IN-32
77727
P-17

NASA Technical Memorandum 105616

Real-Time Demonstration Hardware for Enhanced DPCM Video Compression Algorithm

Thomas P. Bizon and Wayne A. Whyte, Jr.
Lewis Research Center
Cleveland, Ohio

and

Vincent R. Marcopoli
Case Western Reserve University
Cleveland, Ohio

Prepared for the
Data Compression Conference (DCC '92)
sponsored by the Institute of Electrical and Electronics Engineers
Snowbird, Utah, March 24–26, 1992

NASA

Real-Time Demonstration Hardware for Enhanced DPCM Video Compression Algorithm

Thomas P. Bizon and Wayne A. Whyte, Jr.
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Vincent R. Marcopoli
Case Western Reserve University
Cleveland, Ohio 44106

1.0 Introduction

Transmission of television signals in a digital format has been looked upon with promise for a number of years. Digital systems providing teleconferencing quality video have become commonplace in both government and industry; however, digital transmission of high-quality (toll grade or broadcast quality) television signals has yet to achieve anything close to the same kind of acceptance. This has been due in part to the broadcasters' reluctance to have processing of any kind performed on the transmitted signals, but to a greater extent, digital transmission of broadcast quality video has failed to gain acceptance because it has not been cost effective to do so. The lack of available wideband digital links as well as the complexity of implementation of bandwidth efficient digital video CODECs (encoder/decoders) has worked to keep the cost of digital television transmission too high to compete with analog methods. Terrestrial and satellite video service providers, however, are now recognizing the potential gains that digital video compression offers and are proposing to incorporate compression systems to increase the number of available program channels. NASA is similarly recognizing the benefits of and trend toward digital video compression techniques for transmission of high quality video from space.

Advances in very large-scale integration (VLSI) as well as recent work in the field of advanced digital modulation techniques have combined to make digital video processing technically feasible and potentially cost competitive for broadcast quality television transmission. The coupling of a transparent, bandwidth efficient data compression technique with a bandwidth efficient modulation technique offer the potential for transmission of multiple high-quality television signals in the same bandwidth occupied by a single frequency modulated television signal. This paper presents hardware used for a real-time demonstration of a digital television bandwidth compression algorithm which processes standard NTSC (National Television Systems Committee) composite color television signals and produces broadcast quality video at an average of 1.8 bits/pixel. The algorithm is based on differential pulse code modulation (DPCM), but additionally utilizes a non-adaptive predictor, non-uniform quantizer and multilevel Huffman coder to reduce the data rate substantially below that achievable with straight DPCM. The non-adaptive predictor and multilevel Huffman coder combine to set this technique apart from other DPCM encoding algorithms. This CODEC could be used to compress composite NTSC color television signals for transmission direct-to-the-home via direct broadcast satellite systems. Broadcast quality reconstructed video is achieved with minimal hardware complexity.

2.0 Enhanced DPCM Video Compression Algorithm

The Enhanced DPCM video compression algorithm was designed for compression and transmission of composite NTSC video signals. The algorithm incorporates only spatial processing on a field basis to minimize memory requirements and to simplify implementation. Differential pulse code modulation (DPCM) was chosen as the basis of the algorithm for its overall performance characteristics and for simplicity of implementation. To improve upon the basic DPCM performance, two enhancements were developed, a non-adaptive predictor for edge preservation and a multi-level Huffman coding technique for data rate reduction. Figure 1 presents a top level diagram of the steady state operation of the compression algorithm. (Section 3.0 will discuss transitional considerations associated with the beginnings of lines and fields). Figure 1 does not show the functions associated with analog-to-digital and digital-to-analog conversion (ADC and DAC). These functions will be addressed later in the section on the hardware implementation, as they do not explicitly pertain to the operation of the compression algorithm.

DPCM is a predictive compression technique in which a prediction of the picture element (pixel) being encoded is subtracted from the incoming pixel to create a difference value which is then quantized for subsequent transmission. For the Enhanced DPCM algorithm, a two-dimensional prediction is made using two previously encoded pixels having the same color subcarrier phase relationship as the current pixel being encoded. Since both luminance and chrominance information is present in every sample of the composite signal, samples having the same phase relationship tend to be more highly correlated. With sampling of the video signal done at four times the color subcarrier frequency (4 * 3.579545 MHz), the pixel four samples previous on the same line as the current pixel and the same horizontally-spaced pixel from two lines previous will be in phase with the current pixel, and are therefore averaged to form the basis of the DPCM prediction. In a conventional DPCM scheme the resulting difference value would be quantized and the quantization level transmitted to the decoder; however at this point, one of the performance enhancements mentioned previously is incorporated in the algorithm--the non-adaptive predictor (NAP).

For relatively static portions of the image field, the two pixel average provides a good prediction of the pixel being encoded. However, at transition boundaries (edges in the image), the difference values produced can become large due to poor predictions across the edge gradient. The large difference values would result in greater quantization error due to the non-uniform nature of the quantizer. Because the prediction is based in part on the fourth previous pixel, the conventional DPCM scheme would result in slightly blurred edges due to the higher quantization noise from the less accurate predictions. The non-adaptive predictor allows rapid recovery at transitions by compensating for the inaccuracies which result for predictions across transition boundaries. The development of the NAP was based on the likelihood that adjacent pixels would have similar difference values. That is, if pixel A were the first pixel to cross a transition boundary resulting in a large difference value, the next pixel, B, would be likely to have a difference value very close to that for pixel A. The NAP is used to improve the prediction for pixel B by subtracting a value from its prediction which approximates the difference value obtained for pixel A. In this way, good predictions can be reestablished after only one pixel following a transition as opposed to five pixels (based on fourth previous pixel predictions). The result is a subjectively cleaner edge because the quantization noise is small for small difference values. For relatively constant portions of the image, where the prediction is good and difference values are correspondingly small, the NAP value that is subtracted from the difference is zero or a small value corresponding to the size of the difference value of the previously encoded pixel. The NAP is non-adaptive because the estimates are prestored and do not change with differing picture content. These prestored values were generated from statistics of numerous television images covering a wide range of picture content. The NAP values represent the average difference values calculated within the boundaries of the difference values for each quantization level. Table 1 shows the NAP values corresponding to each quantization level. For example, using the values in table 1; if the difference value for the previous pixel was 40, corresponding to quantization level 11, the value of NAP to be subtracted off from the current pixel difference would be 38. To reconstruct the pixel, the decoder uses a lookup table to add back the appropriate NAP value based upon knowledge of the quantization level from the previously decoded pixel.

Table 1 also provides the range of difference values associated with each quantization level (QL) and the corresponding quantization value (QV) which is used in reconstructing the pixel values. The quantizer consists of thirteen (13) non-uniform levels. The quantizer is non-uniform so that more levels are provided for small magnitude differences which result from subtle changes in picture content. The human eye is sensitive to small variations in smooth regions of an image but can tolerate larger variations near transition boundaries where large difference values are more likely to occur. The non-adaptive predictor discussed previously acts to reduce the difference values thus improving image quality by reducing the quantization error. This is because the non-uniform quantizer results in lower quantization error for small magnitude differences than for large magnitude differences. The particular difference value ranges and corresponding quantization values were experimentally derived via computer simulation.

The second major enhancement of the encoding algorithm is multiple level Huffman coding. The output of the DPCM portion of the encoder is a four bit word corresponding to one of thirteen quantization levels for each pixel. These four bit values are then entropy coded to further reduce the amount of data for required transmission. Huffman coding is a lossless process (i.e. fully recoverable) by which shorter codewords are assigned to events having the highest probability of occurrence and longer codewords are assigned to events having the least probability of occurrence. Each codeword is uniquely decodable from all other codewords in the Huffman tree. Every quantization level is assigned a unique Huffman codeword based upon its probability of occurrence across a range of representative video images. A separate set of Huffman codes was generated for each of the thirteen quantization levels. The matrix of code sets is used to reduce the number of data bits required to transmit a given pixel. The particular Huffman code set used for a given quantized pixel is determined by the quantization level of the previous pixel (i.e. if the difference value for the previous pixel resulted in quantization level 4 being selected for that pixel, then the Huffman code set selected for the current pixel would be code set 4, corresponding to the probability of occurrence of pixels falling into the fourth quantization level. This code set would then be constructed so that the shortest codewords are near quantization level 4.). The multiple Huffman code sets were designed to take advantage of the tendency for neighboring pixels to fall into the same or close to the same quantization level. Use of the multilevel Huffman code sets provides a means for significant reductions in the average number of bits per pixel over a single Huffman code set because they allow more pixels to be represented by short length codewords. Simulation results have shown an average reduction of 0.5 bits per pixel for the multilevel code sets compared to a single code set.

Due to the variability of codeword length, use of Huffman coding presents a complication to the transmission process which must be overcome. Data enters the Huffman coder at a constant rate, however, data leaving the Huffman coder is at a variable rate because some pixels are represented by only one bit while other pixels may be represented by as many as eleven bits. This variability cannot be accommodated on a standard communication channel, and therefore the data must be buffered so that transmission can be at a constant serial rate. This buffering is accomplished in practice using a first-in first-out (FIFO) memory. The complication comes from the need to prevent both overflow and underflow of the buffer memory. In order to prevent overflow, the data rate into the FIFO must be reduced. The data reduction technique used here is to process incoming difference values using coarser quantization than that of the original algorithm. The new quantization levels are constructed from the original by simply grouping various levels together as one level, yielding a coarser quantization scheme which can then use a subset of the Huffman codes sets of the original algorithm. Specifically, the groupings chosen are shown in table 2, yielding a five-level quantization scheme having high compression capability while maintaining reasonable picture quality. The groupings represent the combination of quantization levels from the original algorithm into levels which cover a wider range of difference values. Such a scheme minimizes the impact on hardware implementation. Underflow is dealt with by increasing the data rate into the FIFO by simply disabling the compression effect of the Huffman encoder and transmitting at 4 bits/pixel. This can be accomplished by either transmitting the DPCM output directly, or using a Huffman code set whose word lengths are all 4 bits. Note that image quality during underflow is identical to that of the original algorithm, since no data is lost. The FIFO control algorithm operates as follows: If the FIFO depth is above a threshold, called the "full" level, data reduction will be enabled until the FIFO level drops below the full level. Similarly, if the FIFO level is below a threshold, called the "empty" level, data is transmitted at 4 bits/pixel until the FIFO level rises above the empty level. When the FIFO level is between the full and empty levels, processing takes place according to the original algorithm. The serial transmission rate is then chosen such that overflow recovery is minimized to maintain superior image quality across a broad range of picture content. At the same time, the transmission rate should not be set too high so that underflow recovery is also minimized, as this is spectrally inefficient. A transmission rate corresponding to 1.8 bits per pixel has been determined via simulation to represent a good compromise between quality and data rate.

3.0 Practical Considerations to Accommodate NTSC Signals

NTSC composite color television signals transmit thirty frames (screens) of video information per second, with a single frame subdivided into two interlaced fields of 262.5 video lines each. The first nine lines of each of these fields are used for vertical synchronization of the monitor, while the next eleven are used for transmission of a variety of test patterns and data. The remaining lines contain active video information. Each video line (except those in the vertical synch period) includes a horizontal synch pulse and reference color burst for monitor synchronization, followed by the information being transmitted. A diagram of the standard NTSC video signal is shown in figure 2.

In the initial development and computer simulations of the Enhanced DPCM algorithm, it was assumed that only active video would be processed. In the actual implementation, however, the entire NTSC signal is sampled and processed; therefore, it was necessary to modify the original algorithm to avoid as much inefficient processing of video data as possible. Inefficient processing occurs when data uncorrelated with the "current" pixel is used in its prediction, resulting in both a reduction in picture quality and an increase in the amount of compressed video data. As stated previously, pixel predictions used in the DPCM portion of the algorithm are based on the average value of the pixels two lines above and four pixels to the left of the "current" pixel. At various times throughout any given frame, one or both of these values could be uncorrelated with the "current" pixel, leading to an inaccurate prediction (and inefficient processing). For example, trying to predict a pixel at the top of a field using information from the bottom of the previous field will lead to a poor prediction. Similarly, trying to predict a pixel at the beginning of a line using information from the end of the previous line will also lead to a poor prediction. In order to optimize performance of the algorithm, several modifications were made to remove uncorrelated data from "current" pixel predictions throughout each field. A complete description of the Enhanced DPCM video data compression (VDC) algorithm with all modifications is given in figure 3.

Another factor to consider in the implementation of the algorithm was the handling of the half-line in each field. The position of the half-line differs between the fields, with the line being placed at the end of field 1 containing active video and its counterpart being placed at the beginning of field 2 starting the vertical synching period. As with the other data, these lines should be processed in a manner to eliminate as much inefficient processing as possible, while not excessively complicating the algorithm. The description of the Enhanced DPCM VDC algorithm in figure 3 includes processing of the half-lines.

4.0 Demonstration Hardware

As described previously, the Enhanced DPCM VDC algorithm utilizes both lossy (quantized DPCM) and lossless (Huffman coding) compression techniques. Since data loss occurs only during DPCM processing, just that portion needs to be implemented in order to demonstrate the observable effects of the entire compression algorithm on image quality. Referring again to figure 1, the "post-DPCM" RP value stored in the FIFO on the encoder side is identical to the output RP value on the decoder side assuming no channel errors. Taking the above into account, hardware requirements for a real-time demonstration of the Enhanced DPCM algorithm are reduced to an NTSC video sampling board and the encoder-side DPCM quantizer board. These boards were designed, implemented, and tested. A detailed description of each can be found in the following sections.

4.1 Video Sampling Board

The video sampling board performs sampling and reconstruction of an NTSC video signal, generates a phase-locked sample clock, and creates any necessary control signals for synchronization with the quantizer board. Although the sampling and reconstruction circuits are housed on the same board, the integrity of a "real" communications system is maintained because these circuits operate independently and are therefore physically separable. A block diagram of the board can be found in figure 4.

Video conversion is accomplished using an analog-to-digital converter (ADC) preceded by an op-amp circuit to buffer, offset, and amplify the incoming signal. Once converted, the digital video from the ADC is buffered and transmitted to the DPCM quantizer board where compression processing begins. Conversion rate is fixed using a phase-locked loop circuit to maintain the sample clock at a frequency of four times the color subcarrier input. This clock not only controls the ADC, but also acts as the system clock for the quantizer board (and eventually the entire encoder). Additionally, a sync separator circuit is used to create various control signals to synchronize the quantizer circuit with the incoming video signal.

The reconstruction circuit is a mirror image of the video conversion section of the sampling circuit. The decompressed digital video is latched, passed to a digital-to-analog converter (DAC), and sent through an op-amp circuit to undo the previous amplification and offset. Reconstruction rate is controlled with the reconstruction clock, a phase-locked clock to be supplied by the decoder hardware. Since the demonstration hardware does not require implementation of the decoder, the reconstruction clock is temporarily derived from the quantizer board system clock.

4.2 DPCM Quantizer Board

The DPCM quantizer board performs all necessary functions to demonstrate the processed video quality of the Enhanced DPCM VDC algorithm. As shown in figure 5, this board consists of three subcomponents--the DPCM quantizer, the quantizer controller, and a start circuit. A description of each sub-unit can be found in the following paragraphs.

The quantizer circuit performs all algebraic computations to predict the incoming pixel value, calculate the difference between the prediction and the actual value, and quantize that difference. In order to meet "real-world" timing constraints, calculations are performed using a pipelined architecture, where complex computations are subdivided into a number of smaller operations with several elements processing simultaneously. A functional block diagram of the quantizer can be found in figure 6. As is typical of a pipelined structure, all operations are isolated with registers (denoted as Z^{-1} to represent a digital delay). The registers are synchronized with the sampling clock; therefore, the time allocated to perform each operation is just under 70 nsec (the sampling period).

Processing begins with a two-stage summation of the incoming pixel value (PIX), the "negative" predicted value (PV) of that pixel, and its negative non-adaptive prediction (NAP) to yield the difference value (DIF). Using a look-up table addressed with DIF, the difference value is quantized, a quantization level (QL) is assigned, and the negative non-adaptive prediction for the following pixel is generated. The QL is transmitted to the Huffman coder for further compression, while the quantized value (QV) is added to the predicted value and non-adaptive prediction to produce a quantized version of the original pixel--the reconstructed pixel (RP). A hard-limiter is used to prevent overflow and underflow of this value. Once RP is computed, the prediction of the incoming pixel is calculated by averaging that value (already delayed by three samples) and the RP value output from a two-line delay FIFO memory. The result (PV) is inverted and looped back to be added with the incoming pixel.

In order to implement initializations and "special cases" of the Enhanced DPCM VDC algorithm, the quantizer circuit contains four multiplexers to route the desired data to various devices. MUX1 allows transmission of the incoming pixel uncompressed, MUX3 initializes the NAP value to '0', and MUX2/4 determine the pixel prediction method (four pixels previous, two lines previous, or both averaged). These muxes are governed by the quantizer controller circuit, consisting of three programmable logic devices (PLDs) and three digital counters. All devices are synchronized to implement a state machine that generates the multiplexer control signals for proper data routing to execute the algorithm (see figure 3). Due to size limitations, the state machine is divided into three subsections, with each subsection programmed in a single PLD. The counters are included to update and report the location of the incoming pixel within a given field.

An underlying premise of the preceding discussion is that the algorithm is aligned with the real-time video data such that processing begins at the first pixel of a new frame. This synchronization is performed by the start circuit, consisting of two PLDs and two programmable digital counters. These devices cooperate to implement a state machine which detects the top of a video frame and issues the necessary initialization commands to properly start the quantizer and controller. Due to timing constraints, the state machine is divided into two subsections--each one operating one-half clock cycle opposite the other--thus allowing initialization commands to be transmitted on both the rising and falling edges of the sampling clock. The programmable counters are included to allow a user to easily vary the "first pixel" of a frame for experimental purposes.

5.0 Performance Results

Assessment of algorithm performance has been completed in two ways--a computer simulation of the algorithm using single frames of video information and the real-time demonstration of the algorithm described in this paper. Results of these experiments are summarized in the following paragraphs.

In single-frame simulation experiments, a "toggle comparison" between original (unprocessed) and computer-compressed images confirms there are no visible differences between the images. Calculations have shown that data requirements to transmit video information have decreased from 8 bits per pixel (BPP) to an average of 1.8 BPP. In any given frame, however, the amount of compression will vary depending on the complexity of the picture due to variable length codewords being used to represent each pixel. To illustrate this, table 3 shows simulated compression results for several images with transmission requirements ranging from a 2.2 BPP average in the complicated beach scene to a 1.3 BPP average in the relatively simple color bars. Simulations have shown that these variations can be smoothed to a constant transmission rate using the data rate buffer (previously described) with minimal observable impact on the image quality.

For real-time processing experiments, there are two types of video pictures which must be investigated--a stationary NTSC video image (eg. color bars) and actual motion video. When using still video, a slight degradation is observed between the original and processed images due to random noise in the incoming pixels. These "noise errors" occur randomly throughout each frame, thus creating a background motion effect in the stationary picture. When using moving video, the same degradation is not as evident because the background motion effect is concealed with movement in the picture; however, the noise effects are still apparent in large stationary areas of the picture (for example, a blank wall behind the central action). The "noise errors" are induced in the wire-wrapped sampling board; therefore, this degradation should vanish with the upgrade of this board (as described in the next section). Neglecting the noise problem, original and processed images are subjectively indistinguishable for both stationary NTSC video frames and motion video. Motion degradation, a concern with several video processing algorithms, is non-existent in this case since all pixel predictions are made using pixels in the same field.

6.0 Continuing Work

In order to accomplish the goal of a real-time compressed video transmission, development of the additional hardware necessary to implement the full algorithm--the Huffman encoder/decoder pair, the data rate buffer, and the DPCM decoder--has begun. Before implementation of these components can proceed, methods must be developed to perform functions such as synchronizing the decoder (and decoder clock) with the video data and detecting/concealing transmission errors. Resolution of the above issues will be discussed in a follow-up paper describing the completed video processor.

Separate work progressing in parallel with the above is the upgrade of the video sampling board to add features and improve performance. Planned feature additions include a dc-restore circuit to prevent clipping of ac-coupled input signals and a phase-locked loop synchronized with the input video signal as opposed to an external color subcarrier input. The redesigned board will be converted from wire-wrap to pc-board layout, paying precise attention to analog design conventions. Once completed, the upgraded video sampling board will allow input from a variety of sources (both ac and dc-coupled), while reducing the amount of noise added to the input signal. This noise reduction will result in the elimination of the "noise errors" described above.

7.0 Conclusion

A video data compression algorithm has been developed at NASA Lewis Research Center which reduces digital transmission requirements for composite color NTSC video signals from 8 BPP to 1.8 BPP. Computer simulations of the algorithm using a single frame of video information have shown that original and processed images are subjectively identical; therefore, a video processor to demonstrate algorithm performance in real time has been developed. Experiments conducted using this hardware to process full-rate, full-motion video have also shown favorable results; hence, hardware development of the complete algorithm has been initiated. Once completed, the video processor will be used for a real-time compressed video transmission utilizing the Advanced Communications Technology Satellite (scheduled for launch in early 1993).

Table 1: Quantization and Non-Adaptive Prediction

DIF Range	QL	QV	NAP
-255 to -86	1	-100	-85
-85 to -60	2	-66	-61
-59 to -34	3	-42	-38
-33 to -19	4	-25	-22
-18 to -9	5	-14	-11
-8 to -4	6	-6	-4
-3 to 3	7	0	0
4 to 8	8	6	4
9 to 18	9	14	11
19 to 33	10	25	21
34 to 59	11	42	38
60 to 85	12	66	61
86 to 255	13	100	84

Table 2: Coarse Quantization Scheme

Groupings	DIF Range	QL	QV	NAP
1 2 3	-255 to -34	5	-42	-38
4 5	-33 to -9	6	-14	-11
6 7 8	-8 to 8	7	0	0
9 10	9 to 33	8	14	11
11 12 13	34 to 255	9	42	38

Table 3: Algorithm Performance Results

Scene	BPP
Beach	2.228
Make-up	1.738
Lawyer	1.976
Star Trek	1.682
Bees and flowers	1.996
Game/girl	1.689
Woman/couch	1.813
Hospital	1.979
Woman/man	1.802
Woman/gray suit	1.872
Two men	1.949
Man/phone	1.815
Fuller brush	2.037
Plane/biplane	1.711
Woman/hand	1.671
News woman	1.823
Color Bars	1.347
Two women	1.659
Average	**1.822**

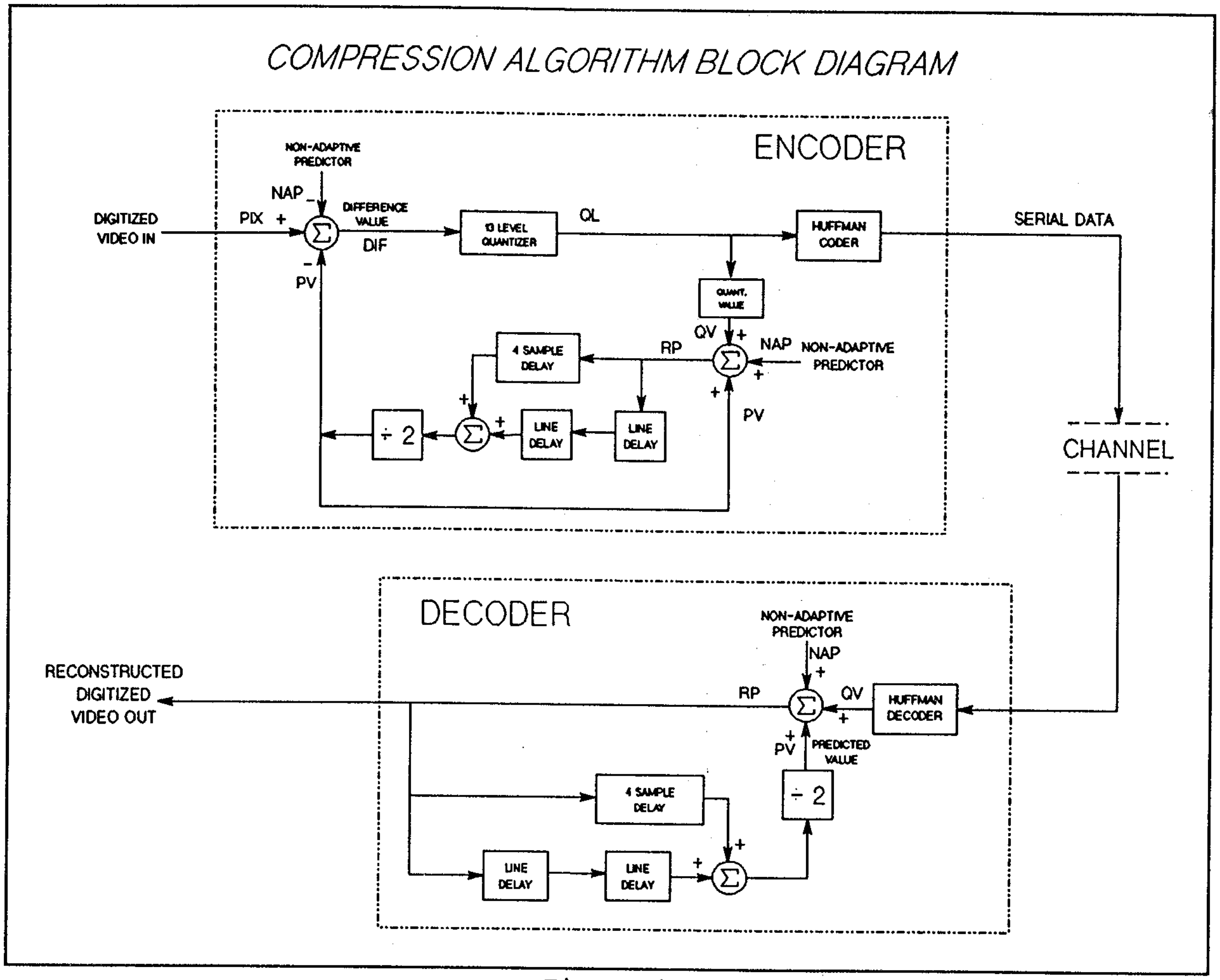

Figure 1

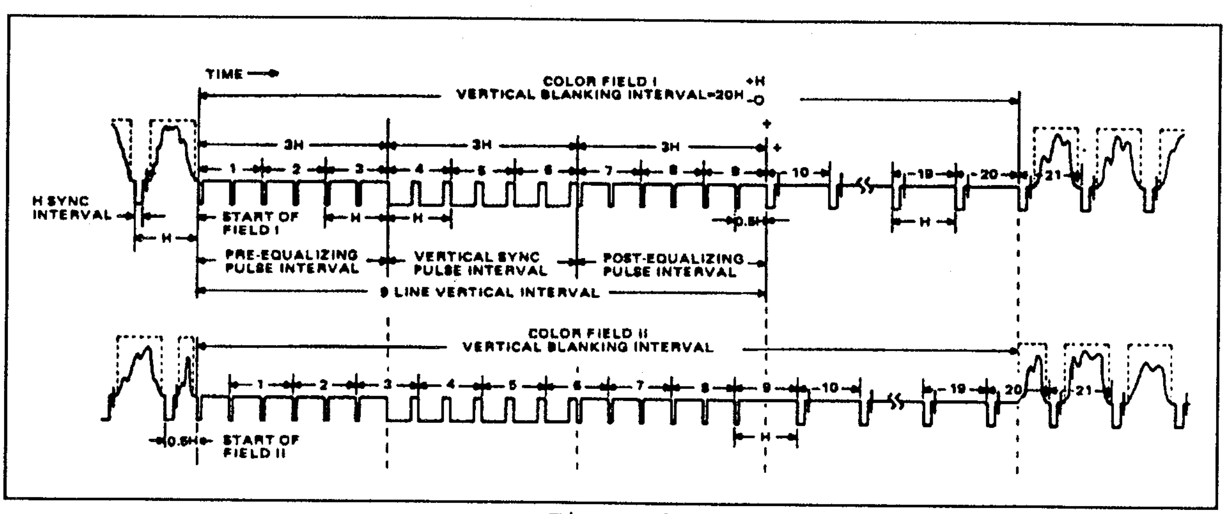

Figure 2

Enhanced-DPCM Video Compression Algorithm

BEGIN FIELD 1:

1) Transmit the first four pixels of the new line uncompressed.
2) Compress the remainder of the line basing prediction only on the sample four pixels previous.
3) Repeat (1) and (2) for lines 1-2.
4) Compress first four pixels of the new line basing prediction only on the sample two lines previous.
5) Compress the remainder of the line basing prediction only on the sample four pixels previous.
6) Repeat (4) and (5) for lines 3-22.
7) Compress first four pixels of the new line basing prediction only on the sample two lines previous.
8) Compress the remainder of the line basing prediction on both the samples four pixels previous AND two lines previous.
9) Repeat (7) and (8) for the remainder of the field (including the final half-line).

END FIELD 1.

BEGIN FIELD 2:

10) Transmit the first four pixels of the new line uncompressed.
11) Compress the remainder of the line basing prediction only on the sample four pixels previous.
12) Repeat (10) and (11) for the initial half-line and lines 1-2.
13) Compress first four pixels of the new line basing prediction only on the sample two lines previous.
14) Compress the remainder of the line basing prediction only on the sample four pixels previous.
15) Repeat (13) and (14) for lines 3-22.
16) Compress first four pixels of the new line basing prediction only on the sample two lines previous.
17) Compress the remainder of the line basing prediction on both the samples four pixels previous AND two lines previous.
18) Repeat (16) and (17) for the remainder of the field.

END FIELD 2.

BEGIN FIELD 1:

19) GoTo (1).

Figure 3

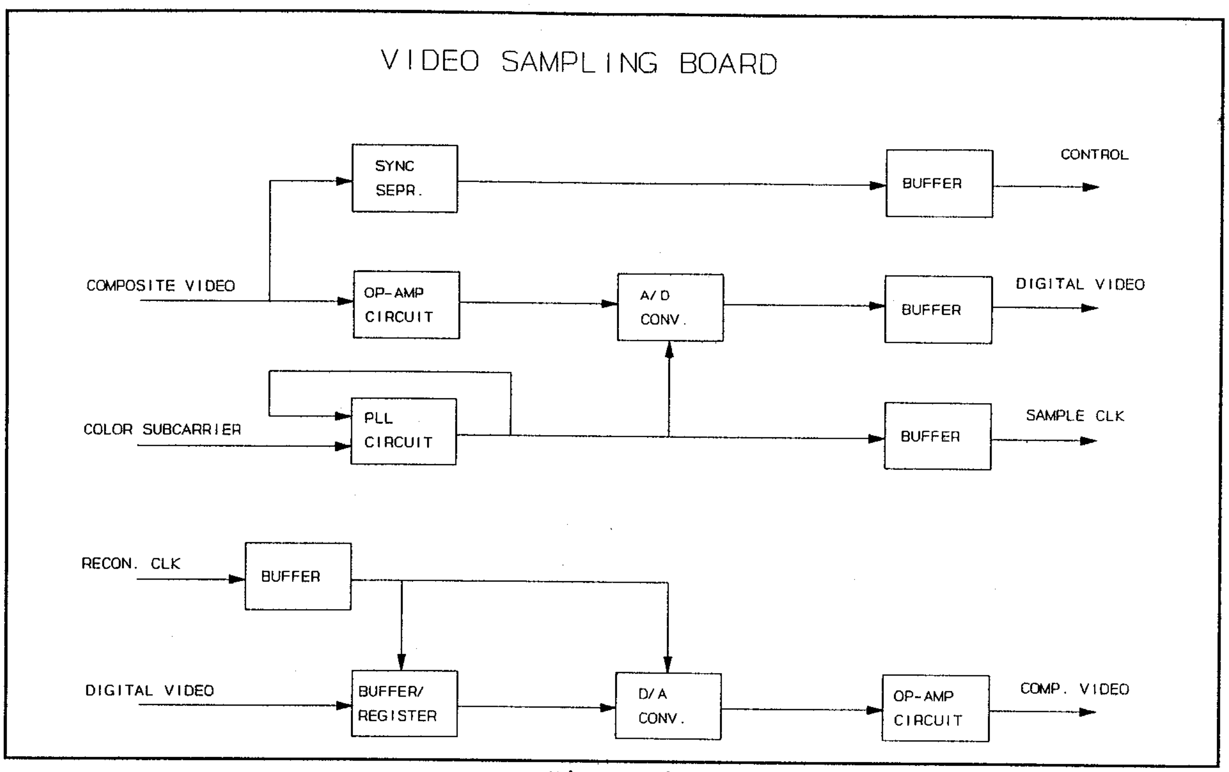

Figure 4

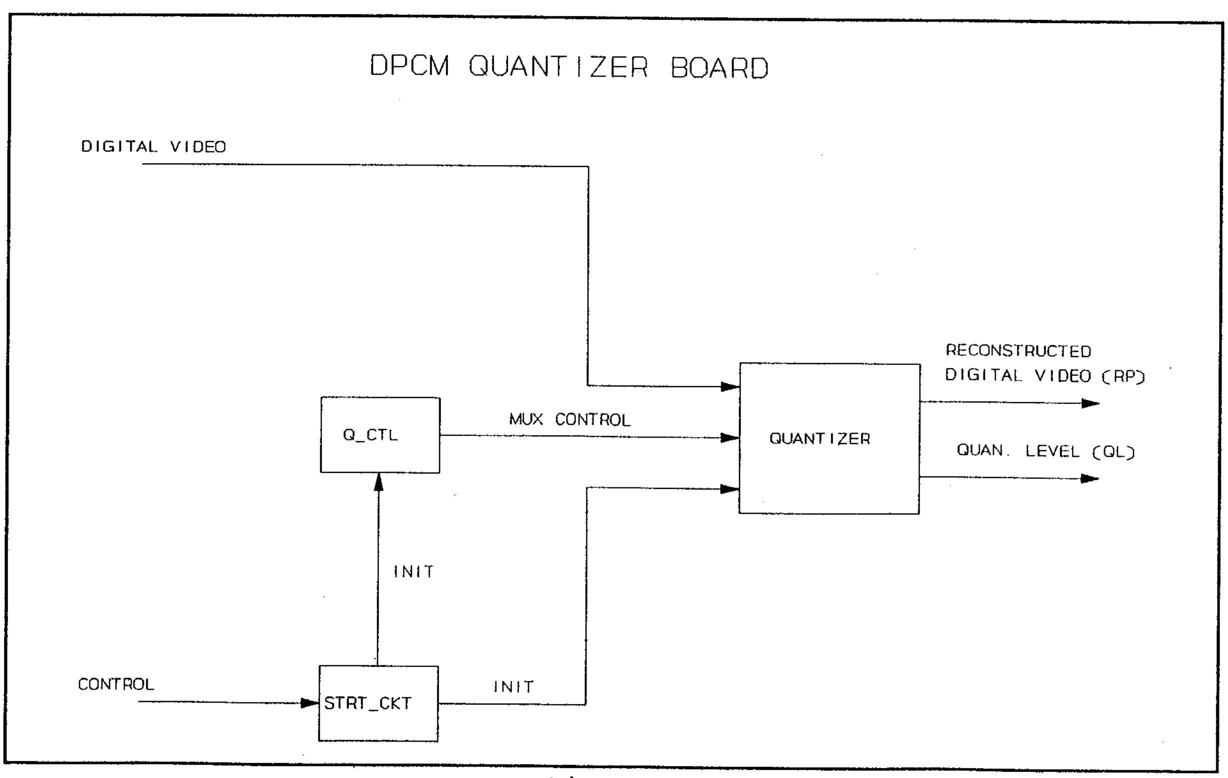

Figure 5

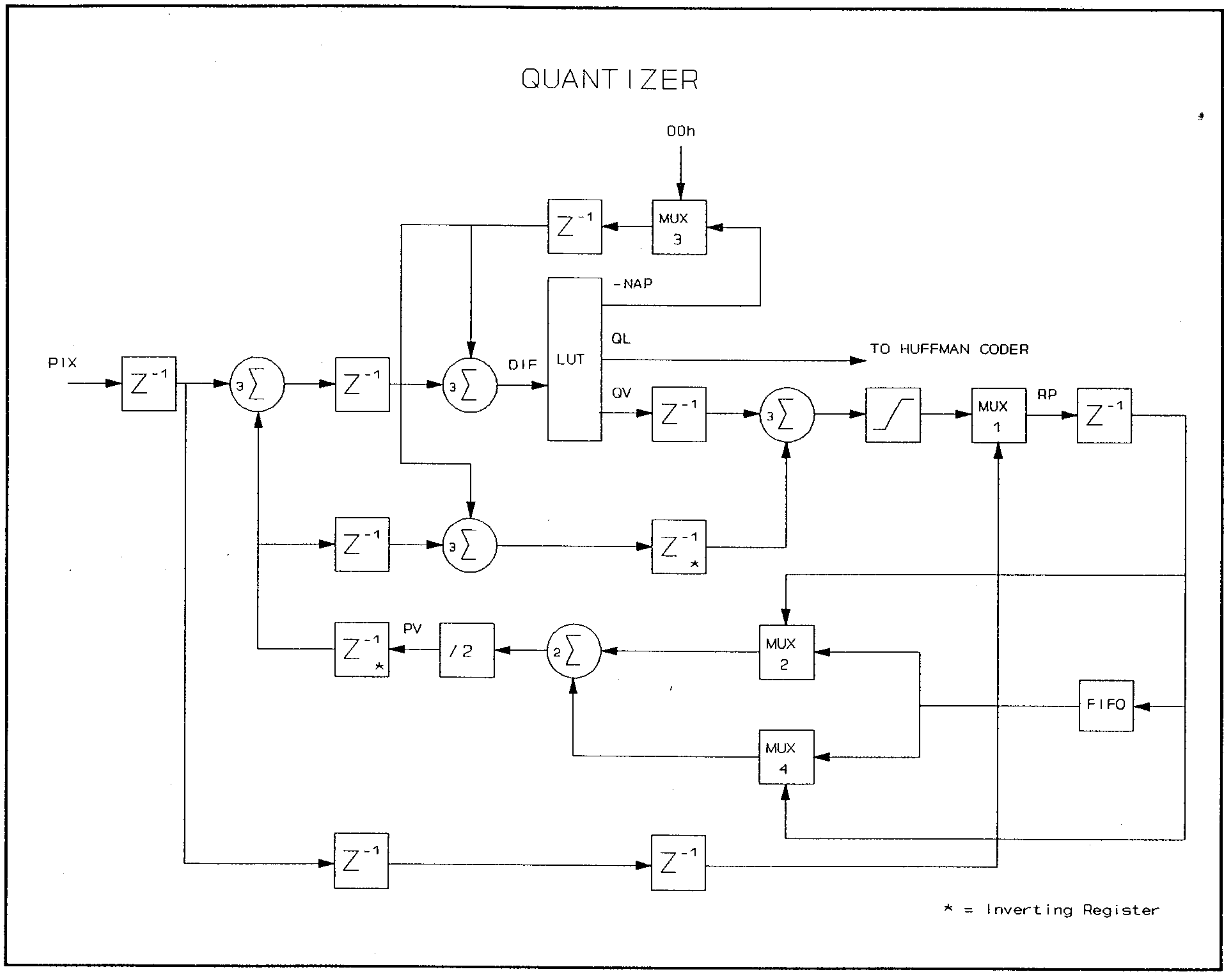
QUANTIZER
00h
MUX 3
-NAP
LUT
QL
TO HUFFMAN CODER
PIX
DIF
QV
MUX 1
RP
PV
/2
MUX 2
FIFO
MUX 4
* = Inverting Register

Figure 6

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
1992

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Real-Time Demonstration Hardware for Enhanced DPCM Video Compression Algorithm

5. FUNDING NUMBERS
WU-144-10-10

6. AUTHOR(S)
Thomas P. Bizon, Wayne A. Whyte, Jr., and Vincent R. Marcopoli

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E-6954

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, D.C. 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM-105616

11. SUPPLEMENTARY NOTES
Prepared for the Data Compression Conference (DCC '92) sponsored by the Institute of Electrical and Electronics Engineers, Snowbird, Utah, March 24-26, 1992. Responsible person, Thomas P. Bizon, (216) 433-8121.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 32

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The lack of available wideband digital links as well as the complexity of implementation of bandwidth efficient digital video CODECs (encoder/decoder) has worked to keep the cost of digital television transmission too high to compete with analog methods. Terrestrial and satellite video service providers, however, are now recognizing the potential gains that digital video compression offers and are proposing to incorporate compression systems to increase the number of available program channels. NASA is similarly recognizing the benefits of and trend toward digital video compression techniques for transmission of high quality video from space and therefore, has developed a digital television bandwidth compression algorithm to process standard NTSC (National Television Systems Committee) composite color television signals. The algorithm is based on differential pulse code modulation (DCPM), but additionally utilizes a non-adaptive predictor, non-uniform quantizer and multilevel Huffman coder to reduce the data rate substantially below the achievable with straight DPCM. The non-adaptive predictor and multilevel Huffman coder combine to set this technique apart from other DPCM encoding algorithms. All processing is done on an intra-field basis to prevent motion degradation and minimize hardware complexity. Computer simulations have shown the algorithm will produce broadcast quality reconstructed video at an average transmission rate of 1.8 bits/pixel. Hardware implementation of the DPCM circuit, non-adaptive predictor and non-uniform quantizer has been completed providing real-time demonstration of the image quality at full video rates. Video sampling/reconstruction circuits have also been constructed to accomplish the analog video processing necessary for the real-time demonstration. Performance results for the completed hardware compare favorably with simulation results. Hardware implementation of the multilevel Huffman encoder/decoder is currently under development along with implementation of a buffer control algorithm to accommodate the variable data rate output of the multilevel Huffman encoder. A video CODEC of this type could be used to compress composite NTSC color television signals where high quality reconstruction is desirable (e.g. Space Station video transmission, transmission direct-to-the-home via direct broadcast satellite systems or cable television distribution to system headends and direct-to-the-home).

14. SUBJECT TERMS
Data compression; DPCM; Video; Huffman coding; Digital television

15. NUMBER OF PAGES
16

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	