] A. G. Tescher, " Transform coding strategies at low rates ", NTC'81, New Orleans, LA, pp. C9.2.1 - C9.2.3, Nov-Dec. 1981.

[48] C. Ekambaram and S. C. Kwatra, " A new architecture for adaptive transform compression of NTSC composite video signal", NTC'81, New Orleans, LA, pp. C9.6.1 - C9.6.4, Nov.-Dec. 1981.

[49] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, " Motion-compensated interframe coding for video conference ", NTC'81, New Orleans, LA, pp. G5.3.1 - G5.3.5, Nov.-Dec. 1981.

[50] J. D. Robbins and A. N. Netravali, " Spatial subsampling in notion-compensated television coders ", BSTJ, vol. 61, no. 8, pp. 1895 - 1910, Oct. 1982.

[51] Y. Ninomiya and Y. Ohtsuka, " A motion-compensated interframe coding scheme for television pictures ", IEEE Trans. Commun., vol. COM-30, no. 1, pp. 201 - 212, Jan. 1982.

[52] S. Brofferio, C. Cafforio, M. Piacentini and F. Rocca, " Motion compensation in teleconference ", ICC'82, Philadelphia, PA, pp. 2G.6.1 - 2G.6.5, June 1982.

[53] K. A Prabhu and A. N. Netravali, " Motion compensated component color coding ", IEEE Trans. Commun., vol. COM-30, no. 12, pp. 2519 - 2527, Dec. 1982.

[54] K. A. Prabhu and A. N. Netravali, " Motion compensated composite color coding ", IEEE Trans. Commun., vol. COM-31, no. 2, pp. 216 - 224, Feb. 1983.

[55] C. E. Li and K. R. Rao, " Composite-component transformation and predictive coding of the component color TV signal ", IEEE Journal on Selected Areas in Communications, vol. SAC-2, no. 2, pp. 353 - 358, March 1984.

[56] C. M. Lin and S. C. Kwatra, " An adaptive algorithm for motion compensated color image coding ", Proc. IEEE GLOBECOM, Atlanta, GA, pp. 47.1.1 - 47.1.4, Nov. 1984.

[57] A. N. Netravali and J. Salz, " Algorithms for estimation of Three-Dimensional Motion ", AT&TTJ, vol. 64, no. 2, pp. 335 - 344, Feb. 1985.

[58] D. R. Davies, K. Baughan and T. Kent, " ISDN signalling standards ", Br Telecom Technol J, vol. 4, no. 2, pp. 26 - 31, Apr. 1986.

[59] N. D. Kenyon, " Audiovisual telecommunications services - a unified approach ", Br Telecom Technol J, vol. 3, no. 2, pp. 5 - 12, Apr. 1985.

[60] H. Kuroda, N. Mukawa, T. Matsuoka and S. Okubo, " 1.5 Mbit/s interframe codec for video teleconferencing signals ", IEEE GLOBECOM'82, pp. E2.5.1 - E2.5.5., 1982

[61] D. N. Hein, "Video compression using conditional replenishment and motion prediction ", Ph.D. Dissertation, Kansas State University, Manhattan, Kansas 1981.

[62] R. W. Hamming, "Coding and Information theory", Prentice - Hall, 1980.

[63] R. G. Gallager, "Information theory and reliable communication ", Wiley, 1968.

[64] S. Sabri and B. Prasada, " Video conferencing systems", Proc. IEEE, vol. 73, no. 4, pp. 671 - 688, Apr. 1985.

[65] L. Weiss, I. M. Paz, and D. L. Schilling, " Video encoding using an adaptive digital delta modulator with overshoot suppression ", IEEE Trans. Commun., vol.COM-23, no. 9, pp. 905 - 920, Sept. 1975.

[66] N. Scheinberg and D. L. Schilling, " Techniques for correcting transmission errors in video adaptive delta modulation channels ", IEEE Trans. Commun., pp. 1064 - 1070, Sept. 1976.

[67] D. L. Schilling, N. Scheinberg, and J. Garodnick, "Video encoding using adaptive delta modulator", IEEE Trans. Commun., vol.Com-26, no. 11, pp. 1682 - 1689, Nov. 1978.

[68] N. Scheinberg, E. Feria, J. Barba, and D. L. Schilling, " A one-stage look-ahead algorithm for delta modulators ", IEEE Trans. Commun., vol.COM-32, no. 7, pp. 861 - 863, July 1984.

[69] C. L. Song, J. Garodnick, and D. L. Schilling, " A variable-step-size robust delta modulator ", IEEE Trans. Commun. Tech., pp. 1033 - 1044, Dec. 1971.

[70] J. O. Limb, C. B. Rubinstein, J. E. Thompson, "Digital Coding of Video Signals - A Review", IEEE Trans. Commun., vol.COM-25, no. 11, pp. 1349 - 1384, Nov. 1977.

[71] A. K. Jain, "Image data compression: A review", Proc. IEEE, vol. 69, no. 3, pp. 349 - 389, Mar. 1981.

[72] J. D. Robbins and A. N. Netravali, " Interframe television coding using movement compensation ", ICC '79, pp. 23.4.1 - 23.4.5, June 1979.

[73] K. Sawadwa and H. Kotera, "32 Mbit/s Transmission of NTSC color signals by composite DPCM coding", IEEE Trans. Commun., vol.COM-26, no. 10, pp. 1432 - 1439, Oct. 1978.

[74] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding", IEEE Trans. Commun., vol.COM-29, no. 12, pp. 1799 - 1808, Dec. 1981.

[75] A. Ploysongsang, K.R. Rao, "DCT/DPCM Processing of NTSC Composite Video Signals," IEEE Trans. Commun., Vol. Com-30, pp. 541 - 549, March 1982.

[76] S. C. Kwatra, H.Fatmi, "NTSC Composite Video at 1.6 Bits/Pel," ICC 1983, pp. B4.6.1-B4.6.5.

[77] W. K. Pratt, " Image transmission techniques ", Academic Press, 1979.

[78] H. F. Harmuth, " Sequency theory foundations and applications ", Academic Press, 1977.

[79] B. Brob, " Basic Television Principles and Servicing", Fourth Edition, McGraw-Hill, 1975.

[80] D. J. Sauer, "Design and Performance of a CCD Comb Filter IC," RCA Review, Vol.41, pp. 29-56, March 1980.

[81] L. Stenger, "Digital Comb-Filter Demodulation of PAL Color Television Signals," IEEE Trans. Commun., Vol. COM-27, No.10, pp 1624-1631, Oct. 1979.

[82] K. Sawada, H. Kotera, " IEEE Trans. on Comm., Vol. COM-26, No.4, pp 458-465, April 1978.

[83] T.Ishiguro, K.Iinuma, Y.Iijima, T.Koga, S.Azami, T.Mune, " Composite Interframe Coding of NTSC Television Signals ", NTC, 1976.

[84] J. W. Wenthworth, "Color television engineering", McGraw-Hill,1955.

[85] E. H. Feria, "Predictive Transform Coding", IEEE Proc. NAECON, Vol. 1, pp. 45-52, May 1986.

[86] E. H. Feria, J. Barba and N. Scheinberg, "A Simple Predictive Transform Coder for Images", IEEE Proc. MILCOM, October 1986.