General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA 94035

AMERICAN REVOLUTION BICENTENNIAL 1776-1976

REPLY TO ATTN OF: NAS 2-8297
AT:241-12

August 18, 1975

NASA Representative
Scientific & Technical Information Facility
P.O. Box 8757
Baltimore/Washington International Airport,
Maryland 21240

Subject: Transmittal of Contractor Report: "Study of Efficient Video Compression Algorithms for Space Shuttle Applications," dated June 11, 1975 by Dr. Zung Poo, Linkabit Corporation, 10453 Roselle Street, San Diego, California.

Reference: Program Code 645-25-01-01-00

The subject report prepared under Contract NAS 2-8297 has been reviewed at Ames and is recommended for release in STAR as CR 137708.

Loretta Vice

(NASA-CR-137708) STUDY OF EFFICIENT VIDEO COMPRESSION ALGORITHMS FOR SPACE SHUTTLE APPLICATIONS Final Report (Linkabit Corp., Los Angeles, Calif.) 66 p HC $4.25 CSCL 17B
N75-29294
Unclas
G3/32 33120

LINKABIT CORPORATION

10453 ROSELLE STREET
UNIVERSITY INDUSTRIAL PARK
SAN DIEGO, CALIFORNIA 92121

AUG 1975
RECEIVED
NASA STI FACILITY
INPUT BRANCH

NASA CR 137-708

STUDY OF EFFICIENT VIDEO COMPRESSION ALGORITHMS FOR SPACE SHUTTLE APPLICATIONS

FINAL REPORT

by

Dr. Zung Poo

11 June 1975

Prepared Under Contract No. NAS2-8297

by

LINKABIT CORPORATION
10453 Roselle Street
San Diego, CA 92121

for

AMES RESEARCH CENTER
National Aeronautics and Space Administration

TABLE OF CONTENTS

Section		Page
1.0	Introduction	1
2.0	Video Source Coding	4
3.0	Results of Study of Two-Dimensional Video Data Compression	9
3.1	Logarithmic Response of Human Eyes	17
3.2	Quantization of the d.c. Component of the Hadamard Transformation	21
3.3	Quantization of non-d.c. Hadamard Components	24
3.4	Selection of Quantization Levels	26
3.5	Further Possible Improvement in Coding Efficiency	29
4.0	Buffer-Free Frame-to-Frame Data Compression	35
4.1	Quantization of Buffer-Free Frame-to-Frame Differencing Technique	37
4.2	Compression Experiments	38
4.3	Requirement for Hardware Implementation	42
4.3.1	Hardware Estimations	47
4.3.1.1	Video Source Encoder	47
4.3.1.2	Video Source Decoder	49
	References	51
	Appendix A	52
	Appendix B	55

TABLE OF FIGURES

Figure		Page
1.	LIM Video Controller Functional Block Diagram	2
2.	Typical Picture Communication Block Diagram	5
3.	Sample Lattice of a Frame of Digitized Video Signal	11
4.	Typical Visual Response	18
5.	Typical Response of Logarithmic DPCM to Step Error	23
6.	Hadamard Component Designation	28
7.	Video Source Encoder	43
8.	Video Source Decoder	44
A1.	Illustration of Inequality (3.14) in 1 x 2 Case	49

1.0 Introduction

This report presents the results of a study on video data compression techniques applicable to space flight communication. This study is exclusively directed towards monochrome (i.e. black and white) picture communication with special emphasis on feasibility of hardware implementation. The primary factors for such a communication system in space flight applications are:

1) picture quality,
2) system reliability,
3) power consumption,
4) hardware weight.

In terms of hardware implementation, these are directly related to hardware complexity, effectiveness of the hardware algorithm, immunity of the source code to channel noise, and data transmission rate (or transmission bandwidth). This report will recommend a system and summarize its hardware requirement. In addition, this report will provide sufficient data on various parameters involved.

Simulations of the study were performed on the improved LIM Video Controller. The LIM Video Controller is computer-controlled by the META-4 CPU. The functional block diagram of the LIM Video Controller is illustrated in Figure 1. The LIM Video Controller processes video signals recorded

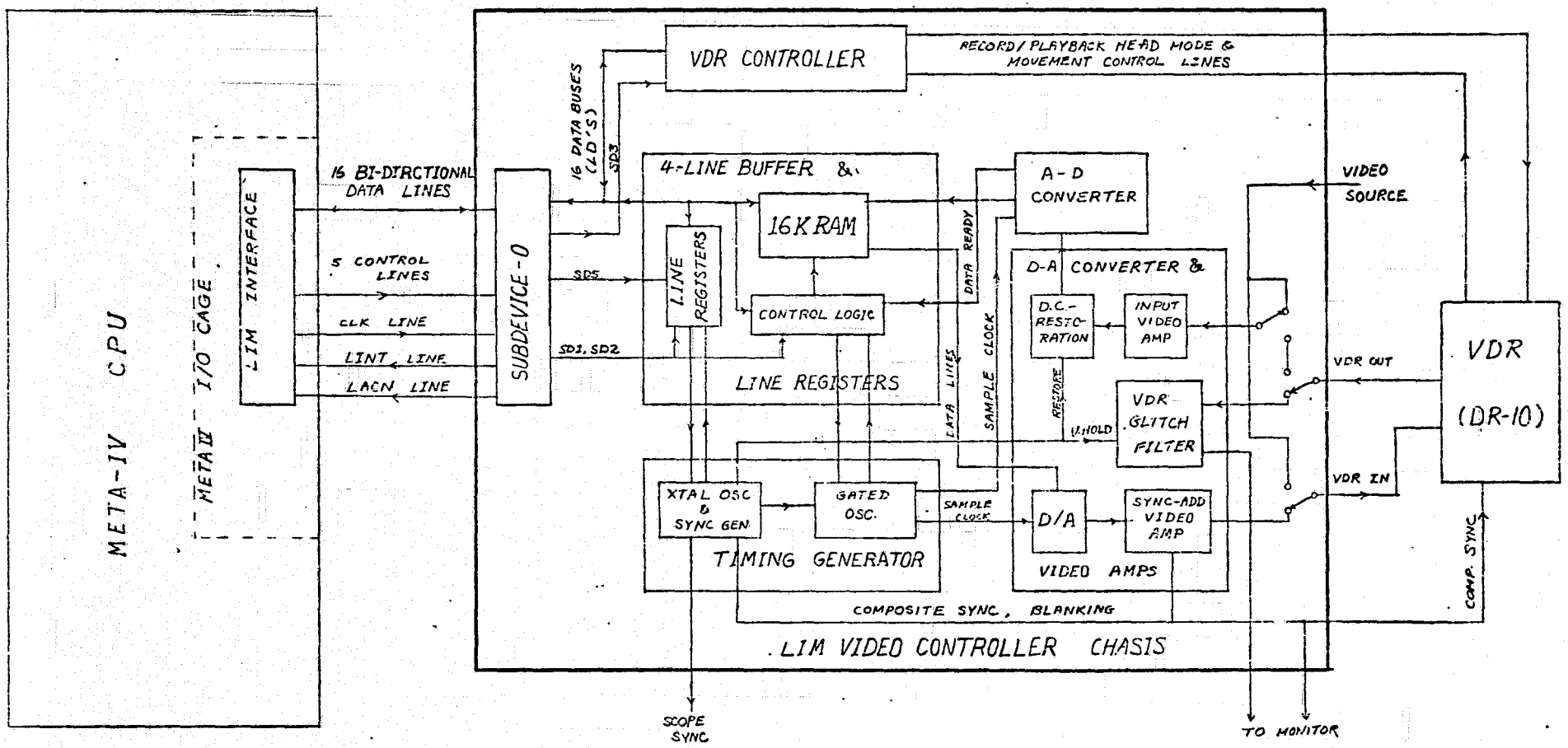

Figure 1. LIM VIDEO CONTROLLER FUNCTIONAL BLOCK DIAGRAM

CHANGE EFF	ECO NO.	REVISIONS			
		LTR	DESCRIPTION	DATE	APPROVED

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES: ANGLES ± .XX± .XXX±
SURFACES ✓
MATERIAL
SPEC
CONTRACT NO.
DRAWN BY DATE
CHECK
ENGINEER
PROJECT

LINKABIT CORPORATION
TITLE
SIZE C
CODE IDENT NO.
DWG NO.
SCALE
WEIGHT
SHEET OF

on the Ampex Video Recorder, Model DR-10. This is done by first A-to-D converting the video signals (in groups of four horizontal lines), transporting the digital data to the META-4 main storage for processing, then the processed signal is reconstructed by D-to-A conversion and finally recorded on the video disc recorder for visual display. The DR-10 has 600 tracks (one frame per track). Half of the disc allocation (i.e. 300 frames of video signals) is used for storage of reference signals. The remaining half is used for recording the processed signals. This corresponds to 10 seconds of real-time video signals. This seems sufficient for demonstration purposes for the effectiveness of the algorithm under study. By preserving the pre-recorded reference signals on the video disc recorder, effectiveness of various schemes can be compared fairly by recording the processed signals on a video tape recorder. The improved LIM Video Controller and its new supporting software enable 300 frames of video signals to be processed in approximately 4-1/2 hours. Each processed frame has a superimposed title and reference frame number for ease of identification. Noise degradation due to simulation is minimal. However, the video tape recorder (Sony model) has limited bandwidth and causes certain noise effects. For a detailed description of the LIM Video Controller, see the LIM Video Controller Operations Manual..

2.0 Video Source Coding

Video data compression using source coding has been under investigation for many years. It has been observed that although any normal scene recognizable to human eyes contains large amount of information in terms of shapes, details, edges, spots, and grey level variations, statistical correlations of video signals within small picture area and time difference are high. These satistical relations are referred to as spatial and time correlations, respectively. Spatial correlation occurs in every frame of a recognizable scene. It merely indicates that only relatively few among all possible producible pictures can be interpreted by the human eyes as recognizable pictures rather than just noise. It should be emphasized that recognizibility of a picture varies from person to person; it is most likely the ability to relate the contents of the picture to certain objects in the past history of a particular person. Time correlation applies only to scenes that involves object movements. Normally, the human eyes can comprehend (without repeating the scene) motion only when the rate of displacement of certain objects in the scene are very small. Transitions of objects with a large rate of displacements normally cannot be distinguished by human eyes.

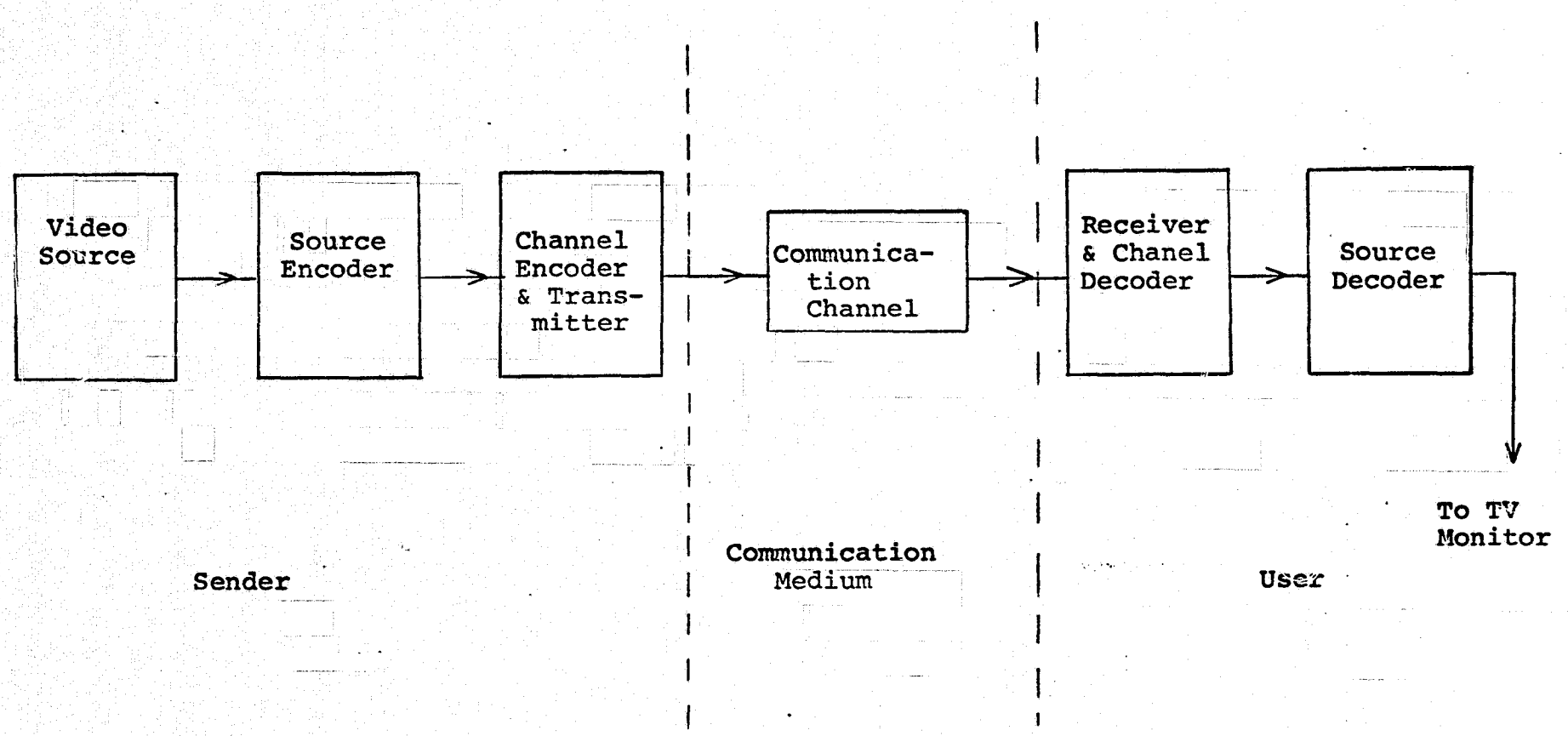

Figure 2. Typical Picture Communication Block Diagram

Most video data compression techniques are based upon the above facts. One of the methods, originally investigated by Landau and Slepian [2], uses spatial statistical correlation exclusively. The picture is first partitioned into small regions of subpictures. Statistical data reduction is performed on each subpicture by considering each subpicture as an independent random vector. Coordinate transformation is applied to the random vector. The objectives of the coordinate transformation is to diagonalize the original covariance matrix of the subpicture and to produce an orthonormal basis similar to the Karhunen-Loeve procedure. Bit rate reduction is obtained by discarding or quantizing with fewer information bits those components that have lower statistical variances. The transformation used is the Hadamard transformation, where the basis vectors corresponds to the row vectors of a Hadamard matrix. This method has the advantage of simple and fast hardware implementation, short delay between the real-time and the processed pictures, and the coding errors due to channel noise are confined to subpictures (i.e. high coding reliability). Compression ratio of 4 to 1 (assuming the original video signals are linearly digitized by an 8-bit A-to-D converter) is achievable without substantial deterioration in picture quality. However, this method does not use the time statistical correlation between adjacent frames of the video sequence.

Another relatively simple method is the frame-to-frame differencing, variable-length coding technique [3]. This method only utilizes the time statistical correlation in recognizable video sequences. Here, a reference frame of full information is transmitted. For the subsequent frames, the video signals are compared with their corresponding video signals of the preceeding frame, and only those differences that exceeded certain pre-chosen threshold are transmitted. At the receiving side, the reference is first reconstructed, the subsequent frames are updated by the information received. This method basically uses the fact that most recognizable sequences have a large proportion of stationary objects; thus, the amount of information changed from frame to frame is rather small. Source coding using this method requires that a substantial portion of the code be allocated for position markers to indicate where in the picture the changes take place. Moreover, a rate buffer is required to achieve fixed rate transmission. Furthermore, this method is very sensitive to channel noise; a decoding error in one frame causes errors in subsequent frames. Thus, to improve picture reliability, reference frames must be transmitted every so often.

In a previous LINKABIT video study report [4], LINKABIT provided a buffer-free technique that utilizes both the spatial and time statistical correlations of recognizable scenes.

This method is an essential combination of the two methods described. Like the Landau and Slepian method, video signals are first transformed into Hadamard coordinates. Bit rate reduction is first achieved utilizing the spatial statistical correlation; a reference frame is transmitted in this manner. For the subsequent 3 frames, the Hadamard components of the new picture are compared with the corresponding components of the reference frame and only the differences of a few selected components (those that have highest statistical variances) are quantized and transmitted. The process is repeated every four frames. This method seems capable of reproducing a scene with a compression ratio of 8 to 1 while retaining recognizability of the scene. It is the sole objective of this report to explore improvements in this technique. Due to its relatively simple hardware implementation and partial frame storage (instead of a full frame storage required in method described above), this method seems to be most promising for space flight application. (Although other image coding methods are available, such a 2-dimension Fourier transformation and other related techniques, their computational complexities and bulk storage required for past frame information limit their usage to ground instruments.)

3.0 Results of Study of Two-Dimensional Video Data Compression

Since the technique suggested in this report uses the Landau and Slepian method for spatial statistical data reduction, it is essential to examine this method and explore possible improvements. We shall begin with a review of the Landau and Slepian two-dimensional transformation technique.

We assume the video signal source being processed is regular commercial NTSC TV. Each frame of the video signal consists of two fields interlaced with each other. The video signals are digitized by an A-to-D converter. The A-to-D converter must sample at a frequency above the Nyquist rate of the desired bandwidth, and a sufficient number of bits of information per sample is required to ensure smooth video reproduction. For monochrome TV signals, sampling at 512 samples per horizontal line and 8-bit grey level resolution seem sufficient to reproduce reasonable quality picture without false image contouring.

Each frame of video signal consists of 525 horizontal lines. Among these, 45 horizontal lines are used for generation of vertical sync pulses and are blanked. Thus, each frame contains, at most, 480 lines of visible information. When digitized accordingly, we can view each frame of digitized video signals as a lattice of 480 x 512 sample points, x_{ij}, ($i = 1, 2, \ldots, 480$; $j = 1, 2, \ldots, 512$). Each x_{ij} has integer representation value between 0 and 255. In other words, each frame of digitized video signal can be represented by a vector in an Euclidean space of dimension 480 x 512. This representation is illustrated in Figure 3.

Since spatial statistical correlations between sampling points are effective only for neighboring points, it is desirable to partition the picture into subpictures where spatial statistical correlations within the subpicture are highest. One method of achieving this is to partition the picture into rectangles of size m x n (m vertically and n horizontally), where m is a divisor of 480 and n is a divisor of 512. For commercial TV signals, since fields overlap each other, the subpictures can be formed within each field. In this way, subpictures of field 1 overlap with subpictures of field 2. An attempt to form subpictures from both fields may not be advantageous, since a delay of at least one field time is required between the real-time and

ORIGINAL PAGE IS
OF POOR QUALITY

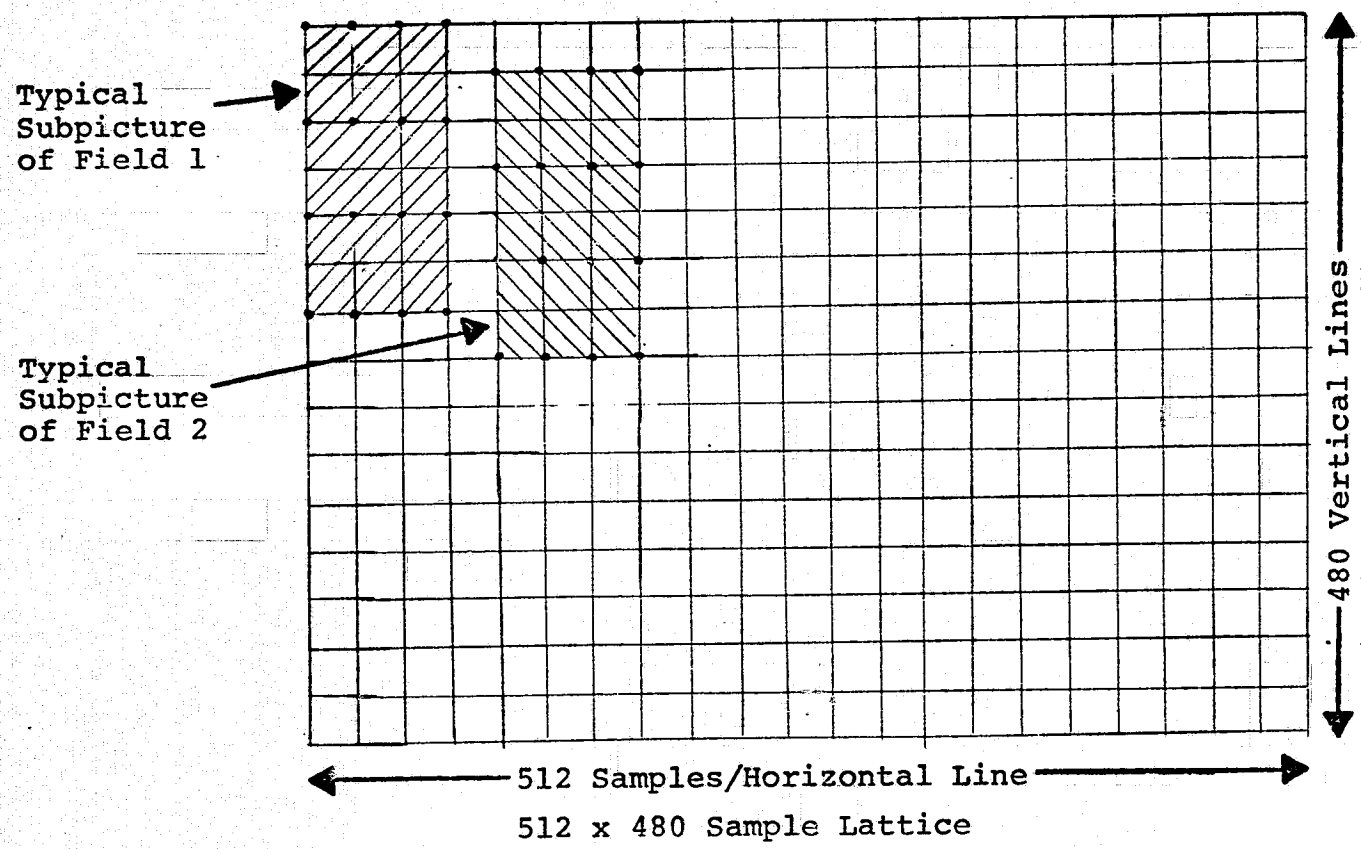

Figure 3. Sample Lattice of a Frame of Digitized Video Signal

processed video signals. Furthermore, field 1 and field 2 are in reality two different pictures at two different instances; consequently, for scenes with a lot of object movements, the spatial statistical correlation between neighbor points with different fields may not be effective.

Choice of m and n are normally determined by considering the following factors:

1) Higher data compression ratio can be achieved with larger m and n.

2) Computational complexity (consequently, hardware implementation) and time delay between real-time and processed signals generally increase with m and n.

3) For subpictures of too large m and n, the spatial satistical correlations between furtherest sample points within the subpicture diminish; thus, data compression ratio may not be further improved by increasing m and n.

Experiments have shown that subpictures of size 4 x 4 seem to be most efficient. Due to the asymmetry resulted from the effect of field interlacing, subpictures of size 4 x 8 should be also a reasonable choice. All experiments in this report are based upon subpictures of size 4 x 4.

Each subpicture is considered as an m x n random vector independent of other subpictures. Symbolically, each subpicture of video can be represented as:

$$Y = \begin{pmatrix} x_{11} & \cdots & x_{in} \\ x_{21} & \cdots & x_{2n} \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \qquad \begin{array}{l} 0 \leq x_{ij} \leq 255 \\ x_{ij} \text{ - integer } \ldots \end{array} \tag{3.1}$$

Each x_{ij} represents the digitized video signal at coordinate (i,j) within the subpicture. The range of this vector is the lattice of $(256)^{mn}$ integral points lying within the mn-dimensional cube of size 255. Due to the spatial statistical correlations of recognizable pictures, not all of the vectors in the range are comprehensible to the human eye. By considering the set of subpictures extracted from recognizable pictures, statistical data reduction in the sense of least mean square error generally can be obtained by using Karhunen-Loeve procedure. This procedure requires generation of the orthonormal basis which diagonalizes the co-

variance matrix of the ensemble of subpictures, (3.1), of recognizable pictures. However, the least mean square error criterion is not generally suitable as a measure of visual fidelity. Other orthonormal bases were sought. One of the intuitive choices, which was judged to be superior to the Karhunen-Loeve basis, is the Hadamard basis. The Hadamard basis are vectors that are the rows of a Hadamard matrix.

An n x n matrix, H, of integer entries is Hadamard if

$$H \cdot H^T = nI$$

Where H^T is the transpose of H. I is the identity matrix of n-dimensional vector space. A Hadamard matrix of order 2^k, k = 2, 3, ..., can be obtained recursively as the tensor product of Hadamard matrix of order 2, H_2:

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad . \; . \; . \tag{3.2}$$

Therefore, a subpicture of size $m = 2^{k_1}$ and $n = 2^{k_2}$ can be coordinate transformed using a Hadamard basis; the basis vectors are obtained from the row vectors of the Hadamard matrix of order $2^{k_1+k_2}$ (which can be obtained by the $k_1 + k_2$ tensor product of H_2). All row vectors of any Hadamard matrix have component values +1 or -1. Thus, the orthonormal basis can be obtained by dividing each row vector by

the constant:

$$\sqrt{mn} = \sqrt{2^{k_1+k_2}} \quad . . . \tag{3.3}$$

The first basis vector thus formed is of the form:

$$\hat{b}_{11} = \frac{1}{\sqrt{nm}} \begin{pmatrix} 1 & 1 & . & . & . & 1 \\ 1 & 1 & . & . & . & 1 \\ . & & & & & \\ . & & & & & \\ 1 & 1 & . & . & . & 1 \end{pmatrix} \quad . . . \tag{3.4}$$

$$\uparrow$$
$$m \times n$$

The remaining basis vectors are all of the form $1/\sqrt{mn}$ multiplied by a vector in which half of the components have value +1 and the remaining half have component value -1. Let us denote the Hadamard basis vector by

$$\hat{b}_{ij} \qquad i = 1, 2, . . . m$$
$$j = 1, 2, . . . n$$

Then each subpicture of the video signal can be expressed as

$$Y = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} \cdot \hat{b}_{ij}, \quad . . . \tag{3.5}$$

where

$$C_{ij} = Y \cdot \hat{b}_{ij} \quad . . . \tag{3.6}$$

C_{ij} is the projection of Y into basis vector $\hat{b}_{ij}$. In particular, the first component, C_{11}, has the following expression:

$$C_{11} = \frac{1}{\sqrt{mn}} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \tag{3.7}$$

This component is also known as the d.c. component of the subpicture, for reason that it is a constant multiple of the sum of individual sample video signals. The remaining components, C_{ij}, have the form $1/\sqrt{mn}$ multipled by the difference of sums of half of the x_{ij}'s. Thus, when transformed, the Hadamard component has the following range:

$$0 \leq C_{11} \leq 255\sqrt{mn}$$

and (3.8)

$$-255\sqrt{nm}/2 \leq C_{ij} \leq +255\sqrt{mn}/2 \quad \text{(to the nearest integer)}$$

$$\text{for } i \neq 1 \text{ and } j \neq 1$$

Given a vector, Z, in the Hadamard basis, the corresponding subpicture, Y, can be obtained by inverse transformation:

$$Y' = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ x_{m1} & \cdots & xmn \end{pmatrix} \tag{3.9}$$

when $x_{ij} = Z \cdot \hat{b}_{ij}$

Data rate reduction is achieved by statistical analysis on the covariance matrix of the transformed Hadamard components on the ensemble of recognizable pictures. Statistical analysis of ordinary recognizable pictures, using a 4 x 4 subpicture size, reveals that the variance of the first component, C_{11}, is the highest and is in excess of 10 to 1 in ratio to the next highest

variance [2] (statistical analyses of this kind are widely available in the literature. LINKABIT has not attempted to duplicate such analysis). The remaining components have relatively small variances. Coarsily quantizing the values of these components close to their mean values, and reconstructing Y (by applying inverse transformation 3.9) using these quantized values, should result in a very small mean square error. This approach is theoretically sound, yet the human visual system does not behave quite that way. Approximations using various quantizations are expected to play a major role in the quality of the picture. Most of the research done so far in this field has been by experimentation. In this report, we shall follow psychovisual reasoning more closely in researching for an optimal choice of quantization for the Hadamard components. The following is a list of phychovisual rationales and findings:

3.1 Logarithmic Response of Human Eyes

It is well-known that the eye, like other sense organs, behaves logarithmically with respect to their inputs. Given a normal recognizable picture, the eyes are generally insensitive to the relative brightness of the picture. However, the eyes are capable of detecting minute amplitude (brightness) changes between adjacent regions. The sensitivity for detecting brightness between adjacent regions likewise behaves

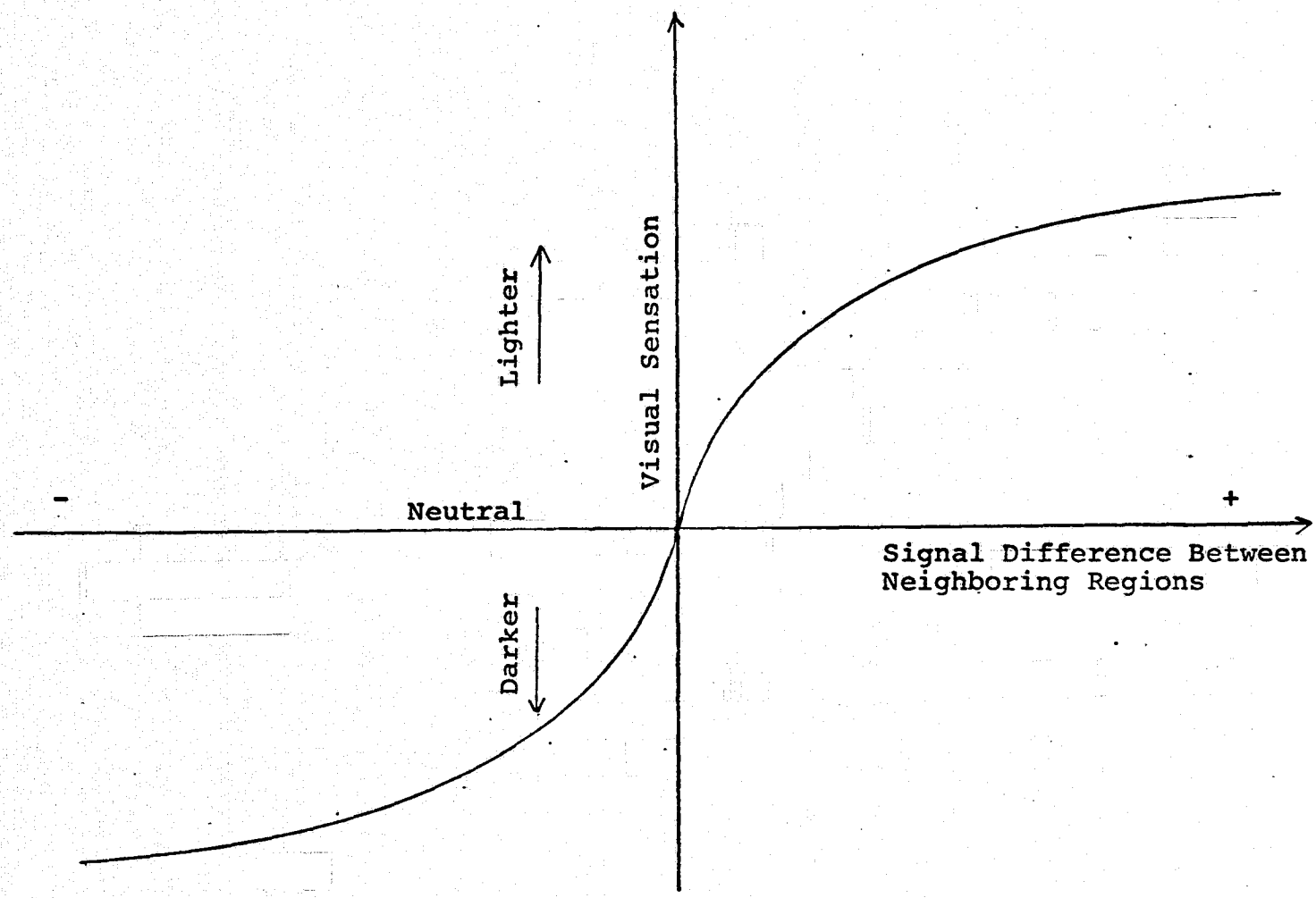

Figure 4. Typical Response of Visual Sensation

logarithmically. The logarithmic response explains the reason why human eyes are extremely sensitive to false image contouring and graininess of the picture. Typical response of human eyes with respect to the difference of video signals of adjacent regions can be expressed as:

$$\mathrm{Sgn}(x) \cdot A.\log\,(B\,|X| + 1) \tag{3.10}$$

where A and B are positive real constants, and x is the difference of the video signals between neighboring regions. (Positive sign implies brighter signal and negative sign implies darker signal.) Based upon this reasoning, the quantization cutpoints for a given number of bits of information should be chosen accordingly, as the inverse of (3.10).

Let the number of quantized levels be $N_1 + N_2 + 1$, which correspond to the integer set:

$$-N_1,\ -N_1+1,\ \ldots,\ -2,\ -1,\ 0,\ 1,\ 2,\ \ldots,$$

$$N_2-1,\ N_2.$$

Then the quantization levels should be chosen as:

$$-A_1[e^{-B_1 k} - 1] \text{ for } k = 0, -1, \ldots, -N_1$$

and (3.11)

$$+A_2[e^{B_2 k} - 1] \text{ for } k = 0, 1, \ldots, N_2$$

In other words the quantization error is allowed to increase exponentially if the difference of adjacent components is large. The formulation (3.11) also allows "zero" representation. Quantization without zero representation has the disadvantage of introducing unwanted "sand paper" effect, which is the inability of the coding to reproduce smooth images. Although some authors advocate to remedy this situation by introducing pseudo-random noise such tactics seem more likely to disguise the bad by worse. Quantization with "zero" representation is highly recommended by this report.

The cutpoints for the quantization (3.11), can be chosen as the $N_1 + N_2$ arithmetic means of the $N_1 + N_2$ adjacent pairs given by (3.11). A_1, B_1, A_2 and B_2 are constants which determine the graininess and the maximum representable value of the quantization. Since the amount of possible quantization levels available for a given number of bits is in the form of 2^k, often either $N_1 = N_2+1$ or $N_2 = N_1+1$ (asymmetric quantization) is desirable for maximum usage of information available. Asymmetric quantization may increase slightly in hardware complexity. For large k (such as 5 or above), introducing asymmetry may not result in much information gain.

3.2 Quantization of the d.c. Component of the Hadamard Transformation

Logarithmic quantizations, though intuitively sound, cannot be applied directly to the d.c. component of the Hadamard transformation. For, the video signals of an arbitrary subpicture, Y, can assume any value within the range of the A-to-D converter, i.e. (3.1), and since the human eyes are relatively insensitive to relative brightness; it is not advantageous to bias the usable video region to any extent. However, the d.c. component can be encoded logarithmically by using DPCM technique. Here, the d.c. component of a subpicture is selected as a reference, the d.c. components of the following subpictures are coded as the differences of these signals. At the receiving end, the reference is first reconstructed and the d.c. components of the succeeding subpictures are reconstructed by updating their preceeding reconstructed components. Since a frame contains 480/m by 512/n subpictures, the DPCM technique can be applied horizontally, vertically or both. "Horizontally" means a reference is sent at the beginning of each horizontal group of subpictures. "Vertically" means a reference is sent at the beginning of each vertical group of subpictures. Only one reference is sent for each frame if both horizontally and vertically. In addition, DPCM vertically requires sufficient data storage for one line group, this is due to the

fact that the TV signal is horizontally oriented.

Encoding the d.c. component using the logarithmic DPCM method has the following advantages:

1) Smooth transition of grey levels between adjacent subpictures, which is characterized by small video amplitude variations, can be achieved with fewer bits than would otherwise be required. The smallest quantum jump using quantization (3.11) is

$$\begin{aligned} &A_1 \left(e^{B_1} - 1\right) \quad \text{for } - \text{ (light to dark)} \\ &A_2 \left(e^{B_2} - 1\right) \quad \text{for } + \text{ (dark to light)} \end{aligned} \tag{3.12}$$

2) When the grey level transition between subpictures is high, it is approximated logarithmically by the quantizer; due to the logarithmic response of human eyes, the corresponding error of visual sensation is relatively low.

3) The logarithmic quantizer can correct a step function to within the error given by (3.12) in M steps, where M is a logarithmic function of the amplitude of the step function and the graininess (3.12). This enables fewer bits of information to be allocated for quantization of the d.c. component than that would otherwise require.

The above statements can also be explained as follows: Since the d.c. component of the Hadamard transform corresponds

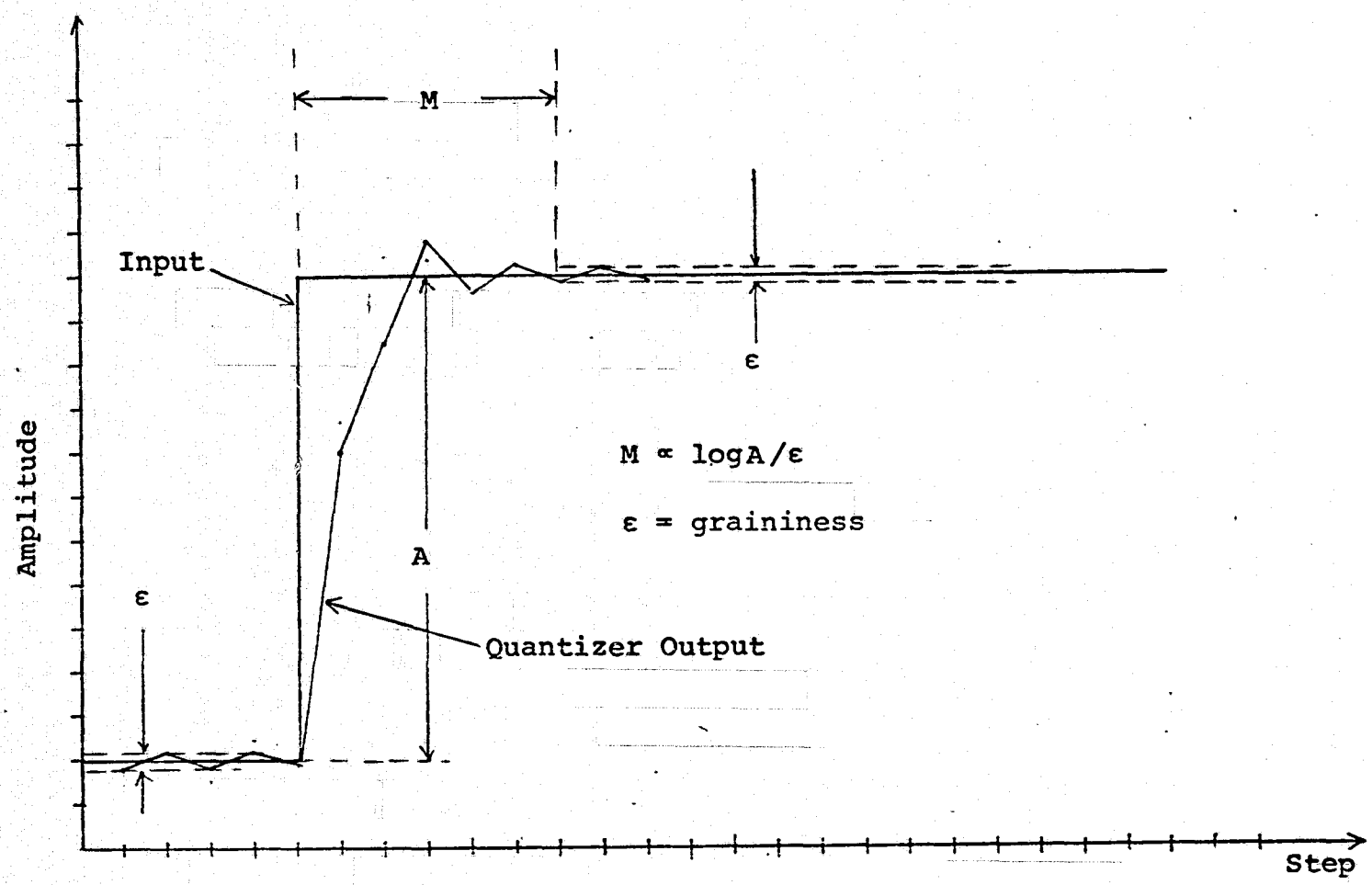

Figure 5. Typical Response of Logarithmic DPCM to Step Error

to an approximation of the original picture by subpictures (i.e. constant grey level), the spatial statistical correlation between subpictures remains valid (if the subpictures were small enough). This spatial statistical correlation is in the form of smooth transitions and logarithmic error tolerance of the human eyes. As in most cases, this spatial statistical correlation enables data reduction by using DPCM.

Our experiments showed that a 5-bit DPCM coding on the d.c. component can effectively reproduce the picture with reasonable visual quality that is free from "false image contouring." In contrast, at least seven bits are required if linear quantization were used.

DPCM coding has the disadvantage of being sensitive to the channel noise. This is caused by the fact that the reconstruction of an element at the receiving end is dependent on the reconstruction of the previous elements. DPCM horizontally will confine the errors to within a horizontal group of subpictures, and this method should be used.

3.3 Quantization of non-d.c. Hadamard Components

All non-d.c. Hadamard components are constant multiples of differences of two sums of video amplitudes within the subpicture. These components are normally very small and close to zero for recognizable pictures. In terms of visual

response, these components become significant only if the subpicture contains edge or spot information. Thus, their presence localized (or clustered) around the boundaries between distinct objects. Since edges or spots within subpictures over the ensemble of recognizable pictures are generally uncorrelated, we can assume the spatial statistical correlation between corresponding components of adjacent subpictures is insignificant, thus, coding these components with DPCM may not be advantageous. Due to their zero mean values and small variance (over the ensemble of recognizable pictures), quantization of these components should be chosen about the value zero. Quantization without zero representation (or nearly zero representation) will result in a "sandpaper" effect. Zero or near zero quantization representation for these components is recommended. Since, given a nonzero number of bits of information, the possible quantization levels are always even, zero representation always causes uneven quantization between positive and negative values. For components quantized with many bits (3 or more) the effect of the shift is negligible, but for quantization with fewer bits (2 or 1), the bias due to the shift may not be desirable, in such cases near zero quantization representation should be used. Often it is advantageous to combine the quantization tables of coarsely quantized

components with that of more finely quantized components. For example: 12 bits are normally used to code 2 components with 16 quantization levels (15 cutpoints) and 2 components with 4 quantization levels (3 cutpoints) each. The same number of bits is sufficient to code 1 component with 13 quantization levels, 1 component with 12 quantization levels, and 2 components with 5 quantization levels. Sharing bits enables most efficient use of amount of information available for a given allocated number of information bits. However, using this method will increase the arithmetic computation and subsequently the overall hardware complexity.

3.4 Selection of Quantization Levels

Based upon the above discussion, quantization levels for experiments performed in this report are chosen according to formulation (3.11). First the number of quantization levels is determined. This is (for non-sharing case) 2^K, where K is the number bits allocated. That is

$$2^K \geq N_1 + N_2 + 1$$

The minimum quantum jump, i.e., the graininess, using this method is given by (3.12) and the maximum change is given by

$$\begin{aligned} &A_1 \left(e^{B_1 N_1} - 1\right) \quad \text{for} \quad - \quad \text{(light to dark)} \\ &A_2 \left(e^{B_2 N_2} - 1\right) \quad \text{for} \quad + \quad \text{(dark to light)} \end{aligned} \qquad (3.13)$$

Normally, $A_1 = A_2$ and $B_1 = B_2$ are chosen for symmetry between hard and light directed transitions. The maximum change and the graininess determine the constants A and B and vice versa. The maximum change for the d. c. component is determined by the range of C_{11}, i.e.

$$0 \leq C_{11} \leq 255 \sqrt{mn}$$

The maximum change is selected as a fraction of $255\sqrt{mn}$. Here we used the fact that the probability for the transition of two adjacent subpictures from extreme darkness to extreme brightness is small. For the non d.c. components, the maximum change of quantization is chosen as a fraction of the range of these components, i.e., a fraction of $255\sqrt{mn}/2$. The fraction is generally determined by the number of quantization levels allocated and judged solely by experiments.

Applying the above discussion to 4 x 4 Hadamard transform, we come up with the following scheme. The component designation is shown in Figure 6. Here

C_{11} has 32 quantization levels using 5 bit DPCM.

C_{12} and C_{21} have 7 quantization levels.

C_{13} and C_{31} have 15 quantization levels.

C_{14} and C_{41} have 9 quantization levels.

+	+	+	+
+	+	+	+
+	+	+	+
+	+	+	+

C_{11}
(5-bit DPCM i.e., 32 Levels)

+	−	+	−
+	−	+	−
+	−	+	−
+	−	+	−

C_{12}
(7 levels)

+	+	−	−
+	+	−	−
+	+	−	−
+	+	−	−

C_{13}
(15 levels)

+	−	−	+
+	−	−	+
+	−	−	+
+	−	−	+

C_{14}
(9 levels)

+	+	+	+
−	−	−	−
+	+	+	+
−	−	−	−

C_{21}
(7 levels)

+	−	+	−
−	+	−	+
+	−	+	−
−	+	−	+

C_{22}
(discard)

+	+	−	−
−	−	+	+
+	+	−	−
−	−	+	+

C_{23}
(discard)

+	−	−	+
−	+	+	−
+	−	−	+
−	+	+	−

C_{24}
(discard)

+	+	+	+
+	+	+	+
−	−	−	−
−	−	−	−

C_{31}
(15 levels)

+	−	+	−
+	−	+	−
−	+	−	+
−	+	−	+

C_{32}
(discard)

+	+	−	−
+	+	−	−
−	−	+	+
−	−	+	+

C_{33}
(5 levels)

+	−	−	+
+	−	−	+
−	+	+	−
−	+	+	−

C_{34}
(5 levels)

+	+	+	+
−	−	−	−
−	−	−	−
+	+	+	+

C_{41}
(9 levels)

+	−	+	−
−	+	−	+
−	+	−	+
+	−	+	−

C_{42}
(discard)

+	+	−	−
−	−	+	+
−	−	+	+
+	+	−	−

C_{43}
(5 levels)

+	−	−	+
−	+	+	−
−	+	+	−
+	−	−	+

C_{44}
(discard)

Figure 6. Hadamard Component Designation

ORIGINAL PAGE IS
OF POOR QUALITY

C_{12}, C_{13}, C_{14}, (likewise C_{21}, C_{31}, and C_{41}) share 10 information bits.

C_{33}, C_{34}, and C_{43} each have 5 quantization levels, and they share 7 information bits.

The overall bit requirement is

$$5 + 10 + 10 + 7 = 32 \text{ bits}$$

or 2 bits per picture element.

The quantization levels and their cutpoints are shown in Table 1.

3.5 Further Possible Improvement in Coding Efficiency

In the above discussion, we have assumed that each Hadamard component can have independent occurrence within their range (3.7) and (3.8). To reproduce the true, original picture, each component must be capable of covering the entire range. Source encoding by truncating the range always results in degradation of the reproduced picture, although the statistical data reduction method intends to limit the degradation to areas that occur rarely. Yet, such effects are generally felt by an observer. Hence, picture quality improvement can be achieved if the true ranges of each Hadamard component can be established. Since the subpicture, Y, has range:

$$Y = \begin{pmatrix} X_{11} \cdots X_{1n} \\ X_{n1} \cdots X_{mn} \end{pmatrix} \qquad \begin{array}{l} 0 \leq X_{ij} \leq 255 \\ X_{ij} \text{ - integer} \end{array}$$

C_{11}: DPCM Logarithmically by 5 Bits (Range: $0 \leq C_{11} \leq 1024$)

Cutpoints	Representative Value
	876
761	
	646
562	
	476
414	
	350
304	
	258
223	
	188
163	
	138
118	
	100
85	
	71
61	
	50
42	
	36
29	
	23
19	
	15
11	
	8
6	
	4
2	
	0
-2	
	-4
-6	
	-8
-11	
	-14
-18	
	-22
-27	
	-32
-39	
	-47
-56	
	-65
-78	
	-91
-107	

Table 1. Quantization Table

C_{11}: DPCM Logarithmically by 5 Bits (Continued)

Cutpoints	Representative Value
	-124
-147	
	-169
-199	
	-229
-209	
	-310
-363	
	-417
-488	
	-560
-655	
	-750
-877	
	-1000

Non-d.c. components have range: -512 to +512

C_{13} and C_{31}: Quantized by 15 Levels

Cutpoints	Representative Value
	±150
±122	
	±94
±76	
	±59
±47	
	±35
±28	
	±20
±15	
	±10
±7	
	±4
±2	
	0

Table 1. Quantization Table (Continued)

C_{14} and C_{41}: Quantized by 9 Levels

Cutpoints	Representative Value
	±70
±53	
	±36
±26	
	±17
±11	
	±6
±3	
	0

C_{12} and C_{21}: Quantized by 7 Levels

Cutpoints	Representative Value
	±60
±43	
	±26
±17	
	±9
±4	
	0

C_{33}, C_{34} and C_{43}: Quantized by 5 Levels

Cutpoints	Representative Value
	±50
±33	
	±15
±8	
	0

Table 1. Quantization Table (Continued)

which lies within the mn-dimensional cube of size 255. When transformed into the Hadamard components, the corresponding boundary condition must also be satisfied. It is not difficult to see that the transformed Hadamard components satisfy the following boundary condition

$$\sum_{i=1}^{m} \sum_{j=1}^{n} |c_{ij}| - |c_{11}| + |c_{11} - \frac{255\sqrt{mn}}{2}| \leq 255\sqrt{mn}/2 \quad . \; . \; . \qquad (3.14)$$

See Appendix A. (3.14) merely states that if some components have very high absolute values (i.e., near the boundary of representable video signals), then the remaining components cannot have large absolute values. In other words, using (3.14), we can improve the estimation for the ranges of occurrence of non-d.c. components of the Hadamard transformation by disregarding the nonaccessible region.

To apply this method, first we have to order the priority of the Hadamard components. This can be normally chosen according to the variances. For 4 x 4 Hadamard transforms, we can choose the following order:

c_{11}, c_{13}, c_{31}, c_{14}, c_{41}, c_{12}, c_{21}, c_{33}, c_{34}, c_{43}
(rest discarded)

c_{11} is quantized in the usual way. c_{13} is quantized as a ratio of

C_{13} to $A_1 = |C_{11} - 255\sqrt{mn}/2|$

C_{31} as a ratio of:

C_{31} to $A_2 = A_1 - |Q(C_{13})|$

when $Q(C_{13})$ = inverse quantized representation of C_{13}.

C_{14} as a ratio of:

C_{14} to $A_3 = A_2 - |Q(C_{14})|$

when $Q(C_{14})$ = inverse quantized representation of C_{14}, and etc.

Alternatively, given a set of quantization, a correction factor based upon A_1, A_2, etc., is premultiplied by C_{13}, C_{31}, etc., and these are compared with the prechosen cutpoints as in the usual manner. The inverse quantization representations are obtained by the representation value and the reciprocal of the correction factors.

4.0 Buffer-Free Frame-to-Frame Data Compression

This technique, proposed in a previous LINKABIT video study report [4], utilizes both the spatial and time statistical correlations of the emsemble of recognizable pictures. In this method, a reference of the 2-dimensional compressed Hadamard components is transmitted; for the subsequent k (k is a positive integer) frames, only the differences between a few major components and their corresponding components of the reference frame are quantized and transmitted. The process is then repeated every k + 1 frames. k was chosen to be 3. The major differencing components are:

$$C_{11}, C_{13}, \text{ and } C_{31}.$$

C_{11} was quantized and represented by 5 information bits. C_{13} and C_{31} are each quantized and represented by 3 information bits. The reference frame was 2 dimensional compressed with an average of 2 bits per picture element. Hence, the overall data requirement is slightly over 1 bit per picture element.

This method has the advantage of using only a partial frame memory. Here, (for the encoder) only the information of major components of the reference frame is required. Moreover, the information storage may not necessarily be the original data with 8-bit precision. This method was implemented, and it was shown to be

capable of reproducing a scene containing normal object movement without degrading the picture quality beyond recognition. The objective of this report is to explore and search for possible improvements in this technique. These are summarized as follows:

1) The data compression on the reference frame is modified per Section 3. There, a new set of quantization and decoding strategy is used.

2) This study found that it is preferrable to update the differencing components with new differences. (In contrast, in the previous LINKABIT video study, the differences were always obtained between the new and the reference frames). In this manner, the coding can be operated in true DPCM mode, and, the errors can be successively corrected. This is particularly desirable for scenes with short movement duration, i.e., containing objects that move and pause, it enables better response to stationary objects.

3) In the previous LINKABIT video study, the d.c. component, C_{11}, is quantized to 6 bits in the reference frame and by 5 bits in the differencing frame. The improvement using DPCM coding reduces the requirement of the d.c. component in the reference frame to 5 bits. Thus unless,

in the differencing frames, this component can be coded by 4 or less information bits, coding this component as differences does not offer special advantage. Experiments have shown that it is generally insufficient to use 4 information bits to encode the frame differencing d.c. component. This can be explained by the fact that human eye is quite sensitive to object displacements, and the fact that the time statistical correlation between corresponding elements of two adjacent frames, given motion existed, is rather small. (In fact, most statistical correlation exists due to stationary objects; its advantage can only be extracted by motion indicators).

Hence, it is preferrable to transmit the d.c. component in the regular mode, i.e., logarithmic DPCM coding with 5 information bits. In this manner, the requirement of frame memory can be further reduced.

4.1 Quantization of Buffer Free Frame-to-Frame Differencing Technique

Based upon the above discussion, the quantizations for the buffer-free frame-to-frame differencing technique is as follows:

1) The reference frame is encoded as described in Section 3. The corresponding quantized table is given in Table I.

2) The d.c. component of the differencing frames is quantized as in 1).

3) The Hadamard component C_{13} and C_{31} are represented by 5 information bits. Their quantized and truncated value obtained in the reference frame are storage in the frame memory (5 bits each). In the difference frames, the difference is quantized by Table II (using 3 information bits each). The frame memory is updated by the quantized values. The total storage requirement for the frame memory is (for the encoder):

$$2 \times (512/4) \times (480/4) \times 5 = 153 \cdot 6K \text{ bits}$$

The 5-bit representation of C_{13} and C_{31} does not have to cover the entire range of C_{13}, C_{31} (i.e., $-255\sqrt{mn}/2 \leq C_{13}, C_{31} \leq +255\sqrt{mn}/2$). It has to cover only the absolute maximum change given by the 2 dimensional quantization, in this, according to Table I, the 5-bit representation linearly partitions the range: -150 to +150.

4.2 Compression Experiments

Experiments were performed on the LIM Video Controller for k = 3, 5, 7, and 9. Correspondingly, the coding efficiencies are 1.0156, .90625, .8516, and .8187 bits per picture element. The results, together with the original A-to-D/D-to-A and the 2-dimensional compressed pictures, were recorded on video tape. The tape presentation consists of processed video first run at normal speed (i.e. 30 frames/second) then immediately followed by instant replay at 8 to 1 slow motion rate. The tape was recorded by a Sony

C_{13} and C_{31}: Quantized by 8 Levels (i.e., 3 Bits) Each

Cutpoints	Representative Value
	87
67	
	52
37	
	23
10	
	0
-8	
	-20
-32	
	-41
-50	
	-62
-75	
	-100

Table II. Quantization Table

video tape recorder, which unfortunately has very limited bandwidth and occasionally develop an appreciable amount of noise.

From the experiments, it is evident that stationary objects in a scene are reproduced with reasonable quality. But for objects with high rate of displacements, edge distortion is clearly visible. The effect becomes highly objectionable to the observer for large value, k. The edge distortion is clearly caused by the lack of high order Hadamard components. This effect can be reduced by introducing more differencing components. This method will increase the size of the frame memory and the average information bit required per picture element. Time sharing, in the form of updating some components at one frame and some others at different frames, can also improve the motion quality.

Frame sharing or updating the differencing components was tried. The differencing strategy is given as follows:

1st and 5th differencing frame:

C_{13}, C_{31}, C_{12} at 3 bits each.

2nd and 6th differencing frame:

C_{13}, C_{31}, C_{21} at 3 bits each.

3rd and 7th differencing frame:

C_{13}, C_{31}, C_{14} at 3 bits each.

4th and 8th differencing frame:

C_{13}, C_{31}, C_{41} at 3 bits each,

repeating every 9 frames. Each differencing frame uses 14 bits. The overall average efficiency is 1 bit per pel. The experiment is recorded on video tape (Sample E).

A computer simulation program is given in Appendix B. From the simulations, the technique described in the above method provides a scheme of transmitting picture information with relative ease for hardware implementation. The degree of quality depends mainly on the factor k, the number of differencing frames, and the number of time sharing differencing components (the hardware complexity increases directly with the number of differencing components used). Hardware implementation for the buffer-free frame-to-frame differencing technique is summarized in the following section. Detailed description of individual functional blocks is skipped. The main points stressed are the general hardware involved and the bulk storage required for the encoder and the decoder.

A possible improvement on the above technique is by interpolation of high order Hadamard components (those that were discarded in the differencing mode) between reference frames. This method has not been simulated.

4.3 Requirement for Hardware Implementation

A buffer-free frame-to-frame differencing technique based upon the discussion of Section 4.1 can be implemented with relatively simple electronic hardware. The block diagrams of the encoder and decoder are illustrated in Figure 7 and 8, respectively.

For the source encoder, due to the horizontal orientation of the regular TV signals, a line buffer (storage for 3 horizontal lines + 4 samples) is required in order to perform the 4 x 4 Hadamard transformation. This line buffer normally requires high read and write speed (these are determined by the period of the A-to-D sampling clock pulses). The Hadamard transformation is performed by a fast serial/parallel Hadamard transformer. Since each 4 x 4 Hadamard transformation requires 16 operations of add or subtract, sufficient bits should be reserved to avoid error due to truncation.

Of all the Hadamard components, only the d.c. component is DPCM coded. This requires one additional data storage, data comparison and updating. Quantizations are carried out by ROM (read-only memory) table search. For components with bit-sharing, additional coding, such as bit mixing, is required. For example, the components C_{33}, C_{34}, and C_{43} are coded by 7 bits, each with 5 levels of quantization. This

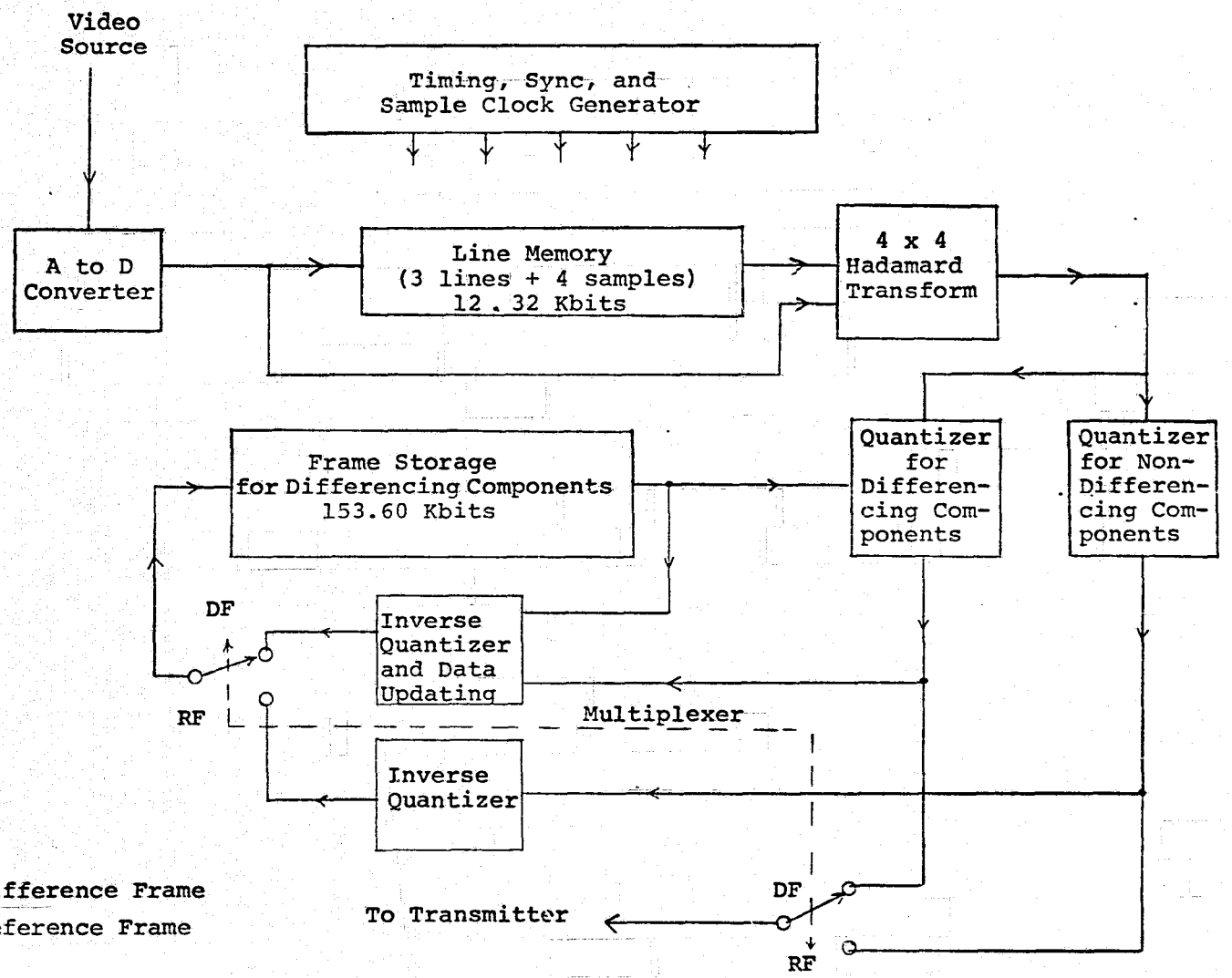

Figure 7. Video Source Encoder

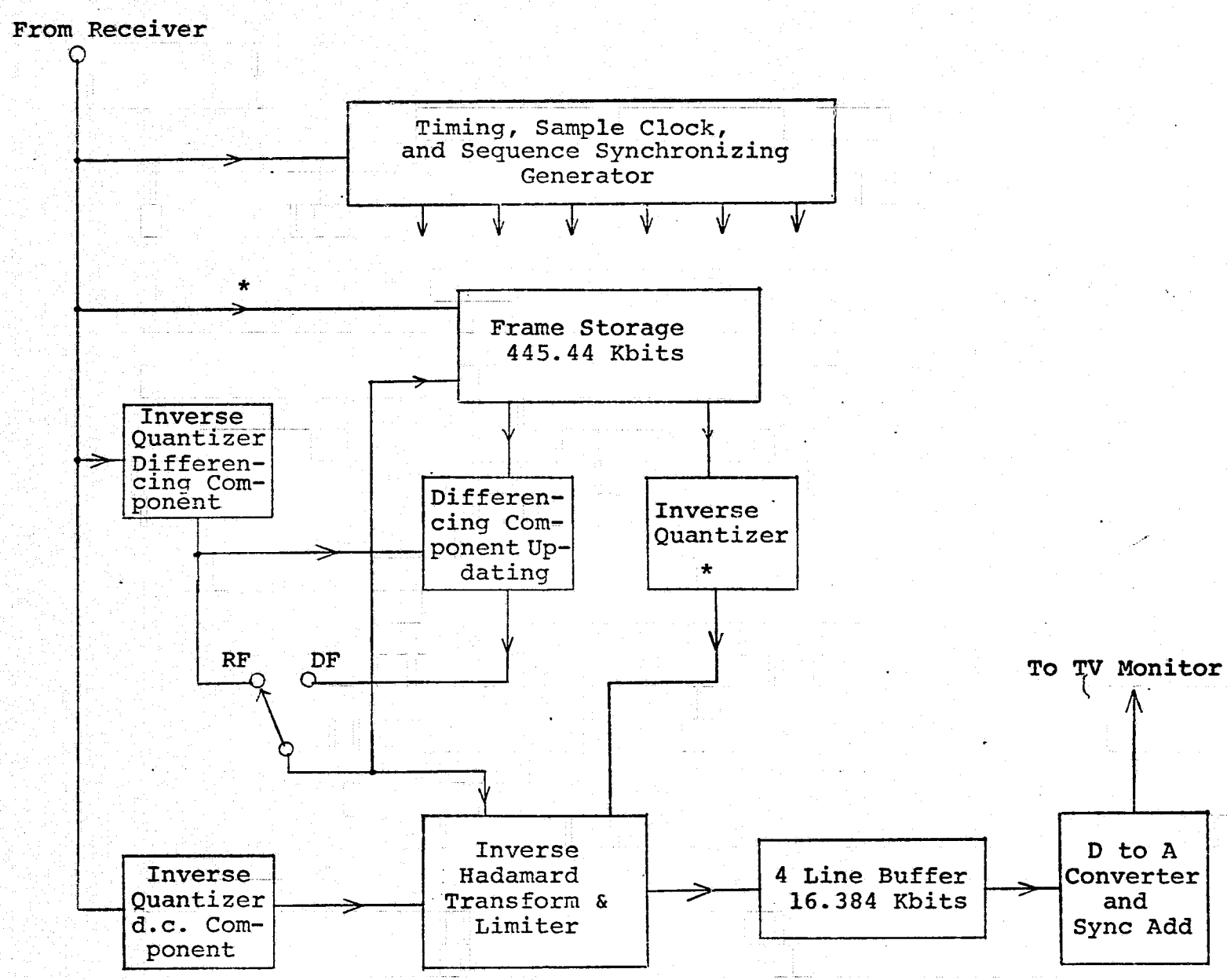

Figure 8. Video Source Decoder

can be done as follows: C_{43} can be represented by an integer having value between 0 and 4. C_{34} can be represented as an integer of the form 5k, k = 0, 1, 2, 3, or 4. C_{33} can be represented as an integer of the form 25k, k = 1, 2, 3, or 4. The resultant code is the sum of the integer representations which has range between 0 and 124 (this is readily represented by 7 bits). If C is the resultant coded integer, then C_{43} can be recovered by the residue of C divided by 5. C_{34} can be recovered by the residue of [C/5] (i.e., the quotient of C divided by 5) divided by 5. And C_{33} can be recovered by the quotient of C divided by 25.

The frame storage of the encoder is required to store the updated information of the differencing components. These components are updated to their nearest representative value. For component C_{13} and C_{31}, (each are represented by 5 bits) the frame memory of size 153.6 K bits is required. (If additional differencing components are introduced, the frame memory will increase correspondingly). Speed requirement for the frame memory is generally very small and can be implemented by many low-speed, low-power devices.

At the receiving end, the decoder requires an almost complete frame memory. Here, data that is transmitted as differences or discarded in differencing, must be preserved. Only the d.c. component, which is transmitted continuously, does not require storage. The differencing components, C_{13}

and C_{31}, each are represented by 5 bits. The overall size of the frame memory is

$$(512/4) \times (480/4) \times (32-5+2) = 445.44 \text{ K bits.}$$

The frame memory stores only the encoded data for those components discarded in the differencing frames (but not discarded in the reference frame), and are retrieved by the inverse quantizer. For the differencing component, they are stored as the 5-bit representation of their updated inverse quantizer. As in the encoder, this memory can be implemented by many low-speed, low-power devices.

The inverse quantized Hadamard components are fed the inverse Hadamard transformer. The resultant data is limited to the usable range of the D-to-A converter and stored in the 4 line buffer before it can be converted into video signal by the D-to-A converter and sync adder.

The frame memory size of the encoder and decoder can be decreased slightly if the updated differencing components are approximated by their 2-dimensional quantization (4 bits each is sufficient). In this case, the frame memory size for the encoder is 122.88 K bits, and that for the decoder is 414.72 K bits. However, a quantizer is needed in the decoder and the appoximation using this method will not be as accurate as the previous case.

4.3.1 Hardware Estimations

A general estimation of power consumption, size and weight for the above system, based upon the LINKABIT Real-Time Video Compression System, model LT8E/LT8D, is itemized in the following paragraphs. (Power supplies, their efficiencies, channel encoder, and additional data rate buffer are not included in the assumption).

4.3.1.1 Video Source Encoder

The video source encoder can be partitioned into the following submodules:

1) Timing Generator: which generates all pertinent timing signals, such as sample clock pulses, control signals, etc.

2) Hadamard Transformer: which performs the fast Hadamard transformations in a "pipe-line" configuration.

3) Line Buffer: which stores 3 horizontal lines of data prior to the Hadamard transformation.

4) Quantizer: which provides all 2-dimensional quantization and quantization for the differencing components.

5) Frame Memory: this storage in the encoder is used to store the information of the differencing components. It can be implemented by high density CCD (Change Coupled Device) shift registers. A low power and compact version of size 16.384 Kbit is available for such an application. Ten such devices are sufficient for the encoder frame memory.

6) A-to-D Converter and Other Related Analog Circuitries: this provides proper d.c. restoration and conversion of video signals into digital data prior to processing.

The estimated power dissipation and number of ICs required are summarized in Table III.

	ESTIMATED POWER DISSIPATION	ESTIMATED NUMBER OF IC's USED
Timing Generator	7.5 watts	40
Hadamard Transform	18 watts	56
Line Buffer	13 watts	35
Frame Memory	7 watts*	25*
Quantizer	10 watts	50
A-to-D Converter and Related Analog Circuitries	13.5 watts	Size: 100 cubic inches Weight: 2.6 lbs**

* This includes data control logic.

**This is based upon the A-to-D converter manufactured by Mirco Consultants, model AN-DI-802 RAD-B, which has a size of 3.7" x 2" x 9.5" and weights 2.2 lbs.

TABLE III.

Using the assumption of packaging density of

1.5 IC's/cubic inch

and 30 IC's/pound,

the estimated power dissipation, weight and size are:

Estimated Power Dissipation:	70 watts
Estimated Weight:	10 lbs
Estimated Size:	240 cubic inches

4.3.1.2 Video Source Decoder

The Video Source Decoder can be partitioned into the following:

1) Timing Generator,
2) Hadamard Transform,
3) 4-Line Buffer,
4) Inverse Quantizer,
5) Frame Memory,
6) D-to-A Converter and Sync Adder.

The estimated power dissipation and number of IC's required are summarized in Table IV.

The estimated power dissipation, weight, and size are:

Estimated Power Dissipation:	83 watts
Estimated Size:	190 cubic inches
Estimated Weight:	8.5 lbs.

	ESTIMATED POWER DISSIPATION	ESTIMATED NUMBER OF IC'S
Timing Generator	7.5 watts	30
Hadamard Transform	18 watts	56
4-Line Buffer	15 watts	36
Frame Memory	20 watts	50
Inverse Quantizer	15 watts	50
D-to-A Converter & Sync Adder	7.5 watts	Size: 40 cubic ins. Weight: 1 lb. (Self-enclosed in box)

TABLE IV.

REFERENCES

1. LIM Video Controller Operations Manual, LINKABIT Corporation, San Diego, California.

2. Landau, H. J., and Slepian, D., "Some Computer Experiments in Picture Processing for Bandwidth Reductions," The Bell System Technical Journal, Volume 50, No. 5, May-June, 1971.

3. Candy, J. C., Franke, M. A., Haskell, B.G., and Mounts, F. W., "Transmitting Television as Clusters of Frame-to-Frame Differences."

4. "Study of Information Transfer Optimization for Communication Satellites," LINKABIT Corporation Final Report, NASA Ames Research Center, Contract NAS2-6810, 3 January, 1973.

5. Cornsweet, T.N., Visual Perception, Academic Press, 1970.

6. Payne, S.E., The Construction of Hadamard Matrics, Technology Incorporated, Dayton, Ohio. Publication No. ARL 73-0117, July, 1973.

7. Wintz P. A., "Transform Picture Coding," Proceedings of the IEEE, Volume 60, No. 7, July, 1972.

APPENDIX A

The objective of this Appendix is to sketch briefly the proof of the inequality

$$\sum_{i=1}^{m} \sum_{j=1}^{n} |c_{ij}| - |c_{11}| + |c_{11} - 255\sqrt{mn}/2| \leq 255\sqrt{mn}/2$$

It should be helpful to observe the case when m = 1 and n = 2. In this case, the Hadamard basis is

$$\hat{b}_{11} = \frac{1}{\sqrt{2}} (1, 1)$$

$$\hat{b}_{12} = \frac{1}{\sqrt{2}} (1, -1) \quad .$$

The original domain of the 1 x 2 subpicture and the transformed coordinates are illustrated in Figure A1. Since the transformed components must lie within the square of the original signal domain, then the components, according to the figure, can be easily verified to satisfy the following inequality

$$|c_{11} - 255/\sqrt{2}| + |c_{12}| \leq 255/\sqrt{2} \qquad \text{(kth nearest integer)}$$

which is the above inequality.

For the arbitrary case $mn = 2^k$, $k \geq 1$, it can be readily proved by first coordinate translation of the original subpicture, Y

$$Y = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & & \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}$$

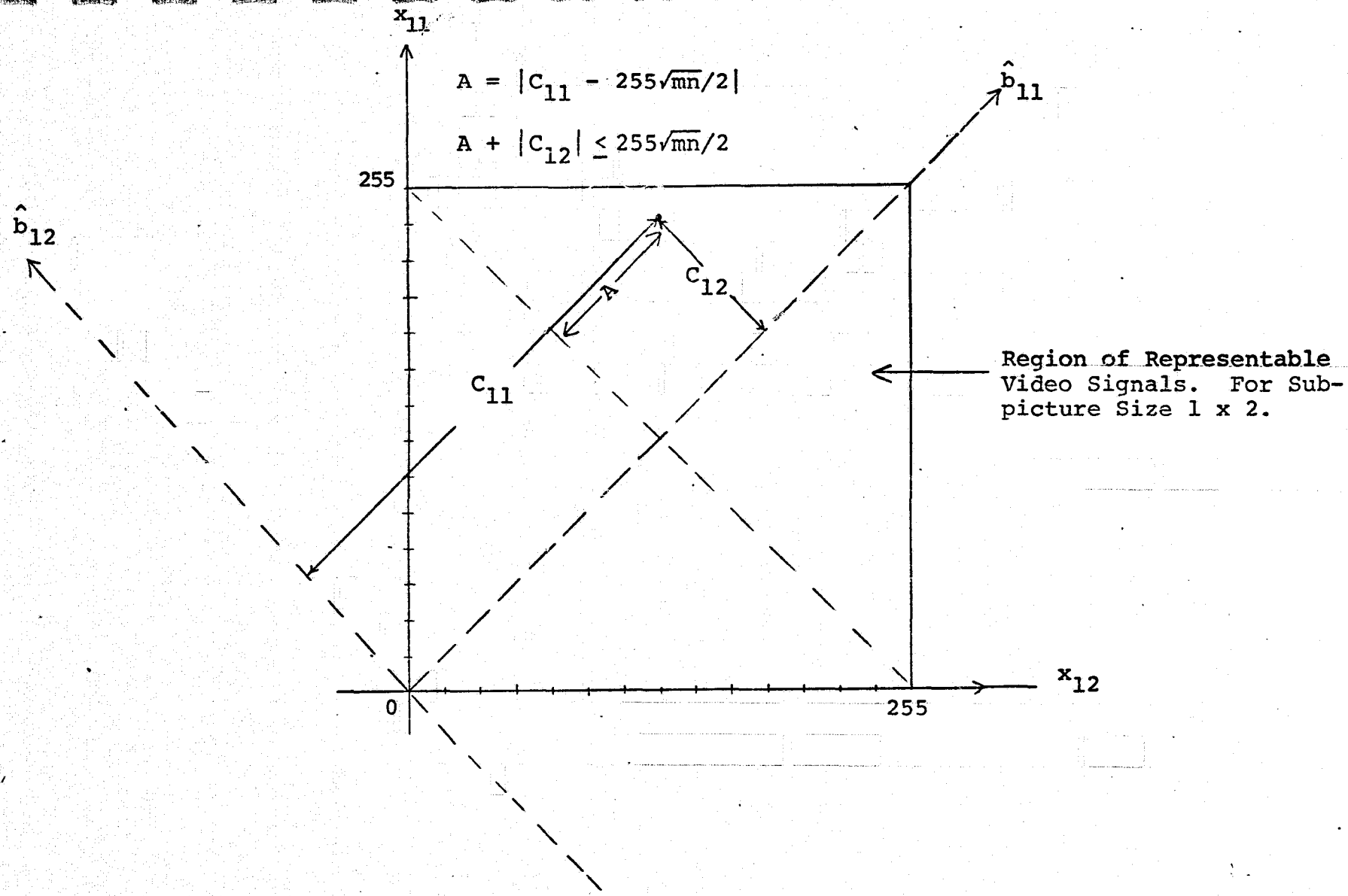

Figure A1. Illustration of Inequality (3.14) in 1 x 2 Case

into

$$Y = \begin{pmatrix} x'_{11} & \cdots & x'_{1n} \\ \vdots & & \\ x'_{m1} & \cdots & x'_{mn} \end{pmatrix}$$

when $x'_{ij} = x_{ij} - 255\sqrt{mn}/2$.

x'_{ij} has range $-255\sqrt{mn}/2$ to $+255\sqrt{mn}/2$. (Note: all components, except the d.c. component, of the Hadamard transform of Y' are identical to those of Y). Then apply the fact that the inverse Hadamard transform in the translated coordinates has peak value (i.e., max $|x_{ij}|$) equal to the sum of the absolute values of the individual Hadamard components (which can be shown by induction on k).

Remark: Source coding without using the inequality (3.14) may result in inverse transformed signals lying outside the domain of the original signal. Using the inequality helps to confine the occurrence of video signals of the ensemble of recognizable picture to more likely area of occurrence.

APPENDIX B

The computer simulation for the buffer-free frame-to-frame differencing technique is simulated by the LIM Video Controller. The Fortran program is listed in the following pages. Most of the subroutines used are explained in the LIM Video Controller Operations Manual. The following is a list of subroutines used in the program but were not included in the LIM Video Controller Operations Manual.

1) MEAN: Calculate the arithmetic mean of a set of numbers. (Note: in the simulation, the average of the d.c. component over a horizontal line group is used. This is used because of hardware implementation of the LIM Videc Controller where substantial sampling occurs in the blanking and sync region.

2) DBNQT, DLNQT: Calculate, successively a set of numbers, their successive differences quantized and updated by inverse quantizations. This simulates the DPCM coding. DBNQT uses logarithmic search technique, while DLNQT uses linear search technique. The quantization cutpoints and inverse representative values are specified as pointers in the arguments of these subroutines.

3) DIFF and RENEW: These subroutines substract and add, respectively, two sets of numbers. They are used to calculate differences and perform updating for the differencing components.

4) ZREAD, ZWRIT, ZTEST and ZINIT: are special fast digital disk routines.

Titles are prewritten on the disk with name "TITLE."

The program allows simulation of time-sharing differencing components. The differencing information is stored in the array, NZX, which indicates which components are to be differenced and provides pointers for the quantization and inverse quantization allocated for the component.

In the video simulations, a grey vertical band is visible to the right side of the picture. This is the glitch filter override, which overrides the switching spikes caused by the video disk recorder.

```
/ FOR
      *LIST ALL
      *IOCS(DISK)
      *IOCS(2501 READER)
      *IOCS(KEYBOARD,TYPEWRITER)
      *ONE WORD INTEGERS

-ERRS...STNO.C.....  F O R T R A N   S O U R C E   S T A T E M E N T S  ........  IDENTFCN    **COMPILER MES

            INTEGER JD1(4162),JD2(4162),BF1(2049,2),BF2(2049,2)
            INTEGER QL2(32),IQL2(32),Q1(16),IQ1(16),Q3(9),IQ3(9)
            INTEGER Q4(7),IQ4(7),Q5(5),IQ5(5)
            INTEGER NZX(15,4),DQ(16,3),IDQ(16,3)
            INTEGER CHAR(16,16),DIGIT(3),FNUM(2),EOF,BOF
            EQUIVALENCE (JD1(1),BF1(1,1)),(JD2(1),BF2(1,1))
            DEFINE FILE 1(780,320,U,K)
            DEFINE FILE 2(130,320,U,K)
            DEFINE FILE 3(40,256,U,K)
            DEFINE FILE 4(130,320,U,K)
            DATA FNUM/1,4/,JD1(4162),JD2(4162)/2*4160/
            DATA DIGIT/440,420,400/,LEV1/50/,LEV2/100/
            CALL ZINIT(FNUM,2,IFG)
            READ (2'1) BF1
            CALL ZWRIT(2,2,JD1(4162))
            READ (8,100) EOF,NLG,NFRM,IFG,M1
            WRITE (1,200) EOF,NLG,NFRM,IFG,M1
            IF (M1) 396,396,395
        396 M1=0
            M2=2
            M3=2
            M4=1
            GOTO 397
        395 M1=2
            M2=0
            M3=1
            M4=2
        397 BOF=EOF-4*NLG+4
            CALL HMODE(M3,M4)
            DO 566 I=1,4
            READ (8,100) (NZX(J,I),J=1,15)
            IF (IFG) 566,566,588
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S ........ IDENTFCN    **COMPILER MES
      588 WRITE (1,200) (NZX(J,I),J=1,15)
      566 CONTINUE
          READ (8,100) QL2
          READ (8,100) IQL2
          READ (8,100) Q1
          READ (8,100) IQ1
          READ (8,100) Q3
          READ (8,100) IQ3
          CALL ZTEST(2,IX,1)
          READ (2'14) BF1
          CALL ZWRIT(2,1,JD1(4162))
          READ (8,100) Q4
          READ (8,100) IQ4
          READ (8,100) Q5
          READ (8,100) IQ5
          IF (IFG) 411,411,413
      413 WRITE (1,333)
          WRITE (1,200) QL2
          WRITE (1,250) IQL2
          WRITE (1,333)
          WRITE (1,200) Q1
          WRITE (1,250) IQ1
          WRITE (1,333)
          WRITE (1,200) Q3
          WRITE (1,250) IQ3
          WRITE (1,333)
          WRITE (1,200) Q4
          WRITE (1,250) IQ4
          WRITE (1,333)
          WRITE (1,200) Q5
          WRITE (1,250) IQ5
      411 WRITE (1,333)
          DO 197 I=1,3
          READ (8,100) (DQ(J,I),J=1,16)
          READ (8,100) (IDQ(J,I),J=1,16)
          IF (IFG) 197,197,194
      194 WRITE (1,200) (DQ(J,I),J=1,16)
```

Read and Output to the Console. The Quantization Cutpoints and their Inverse Representative Values.

ORIGINAL PAGE IS
OF POOR QUALITY

-ERRS...STNO.C..... F O R T R A N S O U R C E S T A T E M E N T S ..... IDENTFCN **COMPILER ME

```
      WRITE (1,250) (IDQ(J,I),J=1,16)
      WRITE (1,333)
  197 CONTINUE
      CALL ZTEST(2,IFG,1)
      DO 808 IFRM=1,NFRM
      IDFX=MOD(IFRM-1,9)
      ILN=EOF
      CALL FATOD(1,ILN,0)
      IRF=2
      DO 800 ILG=1,NLG
      CALL ZTEST(1,IFG,1)
      IF (IDFX) 355,355,356
  356 CALL ZREAD(1,ILG,JD2(4162))
      CALL ZTEST(1,IFG,1)
  355 DO 269 IP=1,2
      BF1(2049,IP)=2048
      CALL RD4LN(0,BF1(2049,IP))
      IF (ILG-NLG) 251,248,248
  248 IF (IP-1) 251,251,259
  251 CALL FATOD(IRF,ILN,1)
      IF (IRF-1) 258,258,256
  258 IRF=2
      GOTO 259
  256 IRF=1
      ILN=ILN-4
  259 CALL SBHM(BF1(2048,IP),BF1(1536,IP),512,128)
      CALL SHM2(BF1(2048,IP),BF1(1024,IP),1024)
      CALL MEAN(125,BF1(2048,IP),BF1(1,IP),0)
      CALL DBNQT(QL2(32),32,IQL2(32), BF1(2048,IP),125, BF1(1,IP))
      IF (IDFX) 347,347,337
  347 CONTINUE
      CALL LNQT(Q4(7),7,IQ4(7),BF1(2047,IP),128)
      CALL LNQT(Q1(16),15,IQ1(16),BF1(2046,IP),128)
      CALL LNQT(Q3(9),9,IQ3(9),BF1(2045,IP),128)
      CALL LNQT(Q4(7),7,IQ4(7),BF1(1536,IP),128)
      CALL SETVL( BF1(1535,IP),128,0)
      CALL SETVL( BF1(1534,IP),128,0)
```

Starts Processing

K + 1 (To change the value of K, this card is replaced by IDFX = MOD [IFRM-1, k + 1])

DPCM-the d.c. component

ORIGINAL PAGE IS OF POOR QUALITY

```
C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S ........ IDENTFCN    **COMPILER ME
      CALL SETVL( BF1(1533,IP),128,0)
      CALL LNQT(Q1(16),15,IQ1(16),BF1(1024,IP),128)
      CALL SETVL( BF1(1023,IP),128,0)
      CALL LNQT(Q5(5),5,IQ5(5),BF1(1022,IP),128)
      CALL LNQT(Q5(5),5,IQ5(5),BF1(1021,IP),128)
      CALL LNQT(Q3(9),9,IQ3(9),BF1(512,IP),128)
      CALL SETVL( BF1(511,IP),128,0)
      CALL LNQT(Q5(5),5,IQ5(5),BF1(510,IP),128)
      CALL SETVL( BF1(509,IP),128,0)
      GOTO 269
  337 CONTINUE
      CALL XFER(BF1(2048,IP),BF2(2048,IP),128)
      IY=MOD(IDFX,4)+1
      DO 805 I=1,15
      IX=NZX(I,IY)
      IFG=I/4
      IFG=(4-IFG)*512
      IFG=IFG-MOD(I,4)
      IF (IX) 805,805,886
  886 CALL DIFF(BF1(IFG,IP),BF2(IFG,IP),128)
      CALL LNQT(DQ(16,IX),DQ(1,IX),IDQ(16,IX),BF1(IFG,IP),128)
      CALL RENEW(BF2(IFG,IP),BF1(IFG,IP),128)
  805 CONTINUE
  269 CONTINUE
      CALL ZTEST(1,IFG,1)
      IF (IDFX) 991,991,992
  991 CALL ZWRIT(1,ILG,JD1(4162))
      GOTO 800
  992 CALL ZWRIT(1,ILG,JD2(4162))
  800 CONTINUE
      CALL HMODE(M1,M2)
      ILN=EOF
      CALL BLANK(1,EOF,2,BOF)
      CALL ZTEST(1,IFG,1)
      CALL ZREAD(1,1,JD1(4162))
      CALL BLANK(2,EOF,1,BOF)
      ILG=2
```

Quantization and Inverse Lookup for Reference Frames

Quantizing and Inverse Lookup for Differencing Components

Store in Digital Disk

ORIGINAL PAGE IS
OF POOR QUALITY

```
C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S ........ IDENTFCN    **COMPILER ME

      IDK=2
      DO 797 MT=1,NLG
      CALL ZTEST(1,IFG,1)
      IF (ILG-NLG) 166,166,985
  166 IF (IDK-1) 984,984,986
  984 CALL ZREAD(1,ILG,JD1(4162))
      GOTO 985
  986 CALL ZREAD(1,ILG,JD2(4162))
  985 DO 757 IFM=1,2
      IF (IDK-1) 381,381,382
  382 BF1(2049,IFM)=2048
      CALL SBHM(BF1(2048,IFM),BF1(1536,IFM),512,128)
      CALL SHM2(BF1(2048,IFM),BF1(1024,IFM),1024)
      CALL LMRN4(BF1(2048,IFM),2048,0,4090)
      CALL WR4LN(0,BF1(2049,IFM))
      GOTO 757
  381 BF2(2049,IFM)=2048
      CALL SBHM(BF2(2048,IFM),BF2(1536,IFM),512,128)
      CALL SHM2(BF2(2048,IFM),BF2(1024,IFM),1024)
      CALL LMRN4(BF2(2048,IFM),2048,0,4090)
      CALL WR4LN(0,BF2(2049,IFM))
  757 CALL FDTOA(IFM,ILN,1)
      ILG=ILG+1
      IDK=MOD(IDK,2)+1
  797 ILN=ILN-4
      CALL ZTEST(1,IFG,1)
      DO 681 IX=1,2
      DO 681 IGP=1,2048
  681 BF1(IGP,IX)=LEV2
      CALL WR4LN(0,BF1(2049,1))
      IP=ILN
      MT=ILN-12
      DO 466 IFM=1,2
      GOTO (771,772),IFM
  771 CALL ZTEST(1,IFG,1)
      CALL ZREAD(2,1,JD1(4162))
      GOTO 773
```

Inverse Hadamard Transform, Limit to Usable Range, and Record on Video Disk Record

ORIGINAL PAGE IS
OF POOR QUALITY

```
-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S ........ IDENTFCN    **COMPILER MES
      772 CALL ZTEST(2,IFG,1)
          CALL ZREAD(2,2,JD2(4162))
      773 CONTINUE
          CALL FDTOA(IFM,MT-8,0)
          CALL FDTOA(IFM,MT-4,0)
          CALL FDTOA(IFM,MT,0)
      466 CALL FDTOA(IFM,ILN,1)
          I=IFRM
          CALL ZTEST(2,IFG,1)
          DO 682 ILG=1,3
          IY=MOD(I,10)+1
          I=I/10
          READ (3'IY) CHAR
          IRX=1
          DO 864 ILN=1,4
          DO 864 IFM=1,2
          IX=512*(5-ILN)-DIGIT(ILG)
          DO 865 MT=1,16
          IFG=CHAR(MT,IRX)
          CALL SGN(IFG)
          BF2(IX,IFM)=IFG*LEV2+LEV1
          IFG=CHAR(MT,IRX+8)
          CALL SGN(IFG)
          BF1(IX,IFM)=IFG*LEV2+LEV1
      865 IX=IX-1
      864 IRX=IRX+1
      682 CONTINUE
          ILN=IP-4
          DO 691 IFM=1,2
          BF1(2049,IFM)=2048
          BF2(2049,IFM)=2048
          CALL WR4LN(0,BF1(2049,IFM))
          CALL FDTOA(IFM,ILN,0)
          CALL WR4LN(0,BF2(2049,IFM))
      691 CALL FDTOA(IFM,ILN-4,0)
          MT=M1
          M1=M2
```

Write Title And Frame Number on the Processed Picture

ORIGINAL PAGE IS
OF POOR QUALITY

```
C-ERRS...STNO.C..... F O R T R A N   S O U R C E   S T A T E M E N T S ........ IDENTFCN    **COMPILER ME

              M2=MT
              MT=M3
              M3=M4
              M4=MT
              MT=M3-1
              CALL MOVE(0,1,MT)
              CALL MOVE(1,1,MT)
          808 CONTINUE
              CALL MOVE(0,0,2)
              WRITE(1,444)
              CALL EXIT
          100 FORMAT(8I5)
          200 FORMAT (5X,8I10)
          250 FORMAT(8I10)
          333 FORMAT ('****')
          444 FORMAT('*END PROCESSING*')
              END
VARIABLE ALLOCATIONS
   JD1(I*2  )=105D-001C    BF1(I*2  )=105D-005C    JD2(I*2  )=209F-105E    BF2(I*2  )=209F-109E    QL2(I*2
  IQL2(I*2  )=20E1-20C2     Q1(I*2  )=20E1-20E2    IQ1(I*2  )=2101-20F2     Q3(I*2  )=210A-2102    IQ3(I*2
    Q4(I*2  )=211A-2114    IQ4(I*2  )=2121-211B     Q5(I*2  )=2126-2122    IQ5(I*2  )=212B-2127    NZX(I*2
    DQ(I*2  )=2197-2168    IDQ(I*2  )=21C7-2198   CHAR(I*2  )=22C7-21C8  DIGIT(I*2  )=22CA-22C8   FNUM(I*2
   EOF(I*2  )=22CD         BOF(I*2  )=22CE           K(I*2  )=22CF         IEG(I*2  )=22D0         NLG(I*2
  NFRM(I*2  )=22D2          M1(I*2  )=22D3          M2(I*2  )=22D4          M3(I*2  )=22D5          M4(I*2
     I(I*2  )=22D7           J(I*2  )=22D8          IX(I*2  )=22D9        IFRM(I*2  )=22DA        IDFX(I*2
   ILN(I*2  )=22DC         IRF(I*2  )=22DD         ILG(I*2  )=22DE          IP(I*2  )=22DF          IY(I*2
   IDK(I*2  )=22E1          MT(I*2  )=22E2         IFM(I*2  )=22E3         IGP(I*2  )=22E4        LEV2(I*2
   IRX(I*2  )=22E6        LEV1(I*2  )=22E7

STATEMENT ALLOCATIONS
  100=2303    200=2306    250=230A    333=230D    444=2311    396=236A    395=237C    397=238C    588=23BD
  413=2436    411=2490    194=24CC    197=2502    356=253A    355=254B    248=256C    251=2572    258=257D
  259=258D    347=2618    337=279D    886=27EF    805=286A    269=2873    991=2885    992=2893    800=289F
  984=28EA    986=28F8    985=2904    382=290E    381=2972    757=29D4    797=29F0    681=2A0C    771=2A48
  773=2A6C    466=2A87    865=2B0E    864=2B1D    682=2B35    691=2B78    808=2BB4

FEATURES SUPPORTED
```

Go to Next Frame

ORIGINAL PAGE IS
OF POOR QUALITY