DIGITAL TELEVISION SYSTEM DESIGN STUDY

FINAL REPORT

Contract No. NAS 9-14863

Prepared for

NASA Lyndon B. Johnson Space Center
Houston, Texas 77058

Prepared by

Gaylord K. Huth
Axiomatix
13900 Panay Way, Suite 110M
Marina del Rey, California 90291

Axiomatix Report No. R7611-3
November 28, 1976

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

If an analog transmission system as shown in Figure 1 is to be utilized for television, the operations which must be performed on the video signal are typically rather straightforward and include premodulation filtering, modulation, post-modulation filtering, and possibly several stages of amplification. In the modulation process itself, the filtered video signal is used to vary (in some manner) the amplitude, phase, or frequency of the RF carrier. For applications such as commercial broadcasting, where bandwidth is at a premium, some form of AM (such as vestigial sideband, VSB) is generally used, while in other applications such as space communications, which allow a power-bandwidth trade, FM is commonly used. Regardless of the form of analog modulation employed, the noise (or some part of it) introduced in the channel appears at the demodulator output.

In recent years, there has been a definite trend towards use of digital instead of analog techniques for transmission of pictorial data, although the trend has been much more pronounced for single-frame images (still pictures) than for multi-frame images (television). The rapid growth in the use of digital computers, as well as the many inherent advantages of digital communications systems has undoubtedly resulted in this trend.

In digital transmission systems, the video signals are processed in some manner which includes quantization and coding such that they are separable from the noise introduced into the channel. The performance of digital television systems, then, is determined by the nature of the processing techniques (i.e., whether the video signal itself or, instead, something related to the video signal is quantized and coded) and to the quantization and coding schemes employed.

It should perhaps be emphasized that digital transmission systems are inherently more complex than analog systems and thus should probably not be considered for those applications in which analog transmission is satisfactory in terms of the power/bandwidth requirements. Digital techniques do offer some potential advantages, however, in certain circumstances. For example,

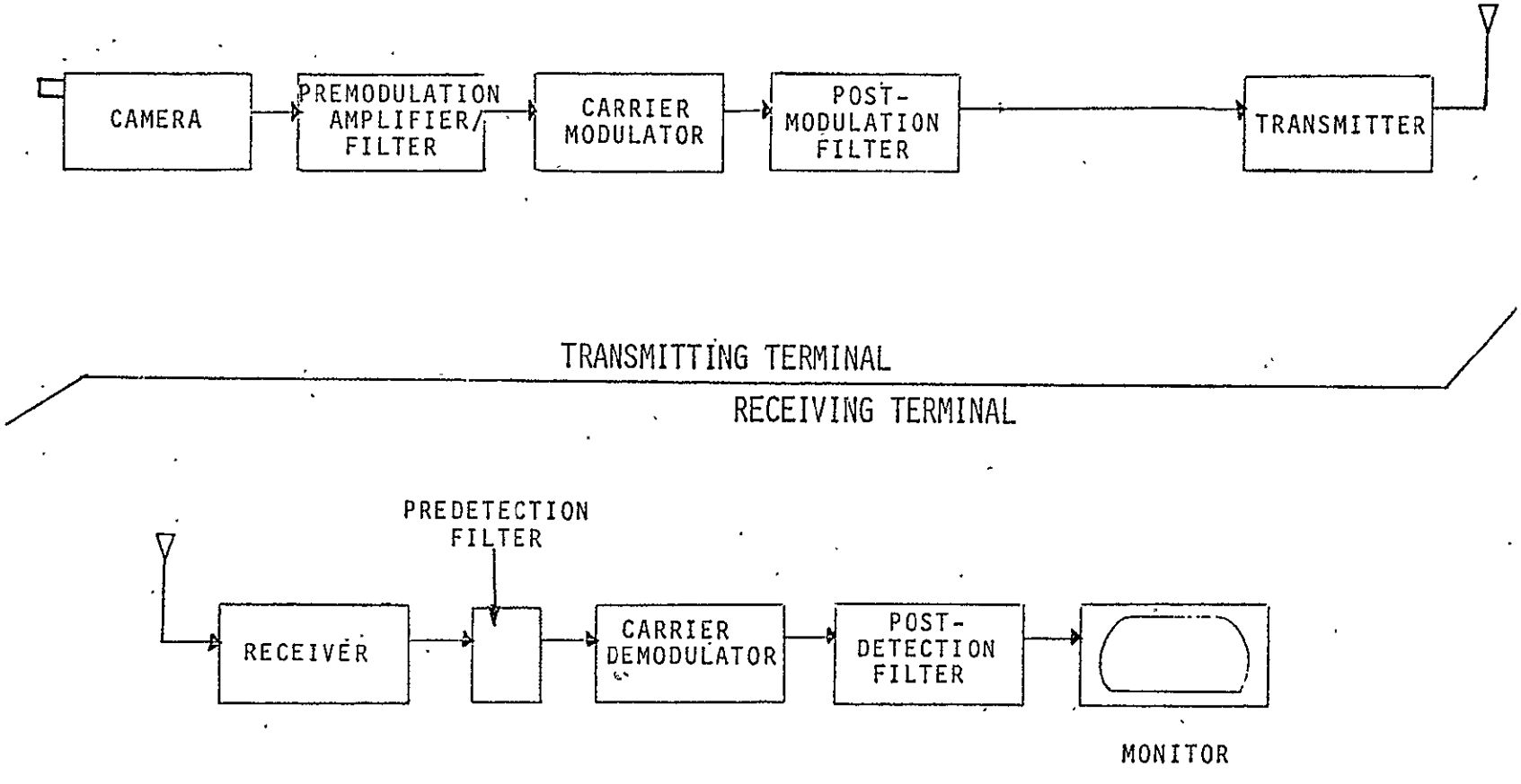


FIGURE 1. ANALOG PROCESSING/TRANSMISSION OF TV

a. Digital transmission of television is potentially much more efficient than analog transmission, in terms of a power-bandwidth tradeoff.

b. More efficient time-division multiplexing techniques can be used if multiple channels of information must be accommodated.

c. Digital techniques are easily adaptable for privacy transmissions.

Figure 2 shows a generalized block diagram of a digital television transmission system. The source encoder is assumed to perform all the functions unique to a particular processing technique (signal transformations, sampling, quantizing, and coding). The basic goal of the source encoder is to reduce the number of digital symbols per second required to represent the television pictures to within some set of acceptable performance criteria. The performance criteria commonly employed for television transmission systems include picture quality (as represented by the judgment(s) of one or more viewers) and picture resolution (which is easy to measure). The source coding goal of digital symbol rate reduction is invariably achieved by eliminating redundant or imperceptible parts of the signals, and care must be exercised in order to insure that picture quality and/or resolution will not be appreciably degraded by the resultant introduction of flicker, spurious contours or patterns, etc. It should be noted that, although digital transmission techniques are generally thought of as requiring an increased RF bandwidth, it is, at least in principle, conceivable that the digital data rate can be reduced such that the required bandwidth is actually less than for analog transmission systems.

The purpose of this survey of television digitization/compression techniques is to summarize the key features, advantages, and disadvantages of several of the various source coding techniques which are in use today or which have been proposed. Since the very nature of any source coding scheme for television involves the process of quantization, then no such schemes are entropy-preserving in the strict sense (i.e., some information is always lost and this loss in picture information is manifested as quantization noise). However, given that the initial quantization process is such that the loss in picture information is acceptable, then it is possible

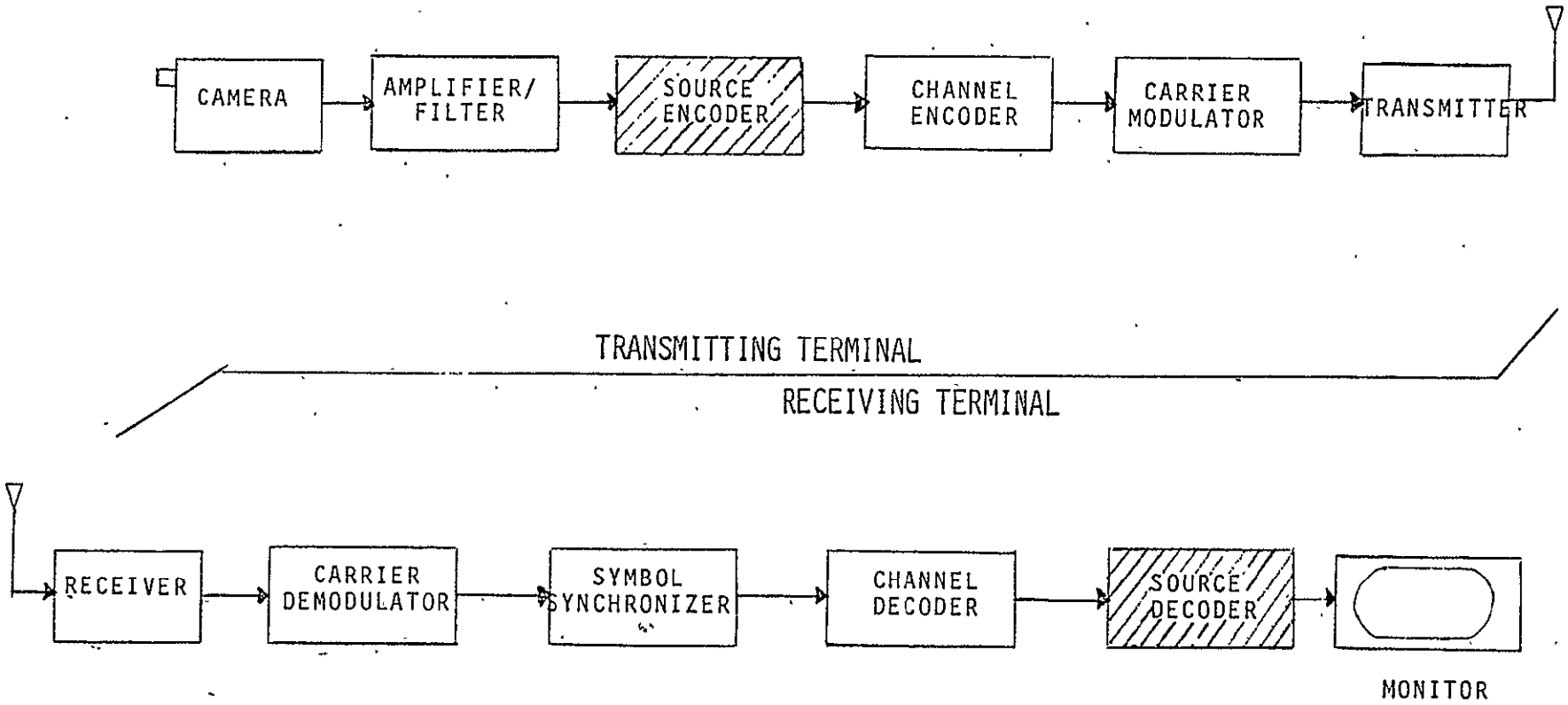


FIGURE 2. DIGITAL TRANSMISSION SYSTEM FOR TELEVISION

to do further source encoding in an entropy-preserving manner. It is also, of course, possible to lose additional information by further source encoding.

Figure 3 illustrates one categorization which can be made of the various source encoding approaches for television, in which a large class of techniques attempts to "track" or "follow" the video waveform exactly, and in which the other class attempts to merely reproduce something which looks nearly like the original image sequence. Many other source encoding categorizations can and have been made.

As will be pointed out in the subsequent discussion, and as indicated in Figure 4, the TV source coding problem can be visualized as a one-, two-, or three-dimensional problem. Various combinations of the source coding techniques shown in Figure 4 can be used in exploiting the spatial and temporal redundancies present in a television signal. The discussion to follow will first (in the classic manner) consider source coding techniques which are applicable to black-and-white images. What can be done to code color images will be considered in a later section of this report.

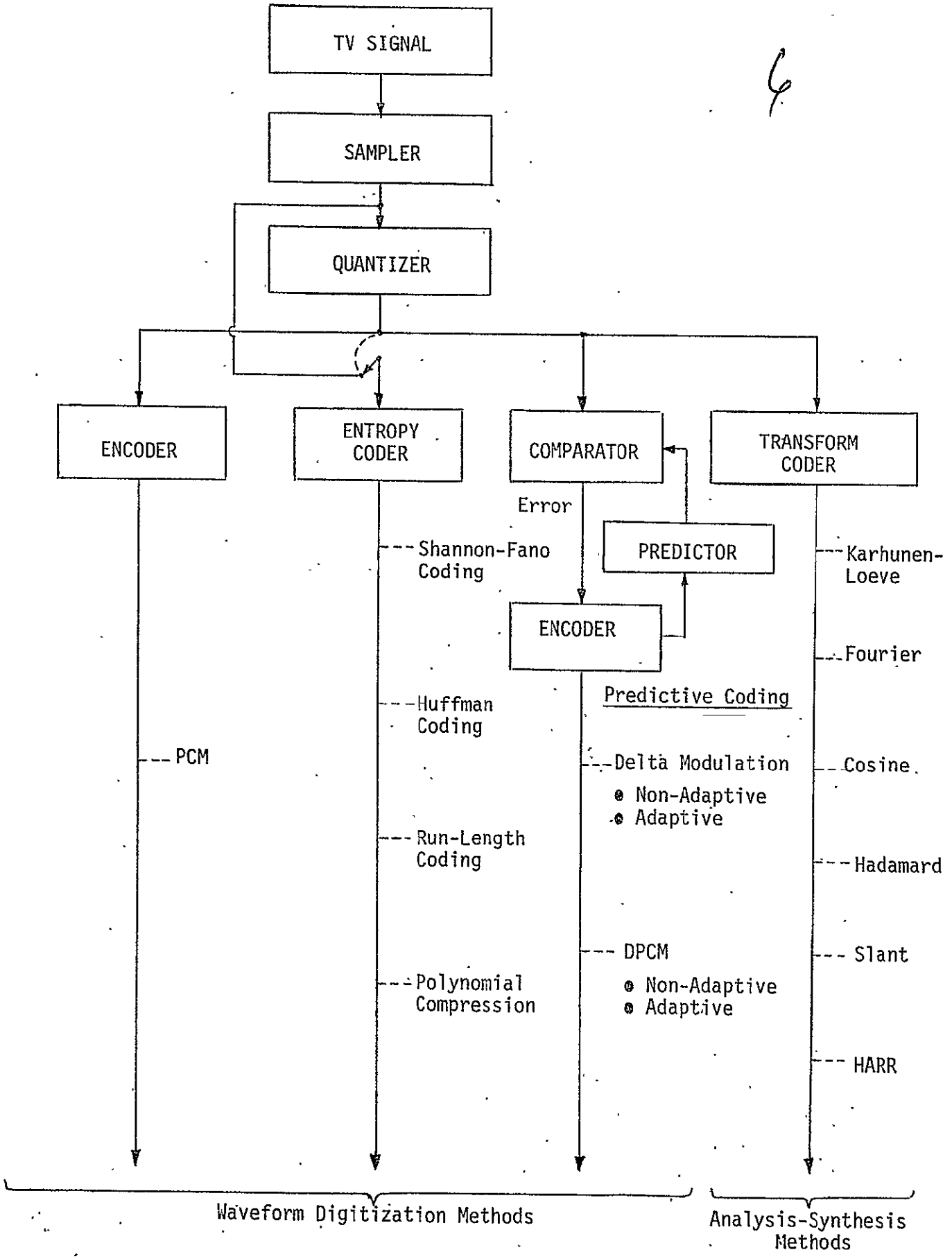
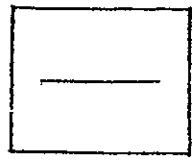


Figure 3. Television Source Encoding Categorization

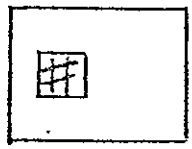
INTRAFRAME

(SPATIAL)

- ONE DIMENSIONAL



- TWO DIMENSIONAL



INTERFRAME

(TEMPORAL)

- THREE DIMENSIONAL

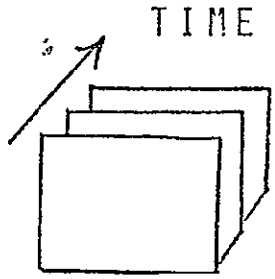


FIGURE 4. ALTERNATIVE METHODS OF TV CODING

2.0 WAVEFORM-DIGITIZATION METHODS

2.1 Pulse Code Modulation (PCM)

As illustrated in Figure 5, when using PCM transmissions for television, the continuous image is first sampled in the spatial domain (within a frame) to produce an M x N array of discrete samples (called picture elements, or pixels) which are then quantized in brightness by using one of 2^K levels to represent the value of each sample. The total number of bits per frame to be transmitted is then given by

$$B = MNK \text{ bits/frame}$$

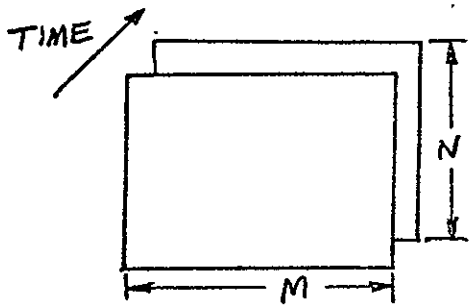
and the required rate to accommodate F frames per second is

$$R_{PCM} = BF = MNKF \text{ bits/second} .$$

As shown in Figure 6, for U.S. commercial television systems, N = 525, M = 512, F = 30, and (typically) K = 6 to 8. Thus, PCM transmission of U.S. commercial television would require a transmission rate of approximately 48 Mbps to 64 Mbps.

A PCM system is capable of transmitting any picture, including those which contain only uncorrelated samples (no redundancy). Pictures containing only uncorrelated samples are, however, not of any particular interest, since considerable correlation must be present between samples in order for an image to be present. In fact, statistical analyses of a wide range of television scenes [1] has indicated that the information content (entropy) of a typical frame is about one bit per pixel. It should therefore be possible, by proper source encoding, to use a transmission rate of only one bit per pixel, with no loss of information. An even further reduction in bit rate (by a factor of perhaps two to five) should be achievable if temporal redundancy is exploited.

Although only a single bit per pixel should theoretically be required to represent an average television frame, it has been found for PCM systems [2,3] that, when K is reduced to less than about six bits per pixel, an image degradation known as the contouring effect results. This effect is due to the formation of discrete rather than gradual changes in brightness. Several techniques have been developed [4] for reducing the contouring effect, including the addition of pre-emphasis and de-emphasis



F = FRAME RATE (FRAMES/SECOND)

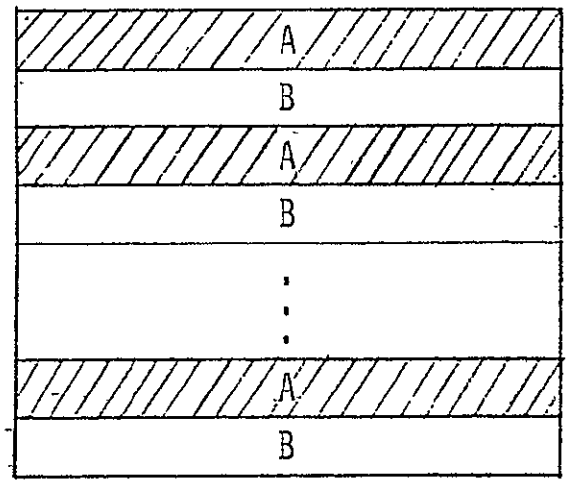
N = NO. OF LINES/FRAME = VERTICAL RESOLUTION

M = NO. OF ELEMENTS/LINE
= HORIZONTAL RESOLUTION

- NO. OF PICTURE ELEMENTS (PIXELS) PER FRAME = MN
- NO. OF PIXELS PER SECOND = MNF
- ALLOWING ONE SAMPLE PER PIXEL AND CODING EACH SAMPLE AS ONE OF 2^K LEVELS GIVES THE REQUIRED PCM TRANSMISSION RATE:

$$R_{PCM} = MNKF \text{ BITS/SECOND}$$

FIGURE 5. PCM PROCESSING OF TV



525 HORIZONTAL LINES (2 INTERLACED FIELDS)
- 45 BLANKED LINES (USED FOR VERT. SYNC PULSES)
480 LINES OF VISIBLE INFORMATION

- HORIZONTAL RESOLUTION GENERALLY ASSUMED TO BE 512 SAMPLES/LINE.
- FRAME RATE = 30 FRAMES/SECOND (60 FIELDS/SECOND)
- INFORMATION CONTENT = 480 X 512 X 30 = 7.3728 X 10^6 PIXELS/SECOND
- $\frac{45}{525}$ X 100 = 8.6% SYNC OVERHEAD

FIGURE 6. BASIC NTSC (COMMERCIAL) TV SIGNAL STRUCTURE

networks, use of pseudorandom dithering, and nonuniform quantizing, but generally it is felt that PCM is not suitable if digital rates below about 40 Mbps are required and high quality/resolution are required.

In summary, the primary disadvantages of PCM television are:

a. The bit rate required for acceptable picture quality is too high (6 bits/sample versus an entropy of ~1 bit/pixel for typical scenes).

b. Picture quality degrades rapidly for lower sampling rates or for a reduced number of bits per sample.

c. No attempt is made to exploit element-to-element, line-to-line, or frame-to-frame redundancy.

d. No consideration is given to the psychovisual properties of the eye.

2.2 Entropy Coding Techniques

2.2.1 General

Entropy coding techniques, as defined in this report, are those techniques which take advantage of the correlation present between adjacent picture samples (quantized or unquantized) along a scan line (i.e., element-to-element redundancy). Such techniques, then, rely on the more likely occurrence of some sample values than others. As a very simple and introductory example of entropy coding, consider the following case in which one of two binary source symbols A (possibly corresponding to no change from the previous sample values) and B (corresponding to a change) having probabilities

$$\begin{aligned} P(A) &= 0.9 \\ P(B) &= 0.1 \ . \end{aligned} \tag{2.1}$$

Using a codeword assignment of

$$\begin{aligned} 0 &\Longleftrightarrow A \\ 1 &\Longleftrightarrow B \ , \end{aligned} \tag{2.2}$$

the average output word length would be

$$\bar{L}_1 = 1.0 \text{ bit/symbol}, \tag{2.3}$$

corresponding to a transmission rate

$$R_1 = S\bar{L}_1 = S \text{ bits/second} \tag{2.4}$$

over the channel, where S is the sampling or source rate (in symbols per second).

By encoding pairs of source symbols (the second order extension of the source) and assigning the shortest codeword to the most likely pair, some reduction in transmission rate should be achievable. Since

$$\begin{aligned} P(AA) &= (0.9)(0.9) = 0.81 \\ P(AB) &= P(BA) = 0.09 \\ P(BB) &= 0.01 , \end{aligned} \tag{2.5}$$

then the codeword assignment

$$\begin{aligned} 0 &\Longleftrightarrow AA \\ 10 &\Longleftrightarrow AB \\ 110 &\Longleftrightarrow BA \\ 111 &\Longleftrightarrow BB \end{aligned} \tag{2.6}$$

results in an average output word length of

$$L_2 = (1)(0.81) + (2)(0.09) + (3)(0.10) = 1.29 \text{ bits/symbol pair}$$

which corresponds (since the rate of occurrence of each symbol pair is S/2) to a transmission rate of

$$R_2 = (S/2)\overline{L}_2 = 0.65\,S \text{ bits/second} . \tag{2.7}$$

Continuing with the third-order extension of the source, and using the codeword assignment below,

	P( )	Codeword
AAA	0.729	0
AAB	0.081	100
ABA	0.081	101
BAA	0.081	110
ABB	0.009	11100
BAB	0.009	11101
BBA	0.009	11110
BBB	0.001	11111

(2.8)

we obtain an average output word length of

$$\bar{L}_3 = (1)(0.729) + (3)(0.243) + (5)(0.028) = 1.598 \text{ bits/symbol trio}$$

which corresponds to a transmission rate of

$$R_3 = (S/3)\bar{L}_3 = 0.53S \text{ bits/second} . \quad (2.9)$$

The above procedure could obviously be continued for higher-ordered source extensions, although it would appear that the improvement must eventually tend toward zero, since the transmission rate must always be non-zero. In fact, one version of the fundamental source coding theorem states that the minimum digital rate R_{min} required to represent a source having entropy H [bits/symbol] and rate R [symbols/sec] can be made as close to HR [bits/second] as desired, where

$$H \equiv \sum_{k=1}^{M} P(x_k) \log_2 \left[\frac{1}{P(x_k)}\right] = -\sum_{k=1}^{M} P(x_k) \log_2 \left[P(x_k)\right] \quad (2.10)$$

$$= \text{average information per source symbol } x_k .$$

For the original source under consideration, with

$$x_1 = A, P(x_1) = P(A) = 0.9$$
$$x_2 = B, P(x_2) = P(B) = 0.1 \quad (2.11)$$
$$R = S ,$$

then

$$H = -0.9 \log_2 0.9 - 0.1 \log_2 0.1 = 0.47 \text{ bits/symbol} . \quad (2.12)$$

Thus, the minimum possible digital rate to represent this source is

$$R_{min} = 0.47 S \text{ bits/second} \quad (2.13)$$

which is nearly achieved by the simple third-order source extension.

As illustrated by the preceding example, entropy coding basically assigns shorter word lengths to those source symbols (or groups of symbols) which occur more frequently and longer word lengths to those which occur less frequently. Subsequent sections of this report will more formally present basic entropy coding concepts and describe several methods for systematically making efficient code word assignments.

2.2.2 Basic Entropy Coding Concepts

Consider a source which provides symbols from a source alphabet

$$\{x_1, x_2, \ldots, x_n\}$$

with a corresponding set of symbol probabilities

$$\{P(x_1),\ P(x_2), \ldots, P(x_n)\}\ .$$

As illustrated by the example of section 2.2.1, an entropy code essentially takes sequences of the source symbols and forms multi-symbol messages

$$\{m_1, m_2, \ldots, m_N\}$$

with a corresponding set of message probabilities

$$\{P(m_1),\ P(m_2),\ \ldots,\ P(m_n)\}\ .$$

If

n_1 = Number of symbols in message m_1

n_2 = Number of symbols in message m_2

n_N = Number of symbols in message m_N ,

then we can define an average message length:

$$\bar{L} = \sum_{i=1}^{M} P(m_i)\, n_i\ . \tag{2.14}$$

The goal of entropy coding is to find a set of messages m_i such that $\bar{L}$ is minimized. To obtain a lower bound on $\bar{L}$, we first define

$$H(X) = -\sum_{i=1}^{n} P(x_i)\, \log_2 P(x_i) \tag{2.15}$$

as the source entropy (in bits/source symbol) and

$$H(M) = -\sum_{j=1}^{N} P(m_j)\, \log_2 P(m_j) \tag{2.16}$$

as the message entropy (in bits/message). Noting that H(X) is maximized when

$$P(x_i) = \frac{1}{n} \quad \text{for all } i \tag{2.17}$$

and that

$$H(x)\big|_{max} = \log_2 n\ , \tag{2.18}$$

then it can be seen that

$$H(X) = \frac{H(M) \text{ [bits/message]}}{\bar{L} \text{ [symbols/message]}} \leq \log n . \tag{2.19}$$

We see that the minimum possible value for $\bar{L}$ is given by

$$\bar{L}\big|_{min} = \frac{H(M)}{\log_2 n} . \tag{2.20}$$

If $\bar{L} = \bar{L}\big|_{min}$, then we say that the redundancy is zero and that the efficiency is 1 (100%), where

$$\eta = \text{Encoding Efficiency} \equiv \frac{H(M)}{\bar{L} \log_2 n} = \frac{\bar{L}\big|_{min}}{\bar{L}} \tag{2.21}$$

and

$$\begin{aligned} \text{Redundancy} &= 1 - \text{efficiency} \\ &= 1 - \frac{H(M)}{\bar{L} \log_2 n} \\ &= \frac{\bar{L} \log_2 n - H(M)}{\bar{L} \log_2 n} . \end{aligned} \tag{2.22}$$

For the encoded messages, the average probability of occurrence of the ith source symbol is given by

$$P(x_i) = \frac{\sum_{k=1}^{N} P(m_k)\, C_{k_i}}{\sum_{k=1}^{N} P(m_k)\, n_k} = \frac{\sum_{k=1}^{N} P(m_k)\, C_{k_i}}{\bar{L}} \tag{2.23}$$

where C_{k_i} = number of x_i's in the kth message,

$$\sum_{k=1}^{N} P(m_k)\, C_{k_i} = \text{average number of } x_i\text{'s per message} . \tag{2.24}$$

To illustrate the utility of the above source coding concepts, consider the first-, second-, and third-order extensions of the binary source used in the example of section 2.2.1:

- First-order extension

$x_1 = A \qquad P(x_1) = 0.9$

$x_2 = B \qquad P(x_2) = 0.1$

$m_1 \Longleftrightarrow 0 \qquad P(m_1) = 0.9$

$m_2 \Longleftrightarrow 1 \qquad P(m_2) = 0.1$

$H(X) = -0.9 \log_2 0.9 - 0.1 \log_2 0.1 = 0.47$ bits/symbol

$H(M) = -0.9 \log_2 0.9 - 0.1 \log_2 0.1 = 0.47$ bits/message

$\bar{L} = (0.9)(1) + (0.1)(1) = 1.0$ symbols/message

$H(X)\big|_{max} = \log_2 n = 1.0$ bit/symbol

$\bar{L}_{min} = \frac{H(M)}{\log_2 n} = \frac{0.47}{1.0} = 0.47$ symbols/message

$\eta = \frac{\bar{L}\big|_{min}}{\bar{L}} = \frac{0.47}{1.0} = 0.47 = 47\%$

Redundancy $= 1 - \eta = 0.53 = 53\%$

- Second-order extension

$m_1 \quad 00 \Longleftrightarrow AA \qquad P(m_1) = P(A)\,P(A) = 0.81$

$m_2 \quad 01 \Longleftrightarrow AB \qquad P(m_2) = P(A)\,P(B) = 0.09$

$m_3 \quad 10 \Longleftrightarrow BA \qquad P(m_3) = P(B)\,P(A) = 0.09$

$m_4 \quad 11 \Longleftrightarrow BB \qquad P(m_4) = P(B)\,P(B) = 0.01$

$H(X) = 0.47$ bits/symbol

$H(M) = -0.81 \log_2 0.81 - 0.09 \log_2 0.09$

$\qquad - 0.09 \log_2 0.09 - 0.01 \log_2 0.01$

$= 0.94$ bits/message

$\bar{L} = (0.81)(2) + (0.09)(2) + (0.09)(2) + (0.01)(2)$

$= 2$ symbols/message

$H(X)\big|_{max} = 1$ bit/symbol

$$\bar{L}_{min} = \frac{H(M)}{\log_2 n} = 0.94 \text{ symbols/message}$$

$$\eta = \frac{\bar{L}|_{min}}{\bar{L}} = \frac{0.94}{2} = 0.47 = 47\%$$

$$\text{Redundancy} = 1 - \eta = 53\%$$

Note that this uncoded second-order source extension provides no improvement in efficiency, or equivalently, no reduction in redundancy. Also note that we still have $P(0) = 0.9$ and $P(1) = 0.1$.

- Second-order extension (Source coding assignment using fewer symbols to represent more likely messages)

m_1 0 $\Longleftrightarrow$ AA $P(m_1) = 0.81$

m_2 10 $\Longleftrightarrow$ AB $P(m_2) = 0.09$

m_3 110 $\Longleftrightarrow$ BA $P(m_3) = 0.09$

m_4 111 $\Longleftrightarrow$ BB $P(m_4) = 0.01$

$H(X) = 0.47$ bits/symbol

$H(M) = 0.94$ bits/message

$$\bar{L} = (0.81)(1) + (0.09)(2) + (0.09)(3) + (0.01)(3)$$

$$= 1.29 \text{ symbols/message}$$

$$H(X)|_{max} = 1 \text{ bit/symbol}$$

$$\bar{L}|_{min} = 0.94 \text{ symbols/message}$$

$$\eta = \frac{\bar{L}|_{min}}{\bar{L}} = \frac{0.94}{1.29} = 0.729 = 72.9\%$$

$$\text{Redundancy} = 1 - \eta = 0.271 = 27.1\%$$

$$P(0) = \frac{(0.81)(1) + (0.09)(1) + (0.09)(1)}{1.29} = 0.77$$

$$P(1) = 1 - P(0) = 0.23$$

- Third-order extension (no source coding)

m_1	000 $\Leftrightarrow$ AAA	$P(m_1)$	= 0.729
m_2	001 $\Leftrightarrow$ AAB	$P(m_2)$	= 0.081
m_3	010 $\Leftrightarrow$ ABA	$P(m_3)$	= 0.081
m_4	100 $\Leftrightarrow$ BAA	$P(m_4)$	= 0.081
m_5	011 $\Leftrightarrow$ ABB	$P(m_5)$	= 0.009
m_6	101 $\Leftrightarrow$ BAB	$P(m_6)$	= 0.009
m_7	110 $\Leftrightarrow$ BBA	$P(m_7)$	= 0.009
m_8	111 $\Leftrightarrow$ BBB	$P(m_8)$	= 0.001

It can be readily shown for this case that

$H(M) = (0.47)(3) = 1.41$ bits/message

$\overline{L}_{min} = 3$ symbols/message

$$\eta = \frac{\overline{L}|_{min}}{\overline{L}} = \frac{1.41}{3} = 0.47 = 47\%$$

Redundancy $= 1 - \eta = 53\%$

$P(0) = 0.9$

$P(1) = 0.1$

- Third-order extension (Source coding assignment using fewer symbols to represent more likely messages)

m_1	0 $\Leftrightarrow$ AAA	$P(m_1)$	= 0.729
m_2	100 $\Leftrightarrow$ AAB	$P(m_2)$	= 0.081
m_3	101 $\Leftrightarrow$ ABA	$P(m_3)$	= 0.081
m_4	110 $\Leftrightarrow$ BAA	$P(m_4)$	= 0.081
m_5	11100 $\Leftrightarrow$ ABB	$P(m_5)$	= 0.009
m_6	11101 $\Leftrightarrow$ BAB	$P(m_6)$	= 0.009
m_7	11110 $\Leftrightarrow$ BBA	$P(m_7)$	= 0.009
m_8	11111 $\Leftrightarrow$ BBB	$P(m_8)$	= 0.001

$$H(X) = 0.47 \text{ bits/symbol}$$

$$H(M) = 1.41 \text{ bits/message}$$

$$\bar{L} = (0.729)(1) + (0.081)(9) + (0.009)(15) + (0.001)(5)$$

$$= 1.598 \text{ symbols/message}$$

$$H(X)\big|_{max} = 1 \text{ bit/symbol}$$

$$\bar{L}\big|_{min} = \frac{1.41}{1} = 1.41 \text{ symbols/message}$$

$$\eta = \frac{\bar{L}\big|_{min}}{\bar{L}} = \frac{1.41}{1.598} = 0.882 = 88.2\%$$

$$\text{Redundancy} = 1 - \eta = 0.118 = 11.8\%$$

$$P(0) = \frac{(0.729)(1) + (0.081)(4) + (0.009)(4)}{1.598} = 0.68$$

$$P(1) = 1 - P(0) = 0.32$$

It is evident from the foregoing discussion that, by forming multi-symbol messages and assigning fewer symbols to more likely messages than to less likely messages, significant improvements in efficiency (corresponding to reductions in average word length $\bar{L}$) can be obtained. By proper code assignment, it is ultimately possible to make

$$\bar{L} = \bar{L}_{min} = \frac{H(M)}{\log_2 n}$$

$$\eta = 100\%$$

$$\text{Redundancy} = 0\% \qquad (2.25)$$

$$P(x_i) = \frac{1}{n} \text{ for all } i \, .$$

As yet, however, we have no information which tells us how to make the best code word assignments to the various messages. Subsequent sections of this report consider various structured algorithms for such code word assignments.

2.2.3 Shannon-Fano Coding

The Shannon-Fano source coding algorithm has the advantage of being probably the simplest of the various entropy coding approaches. For binary coding, given a set of messages, the algorithm may be simply stated

as follows:

a. List messages in order of decreasing probability.

b. Partition messages into the two most equiprobable subsets. Assign a 0 to each message in one set and a 1 to each message in the other.

c. Repeat (b) for each of the new subsets, and continue to repeat until each subset contains only one message.

The above procedure may best be visualized in terms of an example, such as that shown in Table 1, in which a set of five messages is encoded using the Shannon-Fano procedure.

Table 1. Applications of Shannon-Fano Coding Algorithm

Message	Probability	First Partitioning	Second Partitioning	Third Partitioning	Code Word Assignment
m_1	$\frac{1}{2}$	$\frac{1}{2}$ } 0	$\frac{1}{2}$ } 0	$\frac{1}{2}$ } 0	0
m_2	$\frac{1}{6}$	$\frac{1}{2}$ } 1	$\frac{1}{3}$ } 10	$\frac{1}{6}$ } 100	100
m_3	$\frac{1}{6}$			$\frac{1}{6}$ } 101	101
m_4	$\frac{1}{12}$		$\frac{1}{6}$ } 11	$\frac{1}{12}$ } 110	110
m_5	$\frac{1}{12}$			$\frac{1}{12}$ } 111	111

Note that, for the Shannon-Fano coding example shown above, we have

$$H(M) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{3}\log_2\left(\frac{1}{6}\right) - \frac{1}{6}\log_2\left(\frac{1}{12}\right) = 1.96 \text{ bits/message}$$

$$\bar{L} = \left(\frac{1}{2}\right)(1) + \left(\frac{1}{2}\right)(3) = 2 \text{ bits/symbol}$$

$$\bar{L}\Big|_{min} = \frac{H(M)}{\log_2 n} = \frac{1.96}{1} = 1.96$$

$$\eta = \frac{\bar{L}\Big|_{min}}{\bar{L}} = \frac{1.96}{2} = 0.98 = 98\%$$

$$\text{Redundancy} = 1 - \eta = 0.02 = 2\%$$

Thus, the simple Shannon-Fano encoding procedure yields a code which is very nearly optimum (100% efficient). Should it be desired to further improve efficiency, however, this can be accomplished by repeating the above procedure for longer messages and message sets.

The extension of the Shannon-Fano encoding procedure for nonbinary ($n \neq 2$) sources is straightforward. For general n-ary sources, the messages are merely partitioned into the n most equiprobable subsets, with the first source symbol assigned to the first subset, the second symbol assigned to the second subset, etc. This procedure is then repeated as before.

For purposes of encoding television signals, the Shannon-Fano coding procedure is applied to best advantage when the messages represent differences between adjacent sample values, rather than the absolute sample values themselves. This is, of course, because of the high degree of correlation between adjacent samples, resulting in many difference values at or near zero.

As an example of the possible utility of a coding technique such as the Shannon-Fano approach, consider the results of a tabulation of amplitude-difference statistics for a large number of conditons and for several different pictures. Using 4-bit quantization of picture samples, the difference statistics shown in Table 2 are observed. A practical scheme for applying the Shannon-Fano coding algorithm is shown in Table 3, where delta values greater than 3 are represented by the Shannon-Fano 4-bit code word assignment, augmented by the 4 bits representing the absolute difference value.

2.2.4 Huffman Coding

Perhaps the most conceptually important approach which falls in the class of entropy coding is the Huffman coding procedure, which is similar to, although somewhat more complicated than, the Shannon-Fano

Δ	P(Δ)	
0	0.78	
+1	0.08	
-1	0.08	
D	0.02	+3 < D < -3
+2	0.01	
-2	0.01	
+3	0.007	
-3	0.007	
HS	0.004	HS = Horizontal Sync
VS	0.002	VS = Vertical Sync
	1.000	

Table 2. Television Difference Statistics for 4-Bit PCM

Δ	P(Δ)	Code Word Assignment
0	0.78	1
+1	0.08	0 1
-1	0.08	0 0 1
D	0.02	0 0 0 1 X X X X
+2	0.01	0 0 0 0 1 1
-2	0.01	0 0 0 0 1 0
+3	0.007	0 0 0 0 0 1
-3	0.007	0 0 0 0 0 0 1
HS	0.004	0 0 0 0 0 0 0 1
VS	0.002	0 0 0 0 0 0 0 0

Table 3. Application of (Modified) Shannon-Fano Coding Procedure to Television Difference Signal

procedure. The importance of Huffman coding is that it is an optimum approach, in the sense that, for a given message set, the average word length is guaranteed to be the minimum achievable value. The relative complexity of Huffman coding as opposed to, say, Shannon-Fano coding makes it probably not extremely attractive for some applications, but it is felt to be potentially applicable to television transmission and therefore will be described here.

a. For an optimum code, the longer code word should correspond to a message with lower probability; thus, if for convenience the messages are numbered in order of nonincreasing probability,

$$P(m_1) \geq P(m_2) \geq P(m_3) \geq \cdots \geq P(m_N) , \tag{2.26}$$

then the corresponding code word lengths are such that

$$L(m_1) \leq L(m_2) \leq L(m_3) \leq \cdots \leq L(m_N) . \tag{2.27}$$

b. For an optimum code, it is necessary that

$$L(m_{N-1}) = L(m_N) . \tag{2.28}$$

If we assign similar code words to m_N and m_{N-1} except for the final symbol, our purpose is served. Any additional symbol for m_N and m_{N-1} unnecessarily increases $\bar{L}$. Therefore, at least two messages, m_{N-1} and m_N, should be encoded in words of identical length. However, not more than n such messages could have equal length. It can be shown that, for an optimum encoding, n_0 (the number of least probable messages which should be encoded in words of equal length) is the integer satisfying the requirements:

$$\frac{N - n_0}{n - 1} = \text{integer} ; \qquad 2 \leq n_0 \leq n . \tag{2.29}$$

c. Each sequence of length $L(m_N) - 1$ symbols either must be used as an encoded word or must have one of its prefixes used as an encoded word.

In the following, we shall initially restrict ourselves to the binary case (n = 2). Condition (b) requires that the two least probable messages have the same length, and that the two encoded messages be identical except for their last symbols. We shall select these two messages to be the Nth and (N-1)th original messages. After such a

selection, we form a composite message out of these two messages with a probability equal to the sum of their probabilities. The set of messages in which the composite message is replacing the aforementioned two messages will be referred to as an auxiliary ensemble of order 1, or simply AE1. Now, we shall apply the rules for finding optimum codes to AE1; this will lead to AE2, AE3, and so on. The code words for each two least probable members of any ensemble AEk are identical except for their last symbols, which are 0 for one and 1 for the other. This iteration cycle is continued until AEp has only two messages. A final symbol 0 is assigned to one of the messages and 1 to the other. Now we simply trace back our path and remember each two messages which have to differ only in their last digits. The optimality of this procedure is a direct consequence of the previously described optimal steps.

The example shown in Table 4 illustrates clearly the application of the Huffman coding procedure.

The Huffman encoding procedure described above is readily adaptable for nonbinary code word assignments. The key to nonbinary Huffman coding is the requirement of condition (b), as previously described. Thus, for the ternary ($n = 3$) example shown in Table 5, the number of least probable messages which must be encoded in words of equal length is that value of n_0 which satisfies

$$\frac{N - n_0}{n - 1} = \frac{6 - n_0}{3 - 1} = \text{integer} ; \quad 2 \leq n_0 \leq 3 \tag{2.30}$$

or $$n_0 = 2 .$$

After the initial selection of $n_0 = 2$ least probable messages, and formulation of AE1, subsequent subsets of three least probable messages are used to generate new auxiliary ensembles.

2.3 Data Compression Techniques

2.3.1 General

One approach for reducing transmission rate requirements when the data to be transmitted is repetitive in nature (i.e., contains redundancy) is to employ a transformation to remove those samples which are declared redundant. Such redundancy reduction or data compression techniques are

Table 4. Application of Binary Huffman Coding Procedure

Code Word Assignment	Message	Probability	AE1	AE2	AE3	AE4
10	m_1	0.25	0.25	0.25	0.30	0.45
01	m_2	0.25	0.25	0.25	0.25	0.30
00	m_3	0.20	0.20	0.20	0.25	
111	m_4	0.15	0.15	0.30	0.45	0.55
1101	m_5	0.10	0.15	0.25	0.30	0.45
1100	m_6	0.05				

Table 5. Application of Ternary Huffman Coding Procedure

Code Word Assignment	Message	Probability	AE1	AE2
1	m_1	0.25	0.25	0.50
0	m_2	0.25	0.25	0.25
22	m_3	0.20	0.20	
21	m_4	0.15	0.15	
201	m_5	0.10		
200	m_6	0.05		

generally not entropy-preserving in a strictly classical sense, because some information is lost; however, since the processes are reversible to within a specified allowable tolerance or peak error, they are frequently considered to be in the class of information-preserving techniques.

Several types of redundancy reduction algorithms have been formulated and many more are conceptually possible. In general, however, they may be classes as either predictors or interpolators. Because of the correlation that exists from sample to sample, a prediction of the value of the next sample can be made. If the prediction is close enough (to within some allowable tolerance determined by the particular algorithm employed), then there is no need to transmit the sample. Since the same prediction can be made at the receiver, the missing samples can be reconstructed to within the transmitter tolerance.

Polynomial data compression techniques have been widely reported on in the literature and actual hardware has been built to implement such techniques. The polynomial compression schemes generally conceded to be most promising for a wide range of data sources are listed below, and will be described in some detail in subsequent sections of this report.

1. Zero-order Predictor (ZOP)
2. Zero-order Interpolator (ZOI)
3. First-order Predictor (FOP)
4. First-order Interpolator (FOI) .

As a general rule, the type of algorithm to use in a particular application is highly dependent on the data characteristics of the source. Higher-order polynomial compression algorithms (second-order predictors and interpolators, third-order, etc.) are, of course, in most applications because of their inherent design complexity and because of their tendencies to oscillate when operating on data which has small fluctuations such as those which might be caused by noise (the tendency to oscillate when operating on noisy data is also characteristic of the first-order predictor [5]).

2.3.2 Buffer Memory Requirements

When a data compression system is added to the system, the relationship between the data and its channel and the time information changes. The data is input to the data compression algorithm from the computer in

a synchronous manner. After the reversible transfer takes place, some of the channels of a frame will not be output. The result is an asynchronous output from the data compression algorithm. By asynchronous we mean the output appears in a seemingly random fashion and the synchronous frame is eliminated.

A major subsystem of the data compression system is the buffer. The buffer is needed to reestablish a synchronous bit rate so that standard systems can be used for transmission of compressed data. The buffer, when implemented into a hardware system, must be some sort of storage device such as a magnetic core memory. The memory will have a finite storage capability. If the average read-in rate/read-out rate goes close to 1, such as during an active data burst, then many channels will have outputs from the compression algorithm, and the probability of buffer overflow will be large. Buffer overflow is the term used to describe the condition of full buffer memory. Any additional inputs to the memory cannot be stored and are lost—hence overflow. The loss of any data, especially at a time of some important event, cannot be allowed. When it is recalled that one compressed data point represents many uncompressed data points, the solution of the problem of buffer overflow clearly becomes mandatory.

There are several ways of controlling buffer overflow. One method commonly reported in the literature is adaptive aperture control. This type of buffer control operates by sensing either buffer fullness or data activity, or both. Depending upon the fullness or amount of data activity, the tolerances within the data compression algorithm are increased in order to decrease the number of significant samples output from the data compression algorithm to the buffer. An obvious disadvantage of this method is the increase of peak error at an active period when the data may be of the most interest to the experimenter. The philosophy used to justify this method implies that, because of the higher activity, the peak errors are not as noticeable due to steep slopes and sudden changes when viewed by the human eye, assuming human data examination.

A second method of buffer control is called adaptive sampling. With this type of adaptive control, the buffer changes the sampling sequence in the commutator to reduce the sampling rate of active channels if overflow is imminent. If a randomly addressable memory was used for a commutator, this could be a completely flexible operation as any sampling sequence could be set up.

Another method of buffer control is that of adaptive filtering. Here the data is sampled near the frequency of the system noise. An average is taken of N points, thus smoothing the data.. This average is transmitted to the data compression algorithm and from there on to the buffer. If data activity increases, tending to cause the buffer fullness condition, the number of samples averaged (N) will be increased, thus decreasing the total number of points input to the data compression algorithm and buffer.

2.3.3 Sample Identification and Timing Methods

Since the nonredundant samples provided by a data compression device may occur at any time, some means of time tagging them must be provided. This can be done in many ways; however, the simplest way is to follow each nonredundant sample with a run-length code (that is, following them with a code word which tells how many samples were dropped since the last nonredundant sample). The required length of this word is a function of the data. If, for instance, a sample reduction of 8 or 10 is expected, a 4-bit code could be used. Even if the sixteenth sample is redundant, it is considered nonredundant and the process is started over, thus limiting the maximum sample reduction to 15 to 1. Care must be used in choosing the length of this code, since each added bit must be transmitted with every nonredundant sample.

Another possible timing scheme is to transmit the actual time when a nonredundant sample occurs. This requires many bits and, in general, is not used. Some modification may be used, however, such as transmitting a master time at fixed intervals and then transmitting the time since the last master time. This can be dangerous in case the master time should be lost, since all data also will be lost until another master time is received.

Another, and common, method used in multi-channel systems is to transmit a code number at the beginning of each frame, and then tag each nonredundant sample with a channel number. Since a particular channel will always appear at a known location within the frame, time can be established.

As mentioned earlier, the number of nonredundant samples does not increase as the sampling rate is increased once enough samples are available to define the reconstructed waveform. If, however, a run length

timing code is used and the sampling rate is increased, the more times the run length will be exceeded, and the nonredundant samples will increase. When a run length code is used together with large tolerances, there is a good chance the run length will be exceeded.

2.3.4 Description of Polynomial Compression Algorithms

2.3.4.1 Zero-Order Predictor (Figure 7)

This algorithm predicts that the following samples will be within a specified tolerance of the first sample. If this prediction is true, the data sample is considered redundant and is not transmitted. The first sample which falls outside the tolerance level is considered nonredundant, is transmitted, and becomes the new reference for subsequent predictions. Tolerances will normally be different for each channel, but will be variable by command.

2.3.4.2 Zero-Order Interpolator (Figure 8)

In this algorithm, the tolerance is set about the first data point as was done in the Zero-Order Predictor. When the next point is examined, the tolerance is set up about it. The windows of the tolerances of the two points are checked for an overlap. If the windows do overlap, a new window upper bound is found by taking the minimum of the upper bound of the old window versus the new point plus the half-tolerance. A new lower bound is found by similarly taking the maximum of the lower bound versus the new point minus the half-tolerance. The tolerance is set up about the third point, the window formed, checked for overlap against the last window generated, a new window formed, etc. When the windows fail to overlap, an average value is found by taking the average of the upper and lower bounds. This value is then transmitted and a new run started.

2.3.4.3 First-Order Predictor (Figure 9)

In this algorithm, the first point is transmitted. The third point is predicted to be on the line connecting the first and second points, plus or minus the given tolerance. If the point is in tolerance, the next point is predicted on the same line, etc. If the new point was outside the tolerance limits, the previous predicted sample is transmitted and a new run started using the next value.

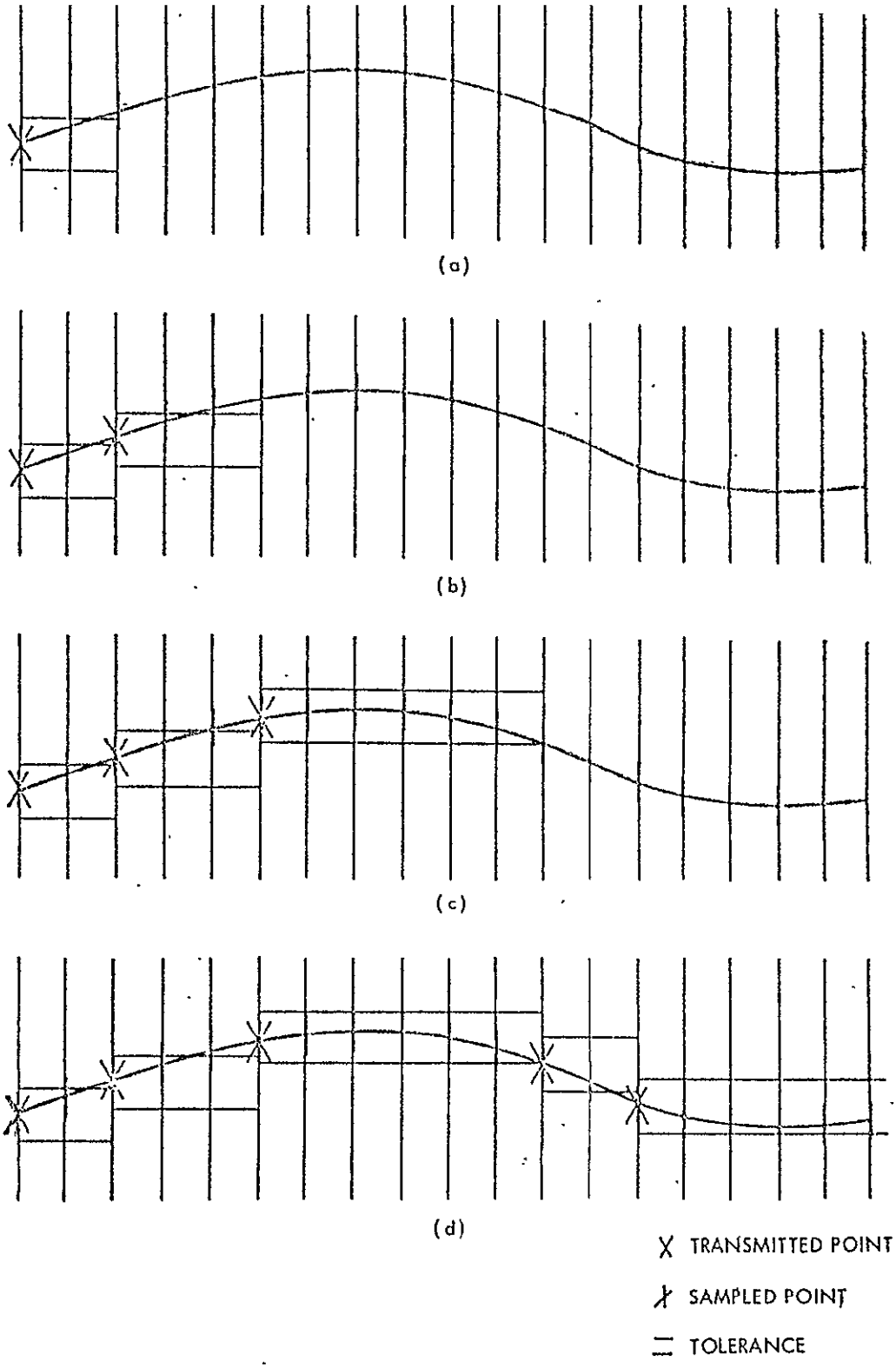


Figure 7. Zero-Order Predictor

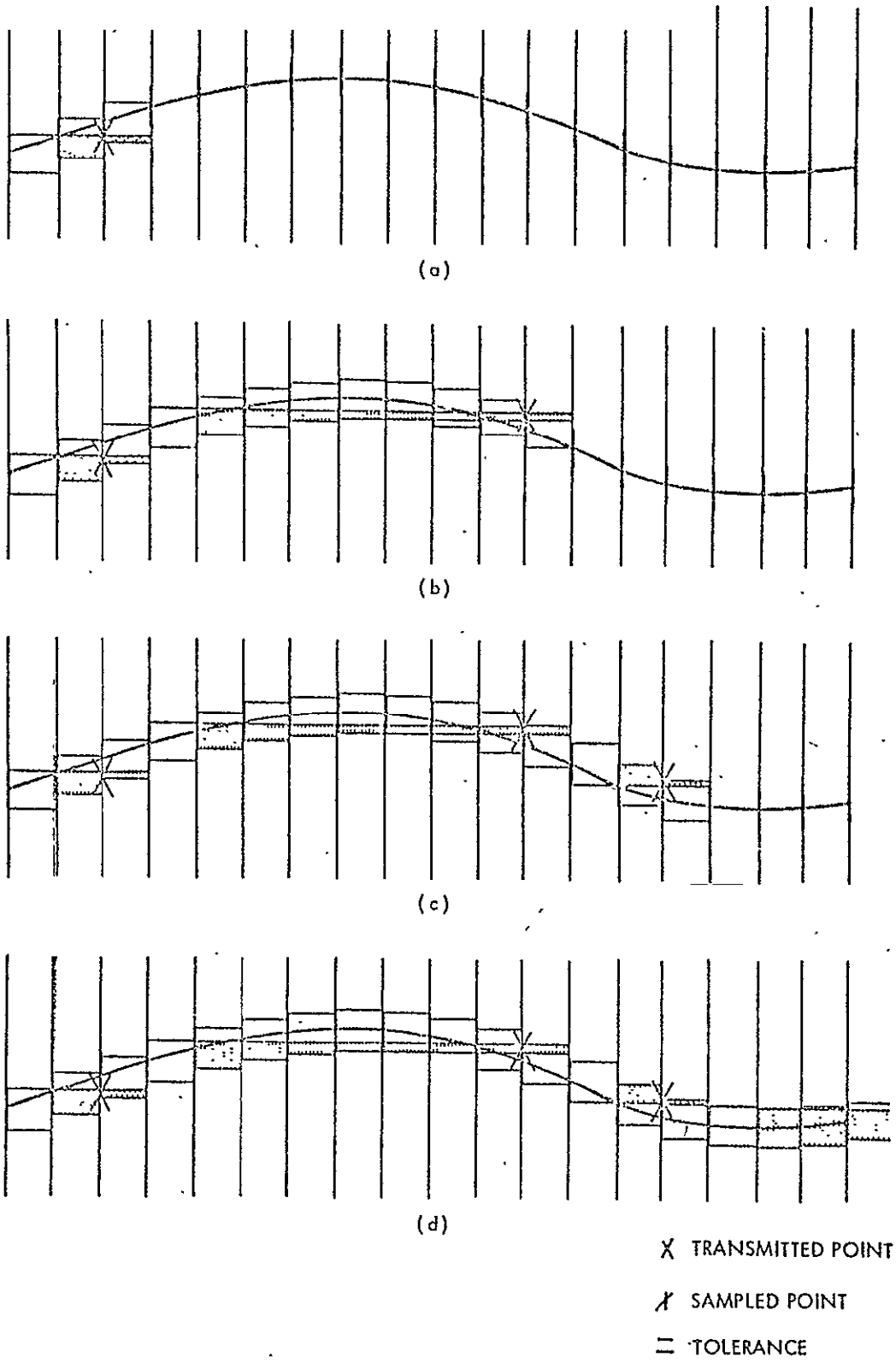


Figure 8. Zero-Order Interpolator

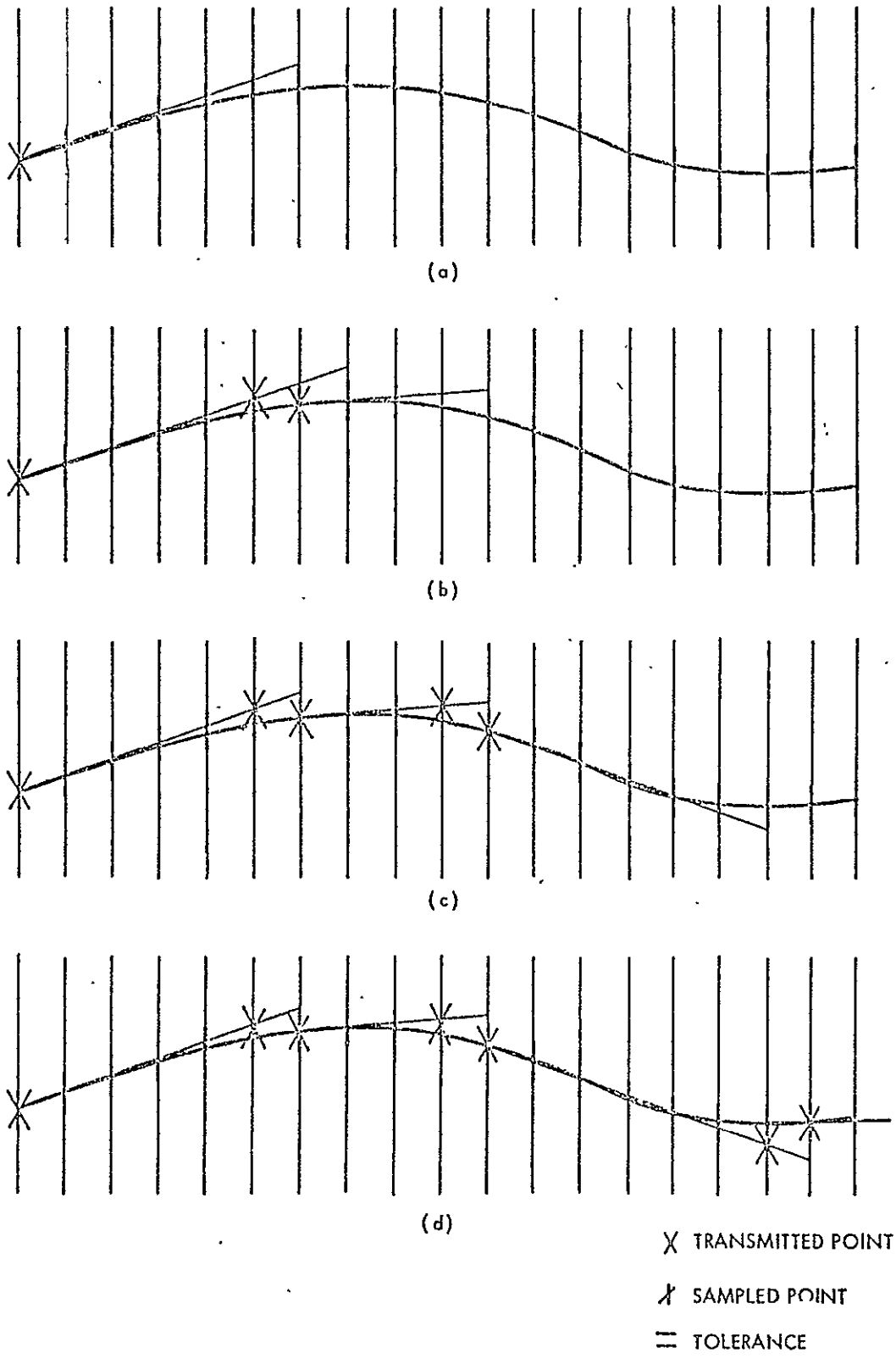


Figure 9. First-Order Predictor

2.3.4.4 First-Order Interpolator (Figure 10)

In this algorithm, the first point is transmitted. The given tolerance is placed about the second point and lines drawn from the first point through the limits of the tolearance. If the third point is within the "fan" thus formed, a new "fan" is started by drawing lines between the first point and the tolerance limits placed about the third point. The new "fan" will be the intersection* of the two "fans." Subsequent "fans" are formed until a point does not fall within the "fan." A mean value of the last tolerance spread is transmitted and a new sequence begun using the last actual point.

2.4 Predictive Coding Techniques

2.4.1 General

As pointed out earlier, PCM systems are inherently inefficient in coding correlated data such as images, since the PCM encoder always assigns the same number of binary digits to each of a number of correlated variables. As will be described later, transform coding systems involve an operation on these variables to uncorrelate them prior to their quantization. Another approach to the problem of generating a set of uncorrelated variables uses classical prediction theory. A predictive coder, as illustrated in general form in Figure 11, forms a prediction $\hat{S}_0$ of each data sample S_0 (generally by using n previous samples S_i, $i = 1,2,\ldots,n$). A differential signal

$$e_0 = S_0 - \hat{S}_0 \tag{2.31}$$

is then formed, and it is this differential signal that is quantized and coded for transmission.

2.4.2 Differential PCM (DPCM)

DPCM is a form of predictive coding in which the predictor provides an estimate $\hat{S}_0$ of each sample S_0, based on the previous n samples by the linear operation

$$\hat{S}_0 = \sum_{i=1}^{n} A_i S_i \tag{2.32}$$

*Intersection here refers to the intersection of two sets, fan 1 and fan 2.

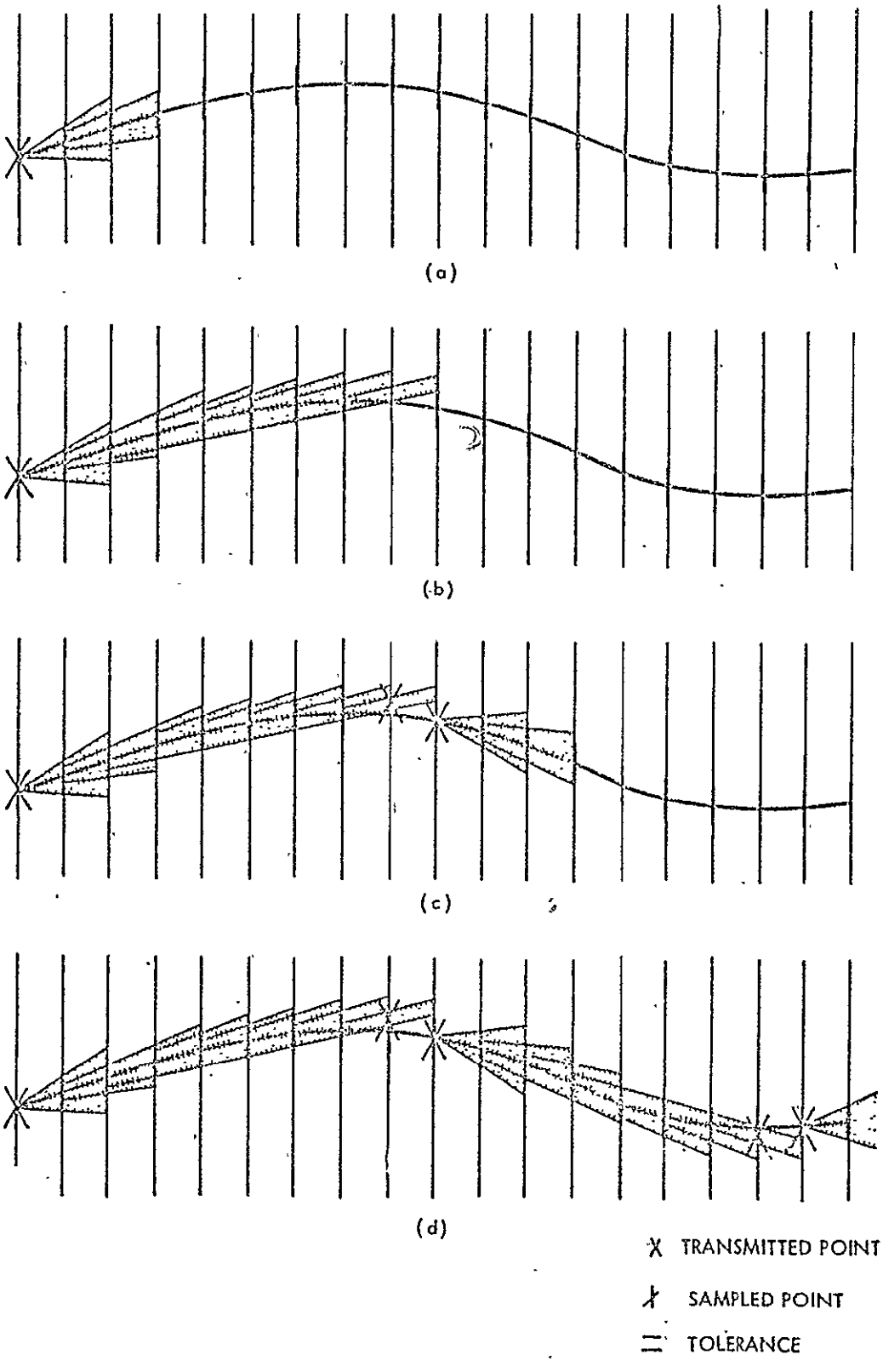
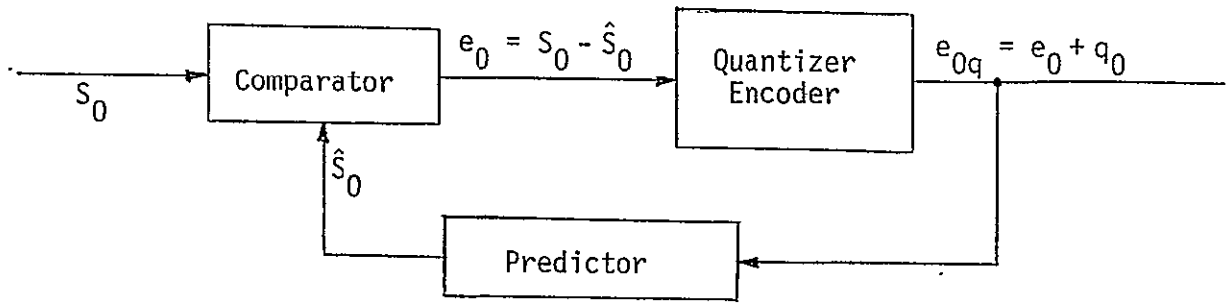
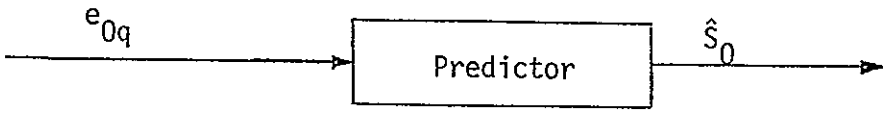


Figure 10. First-Order Interpolator



(a) Transmitter



(b) Receiver

Figure 11. Predictive Coding System (General)

where the weighting coefficients A_i are chosen to minimize the variance of the differential signal for efficient quantization and coding.

A DPCM encoder can form its prediction based on observations (samples) in only one dimension (such as along a scan line), as indicated in Figure 12. Extension of DPCM to two and three dimensions is conceptually straightforward and will be described later in this report.

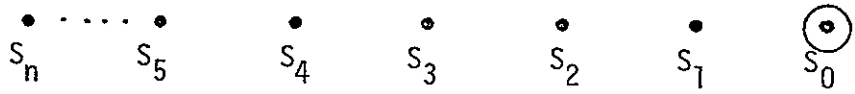
For nonadaptive DPCM, the predictor coefficients A_i are fixed constants, optimized (hopefully) for the statistics of the signal. Such a system performs well only if the original signal is stationary and has about the same statistics with which the predictor coefficients were designed. Since signal statistics for images vary greatly from scene to scene, adaptive techniques must generally be employed for best performance.

2.4.3 Adaptive DPCM (ADPCM)

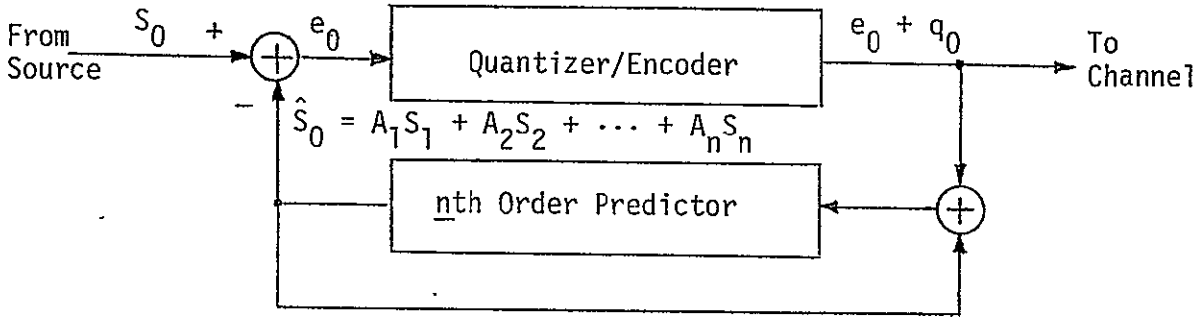
2.4.3.1 DPCM Systems with Adaptive Predictors

Optimal encoding of the nonstationary differential signal requires a variable quantizer which would change to accommodate the variations in the differential signal. In designing an adaptive DPCM system, one must either use a predictor with variable parameters such that the parameters would change with the variations in the signal (always generating a stationary differential signal) or one can use a fixed predictor with a variable quantizer to accommodate the resultant nonstationary differential signal. In addition to the above two adaptive systems, the adaptivity can be incorporated in the system by using a variable sampling rate and fixing both the predictor and the quantizer.

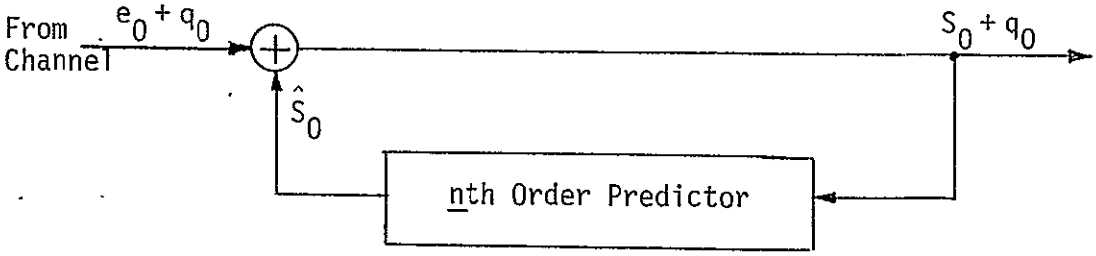
In a DPCM system with an adaptive linear predictor, the weightings on the adjacent samples used in predicting an incoming sample can change according to variations in the signal value. Atal and Schroeder [6] studied the performance of such an adaptive system for voice signals. Their proposed system included a 5 millisecond delay during which the incoming samples were stored in an input buffer and were used to obtain an estimate of signal covariance matrix. The measured covariance matrix was used to obtain a set of weightings for the predictor. These values were then used for processing the stored signals. The updated values of the predictor coefficients need to be transmitted to the receiver once



(a) Pixels Employed in Predicting S_0



(b) Transmitter



(c) Receiver

Figure 12. One-Dimensional DPCM Coding System

every 5 milliseconds. Atal and Schroeder used the variable predictor with a two-level quantizer and reported good coding results. Although identical systems can be implemented for coding pictorial data, this type of system has not been reported on in the open literature. Instead, researchers have used adaptive DPCM systems with a fixed and simple predictor and an adaptive quantizer.

2.4.3.2 DPCM Systems with Adaptive Quantizers

A DPCM system with a fixed predictor will have a nonstationary differential signal for nonstationary data. Using a fixed quantizer, nonstationary differential signals would cause an abnormal saturation or a frequent utilization of the smallest level in the quantizer. To remedy this situation, the threshold and the reconstruction levels of the quantizer must be made variable to expand and contract according to signal statistics. Adaption of the quantizer to signal statistics is accomplished using various approaches. Virupaksha and O'Neal [7] suggested an adaptive DPCM system for speech signal that stores 25 samples of the differential signal to obtain an estimate for the local standard deviation of the signal. Then the stored signal is normalized by the estimated standard deviation and is quantized using a fixed quantizer. Naturally, the scaling coefficient must be transmitted once for every 25 samples for receiver synchronization. Ready and Spencer [8] use a similar approach in a system called Block-Adaptive DPCM, which they use for bandwidth compression of monochrome images. In Block-Adaptive DPCM systems, a block of M samples is stored and is normalized by n possible constants. The total distortion for all M samples using each normalizing constant is calculated at the encoder. The normalizing constant giving the smallest distortion is used to scale the samples in the block prior to their quantization and transmission. The system requires $(\log_2 n)/M$ binary digits per sample overhead information for receiver synchronization. Ready and Spencer use a two-dimensional DPCM system employing 3 adjacent samples in its predictor and use a block of 16 samples with four possible normalizing constants. They report a 36% reduction in bit rate over a similar nonadaptive DPCM system at about 2 bits per sample. The improvement in performance is less at higher bit rates.

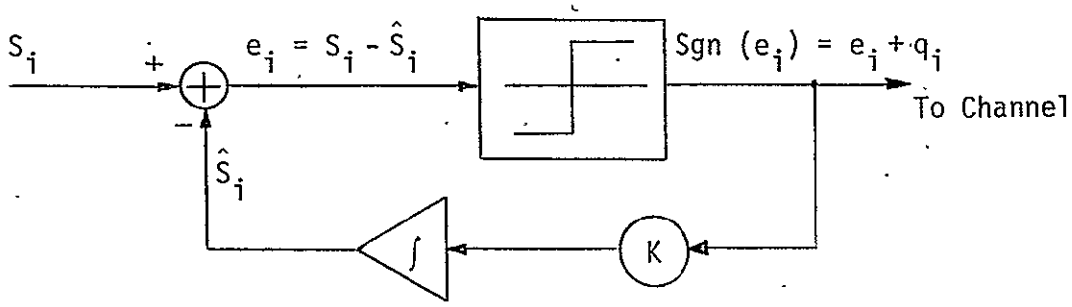
A different approach, which has not appeared in the technical literature, is a DPCM system with a variable set of thresholds and reconstruction levels. This is the self-synchronizing approach used in adaptive delta modulators where the step size contracts and expands depending upon the polarity of sequential output levels. In a DPCM quantizer, the set of threshold and reconstruction levels would contract and expand depending upon the sequential utilization of inner or outer levels of the quantizer. For instance, a variable quantizer can be designed where all reconstruction levels expand by a factor of P (for some optimum value of P) upon two sequential happenings of the outermost level and they would contract by a factor of 1/P upon two sequential happenings of the smallest level. This system has the advantage of being completely adaptive and does not require any overhead information because the receiver is self-synchonizing.

2.4.4 Delta Modulation (DM)

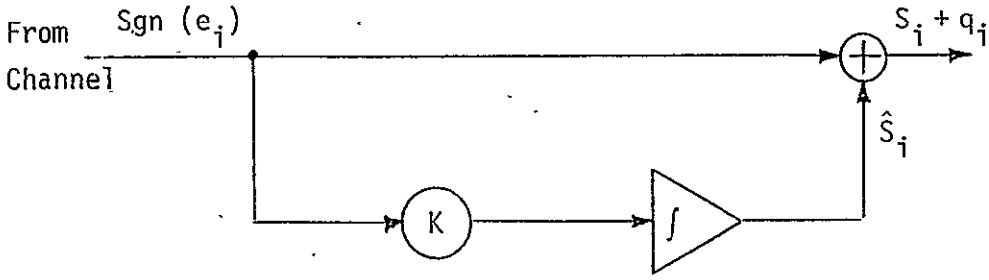
Delta modulation is a simple type of predictive coding system and is essentially a one-digit DPCM system. The delta modulator attempts to transmit the quantized difference between successive samples of the signal rather than the samples themselves. The output of a prediction circuit is differenced with the signal; the difference is quantized and encoded into a PCM sequence.

With linear or simple delta modulation, the quantizer contains only two levels and the predictive circuit is an integrator whose output is fed back to the input. The integrator output is compared with the input signal to produce a difference signal. If, at certain periodic sampling intervals, the difference is positive, a 1 is transmitted; otherwise, a 0 is transmitted. A function diagram of the linear delta modulator is shown in Figure 13. The receiver decodes the received signal by using the integrator (as in the feedback circuit of the delta modulator) to reconstruct the signal which may be added to the error signal. If slightly better quality is desired, the integrator in the feedback circuit may be replaced by a more complex circuit having a number of integrators whose outputs are combined in such a manner as to enhance the prediction function.

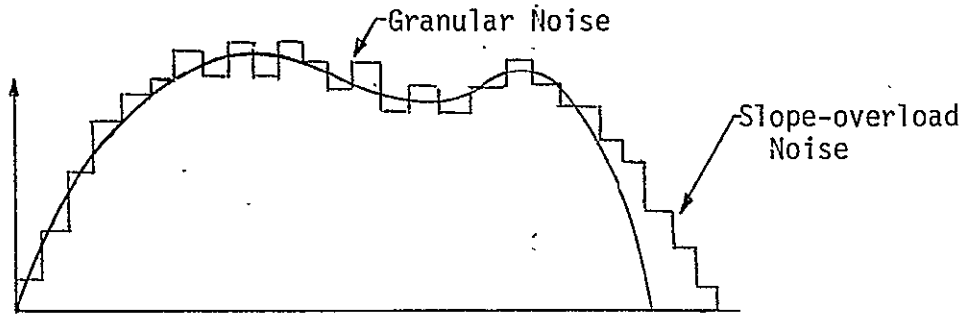
To date, the most comprehensive analysis of delta modulation (linear and adaptive) has been carried out by Abate [9]. This



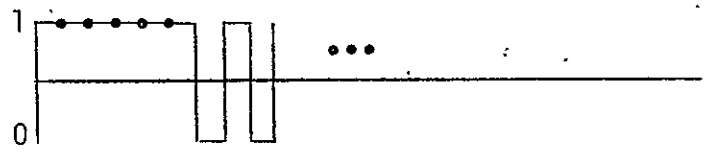
(a) Modulator



(b) Demodulator



(c) Response to Input Signal



(d) Output Signal

Figure 13. Linear DM System

investigation was carried out assuming speech-like signals having an exponential probability function. The investigation considered only single integrators in the feedback loop.

An important term in studying delta modulation is the slope loading factor, since the delta-mod predictor is attempting to follow the changes in the voice signal. The slope capability (i.e, the ability to follow changes in the voice signal) must be greater than the slope of the input voice signal. The slope capability is just the product of the step size k (the quantizer level) and the sampling rate f_s. Therefore, the following condition must hold if the system is not overloaded,

$$k f_s > |f'(t)| \tag{2.33}$$

where $|f'(t)|$ represents the magnitude of the input signal derivative with respect to time.

The slope overload parameter is defined as

$$s = k f_s / \sqrt{D} \tag{2.34}$$

where

$$D = \int_0^{\omega_m} \omega^2 F(\omega)\, d\omega \tag{2.35}$$

and $F(\omega)$ = one-sided power spectrum of the signal

ω_m = the maximum radian frequency to which the signal is bandlimited prior to encoding.

Abate presents values for $F(\omega)$ and s to be used in his analysis; also, s is defined in terms of the bandwidth expansion factor B, which is the ratio of the bandwidth required of the digital channel to that of the signal. For delta modulation systems, B is one-half the ratio of sampling rate to signal bandwidth, or $f_s/2f_m$.

Other parameters which are valuable in evaluating the performance of a delta modulator system are

N_G = granular noise, which is the variance between the predicted feedback pulse and the input signal

N_o = noise due to slope overload.

Both N_G and N_o are functions of s with N_G predominating for large values of s and N_o predominating for small values of s. The total noise power (quantization noise) is the sum of these two noise contributions.

In the linear DM system, the quantization noise ($N_Q = N_G + N_o$) is sensitive to small changes in the mean power or the signal, and the range of s over which S/N_Q is near maximum is small. A change in signal power produces a change in s. When s is not near an optimum value, N_Q will not be near its minimum value and the delta modulator will not perform well. Thus, it is desirable to force the delta-mod system to adaptively vary s as a function of the input signal. The adaptive delta-mod system performs this function and will be discussed in the next section.

2.4.5 Adaptive Delta Modulation (ADM)

In the previous section, the need for an adaptive delta modulator was motivated by considering the signal-to-noise ratio as a function of slope overload factor. The need for an adaptive delta modulator and motivation for varying the step size can be demonstrated by studying equation (2.33).

The relationship between f_s and $|f'(t)|$ in equation (2.33) is seen to be proportional if k is held constant. Thus, if the deviation of the noise signal is large, then f_s must be large. Yet, $|f'(t)|$ varies as a function of time so that, to avoid slope overload, f_s would have to be selected for the condition where $|f'(t)|$ is maximum. Selecting f_s in this manner is not efficient since $|f'(t)|$ will have smaller values much of the time. The sampling rate f_s must be constant for a digital communication system, but k can vary.

The objective of the adaptive delta modulation system is to maintain optimal slope loading and maximum S/N_Q by controlling the value of the slope loading factor by varying the step size k. The distinction between different adaptive delta modulation systems is based on how k is varied. Those to be discussed below are the Abate [9] and the High Information Delta Modulation (HIDM) [10] algorithms.

Before proceeding to the discussion of particular algorithms, it is necessary to discuss concepts pertaining to adaptive delta modulation systems in general. Essentially, there can be both discrete and continuous

methods of varying the step size. The former observes the binary pulse sequence at the quantizer output and changes the step size accordingly and is called "discrete adaptive delta modulation." This system is shown in Figure 14. The latter, called "continuous adaptive delta modulation," is considered to be applicable primarily to voice communications and will not be discussed in this report.

In the discrete adaptive delta modulation system, the gains K_i are chosen as a function of the observation of the sequence of output pulses leaving the two-level quantizer (Figure 14). These gains multiply the step size so that the maximum slope capability of the system remains less than the derivative of $|f'(t)|$ [see equation (2.33)]. Slope overload is no longer the controlling degradation, since the system is able to increase its step size at the sampling rate, until the derivative of the signal is greater than the maximum slope capability of the system so that

$$|f'(t)| > K_n k f_s \tag{2.36}$$

where K_n is the maximum gain. Given this situation, a normalized slope loading factor can be defined:

$$s' \equiv (K_n k f_s)/\sqrt{D} \ . \tag{2.37}$$

With this parameter, many results derived for linear delta modulation can be applied to adaptive delta modulation.

Selection of the step size and the maximum gain K_n are key considerations in designing adaptive delta modulation systems. Abate has developed the following equations which give the step size as

$$k = \left(\frac{\sqrt{D}}{f_s} \ln 2B\right) \sqrt{s_1} \tag{2.38}$$

and the maximum gain as

$$K_n = \sqrt{s_2/s_1} \tag{2.39}$$

where s_1 = smallest value for signal power

s_2 = largest value for signal power.

Control and selection of the intermediate values must also be determined. It is desirable to control the values based on a sequence pulse. This can be justified in two different ways. First, when slope

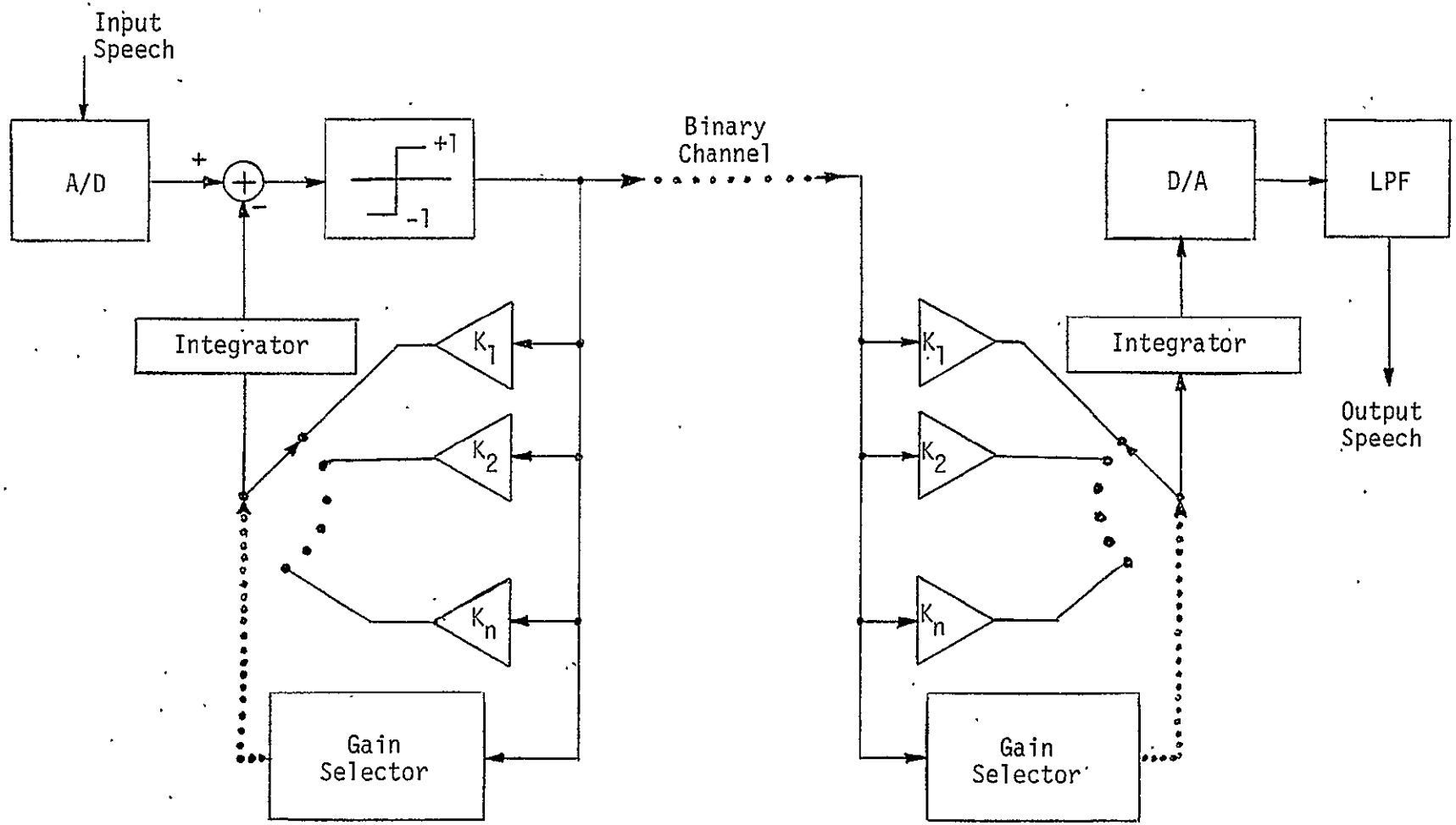


Figure 14. Discrete Adaptive Delta Modulator

overload occurs, causing suboptimal performance, the quantizer output is a series of pulses of the same polarity. In response, a gain K_i greater than K_{i-1} (the previously used gain) is selected such that the new larger step size is K_i multiplied by the smallest step size k, $K_i k$. If the polarity remains unchanged, the step size is incrementally increased to $K_{i+1}k$, $K_{i+2}k$, ..., until the largest value $K_n k$ is reached. The step size incrementally decreases when a polarity reversal occurs. In the decoder, an identical process takes place. The result of switching these gains is to produce a sequence of equally likely pulses of ones and zeroes which insures that slope overload will not occur (or is not likely to occur).

A second way of viewing this situation is from an information theory point of view. It is well known (as shown in any information theory text) that the greatest amount of information is transmitted from a source to a receiver when, at the receiver, the pulse values to be determined are equally likely. Thus, attempting to produce a sequence of equally likely pulse values by adjusting the step size results in maximizing the information content of the signal, thus reducing the redundancy.

With the Abate algorithm, the gains are set to be linearly equal to the consecutive sequence of pulses with the same polarity until the maximum gain is achieved. This represents linear gain factor increments as follows

$$\begin{aligned} K_i &= i \\ K_n &= n \text{ for } i > n \ . \end{aligned} \tag{2.40}$$

For voice transmission, NASA has implemented a modified Abate algorithm with step sizes chosen as shown in Table 6. The difference between this algorithm and the one studied by Abate is the manner in which the gain is decreased when a sequence of the same polarity terminates. With the regular Abate, the gain is incremented to a smaller value, whereas with the modified Abate, the gain goes to the smallest step size k when a sequence of the same polarity is ended by a reversal. Also, the Abate algorithm as implemented by NASA uses eight different gain values which differs from the four used by Abate.

Another possible choice of gain values which was proposed by Winkler [10] and which would appear to be more suitable for image

Table 6. Step-Size Algorithm (Abate)

Encoded Sequence (latest)	Delta Step-Size (Unit)
x x x x x x 0 1	1
x x x x x 0 1 1	2
x x x x 0 1 1 1	3
x x x 0 1 1 1 1	4
x x 0 1 1 1 1 1	5
x 0 1 1 1 1 1 1	6
0 1 1 1 1 1 1 1	7
1 1 1 1 1 1 1 1	8
0 0 0 0 0 0 0 0	-8
1 0 0 0 0 0 0 0	-7
x 1 0 0 0 0 0 0	-6
x x 1 0 0 0 0 0	-5
x x x 1 0 0 0 0	-4
x x x x 1 0 0 0	-3
x x x x x 1 0 0	-2
x x x x x x 1 0	-1

x — don't care, peak-to-peak input analog signal = 256 units.

transmission, in which rapid level changes can occur (at edges of objects), is to let the gains vary exponentially with sequences of pulses of the same polarity as follows:

$$K_i = 2^{i-1} . \tag{2.41}$$

This system has been called High Information Delta Modulation (HIDM) by Winkler.

2.4.6 Multi-Level Adaptive Delta Modulation

A conventional delta modulator has two levels and is, strictly speaking, a one-bit DPCM. It is possible to apply the adaptive approach of the previous section to standard DPCM. In this case, the adaptive DPCM is often referred to as "multi-level adaptive delta modulation." Of particular interest is a tri-state delta modulator where the video signal is divided into three main regions of activity:

(1) Acquisition - The estimate tries to "catch up" with a rapid change in grey level.

(2) Tracking - The estimate follows a slowly changing grey level.

(3) Transition - The estimate must change from acquisition to tracking.

Overshoots (and undershoots) that are inherently connected with fast acquisition behavior of an ADM and are particularly disturbing in video transmission can easily be suppressed by the tri-state delta modulator as it switches to its tracking mode. Because of its importance for video data compression, the tri-state delta modulator is discussed in detail in Section 6.0.

3.0 ANALYSIS-SYNTHESIS METHODS - TRANSFORM CODING

3.1 General

Even though fine sampling and quantization of an image are essential for desirable subjective quality of a digital picture, from the viewpoint of a statistician, the information in the picture can be conveyed quite adequately without all these variables. On the other hand, one cannot simply discard a part of these variates because of their equal statistical significance and the adverse effect this would have on the subjective quality of the picture.

An approach to this problem is to transform the image samples to a new set of variates that have a complementary degree of significance in contributing to both the information content and the subjective quality of the resulting picture. Then one can discard the less significant of these variables without affecting the statistical information content of the picture or causing a severe degradation in the subjective quality of the resultant picture. The method of "principal components" is a coordinate transformation with the above properties.

To illustrate the application of this coordinate transformation to pictorial data, consider two samples on a picture. Let X_1 and X_2 stand for the values these two pixels could assume. From the fact that these variables are correlated, one can easily conclude that the sample values often fall inside the region shown on Figure 15. Now if one rotates the coordinate system to positions indicated by y_1 and y_2, two new variables would be obtained—one having a larger variance than the other, though the sum total of the variances is invariant under the transformation. Performing the rotation of coordinates with all N^2 pixels, one obtains N^2 new variates that are uncorrelated and have monotonically decreasing variances. Indeed, the method of principal components, also known as the discrete Karhunen-Loève Transformation, results in variates with a maximum compaction of energy in the first M components that one desires to keep. Since the information content of the sampled picture is invariant under this linear transformation, and the variance of a variable is a measure of its information content, the coding strategy should be first to discard variates with low variances. Then, since quantizing each number corresponds to approximating its amplitude with the nearest rational number and thus corresponds to a loss of information, one should attempt to

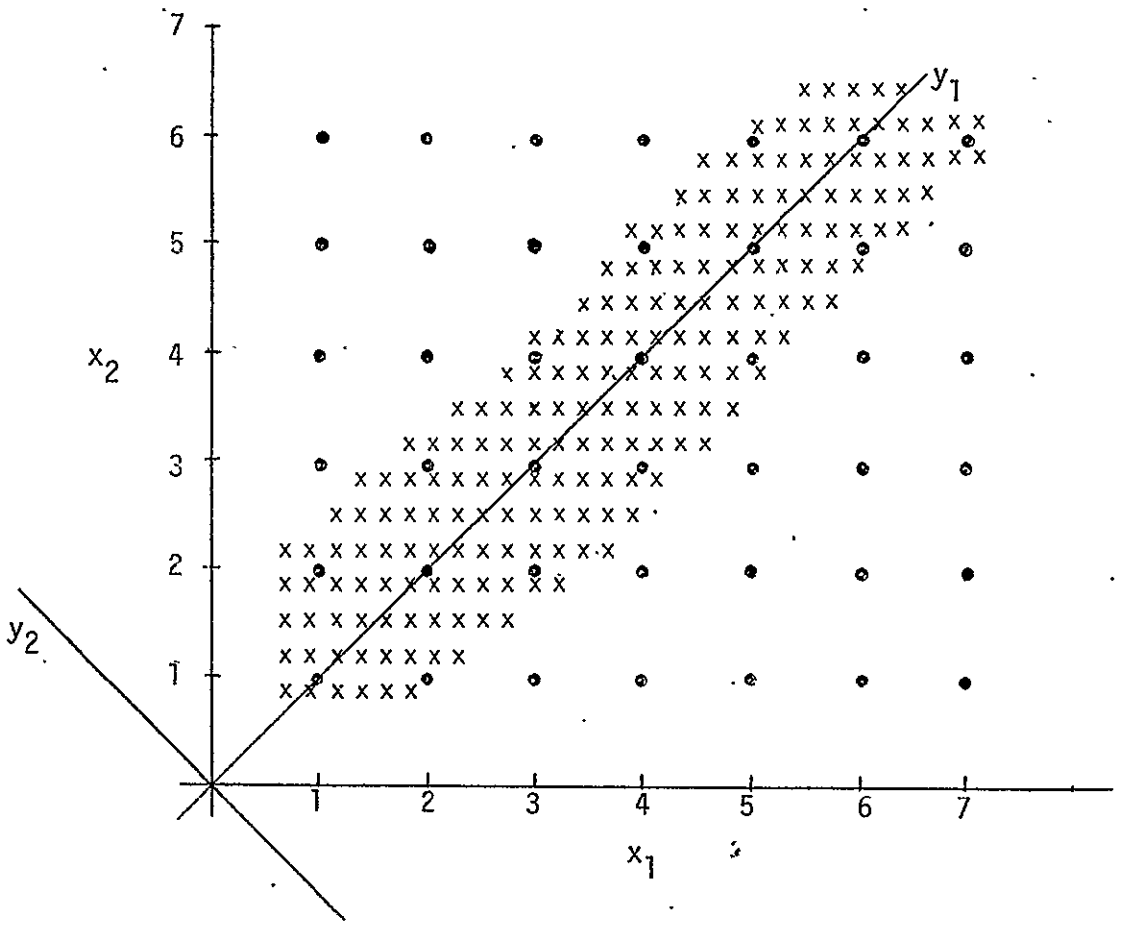


Figure 15. Coordinate System for the Grey Levels of Two Adjacent Pixels x_1 and x_2. The most likely values are the ones in the shaded area. y_1 and y_2 are the new coordinate system.

impose less of a distortion on those variates having larger variances. The above transformation corresponds to a matrix transformation of a vector. For a two-dimensional data (image) it is accomplished by ordering the sampled data in a vector form and performing the transformation. The block diagram of this transformation is shown in Figure 16. Denoting the data vector and the transform vectors by X and Y and the error introduced by quantization with vector Q, the vector X* reconstructed at the receiver is

$$X^* = X + A^{-1}Q \tag{3.1}$$

where A corresponds to the operator in the method of principal components. Then the resulting mean squared error is

$$\varepsilon^2 = \text{tr}\, E\left\{A^{-1} Q Q^T A^{-1^T}\right\} = \text{tr}\left\{A^{-1}\, E\{Q Q^T\}\, A^{-1^T}\right\} \tag{3.2}$$

where tr denotes the trace of a matrix. Since AX is an uncorrelated vector, it follows that Q is also uncorrelated; thus, $E\{Q Q^T\}$ is diagonal with the i^{th} element in its diagonal referring to the variance of the quantization error σ_q^2 in the i^{th} component of the Y vector. It is shown that the overall coding error is minimized if M_b binary digits are assigned to various components of the transformed vector Y so that the variance of the quantization error σ_q^2 for all components of Y are identical. Since the variance of the quantization error is shown to be directly proportional to the variance of the variable being quantized, an equal quantization noise for various components of Y implies assigning more bits to the components with large variances and fewer bits to those with small ones. Of course, assigning zero bits to a particular component of Y means that the particular component is ignored in the process of the transformation storage. Note that the above may not be possible since the number of binary digits assigned to each sample must be an integer. Thus, only an approximate solution is possible in practice.

Elimination of transform coefficients is generally done either on a zonal filtering basis (in which a fixed number of coefficients with largest energy are retained) or on a threshold filtering basis (in which those coefficients which exceed a fixed threshold are retained). Those coefficients which are judged to be less significant than others may be coded using fewer bits or they may be discarded entirely.

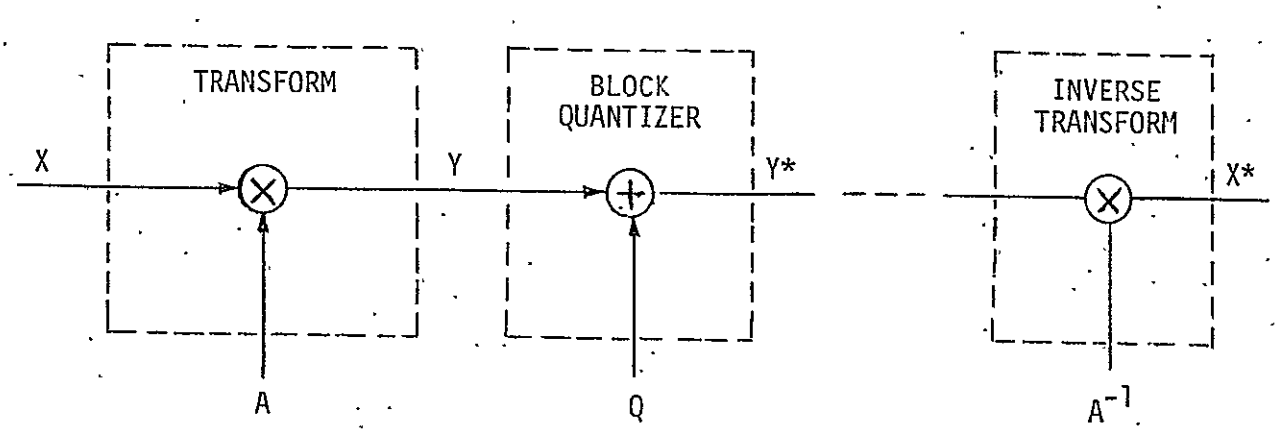


Figure 16. Block Diagram of Transform Coding System

As indicated in Figure 17, a set of N video samples may be used to derive a set of N transform coefficients. The generalized transform relationship is

$$\hat{S}(t) = \sum_{n=0}^{N-1} C_n \phi_n(t) , \quad 0 \leq t \leq T \qquad (3.3)$$

where C_n = transform coefficients

ϕ_n = set of orthogonal basis functions.

Of the N transform coefficients which are calculated, it is assumed that only M ($M \leq N$) of these are transmitted, and that more bits may be used to represent those transmitted coefficients that are deemed to have more significance.

The set of orthogonal basis functions $\phi_n(t)$ that are used to define a particular transform is practically unlimited. It is known that the Karhunen-Loève transformation is optimum in the mean-square-error sense, but it is also known that other suboptimum transformations are computationally simpler due to the availability of fast algorithms.

3.2 Walsh-Hadamard Transform Techniques

A Hadamard matrix is an orthogonal matrix in which all of the elements take the values +1 or -1. One particular subset of Hadamard matrices is the Walsh-Hadamard, for which there exists a fast transform algorithm which reduces the computational requirements. Figure 18 illustrates a set of one-dimensional Walsh functions of length $2^3 = 8$, while Figure 19 shows the normal Hadamard matrices of order N = 2, 4, and 8. By rearranging the rows of the Hadamard matrices in sequency order (sequency ≡ 1/2 × number of sign changes, rounded up to nearest integer), the set of discrete Walsh Function Matrices shown in Figure 20 is obtained. The rows of these matrices can be used as the orthogonal basis functions for the one-dimensional Walsh-Hadamard transform, which is illustrated in Figure 21.

Figure 22 provides an example application of the one-dimensional Walsh-Hadamard transform to a set of four data samples (S_0, S_1, S_2, S_3) to obtain the four transform coefficients C_0, C_1, C_2, and C_3, while Figure 23 shows how the original data samples can be recovered from the

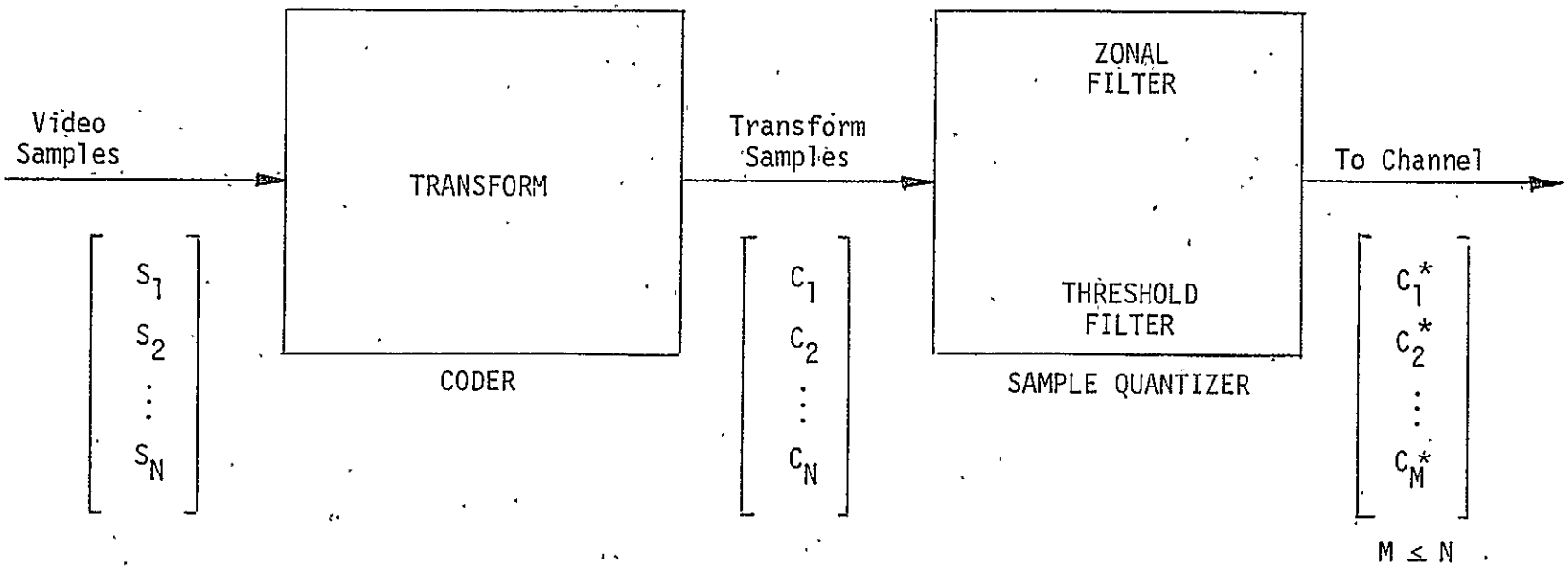


Figure 17. Conceptual Transform Coder

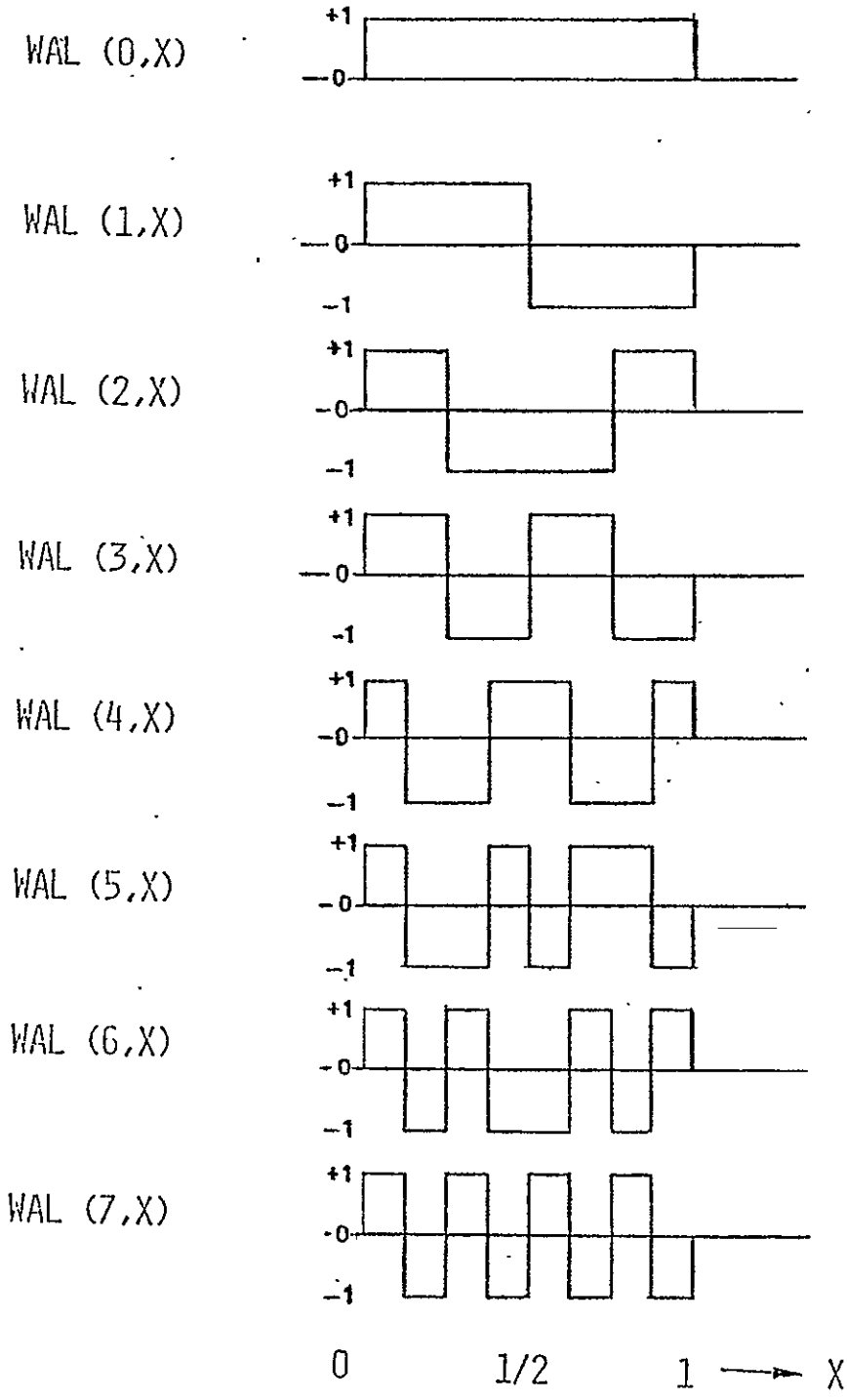


Figure 18. Walsh Functions

$$H_N = (H_2)^{LOG2N}$$

SEQUENCY

N = 2

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

0
1

N = 4

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

0
2
1
1

N = 8

$$H_8 = \begin{bmatrix} H_2 & H_2 & H_2 & H_2 \\ H_2 & -H_2 & H_2 & -H_2 \\ H_2 & H_2 & -H_2 & -H_2 \\ H_2 & -H_2 & -H_2 & H_2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

0
4
2
2
1
3
1
3

Figure 19. Hadamard Matrices

$$\underline{N = 2} \qquad \text{WAL}(k,j) = \begin{matrix} j = 0 & 1 \end{matrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{matrix} k = 0 \\ 1 \end{matrix}$$

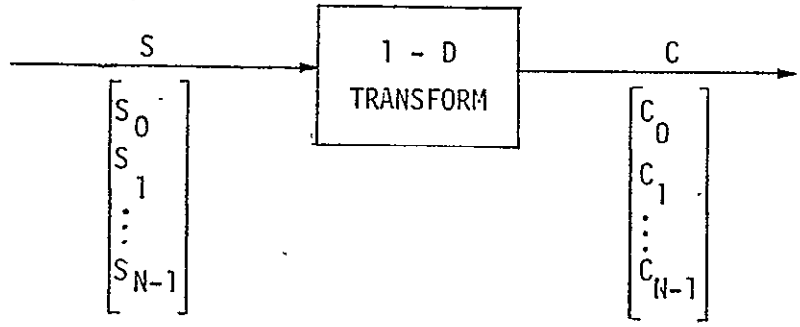
$$\underline{N = 4} \qquad \text{WAL}(k,j) = \begin{matrix} j = 0 & 1 & 2 & 3 \end{matrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{matrix} k = 0 \\ 1 \\ 2 \\ 3 \end{matrix}$$

$$\underline{N = 8} \qquad \text{WAL}(k,j) = \begin{matrix} j = 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \begin{matrix} k = 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix}$$

Figure 20. Discrete Walsh Function Matrices (Hadamard Matrices with Rows in Sequency Order)



$$C_k = \frac{1}{N} \sum_{j=0}^{N-1} S_j \; WAL\,(k,j) \qquad k = 0, 1, \ldots, N-1$$

$$S_j = \sum_{k=0}^{N-1} C_k \; WAL\,(k,j) \qquad j = 0, 1, \ldots, N-1$$

IN MATRIX NOTATION:

$$\underset{N\times 1}{[C]} = \frac{1}{N} \underbrace{[WAL(k,j)]}_{N\times N} \underset{N\times 1}{[S]}$$

$$\underset{N\times 1}{[S]} \qquad \underbrace{[WAL(k,j)]}_{N\times N} \underset{N\times 1}{[C]}$$

Figure 21. Finite One-Dimensional Walsh-Hadamard Transform

$$[S] = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

TRANSFORM COEFFICIENTS:

$$C_0 = \frac{1}{4}\left[S_0\ \mathrm{WAL}(0,0) + S_1\ \mathrm{WAL}(0,1) + S_2\ \mathrm{WAL}(0,2) + S_3\ \mathrm{WAL}(0,3)\right]$$

$$= \frac{S_0 + S_1 + S_2 + S_3}{4} = \frac{6}{4}$$

$$C_1 = \frac{1}{4}\left[S_0\ \mathrm{WAL}(1,0) + S_1\ \mathrm{WAL}(1,1) + S_2\ \mathrm{WAL}(1,2) + S_3\ \mathrm{WAL}(1,3)\right]$$

$$= \frac{S_0 + S_1 - S_2 - S_3}{4} = \frac{2}{4}$$

$$\vdots$$

$$[C] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 6 \\ 2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}$$

Figure 22. Example of One-Dimensional Walsh-Hadamard Transform

$$[S] = [WAL(k,j)][C]$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}$$

$$[S] = \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

Figure 23. Example of Inverse Walsh-Hadamard Transform

transform coefficients. Obviously, from this example, it can be seen that the C_2 and C_3 coefficients need not be transmitted over the channel.

3.3 Other Transform Techniques

In addition to the Walsh-Hadamard transform, there are several other transforms worthy of note. The Karhunen-Loeve transform (KLT) is the optimum in terms of mean-square-error. Any transform can be expressed in the form of (3.3) where $\phi_n(t)$ is the set of orthogonal basis functions. The basis functions for the KLT are the characteristic functions of the integral equation:

$$\int_0^T R(t,r)\ \phi_n(r)\ dr = |\sigma_n|^2\ \phi_n(t)\ , \tag{3.4}$$

where $R(t,r) = E\{S(t),S^*(r)\}$ = the autocorrelation of the video signal $S(t)$. The solutions to (3.4) are the characteristic values $|\sigma_n|^2$ and the characteristic functions $\phi_n(t)$. Note that

$$\int_0^T \phi_n(t)\ \phi_m(t)\ dt = \begin{cases} 1\ , & \text{if } m = n \\ 0\ , & \text{if } m \neq n \end{cases} \tag{3.5}$$

and the transform coefficients are given by

$$C_n = \int_0^T S(t)\ \phi^*_n(t)\ dt \tag{3.6}$$

where

$$E\{C_n C_m\} = \begin{cases} 1\ , & \text{if } m = n \\ 0\ , & \text{if } m \neq n\ . \end{cases} \tag{3.7}$$

The difficulty in using KLT is solving (3.4) to determine characteristic functions. Therefore, other suboptimum transforms are considered that are computationally simpler.

The Fourier transform is often used. The basis functions $\phi_n(t)$ for the Fourier transform are

$$\phi_n(t) = e^{jn\omega t} \tag{3.8}$$

and the transform coefficients C_n are given by

$$C_n = \frac{1}{T}\int_0^T s(t)\, e^{-jn\omega t}\, dt \,. \tag{3.9}$$

If the video signal is sampled every τ seconds, where $N\tau = T$, then the discrete Fourier transform (DFT) is given by

$$C_n = \sum_{k=0}^{N-1} s(kt)\, e^{-j\Omega\tau nk} \tag{3.10}$$

where

$$\Omega = \frac{2\pi}{N\tau} \tag{3.11}$$

and $s(k\tau)$ is sampled video signal. The inverse DFT is given by

$$s(k\tau) = \frac{1}{N}\sum_{n=0}^{N-1} C_n\, e^{j\Omega\tau nk} \,. \tag{3.12}$$

One of the advantages of the DFT is the existence of a fast algorithm (FFT) to compute the transform [11]. The discrete cosine transform (DCT) can also be computed using the FFT algorithm. The performance of the DCT in the mean square error sense is close to the optimum performance of the KLT. The transform coefficients C_n are given by

$$C_0 = \frac{\sqrt{2}}{N}\sum_{k=0}^{N-1} s(k\tau)$$

$$C_n = \frac{2}{N}\sum_{k=0}^{N-1} s(k\tau)\cos\left(\frac{(2k+1)n\Omega}{2N}\right) \tag{3.13}$$

and the inverse DCT is defined as

$$s(k\tau) = \frac{1}{2}C_0 + \sum_{n=1}^{N-1} C_n \cos\left(\frac{(2k+1)n\Omega}{2N}\right) \,. \tag{3.14}$$

Using various transforms in conjunction with scalar Wiener filtering and unity signal-to-noise ratio (white zero mean noise), the performance of the DCT is very close to that of the optimum KLT [12] as shown in Figure 24. Note that the DCT is significantly better in a mean square error sense than the DFT or the Walsh-Hadamard transform.

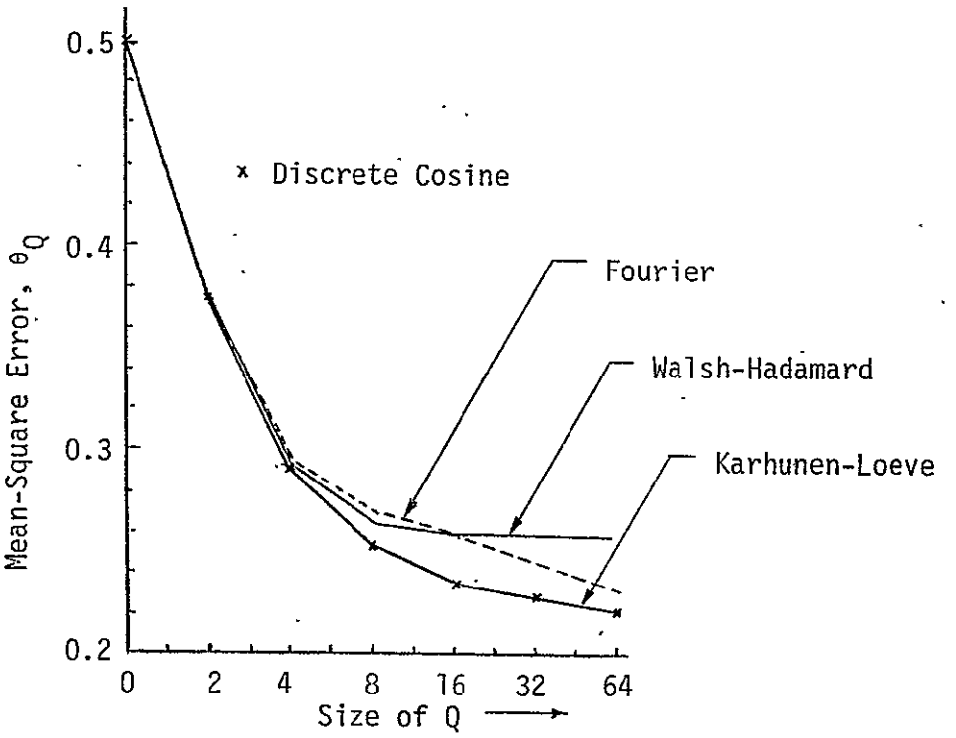


Figure 24. Mean-Square Error Performance of Various Transforms for Scalar Wiener Filtering

In considering using transforms for video signals, it is worth noting that a major attribute of an image transform is the transform compaction of the image energy to a few of the transform domain samples. A high degree of energy compaction will result if the basis vectors of the transform matrix "resemble" typical horizontal or vertical lines of an image. If the lines of a typical monochrome image are examined, it will be found that a large number of the lines are of constant grey level over a considerable length. The Fourier and Hadamard transforms possess a constant valued basis vector that provides an efficient representation for constant grey level image lines, while the Karhunen-Loeve transform has a nearly constant basis vector suitable for this representation. Another typical image line is one which increases or decreases in brightness over the length in a linear fashion. None of the transforms previously mentioned possess a basis vector that efficiently represents such image lines.

Shibata and Enomoto have introduced orthogonal transforms containing a "slant" basis vector for data of vector lengths of four and eight [13]. The slant vector is a discrete sawtooth waveform decreasing in uniform steps over its length, which is suitable for efficiently representing gradual brightness changes in an image line. Their work gives no indication of a construction for larger size data vectors, nor does it exhibit the use of a fast computational algorithm.

With this background, an investigation was undertaken to develop an image-coding slant-transform matrix possessing the following properties: (1) orthonormal set of basis vectors; (2) one constant basis vector; (3) one slant basis vector; (4) sequency property; (5) variable size transformation; (6) fast computational algorithm; and (7) high energy compaction. Figure 25 presents the slant transform basis functions [14] for $N = 16$.

Figure 26 contains a plot of mean-square error as a function of block size for several transforms for coding with an average of 1.5 bits/pixel. The figure indicates that the performance of the slant transform is quite close to the optimal Karhunen-Loeve transform. It is possible to achieve a slightly lower mean-square error for a given channel rate by employing Huffman coding of the quantized coefficients rather than constant word-length coding, but the coder will be much more complex to implement.

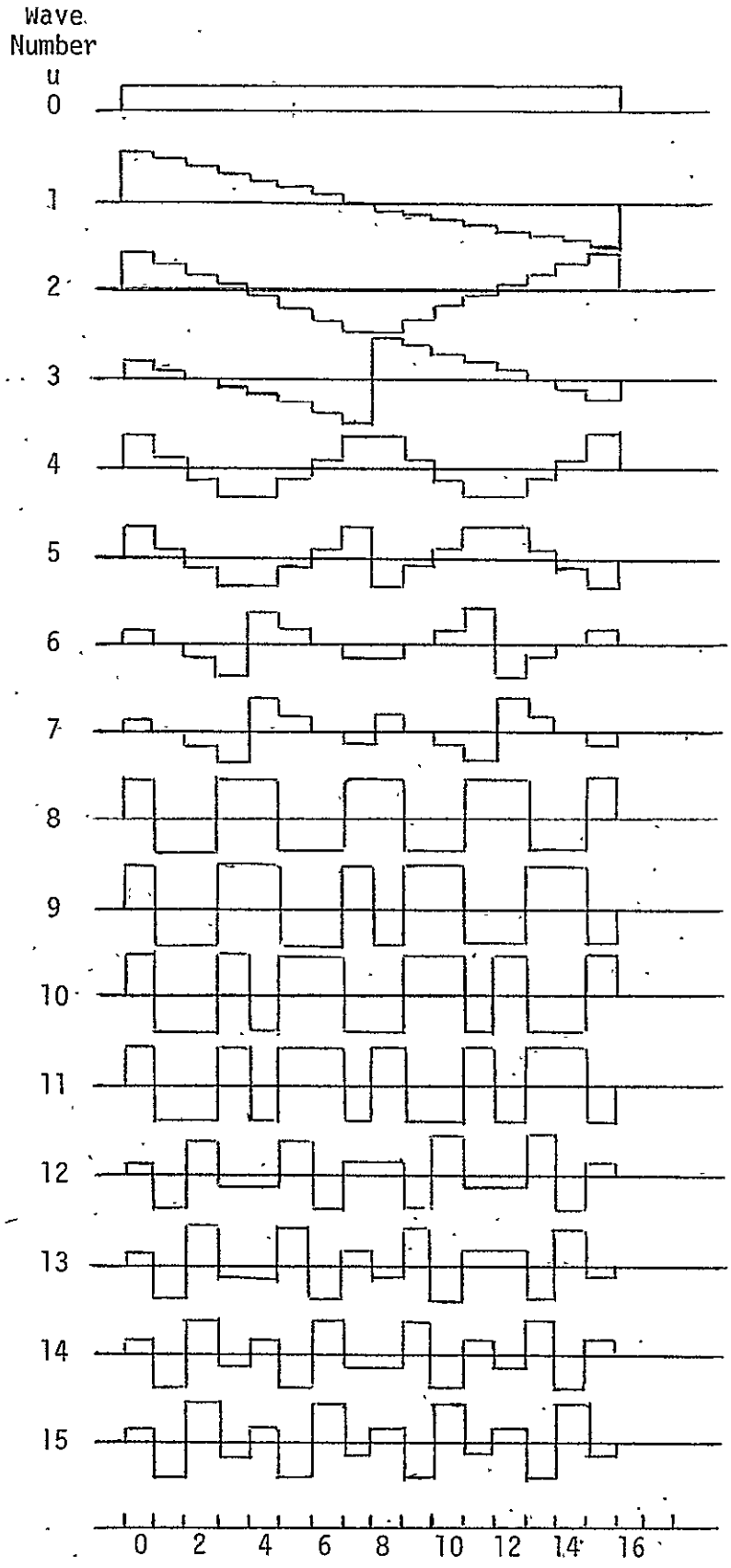


Figure 25. Slant-Transform Basis Waveforms

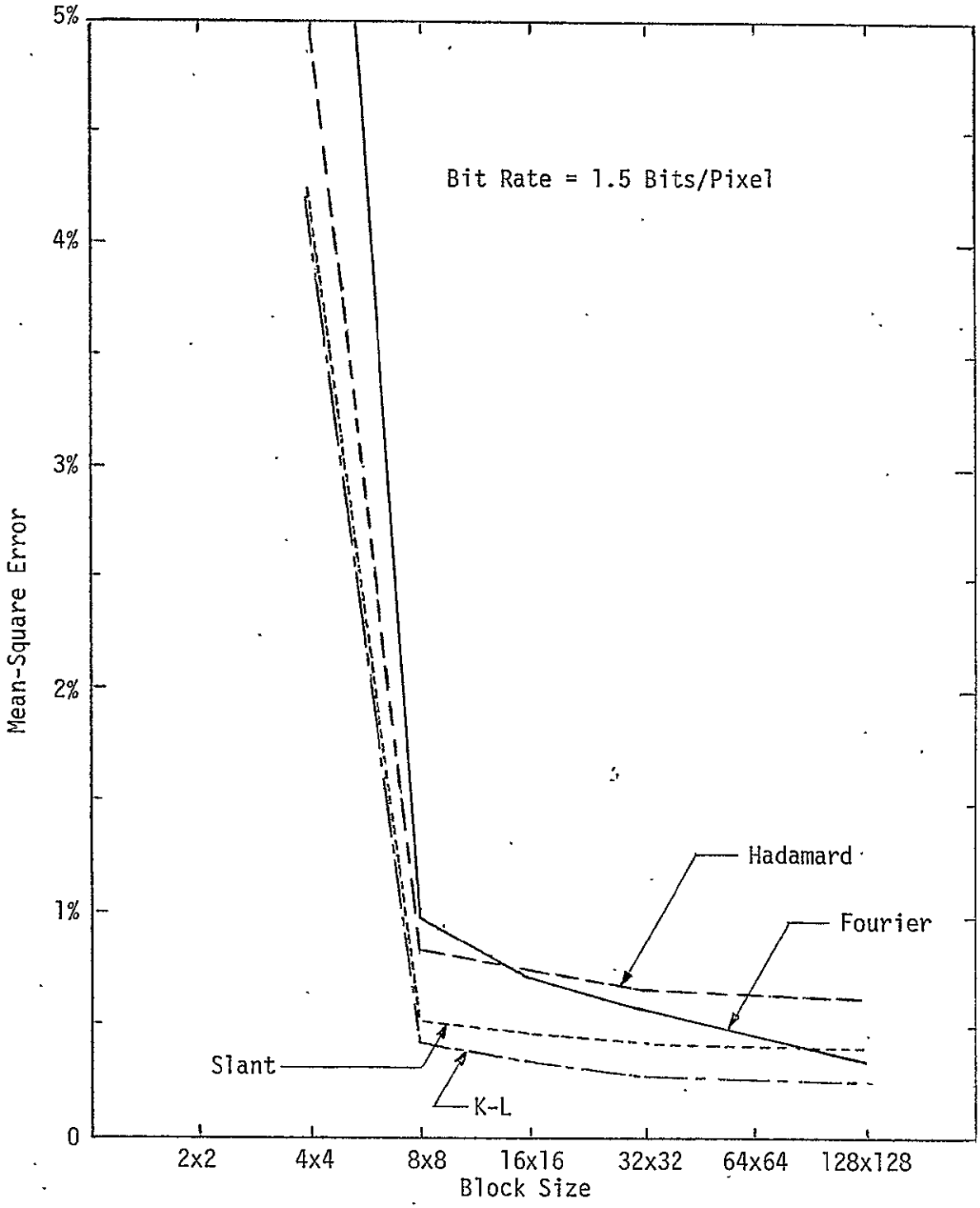


Figure 26. Mean-Square Error Performance of Image Transforms as a Function of Block Size

4.0 MULTIDIMENSIONAL TECHNIQUES

4.1 General

It is possible to achieve greater reduction in video redundancy by coding over the two spatial dimensions and the temporal dimension. It is straightforward to extend the one-dimensional data compression techniques to multidimensions just by applying them sequentially to each dimension. In fact, hybrids of various techniques could be used. For example, analysis of the transform and DPCM image coding techniques has disclosed that each possesses attractive characteristics and some limitations [15,16]. Transform coding systems achieve superior coding performance at lower bit rates; they distribute coding degradation in a manner less objectionable to a human viewer, show less sensitivity to data statistics (picture-to-picture variation), and are less vulnerable to channel noise. DPCM systems, on the other hand, when designed to take advantage of spatial correlation of image data, achieve a better coding performance at a higher bit rate. The equipment complexity and the delay due to the coding operations are minimal. Perhaps the most desirable characteristic of DPCM is the ease of design and the speed of the operation that has made it possible for DPCM systems to be used in coding television signals in real time. The limitations of DPCM are the sensitivity of even well-designed systems to picture statistics and the propagation of channel errors in a coded picture.

A hybrid coding system that combines the attractive features of both transform and DPCM coding has been developed [16]. This system exploits the correlation of the data in the horizontal direction by taking a one-dimensional transform of each line of the picture, then operating on each column of the transformed data using a one-element predictor DPCM system. Since the unitary transformation involved is a one-dimensional transformation of individual lines of the pictorial data, the equipment complexity and the number of computational operations are considerably less than that involved in a two-dimensional transformation. Theoretical and experimental results indicate that the hybrid system has good coding capability—one that surpasses both DPCM and the transform coding systems.

4.2 Two-Dimensional Spatial Techniques

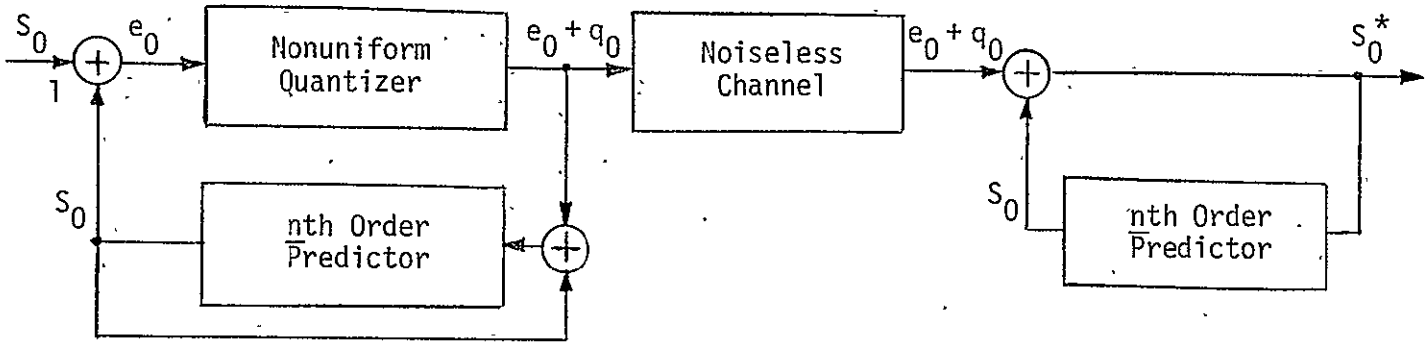
4.2.1 Two-Dimensional DPCM

Two-dimensional DPCM (2D-DPCM) can easily be implemented. In this case, the picture is represented by rows (lines) and columns. The estimate of a picture element is based on the previous adjacent picture element in the row and column as shown in Figure 27.

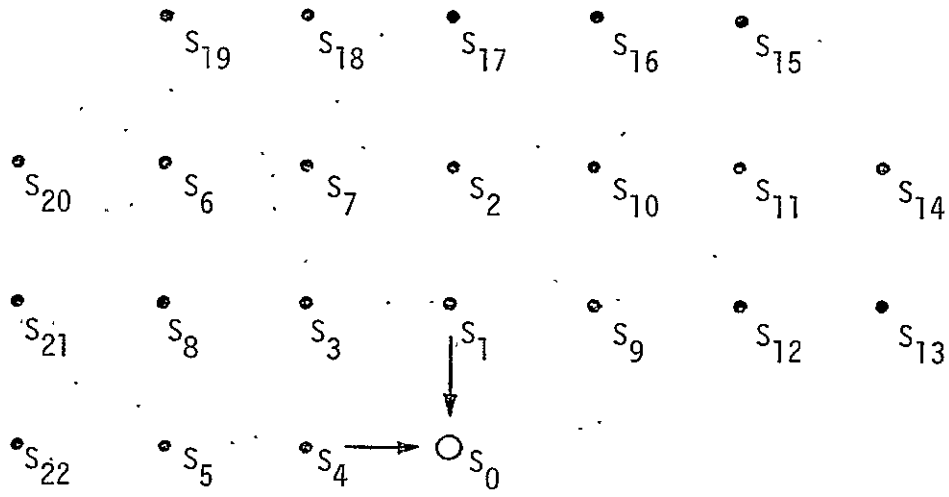
Figure 28 illustrates the functional operation of the 2D-DPCM encoder. Incoming digitized video information is brought in, and the difference between it and its predicted value is quantized by the fixed quantizer. The quantizer contains 8 symmetrically distributed cutpoints, each having its associated output value. The output of the fixed quantizer is recorded for transmission, combined with synchronization information in a multiplexer, and supplied to the transmission link. Additionally, the fixed quantizer output is summed with the previous adjacent sample values, and stored for one sample time in a latch. Simultaneously, the present and previous samples from the line under consideration are each scaled, and their difference stored in the Line Store Memory. Gating logic is provided such that samples are stored only during the active horizontal line time, and not during retrace. The output of the Line Store Memory, representing the adjacent samples from the previous line, is summed with the scaled output from the latch (representing the adjacent sample from the line under consideration), and is subtracted from the next adjacent sample on the line under consideration.

Figure 29 illustrates the functional operation of the Multiple Loop Adaptive 2D-DPCM Encoder. This system utilizes four nonadaptive loops as previously shown in Figure 28. However, each loop is fitted with a different quantizer. The output of each loop is stored in a block storage register (16 samples). The Block Select Logic continuously calculates the mean squared error for each block, at the sample rate. Upon completion of a block of data, the Block Select Logic selects for transmission the data block having the lowest mean squared error.

Figure 30 illustrates the single-loop block-adaptive 2D-DPCM encoder which utilizes multiple quantizers. Here, a 16 sample (data block) delay is provided, while the Gain Computation Logic determines the mean squared error and selects one of eight different quantizers within the Compressor Loop.



DPCM System



Picture Elements Employed in Predicting S_0

Figure 27. Block Diagram of Linear Predictive Coder and the Pixels Employed in Predicting Pixel S_0

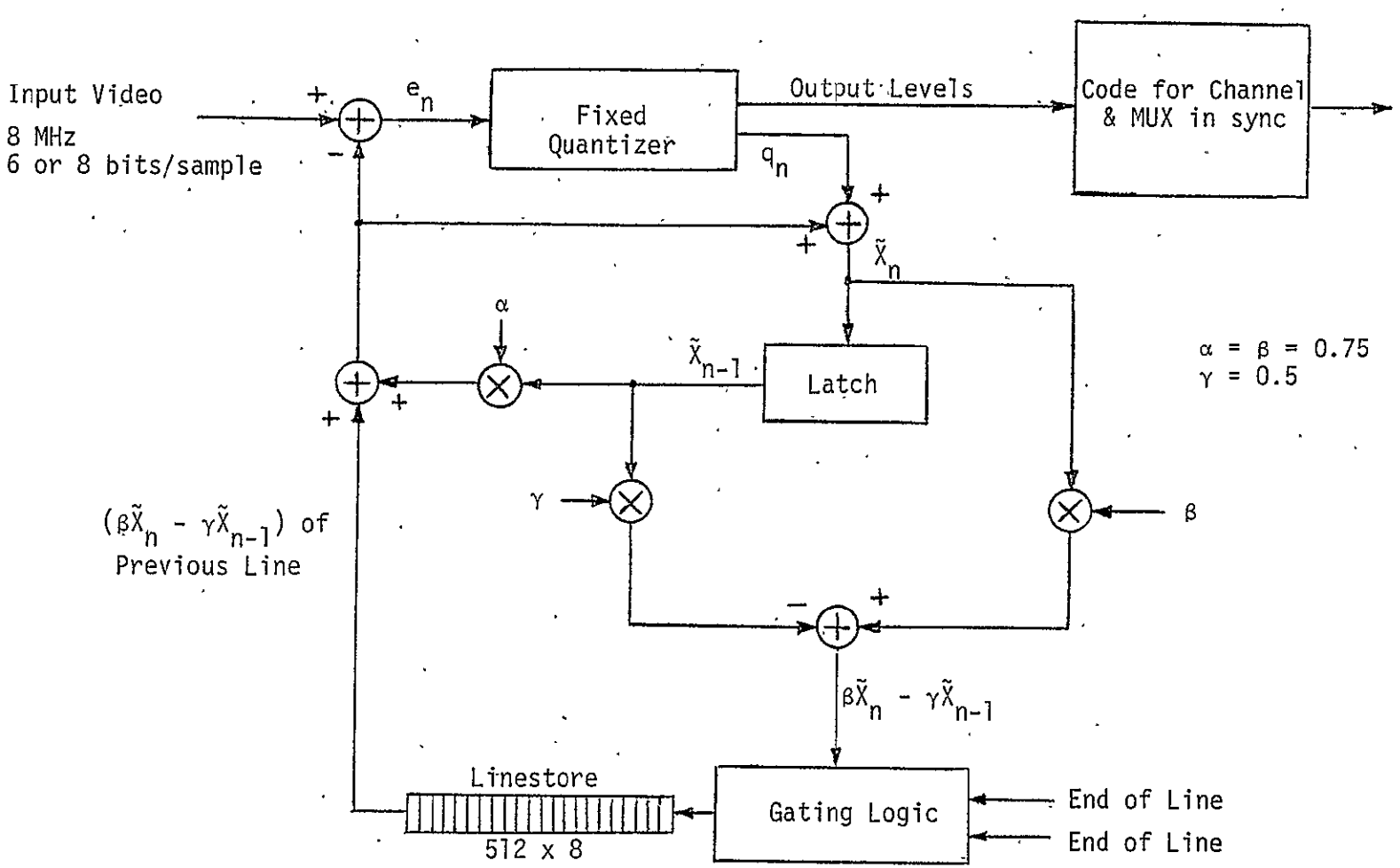


Figure 28. Non-Adaptive 2D-DPCM Encoder

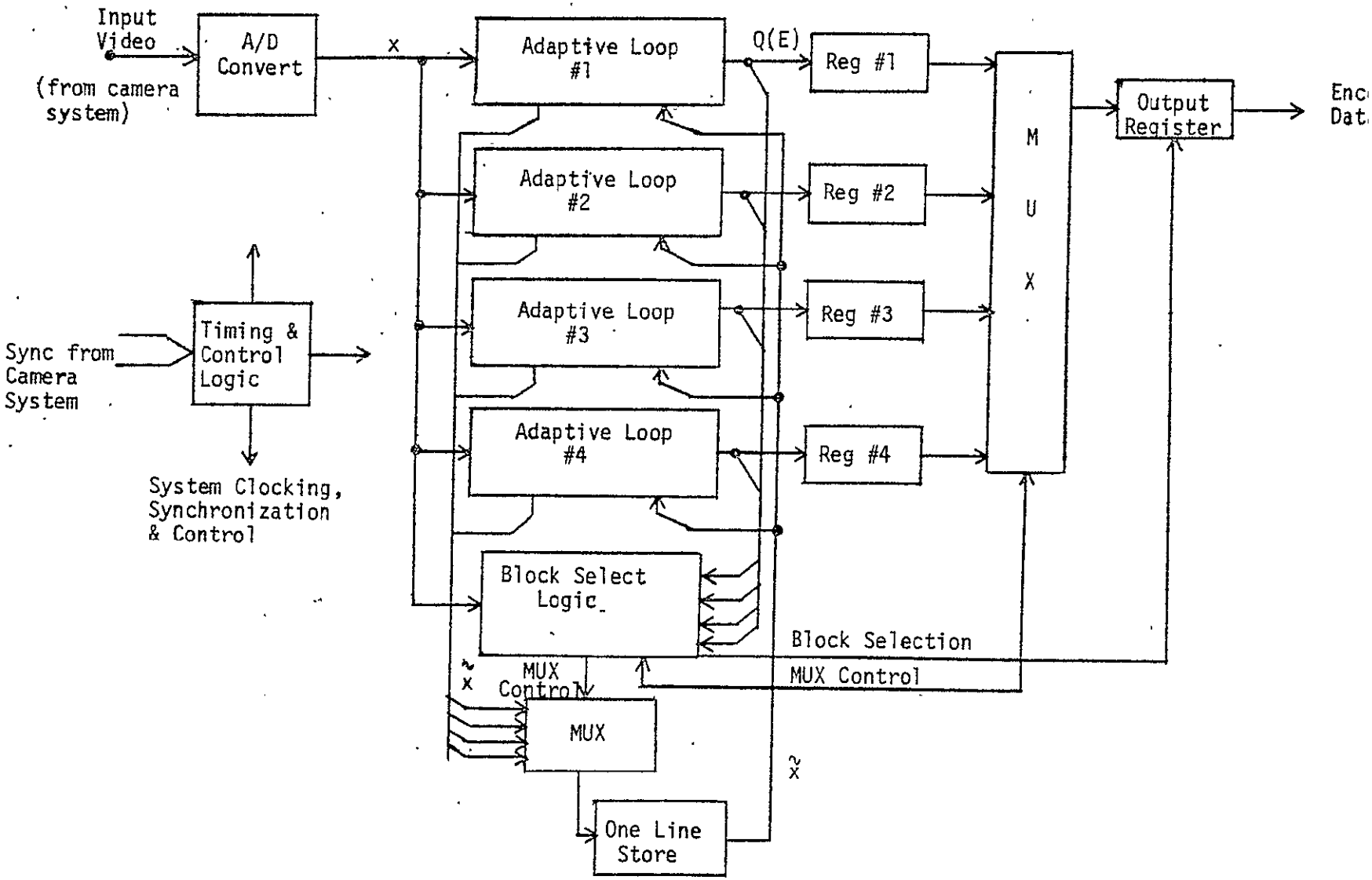


Figure 29. Multiple Loop Adaptive 2D-DPCM Encoder

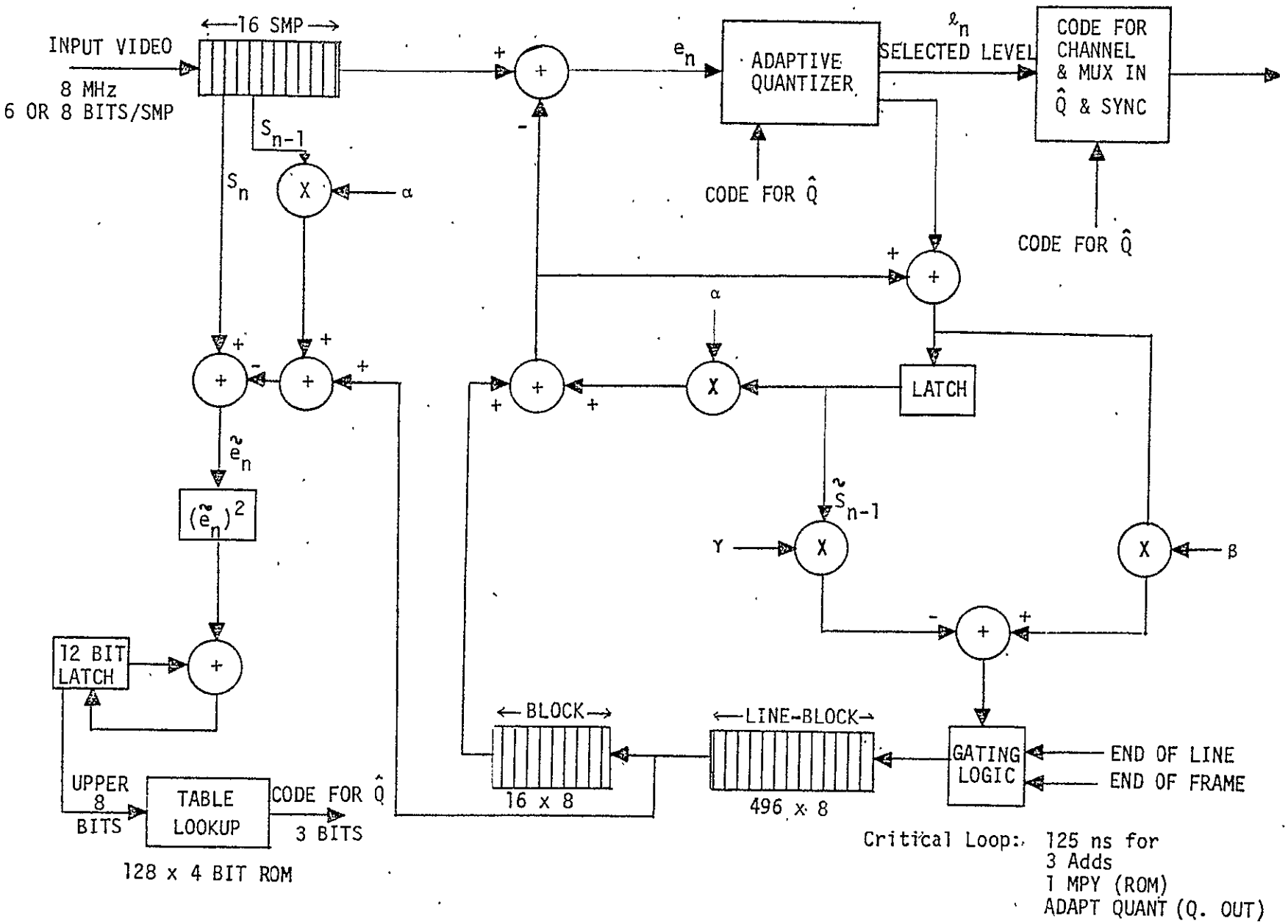


Figure 30. Single Loop Block Adaptive 2D-DPCM Encoder

4.2.2 Two-Dimensional Transforms

Two-dimensional transforms are also possible. The forward and inverse two-dimensional Karhunen-Loeve transformations are given by

$$u_{ij} = \sum_{y=1}^{N} \sum_{x=1}^{N} \phi_{ij}(x,y)\, u(x,y) \tag{4.1}$$

$$u(x,y) = \sum_{i=1}^{N} \sum_{j=1}^{N} \phi_{ij}(x,y)\, u_{ij} \,. \tag{4.2}$$

It has been shown that the "eigenmatrices" of $\phi_{ij}(x,y)$ of the whole picture or subblocks of the picture that are composed of $N \times N$ pixels can be formed from outer products of eigenvectors of the covariance of the data in the horizontal and vertical directions if the covariance of the data is separable and of the form

$$R(x,y,\hat{x},\hat{y}) = R_H(x,\hat{x})\, R_V(y,\hat{y}) \,, \tag{4.3}$$

where R_H and R_V refer to the covariance matrices of the data in the horizontal and vertical directions, and $R(x,\hat{x},y,\hat{y})$ is the covariance "tensor" of the data. In the absence of this assumption, the ordering of the two-dimensional data in a vector form is the only practical solution. Computations indicated by (4.1) and (4.2) correspond to operation on the rows of the image followed by operations on the columns of the horizontally transformed data to obtain the two-dimensional transformation. Approximately N^4 multiplication/addition operations are required to perform the transformation. Often, the two-dimensional Karhunen-Loeve transformation is obtained for the Markov process covariance function

$$R(x,\hat{x},y,\hat{y}) = \exp\left(-\alpha|x-\hat{x}| - \beta|y-\hat{y}|\right) \tag{4.4}$$

where α and β are estimated from the image.

The shortcomings of the method of principal components are the large number of operations required for forward and inverse transformation of (4.1) and (4.2), estimation of the covariance of the data, and calculation of the eigenmatrices. To eliminate these difficulties, a number of other transformations have been considered.

For discrete data, the two-dimensional Fourier transformation corresponds to choosing the basis matrices (images) of the form

$$\phi_{ij}(x,y) = \frac{1}{N} \exp\left[\frac{-2\pi(-1)^{1/2}}{N}(ix+jy)\right] \qquad (4.5)$$

$$i,j,x,y = 1,2,\ldots,N$$

while the two-dimensional Hadamard transform [17,18] corresponds to choosing basis images as

$$\phi_{ij}(x,y) = \frac{1}{N}(-1)^{c} \qquad (4.6)$$

where

$$c = \sum_{h=1}^{\log_2 N-1} [b_h(x)\, b_h(y) + b_h(i)\, b_h(j)] \qquad (4.7)$$

and $b_h(\cdot)$ is the hth bit in the binary representation of $(\cdot)$ and N is a power of 2. Both of the above transformations are members of a class of Kronecker matrix transformations that have $(2N^2 \log_2 N^2)$ degrees of freedom, and can therefore be implemented by $(2N^2 \log_2 N^2)$ computer operations. These transformations remedy the shortcomings of the Karhunen-Loeve transformation by eliminating the necessity of finding an operator matched to the covariance of the image and significantly reducing the computational complexity.

In addition to these transformations, a number of others possessing the above two properties have been considered. For example, Cosine [12] and Slant [13,14] transformations have resulted in a better mean square error performance than either the Fourier or Hadamard transformations. The performance of these transforms is still inferior to the performance of the Karhunen-Loeve transformation, which is the only orthogonal transform that generates a set of uncorrelated signals. However, in most practical applications, the computational simplicity and the ease of implementation of the Hadamard and other suboptimum transformations more than compensates for the suboptimal performance of the transforms.

As an example of using two-dimensional transforms, consider the two-dimensional Walsh-Hadamard transform shown in Figure 31. The video data [S] is an N x N matrix:

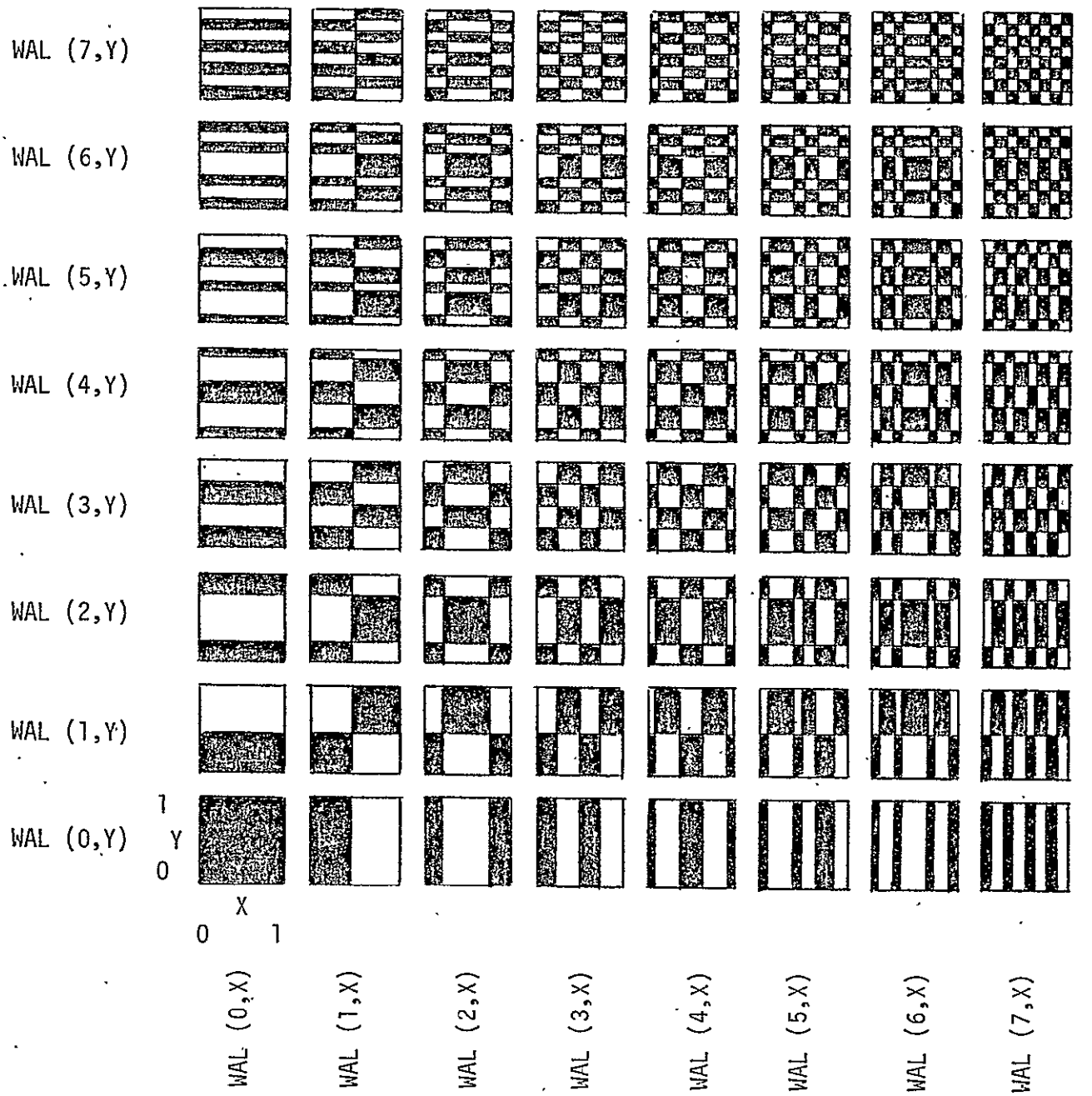


Figure 31. Two-Dimensional Walsh Basis Pictures

$$[S] = \begin{bmatrix} S_{00} & S_{01} & \cdots & S_{0,N-1} \\ S_{10} & S_{11} & \cdots & S_{1,N-1} \\ S_{N-1,0} & \cdots\cdots & & S_{N-1,N-1} \end{bmatrix} \tag{4.8}$$

and the Walsh-Hadamard transformed data is

$$[C] = \begin{bmatrix} C_{00} & C_{01} & \cdots & C_{0,N-1} \\ C_{10} & C_{11} & \cdots & C_{1,N-1} \\ C_{N-1,0} & \cdots\cdots & & C_{N-1,N-1} \end{bmatrix} \tag{4.9}$$

where the elements of $\underline{C}$ are given by

$$C_{k\ell} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} S_{ij}\, WAL(k,i)\, WAL(\ell,j) \tag{4.10}$$

$$k,\ell = 0,1,\ldots,N-1$$

Similarly, by the inverse transform,

$$S_{ij} = \sum_{k=0}^{N-1} \sum_{=0}^{N-1} C_k \; WAL(k,i)\, WAL(\ell,j) \tag{4.11}$$

$$i,j = 0,1,\ldots,N-1$$

or in matrix notation

$$[C] = \frac{1}{N^2} [WAL(k,j)]^T [S] [WAL(k,j)] \tag{4.12}$$

$$[S] = [WAL(k,j)] [C] [WAL(k,j)]^T . \tag{4.13}$$

Figures 32 and 33 present a numerical example of a two-dimensional Walsh-Hadamard transform and its inverse. An alternate approach to the two-dimensional Walsh-Hadamard transform is to use a one-dimensional transform first on the rows of [S] and then on the columns of the resulting matrix [y]. The previous numerical example is repeated in Figure 34. Note that the resulting transform matrix [C] is identical to the matrix [C] in Figure 32 using the two-dimensional transform.

$$[S] = \begin{bmatrix} S_{00} & S_{01} & S_{02} & S_{03} \\ S_{10} & S_{11} & S_{12} & S_{13} \\ S_{20} & S_{21} & S_{22} & S_{23} \\ S_{30} & S_{31} & S_{32} & S_{33} \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

TRANSFORM COEFFICIENTS:

$$[C] = \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 2 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$= \frac{1}{16} \begin{bmatrix} 6 & 6 & 5 & 4 \\ 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$= \frac{1}{16} \begin{bmatrix} 21 & 3 & -1 & 1 \\ 5 & 3 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$[C] = \begin{bmatrix} 21/16 & 3/16 & -1/16 & 1/16 \\ 5/16 & 3/16 & -1/16 & 1/16 \\ 1/16 & -1/16 & -1/16 & 1/16 \\ 1/16 & -1/16 & -1/16 & 1/16 \end{bmatrix}$$

Figure 32. Example of Two-Dimensional Walsh-Hadamard Transform

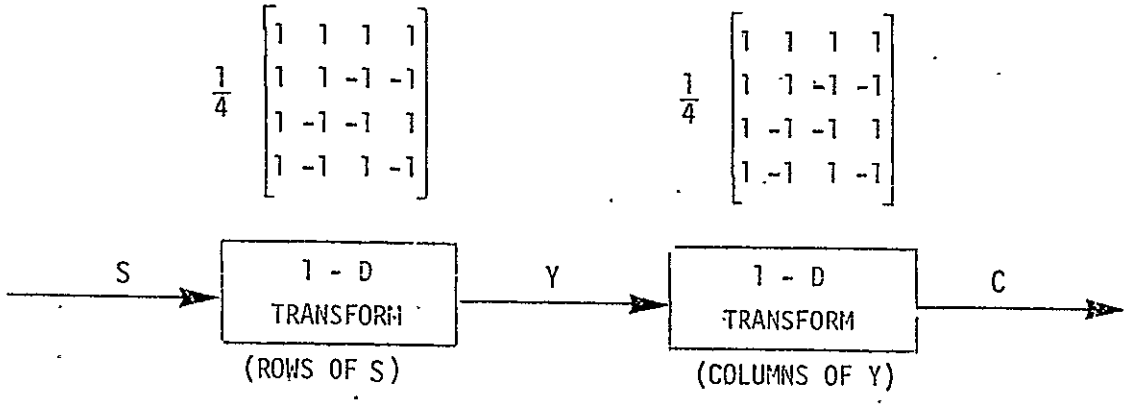
$$[S] = [WAL(k,j)]\,[C]\,[WAL(k,j)]^T$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 21/16 & 3/16 & -1/16 & 1/16 \\ 5/16 & 3/16 & -1/16 & 1/16 \\ 1/16 & -1/16 & -1/16 & 1/16 \\ 1/16 & -1/16 & -1/16 & 1/16 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 28/16 & 4/16 & -4/16 & 4/16 \\ 24/16 & 8/16 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$[S] = \begin{bmatrix} 2 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Figure 33. Example of Inverse Walsh-Hadamard Transform



EXAMPLE:

$$[S] = \begin{bmatrix} 2 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

CALCULATION OF [Y]:

$$\frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 7/4 \\ 1/4 \\ -1/4 \\ 1/4 \end{bmatrix}$$

$$\frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6/4 \\ 2/4 \\ 0 \\ 0 \end{bmatrix}$$

$$\vdots$$

$$[Y] = \begin{bmatrix} 7/4 & 1/4 & -1/4 & 1/4 \\ 6/4 & 2/4 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Figure 34. Alternate Two-Dimensional Walsh-Hadamard Transform Approach

EXAMPLE (continued):

CALCULATION OF [C]:

$$[C] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 7/4 & 1/4 & -1/4 & 1/4 \\ 6/4 & 2/4 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$[C] = \begin{bmatrix} 21/16 & 3/16 & -1/16 & 1/16 \\ 5/16 & 3/16 & -1/16 & 1/16 \\ 1/16 & -1/16 & -1/16 & 1/16 \\ 1/16 & -1/16 & -1/16 & 1/16 \end{bmatrix}$$

Figure 34 (continued)

To illustrate data compression, consider the video data

$$[S] = \begin{bmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} . \tag{4.14}$$

The transformed matrix is

$$[C] = \begin{bmatrix} 20/16 & 4/16 & 0 & 0 \\ 4/16 & 4/16 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} . \tag{4.15}$$

Since only four elements of [C] are nonzero, then only these four elements need to be transmitted. With only four elements received at the decoder, the other elements are assumed zero and the matrix [S] is perfectly reconstructed. However, assume the video data [S] is the same as that used in Figures 32 and 34; then the transformed matrix [C] does not have only four nonzero elements. However, if only the largest four coefficients are transmitted, then at the receiver, the other coefficients are assumed zero. In this case, the reconstructed video data is

$$[S] = \begin{bmatrix} 2 & 2 & 1.25 & 1.25 \\ 2 & 2 & 1.25 & 1.25 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} . \tag{4.16}$$

Note that the difference between the video data [S] in (4.14) differed from the data in Figures 32 and 34 in the upper righthand corner of the matrix [S]. Thus, the assumption of all zeros except for four coefficients causes a smearing of the reconstructed data in the upper righthand corner. Practical systems for high-quality video transmission would probably transmit approximately 10 coefficients.

4.2.3 Hybrid Two-Dimensional Techniques

An attractive hybrid two-dimensional technique is to use a one-dimensional transform on each line of the picture and then operate on each column of the transformed data using a one-element predictor DPCM system. In the hybrid system shown in Figure 35, image data is scanned to form N lines and each line is sampled at the Nyquist rate. This sampled image is then divided into arrays of N by M picture elements $u(x,y)$ where x and y index the rows and the columns in each individual array so that the number of samples in a line of images is an integer multiple of M. The one-dimensional unitary transformation of the data and its inverse are modeled by the set of equations:

$$u_i(y) = \sum_{x=1}^{M} u(x,y)\ \phi_i(x)\ ; \qquad \begin{matrix} i = 1,2,\ldots,M \\ y = 1,2,\ldots,N \end{matrix} \tag{4.17}$$

$$u(x,y) = \sum_{i=1}^{M} u_i(y)\ \phi_i(x)\ , \tag{4.18}$$

where $\phi_i(x)$ denotes a set of M orthonormal basis vectors. Since the correlation of samples in various columns of the transformed array is different, a number of different DPCM systems are used to encode each column of the transformed data.

The performance of hybrid encoders using various transformations is shown in Figure 36 for M = 16 and N = 256 [18]. Performance of the two-dimensional Hadamard and two-dimensional DPCM encoders is included for comparison. This figure clearly shows the superior performance of the hybrid encoder over both the two-dimensional Hadamard and two-dimensional DPCM encoders.

The optimal hybrid coder shown in Figure 35 utilizes a set of different weighting coefficients $A_1, A_2, \ldots, A_M$ in the DPCM predictors at the transmitter and receiver. Also, the quantizers in the DPCM systems are designed based on the statistics of the video data. To simplify the encoder and permit its design to be independent of the signal statistics, a suboptimal system has been developed which uses a common value for A_1 through A_M, and which uses some general statistics (obtained from a number of typical pictures) to obtain variances of the differential signals $w_i(y)$.

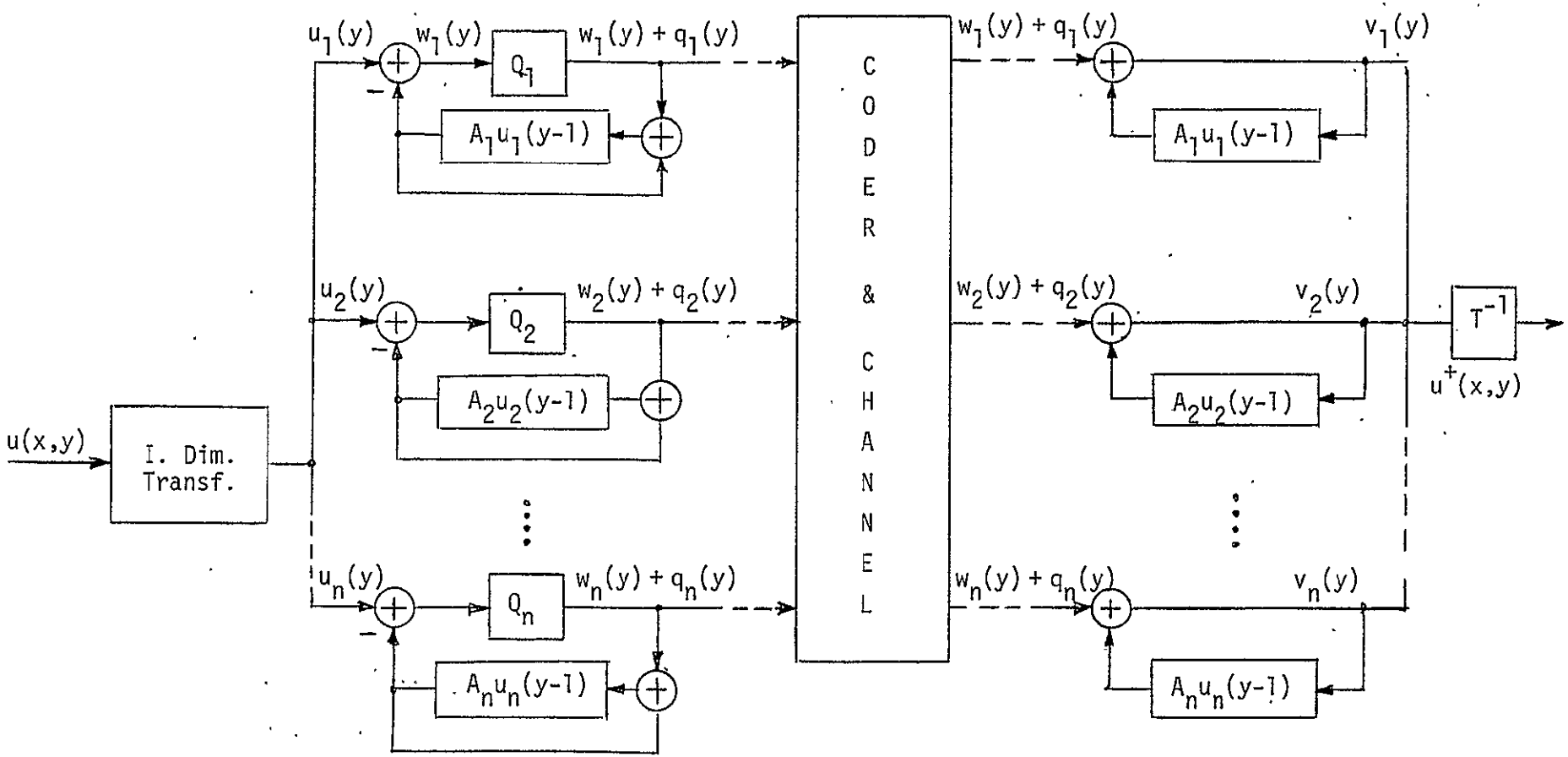


Figure 35. Block Diagram of Hybrid Transform/DPCM Image Coder

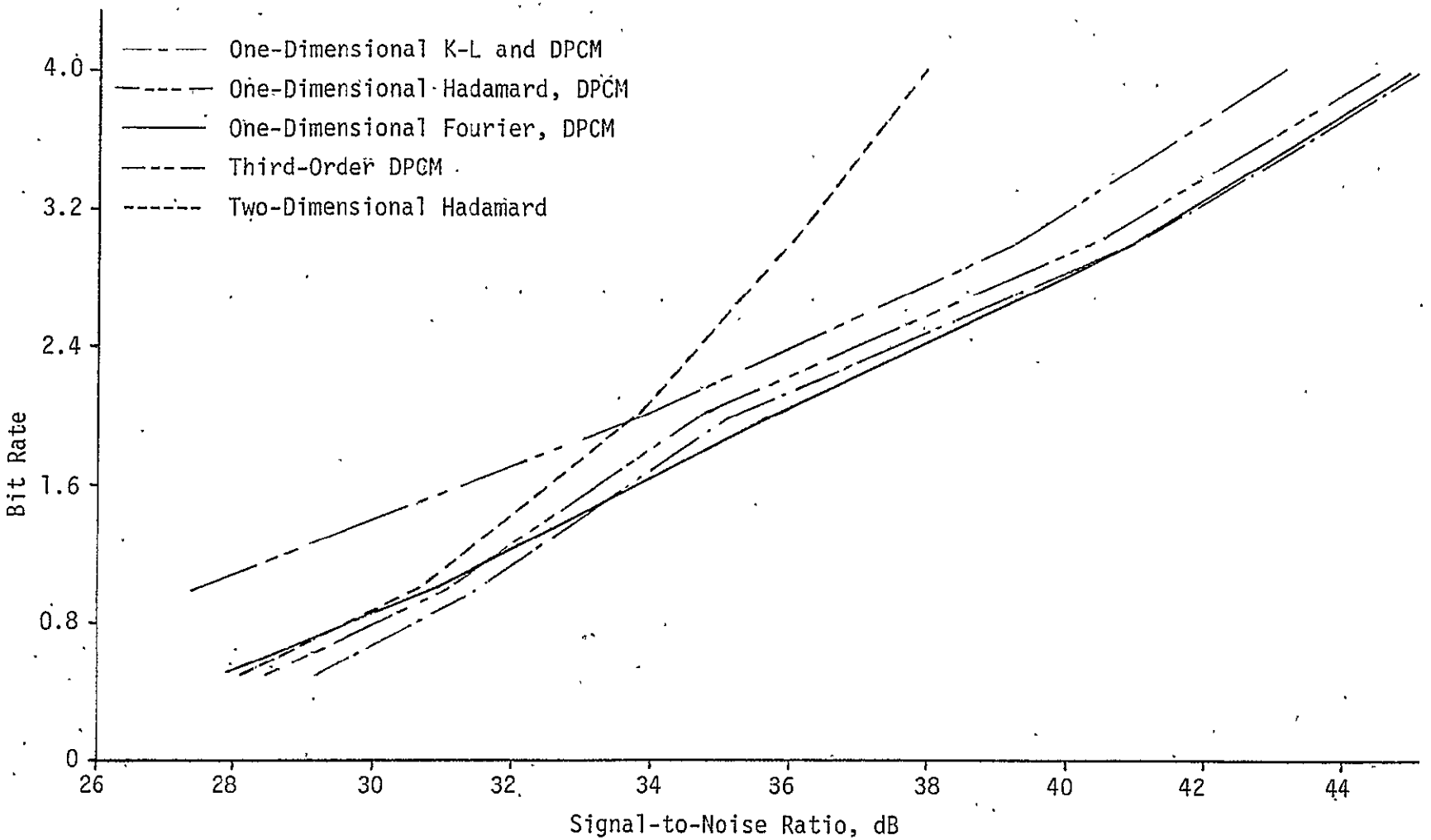


Figure 36. Experimental Results Comparing Performance of Proposed Hybrid Systems. The performance of a third-order simple DPCM and the encoder using two-dimensional Hadamard transform and a block quantizer are included for comparison.

Then only one DPCM encoder can be used to encode all transform coefficients. In the block diagram of the simplified hybrid encoder shown in Figure 37, a single analog-to-digital converter performs the quantization in all DPCM encoders. Differential signals corresponding to various transform coefficients are normalized by dividing them by an appropriate amplitude factor σ_i before being processed by the A/D converter. The system is designed to have a maximum of four bits per coefficient. The bit assignment procedure is programmed in the bit selector which selects 0 to 4 bits per coefficient in a predetermined manner. The nonlinearity of the quantizer is achieved by using a set of fixed nonuniform threshold levels in the A/D converter. The D/A converter uses the corresponding set of nonuniform reconstruction levels.

The simulated results [18] show that performance of the simplified hybrid encoder is only slightly inferior to the performance of the optimum hybrid encoder. The reduction in the signal-to-noise ratio and subjective quality were minimal and well worth the resulting hardware simplification.

4.3 Three-Dimensional Techniques

In a monochrome television signal, a large fraction of picture elements correspond to background material that does not change significantly from one frame to the next, while only a relatively small number of picture elements in a frame convey fresh information. From a statistical standpoint, the similarity of pixels from one frame to the next corresponds to a high level of interframe correlation. Thus, the statistical coding techniques exploiting spatial correlation that have been considered for coding single frames of data could, in principle, be extended to take advantage of the frame-to-frame correlation, thereby further reducing the bit rate required to transmit the data. Indeed, some research in the area of three-dimensional Fourier and Hadamard transformations has indicated that bit rates can be reduced by a factor of about 5 by incorporating the correlation in the temporal direction [19]. However, three-dimensional transform encoding systems suffer from the serious shortcomings of computational complexity and the requirement for large amounts of storage. For this reason, some researchers have avoided extending transform coding systems to a third dimension; instead, they have suggested suboptimum coding systems that do not require extensive amounts of memory or computations.

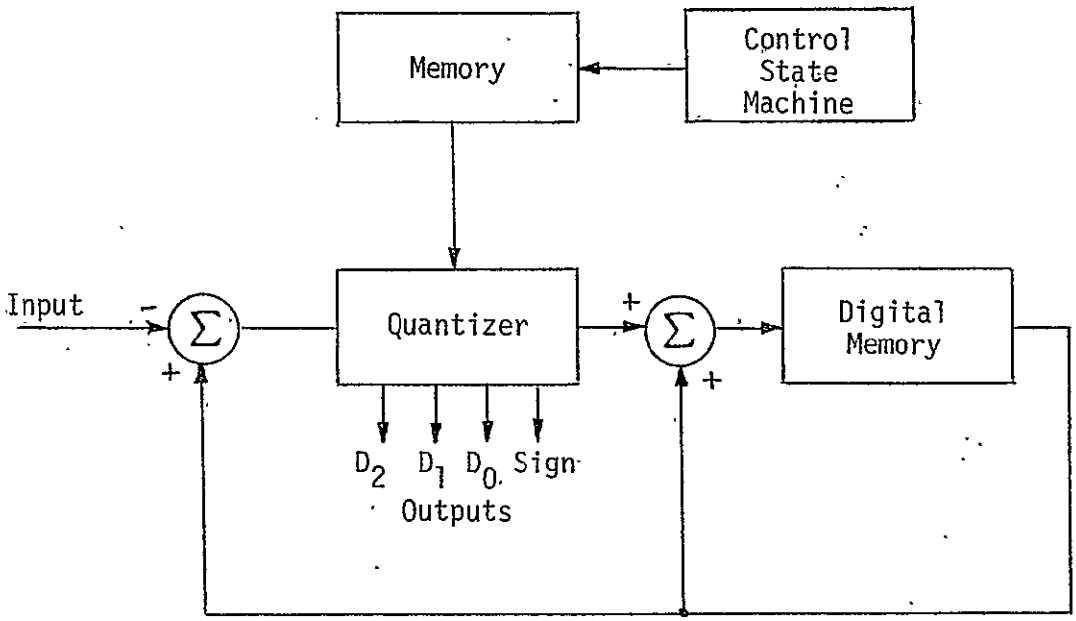


Figure 37. Block Diagram of the DPCM Encoders for the Simplified Hybrid Encoder

An efficient technique of interframe coding of monochrome television images is simply to transmit the grey levels of the elements that have changed in successive frames by replenishing the previous frames with the transmitted data [20,21]. Experiments with Picturephone signals using the conditional frame replenishment technique have indicated good coding results for an average of one bit per pixel. A major shortcoming of the system is that the data is generated at an uneven rate. This is caused by the variation in number of pixels which change beyond a fixed threshold in each frame. To transmit this data over a fixed channel requires buffering the data prior to transmission. The size of the buffer and the bit rate limit the amount of motion in the video data for which this system could be employed. It has been determined that a buffer size of 10 frames is needed to transmit television signals with a moderate degree of motion. However, the buffer size has been reduced to one frame by transmitting only clusters of data. This increases the hardware complexity.

In many cases, it is straightforward to extend two-dimensional techniques to three dimensions. For example, three-dimensional DPCM is illustrated in Figure 38. In this case, the adjacent picture elements in the line, in the column, and in the previous frame are used to predict the current picture element.

Three-dimensional Walsh-Hadamard transforms are also straightforward. Figure 39 presents the three-dimensional Walsh-Hadamard transform. To use the three-dimensional Walsh-Hadamard transforms for data compression, the probability of occurrence of a vector coefficient as a function of its amplitude must be measured. The probability distribution of Walsh-Hadamard vectors shown in Figure 40 illustrate that large vector amplitudes become rare as the vector sequency increases in the horizontal or vertical direction in still pictures. Probabilities of occurrence of large vector amplitudes also fall off with increased temporal sequency, but not as rapidly. This is due to the nature of the source; very fine spatial detail (that would create larger values of high sequency spatial "checkerboard" vectors) is not as common as motion spanning about 20-50 subpictures in four frames (which creates larger values of high sequency temporal vectors).

An alternative approach to the three-dimensional Walsh-Hadamard encoder is a system that would use a two-dimensional Fourier transform for spatial and a DPCM encoder in the temporal direction.

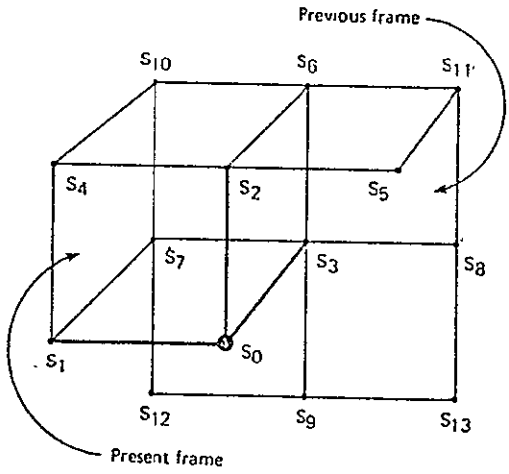


Figure 38. Three-Dimensional DPCM

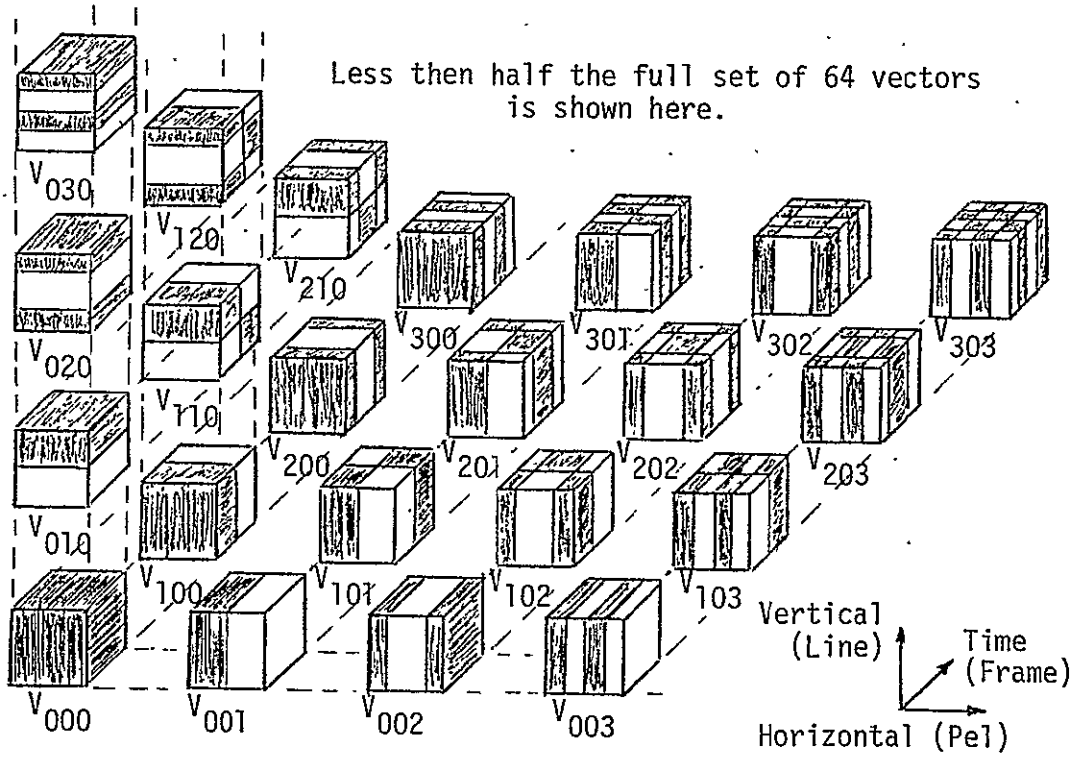


Figure 39. Three-Dimensional Hadamard Basis Vector Representation of a 4 x 4 x 4 Pel Subpicture

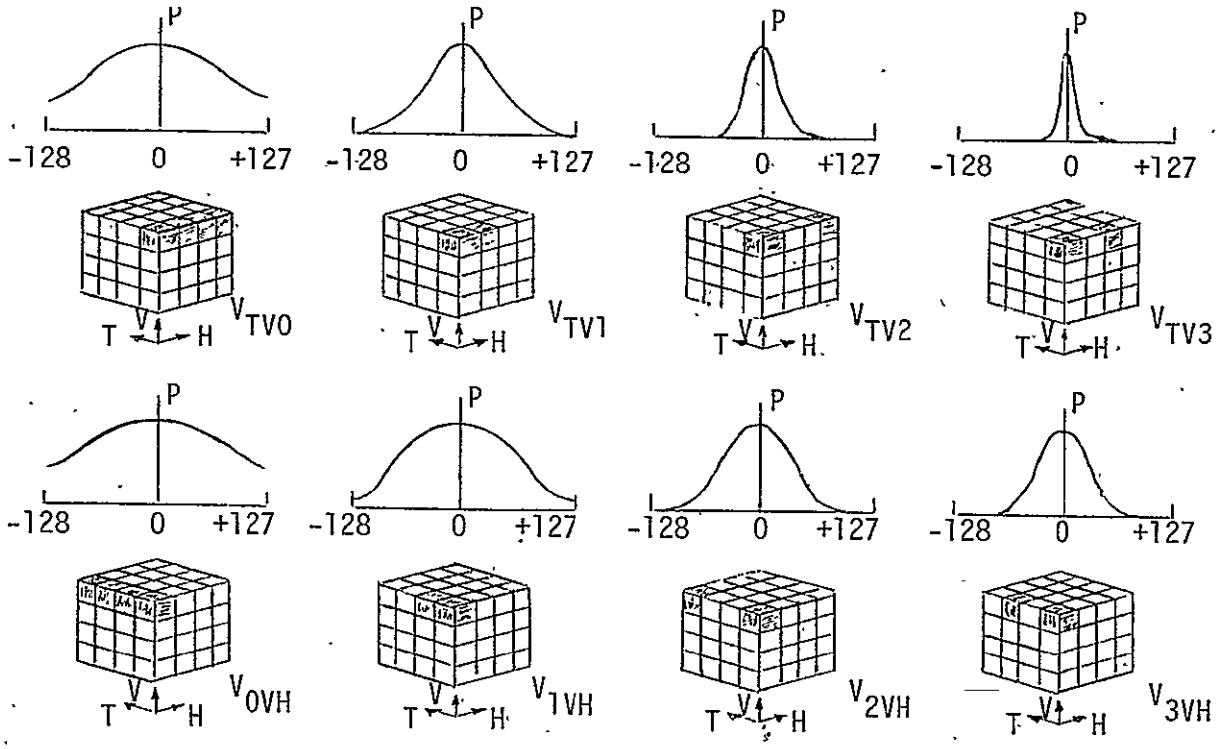


Figure 40. Vector Amplitudes as Functions of Horizontal and Temporal Sequency

This system would generate the two-dimensional Fourier transform of each frame. Referring to the two-dimensional Fourier transform of the kth frame by $f_k(\mu,\nu)$, one can represent $F_k(\mu,\nu)$ by its amplitude and its phase, i.e.,

$$F_k(\mu,\nu) = A_k(\mu,\nu)\, e^{j\theta_k(\mu,\nu)} , \tag{4.19}$$

where $A_k(\mu,\nu)$ and $\theta_k(\mu,\nu)$ refer to the amplitude and phase planes of the kth frame. Many types of motions, such as a panned motion, correspond to significant changes in the phase plane and small changes in the amplitude plane; thus, for an efficient encoder, one would assign a larger fraction of the available binary digits to changes in the phase plane from one frame to the other frame and a smaller number of binary digits to the corresponding changes from one amplitude frame to the other. The other attractive feature of this coding method would be to relate the predictor in the DPCM feedback loop to the motion of the camera relative to the subjects and perform a better prediction for the changes in phase which are caused by the motion. The complexity of this system is significantly less than for the three-dimensional Walsh-Hadamard encoder.

5.0 COLOR TECHNIQUES

5.1 NTSC Color Television System

Color picture reproduction relies mainly on the principle of three primary color decompositions, where the picture is sampled through red, green, and blue filters. Hence, one may treat a color signal as a vector of three separate monochrome signals corresponding to the red, green, and blue contents of the object picture. If the color signal is transmitted in this manner, it would require three times the bandwidth needed for the monochrome transmission. As in the monochrome case, statistical redundancies exist not only as the spatial or time statistical correlations for each color, but also in the form of intra-color redundancy. Data compression can be accomplished in three separate steps. First, the bandwidth can be reduced due to the intra-color redundancy, then the spatial and time statistical data reduction described in section 2.4 can be applied directly. The intra-color statistical redundancy can be described by considering the random vector,

$$\underline{S}(t) = \big(R(t),\ G(t),\ B(t)\big)^T , \tag{5.1}$$

where $R(t)$ = red video signal,
$G(t)$ = green video signal,
$B(t)$ = blue video signal,

as a continuous random process. Bandwidth reduction, in the sense of the least expected mean-square error criterion, can be obtained by applying the Karhunen-Loeve procedure to the set of recognizable color pictures. This normally results in a linear transformation of the original color signal vector,

$$\begin{pmatrix} K_1(t) \\ K_2(t) \\ K_3(t) \end{pmatrix} = M \begin{pmatrix} R(t) \\ G(t) \\ B(t) \end{pmatrix} \tag{5.2}$$

where M is a 3x3 matrix that diagonalizes the covariance matrix

$$E\left[\underline{S}(t)^T \cdot \underline{S}(t)\right] . \tag{5.3}$$

This Karhunen-Loeve procedure depends mainly on empirical analysis. Psychovisual phenomena are not invoked. Other transformations were sought.

One of the methods used in the standard NTSC color TV transmission is to transform the signal vector $\underline{S}(t)$ into the coordinates that consist of the monochrome brightness or luminance and two vectors that lie on the chrominance plane (Figure 41).

$$\begin{pmatrix} Y(t) \\ I(t) \\ Q(t) \end{pmatrix} = M \begin{pmatrix} R(t) \\ G(t) \\ B(t) \end{pmatrix} , \tag{5.4}$$

where

$$M = \begin{pmatrix} .3 & .59 & .11 \\ .6 & -.28 & -.32 \\ .21 & -.52 & .31 \end{pmatrix}, \tag{5.5}$$

$Y(t)$ corresponds to the monochrome brightness, $I(t)$ corresponds to the vector in the "orange red-cyan" direction in the chrominance plane, and $Q(t)$ corresponds to the vector in the "magenta-green" direction. This choice of coordinates has the following advantages:

(1) Black and white monochrome video signal can be readily formed by simply dropping the Q and I components.

(2) Psychovisual phenomenon that the human eyes sense only black and white at very low luminosity.

(3) For small color areas, the human eyes exhibit "tritanopia" (or two-color vision). This corresponds to the decrease of spatial visual sensitivity in the Q component.

Due to these facts, color video signals can be transmitted, with reasonable picture quality, when $Y(t)$ has a bandwidth of 4 MHz, $I(t)$ has a bandwidth of 1.5 MHz, and $Q(t)$ has a bandwidth of only 0.5 MHz. Standard commercial color TV signals are transmitted by modulating the chrominance signals, $I(t)$ and $Q(t)$, by a subcarrier ω_c:

$$\begin{aligned} M(t) &= I(t) \sin \omega_c t + Q(t) \cos \omega_c t \\ &= C_s(t) \sin \omega_c t + C_h(t) , \end{aligned} \tag{5.6}$$

where

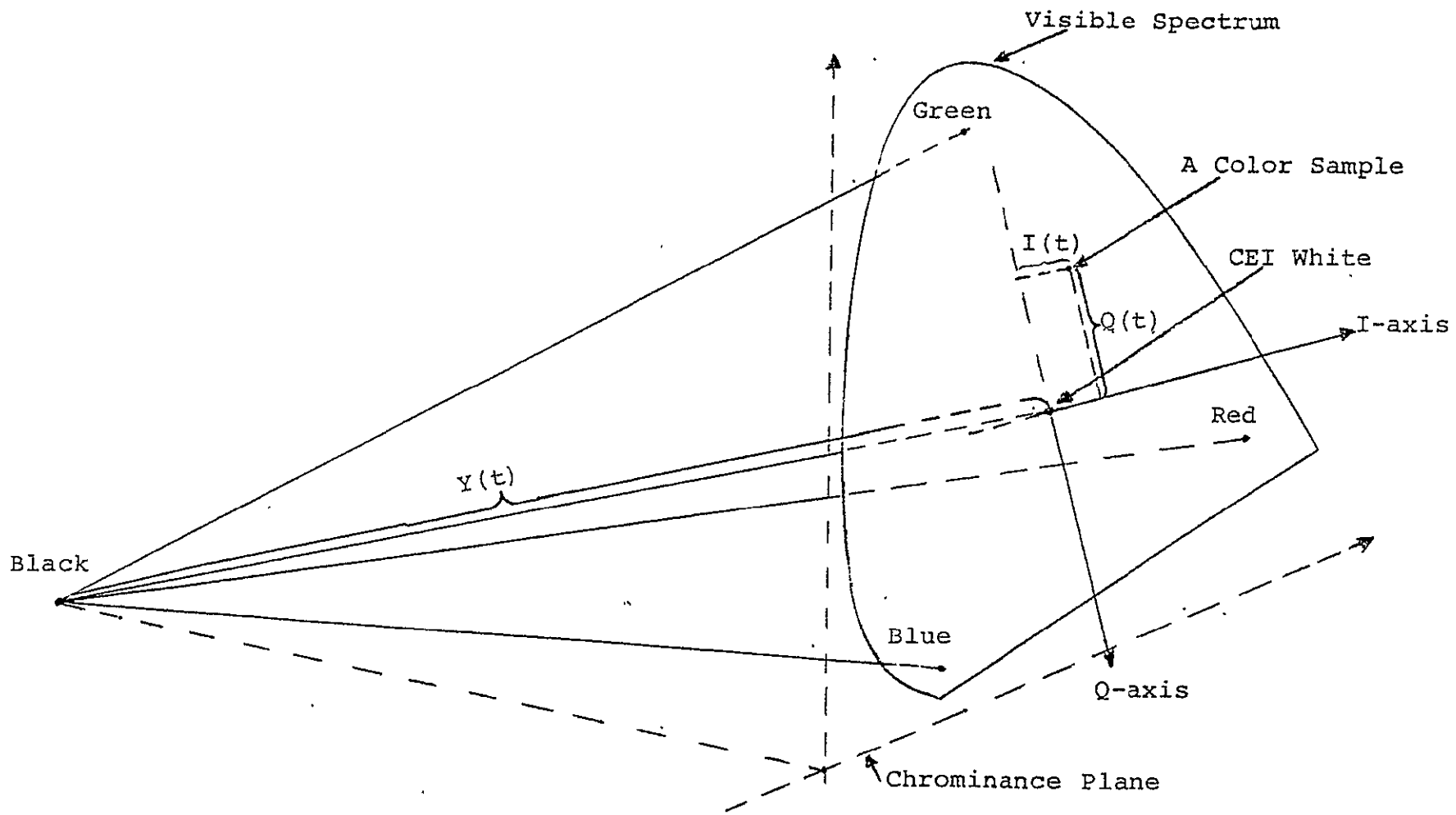


Figure 41. Transformation of Color Sample Into Y, I and Q Coordinates

$$C_s(t) = \sqrt{I^2(t) + Q^2(t)} \tag{5.7}$$

and

$$C_h(t) = \mathrm{Tan}^{-1} \frac{Q(t)}{I(t)} . \tag{5.8}$$

$C_s(t)$ and $C_h(t)$ correspond, very roughly, to the constant hue and constant color saturation stream lines in the plane of chrominance. Thus, a typical commercial color TV signal can be expressed as

$$Y(t) + M(t) = Y(t) + I(t) \sin \omega_c t + Q(t) \cos \omega_c t . \tag{5.9}$$

This composite modulated signal, together with the reference phase of the subcarrier frequency, enables the receiver to demodulate $Y(t)$, $I(t)$, and $Q(t)$. The corresponding primary signals, $R(t)$, $G(t)$, and $B(t)$, are obtained by the inverse linear transformation:

$$\begin{pmatrix} R(t) \\ G(t) \\ B(t) \end{pmatrix} = M^{-1} \begin{pmatrix} Y(t) \\ I(t) \\ Q(t) \end{pmatrix} . \tag{5.10}$$

The general color TV communication scheme is illustrated in Figure 42.

5.2 Field-Sequential Color Television System

Space color television systems use a color camera which is basically a black-and-white camera. It has been converted to a field-sequential color camera by the addition of a rotating color wheel. This technique is very similar to the old CBS field-sequential system developed for color television in the early 1940's. This system was characterized by the use of color band filters at the camera and again at the receiver, with only one camera necessary for viewing a scene and only one picture tube needed at the receiver. This camera, like the old CBS camera, employs a color filter wheel to produce a serial color signal.

The field-sequential system uses a rotating filter wheel to expose the camera's image tube sequentially at the desired broadcast scan rate to the red, blue, and green components of a scene. Thus, the need for complex optical paths and color registration adjustment, such as required in commercial color cameras, is eliminated. This enables the color camera to be lightweight and to require very little power. In addition, it is capable of operating in a large dynamic range of light levels. Since

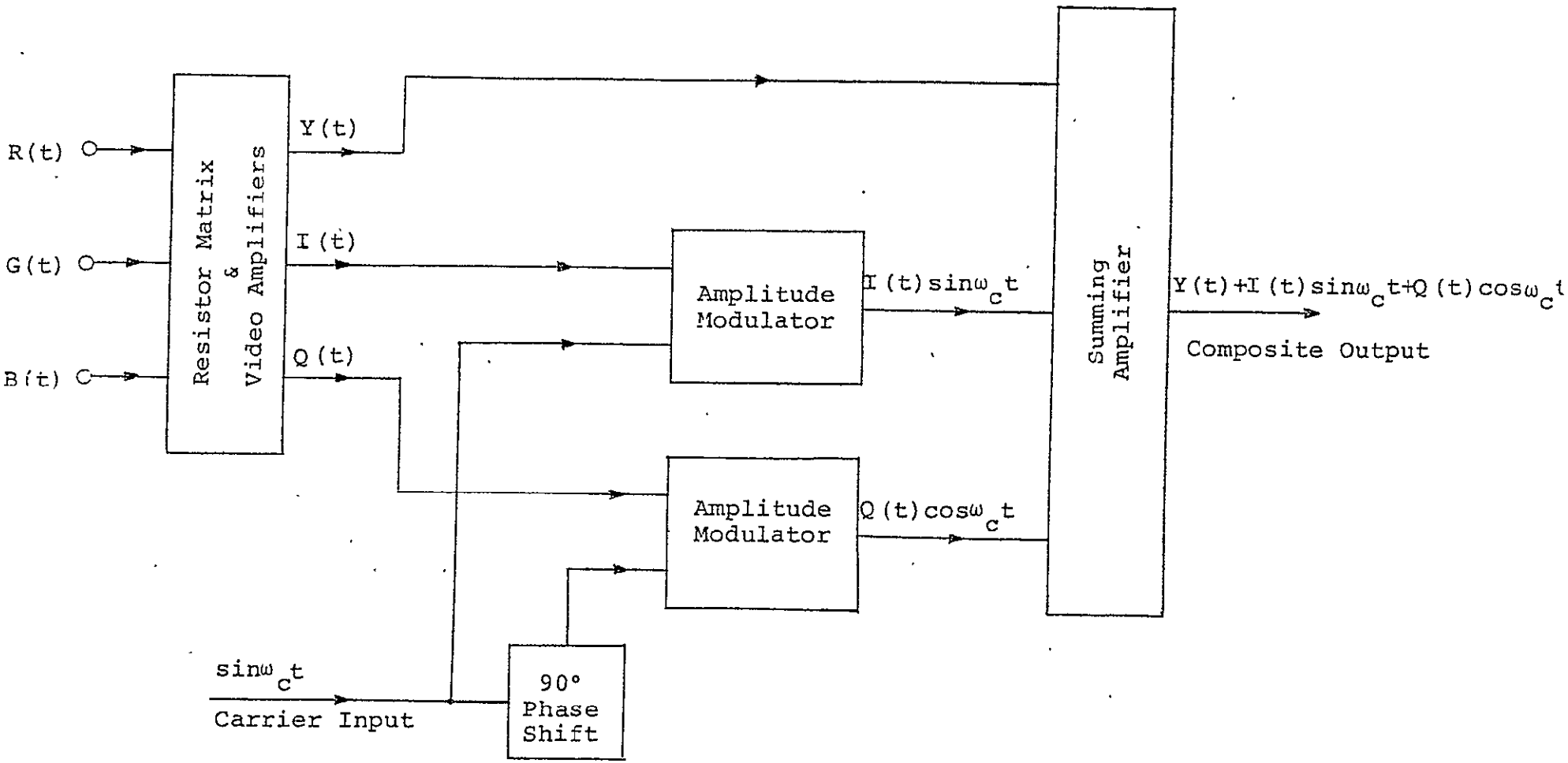


Figure 42. Modulated Composite Color Signal

the output of a field-sequential system is in serial red-blue-green form, it is not compatible with present broadcast standards. This requires that a ground station color converter be utilized to change the sequential color signal to the standard parallel National Television System Committee (NTSC) color TV format so it can be rebroadcast by commercial stations.

A diagram of the color television system [22] is shown in Figure 43. The image is focused by a zoom lens through the color filter wheel onto the faceplate of the image tube. To simplify the problem of synchronization, the scan rate of the wheel as the color filters pass in front of the image tube must be the same as that of the TV networks, which is 60 fields per second. This is achieved by dividing the wheel into six sections, with the colors arranged in red-blue-green, red-blue-green order, and by driving the wheel at 10 revolutions per second. The motor speed is held constant by the timing of the camera's sync generator.

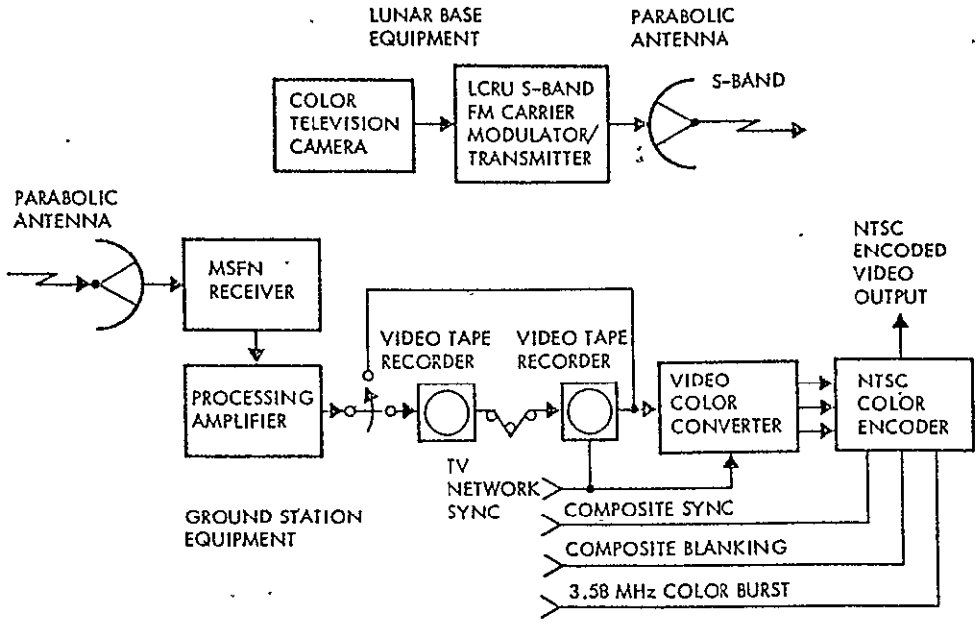


Figure 43. Apollo Color Television System

The field-sequential color signal, which is transmitted by an S-band transmitter (for Apollo), is picked up and amplified by a receiver at the receiving station. The signal is then clamped in a processing amplifier to restore the dc component and reestablish the average light value of the reproduced image.

The processed signal is placed into a series of two tape recorders for the purpose of compensating for doppler shift and presenting real-time information. The sequential color signal is then put into the scan converter that changes the video from the serial color format to the parallel (simultaneous) color format. The scan color converter is a storage and readout device holding the two previous fields in memory and presenting the three fields at once at the output of the incidence of the third field. As the new field is placed into the memory, the oldest field is erased, updating the information at the field rate. Thus, the three colors are simultaneously read out in the same manner as the output from a standard three-tube NTSC color camera. After video color conversion, the signal is sent to an NTSC color encoder which processes it to form the composite video signal.

5.3 Demodulation of Color Signals

Bandwidth compression of television signals exploits the correlation of the signal in both the spatial and temporal directions. To bring out this correlation, the composite color signal must be first demodulated. The field-sequential color signal is already in demodulated form. Spectral correlation of the field-sequential signal can be utilized by simply digitizing two consecutive fields (i.e., corresponding to red and green components) and storing these in digital buffers. The Y, I, and Q components (for each picture element) are then obtained by forming a linear combination of the red and green components with the incoming (blue) component. Demodulation of the NTSC color signal is more complicated. Two popular approaches to this problem are digital color demodulation, using comb filters, and NTSC to time domain multiplexed (TDM) conversion.

5.4 NTSC Digital Color Video Data Compression

Equation (5.9) enables color video information to be packed in a continuous analog signal. Therefore, the monochrome digital video data compression technique can be applied if the mean square error of the

decoded signal is sufficiently small so that the phase and amplitude errors of the reconstructed signal produce tolerable visual sensational errors. This can be done by increasing the sampling rate of the A/D converter and assigning more information bits to the quantization of the Hadamard components. This method has been experimented with by Enomoto and Shibata [13] using 3.75 bits per pel. However, this method is not recommended for spaceborne systems because:

(1) To resolve the phase and amplitude information of the modulated signal to within tolerable visual sensations, a substantially higher sampling frequency must be assigned to the A/D converter. The typical requirement in this case is in the range of 12 MHz to 15 MHz. A/D converters of this type operate in the range of the present state of the art. The corresponding data conversion accuracy, power consumption, weight and cost make them undesirable for limited environments, such as a space vehicle.

(2) The corresponding Hadamard transformer and quantizer must also operate at higher rates. Consequently, power consumption, weight, etc., will increase.

(3) The additional analog modulation-demodulation of the chrominance signals will contribute additional errors to the overall system.

A possible digitized color video data compression algorithm suggested by Linkabit [23] is as follows:

The three primary color video signals [R(t), G(t), B(t)] are first analog transformed into Y(t), I(t), and Q(t), using simple resistor network and video inverting amplifiers. Among these, only the monochrome brightness component, Y(t), conveys most of the picture information and requires highest transmission bandwidth. For video reproduction purposes, 512 samples per horizontal line (approximately 8 MHz sampling frequency) and 8 information bits per sample are sufficient. I(t) and Q(t) have substantially lower bandwidth requirements (this results mainly from the poor spatial response of the human visual system to the chrominance components). Comparatively, either I(t) or Q(t) has bandwidth requirements of less than half that required by Y(t). This enables us to sample I(t) and Q(t) at half the sampling frequency used for Y(t) while maintaining sufficient chrominance resolution. The reduction in horizontal sampling

rate is mainly due to the loss of chrominance response of human eyes to higher spatial frequencies. This reason can be applied equally well vertically. Consequently, I(t) and Q(t) may share the same A/D converter by sampling I(t) and Q(t) at alternate horizontal lines. For ripple-free operation, the sampling frequency for the A/D converter used for Q(t) and I(t) is synchronized to that used for Y(t) by a simple frequency divider. In addition, since the chrominance resolution of human eyes is much less than that of monochrome brightness, a 6-bit resolution seems to be sufficient for the I(t) and Q(t) A/D converter. This corresponds to 31 or more levels of color purity in the directions of white to cyan, white to orange red, white to green, and white to magenta.

Using the above, the additional encoder front-end analog circuit requirements are: a 6-bit A/D converter, a resistor network, and an analog multiplexer for the I(t) and Q(t) signals. For limited environments, as in a space vehicle, this seems to be more preferable than to compress the component modulated signal.

In doing so, the digitized color signal can be represented by a lattice of 512 x 480 sample points for the monochrome brightness component, Y(t), and a lattice of 256 x 240 sample points for the chrominance components, I(t) and Q(t). The monochrome brightness subpictures are chosen with size of 4 x 4 samples, and the chrominance subpictures with 2 x 2 samples. Larger chrominance subpicture sizes are not used because the spatial statistical correlation between sampling points within each subpicture is only a function of its physical dimensions. The bandwidth reduction obtained by lowering the sampling rate results merely from the psychovisual phenomena. Further, the chrominance subpictures are made to coincide with the monochrome brightness subpictures. This is illustrated in Figure 44. The corresponding digitized color subpictures are represented by the following vectors:

Brightness Vector

$$Y = \begin{pmatrix} x_{11} & \cdots & x_{14} \\ \vdots & & \vdots \\ x_{41} & \cdots & x_{44} \end{pmatrix} \qquad (5.11)$$

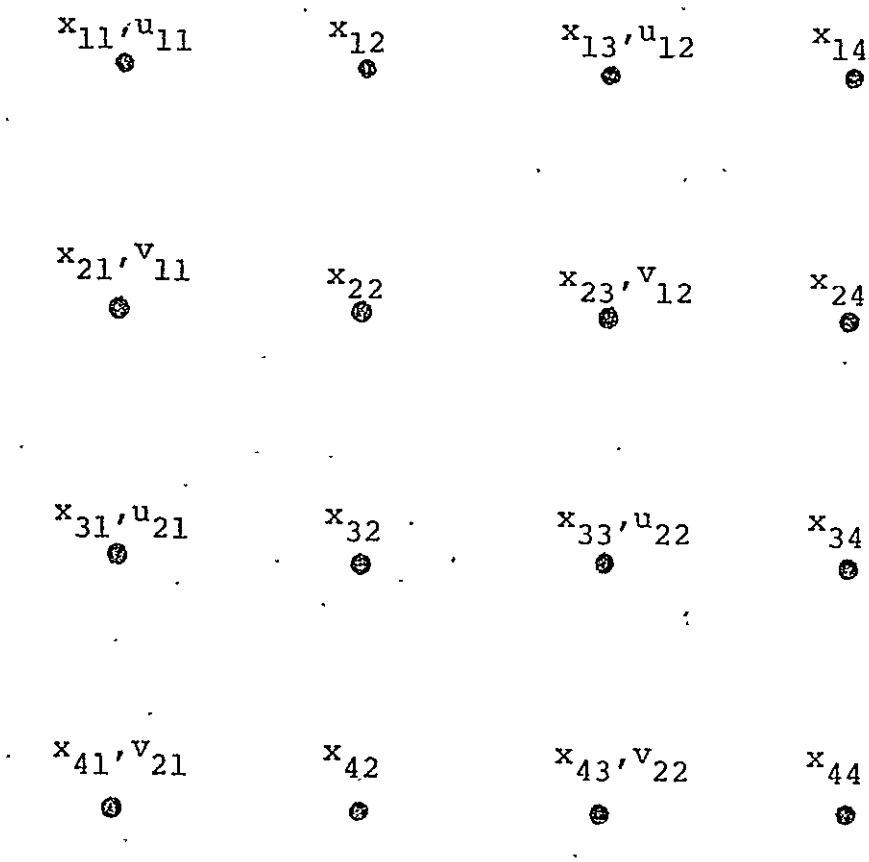


Figure 44. Orientations of Subpicture Sample Points

Cyan-Orange Red Vector

$$I = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \tag{5.12}$$

Green-Magenta Vector

$$Q = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix} \tag{5.13}$$

where x_{ij} is the sampled brightness value at the (i,j) coordinate of the subpicture, and the u_{ij}'s and v_{ij}'s are the sampled chrominance values. This is illustrated in Figure 44. They have integer representations in the following ranges:

$$\begin{aligned} 0 \le x_{ij} \le 255 \\ 0 \le u_{ij}, v_{ij} \le 63 \end{aligned} \tag{5.14}$$

Hadamard transformations can be applied to the above vectors:

$$HY = \begin{pmatrix} C_{11}^{Y} & \cdots & C_{14}^{Y} \\ \vdots & & \vdots \\ C_{41}^{Y} & \cdots & C_{44}^{Y} \end{pmatrix} \tag{5.15}$$

$$H'I = \begin{pmatrix} C_{11}^{I} & C_{12}^{I} \\ C_{21}^{I} & C_{22}^{I} \end{pmatrix} \tag{5.16}$$

$$H'Q = \begin{pmatrix} C_{11}^{Q} & C_{12}^{Q} \\ C_{21}^{Q} & C_{22}^{Q} \end{pmatrix} \tag{5.17}$$

where H is the Hadamard transform for 4 x 4 subpictures, and H' is that for 2 x 2 subpictures. The transformed range is given as follows:

$$\begin{aligned} 0 \le C_{11}^{Y} \le 4 \times 255 \\ -2 \times 255 \le C_{ij}^{Y} \le 2 \times 255 \quad \text{for } i \neq 1 \text{ or } j \neq 1 \end{aligned} \tag{5.18}$$

$$0 \leq C_{11}^{I}, C_{11}^{Q} \leq 2 \times 63$$
$$-63 \leq C_{ij}^{I}, C_{ij}^{Q} \leq 63 \quad \text{for } i \neq 1 \text{ or } j \neq 1 \tag{5.19}$$

The monochrome video data compression procedures described in sections 3.0 and 4.0 can be applied to HY. When two-dimensionally compressed, HY requires 2 bits per pel or, equivalently, 491.52 kbits per frame. Likewise, components for H'I and H'Q can be compressed using similar logarithmic quantization procedures. In particular, due to the low spatial frequency response for the magenta-green chrominance component, C_{12}^{Q}, C_{21}^{Q} and C_{22}^{Q} can be discarded. This follows from the reasoning that using C_{11}^{Q} alone to approximate the Q(t) component, corresponds to sampling Q(t) at one-fourth the sample frequency for Y(t) or, equivalently, sampling at 2 MHz. This is substantially higher than the Nyquist rate required for the 500 kHz bandwidth of the Q(t) signal used in standard commercial color TV signal. Logarithmic DPCM techniques should be applied to the C_{11}^{I} and C_{11}^{Q} components in order to extract maximal advantage due to the residual statistical correlation existing between adjacent subpictures.

The quantization table for chrominance components is given in Table 7. The designation for the Hadamard components is shown in Figure 45.

C_{11}^{I} and C_{21}^{Q} have 23 quantization levels using logarithmic DPCM methods.

C_{12}^{I} and C_{21}^{I} have 15 quantization levels.

C_{22}^{I} has 9 quantization levels.

C_{11}^{I} and C_{11}^{Q} together have 23 x 23 = 529 possible pairs of representative values. To encode them, it would require more than 9 information bits. However, when both I(t) and Q(t) have extreme negative values, the resultant chrominance coordinate lies well outside the reproducible color triangle formed with the three primary color vertices. Hence, by combining the representative values and deleting the cutpoint pairs (C_{11}^{I}, C_{11}^{Q}) outside the reproducible color triangle, as shown in Table 7, C_{11}^{I} and C_{11}^{Q} are effectively encoded by 9 information bits.

C_{12}^{I}, C_{21}^{I}, and C_{22}^{I} share 11 information bits. The overall bit requirement for the chrominance components is 20 bits per subpicture. Thus, the resultant two-dimensionally compressed color data requires

Table 7. Quantization Table

C_{11}^{I} and C_{11}^{Q}: Quantized by 23 Levels

Cutpoint	Representative Value
	±108
+ 96	
	± 84
± 74	
	± 64
± 57	
	± 50
± 44	
	± 38
± 33	
	± 28
± 24	
	± 20
± 17	
	± 14
± 11	
	± 9
± 7	
	± 5
± 3	
	± 2
± 1	
	0

In addition, the following cutpoint pairs of (C_{11}^{I}, C_{11}^{Q}) are deleted; these occur outside the reproducible color triangle: (±96,-96), (±74,-96), (±57,-96), (±44,-96), (±33,-96), (±96,-74), (±74,-74), (±57,-74), (±44,-74). This enables C_{11}^{I} and C_{11}^{Q} to share 9 information bits.

C_{12}^{I} and C_{21}^{I}: Quantized by 15 Levels

Cutpoint	Representative Value
	± 44
± 39	
	± 34
± 29	
	± 25
± 21	
	± 18
± 15	
	± 12
± 10	
	± 8
± 6	
	± 4
± 2	
	0

Table 7 (continued)

C_{22}^{I}: Quantized by 9 Levels

Cutpoint	Representative Value
	± 32
± 26	
	± 20
± 14	
	± 12
± 9	
	± 6
± 3	
	0

C_{12}^{I}, C_{21}^{I}, and C_{22}^{I} have 11 information bits.

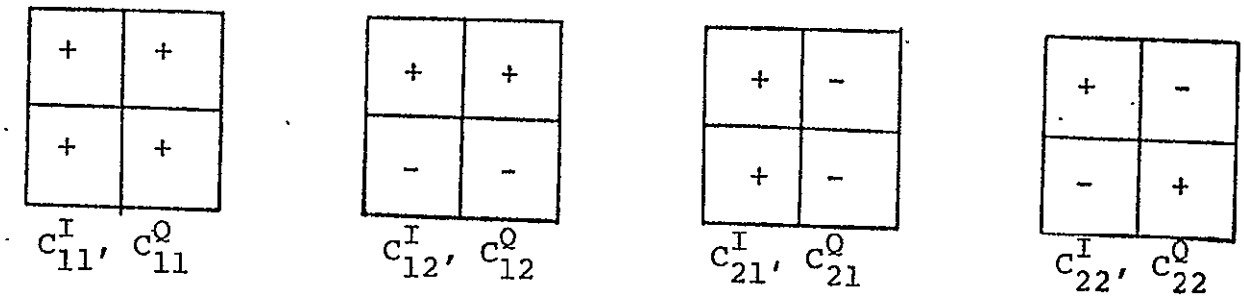


Figure 45. Designations of the Chrominence Hadamard Components

3.25 bits per picture element (based upon 512 x 480 picture elements per frame) or, equivalently, 798.72 kbits per frame. Therefore, by transmitting the color pictures using only two-dimensionally compressed data, the overall bit rate requirement is 23.9616 Mbits per second.

5.5 Field-Sequential Color Video Data Compression

The Field-Sequential Color TV system uses a modified monochrome TV camera with a rotating color wheel. The rotating color filter exposes the camera image tube sequentially, at the commercial broadcast scan rate, to the red, blue and green components of a scene. Therefore, sequential fields differ in spectral components in addition to the field-to-field variations caused by temporal motion. This generates spectral and temporal correlation in addition to the spatial correlation inherent in all pictorial data. However, due to spectral variations, the relative degree of the total spectral and temporal correlation of the Field-Sequential Color TV is less than the temporal correlation between the sequential fields of monochrome television. Therefore, one cannot achieve as high a bandwidth compression with Field-Sequential Color TV as one achieves with monochrome television for the same image fidelity.

The bandwidth of the Field-Sequential signal is significantly smaller than the bandwidth of a standard color TV system. This is because the standard NTSC color television uses three color guns, each with the same bandwidth as the camera used in the field-sequential color TV system. However, the signal generated by NTSC color television systems exhibits more temporal and spectral correlation; therefore, its bandwidth can be compressed by a larger ratio.

The salient feature of the Field-Sequential Color TV signal is that the sequential fields exhibit temporal as well as spectral correlation and exploiting those correlations in addition to spatial correlation is essential to the efficient bandwidth compression of the Field-Sequential Color TV. Spectral correlation is best utilized by using red, green and blue fields to generate the illuminance (Y) and the chromaticity components (I, Q). These are related to the red, green and blue components of the color signal as follows:

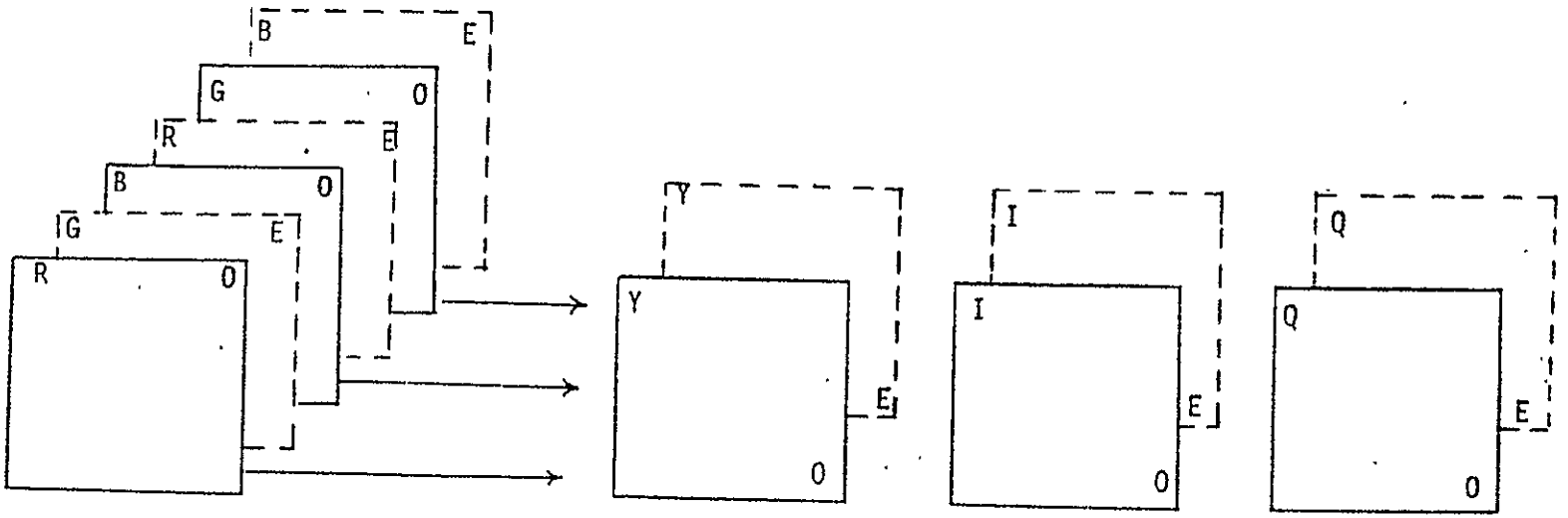
$$\begin{aligned} Y &= 0.30\,R + 0.11\,B + 0.596\,G \\ I &= 0.74\,(R-Y) - 0.27\,(B-Y) \\ Q &= 0.48\,(R-Y) - 0.41\,(B-Y)\ . \end{aligned} \tag{5.20}$$

Sequential fields of the Field-Sequential Color TV signal are composed of the odd and even lines as shown on Figure 46.

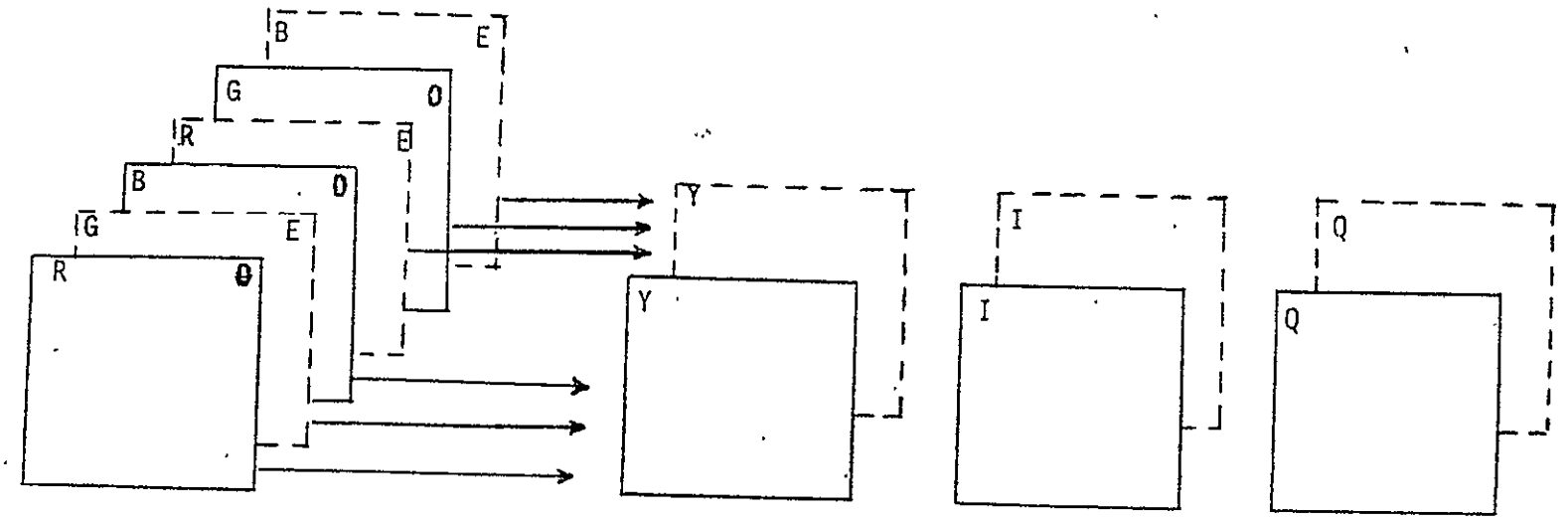
In combining red, green and blue components to generate Y, I, and Q, one utilizes the correlation of the spectral components for a maximum compaction of energy in the illuminance signal. A spectral compaction that results from identical signals for the red, green and blue components produces a maximum value illuminance and a zero grey level for the chromaticity components. On the other hand, the most dissimilar reds, green and blue signals will result in an identical signal for the illuminance and the chromaticity components. In Field-Sequential Color TV, the sequential fields are composed of odd and even lines. The odd red field exhibits spectral similarity with the odd blue and odd green fields. However, these samples are separated by a temporal distance of 4/60 of a second and this causes some spectral decorrelation due to temporal motion. The Y, I, and Q signals formed from all-odd or all-even frames, which requires storing a minimum of 4 frames, are particularly susceptible to spectral decorrelation from rapid temporal motion. An alternate procedure is to mix the odd and even fields in generating the illuminance and the chromaticity components. This requires storing only two fields. The mixing of the odd and even fields results in a smaller correlation among the spectral components but a larger temporal correlation, since the three fields used in generating Y, I, and Q are only 2/60th of a second apart. This gives a larger or smaller compaction of energy in the illuminance signal depending upon the comparative size of spectral similarity and temporal motion.

An approach with attractive implementation properties for data compression uses the standard color wheel and substitutes the green field for the illuminance signal. Then the chromaticity components are obtained by subtracting the green from the red and the blue components. This approach is based on the fact that the green spectral component is very similar to the illuminance component for TV signals. Also, the green component possesses more energy and shows more details than the red and blue components. Using the green component instead of the illuminance, the transmission tristimulus signals are

$$\hat{Y} = G ; \qquad \hat{C}_1 = R - G ; \qquad \hat{C}_2 = B - G . \tag{5.21}$$



(a) Combining Odd and Even Fields Separately to Generate Y, I, and Q Signals.



(b) Combining Odd and Even Fields to Generate Y, I, and Q Signals

Figure 46. Statistics of the Field-Sequential G, R-G, and B-G Fields

$\hat{C}_1$ and $\hat{C}_2$ possess a much smaller bandwidth and a smaller fraction of the signal energy than the G component; therefore, they can be transmitted in a subsampled form utilizing a smaller fraction of the available bit rate.

For data compression, the illuminance signal is used directly, while the chromaticity components are subsampled. The chromaticity signals can be subsampled by taking every other sample without affecting the quality of the reconstructed signal since human vision is rather insensitive to color information at high frequencies. Transmitting the chromaticity components in subsampled form means that the reconstructed composite signal at the receiver has illuminance as well as the chrominance signals at low frequencies, but contains only the illuminance information at high frequencies. Subjective experiments with television viewers at normal viewing distances have indicated that this is, in fact, acceptable [24].

Prior to subsampling the chromaticity signals, they must be filtered to eliminate their high frequency components to prevent aliasing. Although there exist sophisticated filtering techniques to eliminate the high frequency components of discrete signals, experiments with imagery data has shown that a simple 3-point "hanning" filter can be used with comparable results. A 3-point hanning filter uses weightings of 1/4, 1/2 and 1/4 to obtain the filtered signal as follows:

$$\hat{X}_{ij} = \frac{1}{4} X_{i,j-1} + \frac{1}{2} X_{ij} + \frac{1}{4} X_{i,j+1} , \qquad (5.22)$$

where $\hat{X}_{ij}$ is the lowpass filtered form of X_{ij}. This filter is particularly attractive for digital signals since the multiplications can be performed by shift operations. In the proposed system, the chromaticity signals are filtered by the 3-point hanning filter prior to a 2-to-1 subsampling of these signals.

A detailed study by TRW [18] of bandwidth compression algorithms that process the G, R-G and B-G have led to three candidate techniques. These are two-dimensional DPCM, adaptive two-dimensional DPCM, and the Hybrid system combining a Hadamard Transform with a DPCM encoder.

5.5.1 Two-Dimensional DPCM System

Two-dimensional DPCM systems were discussed in Section 4.0. Using a third-order fixed predictor, the picture element X_{ij} is predicted using a linear combination of the adjacent element on the same line, the adjacent element on the same column, and the element diagonally across from X_{ij} as follows:

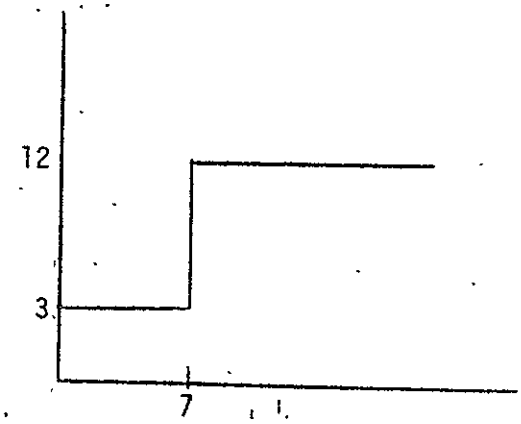
$$\hat{X}_{ij} = 0.75\, X_{i-1,j} + 0.75\, X_{i,j-1} - 0.5\, X_{i-1,\, j-1} \quad (5.23)$$

The fixed values of 0.75, 0.75 and -0.5 have been used for the weightings of the predictor since these weights provided the best overall results in the simulation studies. In addition, digital multiplication by these numbers can be performed by simple shift-and-add operations. The DPCM encoder uses two quantizers. One consists of 8 quantization levels and is used to encode the green signal component. The other consists of 4 quantization levels and is used to encode the chromaticity components. The cutpoints and output levels in the quantizer are selected for the best performance as measured by mean square error and the subjective quality of the reconstructed imagery. The two quantizer characteristics, which are symmetrical, are shown in Figure 47. For convenience, only the positive portion of the quantizer characteristics is shown in the figure.

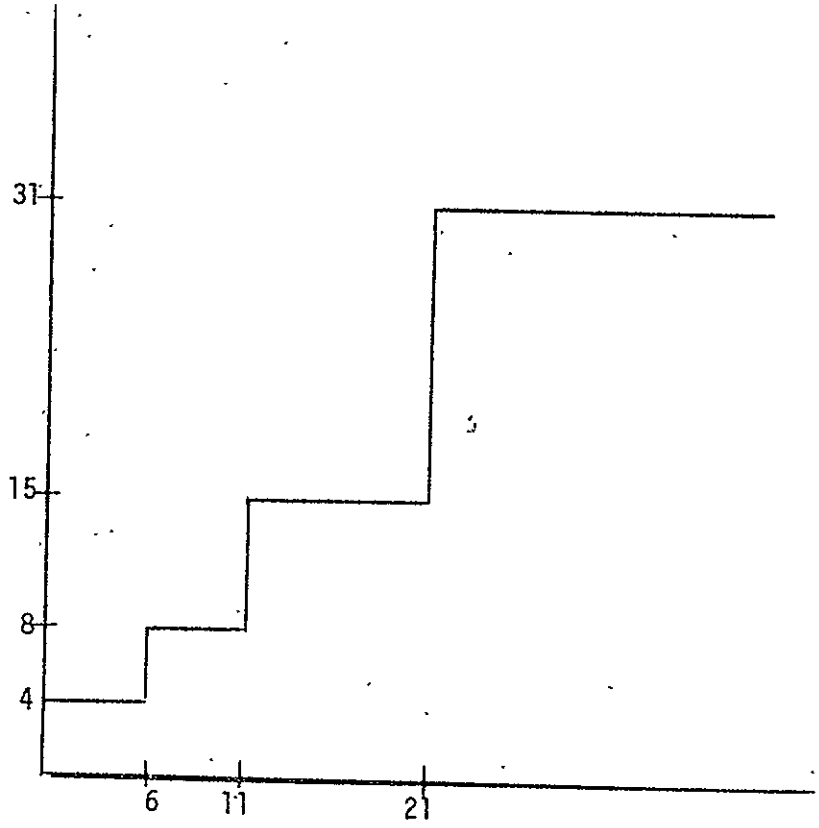
5.5.2 Adaptive Two-Dimensional DPCM Systems

Two adaptive two-dimensional DPCM systems are very promising. These are the block-adaptive DPCM encoder using multiple prediction loops and the block-adaptive DPCM system that uses a single prediction loop with a quantizer characteristic controlled by an auxiliary gain computation loop.

1. Block-Adaptive DPCM Encoder Using Multiple Loops. The block diagram of the encoder is shown in Figure 48. Each DPCM loop uses a third-order predictor defined by equation (5.23). The quantizer in each loop has a different characteristic. The four quantizers are scaled versions of those shown in Figure 47. The scaling constants are 1/2, 1, 2, and 4. Each DPCM loop stores a block of 16 encoded samples in the shift registers and computes the total encoding error. The block select logic compares the distortion (total encoding error) for each loop and transmits the contents of that shift register which corresponds to the smallest



(a) 2-Bit Quantizer



(b) 3-Bit Quantizer

Figure 47. Positive Cutpoints and the Reconstruction Levels of the Quantizers for the 2D-DPCM System

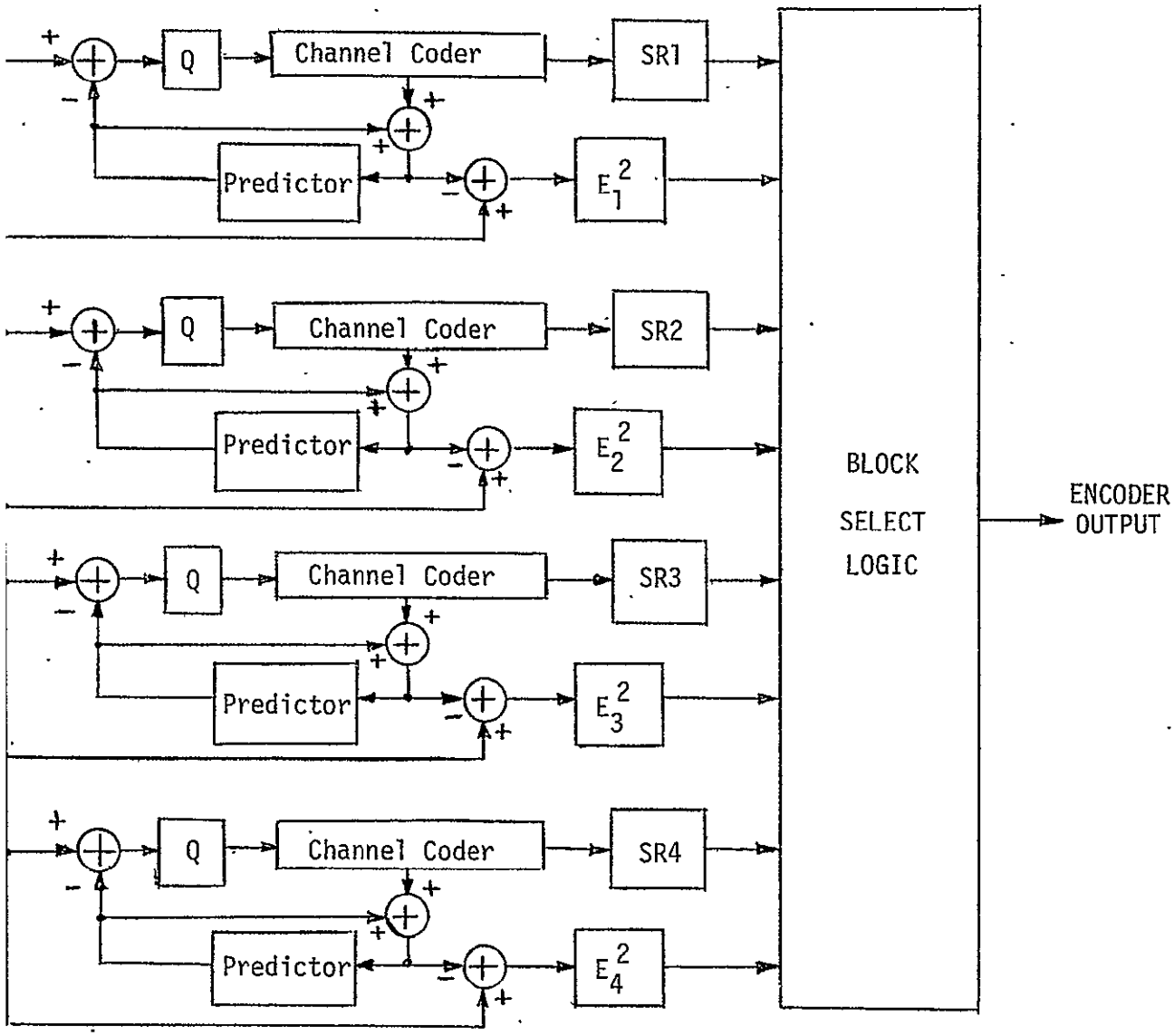


Figure 48. Block-Adaptive 2D-DPCM System Using Multiple Loops

distortion. The receiver needs information as to which DPCM loop was utilized for each block of data; therefore, two bits of overhead information are transmitted with each block indicating which loop (quantizer) was selected. This increases the bit rate by 1/8th of a bit per sample.

2. Block-Adaptive DPCM Encoder Using a Single Loop. In principal, this technique is similar to the adaptive DPCM system with multiple loops. Here, a block of 16 samples is used in the gain computation loop (without a quantizer) to generate an estimate for the variance of the differential signal as shown in Figure 49. Depending upon the value of the variance, one of M gain factors is selected and used to scale the quantizer characteristic in the prediction loop. Eight possible gain factors (1/8, 1/4, 1/2, 1, 2, 4, 8, 16) are used in this system, requiring 3/16th of a bit per sample for transmitting the overhead information.

5.5.3 Hadamard Transform/DPCM System

Hybrid encoders use a concatenation of a unitary transform (i.e., a Hadamard transform) and a DPCM encoder (Figure 50). A block size of four picture elements is used for simple implementation. Each Hadamard coefficient (H_0, H_1, H_2, H_3) is encoded with a DPCM loop using a one-element predictor with a fixed weighting coefficient. The bit assignment and weight coefficient for DPCM loop encoding the green signal and the chromaticity components are listed in Table 8. The quantizer in the DPCM loops is similar to those on Figure 47. However, the scaling of the threshold and reconstruction values for each loop are different. Scaling values of 2, 1, 1/4 and 1/4 are used for the DPCM loops encoding H_0, H_1, H_2, and H_3, respectively.

Table 8. Bit Assignment and Weighting Coefficients per DPCM System in Hybrid Encoder

Type of Signal	Parameters	DPCM for H_0 Coefficient	DPCM for H_1 Coefficient	DPCM for H_2 Coefficient	DPCM for H_3 Coefficient
Green Field	Bits/Sample	4	3	3	2
	Weighting Coefficients	7/8	3/4	3/4	1/2
Chroma-ticity	Bits/Sample	3	2	1	0
	Weighting Coefficients	3/4	3/4	1/2	1/2

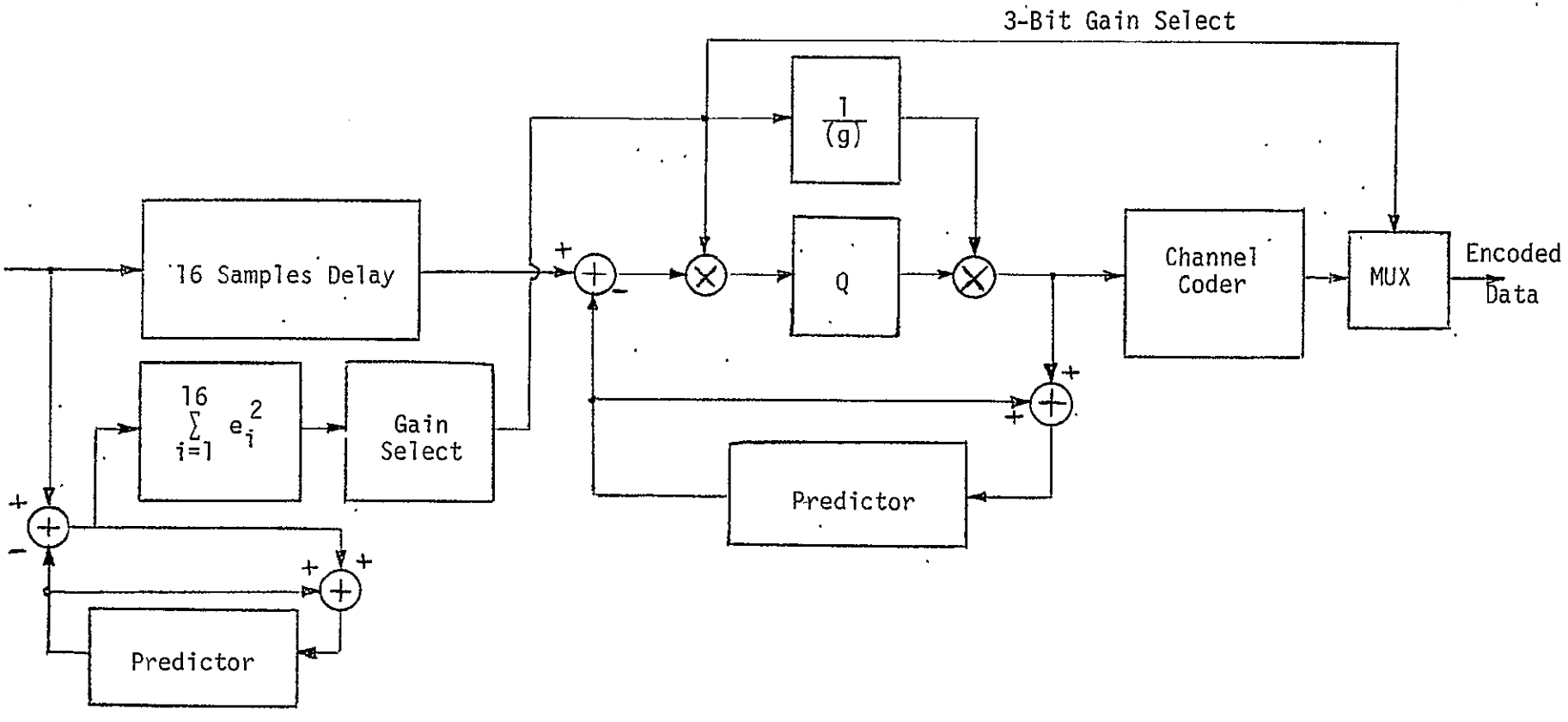


Figure 49. Adaptive DPCM Encoder Using a Single Loop

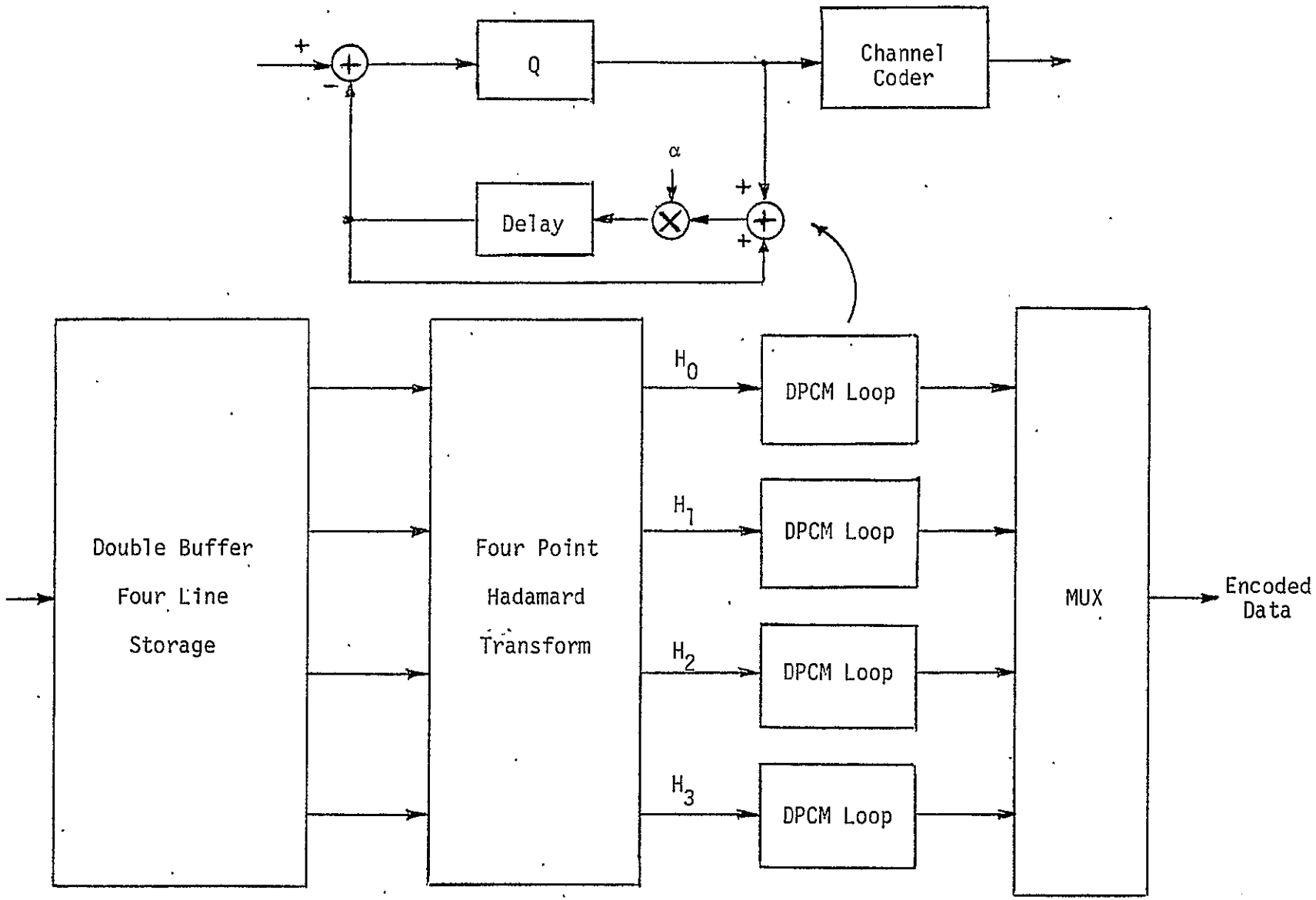


Figure 50. Hybrid Hadamard/DPCM Encoder

6.0 TRI-STATE DELTA MODULATOR (TSDM)

6.1 Introduction and Overview

A technique that was considered of major importance for TV data compression is delta modulation and, in particular, tri-state delta modulation.

A conventional delta modulator (CDM) encoder extracts only the direction of changes in the estimate of the input signal for transmission through the channel. Because of this, such a modulator is simply a two-state, one-bit-per-sample data compression scheme that obtains a better bandwidth utilization efficiency than pulse code modulation (PCM). A significant effort has therefore been made over the last decade to perfect the delta modulator (DM) channels for video and voice transmission. The various solutions suggested are all confronted with the difficult task of finding a good technical compromise between the conflicting requirements to achieve a high transmission fidelity with the minimal possible hardware complexity.

When trying to investigate the performance of a DM in picture transmission, the typical video signals obtained from scanning a picture in the conventional manner should be considered. These video signals are characterized by frequent voltage discontinuities of large amplitudes and very short "rise times" that correspond to abrupt changes in the gray levels of the picture (at its "contours"). In what follows, the delta modulator is considered to be in the "acquisition" mode when trying to catch up with these fast transitions. It shall be considered to be in the "tracking" mode during the other periods in which it "tracks" a constant voltage level (that corresponds to a constant shade, or "gray level," that follows a transition). It should be stressed that the requirements imposed on the channel design by the two mentioned modes are of a conflicting nature. A fast acquisition capability will reduce the "slope-overload" of the channel but will cause a high "granularity" during the tracking periods.

It seems that most of the adaptive delta modulators (ADM) described in the literature assign a higher preference to a high acquisition speed rather than to smooth tracking. A fast acquisition behavior usually leads, however, to instabilities in the signal reconstruction mechanism. The

estimate of the signal then "overshoots" when the channel switches from the acquisition mode to the tracking mode, producing a "multiple-edge" effect in the received picture. It sometimes also overshoots during the acquisition time itself, leading to unwanted "edge busyness" effects.

Attempts to alleviate these instabilities and to ensure that they do converge fast enough to the new level have recently been proposed. Some methods rely on stability bounds imposed on the channel parameters that control the acquisition speed. Others suggest overshoot-suppression (OSS) algorithms that try to smooth the transitions between the two operating modes.

The tri-state delta modulator (TSDM) achieves improved performances over the conventional one (CDM) by breaking the inherent interdependence between the acquisition and the tracking modes of operation. This independence between the two abovementioned modes is obtained by adding a third state (the "level" state) to the usual "rise" and "fall" states of the CDM.

Thus, a conventional delta modulator is clearly a "zero-error seeking feedback loop" type of device. It produces an estimate, X_k, for each (sampled) value of the signal, S_k, applied to its input. The difference between them (the "error") is then used by the device for updating the next estimate. It is therefore obvious that such a device will operate properly only with nonzero errors and this is the basic cause for the "granularity" effect inherent in any delta modulator when tracking constant voltages. This behavior of the CDM in tracking "levels" can also be linked to the differential nature of the device which enables it only to detect (and transmit) changes in the signal voltage.

In comparison, the tri-state delta modulator can detect and handle "no-change" (level) situations in addition to the "up" and "down" changes. It therefore incorporates the "zero-error" case as a third and valid state in its operation. This third state eliminates the very source of the granularity effect mentioned above. Therefore, the TSDM device is characterized by the following three mutually exclusive states:

1. The state in which the voltage level of the signal estimate is continuously rising (the "rise" state).

2. The state in which the voltage level of the estimate falls continuously (the "fall" state).

3. The state in which the estimate equals the signal (within a given voltage tolerance, ε_v); this "level" state is what makes the TSDM device different from its predecessors.

Specifically, a sensing device which continuously splits the video signal or its estimate (as the case may be) into the three activity type components forms the basis of the three-level delta modulator. Thus, the TSDM generates sequences (runs) of +1's during the entire period of time when the estimate of the signal is rising. It generates strings of -1's when the signal is falling, and 0's during level situations. A level situation is declared, at time sample k+1, whenever $|S_k - S_{k+1}| \leq \varepsilon_v$ for $i = 1,2,\ldots$, and the length i of the run of zeroes so obtained will stop increasing when this inequality first fails.

Advantages of the Tri-State Delta Modulator

From the above description, the main inherent advantages of a delta modulator based on such an "activity" sorting device could be summarized as follows:

1. Ease in the run-length encoding of the signal. This could facilitate the implementation of simple bandwidth preserving schemes in the pertinent communication links, as the large portions of any picture merely contain background information that easily qualify as "levels" over long periods of time. These levels can easily be extracted from the video signals and properly coded.

2. No granularity in the received signal. Another important advantage of the TSDM has to do with the mechanism used to reconstruct the video signals at the receiver end of the channel. Here, the splitting into three activity elements, rather than two, facilitates the tracking of the "levels" without the "granularity" associated with the two-state conventional delta modulators.

3. Ease of hardware implementation. Implementing the TSDM appears simpler than would have been expected. The reason for this is that the "level" zone defined by $\pm\varepsilon_v$ can be easily generated within such electronic sensing devices as voltage comparators.

4. Ease of applying smoothing techniques to noisy signals. The voltage zone of $\pm\varepsilon_v$ also provides a powerful, yet extremely simple-to-implement (and to preset) tool for smoothing the video signals and for "clearing" them of excessive noise.

5. More efficient OSS algorithms. Since the overshoots in the voltage level of the estimate X_k pass through one of the three states when entering the tracking mode, their recovery times can be significantly reduced. The increased number of degrees of freedom therefore permits more efficient utilization of possible OSS algorithms.

6.2 Detailed TSDM Operation Description

6.2.1 Functional Block Diagram Description

The functional block diagram of the TSDM is shown in Figure 51. Note that, with the exception of the dead zone in the hard limiter characteristic, this block diagram is similar to the one of a conventional delta modulator (CDM). It is the dead zone in the limiter characteristic which provides this modulator with the tri-state extraction capability. The output of the modulator is then a state indicator, B_k, which provides the information about the type of "activity" characterizing the current signal estimate.

The B_k state indicator lends itself easily to run-length encoding schemes. The information can be thus encoded into either one of the following outputs:

(1) A stream of two-bit serial words; or

(2) A stream of three-bit parallel words.

If the second approach is taken, the modified block diagram of Figure 52 applies. The notations shown in the second block diagram can be used conveniently for describing the encoding of the B_k vector into a two-bit serial word which is entered into the estimator. The translation used is given in Table 9.

Table 9. Representation of the State Information
[Note that $B_k = (r_k - f_k)$]

ℓ_k	f_k	r_k	B_k	b_1	b_0
1	0	0	0	0	0
0	1	0	-1	1	1
0	0	1	1	0	1

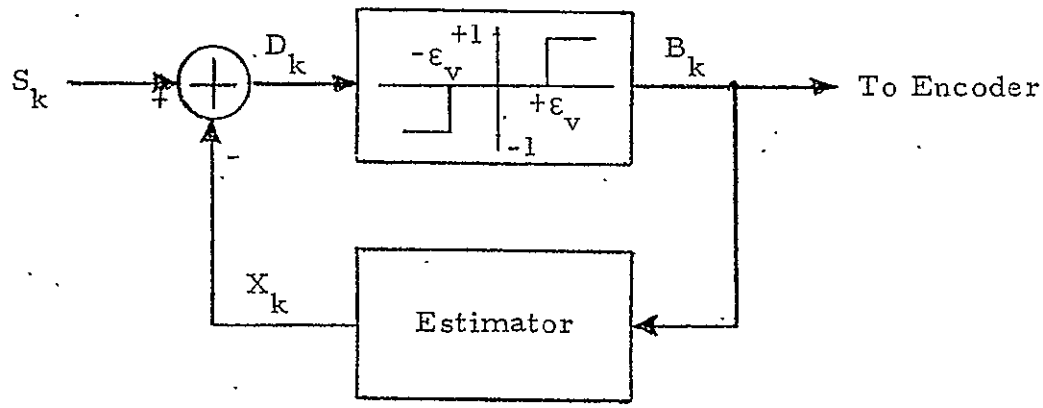


Figure 51. Block Diagram of the TSDM

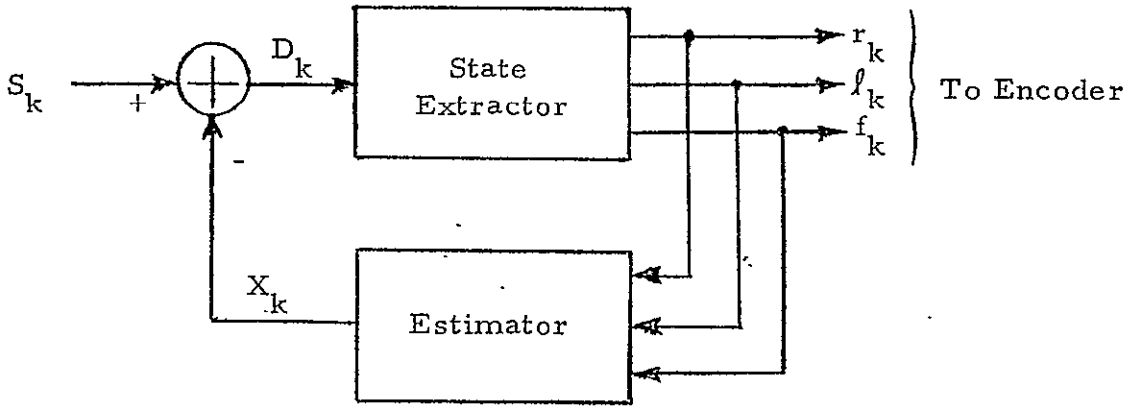


Figure 52. Alternate Block Diagram of the TSDM

6.2.2 Operational Equations

The equations describing the operation of the tri-state delta modulator follow closely those of the conventional ones with modifications as indicated below. The device will behave exactly like a conventional adaptive DM in the acquisition mode. During each tracking mode situation, however, the estimator will simply keep repeating the voltage level estimated when the tracking mode has first been entered.

The equations, in particular, the one calculating the value of the coming step size, can be considered as the generalized versions of those describing the conventional DM. The definitions used are given below. According to the usual convention, k indicates the kth time sample. Let

S_k = the (sampled and digitized) signal to be transmitted

X_k = the estimate of S_k

r_k = an indicator, which when set to 1, indicates that the voltage output of the estimator need "rise" for meeting the signal

f_k = indicator for "falling"

ℓ_k = indicator for maintaining the steady-state condition (level)

a_k = any "activity" element of above (r_k, f_k, or ℓ_k)

B_k = numerical representation of a_k.

It is also required that the state indicators shall be mutually exclusive events, i.e., with

$$\left.\begin{matrix} r_k \\ f_k \\ \ell_k \end{matrix}\right\} \text{ 0 or 1, and letting } \left.\begin{matrix} r_k = a_{k1} \\ f_k = a_{k2} \\ \ell_k = a_{k3} \end{matrix}\right\},$$

equation (6.1) below should be satisfied:

$$\sum_{j=1}^{3} a_{kj} = r_k + f_k + \ell_k = 1. \qquad (6.1)$$

Also let $$D_k = S_k - X_k \,, \tag{6.2}$$

with $$X_k = X_{k-1} + \Delta_k \tag{6.3}$$

as in the CDM. Defining now,

$$B_k = \begin{cases} \text{sign } (D_k), \text{ whenever } |D_k| > \varepsilon_v \\ 0, \text{ otherwise} \,, \end{cases} \tag{6.4a}$$

we have (for the state-selector used) the following requirements:

$$r_k = \begin{cases} B_k, \text{ if } B_k > 0 \\ 0, \text{ otherwise} \end{cases}$$

$$f_k = \begin{cases} -B_k, \text{ if } B_k < 0 \\ 0, \text{ otherwise} \end{cases}$$

$$\ell_k = \begin{cases} 1, \text{ whenever } |D_k| \leq \varepsilon_v \text{ or } B_k = 0 \\ 0, \text{ otherwise} \,. \end{cases} \tag{6.4b}$$

The new equation for calculating the next step size is now:

$$\Delta_{k+1} = |\Delta_k| \cdot [\alpha(r_k - f_k) + \beta(r_{k-1} - f_{k-1})] \cdot (1 - \ell_k) \tag{6.5a}$$

with $$\Delta_{k+1} = 2\Delta_0 B_k \text{ whenever } |\Delta_{k+1}| \leq 2\Delta_0 \text{ and } \ell_k \neq 1 \tag{6.5b}$$

or $$\Delta_{k+1} = 2\Delta_0 B_k \text{ whenever } \ell_k = 0 \text{ and } \ell_{k-1} = 1 \,. \tag{6.5c}$$

In (6.5a) and (6.5b), $2\Delta_0$ is the minimum permitted step size in the acquisition mode. Also, terms α and β are the same positive constants which are used to define the operation of the conventional (adaptive) delta modulators. Note also that (6.5a) can be written as

$$\Delta_{k+1} = |\Delta_k| \cdot [\alpha B_k + \beta B_{k-1}] \cdot |B_k| \,. \tag{6.5d}$$

The ε_v in (6.4) represents, as already mentioned, a small positive constant that takes into consideration the voltage tolerance levels inherent in the amplitude comparators or the signal smoothing requirements.

When EIP (essential information preserving) rather than SIP (strictly information preserving) systems are designed, the value of ε_v will have to be externally predetermined (as an additional parameter) to meet the required smoothing levels. This voltage tolerance parameter can therefore, when needed, serve as a powerful data compression tool in picture transmission systems.

From the equations presented above, it is evident that the operation begins with the minimal step size every time the device leaves the level state. Such an approach provides for simplification of device implementation without limiting significantly the acquisition speed.

It must be noted also that the demodulator, as in the CDM case, is an accurate replica of the estimator portion of the transmitting modulator.

6.2.3 Implementation Block Diagram

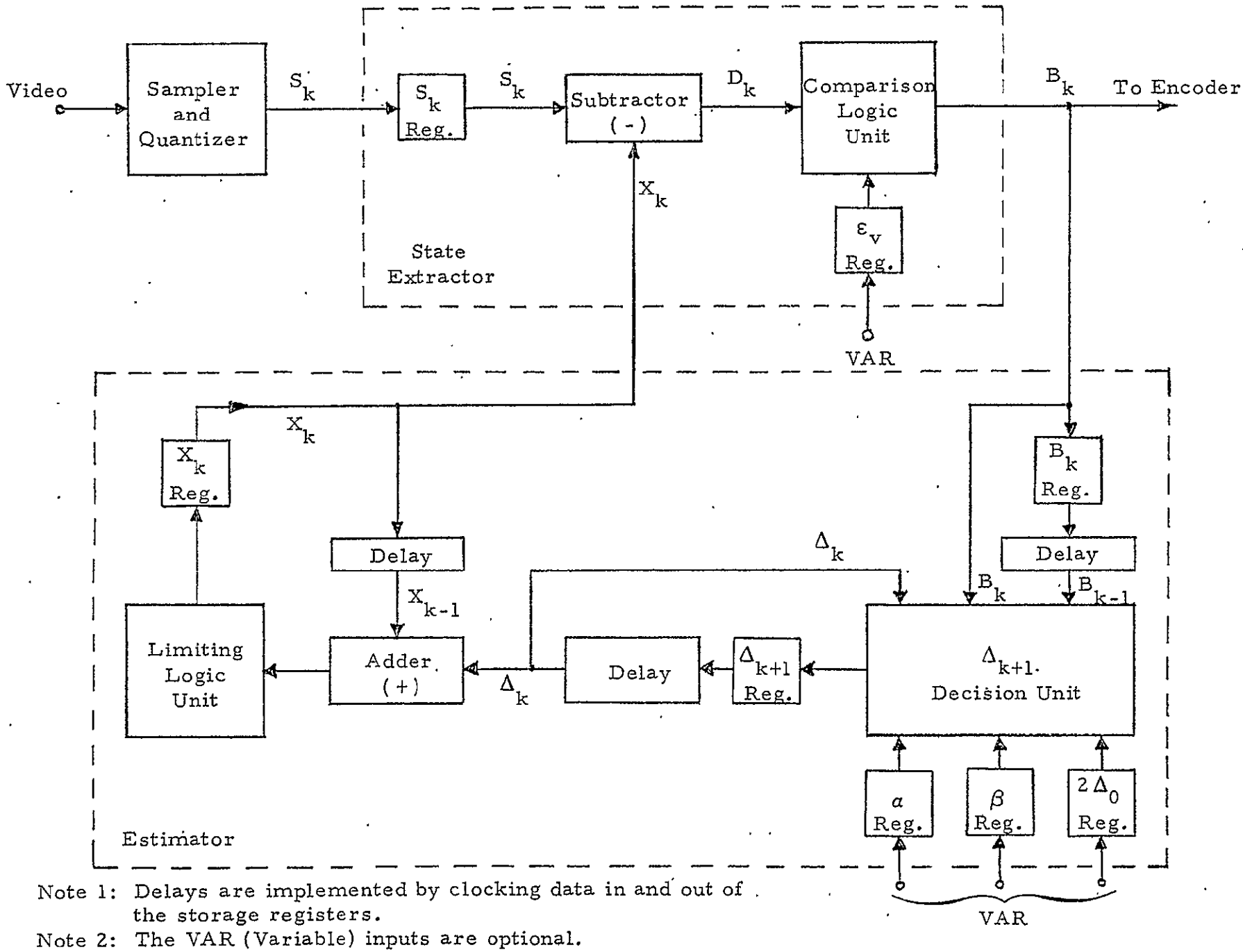
As stated in the preceding section, the equation which defines the next incremental step Δ_{k+1} as a function of the current and the previous state parameters can be written in the form given in (6.5d). The block diagram which carries out the function of this equation, as well as other functions of the delta modulator, is shown in Figure 53. Its salient subunits are described below.

6.2.3.1 State Extractor

The state extractor subunit accepts the quantized signal sample S_k and subtracts from it the present estimate X_k, thus forming the difference D_k. The difference D_k is then applied to a logic unit which compares D_k to ε_v (dead zone value). Depending on the results of this comparison, the logic unit outputs the present activity state information vector B_k. The B_k indicator is applied to the encoder for the ultimate transmission to the receiver; it is also applied to the estimator.

6.2.3.2 The Estimator

The estimator subunit accepts the current state information B_k from the state extractor, delays it until the next sampling period to provide B_{k-1}, and uses both the B_k and B_{k-1} information for generating the next step size Δ_{k+1}. The estimator subunit also uses the previous



Note 1: Delays are implemented by clocking data in and out of the storage registers.

Note 2: The VAR (Variable) inputs are optional.

Figure 53. Implementation Block Diagram of Tri-State Delta Modulator

estimate X_{k-1} and the current step Δ_k for generating the current estimate X_k. The functional subunits comprising the estimator are described below.

(1) The Limiter. The limiter unit keeps the maximum and minimum values of the estimate X_k within the bounds set for this signal.

(2) The Registers. The current values of X_k, B_k and Δ_{k+1} are stored in their respective registers. The required time delays are implemented by clocking data in and out of the registers at the appropriate times.

(3) The Adder. The function of the adder is to generate the new (i.e., present) value of X_k by summing the current step Δ_k with the previous estimate of X_{k-1}.

(4) The Δ_{k+1} Decision Unit. The Δ_{k+1} decision unit is the most important block of the estimator. It determines the magnitude of the next step according to the current and past values of B_k and the current step Δ_k. Table 10 defines the values of Δ_{k+1} as functions of B_k and B_{k-1}.

Table 10. Outcomes of Δ_{k+1} Decision Unit Versus B_k and B_{k-1}

y	B_k	B_{k-1}	Δ_{k+1}
y_0	0	0	0
y_1	0	-1	0
y_2	0	1	0
y_3	-1	0	$-2\Delta_0$
y_4	-1	-1	$-(\alpha+\beta)\cdot\lvert\Delta_k\rvert$
y_5	-1	1	$-(\alpha-\beta)\cdot\lvert\Delta_k\rvert$
y_6	1	0	$+2\Delta_0$
y_7	1	-1	$(\alpha-\beta)\cdot\lvert\Delta_k\rvert$
y_8	1	1	$(\alpha+\beta)\cdot\lvert\Delta_k\rvert$

Note: $|\Delta_k| \geq 2\Delta_0$.

From Table 9, it is evident that the Δ_{k+1} decision unit will do the following:

(a) It will clear the Δ_{k+1} register when $B_k = 0$.

(b) It will set the Δ_{k+1} register to $2\Delta_0$ and assign the sign of B_k to this step when $B_{k-1} = 0$, but $B_k \neq 0$.

(c) For all the remaining possible situations, it will follow the behavior of an adaptive conventional delta modulator (CDM) defined by parameters of α and β.

As shown in Table 10 there are nine possible input combinations of B_k and B_{k-1}. Furthermore, out of these nine combinations, the number that need distinct handling can be reduced, thus simplifying the implementation of the TSDM. To allow such simplification, values $\alpha = 1$ and $\beta = 1/2$ must be used. These values, however, have gained wide acceptance in CDM and thus are the logical candidates for use in TSDM.

When $\alpha = 1$ and $\beta = 1/2$, the ratios between Δ_{k+1} and Δ_k can only be either $\pm 1\text{-}1/2$ or $\pm 1/2$, not considering the cases when Δ_{k+1} must be equal to $2\Delta_0$. Because dividing by 2 is equivalent to shifting Δ_k to the right one time and truncating the result, multiplication by $1\text{-}1/2$ is easily obtained by adding Δ_k to its shifted version. In this manner, the multiplication operation is reduced to a simple shift operation or to a shift-and-add with the proper sign bit taken into consideration.

Pertinent to such an implementation, as well as to its variations, is the timing of various events which take place each time the input is sampled and operated upon. A typical version of such a timing sequence is presented below.

6.2.4 Timing Sequence for TSDM with $\alpha = 1$ and $\beta = 1/2$

Not considering the overshoot suppression (OSS), the timing sequence for each operation cycle of TSDM may typically take up to ten phases, each phase determined by its timing signal C_n. The sequence and the functions of these timing pulses are as follows:

Timing Pulse C_1

The analog video signal is sampled and quantized.

Timing Pulse C_2

(1) The output of the quantizer is transferred to the S_k register.

(2) The estimate of S_k is transferred from the limiting network to the X_k register.

Timing Pulse C_3

The estimate X_k is subtracted from S_k, resulting in output D_k.

Timing Pulse C_4

(1) The magnitude of D_k is compared to the contents of the ε_v register.

(2) The two-bit word B_k is generated in terms of b_1^k and b_0^k.

Timing Pulse C_5

(1) B_k is clocked into a pair of flip-flops.

(2) B_{k-1} is transferred to a second pair of flip-flops.

(3) The contents of the Δ_k register are transferred into the "shifted Δ_k" register.

(4) The trailing edge of C_5 applies B_k and B_{k-1} to a logic network which determines y_k control signal(s) according to Table 10.

Timing Pulse C_6

Control signal(s) y_k are "anded" with C_6 and decision to clear either one or both (Δ_k and "shifted Δ_k") registers is made according to Table 11, the significance of which will be explained later.

Timing Pulse C_7

The outputs of Δ_k and "shifted Δ_k" registers are added to give the magnitude of Δ_{k+1}.

Timing Pulse C_8

(1) The value of $|\Delta_{k+1}|$ is compared against the stored value of $2\Delta_0$, the latter being the minimum step used.

(2) Control signals which determine whether $|\Delta_{k+1}|$ or $2\Delta_0$ is used for the next step are generated.

Timing Pulse C_9

Either $|\Delta_{k+1}|$ or $2\Delta_0$ is passed to the adder/subtracter.

Timing Pulse C_{10}

(1) Depending on the state of b_1^k, Δ_{k+1} or $2\Delta_0$ is either added to or subtracted from the x_k value.

(2) The result of the addition (or subtraction) is passed to the limiting network where it is processed and stored to be transferred to the x_k register upon the reoccurrence of C_1.

Timing Pulse C_{11}

If the OSS is implemented, it is processed after Timing Pulse C_{10}. However, as described in section 6.2.7, additional registers and flip-flops are required and their contents are operated by Timing Pulse C_{11}.

The sequence then repeats with the onset of the next Timing Pulse C_1.

6.2.5 Logic for Clearing Δ_k and Shifted Δ_k Registers

The logic for clearing the Δ_k and shifted Δ_k registers is determined by the states of B_k and B_{k-1}. This logic is summarized in Table 11, which also includes some of the information of Table 10 of section 6.2.3 to indicate how either the values or the coefficients of the Δ_k step are determined.

Table 11. Logic for Clearing the Δ_k and Shifted Δ_k Registers

B_k	B_{k-1}	Required Incremental Step	Clear Δ_k Register?	Clear Shifted Δ_k Register?
0	0	0	Yes (redundant)	Yes (redundant)
0	-1	0	Yes	Yes
0	1	0	Yes	Yes
-1	0	$-2\Delta_0$	Yes (redundant)	Yes (redundant)
-1	-1	$-1.5 \cdot \lvert\Delta_k\rvert$	No	No
-1	1	$0.5 \cdot \lvert\Delta_k\rvert$	Yes	No
1	0	$+2\Delta_0$	Yes (redundant)	Yes (redundant)
1	-1	$-0.5 \cdot \lvert\Delta_k\rvert$	Yes	No
1	1	$1.5 \cdot \lvert\Delta_k\rvert$	No	No

Most of the required, i.e., nonredundant, decisions for clearing or not clearing the Δ_k and shifted Δ_k registers are obvious from determining the coefficients of the $|\Delta_k|$ terms in Table 11. The decisions to clear the registers for the conditions associated with $B_{k-1} = 0$ are considered redundant because for this condition the contents of both registers under consideration are zero anyway. Thus, the decisions to clear both registers for case $B_{k-1} = 0$ should be utilized only if such operation simplifies the logic or is used to preserve the synchronism of the operational cycle.

6.2.6 Parallel Data Handling for High Speed Operation

In section 6.2.4, the timing sequence for the tri-state delta modulator was presented and the functions of the various phases of this sequence were described. The timing sequence presented there was based on the assumption of a serial digital data generation and processing. Such serial data handling, however, requires excessive clock speeds, which in turn may unnecessarily complicate the implementation. This applies particularly to the Analog-to-Digital (A/D) unit which typically consists of a sampler and a serially operated digital quantizer.

Our investigation of parallel data handling techniques and of the corresponding state-of-the-art digital devices indicates that parallel handling of the arithmetic functions within the estimator loop can permit intersample periods as short as 100 nsec and possibly even as short as 50 nsec.

Assuming that a 50 nsec intersample period is achievable, two samples per picture element may be performed with potentially significant improvement of the modulator performance during the fast and large signal level changes. Such deliberate "oversampling" may serve as a good protections against slope overloads, which are the common bane of even the adaptive delta modulators.

Figure 54 shows the block diagram of the tri-state delta modulator based on parallel data handling. Note that the salient feature of this block diagram is the absence of the sampler and quantizer at the input to the subtractor. Instead, the analog estimate X_k' (the prime indicates analog form) is subtracted from the analog video signal S_k' and the analog difference D_k' is applied to the comparator and state extractor. Since

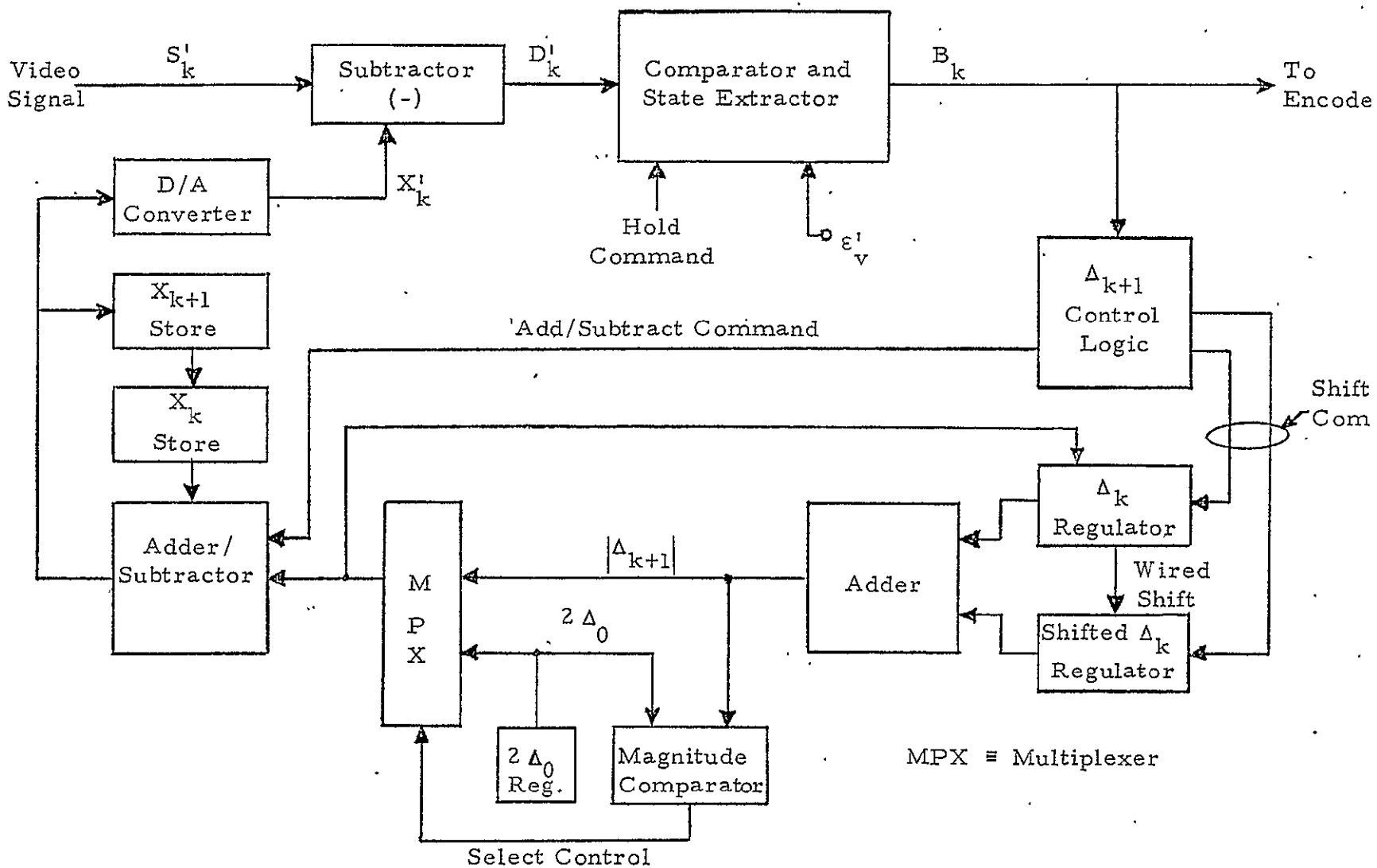


Figure 54. Implementation Block Diagram of Tri-State Delta Modulator (Parallel Data Handling)

all of the functions of the estimator unit are based on the values of B_k and B_{k-1}, the latter being stored in the control logic unit, the estimator is simply a comparator which has to generate only a single two-bit word per sample frame. The word generated in this case supplies the logic with only one of the following three possible messages:

(1) $D_k' > \varepsilon_v'$ (signal rising)

(2) $D_k' > -\varepsilon_v'$ (signal falling)

(3) $D_k' \leq |\varepsilon_v'|$ (signal within dead zone defined by $\pm\varepsilon_v'$)

The third message for the tri-state delta modulator implies that the signal is not changing, i.e., a "level" state is achieved.

Thus, one can see that a multi-level, multi-bit signal amplitude estimator (A/D converter) can be replaced by a far simpler, and faster, tri-state comparator and state extractor. Such simplification is made possible by using a Digital-to-Analog convert (DAC) for transforming the digitally computed signal estimate X_k into an analog voltage X_k'. In this case, we take advantage of the fact that the D/A conversion is generally simpler to perform than A/D conversion.

The operations within the estimator are performed as shown in the block diagram of Figure 54, and they generally follow the timing sequence described in section 6.2.4 with the exception of some modifications in the timing sequence. These modifications are the result of changing the digital data handling operations from a serial to a parallel mode.

6.2.7 Overshoot Suppression (OSS) Implementation

The TSDM described so far, despite its unique features, has all the characteristics of an adaptive delta modulator. The major advantage of an adaptive delta modulator over a linear version are: (a) reduced slope overload and (b) lower level of granularity noise. The penalty paid for these advantages, however, is an increased level of overshoots and of the accompanying oscillations. Consequently, to minimize the effects of the overshoot and oscillations which accompany large transitions from one video signal level to another, an overshoot suppression technique should be considered for the tri-state delta modulator.

In reference [25]. which describes techniques for overshoot and undershoot suppression, the former is detected when the transmitted B_k sequence is 1, 1, -1, -1 and, correspondingly, the undershoot is detected when the sequence is -1, -1, 1, 1. Such sequences, however, can occur following a large change in direction of the video signal. Thus, the overshoot suppression algorithm may introduce an undesirable smoothing of the transmitted signal. Furthermore, if the adaptive steps are small, the above patterns could occur in the normal tracking of the changing video signal.

To overcome the aforementioned problems, it is proposed that two extra delays be added and the overshoot and undershoot be detected according to the following pattern of B_k:

Overshoot: 1, 1, 1, -1, -1, 1

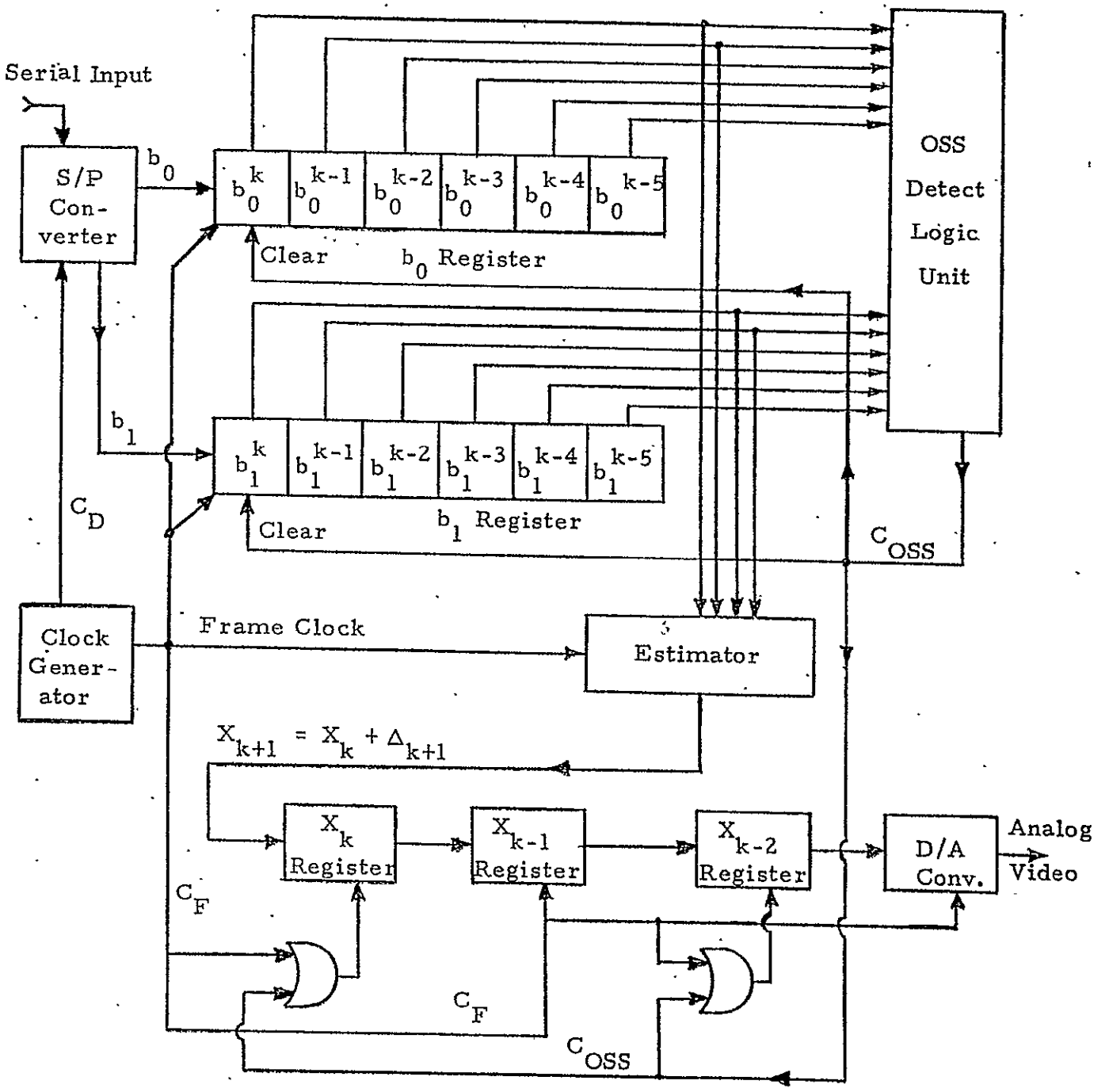
Undershoot: -1, -1, -1, 1, 1, -1 .

Note that the six-character sequences proposed contain in their middle the four-character sequences described in [25]. But the added bit at the beginning of the sequence insures that the step is large enough for an overshoot to have occurred, and the sixth bit at the end indicates that the overshoot has definitely occurred and that the signal is not continuing downward. Similar reasoning applies to the two extreme bits of the undershoot indicator sequence.

Figure 55 shows the proposed implementation for the overshoot suppression (OSS). The implementation shown applies to the receiver, i.e., the demodulator portion of the digital TV link. As shown in the figure, the incoming serial input is converted into parallel outputs of b_0 and b_1. For each kth sample clocked out of the serial-to-parallel (S/P) converter at the frame clock rate C_F, the b_0^k and b_1^k bits are deposited in their respective registers to store the corresponding value of B_k. Thus, a continuously updated sequence of six values of B_k, i.e., B_k through B_{k-5}, are supplied to the OSS Detect Logic Unit.

As in the transmitting modulator, the estimator portion of the receiver shown in Figure 55 determines the values of X_k and Δ_{k+1} based on the information available from state vectors B_k and B_{k-1}. However, the signal applied to the digital-to-analog converter is delayed by two sample periods. In other words, analog video output is derived from X_{k-2} instead of X_k.

Figure 55. Overshoot Suppression (OSS) Implementation for Demodulator



$B_k \equiv f(b_0^k, b_1^k)$

C_F = Sample Frame Clock

C_{OSS} = OSS Detect Pulse

C_D = Decoder Clock

Because the values of B_k and B_{k-1} are available to the estimator at the onset of the X_{k+1} formation cycle, a typical cycle, not involving OSS, is as follows:

Step 1: Form Δ_{k+1} according to

$$\Delta_{k+1} = |\Delta_k| \cdot [\alpha B_k + \beta B_{k-1}] \cdot |B_k|$$

where Δ_k = present step

α, β = weighting factors

B_k = present activity state information (0, 1, or -1)

B_{k-1} = previous sample activity state information (0, 1, or -1).

Step 2: Form X_{k+1} as

$$X_{k+1} = X_k + \Delta_{k+1} .$$

Step 3: Shift B_k registers to the right, set X_{k-2} to X_{k-1}, X_{k-1} to X_k, X_k to X_{k+1}, and return to Step 1.

Now, if an OSS condition is present, the detection logic puts out an extra clock pulse C_{OSS}.* Because this extra pulse precedes the next "normal" frame clock pulse C_F,** the following operation takes place after Step 2:

Step 2a: Set X_k to X_{k+1}, X_{k-2} to X_{k-1}, and $B_k = 0$.

This extra step is then followed by Step 3 as in normal operation. The results are different, however, because now we have set $B_k = 0$ in Step 2a, and this will prevent the OSS pattern from occurring for at least six sampling time intervals. Furthermore, setting $B_k = 0$ will also cause the size of the next step, if required, to be the minimal size of $2\Delta_0$ as shown in Table 11. Such selection takes place when $B_k = 0$ is shifted by C_F to become $B_{k-1} = 0$ for the subsequent frame.

---

*Similar to C_{11} of section 6.2.4.

**Similar to C_1 of section 6.2.4.

7.0 ASSESSMENT OF PREVIOUSLY CONTRACTED EFFORTS

7.1 NASA/Ames Three-Dimensional Hadamard Transfer Coding System

A three-dimensional coding system that would take a transformation in temporal direction, as well as transformations on the spatial domain, exploits the frame-to-frame correlation, as well as the correlation in the temporal direction, thus achieving a better coding performance than the two-dimensional transform coding methods. Although this is possible using any one of the unitary transformations, the complexity of the system is reduced by using a transformation that requires minimal hardware complexity for transform implementation, such as the Hadamard transform.

A three-dimensional Hadamard coder, which employs 2x2 pixel blocks by 2 frames in time, is currently under investigation at Bell Telephone Laboratories. This system exploits the correlation of the data in both spatial and temporal directions, requires only one frame of memory, and is amenable to simple hardware implementation. However, this coding technique is not highly efficient, due to the rather small number of pixels (a total of seven) used to exploit correlations in the data. Hardware for this encoder is in the development stage at Bell Telephone Laboratories [26].

NASA/Ames Research Center at Moffett Field, California, has developed a three-dimensional Hadamard transform coder that uses a block size of 4x4 pixels and 4 frames in time. This coder is designed to reduce the bandwidth of high resolution satellite television signals transmitted at 30 fps. It employs a 4x4 pixel sliding window technique to eliminate visual edge effects due to the Hadamard encoder.

The encoder is designed for standard U.S. commercial television signals; thus, it uses 525 lines per frame. However, each line is sampled to generate 512 samples per line. This corresponds to 8 megasamples per second on a corresponding 4 MHz analog bandwidth for the original television signal. A functional block diagram of this encoder is shown in Figure 56. The Hadamard transformation is applied to sub-blocks of 4x4x4 samples. The bandwidth reduction is obtained by assigning more bits to lower coefficients and less bits to higher coefficients. The system is capable of operating from off-air television signals, video tapes or television cameras. A block diagram showing the field demultiplexing essential

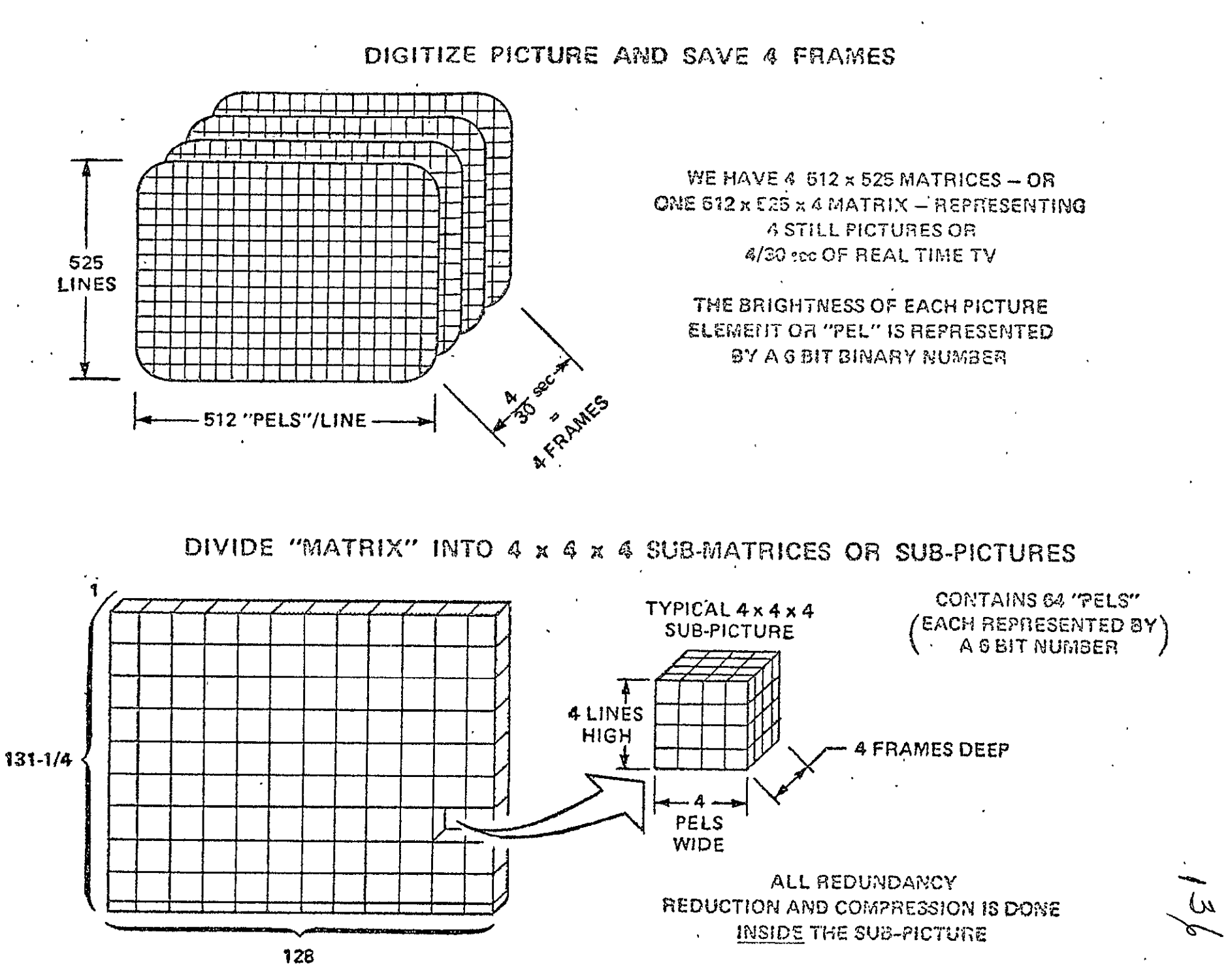


Figure 56. Functional Block Diagram of 3D Hadamard Encoder

to perform the three-dimensional Hadamard transform of interlaced signals is shown on Figure 57. The block diagram of the decoder is shown on Figure 58. The system can be programmed to operate at various bit rates. It also has an adaptive mode of operation that selects a combination of vectors and cut points best fitted to data.

The shortcoming of this system is the large storage requirement which is needed to store 4 frames of data. The system can be modified to encode color television by operating on the illuminance and the chromaticity components individually. This will require storing as many as 12 frames of data at the transmitter, though it is conceivable that this number could be reduced due to the fact that the chromaticity signals require a much smaller bandwidth. The performance of the existing encoder operating on the frame-sequential color video signal is anticipated to be rather poor, due to smaller correlation in the spectral bands as compared to the correlation of sequential frames in monochrome video signal.

7.2 LINKABIT Three-Dimensional Hybrid Encoder

Three-dimensional transform coding systems suffer from the shortcoming of excessive storage requirement which is needed to store previous frames. A three-dimensional hybrid encoder that uses a two-dimensional transformation on the spatial domain cascaded with a DPCM encoder in the temporal domain will require storing only one frame of data and should perform better than the corresponding three-dimensional encoders.

Before discussing the LINKABIT approach in detail, it is worth noting that an investigation of the hybrid encoder using two-dimensional cosine transform cascaded with a DPCM encoder for reducing the bandwidth of RPV imagery is underway at Naval Undersea Center (NUC) in San Diego, California [27].

This hybrid encoder exploits spatial correlation of a television image by taking a two-dimensional discrete cosine transform and exploits temporal correlation of the data by using a DPCM encoder. It is anticipated that this system will reduce the number of binary digits needed for reconstruction of television at the receiver by a factor of about 5 over the two-dimensional hybrid encoder. This system can be modified to encode illuminance and the chromaticity components of a color video signal. This will require storing three frames of data at the transmitter. The

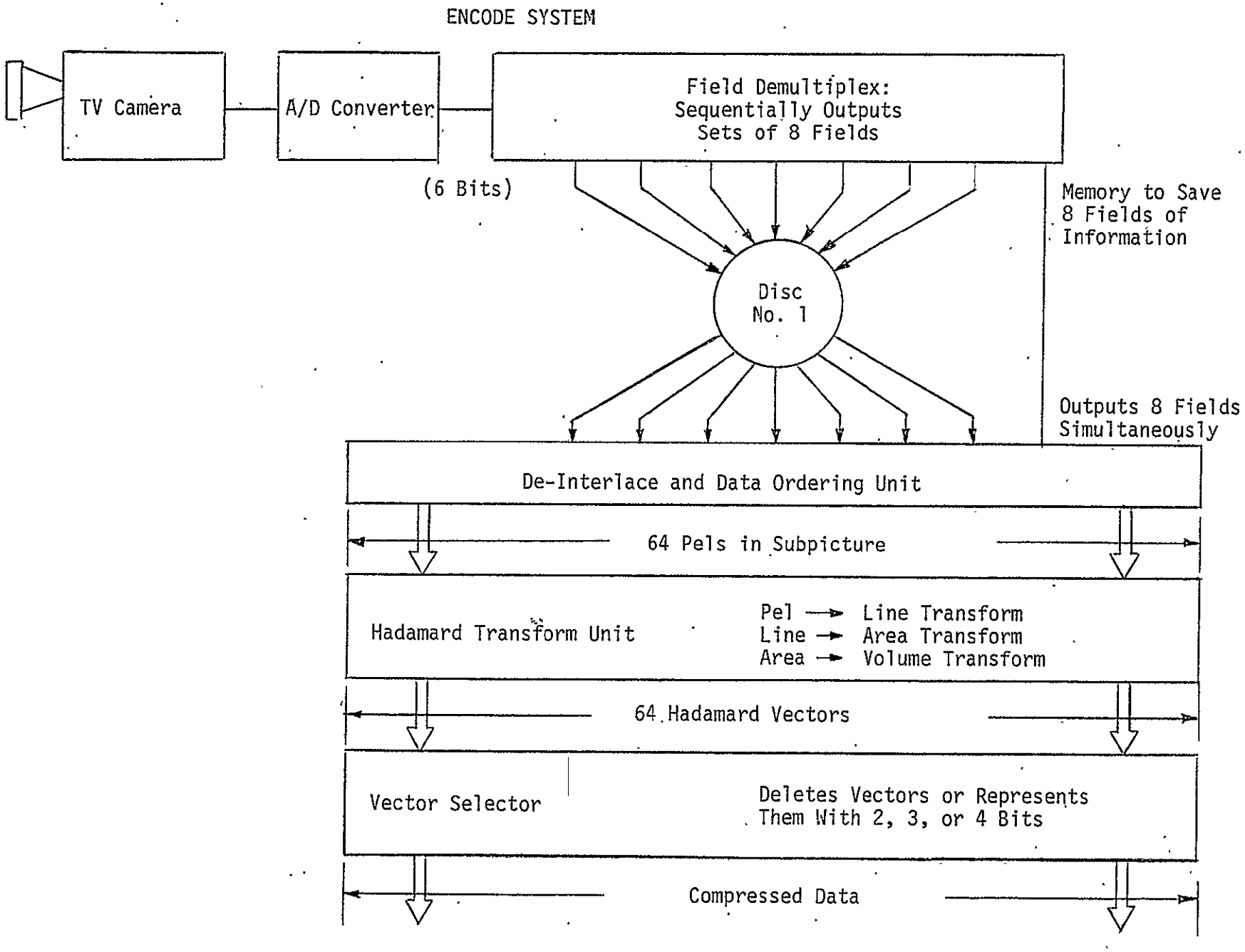


Figure 57. Field Demultiplexer

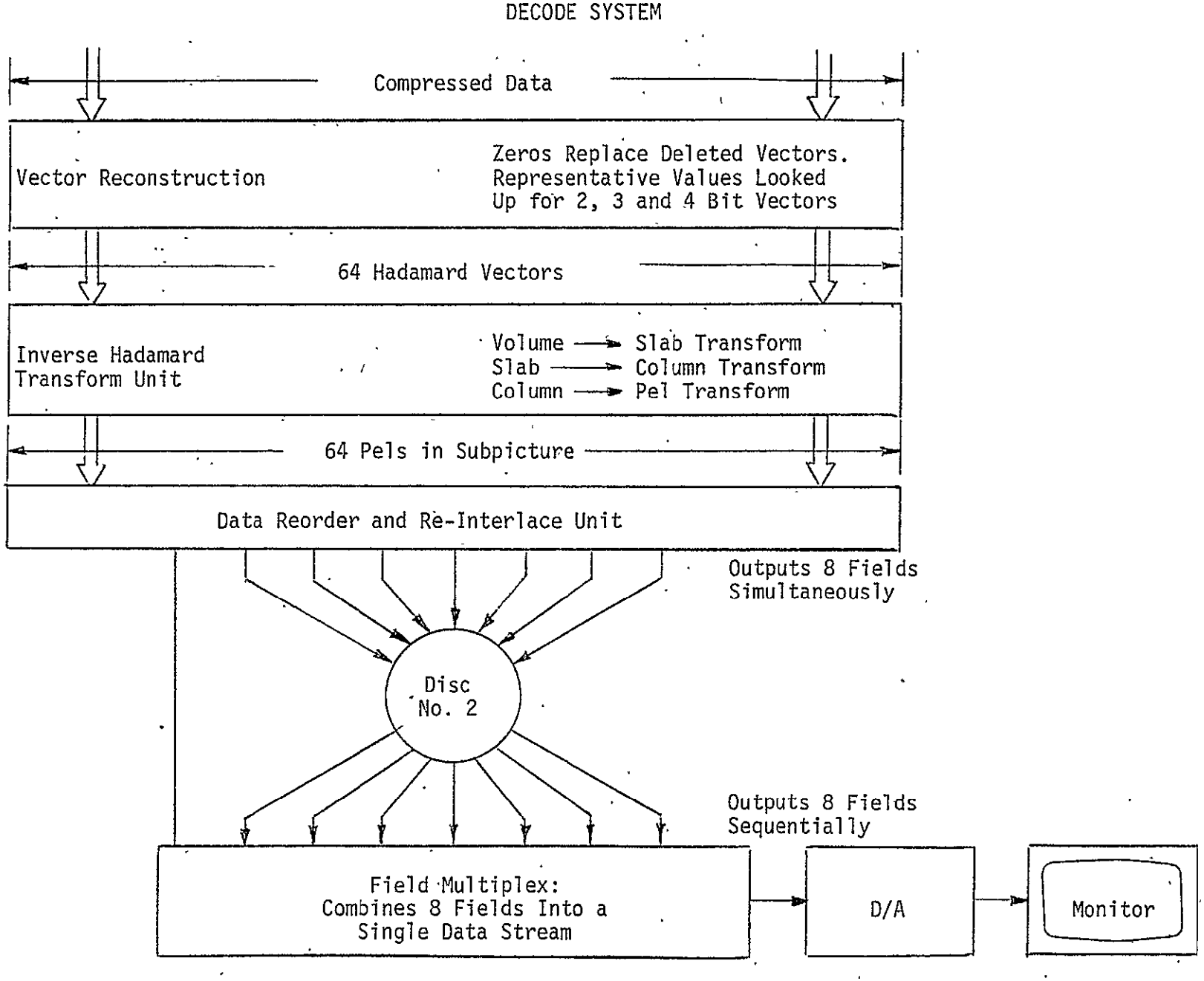


Figure 58. Field Multiplexer

performance of this system operating on the frame-sequential color video data is expected to be inferior to its performance for monochrome television signals.

The LINKABIT scheme is significantly simpler than the NUC design. In the LINKABIT case, a two-dimension 4x4 Hadamard transform is computed per frame and only the difference of transform coefficients from frame to frame is quantized. This technique has many of the advantages of other three-dimensional algorithms but considerably less memory is required and there is no rate buffering due to variable scene activity.

Conceptually, this scheme can be thought of as transmitting the compressed two-dimensional Hadamard transform coefficients for the first frame. The two-dimensional quantization is performed according to [28]. Figure 59 shows the 16 Hadamard subpictures (basis vectors). The numbers in parentheses below each subpicture are the number of bits of quantization allocated to the corresponding coefficients. The total number of bits for 16 pels (the sum of the number in parentheses) is 32, or 2 bits per pel.

For three frames after the first (reference) frame, only quantized differences between the compressed transform coefficients of a subpicture, and the corresponding subpicture in the past frame, are transmitted. Actually, only differences on three of the 16 coefficients are sent for a total of 11, 11 and 10 bits sent per 16 pel array in the second, third, and fourth frames, respectively. This results in an average of one bit per pel. This procedure is repeated by again transmitting a compressed two-dimensional frame independent of previous transmissions. The additional compression over that obtained with a two-dimensional Hadamard transform scheme is possible because the difference coefficients can be quantized much more coarsely than the two-dimensional transform coefficients themselves. The smaller memory is possible because it is sufficient to store a compressed, rather than an uncompressed, form of the reference frame.

The encoder accepts as input a standard composite NTSC black-and-white video signal. The outputs of the encoder are a compressed 8.064 Mbps data stream and its associated clock. The received 8.064 Mbps compressed stream and its recovered clock form the input to the decoder. The decoder output is a reconstructed NTSC composite video signal. Two

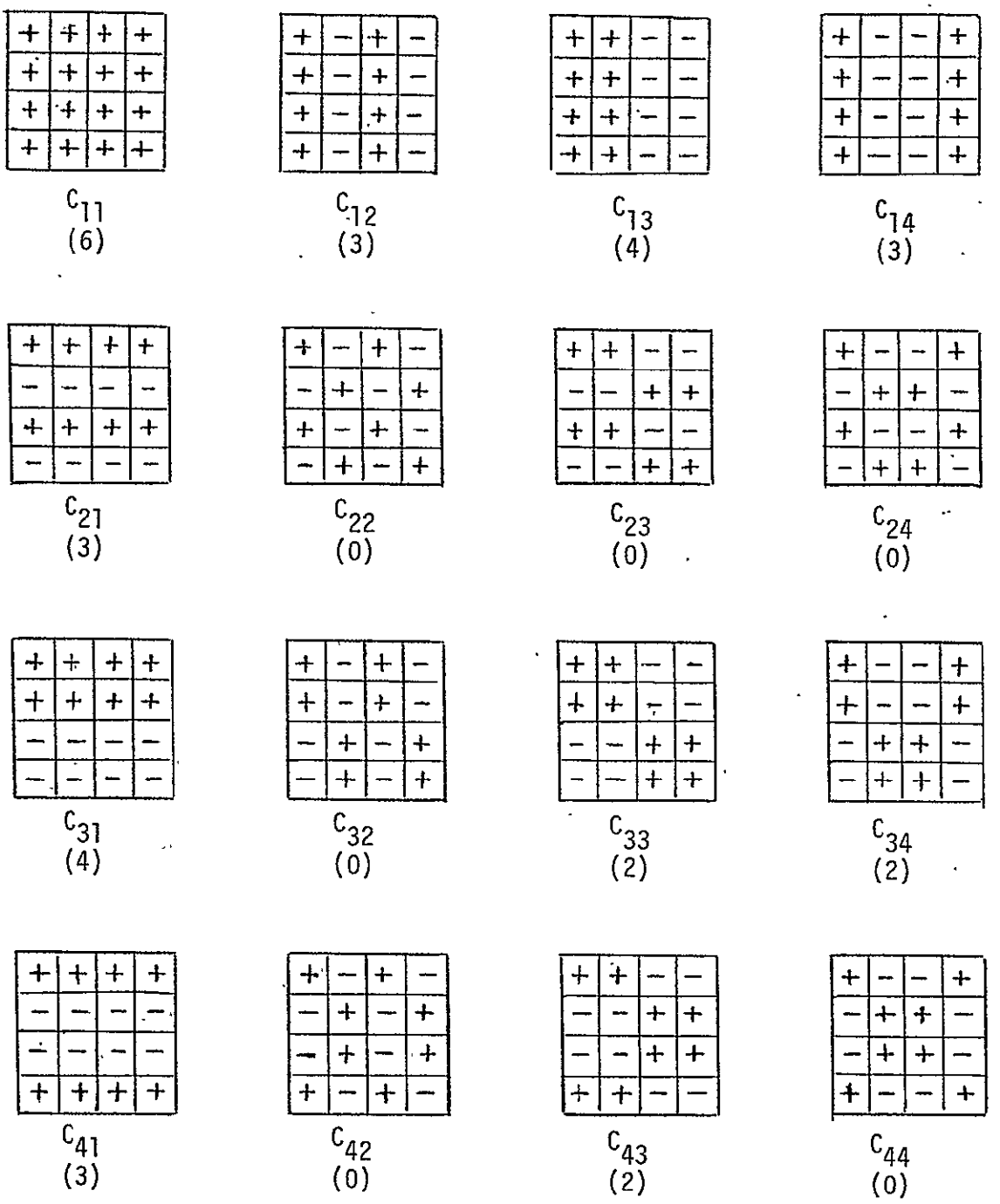


Figure 59. Quantization/Discard Strategy for LINKABIT System

audio channels, sampled and 8-bit quantized at the video line rate (15.75 kHz), are multiplexed into the compressed video data stream during the horizontal sync pulse time. The encoder also inserts a sync word in the output data stream every four frames. Circuitry is provided in the decoder to acquire and track this sync word.

Figures 60 and 61 are block diagrams of the video compression encoder and decoder, respectively.

Operation of the encoder proceeds as follows: The composite video input signal is amplified, DC restored and 8-bit A/D converted. The 4-line buffer stores quantized samples on four video lines and serially provides the Hadamard transformer with sets of 16 samples within 4x4 square arrays. The transform coefficients are further quantized with the number of bits per coefficient given in Figure 59. The quantization is nonlinear following [28].

To minimize data rate smoothing, quantized Hadamard coefficients are sent for one-fourth of the 4x4 pel arrays in each frame (these are called refresh arrays). Quantized coefficient differences are sent for the remaining three-fourths of the 4x4 arrays. For refresh 4x4 arrays, the switches are in the positions shown in Figure 60. The quantized coefficients are stored in the Main (frame) Memory and are sent through the Rate Buffer to form the output data. For differenced 4x4 arrays, both switches in Figure 60 change positions. The Quantizer outputs are subtracted from the corresponding refresh array coefficients at the Main Memory output. These coefficient differences are further quantized and forwarded to the Rate Buffer.

All timing signals and control signals are derived by phase locking to the horizontal and vertical sync stripped from the composite input signal.

The decoder of Figure 61 performs essentially the inverse functions to those of the encoder. Inverting the encoder quantizing operations involves assigning "representative values" to quantization intervals. The decoder also acquires synchronization and regenerates composite sync which is added to the reconstructed video signal.

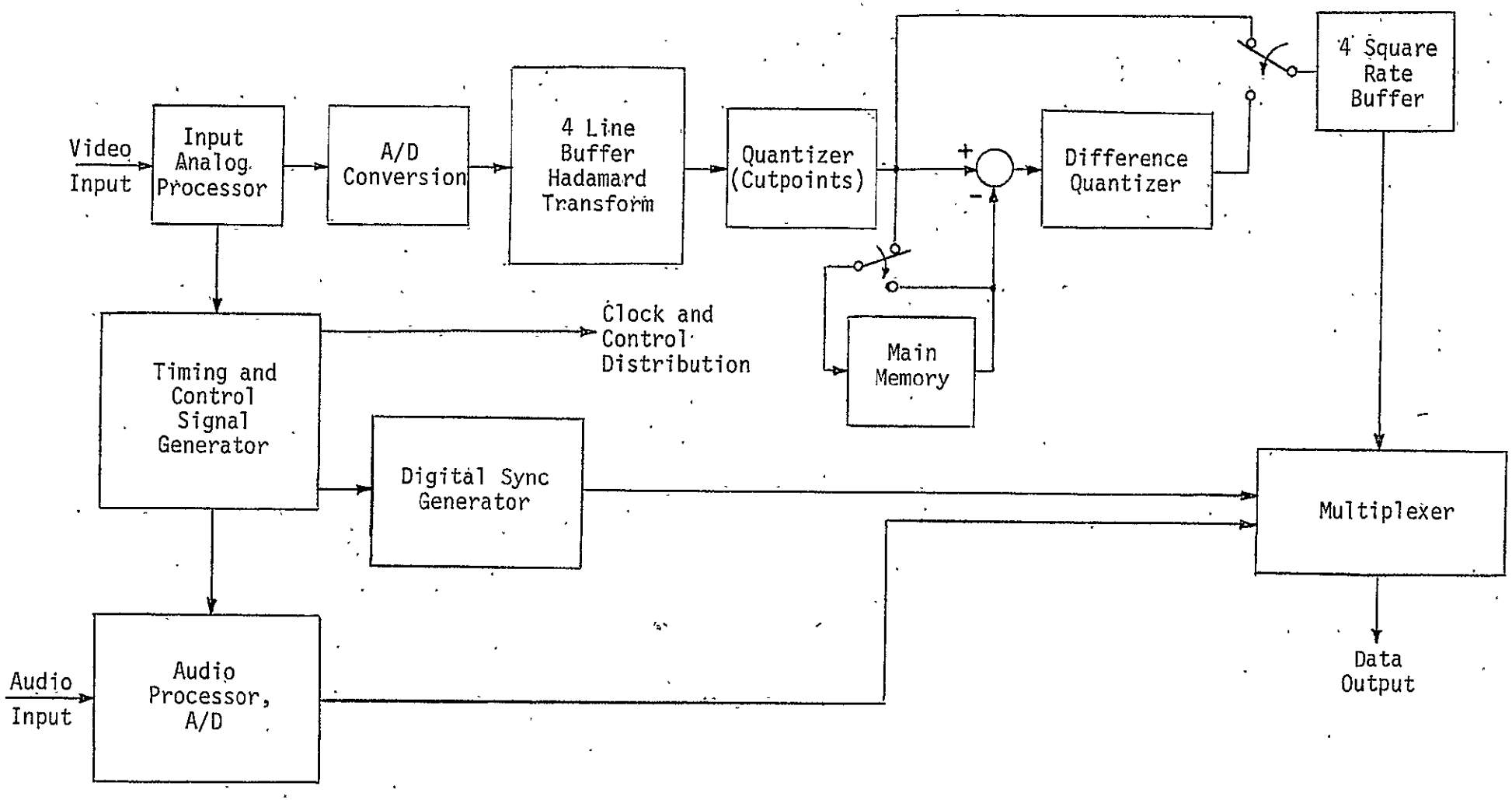


Figure 60. Video Compression Encoder

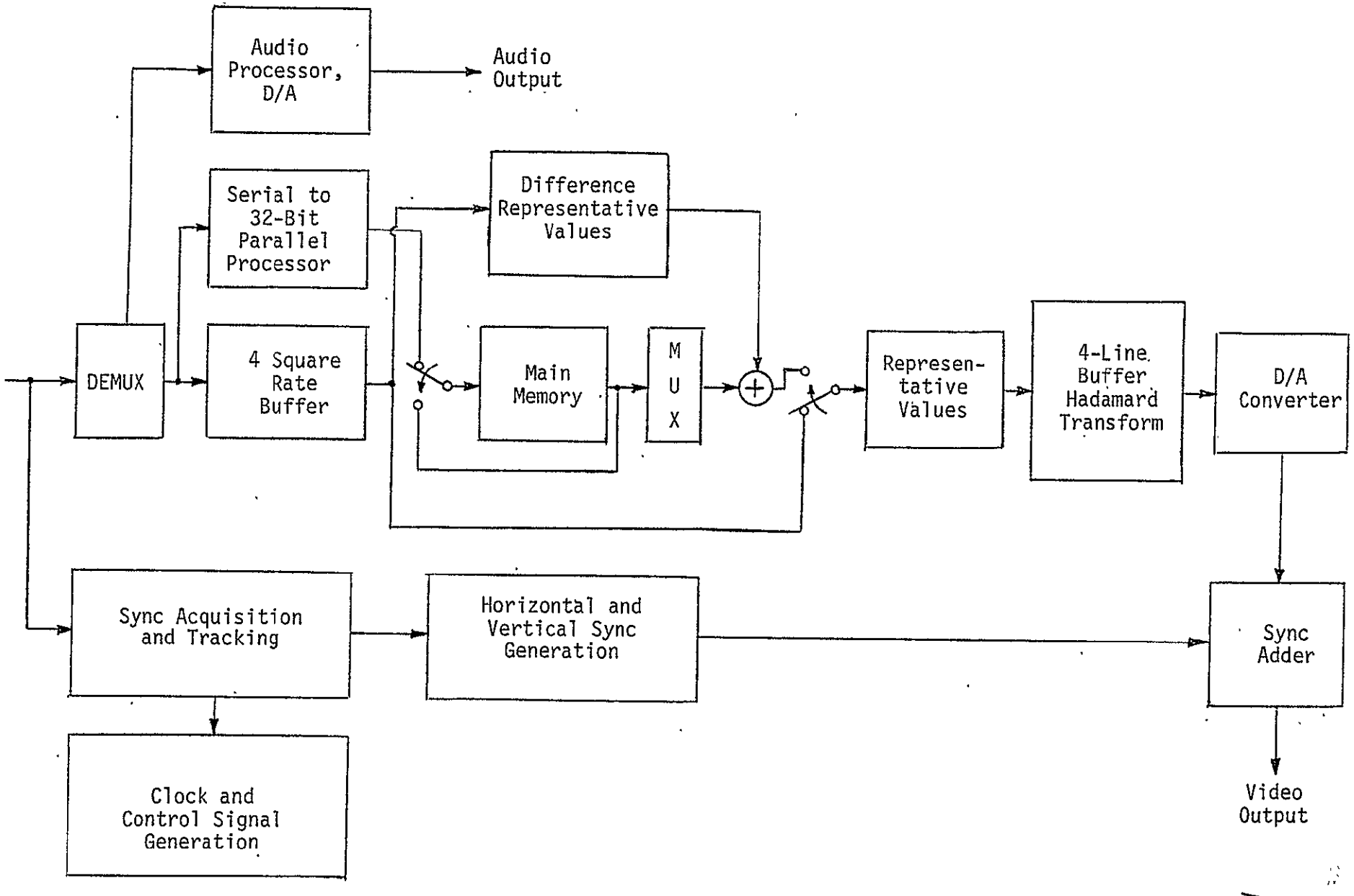


Figure 61. Video Compression Decoder

7.3 TRW 2D-DPCM for Color Data Compression

TRW analyzed the most economical approach to provide data compression of field-sequential color and NTSC color. In both cases, the most economical technique was 2D-DPCM.

Analysis of the field-sequential data, along with the weight and power sizing of the encoder, lead to an approach that consists of using the green (G) field instead of the illuminance, and generating the chromaticity signals by subtracting G from the red (R) and blue (B) fields. This approach is fruitful if the chrominance signals contain less energy and possess a smaller bandwidth than the original fields. Then the chrominance signals can be subsampled by taking every other sample and encoded using a coarser quantization, thus resulting in a further bandwidth compression. Table 12 shows the statistics of the G and two chromaticity signals for three typical fields of the field-sequential data. Figure 62 shows the power spectra of the green, as well as the chromaticity, signals. These results indicate that the chrominance signals indeed possess a smaller energy and a lower bandwidth than the original fields.

Table 12. Statistics of the Field-Sequential G, R-G, and B-G Fields

Fields	Minimum	Maximum	Average	Standard Deviation
G	0.0	255	117.05	51.77
R-G	-101	60	8.03	12.48
B-G	-88	77	1.80	9.30

The above processing of the field-sequential data results in some reduction in its bandwidth. This is due to subsampling of the R-G and B-G fields. A 2-to-1 subsampling of R-G and B-G results in a total bandwidth compression ratio of 1.5 to 1. To achieve additional bandwidth compression, the G, R-G, and B-G fields must be encoded. The performance of the candidate techniques considered by TRW [18] are shown on Figure 63. The two adaptive DPCM systems have essentially identical performances. The performance of the hybrid encoder, on the other hand,

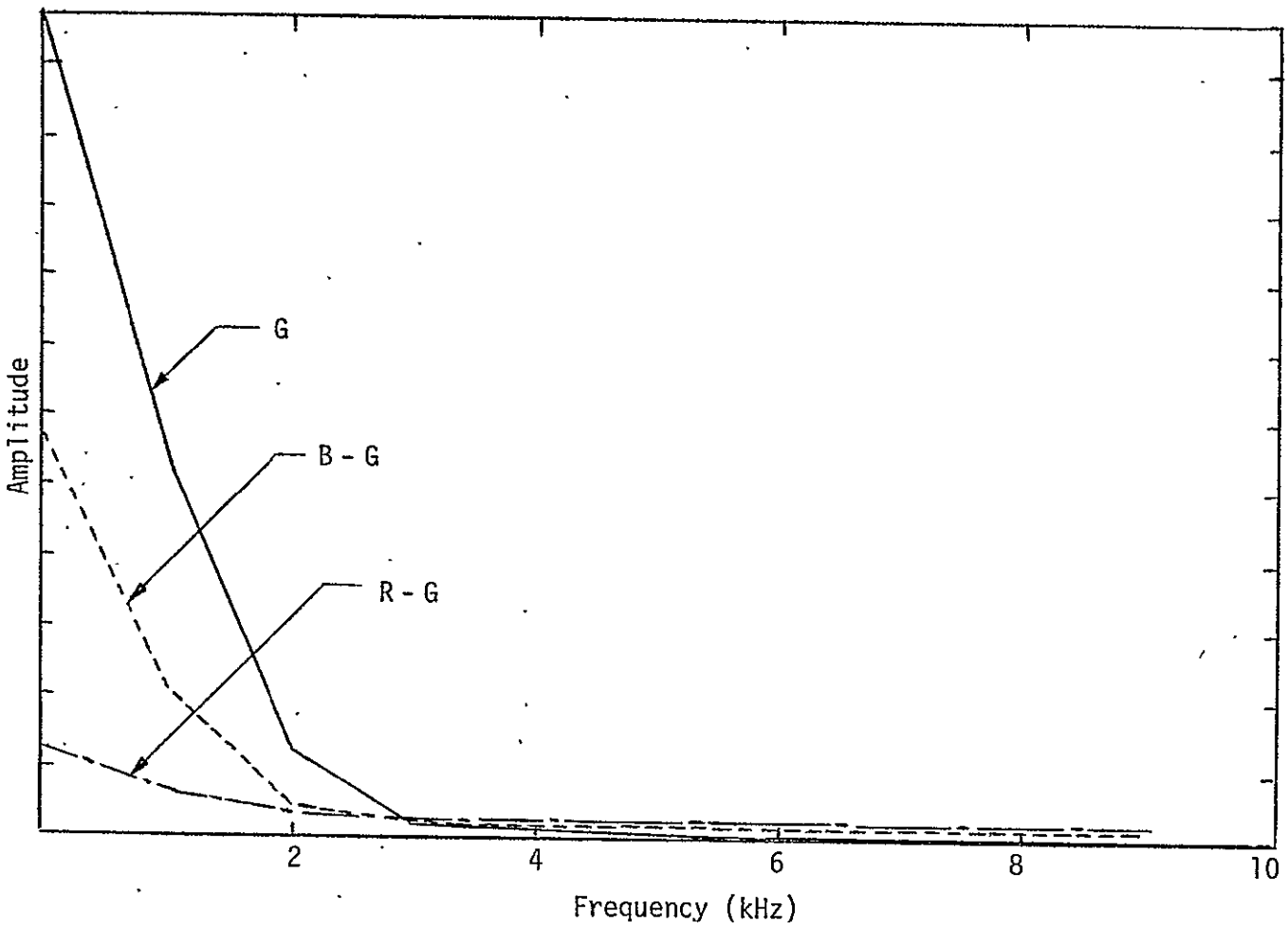


Figure 62. Relative Power Spectra of G, R-G, and B-G Fields

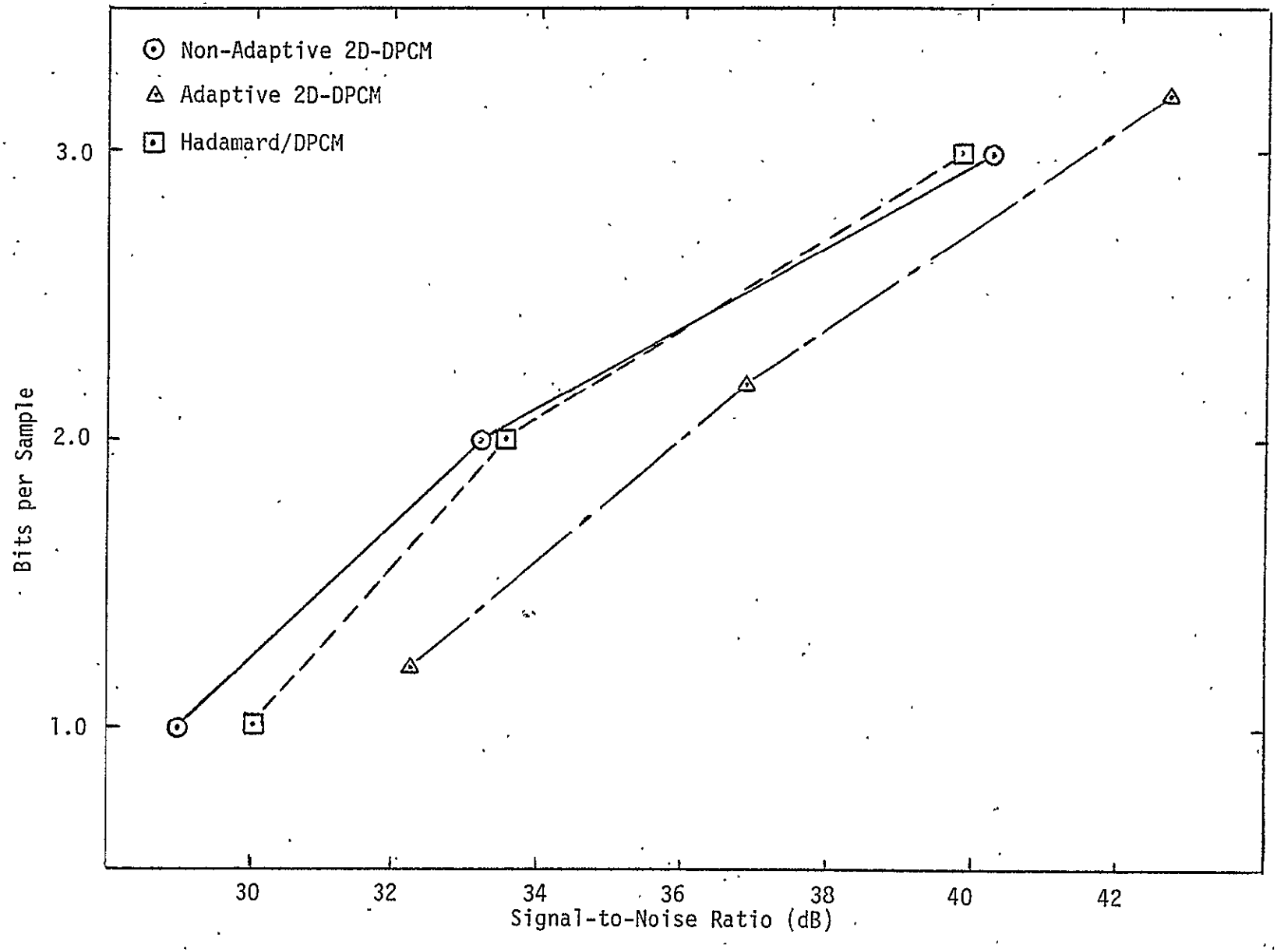


Figure 63. Performance of the Candidate Coding Techniques in Terms of Signal-to-Noise Ratio

is almost the same as the performance of the nonadaptive DPCM encoder. The difference in the performance of the adaptive and nonadaptive DPCM system is fairly small at 3 bits per sample. At 2 bits per sample, the difference is about 3 dB in signal-to-noise ratio and may be significant for some applications. On the other hand, the complexity of the adaptive DPCM encoder is much greater than the complexity of the nonadaptive DPCM encoder. For this reason and the fact that the lighting will be well controlled in the Shuttle, the nonadaptive DPCM encoder is selected over the adaptive DPCM system. The hybrid encoder was rejected because it requires more than twice the parts count of the nonadaptive DPCM encoder.

A block diagram of the proposed bandwidth compression technique for the field-sequential color TV is shown on Figure 64. The system starts its operation upon the receipt of the green field. It is encoded using a 2D-DPCM loop and is transmitted using 3 bits per sample. The green field is also filtered by a 3-point hanning filter, subsampled and stored in the field memory. This requires storing only 256 samples per line. Next, the R field is filtered, subsampled and combined with the G field to generate R-G for each line. R-G is encoded using a DPCM loop with a 4-level quantizer. Finally, the same procedure is used to generate and encode the B-G field. Since G is encoded in full resolution at 3 bits per sample, but R-G and B-G are encoded at 2 bits per sample with reduced resolution, rate buffering at the output of the encoder is required.

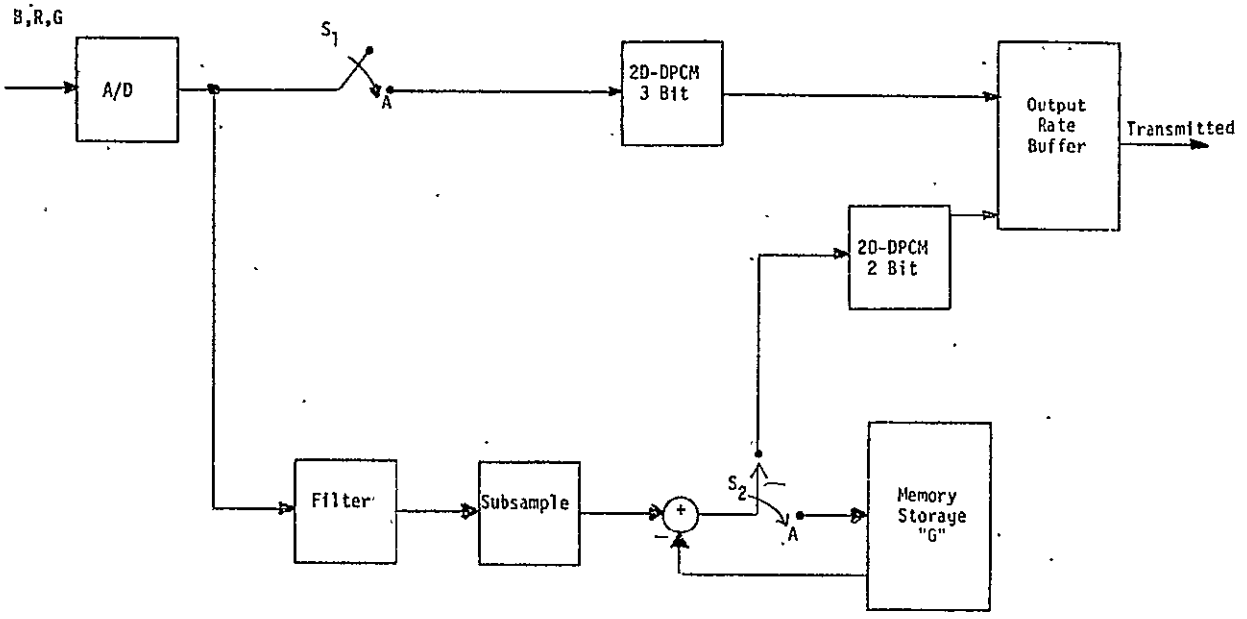


Figure 64. Block Diagram of Proposed Bandwidth Compression Technique for Field-Sequential Color TV

Figure 65 illustrates the operation of the precompression green field memory. Input video data from the A/D converter (8 MHz sampling rate) is applied to the Green Field Select Logic. Green field data is passed via the output select logic to the video compressor circuitry. Also, the green field, as well as the red and blue fields, is subsampled by taking every other sample in the subsampling logic. As a green field emerges from the subsampling logic, it is loaded into the serial green field memory by means of the recirculate switch. This switch is activated by the recirculate logic; thus, data is returned to the memory during the next two successive fields (red and blue). This insures that spatially related video samples differing only in successive field numbers are subtracted from each other, and this difference is fed to the DPCM compressor circuitry. During the initial memory load time, the subtractor output is disabled, and the green field data is fed to the compression logic. Due to the serial nature of the data, 16K CCD (charge coupled device) shift registers have been chosen for implementation of the serial green field memory.

The proposed Field-Sequential Color TV compression system requires a post-compression rate buffer memory. This requirement arises from the unequal sampling rates of the "Green" field and the R-G, B-G fields. In addition, the green field is encoded at 3 bits/sample while the R-G and B-G fields are encoded at 2 bits/sample. Therefore, the output rate changes from 23.6 Mbps for the green field to 7.8 Mbps for the remaining two fields. To maintain a constant output rate, a buffer memory is required to smooth the output rate to 13.1 Mbps.* Figure 65 shows the functional block diagram of the proposed rate buffer memory mechanization. Because of the differing input and output data rates involved, a random access memory (RAM) has been chosen. The output data from the video compressor is input to the demultiplexer for double buffering into the RAM memory. Double buffering is required so that no interruption of the input data will occur while memory loading takes place. Thus, each sample is shifted into the buffer register (a sample at a time) and loaded (rewritten) into the memory, 12 bits at a time. To prevent data loss, two buffer registers (double buffering) are required, one holding data for load,

*The actual rate will be slightly higher due to inclusion of synchronizing signals.

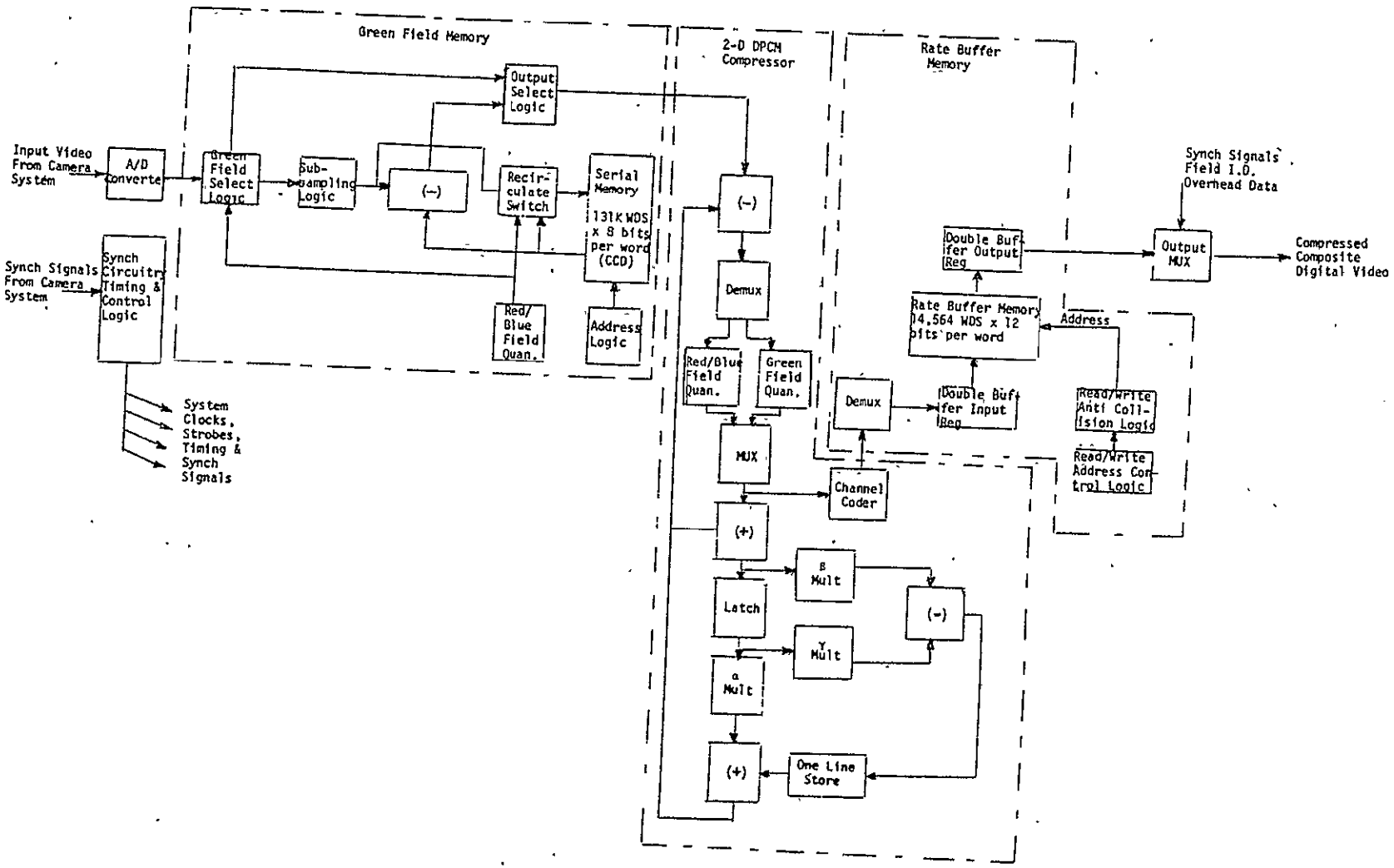


Figure 65. Functional Block Diagram of the Proposed Bandwidth Compression Technique for Field-Sequential Color TV

the other accumulating data. This technique slows the input and output memory data rates to where read/write collision* can be avoided by simple priority logic. The double buffer slows the memory input rate to a point where, in the event of a collision, there is sufficient time to allow a data read before the data write. This anti-collision control is provided by the Read/Write combiner and anti-collision logic. Data read from the memory is stored in the output data buffer register and shifted to the transmission link. Read, Write and Refresh address control are provided by the appropriate counters.

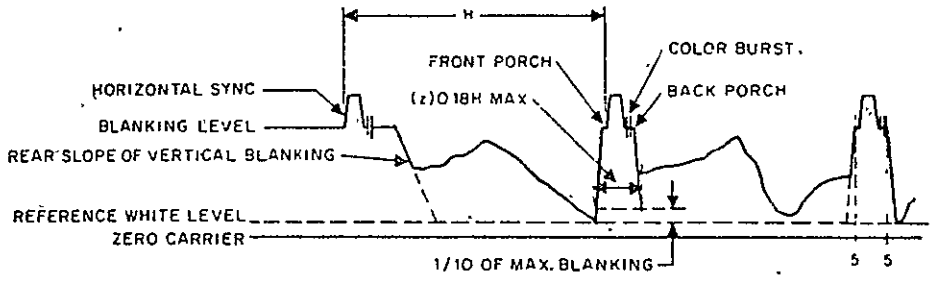
For the standard NTSC color TV system, TRW proposes a modification so that the analog illuminance (Y) and chromaticity signals (I, Q) are available in an unmodulated form. In the absence of such a modification, a comb filter is required to demodulate these signals. The proposed bandwidth compression technique for the NTSC Color TV uses a 2D-DPCM loop.

In the analog transmission of I and Q signals, they are low-pass filtered and multiplexed with the illuminance signals as shown in Figure 66. This technique is practical, since human vision is very insensitive to high frequency components of I and Q signals. Taking advantage of this property, TRW also proposes low-pass filtering of the I and Q signals. The passbands of these filters are about one-fifth of the illuminance signal. Maintaining a spatial resolution of 512 samples per line gives a spatial resolution of about 100 samples per line for I and Q signals. A further bandwidth compression can be achieved by alternating the transmission of the I and Q signals with each line of the illuminance signal. The receiver then restores the missing color component for each line by interpolating between the transmitted components for the previous and the future lines. The performance of such a system was evaluated at Bell Telephone Laboratories for the Color Picturephone [29]. There was no color degradation as a result of alternate transmission of the chromaticity signals.

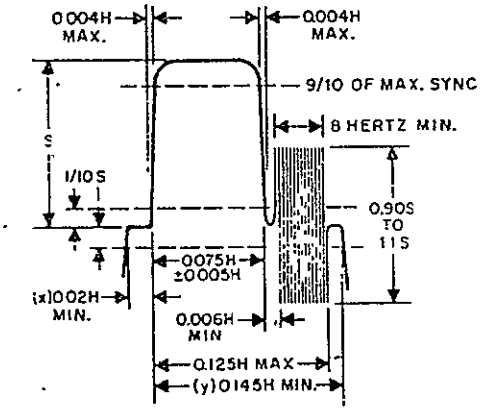
A block diagram of the proposed encoder is shown on Figure 67. The illuminance signal is sampled at a rate of 7.8 Mbps and is encoded by a 2D-DPCM system at 3 bits per sample. The sampling and transmission

---

*Read/write collision may be described as attempts to write into the memory at one location (address) simultaneously with an attempt to read from the memory at another location.



(A) Line Structure



(B) Detail Between 5-5 in (A)

Figure 66. NTSC Color Composite Signal Waveform

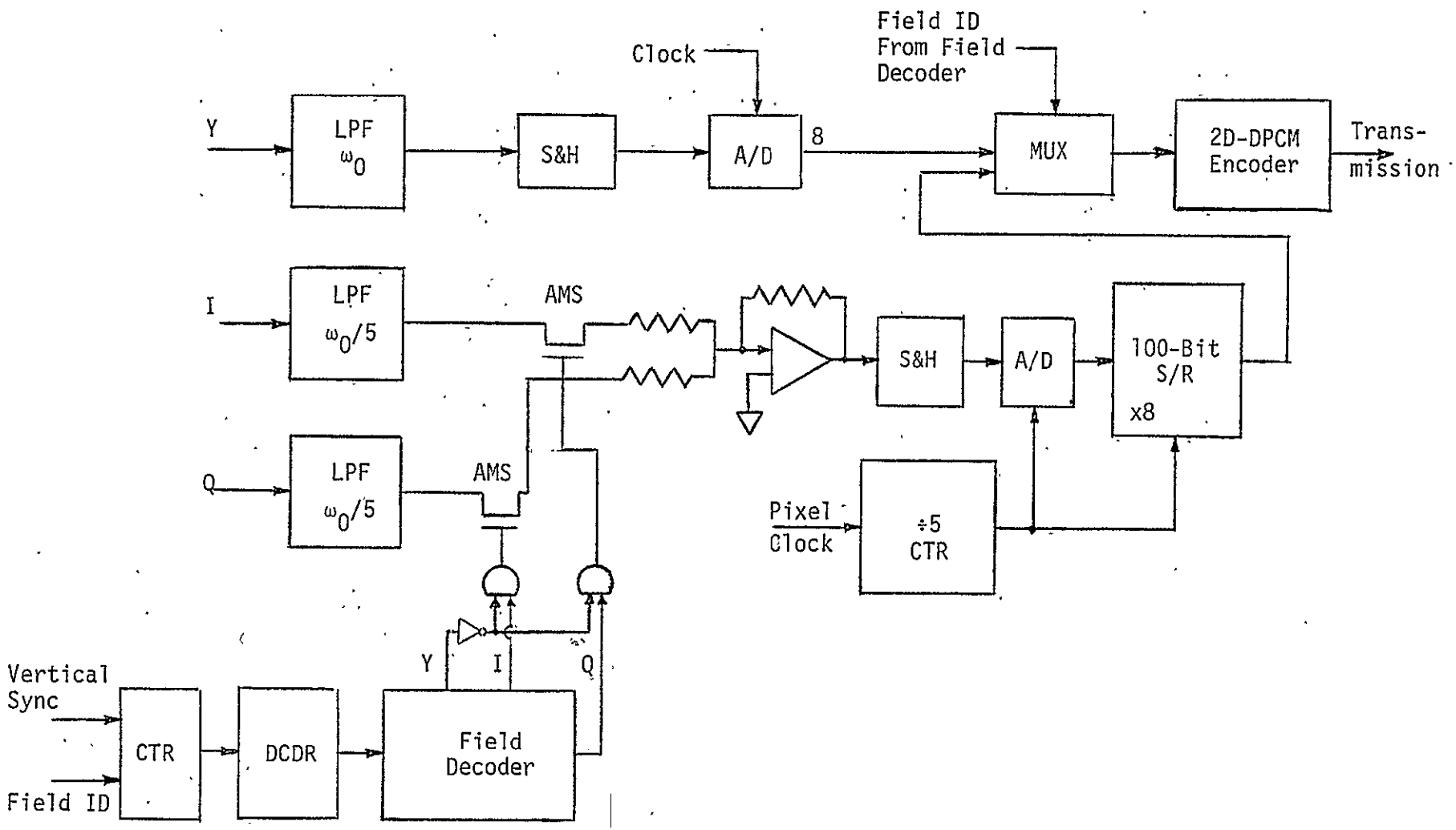


Figure 67. Block Diagram of Proposed Bandwidth Compression System for NTSC Color TV

of the illuminance signal takes place during the active period of the line scan. During the period of blanking, flyback and the interval which is normally used for analog transmission of the modulated color signal, TRW proposes to transmit either I or Q signals. The corresponding time intervals for commercial NTSC Color TV are shown in Figure 68. This arrangement gives sufficient time for transmission of 100 chrominance samples in the nonactive interval. Both illuminance and the chromaticity components can use the same 2D-DPCM encoder if additional memories are provided to store 100 samples of I and 100 samples of the Q signal for use in the DPCM predictor. This additional memory and the memory required to delay I or Q for the active duration of a line scan are the only components that need to be added to the 2D-DPCM encoder.

The performance of this system would be very similar to the performance of the proposed system for Field-Sequential Color TV since they both use the same 2D-DPCM encoder for the bandwidth compression of the illuminance signal. The bandwidth of this system, however, is higher. To maintain the same spatial resolution as that of Field-Sequential Color TV signal requires 28 Mbps.

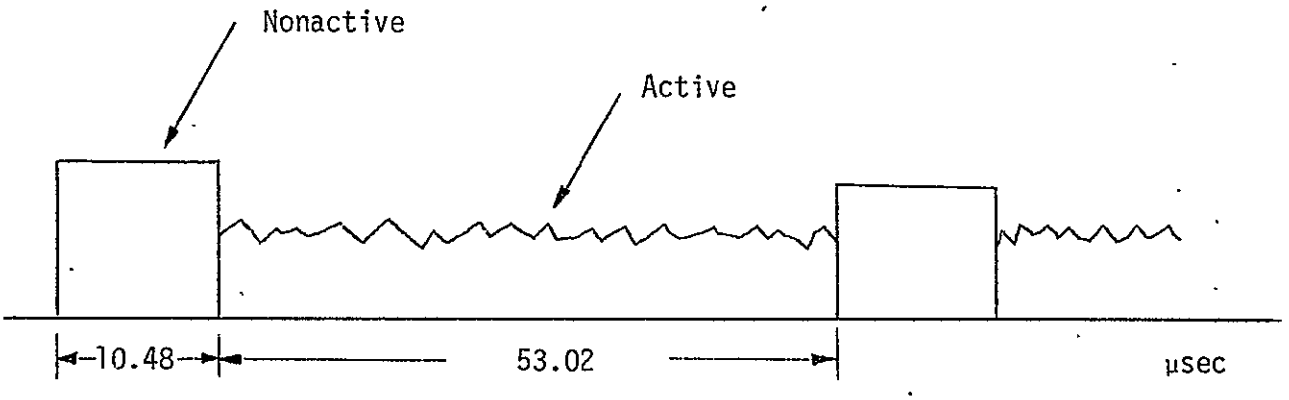


Figure 68. Active and Nonactive Portion of the Scan Line

8.0 SUMMARY AND CONCLUSIONS

The techniques for data compression of digital television has been presented in this report. From the basic PCM techniques of digitizing a signal, Section 2.0 presented entropy-preserving coding such as Shannon-Fano and Huffman coding, as well as polynomial data compression techniques such as the zero-order predictor, the zero-order interpolator, the first-order predictor and the first-order interpolator. Section 2.0 also presented predictive coding techniques such as DPCM, adaptive DPCM, delta modulation (DM), and adaptive delta modulation (ADM). An important form of ADM for digital television compression is tri-state ADM which was discussed in detail in terms of its implementation in Section 6.0. Transform techniques were presented in Section 3.0. Section 4.0 combined the results of Sections 2.0 and 3.0 to present two- and three-dimensional techniques for digital television compression. Thus, the data compression techniques for black-and-white television were presented in Sections 2.0 through 4.0. Section 5.0 extended these techniques to the NTSC color television system, as well as Field-Sequential color television used for spaceborne applications. A number of the techniques have been the subject of contracted efforts to develop a feasible digital television system. Section 7.0 assessed the practicality of each of the systems under development.

There are a number of conclusions that can be drawn from this study of digital television compression techniques. For black-and-white television, at moderate bit rates (15-30 Mbps) and where complexity is at a premium, the best choices are two-dimensional techniques. Three-dimensional techniques require frame storage which is a significant increase in complexity. The best choice for moderate bit rates at the present are probably two-dimensional DPCM as developed by TRW [18]. Two-dimensional delta modulation and tri-state adaptive delta modulation are very promising from preliminary results, but these techniques have not been fully proven. For very low bit rates (i.e., less than 10 Mbps), three-dimensional techniques are required. Transforms will probably be required, at least for the spatial coordinates. The best results to date have been obtained by NASA/Ames using a three-dimensional Walsh-Hadamard transform. However, the hardware complexity of the Ames system is formidable. The Linkabit

two-dimensional Walsh-Hadamard transform with frame differencing is probably the next best choice for low bit rates.

The best techniques for frame-sequential color are identical to the black-and-white schemes for moderate bit rates and small complexity. For low bit rates, where three-dimensional techniques are required, the frame-sequential color will probably have to be converted to standard NTSC color. For NTSC color, the best choice to date seems to be the TRW system where the analog illuminance (Y) and chromaticity signals (I, Q) are available in an unmodulated form. In the absence of such a modification, a comb filter is required to demodulate these signals. The proposed bandwidth compression technique for the NTSC Color TV uses a 2D-DPCM loop.

REFERENCES

1. A. Habibi and G. S. Robinson, "A Survey of Digital Picture Coding," Computer, May 1974, pp. 22-33.

2. T. S. Huang, "PCM Picture Transmission," Spectrum, Vol. 2, No. 12, December 1965, pp. 57-60.

3. F. W. Scoville and T. S. Huang, "The Subjective Effect of Spatial and Brightness Quantization in PCM Picture Transmission," NEREM Proceedings, November 1965, pp. 234-235.

4. R. A. Bruce, "Optimum Pre-emphasis and De-emphasis Networks for Transmission of Television by PCM," IEEE Trans. on Comm. Tech., Vol. COM-12, September 1964, pp. 91-96.

5. G. Hogg, "A Data Compression Primer," Report No. X-521-65-320, Goddard Space Flight Center, Greenbelt, Maryland, August 1965.

6. B. S. Atal and M. R. Schroeder, "Predictive Coding of Speech Signals," Bell System Technical Journal, Vol. 48, No. 8, October 1970, pp. 1973-1986.

7. K. Virupaksha and J. B. O'Neal, Jr., "Entropy Coded Adaptive Differential PCM for Speech," Air Force Office of Scientific Research Contract F44620-69-C-0033, Report No. AFOSR-74-0623TR (AD-778952), January 4, 1974.

8. P. J. Ready and D. J. Spencer, "Block Adaptive DPCM Transmission of Images," Proceedings of NTC'75, New Orleans, Louisiana, December 1975, pp. 22-10 - 22-17.

9. J. E. Abate, "Linear and Adaptive Delta Modulation," Proceedings of the IEEE, Vol. 55, March 1967, pp. 298-308.

10. M. R. Winkler, "Pictorial Transmission with HIDM," IEEE International Convention Record, Pt. 1, 1965, pp. 285-290.

11. J. W. Cooley and J. W. Tukey, "An Algorithm for Machine Computation of Complex Fourier Series," Math. Comput., Vol. 19, April 1965, pp. 297-301.

12. N. Ahmed, T. Natarayan, and K. R. Rao, "Discrete Cosine Transform," IEEE Trans. on Computers, Vol. C-23, January 1974, pp. 90-93.

13. H. Enomoto and K. Shibata, "Orthogonal Transform Coding System for Television Signals," IEEE Trans. on Electromagn. Compat., Vol. EMC-13, August 1971, pp. 11-17.

14. W. K. Pratt, W.-H. Chen, and L. R. Welch, "Slant Transform Image Coding," IEEE Trans. on Commun., Vol. COM-22, August 1974, pp. 1075-1093.

15. A. Habibi, "Comparison of nth-Order DPCM Encoder with Linear Transformations and Block Quantization Techniques," IEEE Trans. on Comm. Tech., Vol. COM-19, No. 6, Part 1, December 1971, pp. 948-956.

16. A. Habibi, "Hybrid Encoding of Pictorial Data," IEEE Trans. on Comm., Vol. 22, No. 5, May 1974, pp. 614-624.

17. P. A. Wintz, "Transform Picture Coding," IEEE Proceedings, Vol. 60, No. 7, July 1972, pp. 809-820.

18. A. Habibi and G. Fultz, "Design of a Digital Compression Technique for Shuttle Television," Final Report, Contract NAS 9-14852, September 1970.

19. A. G. Tescher, H. C. Andrews, and A. Habibi, "Adaptive Phase Coding in Two- and Three-Dimensional Fourier and Walsh Image Compression," 1974 Picture Coding Symposium, Goslar, Germany, August 26-28, 1974.

20. B. G. Haskell, F. W. Mounts, and J. C. Candy, "Interframe Coding of Videophone Pictures," IEEE Proceedings, Vol. 60, No. 7, July 1972, pp. 792-800.

21. F. W. Mounts, "Frame-to-Frame Digital Processing of TV Pictures to Remove Redundancy," Picture Bandwidth Compression, Gordon and Breach, New York, 1972, pp. 653-672.

22. J. D. Drummond, "Color-TV Wheel Takes a Spin in Space," Electronics, July 7, 1969, pp. 114-117.

23. Linkabit Corporation, "Digital TV Processing System," Final Report Contract NAS 9-14561, November 1975.

24. D. G. Fink. Television Engineering Handbook. New York: McGraw-Hill, Inc., 1957, p. 9-7.

25. L. Weiss, I. M. Paz, and A. L. Schilling, "Video Encoding Using an Adaptive Digital Delta Modulator with Overshoot Suppression," IEEE Trans. on Comm., Vol. COM-23, No. 9, September 1975, pp. 905-920.

26. F. W. Mounts, A. N. Netravali, and B. Prasada, "Subjectively Optimum Quantizers for Real-Time Hadamard Coding of Pictures," Proceedings of NTC'74, p. 83.

27. J. A. Roese, G. S. Robinson, W. K. Pratt, and A. Habibi, "Interframe Transform Coding and Predictive Coding Methods," Proceedings of ICC'75, June 1975.

28. A. J. Landau and D. Slepian, "Some Computer Experiments in Picture Processing for Bandwidth Reduction," Bell System Technical Journal, July-August 1971.

29. T. O. Limb, C. B. Rubinstein, and K. A. Walsh, "Digital Coding of Color Picturephone Signals by Element-Differential Quantization," IEEE Trans. on Comm. Tech., Vol. COM-19, No. 6, Pt. 1, December 1971, pp. 992-1006.

BEST SELLERS

FROM NATIONAL TECHNICAL INFORMATION SERVICE

NTIS

Development of Pre-Mining and Reclamation Plan Rationale for Surface Coal Mines
PB-258 041/SET/ PAT 590 p. PC$22.50/MF$7.00

Manual of Respiratory Protection Against Airborne Radioactive Materials
PB-258 052/PAT 147 p. PC$6.00/MF$3.00

Design and Construction of a Residential Solar Heating and Cooling System
PB-237 042/ PAT 233 p. PC$8.00/MF$3.00

1976 Energy Fact Book
ADA-029 331/ PAT 195 p. PC$12.50/MF$12.50

Impacts of Construction Activities in Wetlands of the United States
PB-256 674/ PAT 426 p. PC$11.75/MF$3.00

Standardized Development of Computer Software. Part I: Methods
N76-30-849/ PAT 389 p. PC$10.75/MF$3.00

A Methodology for Producing Reliable Software, Volume I.
N76-29-945/ PAT 228 p. PC$8.00/MF$3.00

Flow and Gas Sampling Manual
PB-258 080/ PAT 102 p. PC$5.50/MF$3.00

Solar Heating and Cooling in Buildings: Methods of Economic Evaluation
COM-75-11070/ PAT 48 p. PC$4.00/MF$3.00

Data Base Directions. The Next Steps
PB-258 103/ PAT 177 p. PC$7.50/MF$3.00

Comparative Study of Various Text Editors and Formatting Systems
ADA-029 050/ PAT 93 p. PC Not Available/MF$3.00

Explaining Energy: A Manual of Non-Style for the Energy Outsider Who Wants In
LBL-4458/ PAT 78 p. PC$4.50/MF$3.00

A Survey of State Legislation Relating to Solar Energy
PB-258 235/ PAT 166 p. PC$6.75/MF$3.00

Cost Estimating Handbook for Transfer, Shredding and Sanitary Landfilling of Solid Waste
PB-256 444/ PAT 85 p. PC$5.00/MF$3.00

Coal Liquefaction Design Practices Manual
PB-257 541/ PAT 372 p. PC$10.50/MF$3.00

HOW TO ORDER

When you indicate the method of payment, please note if a purchase order is not accompanied by payment, you will be billed an additional $5.00 *ship and bill* charge. And please include the card expiration date when using American Express.

Normal delivery time takes three to five weeks. It is vital that you order by number or your order will be manually filled, insuring a delay. You can opt for *airmail delivery* for $2.00 North American continent; $3.00 outside North American continent charge per item. Just check the *Airmail Service* box. If you're really pressed for time, call the NTIS Rush Handling Service (703) 557-4700. For a $10.00 charge per item, your order will be airmailed within 48 hours. Or, you can pick up your order in the Washington Information Center & Bookstore or at our Springfield Operations Center within 24 hours for a $6.00 per item charge.

You may also place your order by telephone or if you have an NTIS Deposit Account or an American Express card order through TELEX. The order desk number is (703) 557-4650 and the TELEX number is 89-9405.

Thank you for your interest in NTIS. We appreciate your order.

METHOD OF PAYMENT

- ☐ Charge my NTIS deposit account no. _____
- ☐ Purchase order no. _____
- ☐ Check enclosed for $_____
- ☐ Bill me. Add $5.00 per order and sign below. (Not available outside North American continent.)
- ☐ Charge to my American Express Card account number

Card expiration date _____

Signature _____

☐ Airmail Services requested

NAME _____

ADDRESS _____

CITY, STATE, ZIP _____

Item Number	Quantity		Unit Price*	Total Price
	Paper Copy (PC)	Microfiche (MF)		

All prices subject to change. The prices above are accurate as of 4/77

Foreign Prices on Request

Sub Total	
Additional Charge	
Enter Grand Total	

Clip and mail to:

NTIS

National Technical Information Service
U.S. DEPARTMENT OF COMMERCE
Springfield, Va. 22161
(703) 557-4650 TELEX 89-9405